

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ФАЙЛООБМІННИКА
SOFTWARE IMPLEMENTATION OF THE FILE SHARING SYSTEM

Виконав: студент(ка) 4 курсу, групи КНТ-119сп
Спеціальності 121Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Котов О.О.

(прізвище та ініціали)

Керівник Коцур М.І.

(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 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(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 ПРОЕКТ (РОБОТУ) СТУДЕНТА(КИ)

Котова Олександра Олександровича
(прізвище, ім'я, по батькові)

- Тема проекту (роботи) Програмна реалізація системи файлообмінника
Software Implementation of the File Sharing System
керівник проекту (роботи) Коцур Михайло Ігорович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від “ 27 ” квітня 2022 року № 102
- Строк подання студентом проекту (роботи) 07 червня 2022 року
- Вихідні дані до проекту (роботи) рекомендована література, технічне завдання
- 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.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Коцур М.І., доцент		
Нормоконтролер	Белова А.В., асистент		

7. Дата видачі завдання “ 01 ” лютого 2022року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (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	2 тиждень	Розділ 2
	розробки		
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	5 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	6 тиждень	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

Котов О.О.
(підпис) (прізвище та ініціали)

Керівник проекту (роботи)

Коцур М.І.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 83 с., 46 рис., 3 табл., 4 дод., 25 джерел.

ФАЙЛООБМІННИК, ФАЙЛОВИЙ ХОСТІНГ, ХМАРНІ ТЕХНОЛОГІЇ, ВІДДАЛЕНЕ ЗБЕРІГАННЯ, РОЗРОБКА, ПРОГРАМУВАННЯ, ВЕБ-ЗАСТОСУНОК.

Об'єкт дослідження – процес програмної реалізації файлообміннику.

Предмет дослідження – методи реалізації файлообміннику.

Метою роботи є розроблення системи файлообміннику для віддаленого зберігання та поширення цифрових даних за допомогою мережі Інтернет.

Матеріали, методи та технічні системи файлообміннику.

Висновки. Розроблено систему файлообміннику для віддаленого зберігання та поширення цифрових даних за допомогою мережі Інтернет.

Галузь використання – користувачі мережі Інтернет.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 83pages, 46 figures, 3 tables, 4 appendixes, 25 sources.

FILE SHARING, FILE HOSTING, CLOUD TECHNOLOGIES, REMOTE STORAGE, DEVELOPMENT, PROGRAMMING, WEB APPLICATION.

The object of research is the process of software implementation of file sharing.

The subject of research is methods of implementing file sharing.

The aim of the work is to develop a file sharing system for remote storage and distribution of digital data via the Internet.

Materials, methods, and technical tools: Object-Oriented Programming, PHP programming language, WordPress version control system, MySQL database management system.

Results. Software has been developed in the form of a file sharing system for remote storage and distribution of digital data via the Internet.

The scientific novelty of the work is the development of a cross-platform file sharing system.

Conclusions. A file sharing system has been developed for remote storage and distribution of digital data via the Internet.

Field of usage: users of the Internet network.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	8
Вступ	9
1 Аналіз предметної області	10
1.1 Файлообмінник або файловий хостинг	10
1.2 Зовнішнє або хмарне збереження файлів	11
1.3 Огляд існуючих рішень	14
1.4 Постановка завдання	19
1.3 Висновки за розділом 1	22
2 Розробка архітектури програми	23
2.1 Вибір інструментарію розробки	23
2.2 Структурна схема розробки	29
2.3 Висновки за розділом 2	30
3 Основні рішення щодо реалізації компонентів системи	31
3.1 Проектування моделей системи	31
3.2 Моделювання системи з точки зору прецедентів	34
3.3 Фізичне проектування	35
3.4 Проектування інтерфейсу	39
3.5 Висновки за розділом 3	43
4 Експлуатація, тестування та експериментальне дослідження програми	44
4.1 Опис застосунку	44
4.2 Умови виконання програми	44
4.3 Інструкція по експлуатації програми	45
4.4 Методика та результати тестування	49
Висновки	53
Перелік джерел посилання	54
Додаток А Технічне завдання	58
Додаток Б Опис програми	63
Додаток В Текст програми	68

Додаток Д Слайди презентації	76
------------------------------------	----

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CMS – системи управління контентом;
HDD – жорсткий диски;
SSD – твердо тільні носії;
URL – uniform resource locator;
UML – unified modeling language;
БД – база даних;
ОЗУ – оперативна пам'ять;
ОС – операційна система;
ПЗ – програмний засіб;
ПК – персональний комп'ютер;
СКБД – системи керування базами даних.

