

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри, канд. техн. наук,

доцент _____ Є. О. Давиденко

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ
ДИСТАНЦІЙНОГО НАВЧАННЯ ЗВО

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21910906

Студентка

_____ А. В. Гайсюк

«___» _____ 2023 р.

Керівник: завідувач кафедри ІІЗ,
канд. техн. наук, доцент

_____ Є. О. Давиденко

«___» _____ 2023 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексєєва

«___» _____ 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії програмного
забезпечення, канд. техн. наук, доцент
_____ Є. О. Давиденко
« ____ » _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано студентці групи 409 факультету комп'ютерних наук

Гайсюк Аліні Валеріївні.

1. Тема кваліфікаційної роботи «Програмне забезпечення системи дистанційного навчання ЗВО».

Керівник роботи Давиденко Євген Олександрович, канд. техн. наук, доцент.

Затверджена наказом по ЧНУ ім. Петра Могили від «17» березня 2023 р. № 60

2. Строк представлення кваліфікаційної роботи «27» червня 2023 р.

3. Початкові дані роботи: концепція, зміст і напрямок розвитку системи дистанційного навчання, цілі та завдання, які вона повинна виконувати; навчальні матеріали, які викладачі будуть використовувати для проведення онлайн-курсів, та методична підтримка для викладачів і студентів; технічне обладнання: комп'ютери, сервери, мережеве обладнання, програмне забезпечення та інші матеріально-технічні засоби, необхідні для роботи системи.

Очікуваний результат: розроблена СДН (Система дистанційного навчання) для широкого кола студентів, що не мають можливості навчатися на місці, зокрема для студентів з інших міст або країн, підвищення

інноваційності та технологічності університетів, оскільки впровадження системи дистанційного навчання вимагає використання новітніх технологій та інфраструктури та розширення навчальної програми і підвищення якості освіти, дистанційна форма навчання дозволяє студентам вивчати матеріал у відповідності зі своїми індивідуальними потребами та ритмом.

4. Перелік питань, що підлягають розробці:

- аналіз сучасних тенденцій розвитку систем дистанційного навчання в світі та в Україні;
- визначення основних функціональних вимог до систем дистанційного навчання та їх застосування в університетах;
- аналіз існуючих систем дистанційного навчання;
- розробка рекомендації щодо подальшого вдосконалення системи дистанційного навчання в університеті;
- проведення тестування СДН;
- аналіз результатів розробки СДН.

5. Перелік графічних матеріалів: _____ презентація. _____

6. Завдання до спеціальної частини: «Вивчення особливостей робочого місця користувача системи дистанційного навчання ЗВО та розробка рекомендацій щодо його організації з точки зору безпеки та здоров'я».

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
канд. техн. наук, доцент Алексеева А. О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи _____ канд. техн. наук, доцент Давиденко Є. О.
(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання _____ Гайсюк А. В.
(прізвище та ініціали)

(підпис)

Дата видачі завдання « _____ » _____ 20 _____ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Програмне забезпечення системи дистанційного навчання ЗВО.

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	16.03.2023р.	17.03.2023р.	Виконано
2.	Огляд літератури за темою роботи	18.03.2023р.	22.03.2023р.	Виконано
3.	Аналіз предметної області	23.03.2023р.	27.03.2023р.	Виконано
4.	Розробка проєктних рішень	28.03.2023р.	08.04.2023р.	Виконано
5.	Моделювання та конструювання ПЗ	10.04.2023р.	17.04.2023р.	Виконано
6.	Кодування, тестування розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	18.04.2023р.	05.05.2023р.	Виконано
7.	Розробка спеціальної частини з охорони праці	06.05.2023р.	11.05.2023р.	Виконано
8.	Оформлення КРБ та презентації	12.05.2023р.	06.06.2023р.	Виконано
9.	Попередній захист	07.06.2023р.	07.06.2023р.	Виконано
10.	Рецензування	08.06.2023р.	08.06.2023р.	Виконано
11.	Завершення оформлення КРБ та презентації	09.06.2023р.	12.06.2023р.	Виконано
12.	Відгук керівника	13.06.2023р.	13.06.2023р.	Виконано
13.	Захист кваліфікаційної роботи	27.06.2023р.	27.06.2023р.	Виконано

Розробила студентка Гайсюк А. В.

(прізвище та ініціали)

(підпис)

«___» _____ 2023 р.

Керівник роботи канд. техн. наук, доцент Давиденко Є. О.

(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

«___» _____ 2023 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Програмне забезпечення системи дистанційного навчання ЗВО»

Студентка 409 гр.: Гайсюк Аліна Валеріївна

Керівник: доцент, канд. техн. наук Давиденко Є. О.

Кваліфікаційна робота присвячена розробці програмного забезпечення системи дистанційного навчання для закладу вищої освіти.

Об'єктом дослідження є процеси проведення навчання.

Предметом дослідження є програмні засоби створення системи дистанційного навчання для ЗВО.

Мета дослідження полягає у підвищенні рівня організації дистанційних занять, а також підтримки очного навчання за рахунок створення універсальної системи, призначеної для взаємодії між адміністраторами, викладачами, студентами та батьками, за допомогою використання технологій апаратної та програмної реалізації React, NEST.JS та бази даних PostgreSQL.

Кваліфікаційна робота бакалавра складається зі вступу, чотирьох розділів, висновків та додатків.

У вступі обґрунтовується актуальність теми, описуються поставлена задача, предмет дослідження та об'єкт дослідження.

У першому розділі описується аналіз предметної області, тобто аналізується низка застосунків-аналогів систем дистанційного навчання закладів вищої освіти (ЗВО), а також система, що розробляється, визначається специфікація вимог до програмного забезпечення (ПЗ). Описуються погодження, що ухвалені в програмній документації та межі проєкту ПЗ, встановлюються спеціальні вимоги та описується інтерфейс, призначений для користувача разом з загальними обмеженнями та різними функціями системи дистанційного навчання ЗВО.

У другому розділі описується моделювання системи дистанційного

навчання ЗВО. Розглядаються етапи реалізації проєкту та створюються різні USE CASE для системи дистанційного навчання. Описуються короткі, поверхневі та альтернативні сценарії використання та функції і ролі всіх дійових осіб. Розробляється алгоритм роботи ПЗ та діаграми розгортання для системи дистанційного навчання. Також в другому розділі детально описуються різні компоненти діаграми розгортання системи. Будуються діаграми взаємодії для системи.

У третьому розділі описується проєктування системи та огляд стеку технологій. Розробляються UML-діаграми, а саме: діаграми станів та переходів, діаграми компонентів та діаграми пакетів. Оглядаються технології, що задіяні у розробці. Описано обрані мови програмування, технології front-end та back-end розробки.

У четвертому розділі надається інформація стосовно програмної реалізації вебзастосунку. Зроблено детальний опис дизайну вебзастосунку, огляд функціоналу відповідно до кожної ролі користувача. Проведено тестування системи, а саме зосереджено увагу на модульному, інтеграційному та системному тестуванні. Перевірено правильність, точність та безпека програмного забезпечення.

У висновках проводиться аналіз роботи та отриманих результатів.

КРБ викладена на 93 сторінки, вона містить 4 розділи, 46 ілюстрацій, 22 таблиці, 19 джерел в переліку посилань.

Ключові слова: *система дистанційних занять, клієнт-серверна архітектура, веброзробка, універсальна система, протоколи, React, NEST.JS та PostgreSQL.*

ABSTRACT

of the Bachelor's Thesis

«Software of the distance learning system of the Higher Education Institution»

Student of group 409: Haisiuk Alina Valeriivna

Supervisor: Ph.D. tech. Sciences, Associate Professor Davydenko Y. O.

This project is devoted to the development of distance learning system software for a higher education institution.

The object of research is the processes of training.

The subject of the study is software tools for creating a distance learning system for higher education.

The aim of the work is to increase the level of organization of distance classes, as well as support of face-to-face education by creating a universal system designed for interaction between administrators, teachers, students and parents, using the hardware and software implementation technologies of React, NEST.JS and the database of PostgreSQL data.

The explanatory note of the bachelor's thesis consists of an introduction, four sections, conclusions and appendices.

The introduction substantiates the relevance of the topic, describes the task, the subject of research and the object of research.

The first section describes the analysis of the subject area, i.e., a number of similar applications of distance learning systems of higher education institutions are analyzed, as well as the system being developed, the specification of software requirements is determined. It describes the agreements adopted in the software documentation and the boundaries of the software project, establishes special requirements and describes the user interface along with the general limitations and various functions of the distance learning system of higher education institutions.

The second section describes the modeling of the distance learning system of higher education institutions. The stages of project implementation are considered and various USE CASES for the distance learning system are created. Brief,

superficial and alternative usage scenarios and functions and roles of all actors are described. The software algorithm and deployment diagrams for the distance learning system are being developed. Also, the second section describes in detail the various components of the system deployment diagram. Interaction diagrams for the system are being built.

The third section describes the system design and overview of the technology stack. UML diagrams are developed, namely: state and transition diagrams, component diagrams and package diagrams. The technologies involved in the development are reviewed. Selected programming languages, front-end and back-end development technologies are described.

The fourth section provides information about the software implementation of the web application. A detailed description of the design of the web application, an overview of the functionality according to each user role was made. System testing was carried out, namely, attention was focused on module, integration and system testing. The correctness, accuracy and security of the software have been verified. In the conclusions, an analysis of the work and the obtained results is carried out.

This thesis contains 93 pages, 4 sections, 46 illustrations, 22 tables, 19 sources in the list of references.

Keywords: *distance learning system, client-server architecture, web development, universal system, protocols, React, NEST.JS and PostgreSQL.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд та аналіз застосунків-аналогів систем дистанційного навчання ЗВО.....	8
1.2 Аналіз системи, що розробляється	15
1.3 Специфікація вимог до ПЗ «Distance Learning Portal»	17
Висновки до розділу 1	29
2 МОДЕЛЮВАННЯ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗВО.....	30
2.1 Етапи реалізації проєкту	30
2.2 Створення USE CASE	31
2.3 Алгоритм роботи ПЗ	43
2.4 Розробка діаграми розгортання.....	47
2.5 Побудова та використання діаграм послідовності.....	51
Висновки до розділу 2	57
3 ПРОЄКТУВАННЯ СИСТЕМИ ТА ОГЛЯД СТЕКУ ТЕХНОЛОГІЙ.....	58
3.1 Розробка UML-діаграм.....	58
3.1.1 Діаграми станів та переходів	59
3.1.2 Діаграми компонентів.....	62
3.1.3 Діаграма пакетів	64
3.2 Огляд технологій	65
3.2.1 Мови програмування	67
3.2.2 Технології front-end розробки	68
3.2.3 Технології back-end розробки	69
Висновки до розділу 3	71

Кафедра інженерії програмного забезпечення	
Програмне забезпечення системи дистанційного навчання ЗВО	
4	ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗВО 72
4.1	Опис дизайну вебзастосунок 72
4.2	Тестування системи..... 82
	Висновки до розділу 4 83
	ВИСНОВКИ..... 84
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 85

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
ПЕРЕЛІК СКОРОЧЕНЬ

СДН	–	система дистанційного навчання
ЗВО	–	заклад вищої освіти
ТЗ	–	технічне завдання
ПЗ	–	програмне забезпечення
БД	–	база даних
ООП	–	об'єктно-орієнтовне програмування
НДІ	–	нормативно-довідкова інформація
LMS	–	Learning Management System

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
ВСТУП

У сучасному світі, особливо в умовах пандемії COVID-19 та повномасштабного вторгнення Росії на територію України, дистанційне навчання є важливою складовою освітнього процесу. Це дозволяє забезпечити доступ до навчання для студентів з різних куточків країни, знизити витрати на житло та транспорт, а також забезпечити безпеку для студентів і викладачів.

Однак, існують певні проблеми, пов'язані з реалізацією дистанційного навчання. До них належать відсутність достатньої взаємодії між студентами та викладачами, обмежений доступ до важливих матеріалів і інформації, неефективне використання часу на заняттях та недостатня мотивація студентів.

Розробка системи дистанційного навчання є актуальною та важливою проблемою у сфері освіти. Підставою для розробки такої системи є різноманітні технології та засоби зв'язку, які дозволяють створювати віртуальні класи, забезпечувати доступ до навчальних матеріалів та підтримувати ефективну комунікацію між викладачами та студентами.

У зв'язку з цим, необхідність розробки системи дистанційного навчання є виправданою, оскільки це дозволить дослідити проблему та розробити практичні рекомендації щодо її вирішення.

Актуальність. Пандемія COVID-19 та повномасштабне російське вторгнення значно змінили спосіб життя людей і спричинили зростання попиту на дистанційне навчання. Школи, університети та інші навчальні заклади змушені переходити на дистанційну форму навчання для забезпечення безпеки своїх студентів і викладачів. Тому питання розробки системи дистанційної освіти є актуальним та дуже своєчасним у наші дні.

Об'єктом дослідження є процеси проведення навчання.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Предметом дослідження є програмні засоби створення системи
дистанційного навчання для ЗВО.

Мета дослідження полягає у підвищенні рівня організації дистанційних занять, а також підтримки очного навчання за рахунок створення універсальної системи, призначеної для взаємодії між адміністраторами, викладачами, студентами та батьками, за допомогою використання технологій апаратної та програмної реалізації React, NEST.JS та бази даних PostgreSQL.

Для досягнення вказаної мети, в роботі поставлено та розв'язано наступні **завдання**:

1. Розробити ТЗ на розробку системи дистанційного навчання.
2. Розробити дизайн та інтерфейс користувацької частини СДН.
3. Розробити функціональні вимоги до системи, включаючи можливості для студентів, батьків, викладачів та адміністраторів.
4. Спроекувати базу даних для збереження інформації про користувачів, курси, завдання тощо.
5. Спроекувати механізми автентифікації та авторизації користувачів.
6. Спроекувати можливості для завантаження та перегляду навчальних матеріалів, відеолекцій, тестів та інших засобів навчання.
7. Спроекувати систему оцінювання результатів навчання студентів та забезпечити можливість підготовки звітів для викладачів та адміністраторів.
8. Спроекувати чат-систему для взаємодії між користувачами та адміністраторами системи.
9. Протестувати розроблену систему та забезпечити її стабільну та безперебійну роботу.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Розробка системи дистанційного навчання є актуальною задачею в сучасному світі, яка може значно полегшити процес навчання та забезпечити більш ефективне засвоєння знань студентами.

Дана робота є важливим кроком у розвитку інформаційних технологій в галузі освіти та може допомогти покращити якість навчання. Результатом роботи буде функціональна та зручна система дистанційного навчання, яка дозволить користувачам ефективно засвоювати знання та отримувати якісну освіту, незалежно від місця та часу.

Сфера застосування:

1. *Дистанційна освіта.* СДН дозволяє забезпечити підготовку студентів за різними напрямками, незалежно від місця їх проживання. Також може бути використана для тих студентів, які не можуть відвідувати заняття в звичайний робочий час через різні життєві обставини.

2. *Професійна підготовка.* СДН може бути застосована для підготовки фахівців різних напрямків або підвищення їх кваліфікації.

3. *Наукова діяльність.* СДН дозволяє забезпечити можливість дистанційного наукового спілкування між вченими, учасниками досліджень, конференцій, тощо.

4. *Організація відкритих занять.* СДН може бути використана для організації різних занять, відкритих лекцій та дискусій з метою розвитку креативності та набуття нового досвіду.

5. *Онлайн-курси.* СДН дозволяє забезпечити доступ до різних онлайн-курсів, що збільшує можливість підвищення рівня професійних знань та навичок.

Отже, система дистанційного навчання може бути використана в різних сферах життєдіяльності для забезпечення можливості навчання та професійного розвитку дистанційним способом.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд та аналіз застосунків-аналогів систем дистанційного навчання ЗВО

Застосунки-аналоги систем дистанційного навчання ЗВО – це різні платформи та системи, які надають змогу навчатися віддалено через Інтернет. Ці системи можуть мати різні функції, інтерфейси та можливості, але мета їх одна – забезпечити якісну, зручну та доступну платформу для навчання.

Аналіз та огляд цих застосунків може бути корисним для ЗВО, які хочуть покращити свою систему дистанційного навчання. Він допоможе визначити найкращі функції та можливості, які можуть бути використані для покращення власної СДН, а також оцінити потенційні переваги та недоліки систем-аналогів. Для огляду та аналізу застосунків-аналогів СДН обрано наступні застосунки: Moodle (табл. 1.1), Canvas (табл. 1.2), Docebo (табл. 1.3).

Moodle

«Moodle» – це одна з найбільш популярних систем управління навчанням (LMS) у світі, розроблена як відкрите програмне забезпечення, що означає, що її можна використовувати безкоштовно та модифікувати за необхідності [1]. Що ж можна віднести до детальних характеристик системи дистанційного навчання «Moodle»?

1. *Модульність.* Ця система є модульною, що дозволяє користувачам встановлювати додаткові модулі та функції за потреби.

2. *Навчальні курси.* Система дозволяє викладачам створювати та редагувати навчальні курси відповідно до власних бажань та потреб. Курси можуть містити різні типи матеріалів, наприклад, текстові файли, відео/аудіо записи, інтерактивні завдання тощо. Ця СДН має різні типи тестів та завдань.

3. *Взаємодія та спілкування.* Система дозволяє студентам та викладачам взаємодіяти та спілкуватися між собою через різні

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
інструменти(чати, форуми, електронні листи, тощо). Особливістю є те, що «Moodle» має можливості для колективної роботи, такі як спільне створення документів та форумів обговорень.

4. *Адаптивність.* Немало важлива характеристика системи «Moodle» – підтримка адаптивного дизайну, що дозволяє користувачам легко користуватися платформою на будь-яких пристроях, включаючи комп'ютери, планшети та мобільні телефони (рис. 1.1).

5. *Безпека.* Система забезпечує високий рівень безпеки, зокрема, захист даних користувачів, авторизація та автентифікація користувачів, захист від зловмисних атак та інше.

Таблиця 1.1 – Основні характеристики СДН «Moodle»

Назва	Moodle
Розробник	Мартін Догіамас
Мова реалізації	PHP
Архітектура	3tier web application, mobile application
Сайт	https://moodle.org/
Перелік функцій, характеристик	<ol style="list-style-type: none"> 1. Створення покрокових інтерактивних курсів шляхом завантаження відео, аудіо, текстів, презентацій та зображень із розмежуванням доступу для студентів. 2. Створення анкет, опитувань та тестувань, що дозволяють оцінити рівень знань студентів та отримати зворотній зв'язок з того чи іншого питання. 3. У режимі редагування курсу доступні форми зв'язку зі студентами, такі як чат, відеочат та коментарі для отримання запитань. 4. Систему можна використовувати як базу знань. Це своєрідне сховище навчальних матеріалів. Усі матеріали зберігаються у системі та доступні студентам у будь-який час. 5. СДН має вбудовану статистику успішності та генерацію автоматичних звітів, що дозволяє відстежувати відвідуваність та успіхи студентів, формувати звіти.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Кінець таблиці 1.1

<p>Переваги</p>	<ol style="list-style-type: none"> 1. «Moodle» є системою з відкритим програмним кодом та в свою чергу безкоштовним програмним забезпеченням. 2. Це гнучка система з можливістю створення додаткових функцій, підвищувати інтерактивність за допомогою спеціальних плагінів налаштування. 3. Система має просту, інтуїтивно зрозумілу навігацію. 4. Доступ до системи можна отримати як в режимі онлайн, так і в режимі офлайн. 5. Навчання в «Moodle» централізоване та інтерактивне, всі ресурси зосереджені в одному місці. 6. В системі є повна видимість статистики та результатів, з можливістю отримання управлінських звітів та показників.
<p>Недоліки</p>	<ol style="list-style-type: none"> 1. Безкоштовна версія системи може здаватися складною в установці та потребувати допомоги спеціалістів. 2. Деякі користувачі відмічають незручний інтерфейс системи та недостатньо привабливі шаблони курсів. 3. Для користувачів хмарної версії може стати проблемою збереження інформації на іноземному сервері. 4. Чим більше студентів отримує доступ до платформи, тим повільнішою стає система. 5. Система може іноді закриватися, блокуючи студентам можливість доступу до матеріалів курсу.



Рисунок 1.1 – Інтерфейс застосунку «Moodle»

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Canvas

«Canvas» – це одна з популярних систем дистанційного навчання, яка використовується в ЗВО та інших освітніх установах. Вона розроблена компанією Instructure та має велику кількість функцій та інструментів для організації та проведення навчання в онлайн-форматі [2]. Що ж можна віднести до детальних характеристик системи дистанційного навчання «Canvas»?

1. *Зручний інтерфейс.* Система має зручний інтерфейс, що дозволяє користувачам легко орієнтуватися в системі та знаходити необхідну інформацію.

2. *Гнучкість та налаштування.* Система дозволяє налаштовувати навчальний процес під конкретні потреби користувачів. Можливість налаштування курсів, зміну прав доступу до контенту та ін.

3. *Інтеграція з багатьма системами.* СНД може бути легко інтегрована з іншими системами, такими як Google Drive, Dropbox, YouTube, Microsoft Office тощо.

4. *Можливості комунікації.* Система надає велику кількість інструментів для комунікації користувачів, сюди можна додати форуми, чати, електронні листи, завдання та інше.

5. *Адаптивність.* Система підтримує адаптивний дизайн, який дозволяє користувачам легко користуватися платформою на будь-яких пристроях: комп'ютерах, планшетах та мобільних телефонах (рис. 1.2).

6. *Підтримка мов.* Система підтримує більше 40 мов, що дозволяє користувачам з різних куточків землі використовувати платформу на рідній мові.

Таблиця 1.2 – Основні характеристики СДН «Canvas»

Назва	Canvas LMS
Розробник	Instructure Company
Мова реалізації	Ruby on Rails

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Кінець таблиці 1.2

Архітектура	3tier web application, mobile application
Сайт	https://www.canvaslms.com
Перелік функцій, характеристик	<ol style="list-style-type: none"> 1. Можливість імпорту раніше розроблених навчальних курсів. 2. Відстеження прогресу студентів для прийняття своєчасних та обґрунтованих рішень. 3. Організація та участь у онлайн-комунікаціях (конференції, лекції, вебінари). 4. Доступ батьків/керівників до даних прогресу навчання для своєчасної допомоги своїй дитині/підлеглому.
Переваги	Основні переваги системи ґрунтуються на наявності сучасних інструментів для створення курсів, відкритому API та підтримці багатьох сторонніх розширень та інтеграцій, за допомогою яких розширюється функціонал системи. Також до плюсів можна віднести зручний інтерфейс, гнучке налаштування, можливість інтеграції з іншими системами та велику кількість інструментів для співпраці та комунікації між користувачами.
Недоліки	<ol style="list-style-type: none"> 1. Відсутність гейміфікації. 2. Висока вартість. 3. Обмежена функціональність в порівнянні з іншими системами дистанційного навчання. Наприклад, не всі функції є доступними для безкоштовної версії системи. 4. Нестабільна робота системи. Деякі користувачі відзначають, що система може бути нестабільною час від часу, зокрема, виникають проблеми з завантаженням сторінок та відео. 5. Проблеми з підтримкою. Відзначається, що іноді може бути важко отримати допомогу та підтримку з боку команди розробників «Canvas».



