

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО РЕСУРСУ
ДЛЯ КОМУНІКАЦІЇ ЗДОБУВАЧІВ АКАДЕМІЧНОЇ ГРУПИ
SOFTWARE IMPLEMENTATION OF AN INFORMATION RESOURCE
FOR STUDENT COMMUNICATION IN AN ACADEMIC GROUP

Виконав: студент(ка) 4 курсу, групи КНТ-218

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

Павленко В.С.

(прізвище та ініціали)

Керівник Субботін С.О.

(прізвище та ініціали)

Рецензент Гофман Є.О.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ

Кафедра програмних засобів

Ступінь вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки

(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.

С.О. Субботін

“ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Павленка Віталія Сергійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація інформаційного ресурсу для комунікації здобувачів академічної групи. Software Implementation of an Information Resource for Student Communication in an Academic Group

керівник проєкту (роботи) Субботін Сергій Олександрович, д.т.н., професор,

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

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 07 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Субботін С.О., професор		
Нормоконтроль	Липовець М.В., асистент		

7. Дата видачі завдання “ 04 ” квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Павленко В.С.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Субботін С.О.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 96 с., 38 рис., 5 табл., 4 дод., 22 джерел.

ІНФОРМАЦІЙНИЙ РЕСУРС, АКАДЕМІЧНА СПІЛЬНОТА, СТУДЕНТСТВО, КОМУНІКАЦІЯ, ПРОГРАМНА РЕАЛІЗАЦІЯ, РОЗРОБКА, ВЕБЗАСТОСУНОК.

Об'єкт дослідження – процес програмної реалізації інформаційного ресурсу для комунікації здобувачів академічної групи

Предмет дослідження – методи програмної реалізації інформаційного ресурсу для комунікації здобувачів академічної групи.

Метою роботи є розроблення інформаційного ресурсу для комунікації здобувачів академічної групи.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування PHP, система контролю контентом Joomla, система керування базою даних MySQL.

Результати. Розроблено програмне забезпечення у вигляді інформаційного ресурсу для комунікації здобувачів академічної групи.

Наукова новизна роботи полягає в розробці кроссплатформного інформаційного ресурсу для комунікації здобувачів академічної групи у вигляді вебзастосунку.

Висновки. Розроблено інформаційний ресурс для комунікації здобувачів академічної групи.

Галузь використання – академічна спільнота та студентство.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 96 pages, 38 figures, 5 tables, 4 appendixes, 22 sources.

INFORMATION RESOURCE, ACADEMIC COMMUNITY, STUDENTS, COMMUNICATION, SOFTWARE IMPLEMENTATION, DEVELOPMENT, WEB APPLICATION.

The object of research is the process of software implementation of an information resource for communication of academic group applicants

The subject of the research is methods of software implementation of an information resource for communication of academic group applicants.

The aim of the work is to develop an information resource for communication between applicants of the academic group.

Materials, methods, and technical tools: Object-Oriented Programming, PHP programming language, Joomla content management system, MySQL database management system.

Results. Software has been developed in the form of an information resource for communication of academic group applicants.

The scientific novelty of the work consists in the development of a cross-platform information resource for communication of academic group applicants in the form of a web application.

Conclusions. An information resource has been developed for communication of academic group applicants.

Field of usage: is the academic community and students.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Академічна група, як соціальне явище	10
1.2 Соціалізація та спілкування всередині академічної групи	13
1.3 Огляд існуючих рішень для соціальної взаємодії всередині академічної групи.....	16
1.3.1 Використання месенджера Telegram	16
1.3.2 Використання спільнот у соціальній мережі Facebook.....	18
1.3.3 Використання хмарних сервісів та платформ колективної роботи Google Workspace	19
1.3.4 Порівняння існуючих аналогів	21
1.4 Постановка завдання.....	22
1.4.1 Визначення стейкхолдерів проєкту.....	23
1.4.2 Функціональність системи.....	23
1.4.3 Вимоги до інтерфейсу.....	25
1.4.4 Вимоги до продуктивності.....	25
1.4.5 Вимоги до безпеки	25
1.4.6 Експлуатаційні вимоги	25
1.4.7 Не функціональні вимоги до продукту.....	26
1.5 Висновки за розділом 1.....	26
2 Розробка архітектури програми	27
2.1 Вибір інструментарію розробки	27
2.1.1 Вибір мови програмування	27
2.1.2 Система керування вмістом	30
2.1.3 Розподіл та реалізація бізнес-логіки	32
2.1.4 Обґрунтування вибору CMS для розробки вебресурсу	33
2.2 Структурна схема розробки	34

	7
2.3 Висновки за розділом 2.....	35
3 Основні рішення щодо реалізації компонентів системи	36
3.1 Проектування моделей системи.....	36
3.2 Проектування інтерфейсу.....	41
3.3 Висновки за розділом 3.....	44
4 Експлуатація, тестування та експериментальне дослідження програми.....	45
4.1 Опис застосунку	45
4.2 Умови виконання програми	45
4.3 Інструкція по експлуатації програми	46
4.3.1 Характеристики програми.....	46
4.3.2 Використані технічні засоби.....	48
4.3.3 Звернення до програми.....	49
4.3.4 Вхідні й вихідні дані	49
4.3.5 Розширення функціональності вебзастосунку за допомогою модулів	49
4.4 Методика та результати тестування.....	54
4.4.1 Тестування функціоналу	54
4.4.2 Тестування за допомогою WAPT	57
Висновки	59
Перелік джерел посилання	60
Додаток А Технічне завдання	63
Додаток Б Опис програми	70
Додаток В Текст програми	75
Додаток Д Слайди презентації.....	90

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CMS – системи управління контентом;

MVC – model-view-control;

БД – база даних;

ЗВО – заклад вищої освіти;

ОС – операційна система;

ПЗ – програмний засіб;

ПК – персональний комп'ютер.

ВСТУП

Інтернет став найбільш ефективним засобом реклами і просування і є одним з важливих елементів сучасної цивілізації. Інтернет може задовольнити всі потреби сучасної людини: покупки, укладення ділових відносин і пошук клієнтів. І в той же час багато користувачів заходять в інтернет читати книги, розважатися, спілкуватися з друзями, дізнаватися щось нове для себе. Тому створення сайтів стало таким популярним явищем.

У всесвітній мережі Інтернет своє місце зайняло зберігання інформації та взаємодія між різними групами людей. Школярі, студенти, робочі колективи користуються соціальними мережами та іншими ресурсами для обміну інформацією. Саме тому практично у кожній організації існує свій офіційний сайт. В даний час існує невелика кількість студентських сайтів. Створення ж такого сайту дозволить реорганізувати роботу навчальної групи.

Завдання, поставлені авторами:

- вивчити функціональні можливості PHP, Joomla;
- провести аналіз предметної області;
- розробити графічний макет;
- розробити UML-модель;
- вивчити можливості верстки і програмування макета;
- наповнити сайт інформацією: розклад, контакти старости, інформація про викладачів, та ін.

При створенні проекту використовувалися метод теоретичного дослідження-аналіз і емпіричний метод – вивчення джерел літератури.

Сайт групи університету розроблений відповідно до позначених завдань. Він призначений для розміщення поточної інформації про групу університету, викладачів, розклад, новини та навчальні матеріали.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Перший розділ роботи присвячено висвітленню питань предметної області – вебресурсу для студентської (академічної) групи. Для цього на цьому етапі буде проаналізовано питання комунікації, в тому числі електронної комунікації, між студентам в групі. Буде досліджено існуючі рішення та зазначено їх переваги і недоліки. Після цього буде можливо розробити перелік функціональних та не функціональних вимог до застосунку, що буде надалі розроблятися.

1.1 Академічна група, як соціальне явище

При вступі молодшої людини до закладу вищої освіти (ЗВО), вона стає членом певної академічної групи. У неї входять люди, які є одногрупниками. Вони разом відвідують заняття, справляються з різними завданнями, а також беруть участь в різних соціальних заходах [1]-[4].

Академічна група – це соціальне об'єднання людей, яке володіє унікальними ознаками, і проходить свої етапи створення [1]-[4].

По-іншому вона називається студентською групою. Це об'єднання студентів, яке має свої закономірності і нюанси. Академічна група створюється автоматично і обумовлена необхідністю ведення навчальної діяльності після вступу до ЗВО [1]-[4]. При утворенні таких груп слід враховувати:

- всі студенти є членами новоствореного колективу, в якому проходять навчання протягом, як мінімум 4 років [1]-[4];
- між учнями і викладачами формуються стійкі відносини для вирішення організаційних і навчальних питань [1]-[4];
- щодня студенти спілкуються один з одним, обмінюються відомостями, формують нові інтереси, а також задовольняють духовні потреби [1]-[4];

– викладачі і керівництво ЗВО легко може простежити зростання всіх студентів в різних областях [1]-[4].

Становлення будь-якого колективу є досить складною процедурою, причому студенти стикаються не тільки з підйомами, а й зупинками, протиріччями, конфліктами і різними проблемами [1]-[4].

Колектив, в якому студент навчається, представлений близьким середовищем. Тут кожен реалізує професійні вміння, бере участь у навчальній діяльності, вчиться спілкуванню і виробляє різні установки, що мають відношення до обраної спеціальності [1]-[4]. Студентська група представлена об'єднанням учнів, які бажають досягти єдиної мети. Вони мають схожі інтереси і орієнтуваннями, спрямованими на спільне навчання, регулярне спілкування і самоврядування. Функціонують такі групи завдяки самоврядуванню, тому в кожному колективі встановлюються як формальні, так і неформальні лідери. Додатково, управління здійснюється за допомогою викладача, призначеного куратором [1]-[4]. Студентська група націлена на отримання освіти, але такий фактор є досить слабким, тому рідко можна спостерігати згуртовані колективи. До головних ознак хорошого колективу відноситься [1]-[4]:

- наявність єдиної і істотної мети;
- спільність інтересів;
- володіння духовною єдністю;
- постійне спілкування і заняття спільним вирішенням різних завдань;
- виконання певних вимог і зобов'язань один перед одним, а також залежність від інших студентів і викладачів;
- утворення органів самоврядування.

До особливостей студентського колективу відноситься:

- студенти бажають отримати потрібні навички, щоб в майбутньому справлятися з професійною діяльністю, тому спільні інтереси складаються за рахунок отримання освіти [1]-[4];

– люди бажають отримувати прекрасні результати не тільки від навчання, а й від майбутньої роботи; в колективі збираються люди, що володіють однаковою освітою і віком, але у них є різні мотиви для вступу в конкретний ЗВО [1]-[4];

– об'єднання людей характеризується послідовністю, тому навчання здійснюється шляхом переходу з одного курсу на інший [1]-[4];

– управління виконується безпосередніми студентами, для чого може виявлятися один або кілька лідерів [1]-[4].

Така студентська група існує протягом періоду, необхідного для отримання освіти. Як тільки мета досягається-люди розходяться, тому припиняють спілкування і постійні зустрічі. У них з'являються нові цілі і мотиви, а також вони влаштовуються на роботу в різні компанії, вступаючи у вже існуючі робочі колективи [1]-[4].

Процедура формування такого колективу ділиться на 4 етапи, які реалізуються послідовно [1]-[4].

Створення колективу. Цей етап припадає на перший курс, коли студенти тільки починають знайомитися один з одним і вивчати правила навчання в обраному закладі. Люди згуртовуються, щоб спростити процедуру адаптації [1]-[4]. Вони ставлять перед собою значущі цілі, визначаються зі спільною роботою, створюють план навчання, а також вступають в особисті відносини. Зазвичай цей етап триває протягом одного року, до кінця якого студенти повністю звикають до нових умов [1]-[4].

Зміцнення колективу. Процедура реалізується протягом двох років. Студенти вже є повноправними членами ЗВО, поважають один одного, ставлять особисті цілі, а також вважають за краще спільно проводити навіть вільний час. У всіх членів колективу з'являються норми, яким вони слідують. Вони володіють єдиною думкою, цінностями і традиціями. Все чіткіше проявляється самоврядування групою [1]-[4].

Удосконалення колективу. Цей етап припадає на 3 і 4 курси. У цей час колектив виступає організатором різних подій. Всі студенти

підпорядковуюються єдиній громадській думці [1]-[4]. У них є різні цінності, а також вони поважають позицію інших студентів, хоча можуть не розділяти її. Люди впевнені в колективі і постійному розвитку. Викладачі ставляться до них з повагою, а також користуються їх допомогою в процесі вирішення різних важливих питань і проблем [1]-[4].

Припинення існування. Цей етап припадає на останній курс навчання. У випускників є потрібні професійні навички та інтереси, щоб влаштуватися на роботу [1]-[4]. У них з'являються власні сім'ї, а також нові турботи, які не мають відношення до навчання. Тому група перестає бути цінною. Після того, як студенти отримують дипломи, колектив припиняє існування. Але при цьому люди ще довго користуються різними цінностями, законами і традиціями, які формувалися у ЗВО [1]-[4].

Етапи є умовними, тому вони можуть значно відрізнятися в різних групах.

1.2 Соціалізація та спілкування всередині академічної групи

Традиційно навчання у ЗВО супроводжується спілкуванням в студентській групі, яке надає неоціненну допомогу в самоактуалізації особистості. Зрозуміти себе, оцінити свої можливості, усвідомити свої потреби можна тільки в контакті з оточуючими [5]-[7].

Студентська група характеризується спільною діяльністю, наявністю групових норм і групових інтересів, нечисленним складом, безпосереднім особистим спілкуванням членів групи. Успішна соціалізація особистості пов'язана з включенням особистості в систему продуктивних соціальних контактів [5]-[7]. Десоціалізація проявляється в відчуженості особистості від колективу (групи), відході людини в особисті проблеми, псевдоінтимності, ослабленні інтересу до суспільного життя, втрати соціальної активності [5]-[7].

При роботі зі студентською групою слід врахувати ряд моментів, що ускладнюють процес соціалізації особистості.

Існування студентської групи обмежене фіксованим часом, тому групі немає необхідності шукати шляхи і способи свого збереження (немає такої практики – відраховувати з ЗВО групу), що дозволяє період студентського життя витратити не на виживання (збереження групи як функціональної одиниці), а на продуктивний особистісний розвиток [5]-[7].

Ефективність існування групи в навчальній діяльності складається з персональної успішності. Навчальний результат жорстко індивідуалізований, тому навчальна діяльність має обмежені можливості в плані групового згуртування [5]-[7]. У сфері навчальної діяльності групове згуртування може приймати різні форми взаємодопомоги. Рівень взаємодопомоги (взаємні консультації, обмін матеріалом, списування, підказки) залежить від групової культури [5]-[7].

У студентській групі в силу відсутності жорсткої взаємозалежності членів (відсутність кого-небудь на лекції не відбивається на інших) складається терпима культура. Але вона несе в собі і негативний момент байдужості до успішності або неуспішності інших членів групи [5]-[7].

Основна проблема сьогодні-низький рівень соціальної активності студентської групи. В силу цього студентські групи сучасних ЗВО далеко не завжди виходять на рівень колективу, тобто групи, де міжособистісні відносини опосередковуються суспільно цінним і особистісно значущим змістом спільної діяльності. Традиції включення кожного в ту чи іншу форму суспільно-значущої діяльності залишилися в радянському минулому, що значно ускладнює соціалізацію особистості [5]-[7].

Сучасна культура соціальних відносин вимагає від особистості компетенцій, пов'язаних зі здатністю працювати в команді. Для того щоб такий досвід особистість отримала в студентській групі, необхідно включати її в різні види групової роботи, де відповідальність за результат несе кожен член групи [5]-[7].

Суспільно корисна діяльність як фактор соціалізації. Громадська діяльність, будучи потужним фактором соціалізації, виховує громадську активність, формує лідерські якості, дозволяє оволодіти необхідними формами соціальної діяльності. У своїй більшості студенти приймають ідеї демократичного суспільства, але суспільно-політична активність молоді достатня низька [5]-[7]. Загалом діяльність громадських об'єднань студентів наразі характеризується відсутністю згуртованості у досягненні поставлених цілей, аморфністю організаційних структур, слабкою розвиненістю їх у регіонах, незначним охопленням студентської молоді, відсутністю єдиного інформаційного простору, нерозробленістю нормативно-правової бази їх діяльності [5]-[7].

Основною формою суспільно корисної діяльності студентів є Студентське самоврядування (студентські ради, Профспілкові організації студентів, громадські об'єднання). Громадська діяльність представлена різними акціями: Організація свят бойової та трудової слави, волонтерська діяльність, участь в експедиції пошукового загону та ін. [5]-[7].

Розширюється практика залучення студентів до волонтерської діяльності, відмітна особливість якої полягає в тому, що вона здатна проникати в ті сфери життєдіяльності суспільства, які слабо зачіпаються існуючими державними структурами. Волонтерська діяльність студентів сприяє отриманню додаткових соціально-особистісних компетенцій майбутнього молодого фахівця [5]-[7].

Успішна соціалізація проявляється в оволодінні різноманітною практикою соціально значущої, в тому числі і соціально-політичної, діяльності [5]-[7].