ВСТУП

У наші часи все більш популярною та затребуваною стає розробка веб-застосунків, як наслідок значного збільшення впливу та поширення мережі Інтернет. Рядовий користувач вимагає все більшого різноманіття та поширення функціоналу існуючих веб-сервісів для задоволення своїх постійно зростаючих потреб.

Звідси випливає мета даної роботи – спроектувати і розробити веб-застосунок.

Об'єктом дослідження даного розрахункового завдання є проектування і розробка веб-застосунку «Файлообмінник», що дозволить користувачу зберігати потрібні йому дані на веб-сервері, маючи до них доступ з будь-якого девайсу, що має підключення до мережі Інтернет.

Також під час виконання роботи розглянуто засоби для проектування та розробки програмного забезпечення.

Актуальність розробки такого файлообміннику пояснюється постійним зростанням попиту на послуги файлового хостингу для більшості підприємств, особливо для тих, у кого є віддалені працівники. Всім підприємствам і організаціям потрібен спосіб безпечного розміщення файлів і обміну ними з співробітниками і клієнтами. Сайти розміщення файлів гарантують, що робота і спільна робота в режимі реального часу як і раніше можуть здійснюватися на відстані.

Існують сотні доступних послуг хмарного файлообміну з різними функціями і тарифними планами. Їх послуги можуть працювати для будь-якої людини, від малого бізнесу до великих корпорацій. Проте існуючі рішення мають свої переваги та недоліки, тому було вирішено розробити новий продукт.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Перший розділ роботи присвячено висвітленню питань віддаленого файлового хостингу та хмарного зберігання інформації. В розділі буде розглянуто основні питання предметної області, досліджено існуючі аналогічні рішення та визначено функціональні вимоги, межі розробки та системи для подальшої реалізації проєкту.

1.1 Файлообмінник або файловий хостинг

Як відомо більшості користувачів та споживачів, файли можна зберігати на різного роду носіях. Це жорсткі диски (HDD) і твердо тільні носії (SSD), флешки, CD або DVD диски, карти пам'яті та інші пристрої. Для передачі тих самих файлів через мережу Інтернет можна використовувати електронну пошту, Skype, соціальні мережі, месенджери [1]-[3].

Але якщо файл занадто великий – його неможливо відправити поштою, в Skype або прикріпити до повідомлення в соціальній мережі. На допомогу приходять файлообмінники [1]-[3].

Файлообмінник – це спеціальний веб-сервіс, сайт, який дозволяє завантажувати файли на віддалені сервера і отримувати зворотні посилання на них. Грубо кажучи, це сховище даних, в якому можна зберігати свої файли [1]-[3].

В Інтернеті розвелось дуже багато файлообмінників. Найпопулярніші з них: Deposit Files, Letitbit, RapidShare, RGhost, Turbobit, Uploading та інші. На різних сайтах для скачування програм, ігор, музики або фільмів ви зустрічали схожі посилання [1]-[3].

Розглянемо як користуватися файлообмінниками. Всі подібні сервіси для зберігання інформації працюють за одним принципом. Користувач просто обирає потрібний файл, натискає кнопку «Завантажити» і отримує зворотне посилання [1]-[3].

Зворотне посилання – це уніфікований локатор ресурсів (URL), за яким можна завантажити файл. Отримавши посилання, користувач може використовувати його в своїх цілях. Наприклад розіслати друзям або залишити на тематичному форумі [1]-[3].

Труднощів виникнути не повинно, тому що відразу кидається в очі величезна кнопка «Завантажити» і впорається може навіть новачок. Єдине – деякі файлообмінники перед завантаженням вимагають реєстрації [1]-[3].