Рисунок 1.2 – Інтерфейс застосунку «Canvas»

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Docebo

«Docebo» – це система дистанційного навчання, яка використовується у закладах вищої освіти. Вона надає можливість створювати та організовувати онлайн-курси, віртуальні класи, тестування та інші навчальні матеріали. «Docebo» також має ряд функцій для аналізу даних, зокрема збору статистики щодо успішності студентів та прогресу навчання [3]. Що ж можна віднести до детальних характеристик системи дистанційного навчання «Docebo»?

1. *Модульність.* Система складається з різних модулів, що дозволяє користувачам обирати тільки ті функції, які їм потрібні.
2. *Мультимовність.* Система підтримує багато мов, що дозволяє користувачам користуватися платформою на своїй рідній мові.
3. *Можливість підтримка системи на мобільних пристроях.* «Docebo» має застосунки для iOS та Android, що дозволяє користувачам мати доступ до курсів та інших навчальних матеріалів з будь-якого пристрою.
4. *Інтерактивність.* Система має багато інтерактивних елементів, наприклад відео, тести, віртуальні класи, які допомагають забезпечувати більш ефективне навчання (рис. 1.3).
5. *Система відстеження прогресу.* СДН має функції, які дозволяють користувачам відслідковувати прогрес та оцінювати їх успішність.
6. *Підтримка соціального навчання* через форуми та чати.
7. *Підтримка віртуальної реальності.* Система має функції, що дозволяють використовувати віртуальну реальність для покращення навчального процесу.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Таблиця 1.3 – Основні характеристики СДН «Docebo»

Назва	Docebo
Розробник	Docebo Company
Мова реалізації	PHP
Архітектура	3tier web application, mobile application
Сайт	https://www.docebo.com/
Перелік функцій, характеристик	<ol style="list-style-type: none"> 1. Можливість створення, редагування, видалення та публікації курсів, завдань, тестів, відео/аудіо матеріалів тощо. 2. Можливість проведення онлайн-занять, конференцій, семінарів з використанням вбудованої технології відеоконференцій. 3. Можливість створення різних груп, форумів, чатів, для спілкування між користувачами. 4. Можливість проведення оцінювання з використанням тестів, завдань тощо. 5. Можливість отримання статистичних даних про активність користувачів, їхній прогрес та результативність навчання. 6. Можливість доступу до системи з мобільних пристроїв. 7. Можливість інтеграції з різноманітними системами, такими як CRM, HRM, ERP тощо. 8. Можливість роботи з різними мовами, що дає можливість використовувати систему для навчання у міжнародному контексті.
Переваги	<ol style="list-style-type: none"> 1. Гнучкість та зручність. 2. Інтерактивність та співпраця з викладачами та іншими студентами. 3. Наявність аналітики, збір та аналіз даних про прогрес та результативність студентів.
Недоліки	<ol style="list-style-type: none"> 1. <i>Ціна.</i> «Docebo» є досить дорогою системою порівняно з іншими СДН. 2. <i>Складність використання.</i> «Docebo» може бути важко налаштувати та використовувати, що може здаватися неприйнятним для користувачів з обмеженим досвідом роботи з інтернет-платформами. 3. <i>Візуальний інтерфейс користувача</i> може бути менш привабливим порівняно з іншими СДН. 4. <i>Проблеми з підтримкою клієнтів.</i> Користувачі можуть стикатися з проблемами у взаємодії з клієнтською підтримкою «Docebo».

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

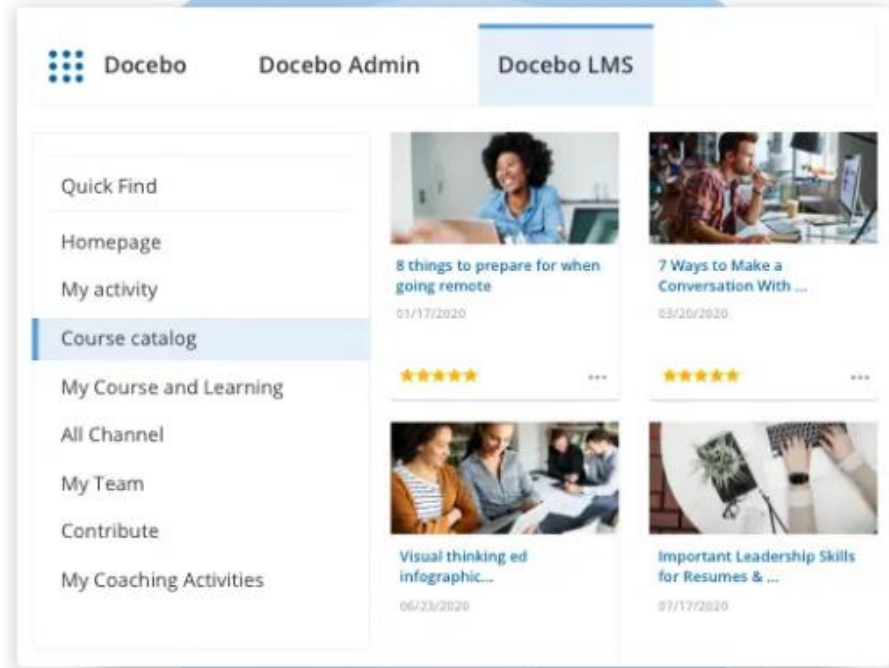


Рисунок 1.3 – Інтерфейс застосунку «Docebo»

Загалом, системи дистанційного навчання відкривають нові можливості для навчання, забезпечуючи гнучкість, доступність та інтерактивність, пристосовуючись до сучасних вимог та потреб студентів.

1.2 Аналіз системи, що розробляється

Розробка системи дистанційного навчання «Distance Learning Portal» (рис. 1.4 – 1.7) призначена для поліпшення рівня дистанційної освіти в Україні, шляхом створення універсальної системи, призначеної для взаємодії між адміністраторами, викладачами, студентами та батьками, та організації традиційних дистанційних занять, а також підтримки очного навчання.

Система повинна працювати з браузером, що підтримує HTML5 та JavaScript. Швидкість доступу до мережі Інтернет для клієнта повинна бути не менше, ніж 60 кб/сек. Система має бути адаптивною та коректно працювати як с комп'ютерами, так і з смартфонами.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Таблиця 1.4 – Опис СДН «Distance Learning Portal»

<p>Користувачі та їх функції</p>	<p>В системі присутні п'ять ролей: супер-адміністратор, адміністратор, викладач, студент та батьки.</p> <p>Супер-адміністратор – створює університет у системі та визначає адміністраторів до цієї системи.</p> <p>Адміністратор – веде інформацію по користувачах, керує обліковими записами, додає дані про стан фінансів.</p> <p>Викладач – створює курс та редагує його, керує тестами, має доступ до засобів спілкування.</p> <p>Студент – проходить тестування і переглядає результат тестування, має доступ до матеріалів курсів.</p> <p>Батьки – переглядають зміст курсів, на яких навчаються їх діти, та їх оцінки, мають доступ до власного кабінету, з метою огляду розділів курсу навчання, оцінок, фінансів (стан рахунку/заборгованостей) та документів (наприклад, копії контрактів навчання, особисті документи, реквізити університету задля оплати та інше).</p>
<p>Сценарії роботи</p>	<p>Сценарій 1. Супер-адміністратор реєструє адміністраторів до окремих університетів задля коректного функціонування системи.</p> <p>Сценарій 2. Адміністратори створюють нових користувачів системи (викладачі, студенти, батьки) задля коректного функціонування системи.</p> <p>Сценарій 3. Кожен користувач в свою чергу виконує ряд дій, який забезпечує функціонал системи СДН, наприклад, створення курсів, тестів, пошук курсів тощо.</p> <p>Сценарій 4. Кожен користувач має доступ до засобів спілкування при необхідності.</p>

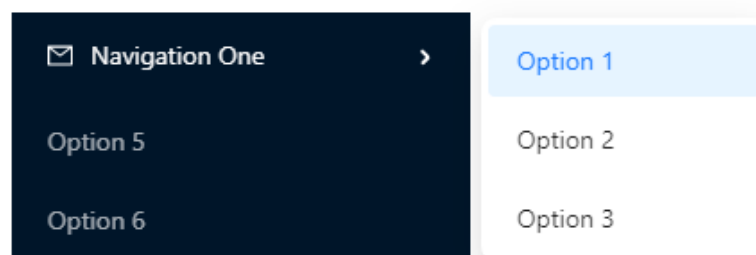


Рисунок 1.4 – Макет теми підменю застосунку «Distance Learning Portal»

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО



Рисунок 1.5 – Макет прямого переходу до сторінки застосунку
«Distance Learning Portal»

Рисунок 1.6 – Макет компонента форми застосунку «Distance Learning Portal»

Система дистанційного навчання, що розробляється, має на меті поєднати сучасні технології, особисту універсальність та широкий доступ до навчального контенту для створення зручної та ефективної навчальної платформи.

1.3 Специфікація вимог до ПЗ «Distance Learning Portal»

Опис створюваного продукту: створення системи дистанційного навчання є актуальною проблемою в сучасному світі. Метою даного проєкту є створення вебплатформи для проведення онлайн-занять з можливістю відслідковування прогресу студентів, викладання лекцій, публікації

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
матеріалів для самостійного вивчення, тестування та оцінювання знань
студентів.

Призначення продукту: система дистанційного навчання призначена для надання можливості навчання студентам з будь-якого місця, використовуючи Інтернет. Система дозволяє викладачам створювати курси та завантажувати на них лекції, завдання та тестування, а студенти можуть відвідувати ці курси та проходити тестування від свого комп'ютера.

Погодження, що ухвалені в програмній документації

Погоджено, що для розробки системи дистанційного навчання «Distance Learning Portal» та його правильного функціонування буде використано технології апаратної та програмної реалізації React, NEST.JS та бази даних PostgreSQL.

Межі проєкту ПЗ

Завершення роботи над розробкою ПЗ «Distance Learning Portal» – 25.05.2023 р.

Спеціальні вимоги

1. Вебплатформа повинна бути доступною для використання на різних пристроях (комп'ютерах, планшетах, смартфонах).
2. Матеріали курсу повинні бути доступні для вивчення у будь-який час, без обмежень у часі та місці.
3. Система повинна мати можливість відслідковувати прогрес студентів та зберігати результати тестування.
4. Система повинна бути зручною та інтуїтивно зрозумілою у використанні для користувачів з різним рівнем комп'ютерної грамотності.

Опис призначеного для користувача інтерфейсу системи дистанційного навчання має включати такі елементи:

1. Головне меню: містить основні пункти меню, такі як «Головна», «Мої курси», «Мій профіль» та «Вихід».

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

2. Сторінка «Головна»: містить огляд останніх новин та актуальних курсів.
3. Сторінка «Мої курси»: містить список курсів, на які користувач підписаний, або які він створив. Кожен курс повинен мати свою сторінку з описом курсу та переліком лекцій, завдань та тестів.
4. Сторінка «Мій профіль»: містить інформацію про користувача, його персональні дані, статистику проходження курсів, налаштування профілю та можливість зміни пароля.
5. Опція «Вихід»: дозволяє користувачеві вийти з системи.
6. Сторінки курсів: містять матеріали курсу, такі як лекції, завдання та тести. Кожна лекція має свою сторінку з текстом лекції та можливістю перегляду матеріалів.
7. Панель інструментів: містить додаткові функції, такі як пошук курсів, фільтри, панель навігації та список активних курсів.
8. Панель налаштувань: містить можливість налаштування повідомлень, мови інтерфейсу, настройки безпеки.
9. Допоміжні елементи: до них відносяться кнопки для перемикання між повідомленнями та лекціями, показу результатів тестів та завдань, список активних та завершених курсів. Для кращого розуміння та зручності використання системи дистанційного навчання використано ясну, легку та дружельобну графіку та кольори.

План розробки системи дистанційного навчання:

1. Аналіз вимог користувачів та створення технічного завдання.
2. Розробка архітектури системи та вибір технологій для реалізації.
3. Розробка бази даних та інтерфейсу для користувачів.
4. Розробка модуля викладача, який дозволяє створювати курси, публікувати лекції та тестування, а також відслідковувати прогрес студентів.
5. Розробка модуля студента, який дозволяє студентам відвідувати лекції, виконувати завдання та проходити тестування.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

6. Розробка модулів супер-адміністратора та адміністратора, який дозволяє керувати користувачами та їх правами, відслідковувати прогрес студентів та керувати курсами.

7. Розробка модуля батьків, який дозволяє підтримувати зв'язок із викладачами та навчальним закладом, переглядати зміст курсів, на яких навчаються їх діти, та їх оцінки, мати доступ до власного кабінету, з метою огляду розділів курсу навчання, оцінок, фінансів (стан рахунку/заборгованостей) та документів (наприклад, копії контрактів навчання, особисті документи, реквізити університету задля оплати та інше).

8. Тестування та відлагодження системи.

9. Оформлення документації.

Сфера застосування

Системи дистанційного навчання знайшли поширене застосування в різних галузях і сферах життя [4]. Основними з них є:

1. *Освіта.* Системи дистанційного навчання широко використовуються у вищій освіті, навчальних закладах або професійних курсах. Вони дозволяють студентам отримувати знання з будь-якого місця, не залежно від різних факторів.

2. *Корпоративне навчання.* Системи дистанційного навчання є ефективним інструментом для проведення тренінгів та курсів навчання в корпоративній сфері.

3. *Медична освіта.* Системи дистанційного навчання також використовуються у медичній освіті та підвищенні кваліфікації медичних працівників. Це дозволяє медичним фахівцям вивчати нові методики та практики без необхідності виїзду на додаткові курси.

4. *Індивідуальне навчання.* Системи дистанційного навчання також можуть використовуватися для індивідуального навчання та самоосвіти. Вони дозволяють людям вивчати нові навички, проходити курси та тренінги

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
в зручний для них час і місце, що дає їм більшу свободу та гнучкість у навчанні.

Отже, системи дистанційного навчання мають широкий спектр застосування і використовуються в різних сферах життя. Вони дозволяють забезпечити доступ до навчання для більшої кількості людей та зменшити бар'єри в доступі до освіти. Вони також можуть знизити витрати на навчання та підвищити ефективність процесу навчання за рахунок інтерактивності та персоналізації матеріалів. Крім того, системи дистанційного навчання можуть бути корисними в умовах епідемії або карантину, коли фізичний контакт обмежений або заборонений.

Характеристики користувачів

Головні характеристики користувачів: ПК, смартфон або планшет з доступом в мережу Інтернет.

Структура і склад системи

Щоб створити програмне забезпечення, необхідно мати наявність таких складових, як сервер, база даних, вебзастосунок та інтерфейс програмування застосунків (API).

Загальні обмеження

Для коректної роботи програмного забезпечення необхідно мати можливість підключення до мережі Інтернет.

ФУНКЦІЇ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗВО

Функція додавання нового завдання

Опис функції

Функція додавання нового завдання в системі дистанційного навчання дозволяє викладачам створювати нові завдання для студентів, які можуть бути виконані в онлайн-форматі.

Вхідна і вихідна інформація

Вхідна інформація – назва завдання, опис завдання, термін виконання, вміст завдання (наприклад, текстове завдання або завдання з використанням

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
мультимедіа матеріалів), критерії оцінювання завдання та інші додаткові параметри, які можуть бути вимогами викладача.

Вихідна інформація – методичні рекомендації, створене завдання, яке буде доступне для виконання студентами в системі дистанційного навчання. Після того, як студент виконає завдання, результати його виконання можуть бути доступні викладачеві для оцінювання.

Функціональні вимоги

1. Система повинна дозволяти викладачам створювати нові завдання для студентів з відповідними параметрами (назвою, описом, терміном виконання, вмістом, критеріями оцінювання та іншими параметрами).

2. Система повинна дозволяти викладачам редагувати створені завдання, якщо необхідно внести зміни у вміст, критерії оцінювання або інші параметри.

3. Студенти повинні мати можливість переглядати завдання, створені викладачем, з усіма необхідними параметрами та вмістом.

4. Система повинна дозволяти студентам виконувати завдання, які були створені викладачем.

5. Система повинна зберігати результати виконання завдання студентами та робити їх доступними для перегляду викладачем.

6. Викладач повинен мати можливість оцінювати результати виконання завдань студентами, встановлюючи відповідну оцінку та відмітки щодо виконання завдання.

Функція проходження тестування

Опис функції

Функція проходження тестування студентами в системі дистанційного навчання дозволяє студентам проходити тести з різних предметів, контролювати свої знання та отримувати оцінки за результатами тестування.

Основні етапи цієї функції включають:

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

1. Студент обирає тест, який хоче пройти.
2. Студент проходить тест, відповідаючи на питання за допомогою відповідних інтерфейсів в системі.
3. Система перевіряє відповіді студента та автоматично розраховує оцінку за тест.
4. Студент отримує результати тестування та оцінку.

Вхідна і вихідна інформація

Вхідна інформація – список питань, які містяться в тесті, відповіді студента на кожне питання та інформацію про обраний студентом тест.

Вихідна інформація – результати тестування та оцінка, яку отримав студент.

Функціональні вимоги

1. Студент повинен мати можливість обрати тести зі списку доступних тестів.
2. Студент повинен мати можливість переглядати список питань тесту та відповідати на них.
3. Система повинна перевіряти відповіді студента та розраховувати його оцінку за тест.
4. Система повинна зберігати результати тестування студента та робити їх доступними для перегляду викладачем.
5. Система повинна мати можливість автоматично перевіряти відповіді студента за допомогою відповідних алгоритмів.
6. Студент повинен мати можливість бачити свої результати тестування та оцінку в системі.

Функція редагування тесту

Опис функції

Функція редагування тесту викладачами дозволяє викладачам редагувати існуючі тести. Ця функція є важливою для забезпечення актуальності та якості тестових завдань.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Вхідна і вихідна інформація

Вхідна інформація – назва тесту та опис, запитання та їх варіанти відповідей, налаштування тесту.

Вихідна інформація – оновлений тест, підтвердження про успішне створення або редагування тесту.

Функціональні вимоги

1. Система повинна забезпечувати доступ до інтерфейсу редагування тестів тільки для викладачів з відповідними правами.
2. Система повинна зберігати історію змін для кожного тесту, щоб відновити попередні версії у разі необхідності.
3. Система повинна дозволяти викладачам додавати нові запитання та варіанти відповідей до тесту.
4. Система повинна надавати можливість зміни порядку запитань та відповідей у тесті.
5. Можливість збереження змін до тестів та їх публікація для студентів.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

Основним джерелом вхідної інформації для системи дистанційного навчання є користувачі, які входять до системи з різних пристроїв (комп'ютерів, планшетів, смартфонів) та мають можливість взаємодіяти з системою через вебінтерфейс або мобільний застосунок. Користувачі можуть бути студентами, викладачами, адміністраторами та іншими учасниками навчального процесу.

Вхідна інформація може включати в себе різні типи даних, такі як особисті дані користувачів, інформацію про курси та предмети, матеріали для навчання, тестові завдання, звіти про виконання завдань студентами та інше.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Важливо, щоб система дистанційного навчання забезпечувала безпеку та конфіденційність даних користувачів, а також надійність та швидкість обробки вхідної інформації.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

НДІ для системи дистанційного навчання може включати в себе різні класифікатори, довідники та інші документи, що визначають норми та правила для навчального процесу. Наприклад:

1. *Класифікатори предметів та курсів.* Такі класифікатори допомагають студентам знаходити необхідні курси та предмети, а також допомагають викладачам класифікувати та організовувати навчальний матеріал.

2. *Довідники з термінів та визначень.* Такі довідники допомагають користувачам зрозуміти терміни та визначення, що використовуються в навчальному процесі.

3. *Довідники з методик викладання та організації навчального процесу.* Ці довідники допомагають викладачам організовувати та проводити заняття, визначати методи та технології викладання.

4. *Нормативні документи щодо організації дистанційного навчання.* Такі документи визначають правила та норми організації дистанційного навчання, включаючи правила використання технічних засобів, захисту персональних даних, та інші.

Вимоги до способів організації, збереження та ведення інформації

БД – PostgreSQL та RESTful API.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Потрібно мати комп'ютери з достатньою потужністю та розширенням екрана для зручної роботи з системою. Стабільне інтернет-з'єднання потрібно для забезпечення безперебійного доступу мережі Інтернет та можливість

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
роботи з браузерами Chrome, Firefox, Opera, Safari, а також достатня кількість оперативної пам'яті для розробки системи.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Система складається з трьох основних компонентів: програмного забезпечення на стороні користувача, програмного забезпечення на серверній стороні та бази даних.

Системне програмне забезпечення

Застосунок повинен бути розроблений на основі фреймворку React. Для розробки front-end частини використовується Ant Design та React.js, а для розробки back-end частини – Nest.js та PostgreSQL. Взаємодія між компонентами застосунку передбачає використання RESTful API. В якості системи управління базами даних для застосунку обрано PostgreSQL.

Мережне програмне забезпечення

Для розробки програмного забезпечення була використана операційна система Windows 10, для написання коду – редактор VS Code, а для перегляду вебзастосунку – браузер Google Chrome.

Програмне забезпечення ведення інформаційної бази

База застосунку обслуговується за допомогою операцій створення, читання, оновлення та видалення даних за допомогою PostgreSQL.

Мова і технологія розробки ПЗ

Розробка програмного забезпечення передбачає використання JavaScript як мову розробки системи та фреймворку React.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Нижче подано деякі вимоги до зовнішніх інтерфейсів користувача системи дистанційного навчання:

1. Простий і зрозумілий інтерфейс для користувачів з різними рівнями технічної грамотності.

Кафедра інженерії програмного забезпечення

Програмне забезпечення системи дистанційного навчання ЗВО

2. Інтуїтивно зрозуміле та логічне розташування функцій та елементів керування.
3. Наявність уніфікованого стилю оформлення для всіх сторінок та елементів інтерфейсу.
4. Адаптивний дизайн, що дозволяє користувачам переглядати сайт на будь-якому пристрої (комп'ютері, планшеті, смартфоні) та з будь-якого місця з доступом до Інтернету.
5. Наявність можливості персоналізації інтерфейсу для користувачів, зокрема налаштування функціональності та відображення інформації відповідно до їхніх індивідуальних потреб.
6. Зручна та швидка навігація між різними сторінками та розділами системи дистанційного навчання.
7. Наявність можливості пошуку інформації та фільтрування результатів.
8. Забезпечення доступу до необхідної інформації та функціональності за допомогою мінімальної кількості кроків.
9. Надійний захист персональної інформації та даних користувачів.

Апаратний інтерфейс

Для взаємодії з вебзастосунком буде використовуватися пристрій користувача, такий як персональний комп'ютер, смартфон або планшет, який і є апаратним інтерфейсом.

Програмний інтерфейс

React – це бібліотека JavaScript для розробки інтерфейсів користувача (UI). Він може використовуватися для створення вебзастосунків, що працюють в браузері, а також для розробки мобільних застосунків на платформі React Native.