1.3 Огляд існуючих рішень для соціальної взаємодії всередині академічної групи

Як було зазначено попередньо комунікація студентів в академічній групі є важливим процесом та етапом її утворення, підтримки та існування. Також комунікація має ще й інформаційну частину, адже такий програмний засіб (ПЗ), що дозволить комунікувати та пересилати мультимедійну інформацію значно полегшить менеджмент та самоорганізацію групи.

На сьогоднішній є декілька способів налагодження комунікації в соціальних групах, в тому числі в академічній спільноті. Така комунікація може бути налагоджена через спільні чати або інформаційні канали в месенджерах (наприклад, Telegram або Viber), спільноти в соціальних мережах (наприклад, у Facebook) або з використанням засобів колективної роботи (Google Workspace). Розглянемо існуючі аналоги.

1.3.1 Використання месенджера Telegram

Telegram – це безкоштовний кроссплатформний хмарний сервіс миттєвих повідомлень [8], [9]. Сервіс також надає наскрізні зашифровані відеодзвінки, загальний доступ до файлів і ряд інших функцій. Він був запущений для iOS 14 серпня 2013 року та Android у жовтні 2013 року. Сервери Telegram розподілені по всьому світу, щоб зменшити часте навантаження на дані, з п'ятьма центрами обробки даних в різних регіонах, в той час як операційний центр базується в Дубаї в Об'єднаних Арабських Еміратах [8], [9]. Різні клієнтські програми доступні для настільних і мобільних платформ, включаючи офіційні програми для Android, iOS, Windows, macOS і Linux (хоча для реєстрації потрібен пристрій iOS або Android і робочий номер телефону). Існують також два офіційних зстосункі-близнюки Telegram web, WebK і WebZ, і численні неофіційні клієнти, що використовують протокол Telegram. Всі офіційні компоненти Telegram

мають відкритий вихідний код, за винятком сервера, який є закритим і пропріє тарним [8], [9].

Telegram надає наскрізні зашифровані голосові та відеодзвінки, а також додаткові наскрізні зашифровані секретні чати [8], [9]. Хмарні чати і групи зашифровані між додатком і сервером, так що інтернет-провайдери та інші сторонні користувачі в мережі не можуть отримати доступ до даних, але сервер Telegram може. Користувачі можуть надсилати текстові та голосові повідомлення, здійснювати голосові та відеодзвінки, а також обмінюватися необмеженою кількістю зображень, документів (2 ГБ на файл), місцеположень користувачів, анімованих стікерів, контактів та аудіо файлів [8], [9]. У січні 2021 року кількість активних користувачів Telegram перевищила 500 мільйонів на місяць, це був самий завантажуваний застосунок у світі в січні 2021 року з 1 мільярдом завантажень по всьому світу станом на кінець серпня 2021 року [8], [9].

Приклад використання Telegram для ведення інформаційного каналу наведено на рисунку 1.1.

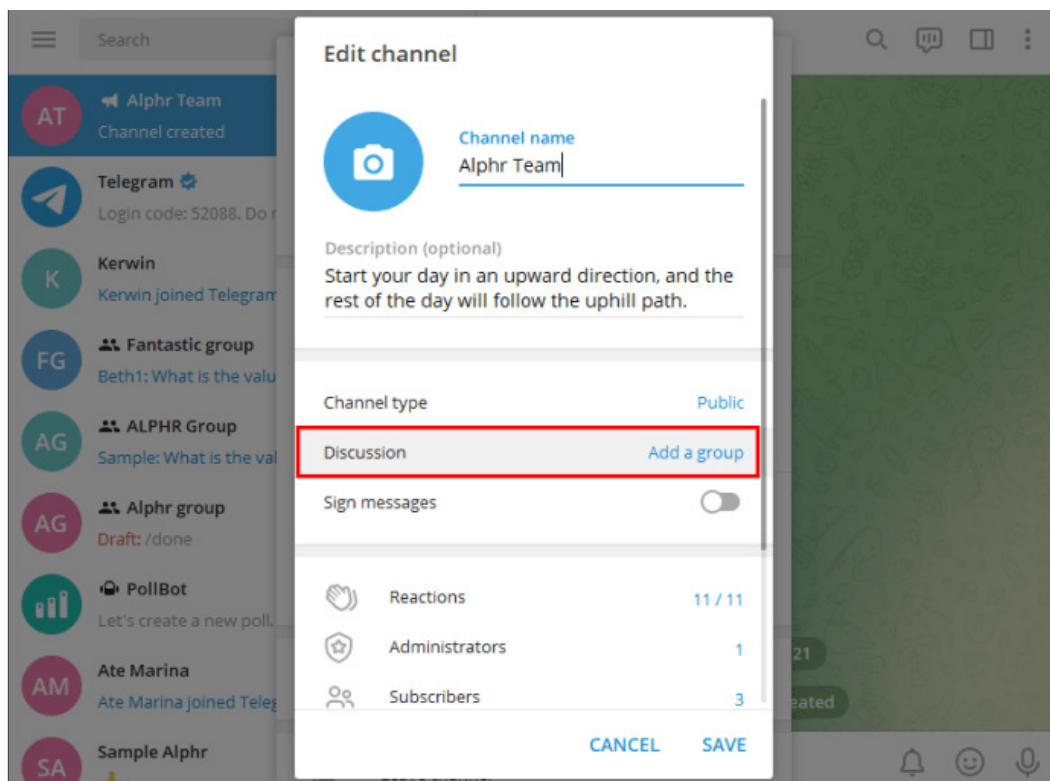


Рисунок 1.1 – Приклад використання Telegram для ведення інформаційного каналу [8]

1.3.2 Використання спільнот у соціальній мережі Facebook

Facebook – це американська онлайн-соціальна мережа та сервіс соціальних мереж, що належить Meta Platforms [10], [11]. Заснована в 2004 році Марком Цукербергом разом з іншими студентами Гарвардського коледжу і сусідами по кімнаті Едуардо Саверіном, Ендрю Макколлумом, Дастіном Московіцем і Крісом Хьюзом, вона отримала назву від каталогів face book, які часто роздають студентам американських університетів [10], [11]. Спочатку членство було обмежено студентами Гарварду, поступово воно поширилося на інші північноамериканські університети, а з 2006 року – на всіх старше 13 років. Станом на 2020 рік Facebook налічував 2,8 мільярда активних користувачів на місяць і займав четверте місце по використанню інтернету в світі. Це був самий завантажуваний мобільний застосунок 2010-х років [10], [11].

Доступ до Facebook можна отримати з пристроїв, підключених до Інтернету, таких як персональні комп'ютери, планшети і смартфони [10], [11]. Після реєстрації користувачі можуть створити профіль, що розкриває інформацію про себе. Вони можуть публікувати текст, фотографії та мультимедіа, якими діляться з будь-якими іншими користувачами, які погодилися бути їх друзями, або, з іншими настройками конфіденційності, публічно [10], [11]. Користувачі також можуть безпосередньо спілкуватися один з одним за допомогою Facebook Messenger, вступати в групи за спільними інтересами і отримувати повідомлення про дії своїх друзів на Facebook і сторінках, на які вони підписані [10], [11].

Будучи предметом численних суперечок, Facebook часто піддавався критиці з таких питань, як конфіденційність користувачів (як у випадку скандалу з даними Cambridge Analytica), політичні маніпуляції (як у випадку виборів в США в 2016 році), масова стеження, психологічні ефекти, такі як залежність і низька самооцінка, а також контент, такий як підроблені новини, теорії змови, порушення авторських прав і розпалювання ненависті [10], [11].

Коментатори звинуватили Facebook в тому, що він добровільно сприяє поширенню такого контенту, а також перебільшує кількість своїх користувачів, щоб привернути увагу рекламодавців [10], [11].

Приклад використання Facebook для ведення спільноти наведено на рисунку 1.2.

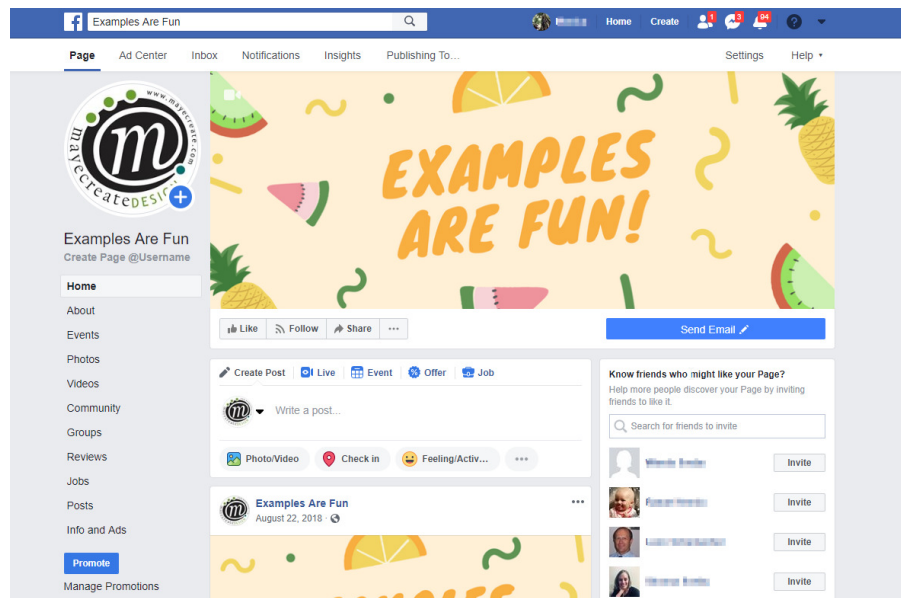


Рисунок 1.2 – Приклад використання Facebook для ведення спільноти [10]

1.3.3 Використання хмарних сервісів та платформ колективної роботи Google Workspace

Google Workspace являє собою набір інструментів для хмарних обчислень, підвищення продуктивності і спільної роботи, ПЗ і продуктів, розроблених і продаваних Google [12], [13]. Вперше він був запусканий в 2006 році як Google Apps для користувацького домену і перейменований в G Suite в 2016 році. Google Workspace складається з Gmail, контактів, календаря, зустрічей і чатів для спілкування; Currents для взаємодії з співробітниками; Drive для зберігання; і пакет документів Google для створення контенту. Для управління користувачами і службами передбачена панель адміністратора. Залежно від версії Google Workspace може також включати цифрову інтерактивну дошку Jamboard і можливість придбання

таких доповнень, як служба телефонного зв'язку Voice. Освітнє видання додає навчальну платформу Google Classroom і сьогодні має назву Workspace for Education [12], [13].

Хоча більшість з цих послуг індивідуально доступні безкоштовно для споживачів, які використовують свої безкоштовні облікові записи Google (Gmail), Google Workspace додає корпоративні функції, такі як настроювані адреси електронної пошти в домені, можливість необмеженого зберігання на диску, додаткові інструменти адміністрування та розширені налаштування, а також 24/7 підтримка по телефону та електронній пошті [12], [13].

Оскільки дані та інформація базуються в центрах обробки даних Google, вони зберігаються безпосередньо, а потім синхронізуються з іншими центрами обробки даних для цілей резервного копіювання [12], [13]. На відміну від безкоштовних, орієнтованих на споживача сервісів, користувачі Google Workspace не бачать реклами під час використання сервісів, а інформація і дані в облікових записах Google Workspace не використовуються в рекламних цілях. Крім того, адміністратори Google Workspace можуть точно налаштувати параметри безпеки та конфіденційності [12], [13].

Станом на квітень 2020 року у G Suite було 6 мільйонів оплачених компаній і 120 мільйонів користувачів G Suite for Education [12], [13].

Приклад використання Google Workspace наведено на рисунку 1.3.

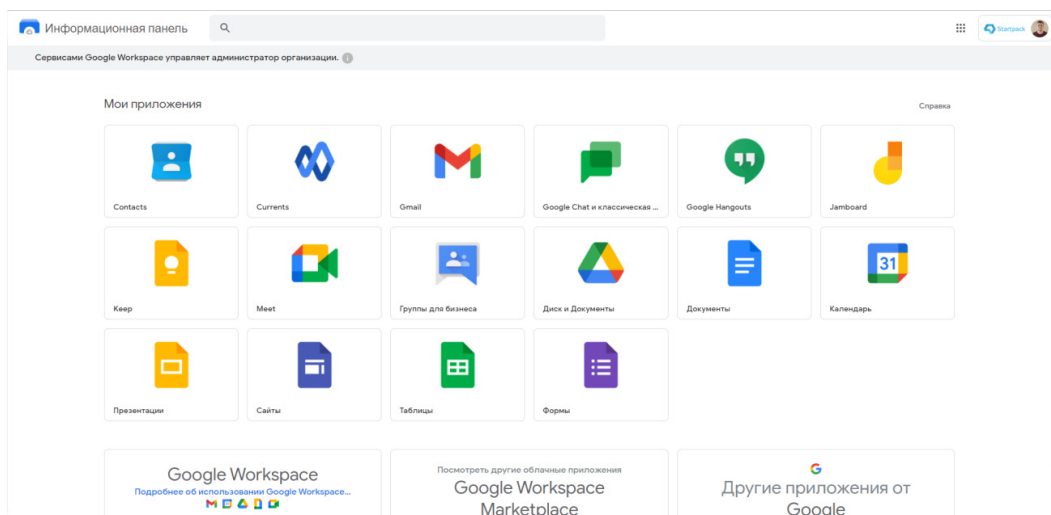


Рисунок 1.3 – Робоча сторінка Google Workspace

1.3.4 Порівняння існуючих аналогів

Проведемо порівняння за певними критеріями, хід якого представимо у вигляді таблиці (табл. 1.1). Серед критеріїв виділимо наступні:

- доступність – чи є ресурс безкоштовним або вимагає додаткової оплати;
- організація чатів/коментарів – чи надає ресурс можливість коментувати новини або просто спілкуватися між користувачами;
- збереження матеріалів – чи надає ресурс можливість зберігати та ділитися матеріали для навчання;
- рівень персоналізації – можливість забезпечити персоналізацію ресурсу під потреби адміністратора.

З результатів порівняння можна зробити висновок, що в більшості випадків існуючі аналоги є безкоштовними та вільно розповсюджуються. Проте, слід зазначити, що наприклад безкоштовна версія засобів Google Workspace має істотні обмеження (місце на Диску та інше). Більш того, без введення колективної реєстрації в корпоративному просторі Google Workspace є проблема із використанням окремих сервісів платформи. В той же час для зберігання інформації Telegram та Facebook не є найбільш вдалим вибором, адже ці платформи не надають найзручнішої організації матеріалів, більш того у Facebook доведеться використовувати зовнішні платформи для розміщення деяких типів файлів. Проте, найбільш цікавим є можливість адаптації та персоналізації платформ під потреби адміністратора. У Facebook це зробити майже не можливо, адже набір функціоналу є стандартним для усіх спільнот, у Google Workspace це обмежено межами безкоштовної версії, а от у Telegram особливих обмежень не має, але в деяких випадках слід використовувати зовнішніх ботів для розширення функціоналу.

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій порівняння	Чат/Канал Telegram	Спільнота Facebook	Засоби Google Workspace	Інформаційний вебресурс
Доступність	Безкоштовно	Безкоштовно	Безкоштовно (але додатковий функціонал – платний)	Безкоштовно
Організація чатів/коментарів	Так	Так	Так (додаткові засоби)	Так
Збереження матеріалів	Так	Через посередні сервіси	Так	Так
Рівень персоналізації	Середній	Низький	Низький	Високий

Саме тому, можна зробити висновок, що не дивлячись на велику кількість аналогів та їх переваги все ж таки проблема розроблення централізованого інформаційного ресурсу для академічної групи є достатньо важливим та актуальним. Адже такий ресурс буде надавати необхідний функціонал із задовільним рівнем персоналізації під потреби адміністратора.

1.4 Постановка завдання

Проаналізувавши усі найбільш поширені і можливі варіанти реалізації проєкту було прийнято рішення про організацію незалежного сайту на окремому хостингу з метою тонкого налаштування його вмісту і доступу до нього. На даний момент не представляється можливим здійснювати оповіщення по телефону, відповідно, потрібна альтернативна заміна, що в умовах сучасного поширення інтернет-технологій формулюється в необхідності створення сайту [14], [15].

1.4.1 Визначення стейкхолдерів проєкту

Зацікавлені сторони (стейкхолдери) проєкту [14], [15] представлені в табл. 1.2.

Таблиця 1.2 – Зацікавлені сторони проєкту

Сторона	Представлена зацікавленість	Роль
Користувач	Представляє користувача системи	Особа, яка використовує програму
Власник	Представляє замовника системи	Особа, яка замовила та оплатила систему
Розробник	Представляє розробників	Особа, що створює систему
Адміністратор	Представляє адміністратора системи	Особа, яка використовує систему в розширеному режимі.
Аналітик	Представляє аналітика	Особа, яка здійснює збір та аналіз вимог, розробка документації.
Тестувальник	Представляє фахівця з Тестування ПЗ	Особа, яка здійснює тестування системи.
Модератор	Представляє користувача, який здійснює контроль.	Особа, яка здійснює контроль.

Загальна схема розподілу функціоналу [14], [15] в проєкті на рис. 1.4.

1.4.2 Функціональність системи

Тепер же визначимо функціональні вимоги відповідно до стейкхолдерів [14], [15].