Переваги файлообмінників. Кожен файлообмінник має свої особливості. Наприклад максимальний обсяг завантажуваних файлів або максимальна швидкість на скачування. Розглянемо переваги файлообмінників в загальному [1]-[3].

Вони дозволяють без проблем пересилати великі файли.

Можна використовувати для економії місця на вінчестері.

Всього за кілька секунд можна отримати посилання на потрібні файли.

Недоліки файлообмінників. Не дивлячись на значні плюси, недоліки у файлообмінників теж є.

Велика кількість реклами. Саме так заробляють власники подібних ресурсів [1]-[3].

Обмеження на швидкість скачування.

Обмеження на обсяг завантажуваних даних.

Через деякий час (тиждень, місяць, рік) користувацькі файли будуть видалені.

Туди можна вільно завантажити віруси. Слід бути більш обережним [1]-[3].

1.2 Зовнішнє або хмарне збереження файлів

Термін «off-site backup» може трактуватися по-різному, проте в загальноприйнятому сенсі під ним мається на увазі розміщення копій

резервних даних на носії (стрічкові або дискові накопичувачі) в місцезнаходження, яке знаходиться поза офісних кордонів [4]-[6]. Цю процедуру можна позначити як план аварійного відновлення даних. Off-site backup є процедурою, яка орієнтована на зберігання даних в чистому вигляді. Тим часом бекап в хмарі зазвичай має на увазі додавання до базової моделі зберігання – «зберігайте свої дані в іншому місці» – додаткових сервісів. Ці сервіси призначені для спрощеного здійснення покупок послуг (вибір тарифів, сховищ, рівня захисту і повернення даних), а також власне для резервного копіювання і відновлення даних у разі необхідності. Таким чином, резервування копій даних в хмарі слід розглядати як сервіс [4]-[6].

Між постачальниками сервісів для зберігання бекапів в хмарі і провайдерами типу off-site backup, які зберігають резервні копії в «чистому» вигляді, може бути багато спільного: і ті, і інші націлені на те, щоб полегшити зберігання критично важливих даних і систем, підключають однакові рівні безпеки як для транзитних даних, так і для даних в місці їх знаходження протягом усього циклу зберігання [4]-[6]. Останнє потрібно для того, щоб дані зберігалися і переміщалися відповідно до певних вимог регуляторів. Відповідність тієї чи іншої категорії вказує на рівень резервування інфраструктури, фізичної безпеки і надійності [4]-[6].

Однак, як вже говорилося, хмарні провайдери не просто зберігають копії даних, а здійснюють сервісне обслуговування. До цього потрібно додати ефективність, яка і «піднесла» хмара так високо. По-перше, оскільки хмарні послуги є підписними, вони оплачуються за спрощеною схемою [4]-[6]. По-друге, управління ними здійснюється за допомогою простих веб-інтерфейсів і, по-третє, провайдери гарантують технічну підтримку в режимі реального часу [4]-[6]. Більшість постачальників настільки вдосконалили свої послуги, що відправка резервних копій в хмарне сховище і їх зберігання порівнялися по простоті з традиційним віддаленим бекапом, однак ці послуги також можуть включати програмне забезпечення (ПЗ), допомога фахівців, які допоможуть вирішити потенційно можливі складнощі з резервним

копіюванням і відновленням даних, цільової інфраструктури, вони також можуть пропонувати преміальні сервіси [4]-[6].

Отже, що ж краще вибрати для резервного зберігання: хмарний бекап або зберігання у неоплачених провайдерів. Нижче пропонується розглянути чотири аспекти, які допоможуть обрати стратегію зберігання [4]-[6].

Які конкретно вимоги пред'являються для збереження резервних даних. Організація може мати певні потреби. Наприклад, вона застосовує до даних, що знаходяться в стані спокою, 448-бітове шифрування Blowfish [4]-[6]. Це той самий окремий випадок, коли не всі хмарні провайдери зможуть піти вам на зустріч, оскільки більшість з них пропонують послуги з певними алгоритмами шифрування, але Blowfish не належить до їх числа. В такому випадку краще розраховувати на постачальника, які дозволять забезпечити належний рівень шифрування [4]-[6].