React дозволяє розробникам створювати складні інтерактивні інтерфейси за допомогою компонентів, які можуть бути повторно використані. Кожен компонент містить відповідний HTML, CSS та

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
JavaScript-код, який описує його поведінку та вигляд. За допомогою React, розробник може легко створювати складні інтерфейси з багатьма взаємодіючими компонентами, що реагують на дії користувача.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. **Модульність.** Система має модульну архітектуру, що дозволяє легко розширювати та модифікувати її функціональність відповідно до потреб користувачів.

2. **Множинна платформа.** Система має можливість працювати на різних платформах, таких як настільні комп'ютери, смартфони, планшети та інші мобільні пристрої.

3. **Безпека.** Система повинна мати механізми захисту від несанкціонованого доступу до користувацьких даних, таких як паролі та інші методи автентифікації.

4. **Можливості звітності.** Система повинна мати можливість генерувати різні звіти та статистику щодо активності користувачів, що дозволить аналізувати їх поведінку та відстежувати прогрес у навчанні.

5. **Керування користувачами.** Система має вбудований механізм керування користувачами, що дозволяє регулювати рівень доступу до різних функцій системи.

6. **Функціональність.** Система має широкий спектр функціональних можливостей, таких як створення та організація курсів, взаємодія між студентами та викладачами, моніторинг навчального процесу та інші.

7. **Легкість використання.** Система повинна мати інтуїтивно зрозумілий інтерфейс, що дозволяє легко використовувати її функціональність без необхідності вивчення складної документації або проходження спеціальної підготовки.

Всі ці властивості є дуже важливими для ефективної та успішної роботи системи дистанційного навчання.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Висновки до розділу 1

В першому розділі кваліфікаційної роботи бакалавра проведено аналіз предметної області. Розглянуто та проаналізовано низку застосунків-аналогів систем дистанційного навчання ЗВО. Проаналізовано систему, що розробляється та визначено специфікацію вимог до ПЗ.

Описано створюваний продукт та визначено його призначення. Описані погодження, що ухвалені в програмній документації та межі проєкту ПЗ. Встановлено спеціальні вимоги та описано інтерфейс, що призначений для користувача. Зроблено план розробки системи дистанційного навчання та виокремлено сферу її застосування. Визначено характеристики користувачів, структуру і склад системи. Описані загальні обмеження та різні функції системи дистанційного навчання ЗВО.

Визначено вимоги до інформаційного забезпечення, джерела і зміст вхідної інформації, нормативно-довідкову інформацію та вимоги до способів організації, збереження та ведення інформації. Зазначено вимоги до технічного забезпечення, вимоги до програмного забезпечення, описано архітектуру програмної системи, системне програмне забезпечення, мережне програмне забезпечення, програмне забезпечення ведення інформаційної бази. Визначено мову і технологію розробки ПЗ, вимоги до зовнішніх інтерфейсів, а саме інтерфейс користувача, апаратний інтерфейс, програмний інтерфейс, описано комунікаційний протокол та властивості програмного забезпечення.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

2 МОДЕЛЮВАННЯ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗВО

2.1 Етапи реалізації проєкту

Реалізація проєкту моделювання системи дистанційного навчання ЗВО може бути поділена на декілька етапів:

1. *Аналіз вимог та потреб.* На цьому етапі потрібно провести дослідження ринку та визначити потреби студентів, викладачів та інших учасників навчального процесу у дистанційному навчанні. Важливо визначити, які функції має виконувати система дистанційного навчання та які її можливості.

2. *Проектування архітектури системи.* На цьому етапі створюються детальні плани структури системи, обираються необхідні технології та розроблюються сценарії її роботи.

3. *Розробка та тестування.* На цьому етапі розробляється програмний код та інтерфейс користувача. Після цього проводиться тестування системи, щоб переконатися у її працездатності та відповідності вимогам.

4. *Впровадження та налаштування.* На цьому етапі система дистанційного навчання встановлюється на сервері та налаштовується для роботи зі студентами та викладачами. Важливо переконатися, що система працює стабільно та відповідає вимогам користувачів.

5. *Підтримка та розвиток.* Після впровадження системи потрібно забезпечувати її підтримку та розвиток. Це означає вирішення технічних проблем, оновлення програмного забезпечення та постійне вдосконалення функцій системи.

Крім того, важливо враховувати те, що реалізація проєкту моделювання системи дистанційного навчання ЗВО може зайняти тривалий час та вимагати значних фінансових та людських ресурсів [5]. Тому на

Кафедра інженерії програмного забезпечення

Програмне забезпечення системи дистанційного навчання ЗВО

кожному етапі реалізації проєкту важливо забезпечувати ефективне управління проєктом, встановлювати реалістичні терміни та бюджет, використовувати ефективні методики та інструменти управління проєктами.

Усі ці етапи та вимоги мають бути враховані при реалізації проєкту моделювання системи дистанційного навчання ЗВО, щоб забезпечити успішну та ефективну реалізацію проєкту та задоволення потреб користувачів.

Під час проєктування архітектури застосунку, програмне забезпечення розробляється шляхом створення різних типів UML-діаграм, таких як варіанти використання, діаграми послідовності, діаграма класів, станів, діяльності, компонентів та розгортання, використовуючи спеціалізовані продукти, такі як draw.io.

Створення UML-діаграм допомагає розробникам краще зрозуміти структуру проєкту. Після цього наступним етапом є розробка мокапів застосунку відповідно до вимог, включаючи логотип та вибір колірної гами.

Після цього відбувається розробка front-end та back-end частин застосунку, а також тестування на помилки для запобігання проблем при роботі ПЗ.

2.2 Створення USE CASE

Сценарії використання (use cases) – це техніка вимог до програмного забезпечення, яка описує функціональність системи з точки зору користувача або зовнішньої системи. Вони використовуються для опису того, як користувачі будуть взаємодіяти з системою та як система повинна реагувати на їх дії (рис. 2.1).

Кожен сценарій використання містить список кроків, які користувач повинен виконати, щоб виконати певну функцію, а також список можливих результатів, які повинна повернути система. Сценарії використання можуть бути використані для тестування програмного забезпечення та визначення

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
його вимог. Вони допомагають уточнити функціональність системи та
забезпечують взаєморозуміння між розробниками, тестувальниками та
замовниками.

Короткий сценарій використання

Користувач відкриває в браузері вебсайт системи дистанційного навчання та запитує логін/пароль. Після перевірки логіну/пароллю система виводить головне меню, де користувач обирає необхідні для нього опції. Він визначає мету задля якої зайшов у систему та відкриває необхідні вкладки та робить різні запити. Система працює коректно, запити виконуються задовільно. Коли мета користувача була досягнута, він обирає опцію «Log out» та закриває систему.

Поверхневий сценарій використання «Пошук необхідних матеріалів у СДН»

Головний сценарій (успішний):

1. Користувач заходить до СДН та вводить коректні дані (логін, пароль) для того, щоб увійти в свій обліковий запис.
2. Система коректно відповідає на запит та завантажує обліковий запис користувача.
3. Система відображає всю необхідну інформацію про курси, оцінки тощо.
4. Користувач робить відповідні запити, щоб знайти зміст курсів та оцінок, система видає йому ці дані.
5. Користувач робить відповідні запити, щоб працювати з засобами для групової роботи, система видає йому ці дані.
6. Користувач робить відповідні запити, щоб переглянути результати проходження дистанційного курсу, система видає йому ці дані.
7. Користувач робить відповідні запити, щоб переглянути результати проходження тестів, система видає йому ці дані.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

8. Користувач робить відповідні запити, щоб завантажити файли з виконаними завданнями, система видає йому ці дані.

9. Користувач робить відповідні запити, щоб розмістити навчальні матеріали на різних курсах, система працює коректно.

10. Користувач робить відповідні запити, задля проведення швидкої модифікації навчальних матеріалів, система працює коректно.

Альтернативні сценарії:

1. Користувач є відрахованим, система не може завантажити обліковий запис користувача.

2. Користувач є заблокованим, система не може завантажити обліковий запис користувача.

3. Система не може коректно відповідати на запити, тому що не витримує навантаження кількості відвідувачів на сайті.

4. Дані на формі автентифікації не пройшли валідацію. Користувач не може отримати доступ до необхідної інформації.

5. Технічний збій роботи системи. Видається повідомлення «Немає зв'язку з сервером». Користувач викликає адміністратора системи.

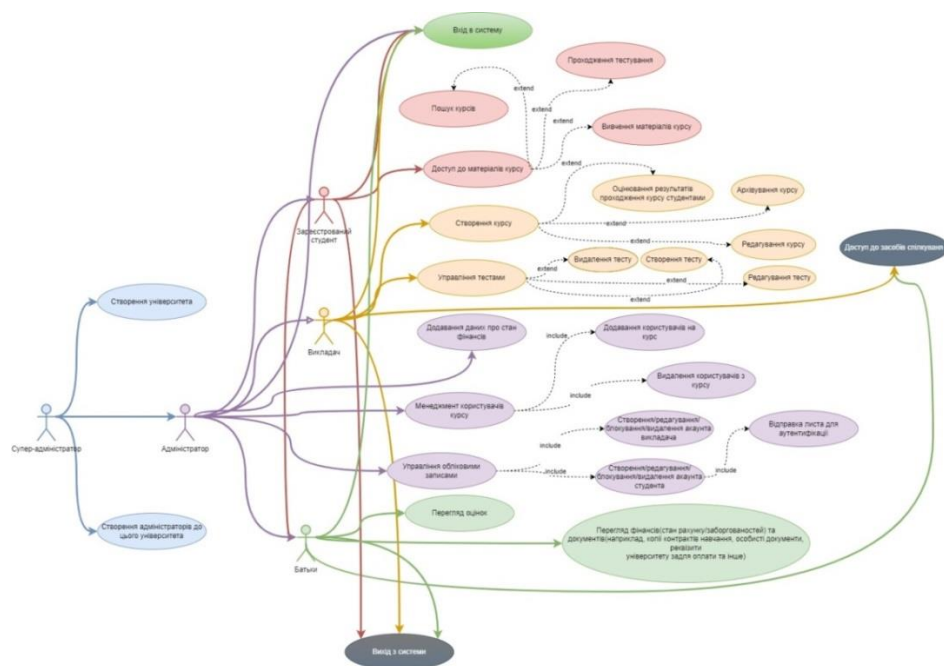


Рисунок 2.1 – Діаграма використання системи дистанційного навчання ЗВО

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Опис дійових осіб

Супер-адміністратор – створює університет у системі та визначає адміністраторів до цієї системи.

Адміністратор – веде інформацію по викладачах і студентах; відстежує список результатів тестування, видаляє застарілі.

Викладач – автор навчальних матеріалів (тестів). Має право створювати, змінювати, видаляти тести. Може переглядати результати проходження тесту.

Студент – проходить тестування і переглядає результат тестування, має доступ до матеріалів курсів.

Батьки – переглядають зміст курсів, на яких навчаються їх діти, та їх оцінки, мають доступ до власного кабінету, з метою огляду розділів курсу навчання, оцінок, фінансів (стан рахунку/заборгованостей) та документів (наприклад, копії контрактів навчання, особисті документи, реквізити університету задля оплати та інше).

Опис варіантів використання. Варіант використання «Вхід в систему СДН»

Передумови: відсутні.

Постумови: якщо варіант використання виконаний успішно, користувач входить в систему. В іншому випадку стан системи не змінюється.

Таблиця 2.1 – Головний розділ сценарію виконання варіанта використання «Вхід в систему»

Use case section	Comment
Use Case Name	Увійти до системи
Scope	Система дистанційного навчання
Level	Успішно пройти процес авторизації в системі, отримати доступ до власного профілю.
Primary Actor	Студент, викладач, батьки, адміністратори.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Продовження таблиці 2.1

Stakeholders and interests	<p>1. Студент: зацікавлений в точному швидкому доступу до необхідної інформації.</p> <p>2. Викладач: зацікавлений в швидкому завантаженні необхідної інформації на курс та зручному відображенні звітів щодо проходження студентами курсу.</p> <p>3. Батьки: зацікавлені у швидкому доступі до інформації(документів, стану рахунків тощо).</p> <p>4. Супер-адміністратор/адміністратор: зацікавлений у забезпеченні доступу та підтримці технологій, необхідних для дистанційного та змішаного навчання, а також швидкому розв'язанні технічних проблем під час освітнього процесу.</p>
Preconditions	<p>Адміністратор/студент/викладач/батьки мають бути зареєстровані, їх профіль має бути активований.</p> <p>Адміністратор/студент/викладач/батьки виконали вхід до системи (авторизація).</p>
Success guarantee	<p>1) Адміністратор/студент/викладач/батьки використовують браузер, що підтримує HTML5 та JavaScript.</p> <p>2) Адміністратор/студент/викладач/батьки мають використовувати стабільне підключення до мережі Інтернет.</p> <p>3) Адміністратор/студент/викладач/батьки мають вводити коректні дані профілю (логін та пароль).</p>
Main Success Scenario	<p>1) Користувач відкриває розділ авторизації;</p> <p>2) Користувач вводить свій логін та пароль у спеціалізовані поля;</p> <p>3) Користувач відкриває сторінку його профілем.</p>

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Кінець таблиці 2.1

Extensions	<p>– Користувач не вводить дані:</p> <p>1) Адміністратор/студент/викладач/батьки відкривають розділ авторизації;</p> <p>2) Адміністратор/студент/викладач/батьки залишають поля вводу даних незаповненими;</p> <p>3) Адміністратор/студент/викладач/батьки натискають кнопку вхід;</p> <p>4) Система видає повідомлення про незаповнені поля вводу даних</p> <p>– Користувач допускає помилку при заповненні даних:</p> <p>1) Адміністратор/студент/викладач/батьки натискають кнопку вхід;</p> <p>2) Адміністратор/студент/викладач/батьки вводять не вірні дані;</p> <p>3) Адміністратор/студент/викладач/батьки натискають кнопку вхід;</p> <p>4) Система повідомляє про невірно задані дані</p>
Special Requirements	1. Швидкість доступу до мережі Інтернет клієнту має бути більшою ніж 60 кб/сек.
Frequency of Occurrence	85%
Miscellaneous	<p>1. Чи необхідно проводити перевірку наявності логіну користувача у базі даних при неправильно введених даних.</p> <p>2. Вивчити можливість відновлення роботи системи після збою.</p>

Альтернативні сценарії:

1. Користувач не є зареєстрованим, система не може завантажити обліковий запис користувача.
2. Користувач не має доступу в Інтернет, тому система не працює коректно.
3. Користувач використовує браузер **Internet Explorer**, тому система не працює коректно.
4. Швидкість доступу до мережі Інтернет для клієнта менше, ніж 60 кб/сек.

Варіант використання «Створення тесту»

Передумови: користувач вибрав дію.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
Постумови: якщо варіант використання виконаний успішно, зміни, внесені викладачем, будуть збережені системою. В іншому випадку стан системи не змінюється.

Таблиця 2.2 – Головний розділ сценарію виконання варіанта використання «Створення тесту»

Варіант використання	Створення тесту
Актори	Викладач
Короткий опис	Створення нового тесту
Мета	Створити тест
Тип	Підлеглий
Посилання на інші варіанти використання	Виконується в рамках варіанту використання «Управління тестами».

Таблиця 2.3 – Типовий хід подій сценарію виконання варіанта використання «Створення тесту»

Дії актора	Відгук системи
3. Викладач обирає дію.	4. Система виводить форму для створення тесту.
5. Викладач вводить назву тесту і предмет. 6. Викладач по черзі вводить питання, відповіді на них, позначку про правильну відповідь. 7. Викладач дає команду на збереження тесту. Виняток №1. Викладач помилився в кількості при введенні варіантів відповідей.	8. Система створює тест і зберігає його.

Таблиця 2.4 – Винятки сценарію виконання варіанта використання «Створення тесту»

Дії актора	Відгук системи
Виняток №1. Викладач помилився в кількості при введенні варіантів відповідей.	
7. Викладач дає команду на збереження тесту.	8. Система попереджає про помилку, збереження не відбувається.
9. Викладач повертається до пункту 6 типового ходу подій.	

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
Варіант використання «Управління тестами»

Передумови: користувач повинен бути зареєстрований і володіти необхідними правами.

Постумови: якщо варіант використання виконаний успішно, зміни, внесені викладачем, будуть збережені системою.

Таблиця 2.5 – Головний розділ сценарію виконання варіанта використання «Управління тестами»

Варіант використання	Управління тестами
Актори	Викладач
Короткий опис	Створення, редагування, видалення тесту
Мета	Створити / редагувати / видалити тест
Тип	Базовий
Посилання на інші варіанти використання	Включає в себе варіанти використання: – створення тесту; – редагування тесту; – видалення тесту.

Таблиця 2.6 – Типовий хід подій сценарію використання «Управління тестами»

Дії актора	Відгук системи
1. Викладач хоче почати роботу з тестами.	2. Система запрошує необхідну дію (створити тест, відредагувати існуючий, видалити).
3. Викладач обирає дію.	4. Система виконує один з підлеглих потоків.

Варіант використання "Створення тесту"

Передумови: користувач вибрав дію.

Постумови: якщо варіант використання виконаний успішно, зміни, внесені викладачем, будуть збережені системою. В іншому випадку стан системи не змінюється.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Таблиця 2.7 – Головний розділ сценарію виконання варіанта
 використання «Створення тесту»

Варіант використання	Створення тесту
Актори	Викладач
Короткий опис	Створення нового тесту
Мета	Створити тест
Тип	Підлеглий
Посилання на інші варіанти використання	Виконується в рамках варіанту використання «Управління тестами».

Таблиця 2.8 – Типовий хід подій сценарію виконання варіанта
 використання «Створення тесту»

Дії актора	Відгук системи
3. Викладач обирає дію.	4. Система виводить форму для створення тесту.
5. Викладач вводить назву тесту і предмет. 6. Викладач по черзі вводить питання, відповіді на них, позначку про правильну відповідь. 7. Викладач дає команду на збереження тесту. Виняток №1. Викладач помилився в кількості при введенні варіантів відповідей.	8. Система створює тест і зберігає його.

Таблиця 2.9 – Винятки сценарію виконання варіанта використання
 «Створення тесту»

Дії актора	Відгук системи
Виняток №1. Викладач помилився в кількості при введенні варіантів відповідей.	
7. Викладач дає команду на збереження тесту.	8. Система попереджає про помилку, збереження не відбувається.
9. Викладач повертається до пункту 6 типового ходу подій.	

Варіант використання «Редагування тесту»

Передумови: користувач вибрав дію.

Постумови: якщо варіант використання виконаний успішно, зміни, внесені викладачем, будуть збережені системою. В іншому випадку стан системи не змінюється.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Таблиця 2.10 – Головний розділ сценарію виконання варіанта
 використання «Редагування тесту»

Варіант використання	Редагування тесту
Актори	Викладач
Короткий опис	Редагування тесту, створеного даним викладачем.
Мета	Відредагувати тест
Тип	Підлеглий
Посилання на інші варіанти використання	Виконується в рамках варіанту використання «Управління тестами».

Таблиця 2.11 – Типовий хід подій сценарію виконання варіанта
 використання «Редагування тесту»

Дії актора	Відгук системи
3. Викладач обирає дію по редагуванню тесту. Виняток №1. Перелік тестів даного викладача порожній.	4. Система виводить список вже створених даним викладачем тестів.
5. Викладач обирає потрібний тест.	6. Система завантажує обраний викладачем тест.
7. Викладач здійснює необхідну йому роботу. 8. Викладач дає команду на збереження тесту.	9. Система зберігає тест.

Таблиця 2.12 – Винятки сценарію виконання варіанта використання
 «Редагування тесту»

Дії актора	Відгук системи
Виняток №1. Перелік тестів даного викладача порожній.	
3. Викладач обирає дію по редагуванню тесту.	4. Система виводить попередження про те, що список тестів порожній.
5. Викладач повертається до пункту 3 типового перебігу подій (вибір дії).	

Варіант використання «Видалення тесту»

Передумови: користувач вибрав дію.

Постумови: якщо варіант використання виконаний успішно, зміни, внесені викладачем, будуть збережені системою. В іншому випадку стан системи не змінюється.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Таблиця 2.13 – Головний розділ сценарію виконання варіанта
 використання «Видалення тесту»

Варіант використання	Видалення тесту
Актори	Викладач
Короткий опис	Видалення тесту, створеного даним викладачем.
Мета	Видалити тест
Тип	Підлеглий
Посилання на інші варіанти використання	Виконується в рамках варіанту використання «Управління тестами».

Таблиця 2.14 – Типовий хід подій сценарію виконання варіанта
 використання «Видалення тесту»

Дії актора	Відгук системи
3. Викладач обирає дію по видаленню тесту. Виняток №1. Перелік тестів даного викладача порожній.	4. Система виводить список вже створених даним викладачем тестів.
5. Викладач обирає потрібний тест.	6. Система запитує підтвердження видалення.
7. Викладач підтверджує видалення. Виняток №2. Викладач скасовує видалення.	8. Система здійснює видалення обраних матеріалів.

Таблиця 2.15 – Винятки сценарію виконання варіанта використання
 «Видалення тесту»

Дії актора	Відгук системи
Виняток №1. Перелік тестів даного викладача порожній.	
3. Викладач обирає дію по видаленню тесту.	4. Система виводить попередження про те, що список тестів порожній.
9. Викладач повертається до пункту 3 типового перебігу подій (вибір дії).	
Виняток №2. Викладач скасовує видалення.	
7. Викладач не підтверджує видалення.	8. Система не виробляє дій по видаленню.
9. Викладач повертається до пункту 3 типового перебігу подій (вибір дії).	

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
Варіант використання «Проходження тестування»

Передумови: користувач повинен бути зареєстрований і володіти необхідними правами.

Постумови: якщо варіант використання виконаний успішно, студент отримає оцінку за проходження тесту, інформація про це буде збережена системою.

Таблиця 2.16 – Головний розділ сценарію виконання варіанта використання «Проходження тестування»

Варіант використання	Проходження тестування
Актори	Студент
Короткий опис	Студент вибирає тест, виконує його і отримує оцінку. Якщо раніше тест вже був пройдений даними студентом, то систем дає відбій.
Мета	Пройти тестування
Тип	Підлеглий
Посилання на інші варіанти використання	Виконується в рамках варіанту використання «Доступ до матеріалів курсу».

Таблиця 2.17 – Типовий хід подій сценарію виконання варіанта використання «Проходження тестування»

Дії актора	Відгук системи
1. Студент хоче почати тестування.	2. Система надає список доступних тестів.
3. Студент обирає тест.	4. Система перевіряє можливість проходження тесту.
Виняток №1. Студент вже проходив обраний тест.	5. Система надає тест для проходження.
6. Студент виконує тестові завдання	7. Система аналізує результати і зберігає їх в особистій статистиці студента.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Таблиця 2.18 – Винятки сценарію виконання варіанта використання
 «Проходження тестування»

Дії актора	Відгук системи
Виняток №1. Студент вже проходив обраний тест.	
3. Студент обирає тест.	4. Система перевіряє можливість проходження тесту. 5. Система виводить попередження про те, що вже пройдено даним студентом.
8. Студент повертається до пункту 3 типового перебігу подій (вибір тесту).	

Описані варіанти використання системи дистанційного навчання демонструють широкий спектр можливостей, які дозволяють ЗВО, студентам та викладачам гнучко адаптуватися до вимог сучасного освітнього середовища та досягати успіху у дистанційному навчанні.