Користувальницькі функції:

- додавання контенту;
- пошук по існуючому контенту;
- редагування та видалення вмісту;

- коментарі до контенту;
- завантаження файлів (медіа, документи);
- видалення файлів;
- інтерактивне голосування;
- отримання повідомлень про зміни або новинах.

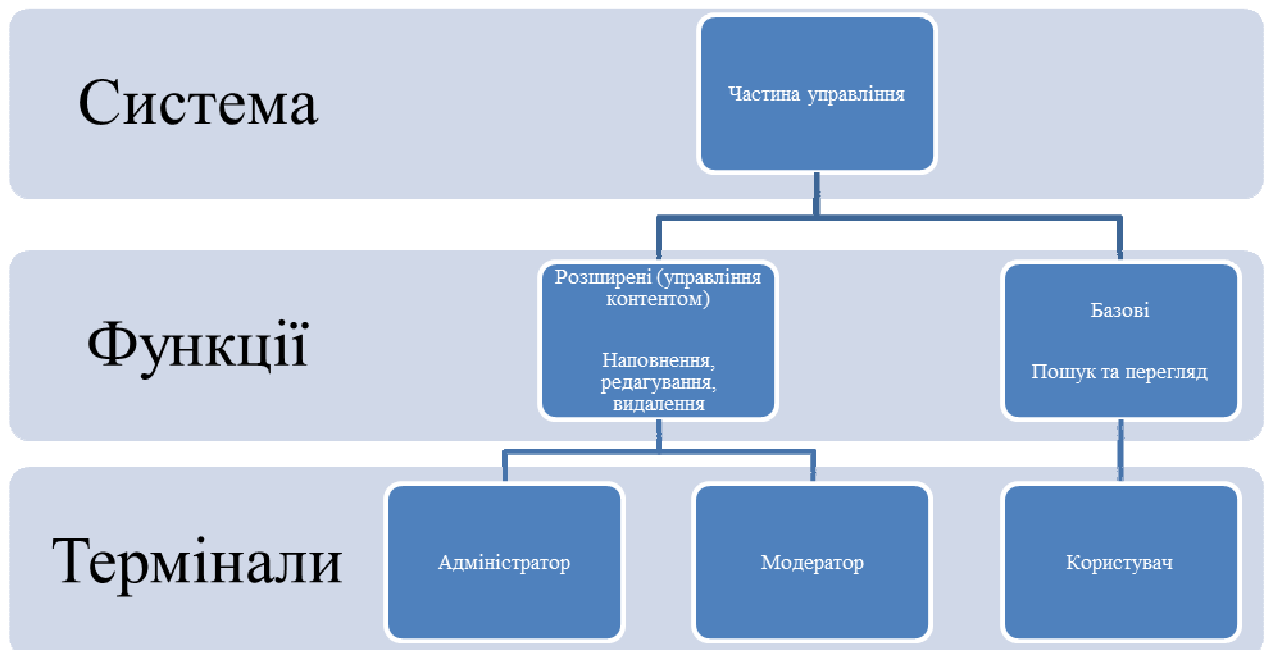


Рисунок 1.4 – Загальна схема розподілу функціоналу в проєкті

Доступ до контенту:

- простота доступу через форму входу;
- використання для входу зв'язки «логін-пароль».

Функції сайту систематизовано та перераховано в таблиці 1.3.

Таблиця 1.3 – Функції сайту

Переваги користувача	Підтримувані функції
Прискорення процесу обміну інформацією	Функція оповіщення про оновлення
Прискорення процесу пошуку інформації	Функція пошуку по сайту
Нові користувачі можуть швидко включитися в роботу	Детальна довідка допоможе в отриманні інформації по роботі з системою

1.4.3 Вимоги до інтерфейсу

Інтерфейс розроблюваної системи повинен бути легким у використанні, ергономічним, зрозумілим для користувача, захищеним від некоректних дій. Дизайн повинен бути сучасним а багатофункціональним:

- зрозумілість;
- зручність використання;
- мінімалізм.

1.4.4 Вимоги до продуктивності

Висуваються такі вимоги:

- очікування відгуку сервера в межах 8 сек;
- автооновлення сторінок.

Швидкодія:

- відмовостійкість;
- стабільність роботи системи.

1.4.5 Вимоги до безпеки

Захист даних:

- можливість наповнення контентом тільки адміністраторами і модераторами;
- захист паролів алгоритмом MD5.

1.4.6 Експлуатаційні вимоги

Висуваються такі вимоги:

- наявність браузера для перегляду сайту;
- наявність встановленого на клієнтській стороні Flashplayer;

- наявність вебсервера Denwer для тестування і налагодження.

1.4.7 Не функціональні вимоги до продукту

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- IBM-сумісна серверна робоча станція під управлінням встановленої та налагодженої серверної операційної системи (ОС);
- вебсервер: HTTP-сервер, або сервер Microsoft Internet Information або Netscape Enterprise;
- інтерпретатор PHP5;
- MySQL-сервер для керування користувачами та організації доступу до бази даних (БД).

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- IBM-сумісний персональний комп'ютер (ПК);
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – від 3GB (для безперебійної роботи браузерів);
- підключення до мережі Internet;
- встановлений браузер останньої версії.

1.5 Висновки за розділом 1

На першому етапі було проаналізовано існуючі рішення та досліджено їх переваги та недоліки. На основі існуючих недоліків було сформовано список конкурентоспроможних характеристик для майбутнього застосунку. Також було з'ясовано перелік стейкхолдерів та їх функціональних вимог.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

На попередньому етапі було досліджено предметну область: питання електронної комунікації здобувачів в рамках студентської групи, існуючі рішення для цих задач, їх переваги та недоліки. На основі цього вдалося сформулювати перелік стейкхолдерів та їх функціональних вимог до застосунку. Після чого було встановлено також перелік не функціональних вимог.

Цей розділ буде присвячено висвітленню питань вибору технології застосунку: технології, мова програмування, програмні надбудови, патерни проєктування, тощо. Визначення цих питань дозволить обґрунтувати наступні рішення з розробки застосунку.

2.1 Вибір інструментарію розробки

2.1.1 Вибір мови програмування

PHP – це універсальна серверна скриптова мова з відкритим вихідним кодом, призначена для розробки динамічних та інтерактивних вебсторінок. PHP розшифровується як Hypertext Pre-Processor, що означає, що ця мова має здатність обробляти всі дані, отримані з боку клієнта або користувача, перед відправкою назад у вигляді гіпертексту [16], [17].

Це одна з перших розроблених серверних скриптових мов, яка впроваджується у вихідний документ HTML замість виклику зовнішнього файлу для обробки даних [16], [17].

Розробка PHP мала великий вплив на створення динамічних вебсайтів, які пропонують розширену функціональність вебсторінки. PHP дозволяє вебсайтам виконувати обробку даних в режимі реального часу, що є основною причиною його популярності [16], [17].

Єдиною повною реалізацією PHP є оригінал, відомий просто як PHP. Він є найбільш широко використовуваним і працює на движку Zend. Щоб

відрізнити його від інших реалізацій, його іноді неофіційно називають Zend PHP [16], [17]. Движок Zend компілює вихідний код PHP на льоту у внутрішній формат, який він може виконувати, таким чином, він працює як інтерпретатор. Це також еталонна реалізація PHP, оскільки PHP не має формальної специфікації, і тому семантика Zend PHP визначає семантику PHP. Через складну та нюансовану семантику PHP, визначену тим, як працює Zend, конкуруючим реалізаціям важко забезпечити повну сумісність [16], [17].

Модель виконання PHP з одним запитом на сценарій і той факт, що движок Zend є інтерпретатором, призводять до неефективності; в результаті були розроблені різні продукти, які допомагають підвищити продуктивність PHP [16], [17]. Щоб прискорити час виконання і не компілювати вихідний код PHP при кожному зверненні до вебсторінки, PHP-скрипти також можуть бути розгорнуті у внутрішньому форматі PHP engine за допомогою кешу кодів операцій, який працює шляхом кешування скомпільованої форми PHP-скрипта (кодів операцій) в загальній пам'яті щоб уникнути накладних витрат на синтаксичний аналіз і компіляцію коду при кожному запуску скрипта. Кеш кодів операцій, Zend Opcode, вбудований в PHP починаючи з версії 5.5. Іншим прикладом широко використовуваного кешу кодів операцій є альтернативний кеш PHP (APC), який доступний як розширення PECL [16], [17].

У той час як Zend PHP як і раніше залишається найпопулярнішою реалізацією, було розроблено кілька інших реалізацій [16], [17]. Деякі з них є компіляторами або підтримують JIT-компіляцію і, отже, забезпечують переваги в продуктивності в порівнянні з Zend PHP за рахунок відсутності повної сумісності з PHP [16], [17]. Альтернативні варіанти реалізації включають наступне:

- HipHop Virtual Machine (HHVM) – розроблений у Facebook і доступний з відкритим вихідним кодом, він перетворює PHP-код у байт-код високого рівня (зазвичай відомий як проміжна мова), який потім динамічно

перетворюється на машинний код x86-64 під час виконання за допомогою JIT-компілятора, в результаті продуктивність покращилася в 6 разів. Однак, починаючи з версії 7.2, Zend перевершив HHVM, а HHVM 3.24 – остання версія, офіційно підтримує PHP [16], [17];

- Parrot – віртуальна машина, призначена для ефективного запуску динамічних мов; Pipp перетворює вихідний код PHP в проміжне представлення Parrot, яке потім перетворюється в байт-код Parrot і виконується віртуальною машиною [16], [17];

- ReachPie – компілятор другого покоління для байт-коду. Net Common Intermediate Language, побудований на платформі Roslyn; наступник Phalanger, що розділяє кілька архітектурних компонентів [16], [17];

- Phalanger-компілює PHP в .Байт-код Net Common Intermediate Language – попередник ReachPie [16], [17];

- Quercus-компілює PHP в байт-код Java;

- HipHop – розроблений у Facebook і доступний з відкритим вихідним кодом, він перетворює PHP-скрипти в код C++, а потім компілює отриманий код, знижуючи навантаження на сервер до 50%. На початку 2013 року Facebook відмовився від нього на користь HHVM з кількох причин, включаючи труднощі з розгортанням та відсутність підтримки всієї мови PHP, включаючи конструкції `create_function ()` та `eval ()` [16], [17].

Проведемо порівняння мов програмування для реалізації вебзастосунків, результати порівняння представимо в табличній формі (табл. 2.1). У якості критеріїв будемо використовувати наступні:

- простота вивчення – швидкість та простота початку розробки;
- матеріали та спільнота – наявність матеріалів та активна спільнота для підтримки розробників та поширення навчальних матеріалів;
- швидкість та простота розгортання застосунків – загальна оцінка швидкості та простоти розгортання вебзастосунку на сервері;

- програмні надбудови для розробки – наявність спеціалізованих програмних надбудов (систем керування вмістом (CMS) або фреймворків) для пришвидшення розробки;
- підтримка та масштабування застосунків – легкість та загальна можливість наступної підтримки застосунків.

Таблиця 2.1 – Порівняння мов програмування

Критерій	Python	PHP
Простота вивчення	++	++
Матеріали та спільнота	++	++
Швидкість та простота розгортання застосунків	+	++
Програмні надбудови для розробки	+	++
Підтримка та масштабування застосунків	+	++

З таблиці видно, що PHP є найоптимальнішим вибором для розробки, а наявна велика база фреймворків та CMS гарантує спрощення, пришвидшення та надійність розробки.

2.1.2 Система керування вмістом

Згідно з визначеними раніше вимогами та проектуванням системи, вибір зупиняємо на Joomla [18], [19].

Основні переваги даної системи управління вмістом:

- можливість створювати необмежену кількість сторінок [18], [19];
- чітка організація структури сайту;
- для кожної динамічної сторінки можна створити свій опис і ключові слова з метою підвищення рейтингу в пошукових системах [18], [19];

- початок і закінчення публікації будь-яких матеріалів можна запрограмувати за календарем [18], [19];
- можливість обмежити доступ до певних розділів Сайту тільки для зареєстрованих користувачів [18], [19];
- секції новин, продукції або послуг легко редаговані і керовані;
- повністю настроюються схеми розташування елементів по 5 областям, що дозволяє зробити оригінальний дизайн [18], [19];
- закачування зображень за допомогою браузера в вашу власну бібліотеку - для подальшого використання з будь-якого місця сайту [18], [19];
- опитування та голосування для ефективного зворотного зв'язку;
- різні модулі-такі як останні новини, лічильник відвідувань, докладна статистика відвідувань, гостьова книга, форум і т. д. [18], [19];
- причому ви самі вибираєте, чи будуть показані ці модулі і на яких сторінках [18], [19];
- можливість створення не однієї, а декількох форм зворотного зв'язку для кожного контакту [18], [19];
- зміна порядку об'єктів, включаючи новини, питання, статті і т. д;
- генератор показу випадкової новини;
- модуль прийому від віддалених авторів новин, статей і посилань [18], [19].

Діаграма взаємодії між компонентами Joomla зображена на рис. 2.1.

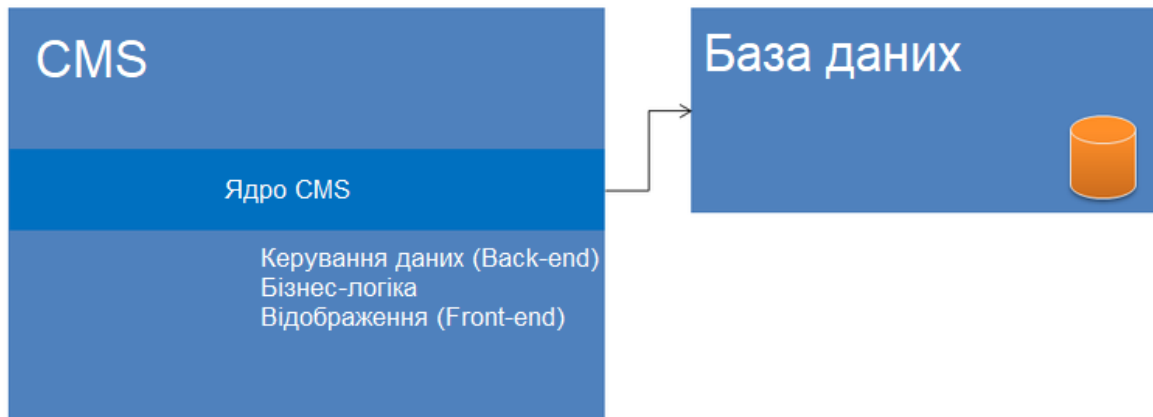


Рисунок 2.1 – Діаграма взаємодії між компонентами Joomla

2.1.3 Розподіл та реалізація бізнес-логіки

Сама організація Joomla має модульні компоненти, що обумовлює специфічну внутрішню організацію сайту. Для кращого розуміння функцій системи наведемо діаграму класів (рис. 2.1).

Найбільш повно організацію підключаються компонентів (як видно з діаграми-це будуть модулі коментарів, чату, пошуку і голосування) відображають діаграми пакетів і компонентів/

Кожен модуль взаємодіє з БД сайту, зберігаючи там свої результати роботи (крім модуля пошуку, який повертає в якості підсумку запиту HTML-сторінку). У свою чергу, інтерпретацією коду модулів займається безпосереднє вебсервер і його розширення для перетворення js, jsp php-скриптів [18], [19].

Модуль чату передбачає відправку на сервер тексту повідомлення, зберігання логів в БД і відображення їх у своєму функціональному вікні.

Модуль голосування надає інтерфейс у вигляді списку вибору варіантів відповіді на опитування, також зберігає свої дані в БД, відправляючи код вибору на інтерпретацію [14], [15], [18], [19].

Модуль коментарів по функціональності схожий з модулем чату, прикріплює текстові повідомлення до коментованого матеріалу, підтримує також кілька гілок коментарів, дозволяючи здійснювати повноцінні обговорення [14], [15], [18], [19].

Модуль пошуку приймає в якості параметрів запиту ключові слова і повертає масив посилань на знайдені матеріали у вигляді HTML-сторінки.

2.1.4 Обґрунтування вибору CMS для розробки вебресурсу

Порівняльна характеристика найпоширеніших CMS для розробки інтернет-магазинів наведена у таблиці 2.2.

Серед критеріїв порівняння будемо розглядати наступні:

- простота розгортання – простота розгортання та старту подальшої роботи із CMS;
- можливість персоналізації застосунку для адміністратора – гнучкість CMS до побажань та вимог адміністратора;
- вимоги до хостингу – загальні вимоги CMS до хостингу для розгортання застосунку;
- підтримка розширень – наявність та простота застосування розширень для спрощення розробки та захисту даних в застосунку.

Таблиця 2.2 – Порівняння php-фреймворків

	WordPress	Drupal	Joomla!
Простота розгортання	++	+	++
Можливість персоналізації застосунку для адміністратора	++	+	++
Вимоги до хостингу	±	+	++
Підтримка розширень	+	+	++

З порівняння наведеного в таблиці можна зробити висновок, що використання Joomla! є найбільш оптимальним варіантом. Це пов'язано із легкістю розгортання CMS та мінімальними вимогами до хостингу. Також важливо відмітити низький рівень проблем із сумісністю плагінів на відміну від Wordpress, наприклад.