Чи потрібна вашій організації допомога сторонніх фахівців. Постачальники послуг із зовнішнього зберігання можуть розглядатися в якості IaaS-вендорів, оскільки вони надають сховище за межами вашої організації. Однак якщо для створення і налаштування резервних копій даних або відновлення потрібні спеціальні навички, то в цьому випадку краще скористатися послугами SaaS-провайдерів [4]-[6].

Яка модель ціноутворення підходить вашій організації. Це питання можна задати по-іншому: капітальні витрати або операційні витрати. Вибір хмарних провайдерів має сенс в тому випадку, якщо ви прагнете до помісячної оплати за вигідною ціною [4]-[6]. Не виключено, що для зберігання копій доцільніше придбати обладнання і розмістити його в орендованому просторі. Вибір того чи іншого варіанту лягає на плечі фінансової служби [4]-[6].

Хмарний бекап або off-site backup: який з них забезпечить найкращий захист даних. Чи можете ви створити краще середовище захисту даних самостійно або вам краще застосувати ту, яка створена постачальником хмарних послуг [4]-[6]. Якщо вимоги досить прості, хмарні провайдери

напевно захистять ваші дані стандартними, але надійними методами шифрування [4]-[6].

Однозначної відповіді на те, що краще, Хмарний бекап або off-site backup, немає. Потрібно зважити всі переваги обох підходів і витрати, і потім визначити, наскільки кожен з них задовольняє потреби вашої організації [4]-[6]. Тим не менш, хмарні провайдери постійно розширюють коридор можливостей для зберігання і відновлення резервних даних, залишаючи все менше місця для провайдерів, які займаються зовнішнім бекапом [4]-[6].

1.3 Огляд існуючих рішень

ПЗ типу файлообміннику розробляється з метою обміну різними типами файлів, розмір яких не дозволяє їх відправляти за допомогою, наприклад електронної пошти чи повідомлень у соціальних мережах, або ж занадто ускладнює цей процес. Передбачується, що завдяки програмному продукту процес обміну файлами можна буде значно полегшити та пришвидшити. Також такий ресурс надає свого роду можливість хмарного збереження даних та бекапів.

Для розробки конкурентоспроможного застосунку слід на початку проаналізувати існуючі аналогічні рішення та виявити їх переваги або недоліки.

1.3.1 Сервіс «Dropbox»

Dropbox (рис. 1.1). Даний проект надає можливість безкоштовно зберігати файли певного обсягу (2 Гб) та можливість придбати більший розмір онлайн-сховища (до 100 Гб) [7]-[9]. Також сервіс надає можливість синхронізації файлів між персональними комп'ютерами (ПК) та мобільними девайсами. Присутня реферальна програма [7]-[9].