2.3 Алгоритм роботи ПЗ

Алгоритм роботи ПЗ системи дистанційного навчання ЗВО може бути наступним:

1. *Реєстрація користувачів.* Нові користувачі повинні зареєструватися в системі, заповнивши необхідну інформацію про себе.
2. *Авторизація користувачів.* Користувачі повинні авторизуватися в системі, ввівши свій логін та пароль.
3. *Відображення курсів.* Користувачеві надається доступ до переліку курсів, які доступні для навчання.
4. *Навчання.* Користувач має доступ до матеріалів курсу, включаючи відео-лекції, тексти, тести та завдання. Користувачі можуть виконувати завдання та тести, які можуть бути автоматично перевірені системою.
5. *Взаємодія з викладачами.* Користувачі можуть звернутися до викладачів для отримання додаткової допомоги та відповідей на питання, використовуючи вбудовані інструменти зв'язку, такі як електронна пошта або чат.

Кафедра інженерії програмного забезпечення

Програмне забезпечення системи дистанційного навчання ЗВО

6. *Моніторинг прогресу.* Система дистанційного навчання може моніторити прогрес користувачів та надавати статистику щодо їх успішності.

7. *Оцінювання.* Користувачі можуть бути оцінені за свої успіхи в навчанні, а також можуть оцінювати викладачів та матеріали курсів.

8. *Адміністрування.* Адміністратори системи мають доступ до управління користувачами, курсами та матеріалами, а також до вирішення проблем та питань, які можуть виникнути в процесі використання системи.

Для зображення графічного представлення алгоритмів та процесів, зазвичай використовують блок-схеми, які позначають кожен етап роботи користувача з системою та різні варіанти розвитку подій [6]. Діаграма діяльності є аналогом блок-схеми та відображає послідовність дій, яка виконується під час процесу реалізації якогось варіанту використання або функціонування системи в цілому.

Діаграми діяльності (activity diagrams) відображають послідовність дій, що виконується в процесі реалізації певного варіанту використання або функціонування системи в цілому. Діаграми діяльності є аналогом блок-схеми будь якого алгоритму. Вони, як і діаграми станів та переходів, відображаються у вигляді орієнтованого графу, вершинами якого є дії, а ребрами – переходи між діями [7].

Діяльність (activity) є частковим випадком стану (state) без назви, який має одну вхідну подію (OnEntry action). Тому для кожної діяльності назва складається з дієслова та декількох пояснюючих слів, наприклад «Розрахувати заробітну платню» чи «Перевірити результати запиту».

Події (events) на переходах діаграми діяльності не задаються, оскільки вважається, що перехід від однієї дії до іншої здійснюється безумовно. Гранична умова (guard condition) використовується лише для визначення дії, до якої переходить керування у випадку неоднозначності. Тобто, якщо з даної вершини на діаграмі діяльності можна перейти до декількох інших вершин для всіх переходів необхідно визначити граничну умову [8].

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Характеристика дії (action) для переходу також не має сенсу, оскільки

всі дії на діаграмі представлені вершинами графу.

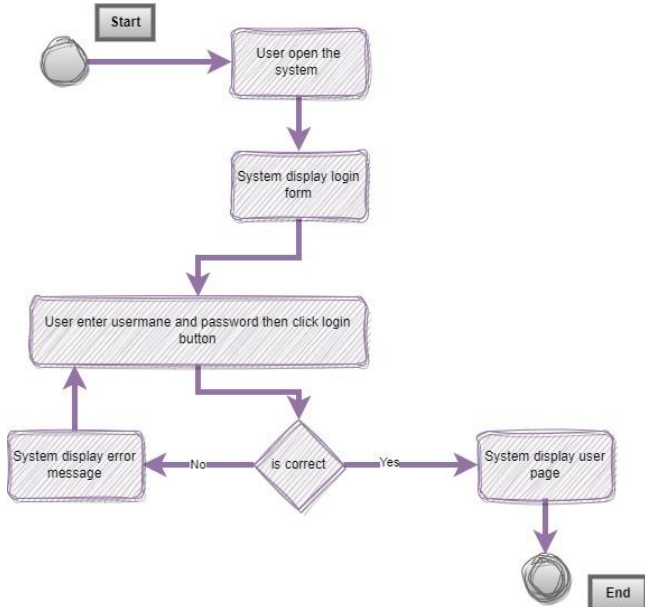


Рисунок 2.2 – Діаграма діяльності для створення облікового запису

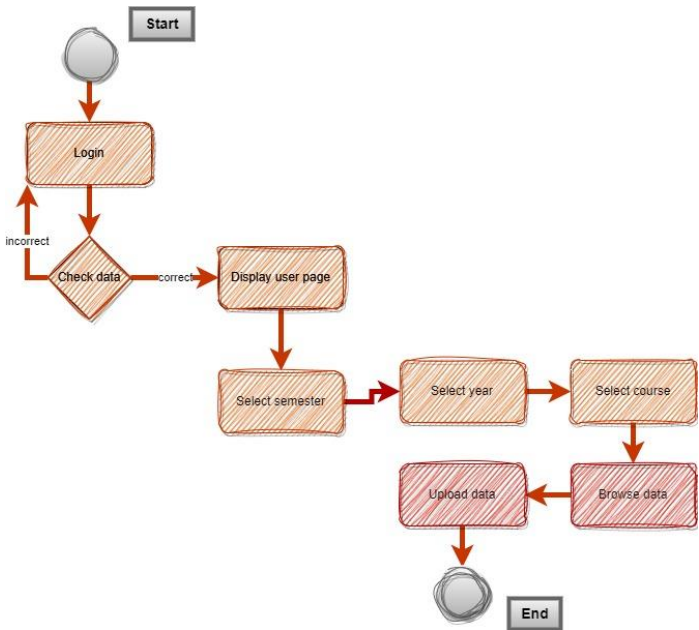


Рисунок 2.3 – Діаграма діяльності для завантаження матеріалу

Застосування діаграм діяльності сприяє зрозумілому визначенню кроків, виконанню завдань та оптимізації робочих процесів.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

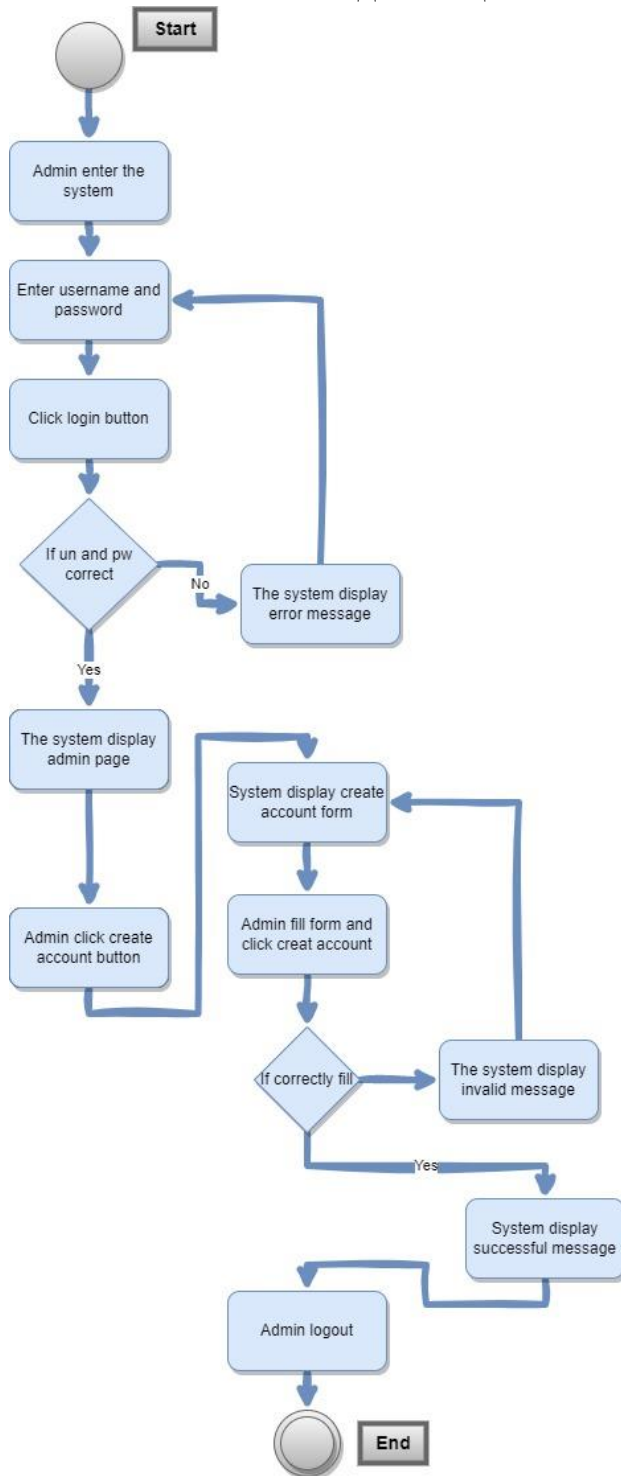


Рисунок 2.4 – Діаграми діяльності для створення облікового запису

Отже, діаграми діяльності є потужним інструментом для моделювання послідовності дій, процесів та взаємодії між об'єктами в системі, що допомагає зрозуміти та візуалізувати робочі процеси та логіку виконання завдань.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
2.4 Розробка діаграми розгортання

На рис. 2.5 представлена діаграма розгортання застосунку.

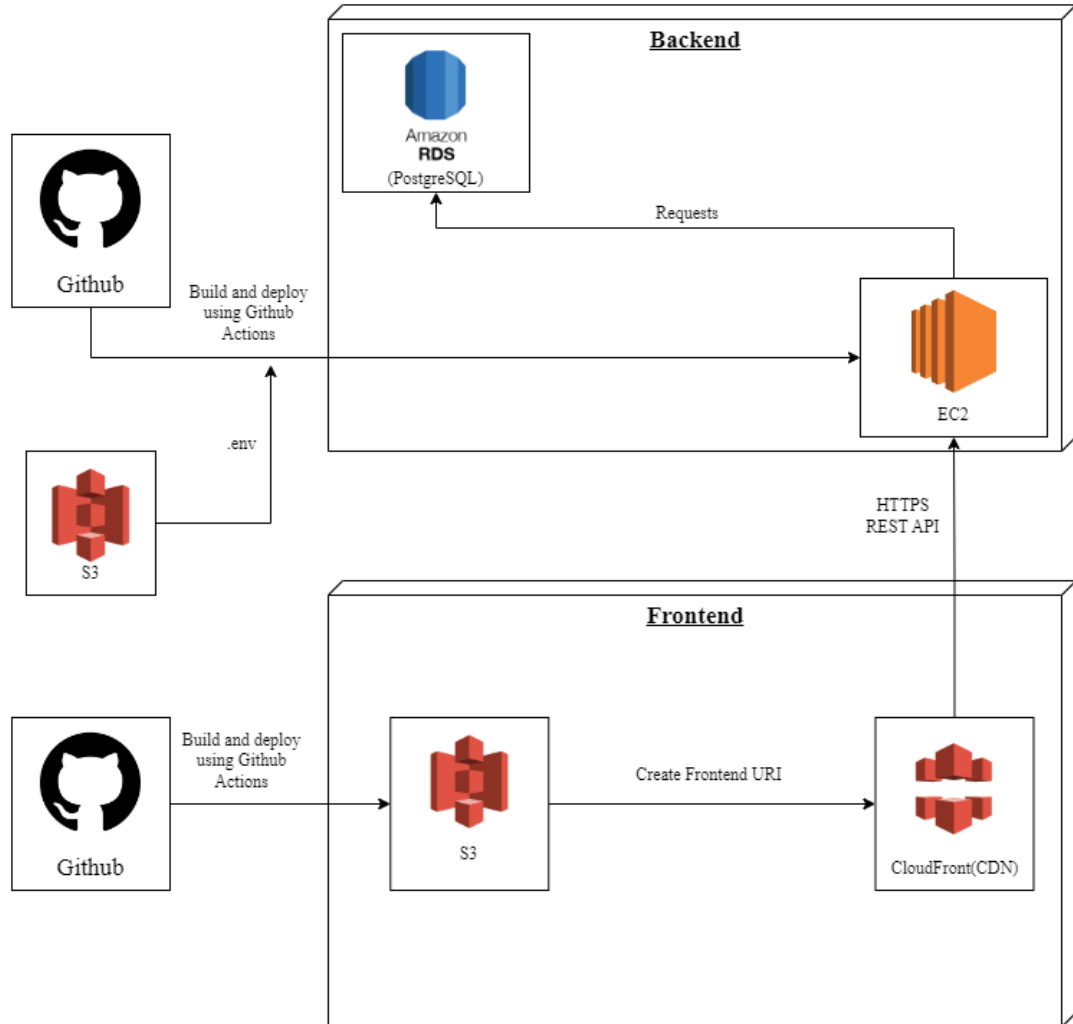


Рисунок 2.5 – Діаграма розгортання системи дистанційного навчання ЗВО

Опис компонентів діаграми розгортання

AMAZON ELASTIC COMPUTE CLOUD (AMAZON EC2)

Amazon Elastic Compute Cloud (Amazon EC2) пропонує найширшу та найглибшу обчислювальну платформу з більш ніж 500 інстансами та набором новітніх процесорів, сховищ, мереж, операційних систем та моделей покупок, забезпечуючи належну відповідність потребам конкретного робочого навантаження. Це перший великий хмарний провайдер, який підтримує роботу процесорів Intel, AMD та Arm, єдина хмара з інстансами

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
EC2 Mac на вимогу та з мережею Ethernet 400 Гбіт/с. Тут пропонується найкраще співвідношення ціни та продуктивності машинного навчання, а також найнижчу вартість інстансів логічних висновків у хмарі. На AWS виконується більше робочих навантажень SAP, високопродуктивних обчислень (HPC), машинного навчання та Windows, ніж у будь-якій іншій хмарі [9].

Приклади використання

1. Запуск хмарних та корпоративних застосунків.

Amazon EC2 пропонує безпечну, надійну, високопродуктивну та економічну обчислювальну інфраструктуру для виконання складних бізнес-завдань.

2. Масштабування для програм HPC.

Доступ до вимог інфраструктури та ресурсів, необхідних для швидкого та економічного запуску програм HPC.

3. Розробка для платформ Apple.

Створення, тестування та випуск робочих навантажень macOS на вимогу. Доступ до середовищ за лічені хвилини, динамічне масштабування ресурсів у разі потреби та цінову перевагу при оплаті AWS у міру використання.

4. Навчання та розгортання ML-застосунків.

Amazon EC2 надає найширший вибір обчислювальних та мережевих (до 400 Гбіт/с) сервісів, а також сервісів зберігання даних, спеціально розроблених для оптимізованого співвідношення ціни та продуктивності у проєктах машинного навчання [9].

AMAZON CLOUDFRONT

Amazon CloudFront – це сервіс мережі доставки контенту (CDN), створений для високої продуктивності, безпеки та зручності розробників [10].

Приклади використання

Кафедра інженерії програмного забезпечення

Програмне забезпечення системи дистанційного навчання ЗВО

1. Надайте швидкі безпечні вебсайти.

Охоплюйте глядачів по всьому світу протягом мілісекунд завдяки вбудованому стиску даних, можливостям периферійних обчислень та шифруванню на рівні поля.

2. Прискорюйте доставку динамічного контенту та API.

Оптимізуйте доставку динамічного вебконтенту за допомогою спеціально створеної багатофункціональної глобальної мережної інфраструктури AWS, яка підтримує периферійне завершення та WebSockets.

3. Поточкова передача відео в режимі реального часу або на вимогу.

Оперативно запускайте потокову передачу, відтворюйте їх послідовно та доставляйте високоякісне відео на будь-який пристрій завдяки AWS Media Service та інтеграції AWS Elemental.

4. Поширюйте виправлення та оновлення.

Автоматично масштабуйтеся для швидкої доставки оновлень програмного забезпечення, виправлень ігор та IoT бездротової мережі (OTA) [10].

AMAZON RELATIONAL DATABASE SERVICE (AMAZON RDS)

Amazon Relational Database Service (Amazon RDS) – це набір керованих сервісів, який спрощує налаштування, використання та масштабування бази даних у хмарі. Виберіть одне із семи популярних ядер: версія Amazon Aurora, сумісна з MySQL, версія Amazon Aurora, сумісна з PostgreSQL, MySQL, MariaDB, PostgreSQL, Oracle та SQL Server – і розгорніть його в локальному середовищі за допомогою Amazon RDS в AWS Outposts [11].

Приклади використання

1. Створюйте мобільні та інтернет-застосунки.

Забезпечте розвиток програм за рахунок високої доступності, пропускної спроможності та масштабованості сховища. Скористайтеся

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
перевагою гнучкої моделі ціноутворення з оплатою за фактом використання,
яка підходить для різних варіантів використання програм.

2. Перехід до керованих баз даних.

Впроваджуйте інновації та створюйте нові програми за допомогою Amazon RDS, не обтяжуючи себе трудомістким, складним та дорогим процесом управління базами даних.

3. Звільнення від застарілих баз даних.

Звільніть себе від дорогих комерційних баз даних із складними умовами ліцензування – виконайте міграцію на Amazon Aurora. Виконавши міграцію на Aurora, ви отримуєте масштабованість, продуктивність та доступність комерційних баз даних у десять разів дешевше [11].

AMAZON SIMPLE STORAGE SERVICE (AMAZON S3)

Amazon Simple Storage Service (Amazon S3) – це сервіс зберігання об'єктів, що пропонує найкращі в галузі показники продуктивності, масштабованості, доступності та безпеки даних. Клієнти будь-якої величини та з будь-якої промислової галузі можуть зберігати та захищати необхідний обсяг даних для практично будь-якого прикладу використання. Наприклад, для озер даних, хмарних та мобільних застосунків. Вигідні класи сховища та прості у використанні інструменти адміністрування дозволяють оптимізувати витрати, організувати дані та точно налаштувати обмеження доступу відповідно до потреб бізнесу чи законодавчих вимог [12].

Приклади використання

1. Створення озера даних.

Запустіть програми для аналітики великих даних, штучного інтелекту (AI), машинного навчання (ML) та високопродуктивних обчислень (HPC), щоб отримати корисну інформацію.

2. Резервне копіювання та відновлення критично важливих даних.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Виконуйте нормативи за часом відновлення (RTO), точками відновлення (RPO) та законодавчими вимогами за допомогою надійних можливостей реплікації S3.

3. Архівація даних за найнижчою ціною.

Переміщуйте архівні дані до класів сховища Amazon S3 Glacier, щоб скоротити витрати, позбутися операційних складнощів та отримати нову корисну інформацію.

4. Запуск програм з оптимізацією для хмари.

Створюйте швидкі, потужні мобільні та оптимізовані для хмари інтернет-програми, які автоматично масштабуються до найвищої доступної конфігурації [12].

GITHUB ACTIONS

GitHub Actions – це платформа безперервної інтеграції та безперервної доставки (CI/CD), яка дозволяє вам автоматизувати збірку, тестування та розгортання. Ви можете створювати робочі процеси, які створюють і тестують кожен запит на отримання до вашого сховища, або розгортати об'єднані запити на отримання в робочому середовищі.

GitHub Actions виходить за рамки просто DevOps і дозволяє запускати робочі процеси, коли у вашому сховищі відбуваються інші події. Наприклад, ви можете запустити робочий процес, щоб автоматично додавати відповідні мітки щоразу, коли хтось створює нову проблему у вашому сховищі.

GitHub надає віртуальні машини Linux, Windows і macOS для запуску ваших робочих процесів, або ви можете розмістити власні програми запуску у власному центрі обробки даних або хмарній інфраструктурі [13].

2.5 Побудова та використання діаграм послідовності

На рис. 2.6 представлена діаграма послідовності для основного потоку подій варіанту використання «Вхід в систему». Під рисунком наводиться короткий опис даної діаграми.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

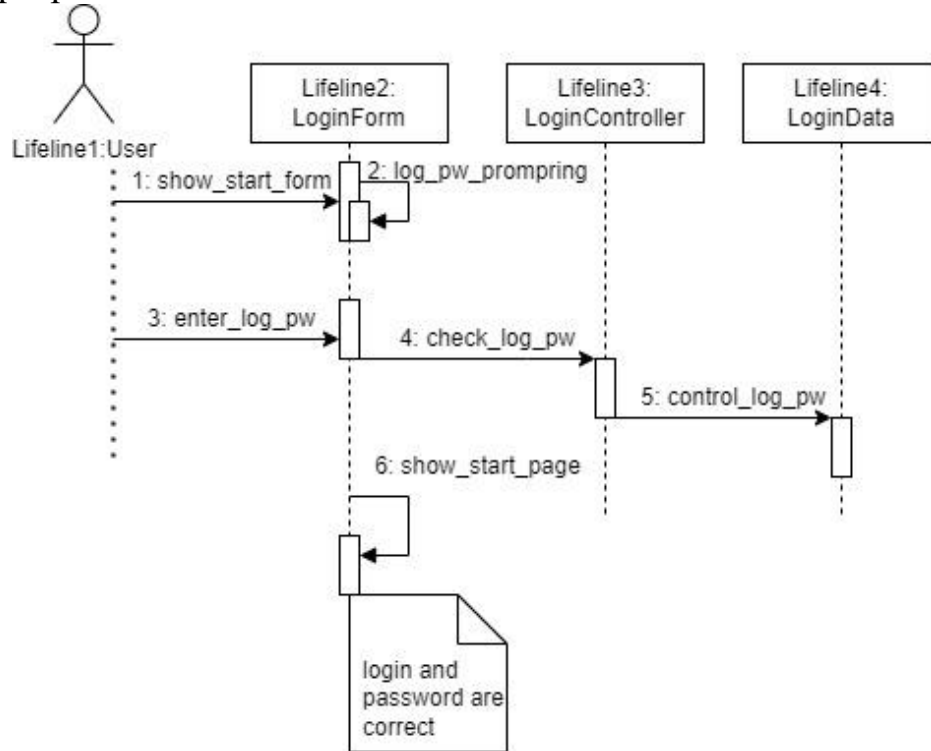


Рисунок 2.6 – Діаграма послідовності для основного потоку подій варіанту використання «Вхід у систему»

Даний варіант використання починається з ініціативи користувача, який може звернутися до меню за допомогою миші або клавіатури. Це викликає появу спеціальної форми, в яку користувач повинен ввести свій логін і пароль. Після отримання цих даних, система перевіряє їх, використовуючи інформацію, яку заздалегідь вніс у систему адміністратор. Якщо перевірка пройшла успішно, система надає користувачеві відповідний рівень доступу залежно від його ролі і виводить на екран відповідну інформацію.

Створення діаграми послідовності для основного потоку подій варіанту використання «Управління тестами»

На рис. 2.7 – 2.10 представлені діаграми послідовності для основного потоку подій варіанту використання «Управління тестами».