2.2 Структурна схема розробки

Інтерфейс користувача містить тільки форми опитування даних, у вигляді полів, посилань, радіокнопок і т.д. безпосередньою їх обробкою займається ПЗ на стороні сервера [14], [15], [18], [19].

Згідно спроектованої раніше концепції, показаної на діаграмі компонентів (рис. 2.2), попередня обробка даних користувацьких форм на стороні клієнта буде виконуватися модулями, які, в свою чергу, обробляються вебсервером як окремі інтерпретовані програми

Оброблені дані формують транзакції, що направляються в БД для збереження і формують відгук(у вигляді повідомлення або HTML-сторінки), який відправляється клієнту на відображення успішності\неуспішності скоєних дій, наприклад, входу на сайт або додавання коментаря [14], [15], [18], [19].

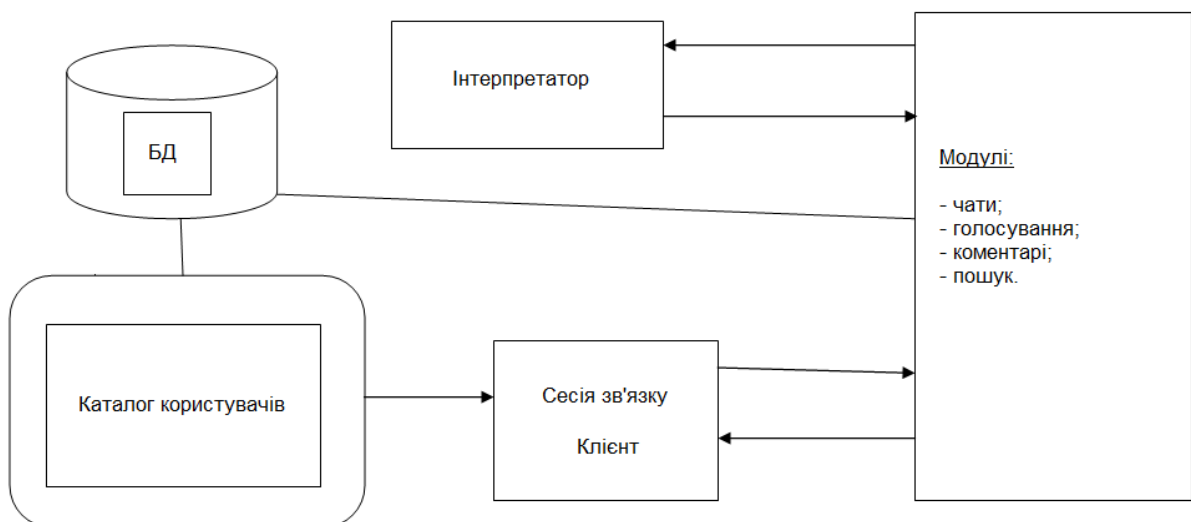


Рисунок 2.2 – Архітектура вебзастосунку

2.3 Висновки за розділом 2

В цьому розділі було досліджено питання вибору технології розробки застосунку:

- так у якості технології обрано – розробку вебзастосунку;
- у якості мови програмування – серверну мову PHP;
- у якості програмної надбудови – CMS Joomla.

Патерн проектування MVC в повній мірі дозволить організувати та розподілити бізнес логіку застосунку.

Визначення цих питань дозволяє обґрунтувати наступні рішення з розробки застосунку.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проєктування моделей системи

Unified Modeling Language (UML) — це мова моделювання загального призначення [20], [21]. Основна мета UML - визначити стандартний спосіб візуалізації того, як була розроблена система. Він дуже схожий на креслення, які використовуються в інших галузях техніки [20], [21].

UML – це не мова програмування, це скоріше візуальна мова. Ми використовуємо діаграми UML, щоб зобразити поведінку та структуру системи. UML допомагає інженерам програмного забезпечення, бізнесменам і системним архітекторам моделювати, проєктувати та аналізувати [20], [21]. Група управління об'єктами прийняла уніфіковану мову моделювання як стандарт у 1997 році. Відтоді нею керує OMG. Міжнародна організація зі стандартизації опублікувала UML як затверджений стандарт у 2005 році. UML переглядався протягом багатьох років і періодично переглядається [20], [21].

Діаграма класів UML – це графічне позначення, яке використовується для побудови та візуалізації об'єктно-орієнтованих систем [20], [21]. Діаграма класів на UML – це тип діаграми статичної структури, яка описує структуру системи, показуючи:

- класи;
- їх атрибути;
- операції (або методи);
- і відносини між об'єктами [20], [21].

Діаграма класів для застосунку зображена на рис. 3.1. В даному випадку існує два батьківських класу: Користувач і Модуль. Вони являють собою узагальнення для подальшої деталізації щодо прав доступу та підтипів модулів.

Основним класом є клас Сайт, що містить послідовності, в які входить каталог сторінок, модулі, дані користувачів.

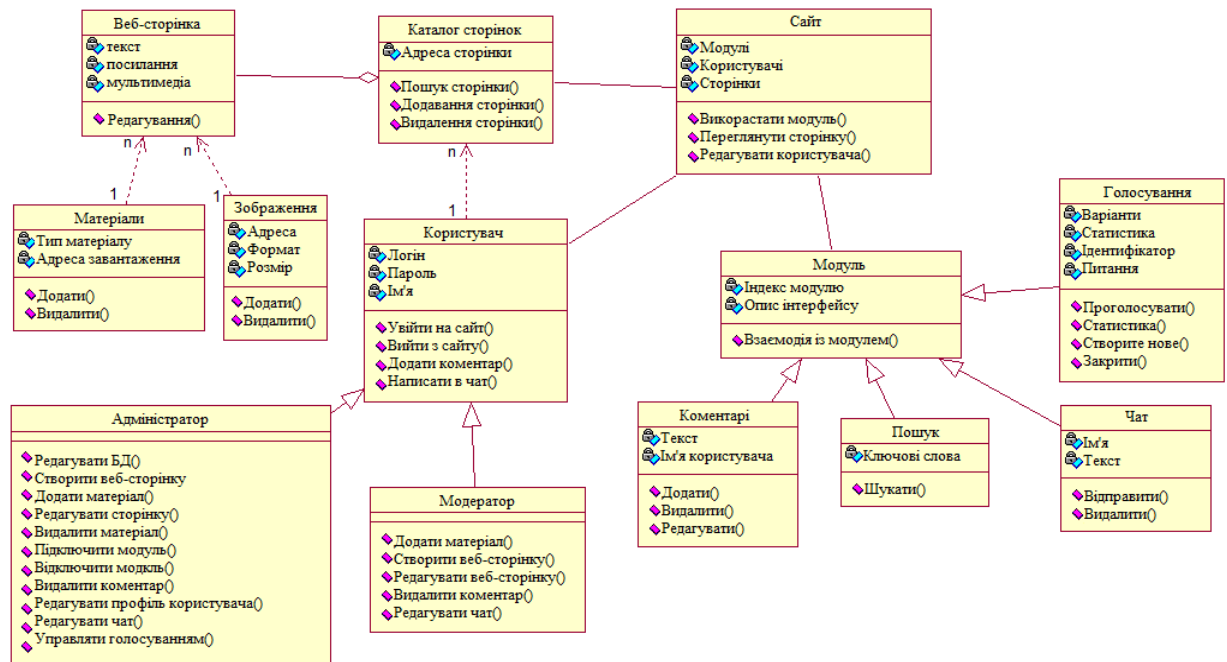


Рисунок 3.1 – Діаграма класів

Діаграми пакетів – це структурні діаграми, які використовуються для показу організації та розташування різних елементів моделі у вигляді пакетів [20], [21]. Пакет – це групування пов'язаних елементів UML, таких як діаграми, документи, класи чи навіть інші пакети. Кожен елемент вкладений в пакунок, який зображений у вигляді папки файлів на діаграмі, а потім впорядкований ієрархічно в межах діаграми. Діаграми пакетів найчастіше використовуються для наочної організації багатoshарової архітектури в будь-якому класифікаторі UML, наприклад, у програмній системі [20], [21].

Діаграма пакетів зображена на рис. 3.2. В даному випадку існує чотири пакети, один з яких безпосередньо залежить від трьох інших. Такий поділ на пакети обумовлено функціональним призначенням кожної з груп.

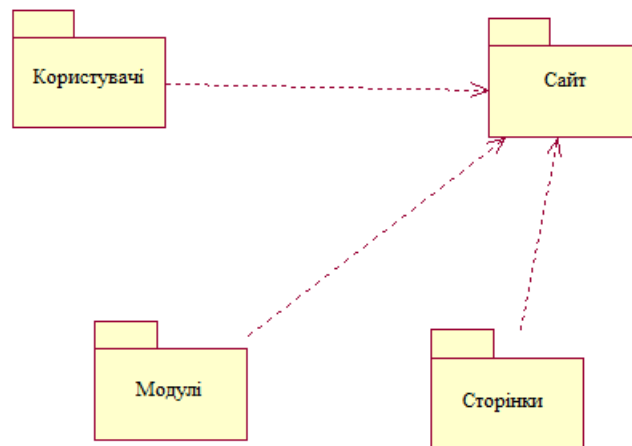


Рисунок 3.2 – Діаграма пакетів

Мета діаграми компонентів – показати взаємозв’язок між різними компонентами в системі. Для цілей UML 2.0 термін «компонент» відноситься до модуля класів, які представляють незалежні системи або підсистеми з можливістю взаємодії з рештою системи [20], [21]]. Існує цілий підхід до розробки, який обертається навколо компонентів: розробка на основі компонентів. У цьому підході діаграми компонентів дозволяють планувальнику ідентифікувати різні компоненти, щоб вся система виконувала те, що вона повинна робити. Частіше, в підході до програмування, діаграма компонентів дозволяє старшому розробнику групувати класи разом на основі спільної мети, щоб розробник та інші могли дивитися на проект розробки програмного забезпечення на високому рівні [20], [21].

Діаграми компонентів для застосунку зображена на рис. 3.3. Все передача даних між компонентами ведеться через інтерфейс. У свою чергу, саме сервер вирішує питання про передачу запитів і скриптів інтерпретатору, і результат передається через інтерфейс клієнту.

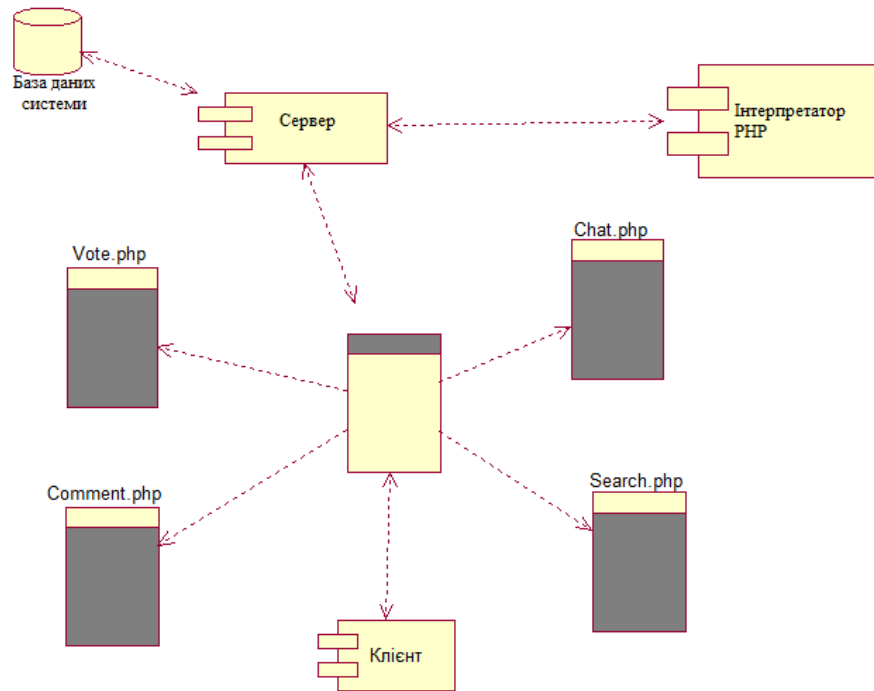


Рисунок 3.3 – Діаграма компонентів

Діаграми кооперації використовуються, щоб показати, як об'єкти взаємодіють, щоб виконувати поведінку конкретного варіанту використання або частини варіанту використання [20], [21]. Поряд із діаграмами послідовності, кооперація використовується дизайнерами для визначення й уточнення ролей об'єктів, які виконують певний потік подій у варіанті використання [20], [21]. Вони є основним джерелом інформації, яка використовується для визначення обов'язків класу та інтерфейсів [20], [21].

Діаграми кооперації зображені на рис. 3.4 та 3.5.

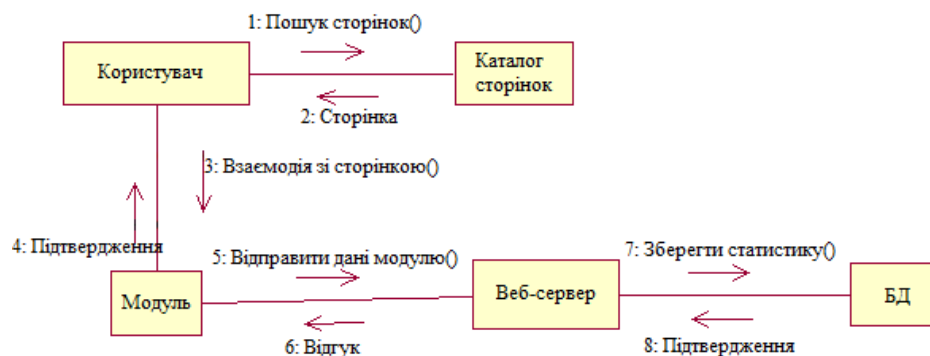


Рисунок 3.4 – Діаграма кооперації для користувачів



Рисунок 3.5 – Діаграма кооперації для адміністратора

Діаграми послідовностей зображені на рис. 3.6 та 3.7.

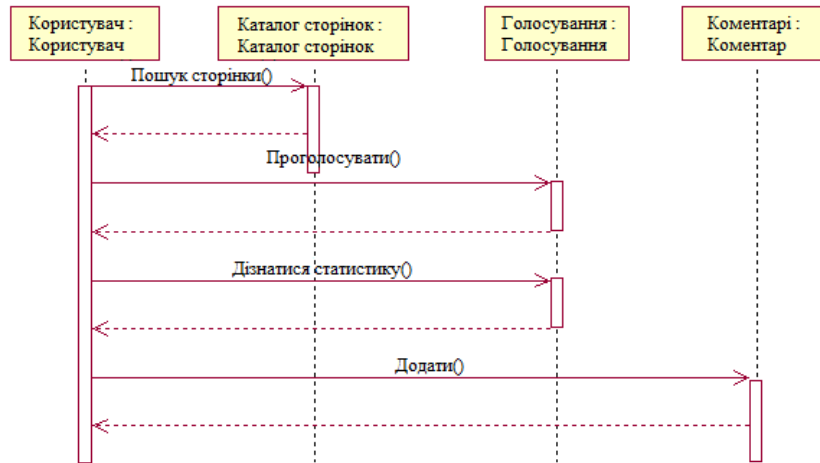


Рисунок 3.6 – Пошук диску авторизованим користувачем

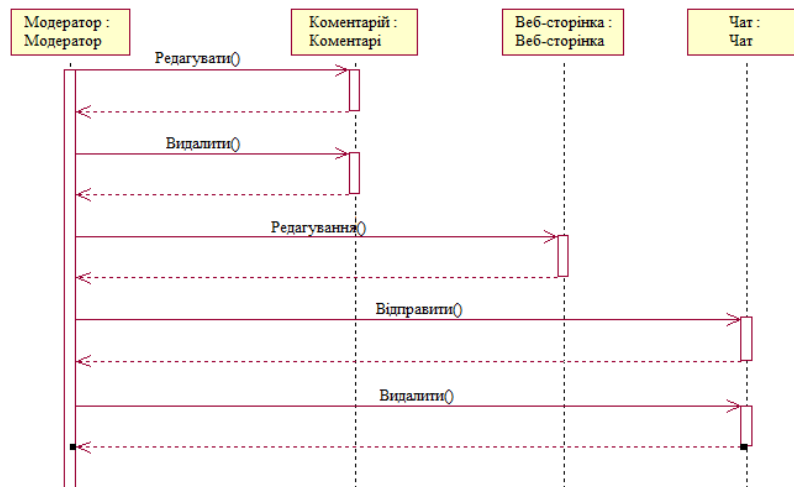


Рисунок 3.7 – Оформлення замовлення авторизованим користувачем

3.2 Проктування інтерфейсу

Сайт, згідно з вимогами технічного завдання, повинен являти собою блокову структуру, тобто бути розділеним на інформаційні зони. У технічному завданню був представлений попередній макет, згідно з яким був перероблений і побудований остаточний (рис. 3.8).

Блоки чату і голосування є динамічними, тобто після натискання відповідного посилання розкриваються і згортаються. Таким чином, області матеріалів будуть змінювати свій розмір, в залежності від потреб користувача.

Блок матеріалів – це основне місце відображення необхідної інформації, будь то каталог статей \ новин \ іншої інформації.