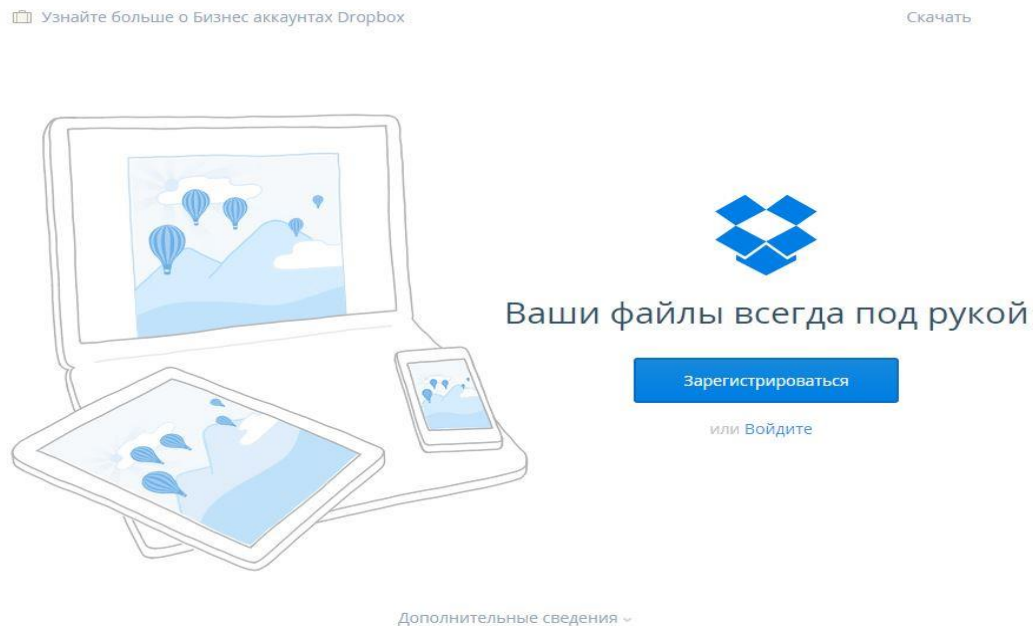


Рисунок 1.1 – Головна сторінка сервісу Dropbox [7]

1.3.2 Сервіс «Google Drive»

GoogleDrive (рис. 1.2). Даний сервіс аналогічний попередньому, але на відміну від нього має синхронізацію з GoogleDocs. Також має значно більший (15 Гб) безкоштовний об'єм і дешевше розширення [10]-[12].

Після запуску сервісу Політика конфіденційності Google Drive піддалася різкій критиці з боку деяких представників засобів масової інформації [10]-[12]. У Google є єдиний набір умов надання послуг і угод про політику конфіденційності, які охоплюють всі його послуги, а це означає, що формулювання в угодах надають компанії широкі права на відтворення, використання і створення похідних творів з контенту, що зберігається на Google Диску [10]-[12]. Хоча політика також підтверджує, що користувачі зберігають права на інтелектуальну власність, захисники конфіденційності висловили стурбованість тим, що Ліцензії надають Google право використовувати інформацію та дані для налаштування реклами та інших послуг, що надаються Google [10]-[12]. Навпаки, інші представники ЗМІ зазначили, що угоди були не гіршими, ніж у конкуруючих сервісів хмарного зберігання, але що конкуренти використовують в угодах «більш хитру мову»,

а також заявили, що Google потрібні права для того, щоб «переміщати файли на своїх серверах, кешувати ваші дані, або зробити мініатюри зображень» [10]-[12].