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

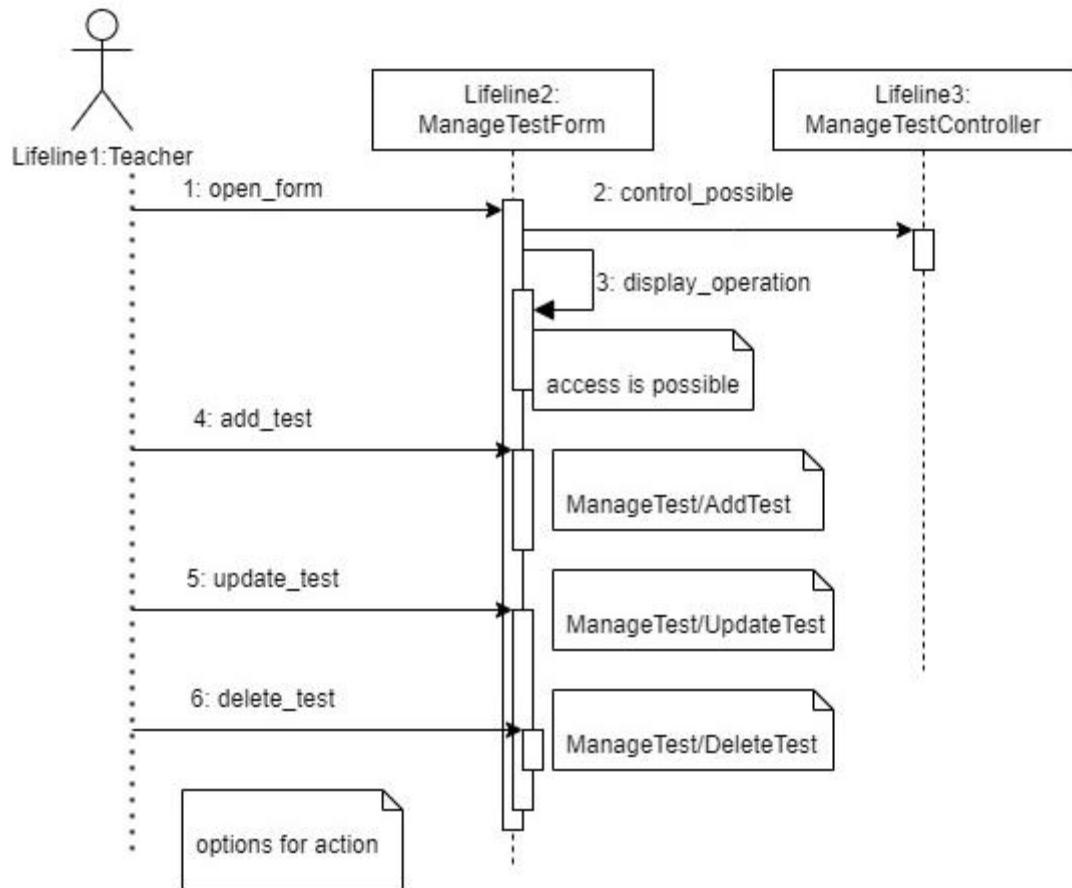


Рисунок 2.7 – Діаграма послідовності для основного потоку подій варіанту використання «Управління тестами»

Ініціатором даного варіанту використання є викладач. Після успішної авторизації в системі, викладачу відкривається форма, яка містить перелік доступних дій. Якщо функція повертає значення, яке вказує на можливість вибору, то викладач може обрати необхідну дію з запропонованих варіантів. Після отримання вибору викладача, система відкриває відповідну форму.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
Створення діаграми послідовності для додаткового потоку подій
варіанту використання «Створення тесту»

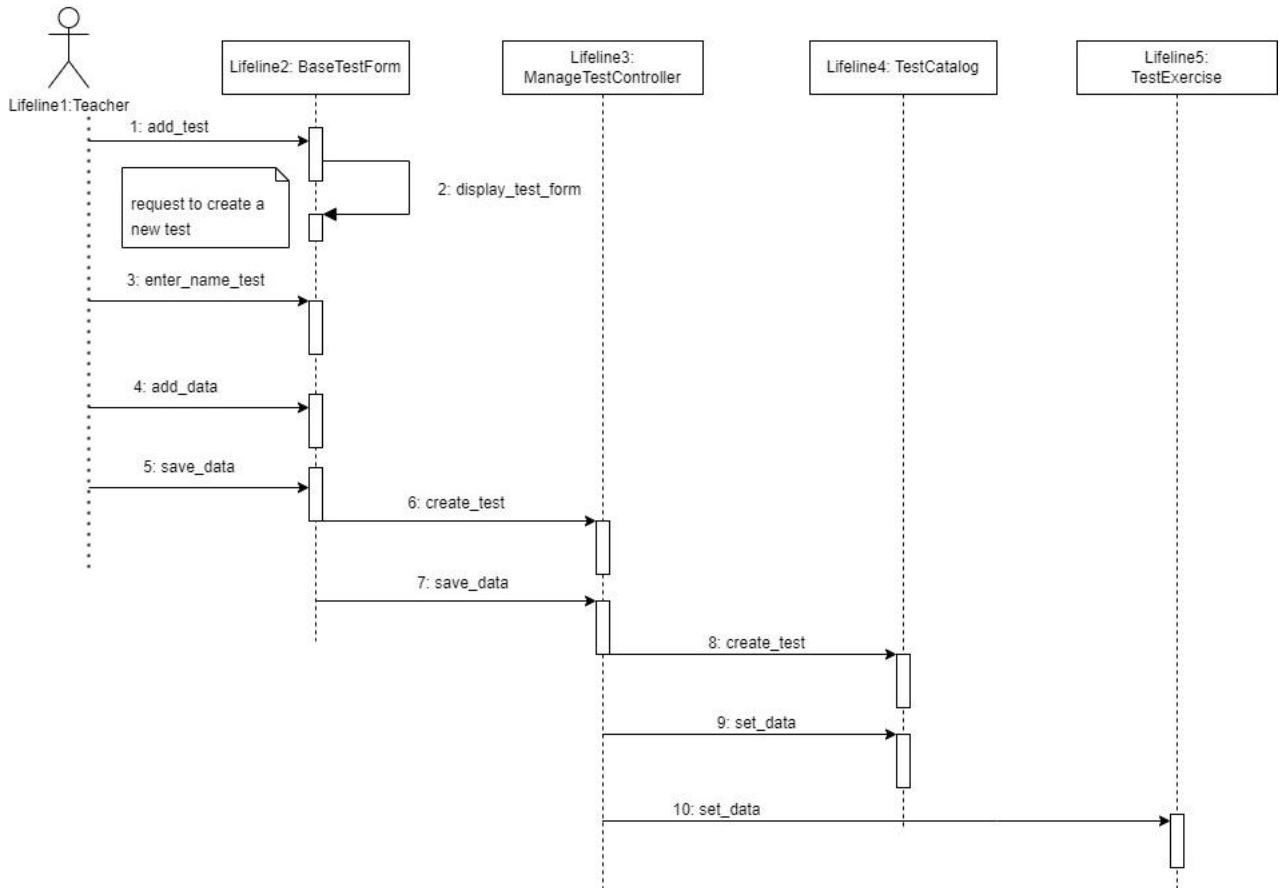


Рисунок 2.8 – Діаграма послідовності для додаткового потоку подій варіанту використання «Створення тесту»

Викладач є ініціатором створення додаткового потоку подій шляхом обрання дії «Створення тесту». Після цього відкривається форма для введення загальної інформації про тест та тестових завдань з варіантами відповідей. Після отримання команди на збереження даних, вони перевіряються і зберігаються в базі даних тестів.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
Створення діаграми послідовності для додаткового потоку подій
варіанту використання «Редагування тесту»

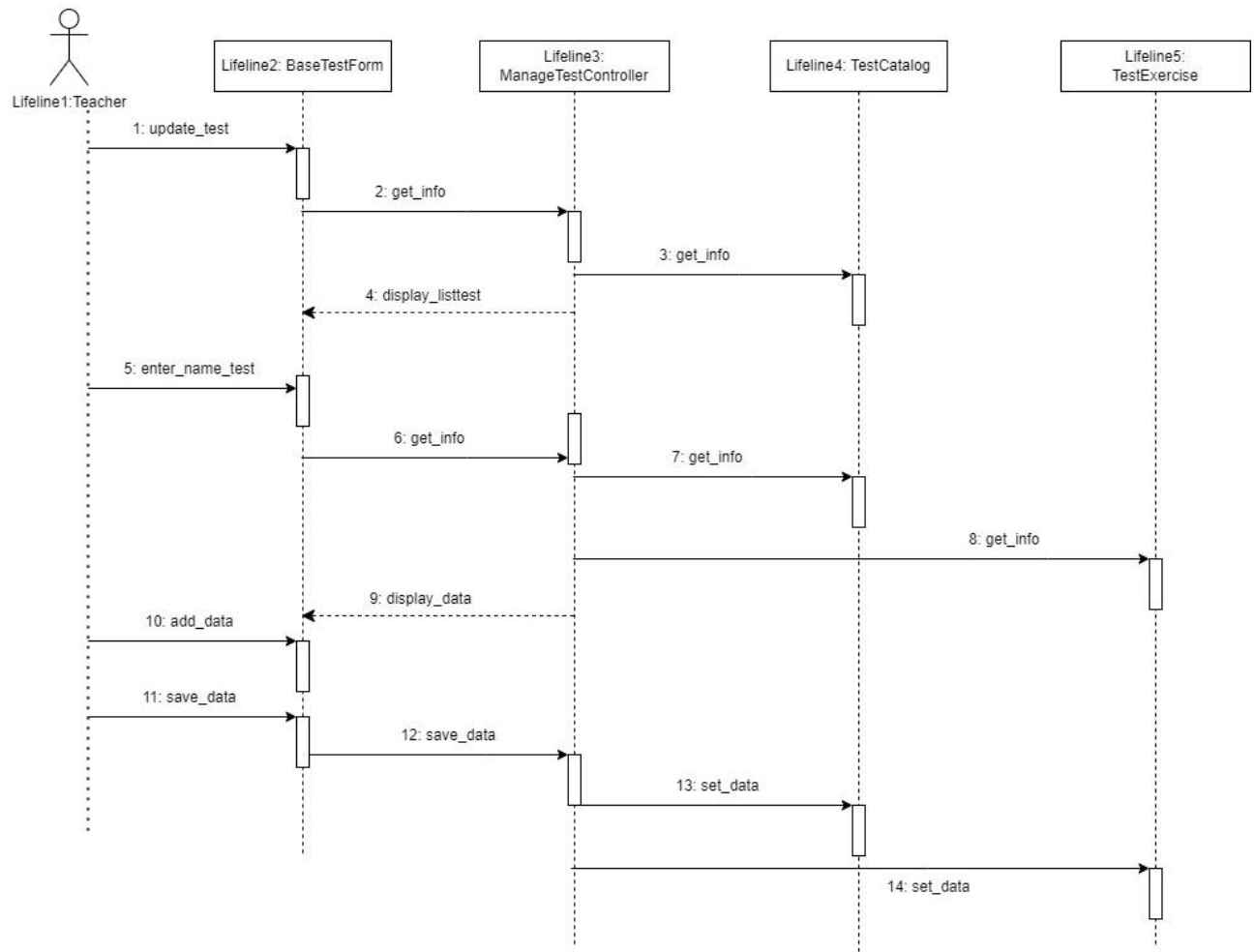


Рисунок 2.9 – Діаграма послідовності для додаткового потоку подій варіанту використання «Редагування тесту»

Викладач є ініціатором додаткового потоку подій, коли він обирає дію «Редагування тесту». Далі відбувається відкриття форми, яка містить список тестів, створених раніше Викладачем. Після вибору тесту для редагування, система завантажує відповідну форму, що містить загальну інформацію про тест та завдання з варіантами відповідей. Після внесення змін користувач дає команду на збереження даних. Результати редагування перевіряються та зберігаються в базі даних тестів.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
**Створення діаграми послідовності для додаткового потоку подій
 варіанту використання «Видалення тесту»**

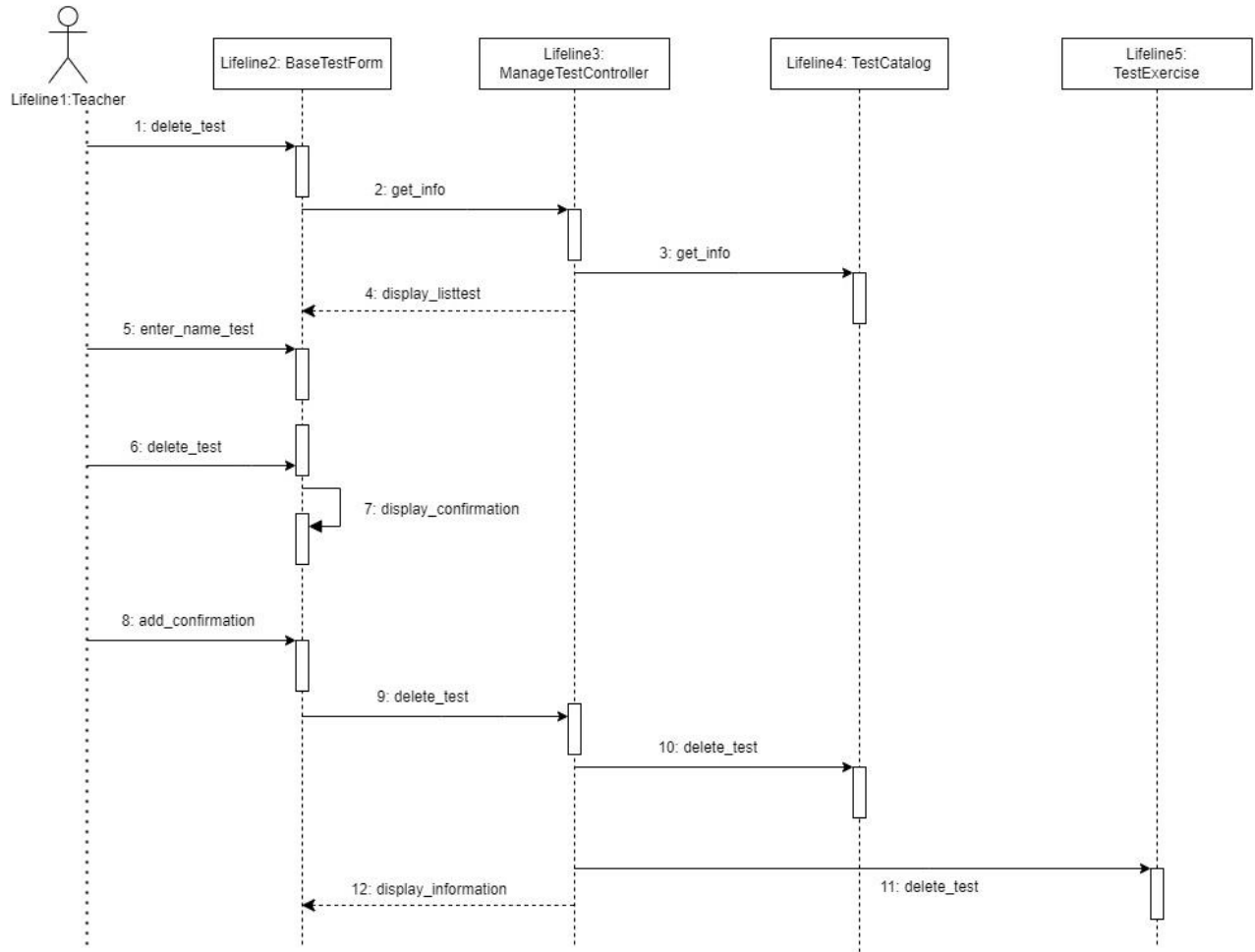


Рисунок 2.10 – Діаграма послідовності для додаткового потоку подій
 варіанту використання «Видалення тесту»

Викладач є ініціатором додаткового потоку подій, коли він обирає дію «Видалення тесту». Після цього відкривається форма зі списком тестів, які були раніше створені Викладачем. Після вибору тесту для видалення система запитує підтвердження видалення. Якщо підтвердження надійшло, тоді тест видаляється з бази даних. Після успішного видалення тесту виводиться повідомлення.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Висновки до розділу 2

В другому розділі кваліфікаційної роботи бакалавра проведено моделювання системи дистанційного навчання ЗВО. Розглянуто етапи реалізації проєкту. Створено різні USE CASE для системи дистанційного навчання. Розписано короткі, поверхневі та альтернативні сценарії використання.

Описано функції та ролі всіх дійових осіб. Розроблено алгоритм роботи ПЗ та діаграми розгортання для системи дистанційного навчання. Також в другому розділі детально описано різні компоненти діаграми розгортання системи, а саме Amazon Elastic Compute Cloud (Amazon EC2) та приклади його використання, Amazon CloudFront та приклади його використання, Amazon Relational Database Service (Amazon RDS) та приклади його використання, Amazon Simple Storage Service (Amazon S3) та приклади його використання, GitHub Actions. Побудовано та описано діаграми послідовності для системи дистанційного навчання ЗВО.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО

3 ПРОЄКТУВАННЯ СИСТЕМИ ТА ОГЛЯД СТЕКУ ТЕХНОЛОГІЙ

3.1 Розробка UML-діаграм

Огляд розробки UML-діаграм є необхідною складовою процесу розробки програмного забезпечення, оскільки UML надає стандартну мову моделювання для візуалізації, специфікації, конструювання та документування складних систем. Цей огляд допомагає покращити якість та ефективність розробки, забезпечує узгодженість у проєкті та зменшує ризик виникнення помилок під час розробки системи [14].

UML-діаграми поділяються на 14 типів (рис. 3.1):

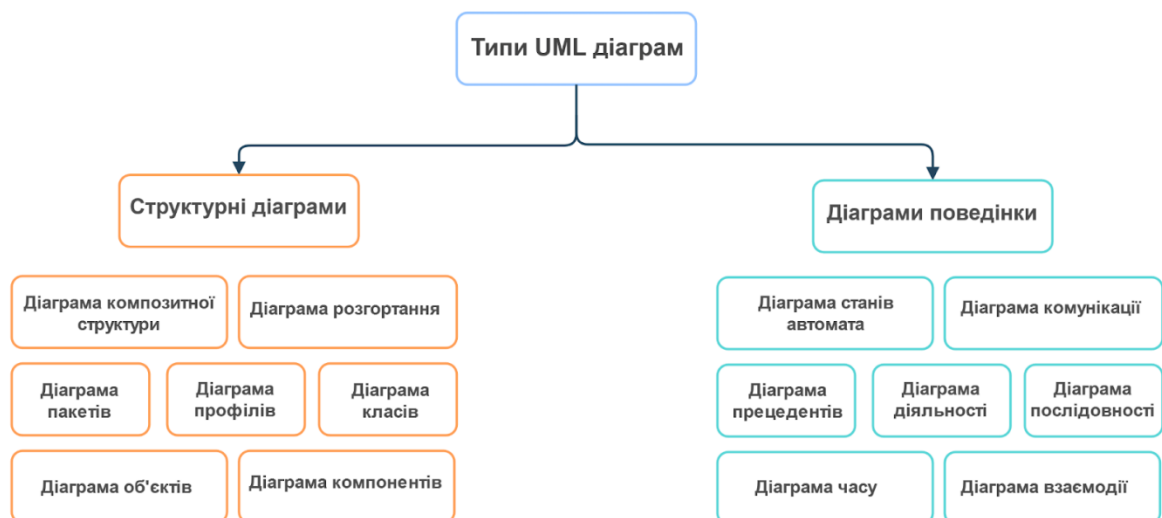


Рисунок 3.1 – Різновид UML-діаграм

Під час моделювання системи дистанційного навчання ЗВО у другому розділі створено діаграму діяльності, діаграму розгортання, діаграму послідовності та діаграму прецедентів (діаграму варіантів використання) тому, що кожна з цих діаграм відображає різні аспекти системи та допомагає зрозуміти її функціонування, взаємозв'язки між компонентами та послідовність виконання операцій.

Діаграма діяльності надає уявлення про послідовність дій та процесів, які відбуваються в системі. Діаграма розгортання дозволяє візуалізувати розташування та взаємозв'язки між апаратними та програмними

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
компонентами системи. Діаграма послідовності показує, як об'єкти взаємодіють між собою та обмінюються повідомленнями в певному порядку. Діаграма прецедентів (або діаграма варіантів використання) допомагає ідентифікувати основні функціональні можливості системи та сценарії їх використання користувачами.

Використання цих діаграм допомагає краще розуміти систему, виявляти потенційні проблеми та забезпечувати відповідність вимогам та очікуванням користувачів.

3.1.1 Діаграми станів та переходів

Діаграми станів та переходів (statechart diagrams) спільно з діаграмами діяльності та взаємодії відображають конкретний сценарій, що відбувається під час функціонування системи в цілому або певної її частини.

Діаграма станів відображає різні стани, в яких може перебувати об'єкт або система, а також показує переходи між цими станами. Ця діаграма моделює поведінку системи та дозволяє спостерігати, як система реагує на події та які зміни відбуваються у її станах [14].

Стан (state) – це логічна сутність, яка використовується для моделювання певної ситуації, дії чи процесу. Кожен стан має ім'я та список внутрішніх дій. Зазвичай для іменування станів використовуються іменники, наприклад: «Введення паролю», «Перевірка параметрів» тощо. Список внутрішніх дій містить перерахування дій, які виконуються під час перебування системи або об'єкту в даному стані.

Користувачу «Супер-адміністратор» доступні два основні стани, в яких може перебувати курс: перегляд курсу та управління курсом (рис. 3.2). Обидва стани є компонентами системи.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

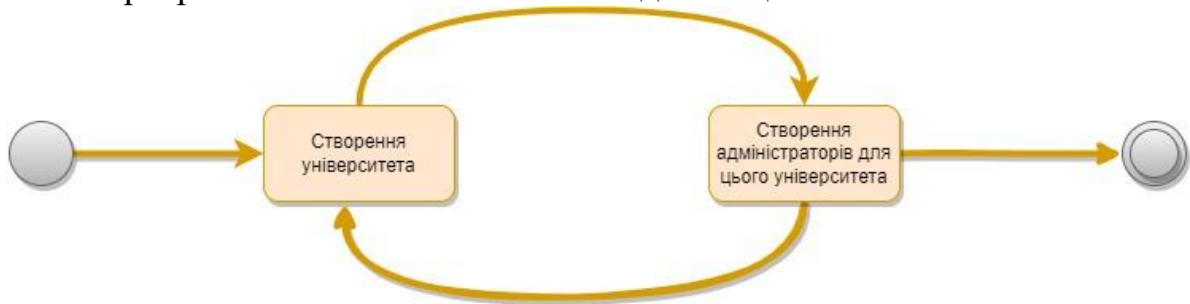


Рисунок 3.2 – Діаграма станів користувача «Супер-адміністратор»

На рис. 3.3 наведено діаграму стану «Управління курсом». Користувачеві «Викладач» доступні переходи на додавання курсу, редагування курсу та архівування курсу. З додавання та редагування курсу можна перейти у стан збереження курсу. Зі стану «Архівування курсу» можна перейти до архівування даних курсу.