Профіль користувача містить наступну інформацію: Ім'я користувача, логін, Дата реєстрації, Дата останнього входу, використовуваний редактор статей(для адміністрації), часовий пояс. Увійти на сайт можна за допомогою форми входу. Якщо Користувач не зареєстрований, то можна залишити заявку на реєстрацію (рис. 3.8), яку повинен підтвердити адміністратор. Форма реєстрації або входу відображається замість головної сторінки сайту.

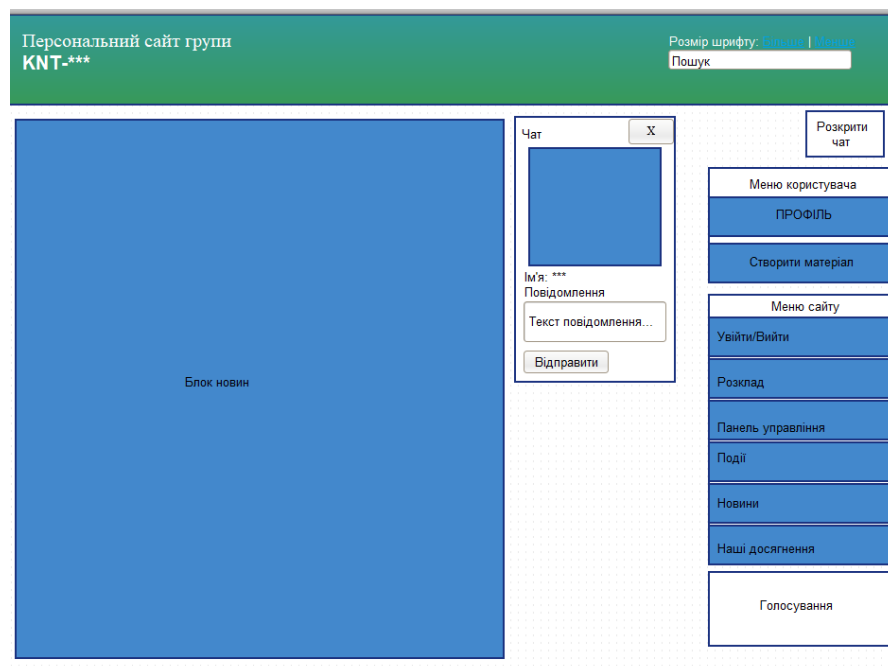


Рисунок 3.8 – Головний прототип сайту

На рисунку 3.9 наведено прототип входу до системи.

The image shows a wireframe of a login page. At the top, there is a green header bar with the text "Персональний сайт групи KNT-***" on the left and "Розмір шрифту: [Збільше](#) | [Зменше](#)" and a search box labeled "Пошук" on the right. The main content area is a large white rectangle with a blue border. Inside this area, there is a smaller white box with a blue border containing the login form. The form has two text input fields: "Логін:" and "Пароль:", each followed by a "Text box" placeholder. Below these is a "Увійти" button. To the right of the main content area, there is a vertical sidebar with a blue border containing two buttons: "Меню сайту" and "Вхід на сайт".

Рисунок 3.9 – Прототип входу до системи

На рисунку 3.10 представлено прототип реєстрації.

The image shows a wireframe of a registration form. It consists of a white box with a blue border. Inside, there are five text input fields, each with a "Text box" placeholder. The labels for the fields are "Логін", "Пароль", "Повторний пароль", "e-mail", and "Повторний e-mail". At the bottom of the box, there are two buttons: "Реєстрація" and "Скасувати".

Рисунок 3.10 – Прототип реєстрації

Розклад містить список щоденних занять. За замовчуванням відображається на головній сторінці.

Події-містять звіти про минулі або оголошення про майбутні заходи. Кожне оголошення \ звіт оформляється у вигляді окремої статті.

Новини-містить інформацію, необхідну для донесення до аудиторії. Як і інші каталоги матеріалів \ статей, підлягає редагуванню тільки з боку адміністрації.

Наші досягнення-містить інформацію по досягненням в різних областях членів групи, також являє собою каталог матеріалів.

Панель управління-кнопка входу в панель адміністрування сайтом.

Чат являє собою динамічний блок, що містить поле введення тексту, інформацію про відправника, кнопку відправки і форму виведення всіх повідомлень.

Голосування (рис. 3.11) являє собою список прапорців одиночного вибору з варіантами відповідей, які підтверджуються кнопкою. Створення голосування доступне через панель адміністрації за допомогою спеціального модуля.

Two wireframe prototypes of a voting interface. The left prototype shows a question "Чи задоволені ви дизайном?" with two radio buttons labeled "Так" and "Ні", and a button labeled "Vote>>". The right prototype shows the same question with two progress bars, one for "Так" and one for "Ні", indicating the current state of the vote.

Рисунок 3.11 – Прототип голосування

На рисунку 3.12 наведено прототип авторизації.

A wireframe prototype of a login form. It features two text input fields: "Логін" (Login) and "Пароль" (Password), both labeled "Text box". Below the fields is a button labeled "Увійти" (Login).

Рисунок 3.12 – Прототип авторизації

3.3 Висновки за розділом 3

Було проведено ознайомлення з видами структурних діаграм та особливостями їх застосування, розроблено абстрактну модель системи мовою UML на основі структурних діаграм. Розроблено абстрактну модель системи за обраною темою на основі діаграму класів, пакетів та компонентів, на основі діаграм поведінки, прецедентів, діяльності та станів, кооперацій та послідовностей.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Опис застосунку

Функціональне призначення: сайт надає умови для обміну інформацією, з підтримкою функції оповіщення користувачів і онлайн обміну повідомленнями.

Експлуатаційне призначення сайту:

- надання інформації про групу;
- надання інформації всередині групи;
- обмін та обговорення інформації всередині групи.

Системні вимоги до використання: наявність інтернет-браузера.

4.2 Умови виконання програми

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- IBM-сумісна серверна робоча станція під управлінням встановленої та налагодженої серверної ОС;
- вебсервер: HTTP-сервер, або сервер Microsoft Internet Information або Netscape Enterprise;
- інтерпретатор PHP5;
- MySQL-сервер для керування користувачами та організації доступу до БД.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- IBM-сумісний ПК;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – від 3GB (для безперебійної роботи браузерів);

- підключення до мережі Internet;
- встановлений браузер останньої версії.

4.3 Інструкція по експлуатації програми

4.3.1 Характеристики програми

Логічна взаємодія найбільш повно представлено на рис. 4.1.

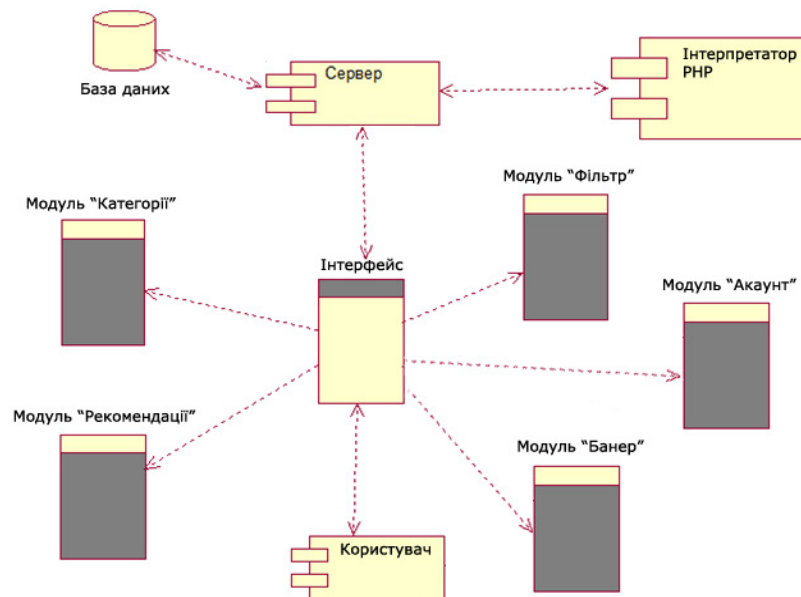


Рисунок 4.1 – Компоненти та зв'язки

Оброблені дані формують транзакції, що направляються в БД для збереження і формують відгук(у вигляді повідомлення або HTML-сторінки), який відправляється клієнту на відображення успішності\неуспішності скоєних дій, наприклад, входу на сайт або додавання коментаря.

При цьому ніякий модуль не може функціонувати, якщо ідентифікатора користувача, що звертається до нього, немає в базі даних. Таким чином, модулі голосування, коментарів, чату і пошуку будуть недоступні для сторонніх користувачів, однак скрипт входу на сайт, виконаний у вигляді інтегрованого в середу модуля, буде доступний.

Найбільш повно концепція логіки відображена на діаграмі класів, за допомогою функцій, доступних тим чи іншим категоріям користувачів (рис. 2.1).

Основний поділ функціональності проводиться за категоріями користувачів. Прості користувачі, тобто особи, зареєстровані в системі, мають можливість незворотної взаємодії з модулями: залишення коментаря без можливості редагування, відправка повідомлень в чат без можливості видалення, і т. д. [18], [19].

Особі категорії «Модератор» мають можливість редагувати результати взаємодії з модулями, свої або інших користувачів.

Особі категорії «Адміністратор» мають можливість глибинної взаємодії з системою, тобто додавання\видалення модулів, внесення безпосередньо змін до БД сайту [18], [19].

Кожен модуль взаємодіє з БД сайту, зберігаючи там свої результати роботи (крім модуля пошуку, який повертає в якості підсумку запиту HTML-сторінку). У свою чергу, інтерпретацією коду модулів займається безпосереднє вебсервер і його розширення для перетворення js, jsp php-скриптів [18], [19].

Модуль чату передбачає відправку на сервер тексту повідомлення, зберігання логів в БД і відображення їх у своєму функціональному вікні.

Модуль голосування надає інтерфейс у вигляді списку вибору варіантів відповіді на опитування, також зберігає свої дані в БД, відправляючи код вибору на інтерпретацію [18], [19].

Модуль коментарів по функціональності схожий з модулем чату, прикріплює текстові повідомлення до коментованого матеріалу, підтримує також кілька гілок коментарів, дозволяючи здійснювати повноцінні обговорення [18], [19].

Модуль пошуку приймає в якості параметрів запиту ключові слова і повертає масив посилань на знайдені матеріали у вигляді HTML-сторінки.

4.3.2 Використані технічні засоби

Вся робота сайту базується на основі CMS Joomla! [18], [19].

Основні переваги даної системи управління вмістом [18], [19]:

- можливість створювати необмежену кількість сторінок;
- чітка організація структури сайту;
- для кожної динамічної сторінки можна створити свій опис і ключові слова з метою підвищення рейтингу в пошукових системах;
- можливість обмежити доступ до певних розділів Сайту тільки для зареєстрованих користувачів;
- повністю настроюються схеми розташування елементів по 5 областям, що дозволяє зробити оригінальний дизайн;
- закачування зображень за допомогою браузера в вашу власну бібліотеку - для подальшого використання з будь-якого місця сайту;
- опитування та голосування для ефективного зворотного зв'язку.
- Відповідно, дана система має наступні характеристики:
- повністю заснований на БД движок з використанням PHP / MySQL;
- модуль безпеки для багаторівневої аутентифікації користувачів / адміністраторів;
- робота під Linux, FreeBSD, MacOSX, Solaris, AIX, SCO, WinNT, Win2K.

В якості системи тестування був обраний Denwer-це набір програм для створення сайту на локальному комп'ютері, без виходу в інтернет [18], [19]. Він включає в себе:

- Apache, SSI, mod_rewrite, mod_php;
- PHP4 з підтримкою GD і MySQL;
- MySQL з підтримкою транзакцій (mysqld-max);

- систему управління віртуальними хостами, засновану на шаблонах;
- систему управління запуском і завершенням;
- phpMyAdmin - систему управління MySQL через Вебінтерфейс;
- ядро Perl без стандартних бібліотек (вони поставляються окремо);
- емулятор sendmail, підтримується робота спільно з PHP і Perl;
- інсталятор.

4.3.3 Звернення до програми

Для завантаження сайту і взаємодії з ним досить набрати в адресному рядку браузера «knt123.nuzp». Для входу в панель управління: «knt123.nuzp \administrator».

4.3.4 Вхідні й вихідні дані

Програма не має конкретних вхідних даних для обробки, і ключем до доступу до роботи з нею є комбінація з логіна і пароля користувача\адміністратора. Спосіб кодування пароля-MD5.

4.3.5 Розширення функціональності вебзастосунку за допомогою модулів

Модуль голосування. AcePolls – дуже хороший компонент для опублікування свого голосування і опитування на сайті, що працює на движку Joomla. Елементи налаштування [18], [19].

Вкладка «Детальніше». «Заголовок» – сюди вписуємо назву вашого опитування або голосування, тобто саме питання, яке ви будете задавати відвідувачам вашого сайту [18], [19].

«Аліас» – це поле відповідає за створення людино-зрозумілого посилання для нашого голосування. Можна не писати-компонент сам напише ваш заголовок латинськими літерами після того, як ви збережете своє опитування [18], [19].

«ЛАГ» – сюди вписуйте проміжок часу в годинах, яке повинно пройти, щоб один і той же чоловік міг знову проголосувати в цьому опитуванні, якщо він вже до цього голосував [18], [19].

«Дата початку» і «Дата закінчення» – дуже хороша настройка, вона дозволяє обмежити ваше голосування в часі. Це може вам стане в нагоді, наприклад, якщо ваше опитування стосується будь-якої події, що триває деякий час [18], [19]. Наприклад, ви створюєте опитування про який-небудь товар, термін продажу якого обмежений. Відповідно «Дата початку» це день і час, коли почне працювати опитування, а «Дата закінчення» день і час, коли проголосувати буде вже не можна [18], [19].

«Опубліковано» – опублікувати ваше опитування, або поки не потрібно [18], [19].

Налаштування вкладки «Основні». «Тільки зареєстровані користувачі можуть голосувати» – якщо у вас на сайті є Реєстрація, то ви можете дозволити голосування тільки для зареєстрованих користувачів, а також і для гостей сайту [18], [19].

«Один користувач – один голос» – якщо ви дозволили голосувати тільки зареєстрованим, то тут ви можете налаштувати голосування так, щоб одна людина могла голосувати тільки раз [18], [19].

«Один голос з однієї IP адреси» – дана опція дозволяє заборонити голосувати двічі з одного IP. Я завжди для реальних голосувань ставити «так» [18], [19].

«Показати повідомлення» – якщо користувач, з якихось причин не може проголосувати, припустимо, що він вже відповідав на це опитування, то ви можете повідомити його про це за допомогою спеціальних повідомлень закладених в компоненті [18], [19].

«Показувати опитування в компоненті» – чи показувати опитування з можливістю голосування, на сторінці результатів голосування в компоненті AcePolls [18], [19].

Налаштування вкладки «Результати». Відповідає за Налаштування результатів опитування, коли вони відображаються на окремій сторінці [18], [19].

«Показати результат як» – тут доступно два варіанти відображення результатів голосування [18], [19].

«Діаграма опитування» – результат буде показаний у вигляді кругової діаграми прямо в модулі опитування [18], [19].

Модуль чату. Аjax-чат був обраний як додаток для управління чатом. Налаштування має наступні [18], [19].

Кешування-власне там і налаштування не вибираються, так як чат втрачає будь-який сенс, якщо почати кешувати його дані. Чат повинен працювати в живу, залишаємо цю настройку і рухаємося далі [18], [19].

Суфікс класу модуля - цей пункт Налаштування загальний для всіх модулів Joomla по ньому потрібно робити окрему статтю, тут зупинятися на ньому не буду, єдине скажу що за допомогою цієї настройки можна задати кожному модулю своє унікальне оформлення [18], [19].

Display a URL input field-чи показувати поле введення для url сайту. Якщо стоїть «Вкл» то користувачі зможуть вписати туди адресу свого сайту і їх імена в чаті будуть гіперпосиланнями на цей сайт. Рекомендую даний параметр вимикати, для захисту від спаму [18], [19].

Коментарі. В якості розширення для коментарів використовується компонент JComments.

Компонент JComments дозволяє відвідувачам сайту залишати коментарі до різних матеріалів, розміщених на сайті. Гнучка система налаштування прав доступу до функцій компонента, велика кількість налаштувань зовнішнього вигляду, підтримка шаблонів, використання технології AJAX і багато іншого [18], [19].