Рисунок 3.3 – Діаграма для стану «Керування курсом»

Стан «Перегляд курсу» (рис. 3.4) включає переходи до перегляду історії зміни сторінки, перегляду посилань на сторінку, редагування на сторінки, додавання сторінки, перегляду сторінки, збереження сторінки.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

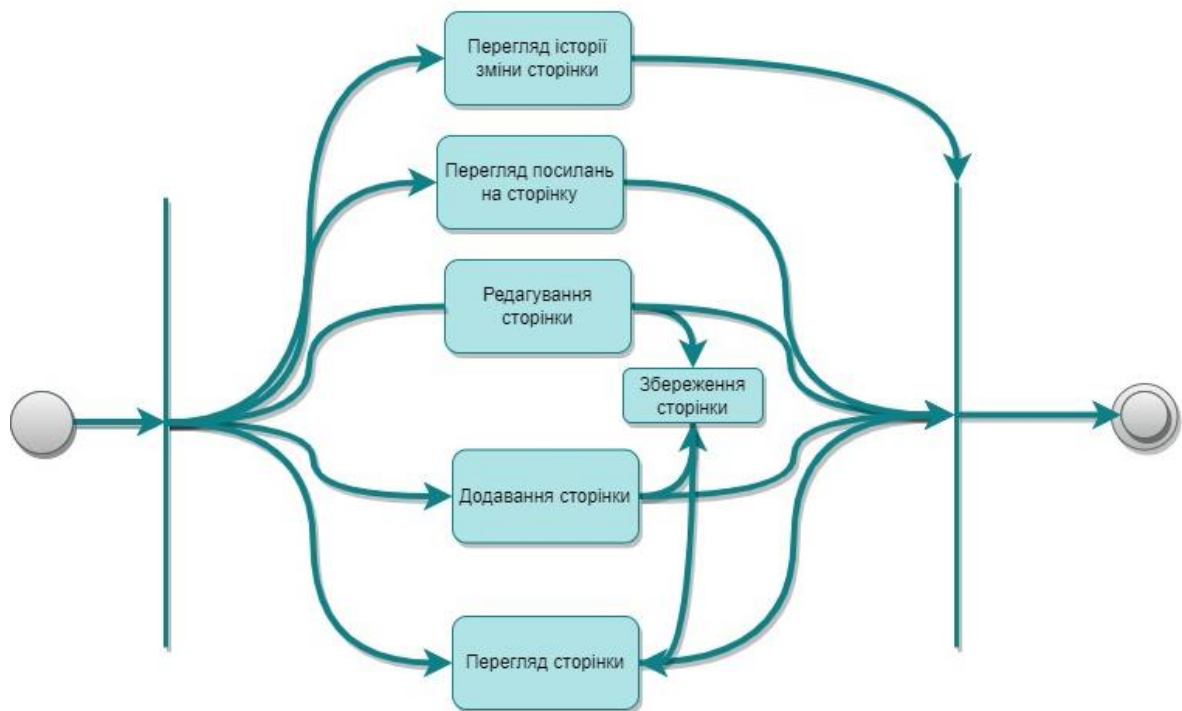


Рисунок 3.4 – Діаграма стану «Перегляд сторінок»

Стан «Перегляд даних курсу» (рис. 3.5) визначає переходи до стану:

- перегляд найбільш відвідуваних сторінок;
- перегляд необхідних сторінок;
- перегляд одиноких сторінок;
- перегляд нових сторінок;
- перегляд структури сторінок;
- перегляд оновлених сторінок;
- перегляд часто змінюваних сторінок.

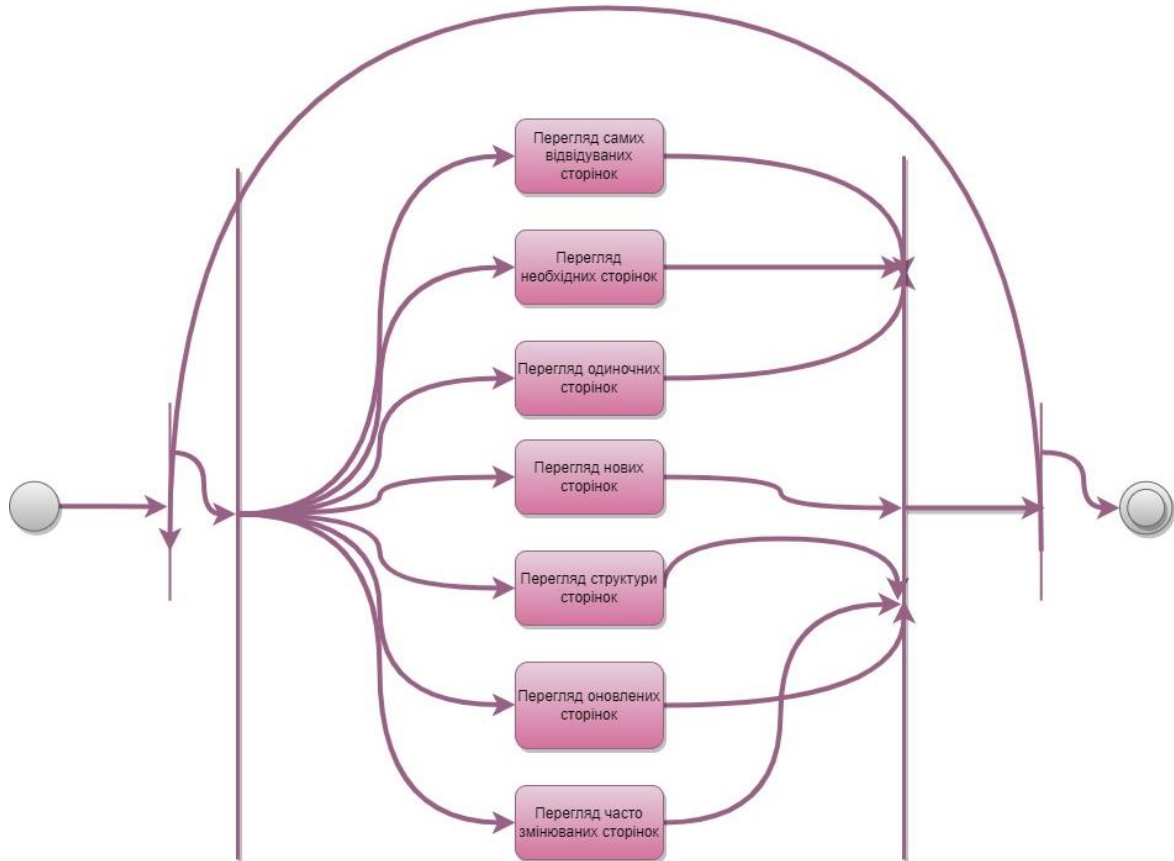


Рисунок 3.5 – Діаграма стану «Перегляд даних курсу»

Отже, діаграми стану є корисним інструментом для моделювання та візуалізації поведінки системи, допомагаючи зрозуміти різні стани системи та переходи між ними.

3.1.2 Діаграми компонентів

Діаграми компонентів (component diagrams) використовуються для візуального відображення фізичного представлення програмної системи. Вони складаються з набору елементів, що називаються компонентами. Кожен компонент має своє ім'я, мову реалізації та перелік призначених класів. Компоненти можуть бути представлені фізично як окремі файли, директорії або програмні продукти [16].

В діаграмі компонентів системи дистанційного навчання ЗВО показані компоненти, інтерфейси, порти та зв'язки, пов'язані з курсами, розкладами тощо. Цей тип діаграми часто використовується в компонентній розробці

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 (Component-Based Development) для опису систем з сервісно-орієнтованою
 архітектурою (Service-Oriented Architecture). UML-схема компонентів
 системи керування електронним навчанням відображає організацію та
 підключення фізичних компонентів системи.

Компоненти, які складають діаграму компонентів системи дистанційного навчання ЗВО, включають (рис. 3.6):

- компонент курсу;
- розкладу;
- студентського компонента;
- предметного компонента тощо.

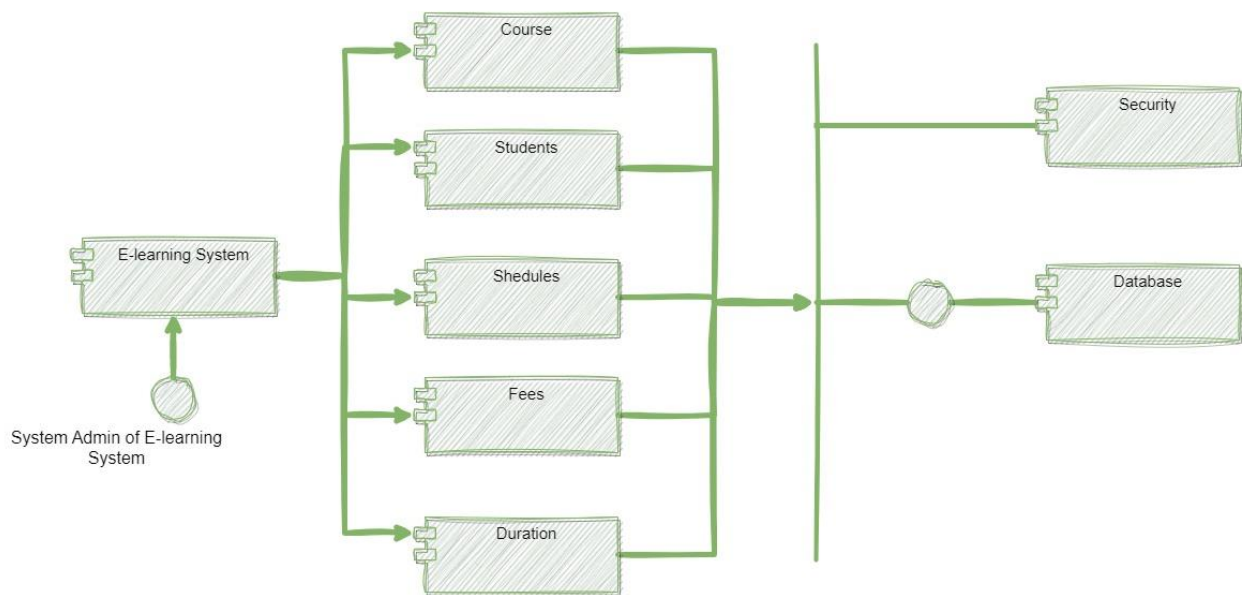


Рисунок 3.6 – Діаграми компонентів системи керування електронним навчанням

Діаграми впровадження (deployment diagrams) використовуються для моделювання фізичної інфраструктури та апаратної реалізації програмного комплексу. Вони надають візуальне представлення фізичної моделі предметної області [17].

При розгляді діаграми впровадження для системи дистанційного навчання ЗВО, яка була описана раніше на діаграмі компонентів, можна

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 встановити, що програмний комплекс має тришарову архітектуру (рис. 3.7).
 Перший шар представлений робочими столами з пристроями. Другий шар
 включає вебсервер, на якому виконується бізнес-логіка програми. Третій шар
 складається з сервера баз даних, до якого здійснюється доступ з вебсервера.

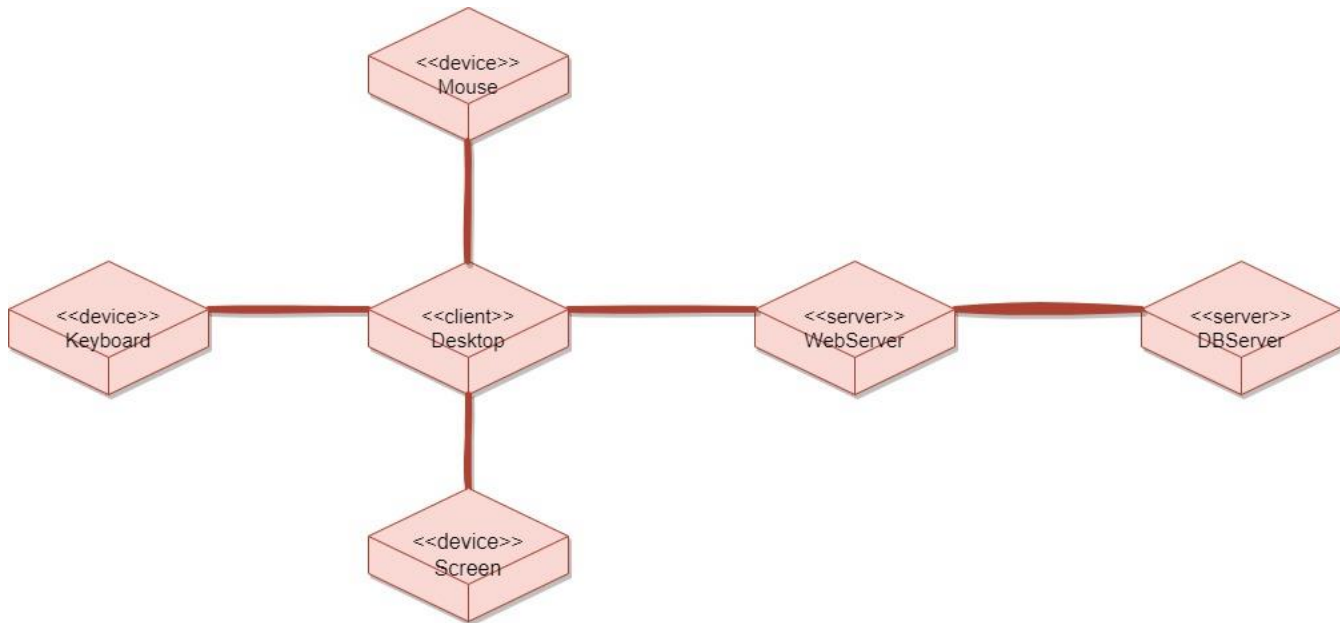


Рисунок 3.7 – Діаграма впровадження для обраного програмного продукту

Отже, діаграми впровадження є ефективним інструментом для візуалізації та планування процесу впровадження системи або програмного продукту.

3.1.3 Діаграма пакетів

Діаграма пакетів (package diagram) є одним з типів діаграм в рамках мови моделювання UML (Unified Modeling Language). Вона використовується для візуалізації організації та структури системи шляхом групування елементів моделі в пакети [18]. На рис. 3.8 зображена діаграма пакетів системи дистанційного навчання ЗВО.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

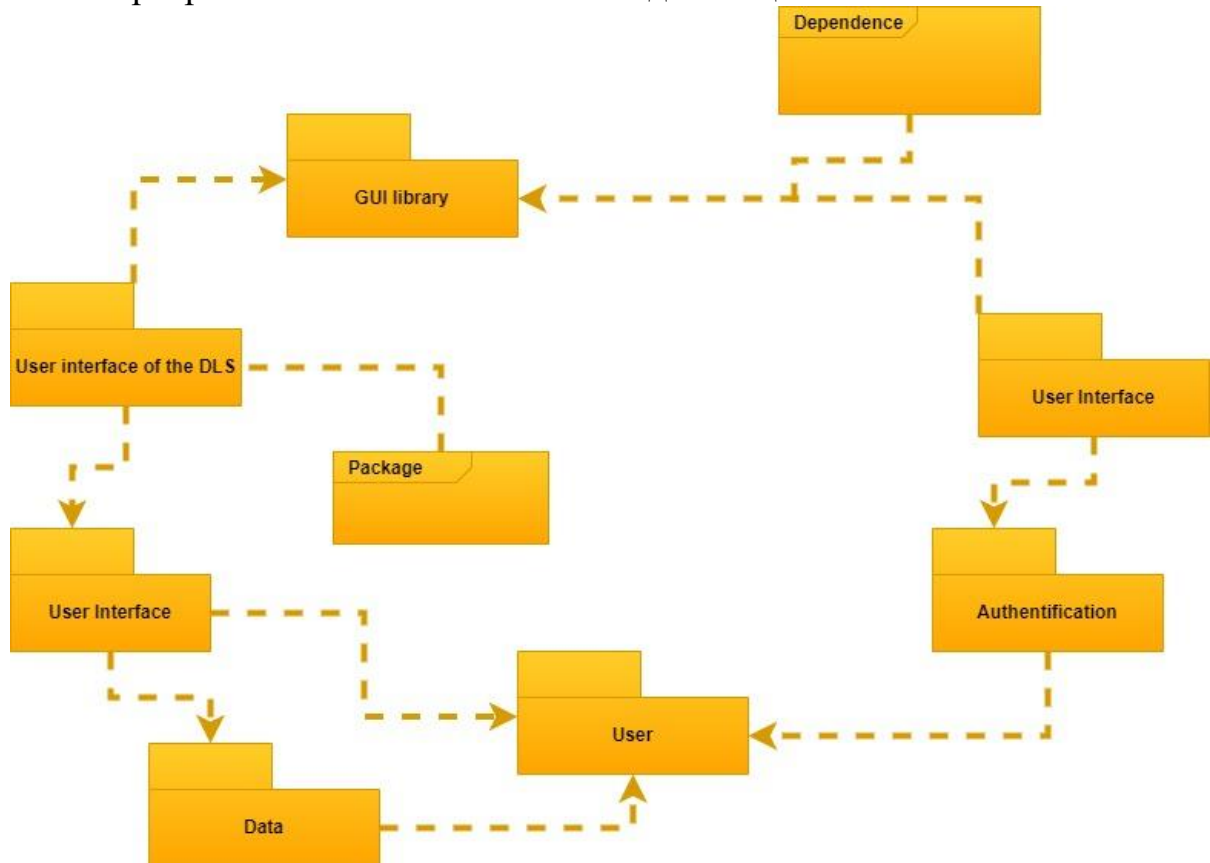


Рисунок 3.8 – Діаграма пакетів системи дистанційного навчання ЗВО

У діаграмі пакетів пакети представлені у вигляді прямокутників, які містять ім'я пакету. Вони допомагають організувати компоненти системи в логічні групи та показати залежності між ними.

Діаграма пакетів дозволяє зосередитись на високорівневій структурі системи та логічному розподілі функціональності між пакетами. Вона полегшує розуміння складної системи та сприяє її керованості, документації та підтримці.

Узагальнено, діаграма пакетів допомагає візуалізувати та організувати структуру системи, показати залежності між її компонентами та сприяє кращому розумінню та управлінню системою під час розробки та підтримки.

3.2 Огляд технологій

У процесі розробки системи дистанційного навчання ЗВО було використано конкретний набір технологій, що включає (рис. 3.9):

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

- Для програмування використано дві мови: JavaScript і TypeScript. JavaScript є популярною мовою програмування для веброзробки, а TypeScript є надмножиною JavaScript, яка надає додаткові можливості статичної типізації та підтримку сучасних функцій програмування.
- React як технологію front-end розробки. React є популярною бібліотекою JavaScript для побудови користувацьких інтерфейсів вебзастосунків. Вона дозволяє розробникам ефективно створювати і керувати компонентами інтерфейсу користувача [19].
- NestJS як технологію back-end розробки. NestJS є фреймворком для побудови масштабованих і модульних серверних застосунків на базі Node.js. Він надає структурований підхід до розробки, підтримку модульності і інструменти для створення RESTful API.



Рисунок 3.9 – Стек технологій з логотипами та коротким описом

Цей стек технологій був обраний з метою реалізації функціональності системи дистанційного навчання ЗВО, забезпечення швидкості та ефективності розробки, а також забезпечення потужності та масштабованості серверної частини. Використання цих технологій дозволяє розробникам

2023 р. Гайсюк А. В. 121 – КРБ.01 – 409.21910906

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
ефективно створювати функціональний та сучасний дистанційний
навчальний застосунок.

3.2.1 Мови програмування

У процесі вибору технологій для розробки системи дистанційного навчання ЗВО було обумовлено використання JavaScript і TypeScript як мов програмування. JavaScript є широко використовуваною мовою програмування вебзастосунків, має велику спільноту розробників і багатий екосистему інструментів та бібліотек. TypeScript, який є надмножиною JavaScript, був обрано з метою покращення типізації та роботи зі складними структурами даних, що сприяє підвищенню якості коду та полегшує підтримку та розробку системи у майбутньому.

Для front-end частини обрано фреймворк React. Цей вибір обумовлений його популярністю, широким спектром підтримуваних бібліотек і компонентів, а також швидкістю розробки завдяки використанню компонентної моделі. React надає зручні інструменти для побудови ефективних та інтерактивних користувацьких інтерфейсів.

У розробці back-end частини обрано фреймворк NestJS. Його вибір був обумовлений його модульністю, гнучкістю і підтримкою сучасних підходів до розробки. NestJS базується на Node.js і надає розробникам можливість швидко створювати масштабовані та надійні серверні застосунки з використанням популярних стандартів і шаблонів [19].

У контексті бази даних обрано PostgreSQL, оскільки він є потужною та надійною реляційною базою даних, яка підтримує розширення і масштабованість. PostgreSQL надає різноманітні функції для роботи з даними, а також має велику спільноту активних користувачів і підтримує стандарти SQL. Це забезпечує надійність та ефективність зберігання та обробки даних для системи дистанційного навчання ЗВО.

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Для середовища розробки обрано Visual Studio Code (VS Code). Це

популярний та потужний редактор коду з багатофункціональними можливостями, широким спектром розширень та підтримкою для багатьох мов програмування. VS Code надає зручне інтерфейсне середовище для розробки, засоби для реалізації автоматичного завершення коду, налагодження та інші корисні функції, що полегшують роботу розробників у процесі створення системи дистанційного навчання ЗВО.

Загалом, обумовленість вибору JavaScript і TypeScript, фреймворка React, NestJS для розробки back-end частини, PostgreSQL для бази даних та VS Code для середовища розробки полягає у їхній популярності, підтримці сучасних підходів до розробки, надійності та ефективності, що допомагає забезпечити високу якість та продуктивність системи дистанційного навчання ЗВО.

3.2.2 Технології front-end розробки

Вибір React для front-end частини системи дистанційного навчання ЗВО обумовлений його широкою підтримкою та активним розвитком. React є однією з найпопулярніших технологій для front-end розробки, що означає, що існує велика кількість розробників та ресурсів, доступних для вирішення проблем та підтримки проекту [19].



Рисунок 3.10 – Популярність пошукового терміна React відносно найвищої точки на графіку

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

На рис. 3.10 показано популярність пошукового терміна відносно найвищої точки на графіку для певного регіону та періоду часу:

- 100 – це пік популярності терміна;
- 50 означає, що популярність терміна вдвічі менша;
- 0 означає, що було замало даних про цей термін.

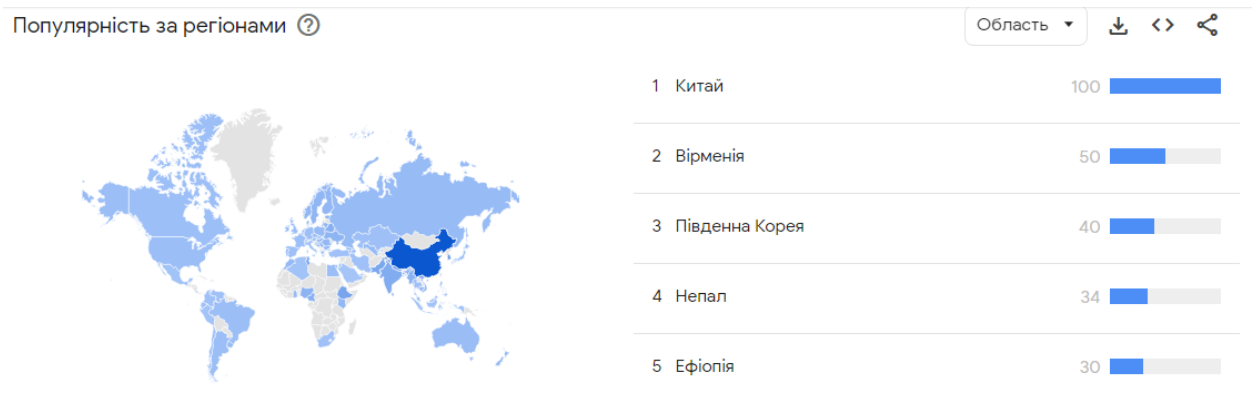


Рисунок 3.11 – Популярність за регіонами

На основі аналізу Google Trends, можна встановити популярність React за регіонами в контексті пошукових запитів (рис. 3.11). Зауважимо, що рейтинги популярності можуть відрізнятися в залежності від регіону та часового періоду.

3.2.3 Технології back-end розробки

Технологія back-end розробки, використана в системі дистанційного навчання ЗВО, є NestJS. NestJS – це фреймворк для розробки серверних застосунків на базі Node.js [19]. Основні причини вибору NestJS для розробки back-end частини системи дистанційного навчання можуть включати:

1. **Ефективність.** NestJS використовує ефективний та масштабований підхід до розробки, що дозволяє створювати швидкі та потужні серверні застосунки. Він базується на платформі Node.js, яка відома своєю продуктивністю та швидкодією.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

2. **Типізація.** NestJS побудований на мові програмування TypeScript, що надає можливість використовувати типізацію і підвищує безпеку та стабільність розробки. TypeScript дозволяє виявляти помилки на етапі компіляції та полегшує розробку та підтримку коду.

3. **Модульність.** NestJS пропонує модульну структуру розробки, що сприяє організації коду та полегшує управління складними проєктами. Модулі дозволяють групувати функціональність за логічними блоками та полегшують розширення та підтримку системи.

4. **Вбудована підтримка HTTP.** NestJS надає вбудовану підтримку для роботи з HTTP-запитами та розробки API. Він надає зручний спосіб визначення маршрутів, обробки запитів та валідації даних.

5. **Розширюваність.** NestJS надає можливість розширювати функціональність системи шляхом використання плагінів, middleware і інших розширень. Це дозволяє додавати нові функції та інтегрувати зовнішні сервіси з легкістю.



Рисунок 3.12 – Популярність пошукового терміна NestJS відносно найвищої точки на графіку

На рис. 3.12 показано популярність пошукового терміна відносно найвищої точки на графіку для певного регіону та періоду часу.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

Популярність за регіонами ?

Область ▾ ⬇ ⏪ ⏩



1	Китай	100	<div style="width: 100%;"></div>
2	Вірменія	89	<div style="width: 89%;"></div>
3	Південна Корея	58	<div style="width: 58%;"></div>
4	Ефіопія	55	<div style="width: 55%;"></div>
5	Молдова	44	<div style="width: 44%;"></div>

Рисунок 3.13 – Популярність пошукового терміна NestJS відносно найвищої точки на графіку

На основі аналізу Google Trends, можна встановити популярність NestJS за регіонами в контексті пошукових запитів (рис. 3.13). Рейтинги популярності можуть відрізнятися в залежності від регіону та часового періоду.

Висновки до розділу 3

В третьому розділі кваліфікаційної роботи бакалавра спроектовано систему та зроблено огляд стеку технологій. Розроблено UML-діаграми, а саме діаграми станів та переходів, діаграми пакетів, діаграми впровадження та діаграми компонентів. Описано технології, які використовувалися при розробці системи. Розглянуто та детально описано обрані мови програмування. Зроблено опис технологій front-end та back-end розробки та обґрунтовано їх використання. Відстежено пошукову популярність обраних технологій за часом та регіонами у Google Trends.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗВО

Поставлене завдання вимагає від себе використання сучасних технологій та інструментів для побудови вебзастосунку. Тому для його вирішення обрано одні з найкращих технологій в області клієнт-серверної розробки – React для розробки front-end частини та NestJS для розробки back-end частини. Ці технології забезпечують простоту використання, зрозумілу семантику, а також вони займають мало ресурсів. У якості редактору коду був обраний VS Code. Вибір зумовлений надійною роботою цього середовища і наявністю в ньому вбудованих інструментів, модулів та розширень, які полегшують процес розробки.

4.1 Опис дизайну вебзастосунку

В системі є користувачі з різними ролями. На вході в систему користувач може побачити сторінку авторизації (рис. 4.1).

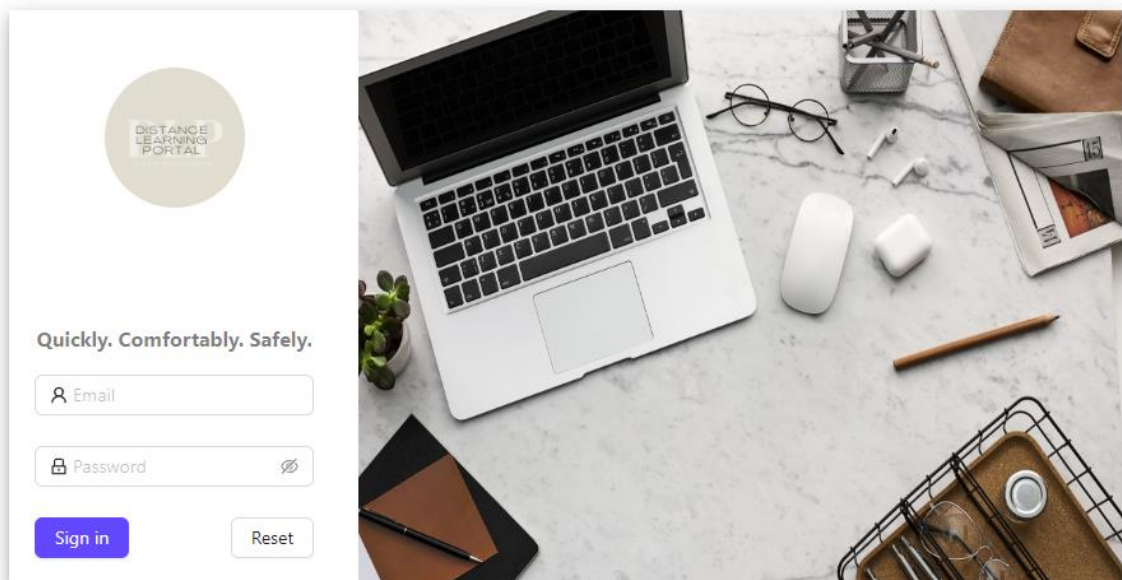


Рисунок 4.1 – Сторінка авторизації

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
На поточній сторінці користувач має заповнити форму відповідними даними, а саме поштою та паролем. На етапі авторизації система автоматично відкриває необхідну сторінку у відповідності з роллю користувача. Для того аби користувачі розпочали повноцінну роботу в системі супер-адміністратору необхідно зайти в свій обліковий запис у вебзастосунку (рис. 4.2) і створити відповідний університет з адміністраторами (рис. 4.3).

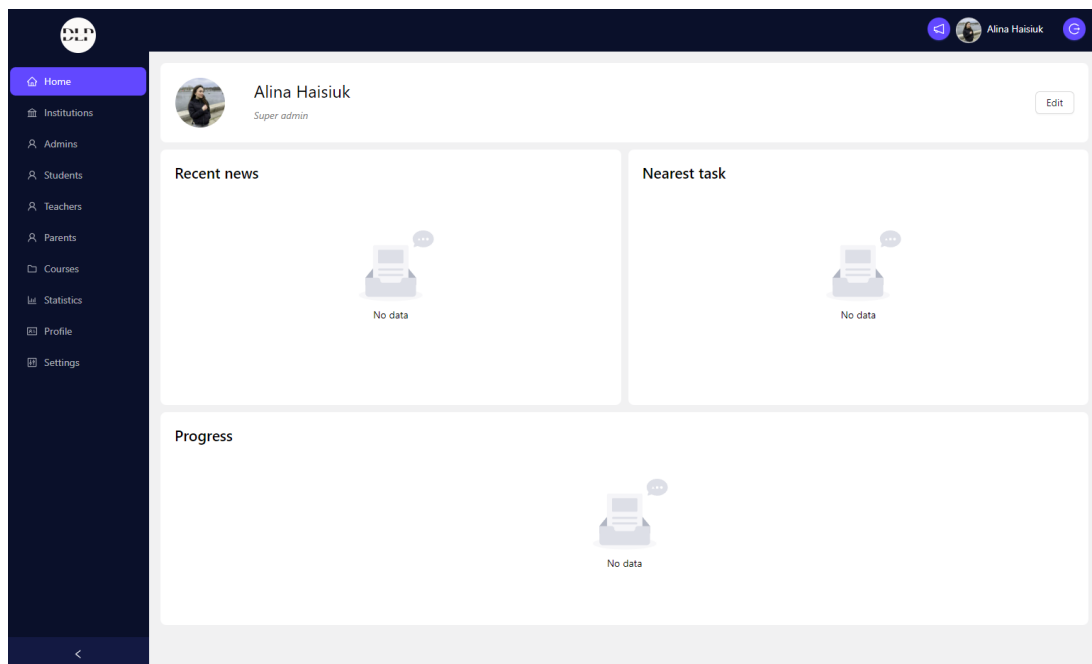


Рисунок 4.2 – Обліковий запис супер-адміністратора

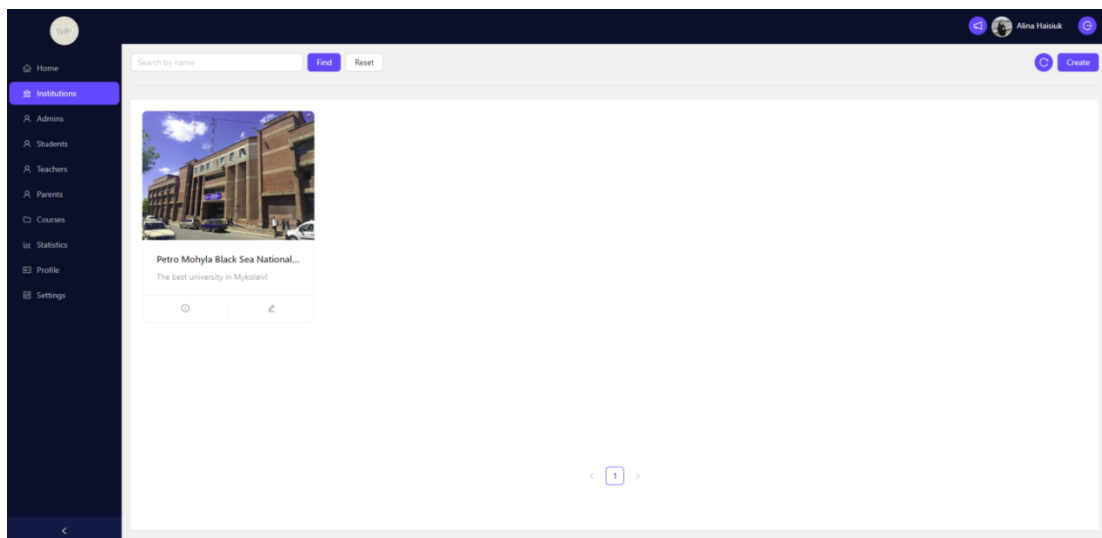
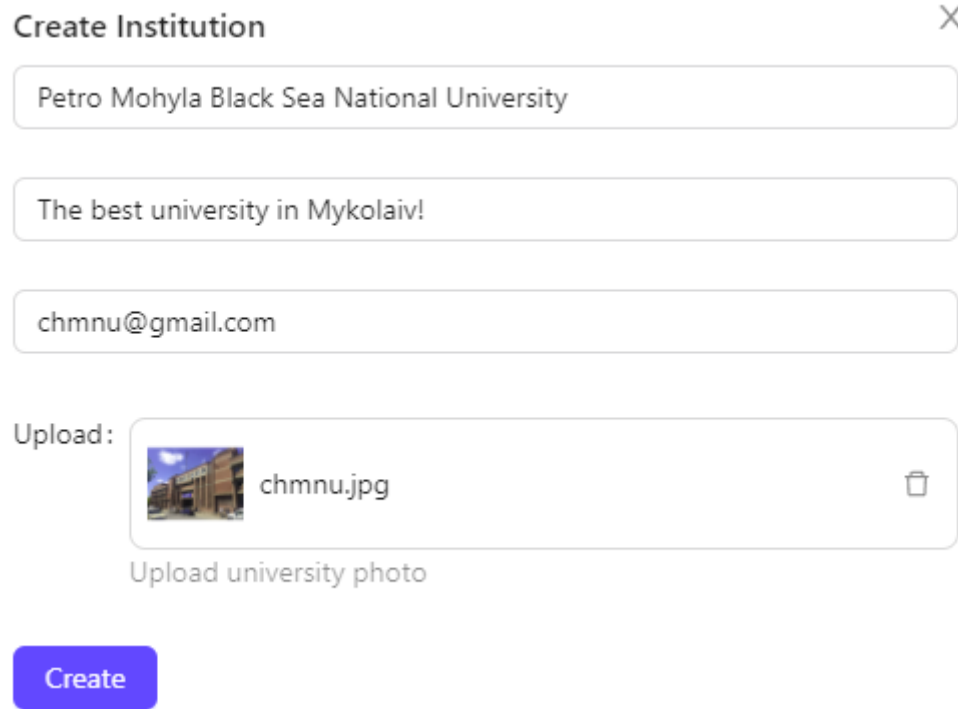


Рисунок 4.3 – Створення відповідного університету з адміністраторами

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Для створення сутності університету адміністратору необхідно заповнити форму з наступними полями: назва, опис, пошта та фото (рис. 4.4).





Create Institution X

Petro Mohyla Black Sea National University

The best university in Mykolaiv!

chmnu@gmail.com

Upload:  chmnu.jpg 

Upload university photo

Create

Рисунок 4.4 – Заповнення форми для створення сутності університета

Після створення цієї сутності супер-адміністратор може перейти на сторінку «Admins» та створити адміністраторів до конкретного університету заповнивши поля: пошта, ПІБ, назва університету, до якого він відноситься, а також тимчасовий пароль, який адміністратору необхідно змінити на власний після першої авторизації в системі (рис. 4.5).

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

Create Admin X

Email

First name

Last name

Temp password

Institution

Create

Рисунок 4.5 – Створення адміністраторів до конкретного університета

Після чого на вказану пошту адміністраторів університету надходить лист з тимчасовим паролем, що дає змогу зайти в систему (рис. 4.6).

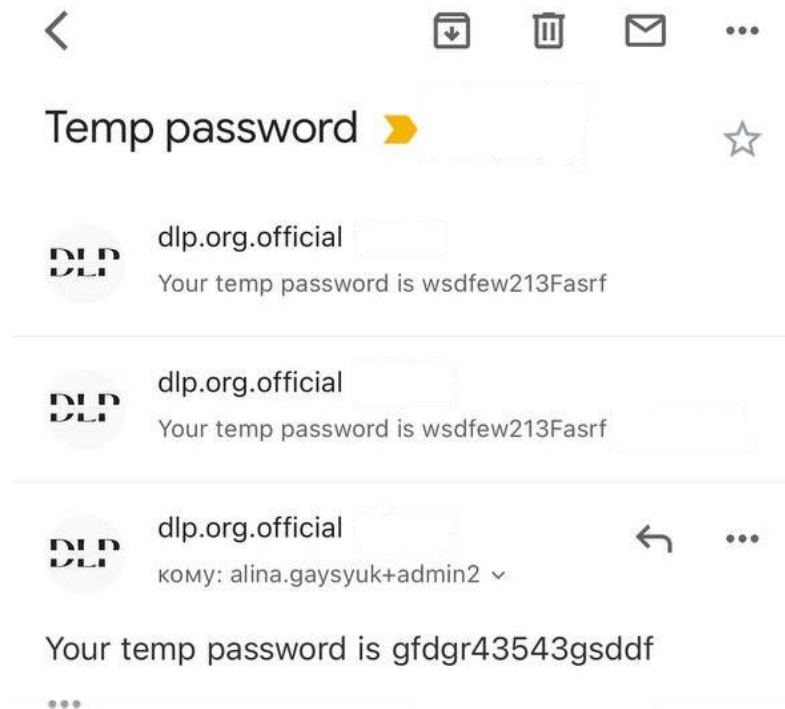


Рисунок 4.6 – Лист з тимчасовим паролем

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
При першій авторизації в системі адміністратору необхідно змінити пароль перш ніж отримати доступ до іншого функціоналу (рис. 4.7).

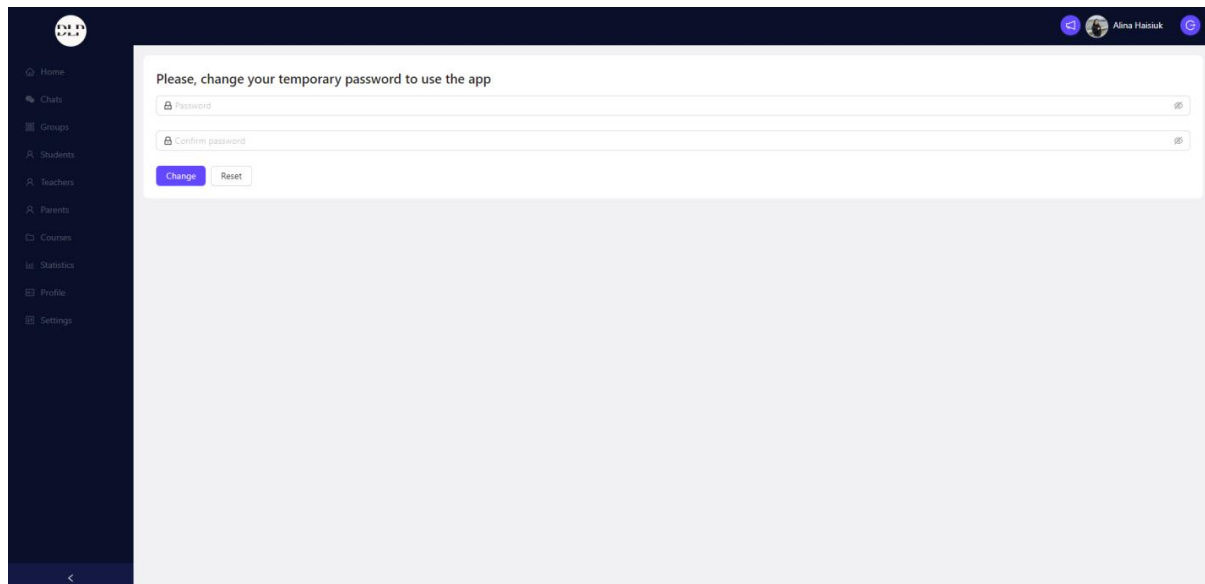


Рисунок 4.7 – Зміна паролю

Після чого користувача буде переадресовано на домашню сторінку (рис. 4.8).

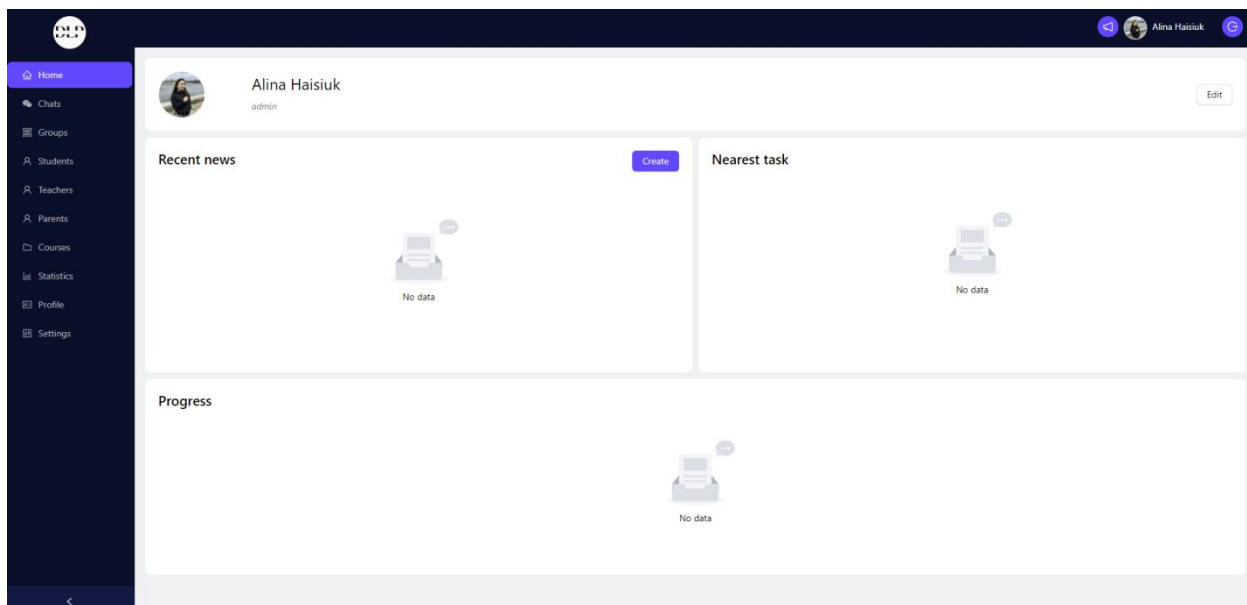


Рисунок 4.8 – Домашня сторінка

Переходячи на сторінку «Groups», адміністратор спершу має створити групи, щоб потім додати до них студентів (рис. 4.9).

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

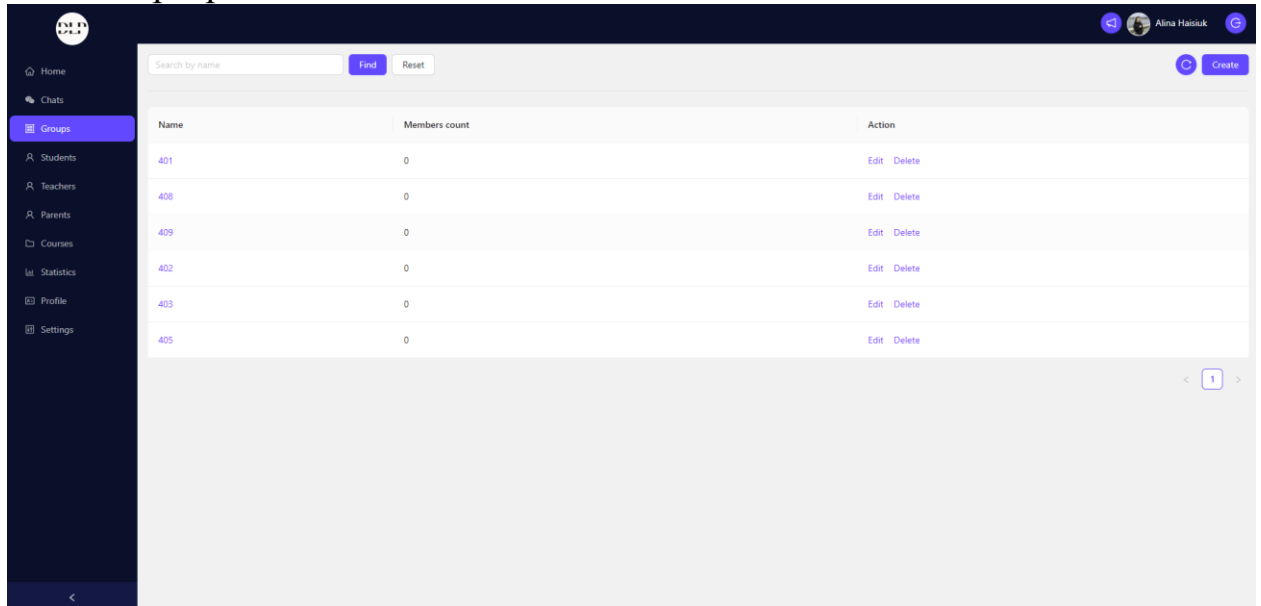


Рисунок 4.9 – Створення групи

Після створення групи, адміністратор може додати до неї студентів на сторінці «Students» (рис. 4.10).