Основні функції [18], [19]:

- включення/відключення коментарів як для цілої категорії, так і для конкретного матеріалу (в тому числі і статичних);
- налаштування автопублікації коментарів для існуючих груп користувачів;
- налаштування прав на редагування/видалення коментарів для існуючих груп користувачів;
- підтримка захисту від флуду і спамботів;
- повідомлення адміністраторам сайту про нові коментарі;
- можливість підписки користувачів на повідомлення про нові коментарі;
- можливість редагування коментарів з фронтенда;
- можливість посторінкового відображення коментарів на сторінки та налаштування відображення положення номерів сторінок;
- налаштування порядку виведення коментарів (показувати останні коментарі спочатку або в кінці списку);
- підтримка смайлів, з можливістю налаштування порядку відображення смайлів в списку і відповідність зображення смайла довільної комбінації символів;
- підтримка BBCode, з можливістю налаштування прав доступу до тегів різних груп користувачів;
- підтримка гарячих клавіш для відправки коментарів і вставки тегів BBCode;
- використання технології AJAX дозволяє додавати, видаляти і редагувати коментарі без перевантаження сторінки;
- можливість показувати користувачам певних груп правила додавання коментарів;
- підтримка RSS-стрічок коментарів для кожного матеріалу і загальної RSS-стрічки для всіх коментарів;

- підтримка шаблонів оформлення для налаштування зовнішнього вигляду;
- функція конвертації з трансліта в кирилицю на льоту при додаванні коментаря;
- можливість налаштування відображення у формі редагування і перевірки заповнення полів «Адреса сайту» і «E-mail»;
- найпростіший фільтр нецензурних виразів;
- підтримка багатомовності (Російська, Українська, Білоруська, Болгарська, Англійська, Німецька, Грецька, Іврит, Латиська);
- підтримка пошуку по коментарям за допомогою пошукового спамбота.

Допоміжні функції [18], [19]:

- автоматична заміна повторюваних перекладів рядків на один;
- настройка обмеження довжини імені користувача;
- можливість вказувати заборонені до використання імена користувачів;
- можливість задати максимально допустиму довжину коментаря;
- виділення зовнішніх посилань у тексті коментарів;
- перевірка на наявність дублікатів при додаванні коментарів;
- вбудований захист e-mail адрес в коментарях від спамботів;
- механізм боротьби з дуже довгими посиланнями і словами, що дозволяє уникнути розриву дизайну.

Встановлення модулів. Щоб встановити модуль в Joomla! потрібний:

- увійти в Адмінцентр сайту;
- у закладці «Розширення» вибрати пункт «Установка/Видалення»;
- на сторінці потрібно завантажити скачаний архів з модулем. Для цього, натиснути «огляд» і вибрати файл на комп'ютері;
- після цього натиснути на кнопку «Завантажити та встановити».

Щоб відобразити модуль на сайті треба [18], [19]:

- пройти в наступний розділ: Адмінцентр, потім Розширення, і нарешті Менеджер модулів;
- тут знайти наш модуль, який встановлювали;
- натискаємо на його назву, переходимо в Налаштування;
- тепер ми можемо редагувати конфігурації модуля (змінювати назву, положення і т.п.);
- активуємо галку над пунктом «включений» (в новіших версіях-опублікований);
- вибираємо місце в шаблоні. З'являються назви блоків. Шукаємо потрібний, і вибираємо позицію.

4.4 Методика та результати тестування

Тестування інформаційного порталу виконаємо в декілька етапів. На початку протестуємо функціональність застосунку, а на другому етапі виконаємо тестування за допомогою спеціального ПЗ –WAPT [22].

4.4.1 Тестування функціоналу

Для завантаження сайту і взаємодії з ним досить набрати в адресному рядку браузера «knt123.nuzp».

Користувач може взаємодіяти з рядком пошуку, панеллю меню а також додатковими функціями (рис. 4.2).

Пункт «Ваш профіль» відобразить основну інформацію про Користувача – ім'я, дату реєстрації, інформацію про налаштування сайту.

Пункт «Створити допис» доступний тільки користувачам-адміністраторам, що дозволяє розмістити на сайті новину.

Пункт «Вхід на сайт» дозволяє вийти з профілю/увійти заново.

Пункт «Панель управління» доступний тільки адміністраторам і перенаправляє в панель управління сайтом на Joomla!

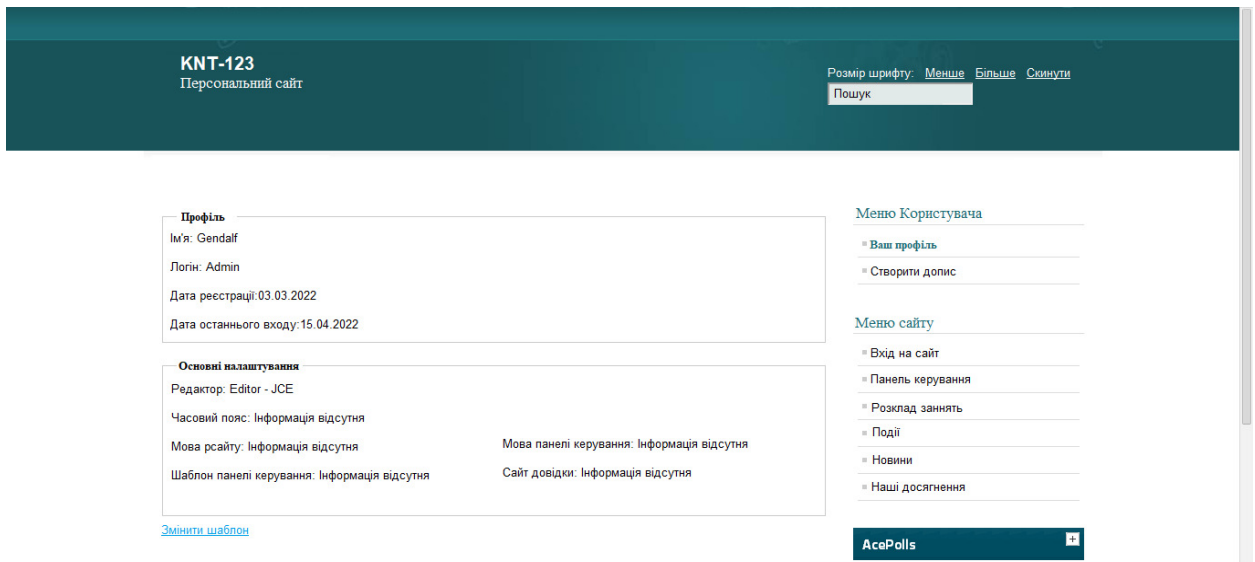


Рисунок 4.2 – Головна сторінка

Пункт «Розклад занять» відображає головну сторінку з поточним розкладом.

Пункт «Події» відображає список публікацій про недавні події в групі.

Пункт «Новини» містить список оновлень на сайті.

Пункт «Наші досягнення» представляє інформацію про успіхи групи.

Крім того, користувачам доступна взаємодія з коментарями статті (рис. 4.3. та 4.4).

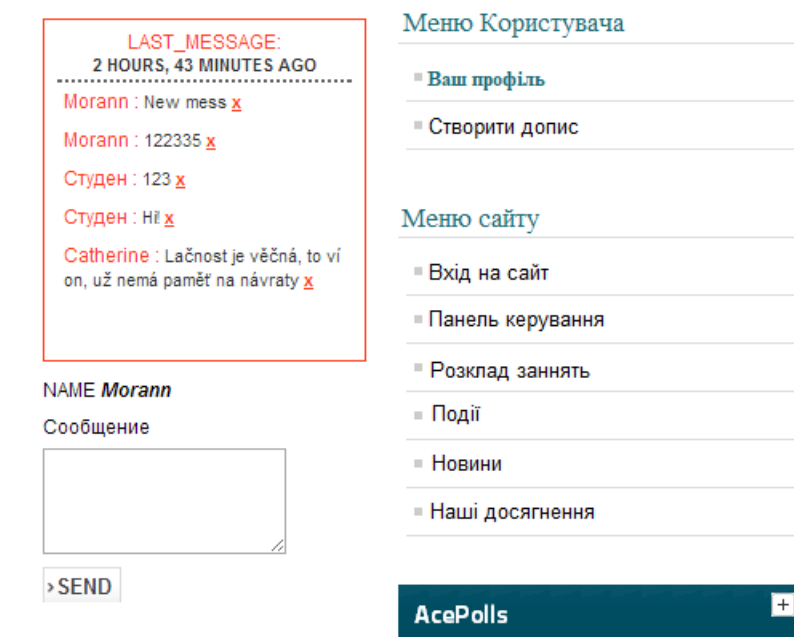


Рисунок 4.3 – Категорії взаємодії

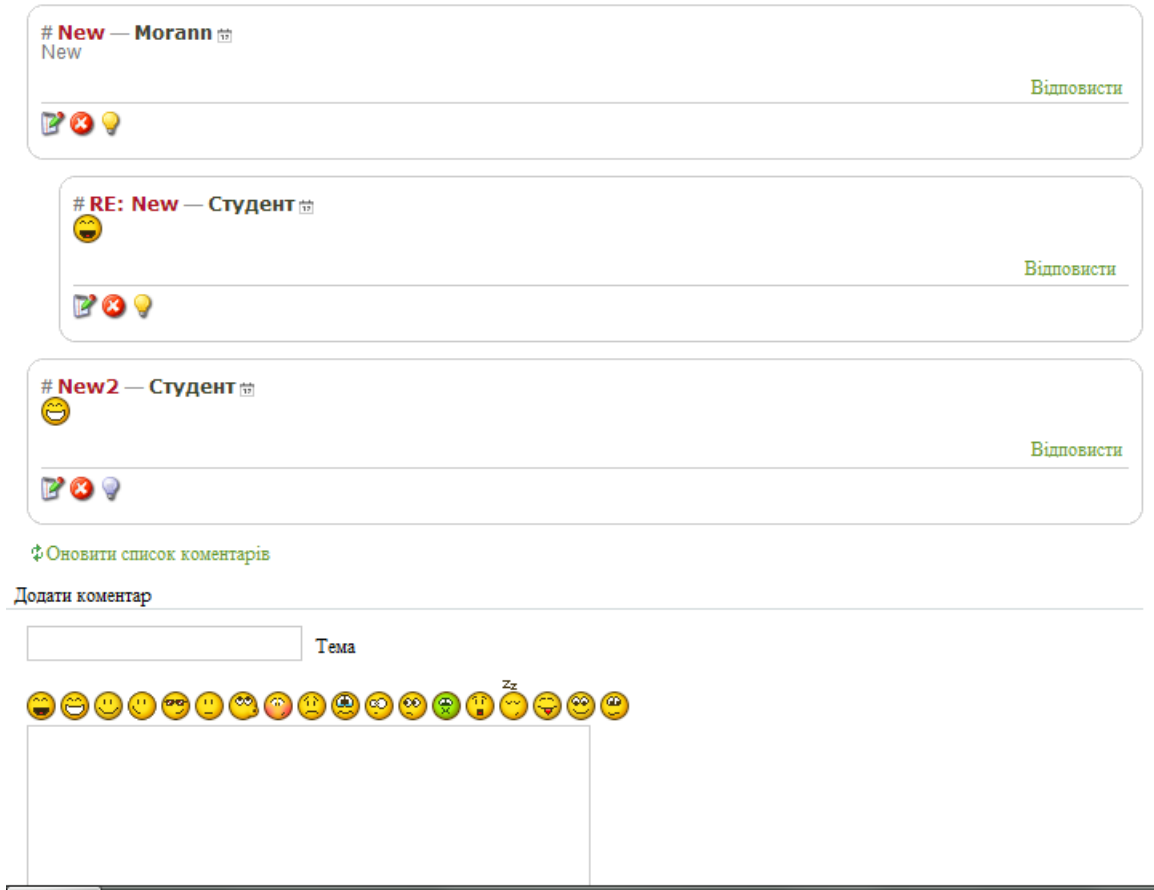


Рисунок 4.4 – Взаємодія із коментарями

Після заповнення пункту «Тема» і основного поля з'являється можливість додати коментар під статтю. Є можливість прокоментувати вже існуючий коментар, для цього треба натиснути на кнопку «Відповісти» у необхідного коментаря.

Також користувач має можливість брати участь у голосуваннях, що проводяться адміністрацією сайту (рис. 4.5).

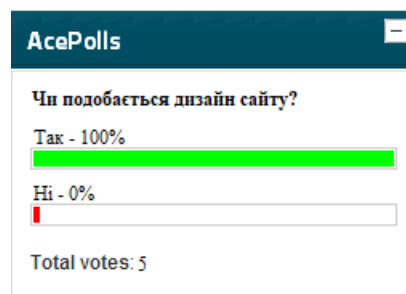


Рисунок 4.5 – Результати голосування

Для голосування за вподобаний відповідь, потрібно натиснути на смужку з його ім'ям. Один користувач, як правило, має можливість голосувати один раз, щоб уникнути накрутки рейтингу.

4.4.2 Тестування за допомогою WAPT

WAPT – це надійний, простий і економічний інструмент для навантажувального і стресового тестування будь-яких вебсайтів, включаючи бізнес-додатки, мобільні сайти, вебпортали і т.д. [22]. Він здатний дати відповіді на наступні питання продуктивності [22]:

- скільки одночасних користувачів може витримати вебсайт;
- скільки часу користувач чекає відповіді сервера на свій запит;
- як часто користувачі побачать помилки та перевантажують сторінки;
- чи витримає сайт стресове навантаження;
- чи зможе він повернутися до нормального функціонування після цього.

Візуальне представлення тестування створеного вебзастосунку приведено нижче (рис. 4.6, рис. 4.7).

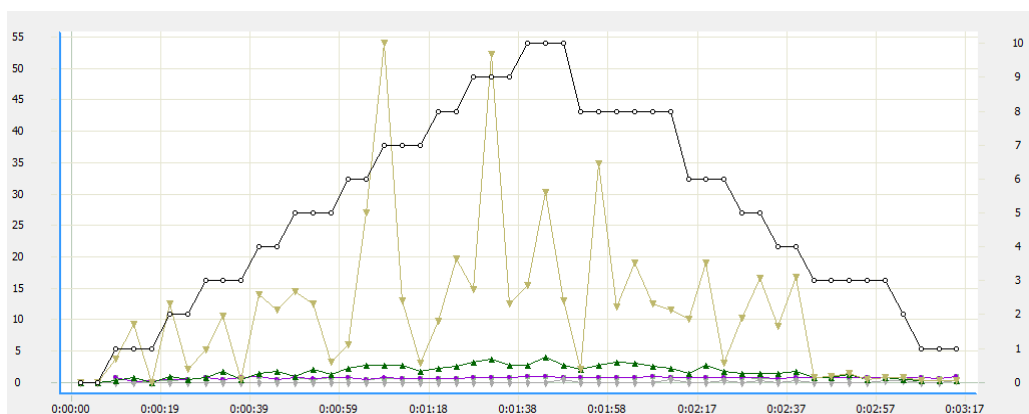


Рисунок 4.6 – Графік продуктивності

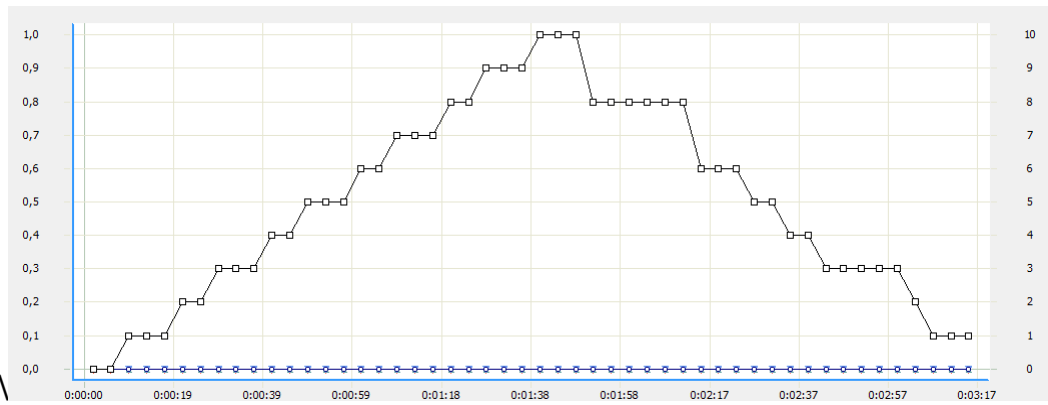


Рисунок 4.7 – Графік ширини каналу

ВИСНОВКИ

У дипломній кваліфікаційній роботі бакалавра виконано аналіз проблеми реалізації вебзастосунку для студентської групи. Так було розроблено інформаційний ресурс для комунікації здобувачів вищої освіти. Для цього було виконано наступні завдання:

- проаналізовано проблемну комунікації студентів;
- досліджено існуючі аналогічні рішення для електронної комунікації здобувачів;
- обрано технологію розробки застосунку, що відрізняється кращими характеристиками від аналогів;
- спроектовано архітектуру вебзастосунку;
- розроблено вебзастосунок;
- проведено тестування.

Розроблена структура вебзастосунку була реалізована із використанням CMS Joomla. Вибір CMS був висвітлений та обґрунтований.

Використали модулі розширення функціональності.