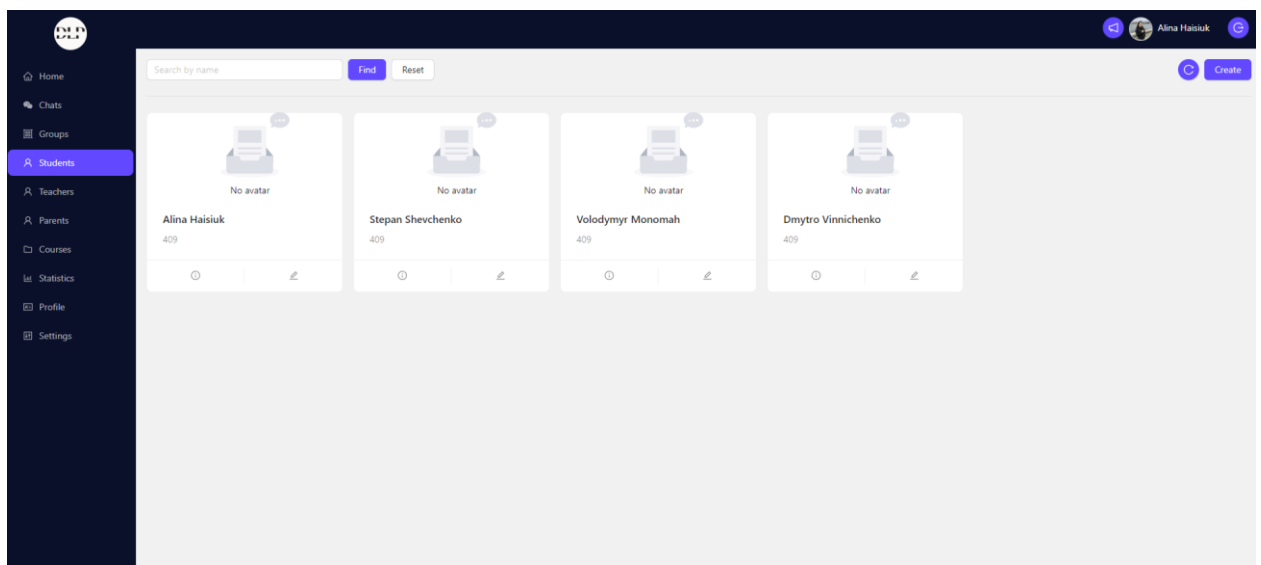


Рисунок 4.10 – Додавання до групи студентів

Також, однією з основних функцій системи є створення сутності викладача. На сторінці «Teachers» адміністратор створює обліковий запис для того, аби викладачі мали змогу створювати курси та за потреби керувати ними (рис. 4.11).

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

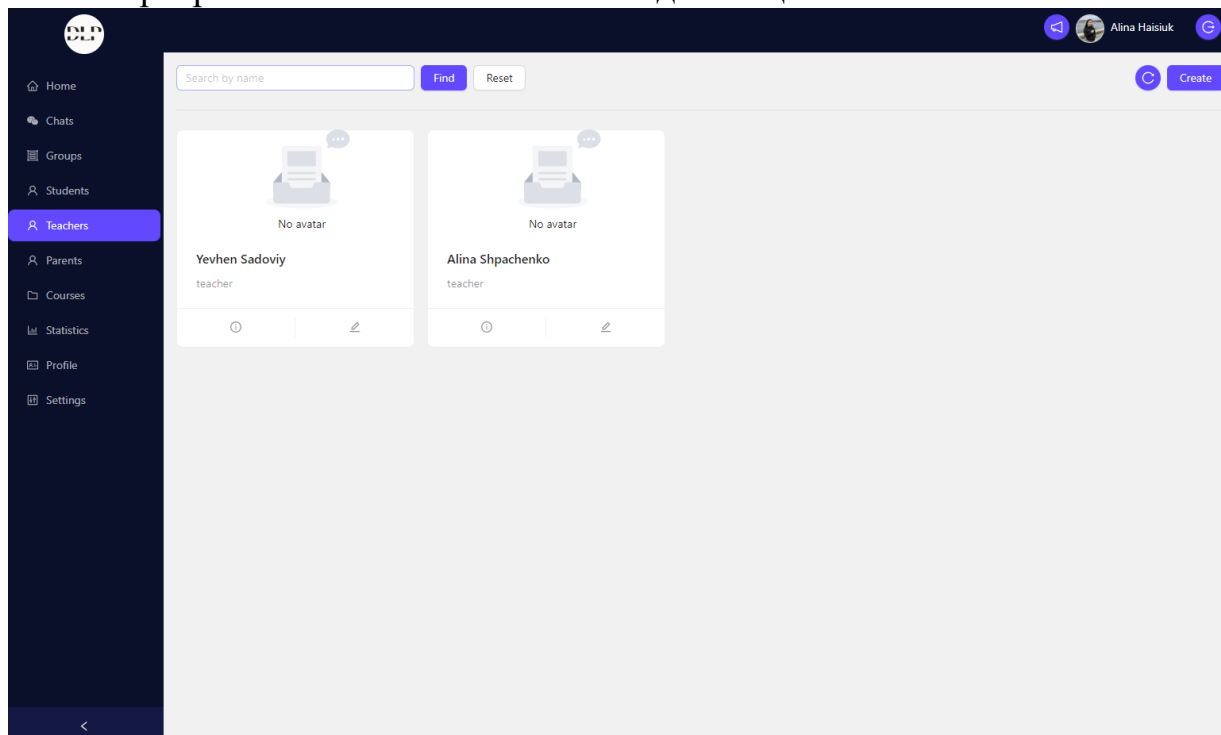


Рисунок 4.11 – Створення сутності викладача

Для підтримки комунікації між користувачами системи існує вбудований онлайн чат (рис. 4.12).

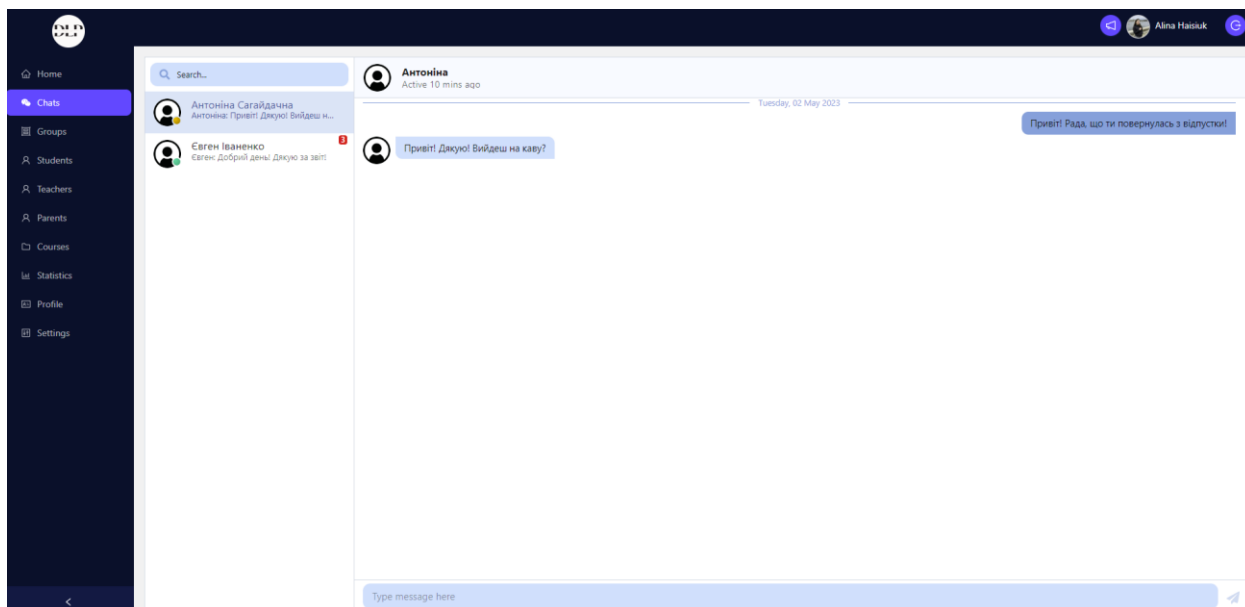


Рисунок 4.12 – Вбудований онлайн чат

Кафедра інженерії програмного забезпечення
 Програмне забезпечення системи дистанційного навчання ЗВО
 Також попри основний функціонал система додатково відображає певну статистику, яка може допомогти в аналізі ситуації в університеті в цілому (рис. 4.13).

Active Users	Students	Teachers	Parents
8	4	2	2
Active Groups	No. of members	Active Chats	No. of messages
8	4	2	421
Active Courses	Finished Courses	Pending Courses	Deleted Courses
24	4	0	2
Active Tasks	Finished Tasks	Pending Tasks	Deleted Tasks
37	46	51	22

Рисунок 4.13 – Статистика

Після створення сутностей користувачів різних ролей можна перевірити базу даних на наявність всіх необхідних даних (рис. 4.14).

id	abc email	abc role	abc lastName	abc firstName	abc password	abc tempPassword
1	1e5e264a-ba98-45f5-882a-429e4fb7b3fe	superadmin	Haisiuk	Alina	\$2b\$10\$5B7WoGAeS6Y	[NULL]
2	fd5a6839-5d26-4a26-ab12-cc3b8fc756a0	admin	Haisiuk	Alina	\$2b\$10\$5B7WoGAeS6Y	[NULL]
3	b60022af-0307-4b11-8061-2a1372761d2c	student	Haisiuk	Alina	[NULL]	tempPassword
4	4adcb50c-41a5-43f4-90a4-efb1f74e62b9	student	Shevchenko	Stepan	[NULL]	tempPassword
5	ed38ef5e-de35-4424-a7b8-08285ec0fcde	student	Monomah	Volodymyr	[NULL]	tempPassword
6	7d144bcd-0264-43fe-be2e-78371502cdcd	student	Vinnichenko	Dmytro	[NULL]	tempPassword
7	5687dc92-0630-4104-aef2-b1a8fd5bd952	teacher	Sadoviy	Yevhen	[NULL]	tempPassword
8	78fa4bde-5cc6-4c89-92ad-b2b4a3c5c81a	teacher	Shpachenko	Alina	[NULL]	tempPassword

Рисунок 4.14 – Перевірка бази даних

Колонка «tempPassword» відповідає за тимчасовий пароль, наданий адміністратором до створених ним користувачів. Наявність значень в даній колонці означає, що користувачі ще не змінили тимчасовий пароль на власний.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
Основою функцією системи є керування процесом навчання, а саме створення курсів, створення завдань, завантаження та перевірка відповідей.

Для того, аби повноцінно користуватись всіма можливостями вебзастосунку, адміністратору або викладачу необхідно створити курси для обраних груп (рис. 4.15).

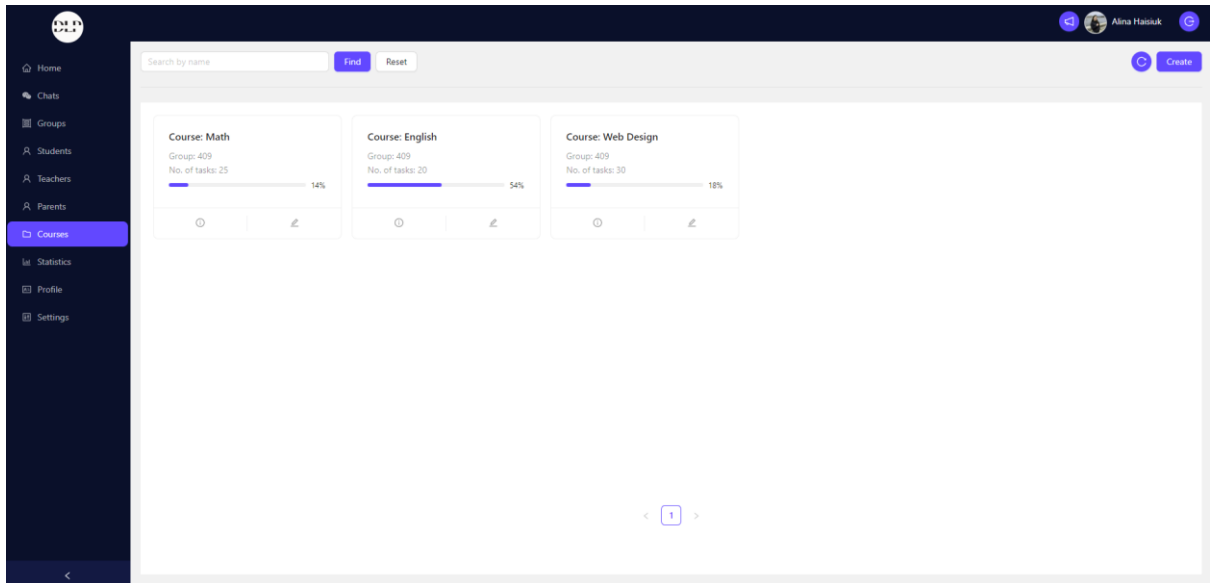


Рисунок 4.15 – Створення курсів для обраних груп

Після чого викладач може конфігурувати власний курс, додавати до нього завдання, завантажувати файли та визначати максимальні бали за завдання (рис. 4.16).

Кафедра інженерії програмного забезпечення Програмне забезпечення системи дистанційного навчання ЗВО

The screenshot shows the 'Web Design' course configuration interface. It features a dark sidebar with navigation options: Home, Chats, Groups, Students, Teachers, Parents, Courses, Statistics, Profile, and Settings. The main content area is titled 'Web Design' and contains a list of 10 items. Each item has a 'Name' field, a 'Max points' field, and a 'File' field with a 'Click to Upload' button. At the bottom of the list, there is an 'Add work' button and a 'Submit' button.

Рисунок 4.16 – Конфігурування курсів

При натисканні на кнопку «Submit», курс оновлюється, та усі групи, які мають дозвіл на даний курс, мають можливість виконувати завдання, завантажувати відповідь, та переглядати результати (рис. 4.17).

The screenshot shows the 'Web Design' course update interface. It features the same dark sidebar as the previous screenshot. The main content area is titled 'Web Design' and contains several fields: 'Work name' (Photoshop (part 1)), 'Max points' (6), 'Due date' (07.05.23 18.00), and 'Uploaded file' (none). Below these fields is a large dashed box for file upload with a 'Click or drag file to this area to upload' message and a 'Save' button.

Рисунок 4.17 – Оновлення курсів

Переглядаючи функціонал системи варто зазначити, що простота інтерфейсу та багатство на функціональність гарантує користувачам вкрай

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
комфортний процес навчання. Попри те, що вебзастосунок має широкий спектр функцій в першій версії, будуть розглядатися усі майбутні фідбеки користувачів, а також оновлюватись в залежності від рекомендацій.

4.2 Тестування системи

Перевірка функціональності системи є процесом аналізу системи або її компонентів з метою виявлення відмінностей між очікуваним (необхідним) та спостережуваним поведінкою. Нижче наведено опис використаних процедур.

1. *Модульне тестування.* Модульне тестування – це метод перевірки, при якому розробник перевіряє придатність окремих компонентів вихідного коду для використання. Системні блоки перевіряються по черзі шляхом введення неприпустимих даних.

2. *Інтеграційне тестування.* Інтеграційне тестування включає об'єднання модулів та їх перевірку. Тестування інтеграції відбувається поетапно. Спочатку тестується найвищий рівень, або координуючий модуль, та лише один із його підпорядкованих модулів. Після модульного тестування система перевіряється на те, чи всі блоки взаємодіють між собою належним чином.

3. *Системне тестування.* Системне тестування програмного забезпечення є перевіркою повної системи з метою оцінки її відповідності визначеним вимогам. Цей процес тестування спрямований на переконання в тому, що вся система працює відповідно до вимог.

Функціональне тестування системи базується на моделі варіантів використання, розроблених на етапі аналізу. Система також проходить операційне тестування відповідно до вимог. Перевіряються такі вимоги, як коректність, точність, безпека та інші.

Перевірка коректності полягає в тому, що програмне забезпечення поводить відповідно до очікуваного сценарію. Забезпечується легкість

Кафедра інженерії програмного забезпечення

Програмне забезпечення системи дистанційного навчання ЗВО взаємодії з програмою завдяки зрозумілому інтерфейсу, а програма реагує на коректні введення користувача згідно очікуваного результату.

Перевірка точності гарантує, що система надає лише правильні результати. Це означає, що в разі відсутності відповідних даних система не повинна генерувати відповідну відповідь. Завдяки виконанню цих характеристик програма забезпечує точність.

Перевірка безпеки забезпечує захист програми. Користувачі повинні увійти до системи, використовуючи логін та пароль. Система дозволяє доступ лише авторизованим користувачам, які мають створені облікові записи з використанням імені користувача та пароля. Це забезпечує безпеку програми.

Таким чином, процес тестування системи включає модульне тестування, інтеграційне тестування та системне тестування, з фокусом на перевірці правильності, точності та безпеки програмного забезпечення.

Висновки до розділу 4

В четвертому розділі кваліфікаційної роботи бакалавра описана програмна реалізація вебзастосунку системи дистанційного навчання ЗВО. Зроблено детальний опис дизайну вебзастосунка, огляд функціоналу відповідно до кожної ролі користувача. Проведено тестування системи, а саме зосереджено увагу на модульному, інтеграційному та системному тестуванні. Перевірено правильність, точність та безпека програмного забезпечення.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра підвищено рівень організації дистанційних занять, а також забезпечено підтримку очного навчання за рахунок створення універсальної системи дистанційного навчання ЗВО.

Для досягнення вказаної мети вирішено поставлені завдання:

1. Розроблено ТЗ на розробку системи дистанційного навчання.
2. Розроблено дизайн та інтерфейс користувацької частини СДН.
3. Розроблено функціональні вимоги до системи, включаючи можливості для студентів, батьків, викладачів та адміністраторів.
4. Спроектовано базу даних для збереження інформації про користувачів, курси, завдання тощо.
5. Спроектовано механізми автентифікації та авторизації користувачів.
6. Спроектовано можливості для завантаження та перегляду навчальних матеріалів, відеолекцій, тестів та інших засобів навчання.
7. Спроектовано систему оцінювання результатів навчання студентів та забезпечено можливість підготовки звітів для викладачів та адміністраторів.
8. Спроектовано чат-систему для взаємодії між користувачами та адміністраторами системи.
9. Протестовано розроблену систему та забезпечено її стабільну та безперебійну роботу.

Розглянуто етапи реалізації проєкту, розроблено UML-діаграми та зроблено огляд задіяних технологій для розробки. Описано дизайн вебзастосунку та проведено тестування задля коректної роботи системи. Результатом проведеної роботи є система дистанційного навчання ЗВО.

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Moodle: Open-source learning platform. URL: <https://moodle.org/?lang=uk> (Last accessed: 24.04.2023).
2. Canvas by Instructure | World's #1 Teaching and Learning Software. Instructure. URL: <https://www.instructure.com/canvas> (Last accessed: 25.04.2023).
3. Docebo Learning Suite: Learn. Develop. Succeed. Docebo. URL: <https://www.docebo.com/> (Last accessed: 27.04.2023).
4. Nagy A. The impact of E-Learning. E-Content: Technologies and Perspectives for the European Market. 2005. P. 79–96.
5. Burac M. A. P., Fernandez J. M., Cruz M. M. A., Cruz J. Dela. Assessing the impact of e-learning system of higher education institution's instructors and students. IOP Conference Series: Materials Science and Engineering. 2019. Vol. 482, № 1. P. 012009.
6. Ellis R. A., Goodyear P. Students' Experiences of E-learning in Higher Education: The Ecology of Sustainable Innovation. Students' Experiences of E-learning in Higher Education: The Ecology of Sustainable Innovation. 2013, P. 1–209.
7. Lee J., Song H. D., Hong A. J. Exploring Factors, and Indicators for Measuring Students' Sustainable Engagement in e-Learning. Sustainability 2019, Vol. 11, № 4. Page 985.
8. Kirsal Ever Y., Dimililer K. The effectiveness of a new classification system in higher education as a new e-learning tool. Quality and Quantity. 2018, Vol. 52, № 1. P. 573–582.
9. Secure and resizable cloud compute – Amazon EC2 – Amazon Web Services. URL: <https://aws.amazon.com/ec2/> (Last accessed: 01.05.2023).

Кафедра інженерії програмного забезпечення
Програмне забезпечення системи дистанційного навчання ЗВО

10. Low-Latency Content Delivery Network (CDN) – Amazon CloudFront – Amazon Web Services. URL: <https://aws.amazon.com/cloudfront/> (Last accessed: 02.05.2023).
11. Fully Managed Relational Database – Amazon RDS – Amazon Web Services. URL: <https://aws.amazon.com/rds/> (Last accessed: 03.05.2023).
12. Cloud Object Storage – Amazon S3 – Amazon Web Services. URL: <https://aws.amazon.com/s3/> (Last accessed: 04.05.2023).
13. Understanding GitHub Actions – GitHub Docs. URL: <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions> (Last accessed: 05.05.2023).
14. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. URL: <https://dou.ua/forums/topic/40575/> (Дата звернення: 23.05.2023).
15. Моралес Дж. What is A UML component diagram. URL: <https://www.mindonmap.com/uk/blog/uml-component-diagram/> (Last accessed: 25.05.2023).
16. Елементи UML. Діаграма випадків використання. URL: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-elements.html> (Дата звернення: 24.05.2023).
17. Репозитарій Національного Авіаційного Університету: Номе. URL: https://er.nau.edu.ua/bitstream/NAU/44152/2/Методичні%20вказівки%20модельовання%20в%20ЦЕ_Бойко.pdf (дата звернення: 25.05.2023).
18. ЕФЕКТИВНІСТЬ ЗАСТОСУВАННЯ БІБЛІОТЕКИ REACT. Інформаційні технології та суспільство. URL: <http://journals.maup.com.ua/index.php/it/article/view/2096> (Дата звернення: 24.05.2023).
19. Рейтинг популярних в Україні мов програмування 2022 року – в лідерах JavaScript, Java й Python. URL: <https://itc.ua/ua/novini/rejtyng-populyarnyh-v-ukrayini-mov-programuvannya-2022-roku-v-liderah-javascript-java-j-python/> (Дата звернення: 25.05.2023).