Був розроблений сценарій та проведено тестування створеного вебзастосунку за допомогою WAPT. Отриманий результат свідчить про те, що тестування дало позитивний результат.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Тіверіадська Л. В. Студентська група як елемент педагогічної системи та виховного процесу у ВНЗ [Текст] / Л. В. Тіверіадська // Вісник Придніпровської державної академії будівництва та архітектури. - 2013. - № 6. - С. 46-51.
2. Крисюк Г. Студентська група як чинник формування нормативних стандартів поведінки [Текст] / Г. Крисюк // Збірник наукових праць: філософія, соціологія, психологія. – 2011. - Вип. 16(1). - С. 61-68.
3. Шевчук І. В. Студентська академічна група як об'єкт виховання на початковому етапі навчання у вищих економічних навчальних закладах : автореф. дис. канд. пед. наук: 13.00.07 / І. В. Шевчук; Нац. пед. ун-т ім. М.П.Драгоманова. - К., 2010. - 21 с.
4. Лякішева А. В. Психологічні та соціально-педагогічні особливості функціонування студентської групи / А. В. Лякішева // Педагогіка і психологія. - 2012. - № 4. - С. 69-74.
5. Прадивляний М. Г. Використання особистісно орієнтованого підходу у формуванні готовності до іншомовного професійного спілкування студентів технічних спеціальностей [Текст] / М. Г. Прадивляний // Вісник Вінницького політехнічного інституту. - 2016. - № 4. - С. 115-119.
6. Пономаренко Н. В. Особливості використання автентичних текстів для розвитку навичок міжкультурного спілкування в студентів комп'ютерних спеціальностей [Текст] / Н. В. Пономаренко, Г. О. Неустроева // Науковий вісник Міжнародного гуманітарного університету. Серія : Філологія. - 2018. - Вип. 34(2). - С. 112-114.
7. Можаровська О. Е. Навчання професійно орієнтованого іншомовного спілкування студентів технічних освітніх закладів як педагогічна проблема [Текст] / О. Е. Можаровська // Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми. - 2020. - Вип. 57. - С. 125-130.

8. Тонкіх І. Ю. Особливості функціонування публічних Telegram-каналів як кросмедійної платформи [Текст] / І. Ю. Тонкіх // Держава та регіони. Серія : Соціальні комунікації. - 2021. - № 4. - С. 47-54.
9. Telegram / [Електрон. ресурс] – Режим доступу: <https://telegram.org/?setln=uk>
10. Дзюбіна О. І. Комунікативний аспект соціальних мереж Facebook і Twitter [Текст] / О. І. Дзюбіна // Вісник Дніпропетровського університету імені Альфреда Нобеля. Серія : Філологічні науки. - 2016. - № 2. - С. 218–222.
11. Facebook / [Електрон. ресурс] – Режим доступу: <https://uk-ua.facebook.com/>
12. Петренко С. В. Використання хмарних сервісів G SUITE (GOOGLE APPS) у навчально-виховному процесі [Текст] / С. В. Петренко // Інноватика у вихованні. - 2016. - Вип. 4. - С. 211-220.
13. G Suite / [Електрон. ресурс] – Режим доступу: <https://support.google.com/a/answer/6047848?hl=ua>
14. Костенко О. П. Розробка методу захисту інформації у веббазованих додатках [Текст] / О. П. Костенко, С. М. Маркевич, Ю. О. Величай // Нові технології. - 2013. - № 1-2. - С. 78-81.
15. Канакін А. Е. Розробка елементів рекомендаційної системи на базі вебаналітики [Текст] / А. Е. Канакін, І. С. Скарга-Бандурова, М. Є. Щербакова // Вісник Східноукраїнського національного університету імені Володимира Даля. - 2018. - № 6. - С. 199-204.
16. What is PHP? The PHP Programming Language Meaning Explained / [Electronic resource]. – Access mode: <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/>
17. PHP Introduction / [Electronic resource]. – Access mode: https://www.w3schools.com/php/php_intro.asp
18. North B.M. Joomla! A user's guide [Text] : building a successful Joomla! Powered website / Barrie M. North. - Upper Saddle River, New Jersey[etc.] : Prentice Hall, 2008.

19. Рамел Д. Joomla! для профессионалов [Текст] / Дэн Рамел ; [пер. с англ. и ред. И. В. Берштейна]. - Киев : Вильямс, 2014. - 441 с.

20. Гудзовата О. О. Використання уніфікованої мови візуального моделювання UML (Unified Modeling Language) як інструменту підтримки проєктування інформаційних систем [Текст] / О. О. Гудзовата, В. І. Костирко, І. В. Артищук // Підприємництво і торгівля. - 2019. - Вип. 24. - С. 108-114.

21. Богдан И. Классификация ошибок на UML-диаграммах, возникающих в ходе разработки IT-проектів [Текст] / И. Богдан, А. Задорожний // Технічні науки та технології. - 2018. - № 1. - С. 68-78.

22. WAPT / [Electronic resource]. – Access mode: <https://www.loadtestingtool.com/>

ДОДАТОК А
Технічне завдання

Вступ

Предметом розробки є інформаційний ресурс для комунікації здобувачів академічної групи у вигляді вебзастосунку.

У цьому документі наводиться повний набір вимог до реалізації сайту. Підпис замовника і виконавця на цьому документі підтверджує їх згоду з нижченаведеними фактами та умовами:

- виконавець підготував і розробив цей документ, іменований технічне завдання, який містить перелік вимог до виконуваних робіт;
- замовник згоден з усіма положеннями цього технічного завдання;
- замовник не має права вимагати від виконавця в рамках поточного договору виконання робіт або надання послуг, прямо не описаних в цьому технічному завданні;
- виконавець зобов'язується виконати роботи в обсязі, зазначеному в цьому технічному завданні;
- замовник не має права вимагати від виконавця дотримання будь-яких форматів і стандартів, якщо це не зазначено в цьому Технічному завданні.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу, яке було визначене за темою «Програмна реалізація інформаційного ресурсу для комунікації здобувачів академічної групи» і затверджене наказом № 102 від 27 квітня 2022 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Функціональне призначення: сайт надає умови для обміну інформацією, з підтримкою функції оповіщення користувачів і онлайн обміну повідомленнями.

Експлуатаційне призначення сайту:

- надання інформації про групу;
- надання інформації всередині групи;
- обмін та обговорення інформації всередині групи.

Цільова аудиторія сайту – студенти академічної групи ЗВО, наповненням буде новинний контент, такий як: розклади, лекції, аудіо-відео конспекти, методичні посібники, книги та інша література.

A.3 Основні вимоги до програми, що розробляється

При розробці сайту повинні бути використані переважно світлі стилі.

Основні розділи сайту повинні бути доступні з першої сторінки.

На першій сторінці не повинно бути великого обсягу текстової інформації, а повинно розташовуватися актуальне розкладу занять.

У дизайні сайту не повинні бути присутніми:

- миготливі банери;
- багато зливається тексту.

A.3.1 Вимоги до функціональних характеристик

Головна сторінка сайту повинна містити графічну частину, навігаційне меню сайту, а також контентну область для того, щоб відвідувач сайту з першої сторінки міг отримати інформації про розклад, а також Головне меню. Для зареєстрованих користувачів має бути доступно окреме меню користувача.

Меню сайту першої сторінки має ділитися на наступні розділи:

- головна сторінка;
- вхід на сайт;
- панель управління (доступно пройшли авторизацію адміністраторам і модераторам);
- розклад занять;
- новини (оцінки за минулими іспитами, відомості про консультації викладачів і т. д.);
- наші досягнення (містить матеріали щодо досягнень учасників у галузі спорту та навчання);
- наші заходи (містить фот-відео звіти з спільних заходів);
- бібліотечка (містить навчальну літературу);
- лекціонаріум (конспекти лекцій у текстовому та/або медіа варіантах);
- дозвілля (матеріали розважального змісту: книги, журнали, фільми);
- довідка по сайту.

Меню користувача повинно містити:

- ваш профіль (профіль користувача з відомостями реєстрації);
- створити матеріал (для адміністрації та модераторів).

Крім того, шапка сайту повинна містити функцію підстроювання розміру шрифту і пошуковий рядок. Нижня панель сайту повинна містити модуль відправки повідомлень на е-мейл, а модуль чату повинен бути вбудований між основною панеллю і меню.

Функціонал сайту і панелі управління повинен забезпечувати наступне:

- можливість реєстрації нового користувача;
- можливість відправки електронного повідомлення адміністратором/модератором будь-якому користувачеві;

- залишати коментарі до матеріалів;
- завантажувати файли зареєстрованим користувачам;
- розміщення файлів адміністраторами \ модераторами;
- для отримання доступу до адміністративної частини сайту необхідно вказати певну адресу в рядку браузера і пройти авторизацію;
- підтримка онлайн-чату;
- можливість змінювати шрифт відображення сайту;
- здійснення пошуку по сайту;
- участь зареєстрованих учасників у голосуванні, що організовується адміністрацією;
- можливість залишити коментар до матеріалу.

A.3.2 Вимоги до надійності

Були висунуті наступні вимоги до надійності:

- відмовостійкість;
- час очікування відгуку сервера не більше 8 сек;
- захист паролів алгоритмом MD5.

A.3.3 Умови експлуатації

Пред'являються такі вимоги:

- наявність браузера на клієнтській стороні;
- використання в розробці PHP 5.1 і вище;
- використання CMS Joomla для розробки;
- використання вебсервера Denwer для тестування;
- використання бази даних MySQL в якості сховища даних на серверній стороні.

А.3.4 Вимоги до складу та параметрів технічних засобів

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- IBM-сумісна серверна робоча станція під управлінням встановленої та налагодженої серверної ОС;
- вебсервер: HTTP-сервер, або сервер Microsoft Internet Information або Netscape Enterprise;
- інтерпретатор PHP5;
- MySQL-сервер для керування користувачами та організації доступу до БД.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- IBM-сумісний ПК;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – від 3GB (для безперебійної роботи браузерів);
- підключення до мережі Internet;
- встановлений браузер останньої версії.

А.4 Стадії та етапи розробки

На цьому етапі необхідно сформулювати основну ідею майбутнього сайту.

Визначити мету сайту – музичний інтернет-магазин.

Визначити цільову аудиторію (стать, вік, освіта), тобто для кого буде цей сайт.

З'ясувати, яку інформацію очікують відвідувачі, що допоможе з визначенням контенту (текстовою інформацією) на сайті.

Визначитися з технічною стороною. Важливо враховувати середнє підключення до мережі, а так само програмним забезпеченням.

На цьому ж етапі складається ТЗ (Технічне завдання).

Проектування системи

Проектується система, враховуючи усі потреби замовника.

Проектування інтерфейсу

На цьому етапі проектується інтерфейс сайту. Він повинен бути зручний та зрозумілий у використанні.

Розробка програмного продукту

На цьому етапі відбувається аналіз структури та розгортання вебдодатків.

Програмується вебсайт з урахуванням усіх проектних рішень.

Тестування системи

Після завершення всіх робіт з інтеграції систем і налагодження обладнання інтернет-магазин буде надано на тестування експерту.

Наповнення контентом

Ефективність роботи сайту пов'язана з поняттям контенту, тобто текстової та графічної інформації. На цьому етапі сайт наповнюється контентом за бажанням Замовника.

Оформлення документації та передання замовнику

Після завершення всіх процедур і робіт з тестування сайту і навчання фахівців групи підтримки проєкт буде готовий до відкриття для відвідувачів. На даному етапі оформлюється документація та підписуються необхідні документи про приймання – здачі програмного продукту.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Предметом розробки є інформаційний ресурс для комунікації здобувачів академічної групи у вигляді вебзастосунку.

Б.2 Функціональне призначення

Функціональне призначення: сайт надає умови для обміну інформацією, з підтримкою функції оповіщення користувачів і онлайн обміну повідомленнями.

Експлуатаційне призначення сайту:

- надання інформації про групу;
- надання інформації всередині групи;
- обмін та обговорення інформації всередині групи.

Цільова аудиторія сайту – студенти академічної групи ЗВО, наповненням буде новинний контент, такий як: розклади, лекції, аудіо-відео конспекти, методичні посібники, книги та інша література.

Б.3 Опис логічної структури

Логічна взаємодія найбільш повно представлено на рис. Б.1.

Оброблені дані формують транзакції, що направляються в БД для збереження і формують відгук(у вигляді повідомлення або HTML-сторінки), який відправляється клієнту на відображення успішності\неуспішності скоєних дій, наприклад, входу на сайт або додавання коментаря.

При цьому ніякий модуль не може функціонувати, якщо ідентифікатора користувача, що звертається до нього, немає в базі даних. Таким чином, модулі голосування, коментарів, чату і пошуку будуть недоступні для сторонніх користувачів, однак скрипт входу на сайт, виконаний у вигляді інтегрованого в середу модуля, буде доступний.

Найбільш повно концепція логіки відображена на діаграмі класів, за допомогою функцій, доступних тим чи іншим категоріям користувачів (рис. Б.1).

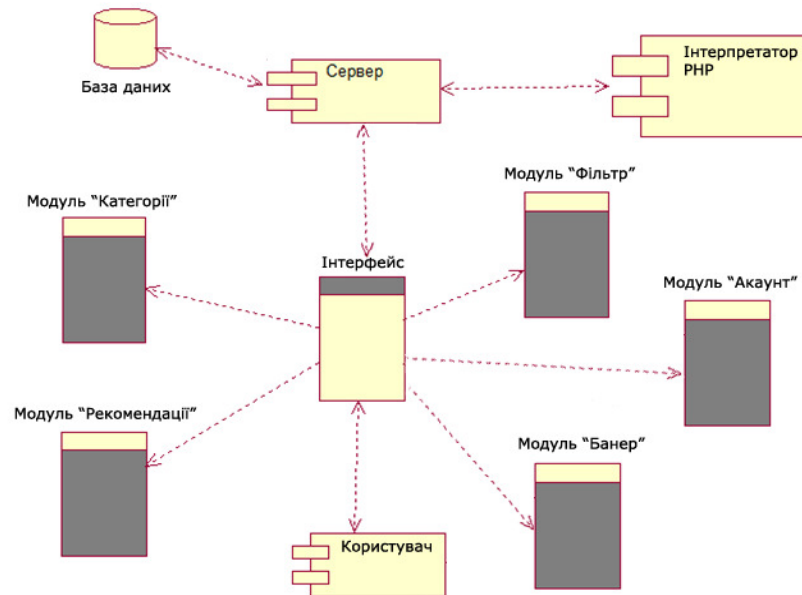


Рисунок Б.1 – Компоненти та зв'язки

Основний поділ функціональності проводиться за категоріями користувачів. Прості користувачі, тобто особи, зареєстровані в системі, мають можливість незворотної взаємодії з модулями: залишення коментаря без можливості редагування, відправка повідомлень в чат без можливості видалення, і т. д.

Особі категорії "Модератор" мають можливість редагувати результати взаємодії з модулями, свої або інших користувачів.

Особі категорії "Адміністратор" мають можливість глибокої взаємодії з системою, тобто додавання\видалення модулів, внесення безпосередньо змін до БД сайту.

Кожен модуль взаємодіє з БД сайту, зберігаючи там свої результати роботи (крім модуля пошуку, який повертає в якості підсумку запиту HTML-сторінку). У свою чергу, інтерпретацією коду модулів займається безпосереднє вебсервер і його розширення для перетворення js, jsp php-скриптів.

Модуль чату передбачає відправку на сервер тексту повідомлення, зберігання логів в БД і відображення їх у своєму функціональному вікні.

Модуль голосування надає інтерфейс у вигляді списку вибору варіантів відповіді на опитування, також зберігає свої дані в БД, відправляючи код вибору на інтерпретацію.

Модуль коментарів по функціональності схожий з модулем чату, прикріплює текстові повідомлення до коментованого матеріалу, підтримує також кілька гілок коментарів, дозволяючи здійснювати повноцінні обговорення.

Модуль пошуку приймає в якості параметрів запиту ключові слова і повертає масив посилань на знайдені матеріали у вигляді HTML-сторінки.

Б.4 Використані технічні засоби

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- IBM-сумісна серверна робоча станція під управлінням встановленої та налагодженої серверної ОС;
- вебсервер: HTTP-сервер, або сервер Microsoft Internet Information або Netscape Enterprise;
- інтерпретатор PHP5;
- MySQL-сервер для керування користувачами та організації доступу до БД.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- IBM-сумісний ПК;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – від 3GB (для безперебійної роботи браузерів);
- підключення до мережі Internet;

- встановлений браузер останньої версії.

Б.5 Виклик та завантаження

Для завантаження сайту і взаємодії з ним досить набрати в адресному рядку браузера «knt123.nuzp». Для входу в панель управління: «knt123.nuzp \administrator».

Б.6 Вхідні та вихідні дані

Програма не має конкретних вхідних даних для обробки, і ключем до доступу до роботи з нею є комбінація з логіна і пароля користувача\адміністратора. Спосіб кодування пароля-MD5.

ДОДАТОК В
Текст програми

```

static function &getInstance($dbtype, $dbhost="", $dbuser="", $dbpass="",
$dbname="", $dbprefix=")
{
    static $instance;
    static $hash_current;

    // Forcibly create new instance if dbhost parameter exists and
the requested connection has
    // different parameters than the existing one
    if(!empty($dbhost))
    {
        // Create a parameter hash of the current connection (if
any)
        $hash_requested =
$dbtype.$dbhost.$dbuser.$dbpass.$dbname.$dbprefix;
        if( ($hash_current != $hash_requested) ||
empty($hash_current) )
        {
            $hash_current = $hash_requested;
            if(is_object($instance)) $instance->disconnect();
            $instance = null;
        }
    }

    // Create a new instance if it doesn't exist
    if( !is_object($instance) || !(empty($host)) )
    {
        $instance = new ABIDatabase($dbtype, $dbhost,
$dbuser, $dbpass, $dbname, $dbprefix);
    }
}

```

```

        return $instance;
    }

/**
 * Tries to connect to the database
 * @return bool True on success
 */
function connect()
{
    // For the shake of Windows users, I'll try persistent connections
    first, or reusing an existing connection
    // Ref: http://dev.mysql.com/doc/refman/5.0/en/can-not-
    connect-to-server.html#can-not-connect-to-server-on-windows
    switch($this->dbtype)
    {
        case 'mysql':
            $this->handle = false;
            if( function_exists('mysql_pconnect') )
                $this->handle = @mysql_pconnect($this-
                >dbhost, $this->dbuser, $this->dbpass);
            if($this->handle === false)
                $this->handle = @mysql_connect($this-
                >dbhost, $this->dbuser, $this->dbpass, false);
            break;

        case 'mysqli':
            // Unlike mysql_connect(), mysqli_connect() takes
            the port and socket

```

```

// as separate arguments. Therefore, we have to
extract them from the

// host string.
$host = $this->dbhost;
$port = NULL;
$socket = NULL;
$targetSlot = substr( strstr( $host, ":" ), 1 );
if (!empty( $targetSlot )) {
    // Get the port number or socket name
    if (is_numeric( $targetSlot )) {
        $port = $targetSlot;
    } else {
        $socket = $targetSlot;
    }

    // Extract the host name only
    $host = substr( $host, 0, strlen( $host ) -
(strlen( $targetSlot ) + 1) );

    // This will take care of the following
notation: ":3306"

    if($host == "") {
        $host = 'localhost';
    }
}

$this->handle = @mysqli_connect($host, $this-
>dbuser, $this->dbpass, $this->dbname, $port, $socket);
break;
}

```

```

if($this->handle === false)
{
    $this->error = 'Could not connect to MySQL server';
    return false;
}

// MySQL Connection Parameters
switch($this->dbtype)
{
    case 'mysql':
        $result = @mysql_query("SET NAMES 'utf8'",
$this->handle);

        break;

    case 'mysqli':
        $result = @mysqli_query($this->handle, "SET
NAMES 'utf8'");

        break;
}

// Select database
switch($this->dbtype)
{
    case 'mysql':
        $result = @mysql_select_db($this->dbname, $this-
>handle);

        break;

    case 'mysqli':

```

```

        $result = @mysql_select_db($this->handle, $this-
>dbname);

        break;
    }

    if(!$result)
    {
        // Database does not exist. Let's try create it.
        switch($this->dbtype)
        {
            case 'mysql':
                $result = @mysql_query('CREATE
DATABASE `'.$this->dbname.` DEFAULT COLLATE utf8_general_ci');
                break;

            case 'mysqli':
                $result = @mysqli_query($this->handle,
'CREATE DATABASE `'.$this->dbname.` DEFAULT COLLATE
utf8_general_ci');
                break;
        }
        // Check if the database was created
        if(!$result)
        {
            $this->error = 'Could not select the '.$this-
>dbname.' database.';

            return false;
        }
        // Now that it's created, select it
        switch($this->dbtype)

```



```

        {
            case 'mysql':
                $result = @mysql_select_db($this->dbname,
$this->handle);

                break;

            case 'mysqli':
                $result = @mysqli_select_db($this->handle,
$this->dbname);

                break;
        }
        // Check if the database was selected
        if(!$result)
        {
            $this->error = 'Could not select the '.$this-
>dbname.' database.';

            return false;
        }
    }

    switch($this->dbtype)
    {
        case 'mysql':
            @mysql_query("SET NAMES 'utf8'", $this-
>handle);

            break;

        case 'mysqli':
            @mysqli_query($this->handle,"SET NAMES
'utf8");

```

```

        break;
    }

    // If MySQL 5.x is used, switch to MySQL 4.0 compatibility
mode to work around strict mode issues
    if ( strpos( $this->getVersion(), '5' ) === 0 ) {
        switch($this->dbtype)
        {
            case 'mysql':
                @mysql_query("SET sql_mode =
'MYSQL40'", $this->handle);
                break;

            case 'mysqli':
                @mysqli_query($this->handle,"SET
sql_mode = 'MYSQL40'");
                break;
        }
    }

    return true;
}

/**
 * Disconnects from the database server
 */
function disconnect()
{
    switch($this->dbtype)
    {

```

```

case 'mysql':
    @mysql_close($this->handle);
    break;

case 'mysqli':
    @mysqli_close($this->handle);
    break;
    }
}

/**
 * Sets the query to execute, replacing the #__ placeholder with the
database prefix, if any
 * @param $sql string An SQL query
 * @param $onlyFirstInstance bool Only replace the first instance
 */
function setQuery($sql, $onlyFirstInstance = true)
{
    if( !empty($this->dbprefix) )
    {
        // Substitute the db prefix
        if(!$onlyFirstInstance)
        {
            $sql = str_replace('#__', $this->dbprefix, $sql);
        }
        else
        {
            $pos = strpos($sql, '#__');
            // # FIX 2.3.1 -- It is possible that the #__ doesn't
exist. Stupid me!

```

```

        if($pos !== false)
            $sql = substr_replace( $sql, $this->dbprefix,
$pos, 3);
    }
}

$this->sql = $sql;
}

/**
 * Runs a query against the server.
 * @param $sql string If set, this is the query to run, otherwise a call
to setQuery() must have been preceded.
 * @return unknown_type
 */
function query($sql = null)
{
    // Reset error statuses
    $this->errno = 0;
    $this->error = "";

    // Should we set a new query? Allows for tight coding :)
    if(!is_null($sql))
    {
        $this->setQuery($sql);
    }

    // Catch empty SQL statements
    if(empty($this->sql))
    {

```

```

        $this->sql = "";
        return true;
    }

    // If there is no active connection, try to connect
    if(empty($this->handle))
    {
        if(!$this->connect())
        {
            // If conenction failed, exit
            return false;
        }
    }

    switch($this->dbtype)
    {
        case 'mysql':
            $result = @mysql_query($this->sql, $this-
>handle);

            break;

        case 'mysqli':
            $result = @mysqli_query($this->handle, $this-
>sql);

            break;
    }

    if($result === FALSE)
    {
        switch($this->dbtype)
        {

```

```

        case 'mysql':
            $this->error = 'MySQL query failed with
error '.mysql_erro($this->handle).' (. mysql_error($this->handle) .'). The query
was:'. $this->sql;

            $this->errno = mysql_errno($this->handle);
            break;

        case 'mysqli':
            $this->error = 'MySQL query failed with
error '.mysqli_erro($this->handle).' (. mysqli_error($this->handle) .'). The query
was:'. $this->sql;

            $this->errno = mysqli_errno($this->handle);
            break;
    }
    return false;
}
else
{
    $this->resource = $result;
    $this->sql = "";
    return true;
}
}

/**
 * Returns a query result as an associative array
 * @param $sql string The SQL query to execute (optional)
 * @return array The result, or an empty array if nothing was fetched
 */
function getAssocArray($sql = null)

```

```

{
    $output = array();
    $result = $this->query($sql);
    if($result == true)
    {
        switch($this->dbtype)
        {
            case 'mysql':
                $numrows = @mysql_num_rows($this-
>resource);
                break;

            case 'mysqli':
                $numrows = @mysqli_num_rows($this-
>resource);
                break;
        }
        if($numrows > 0)
        {
            switch($this->dbtype)
            {
                case 'mysql':
                    while ($row =
@mysql_fetch_assoc($this->resource))
                    {
                        $output[] = $row;
                    }
                    break;

                case 'mysqli':

```

```

                                while ($row =
@mysqli_fetch_assoc($this->resource))
                                {
                                    $output[] = $row;
                                }
                                break;
                            }
                        }
                    }
                return $output;
            }

/**
 * Escapes a value string to be used in a SQL query
 * @param $string string The string to escape
 * @return string The escaped string
 */
function escape($string)
{
    switch($this->dbtype)
    {
        default:
        case 'mysql':
            return @mysql_real_escape_string($string, $this-
>handle);
            break;

        case 'mysqli':
            return @mysqli_real_escape_string($this->handle,
$string);

```



```
                break;
            }

        }

function getError()
{
    return $this->error;
}

function reset()
{
    $this->errno = 0;
    $this->error = null;
}

function getVersion()
{
    switch($this->dbtype)
    {
        case 'mysql':
            return mysql_get_server_info($this->handle);
            break;

        case 'mysqli':
            return mysqli_get_server_info($this->handle);
            break;
    }
}
}
```

ДОДАТОК Д
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»

○

**Програмна реалізація інформаційного
ресурсу для комунікації здобувачів
академічної групи**

Виконав	Павленю В.С. (КНТ-218)
Керівник	д.т.н., професор Субботін С.О.
Рецензент	к.т.н., доцент Гофман Є.О.

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

○

Об'єкт дослідження – процес програмної реалізації інформаційного ресурсу для комунікації здобувачів академічної групи

Предмет дослідження – методи програмної реалізації інформаційного ресурсу для комунікації здобувачів академічної групи.

Метою роботи є розроблення інформаційного ресурсу для комунікації здобувачів академічної групи.

Рисунок Д.2 – Слайд 2

Постановка завдання

Проаналізувавши усі найбільш поширені і можливі варіанти реалізації проекту було прийнято рішення про організацію незалежного сайту на окремому хостингу з метою тонкого налаштування його вмісту і доступу до нього. На даний момент не представляється можливим здійснювати оповіщення по телефону, відповідно, потрібна альтернативна заміна, що в умовах сучасного поширення інтернет-технологій формулюється в необхідності створення сайту.

Переваги Користувача	Підтримувані функції
Прискорення процесу обміну інформацією	Функція оповіщення про оновлення
Прискорення процесу пошуку інформації	Функція пошуку по сайту
Нові користувачі можуть швидко включитися в роботу	Детальна довідка допоможе в отриманні інформації по роботі з системою

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Чат/Канал Telegram	Спільнота Facebook	Засоби Google Workspace	Інформаційний веб-ресурс
Доступність	Безкоштовно	Безкоштовно	Безкоштовно (але додатковий функціонал – платний)	Безкоштовно
Організація чатів/коментарів	Так	Так	Так (додаткові засоби)	Так
Збереження матеріалів	Так	Через посередні сервіси	Так	Так
Рівень персоналізації	Середній	Низький	Низький	Високий

Рисунок Д.4 – Слайд 4

Порівняння мов програмування

Критерій	Python	PHP
Простота вивчення	++	++
Матеріали та спільнота	++	++
Швидкість та простота розгортання застосунків	+	++
Програмні надбудови для розробки	+	++
Підтримка та масштабування застосунків	+	++

Рисунок Д.5 – Слайд 5

Порівняння CMS

	WordPress	Drupal	Joomla!
Простота розгортання	++	+	++
Можливість персоналізації застосунку адміністратора для	++	+	++
Вимоги до хостингу	±	+	++
Підтримка розширень	+	+	++

Рисунок Д.6 – Слайд 6



Рисунок Д.7 – Слайд 7



Рисунок Д.8 – Слайд 8

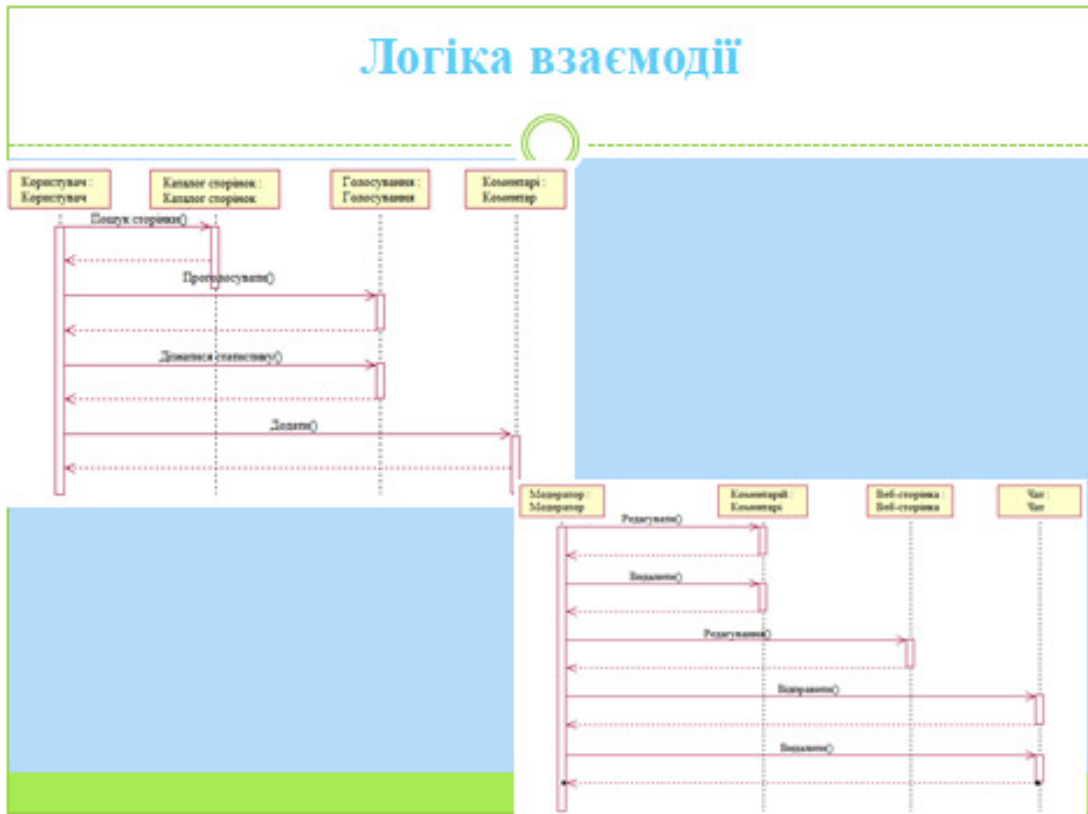


Рисунок Д.9 – Слайд 9

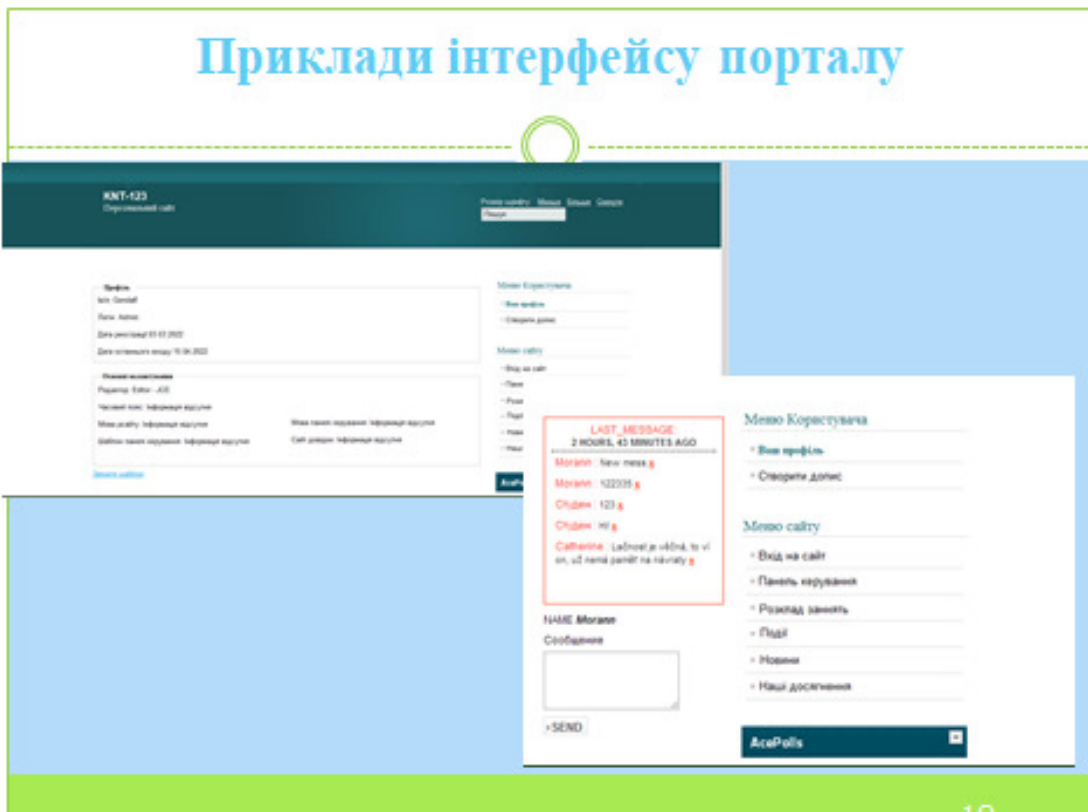


Рисунок Д.10 – Слайд 10



Рисунок Д.11 – Слайд 11

Висновки

У випускній дипломній роботі бакалавра виконано аналіз проблеми реалізації веб-застосунку для студентської групи. Так було розроблено інформаційний ресурс для комунікації здобувачів вищої освіти. Для цього було виконано наступні завдання:

- проаналізовано проблемну комунікації студентів;
- досліджено існуючі аналогічні рішення для електронної комунікації здобувачів;
- обрано технологію розробки застосунку, що відрізняється кращими характеристиками від аналогів;
- спроектовано архітектуру веб-застосунку;
- розроблено веб-застосунок;
- проведено тестування.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

Рисунок Д.12 – Слайд 12