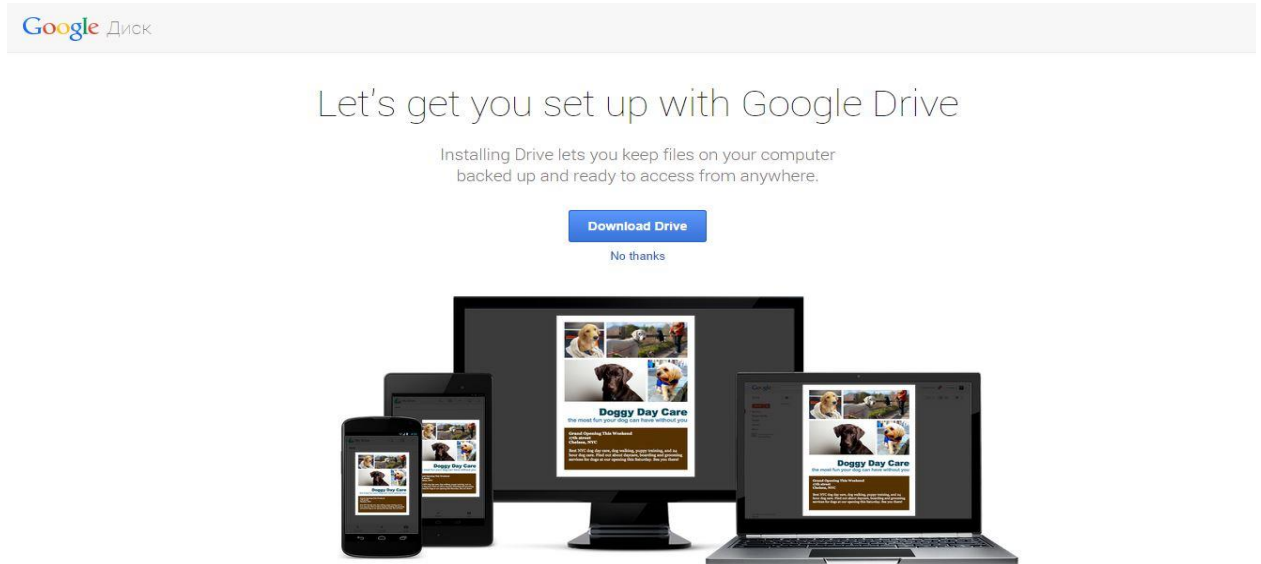


Рисунок 1.2 – Приклад головної сторінки Google Drive [12]

1.3.3 Сервіс «Fex.net»

Fex.net – найбільший український файлообмінник. Був заснований в 2016 року, як часткова заміна сервісу «Ex.ua» [13], [14]. Всі сервери сервісу були розташовані на території України. Сервіс підтримує завантаження файлів до 50 гігабайт. За даними Alexa.com, станом на червень 2016 року Fex.net знаходиться на 13-му за відвідуваністю місці серед українських інтернет-користувачів [13], [14]. Трафік сервісу становить близько 15% всього трафіку зони UA-IX. В середині листопада 2016 року на офіційному сайті Ex.ua з'явилася новина про те, що файлообмінник припиняє роботу, у зв'язку прийняттям нового закону про держпідтримку українського кіно [13], [14]. Всіх користувачів попросили вилучити файли зі своїх архівів до 31.12.2016. А вже на початку 2017 сервіс Fex.net запрацював в повній мірі. При звичайному тарифному пакеті файли зберігаються 7 днів [13], [14] (рис. 1.3).

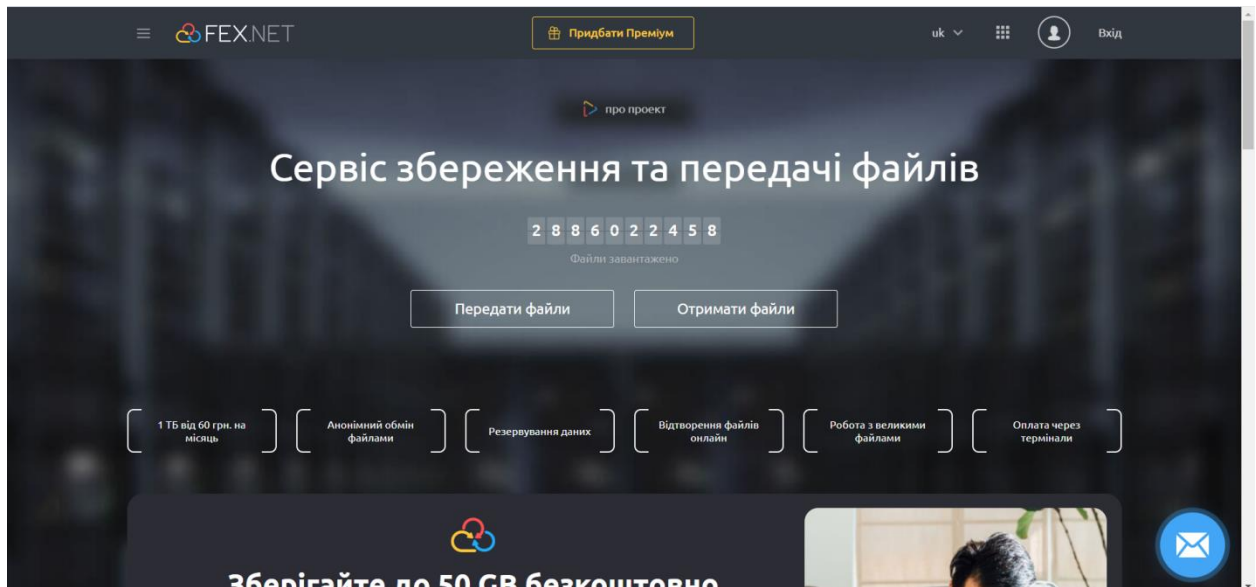


Рисунок 1.3 – Головна сторінка сервісу «Fex.net»

1.3.4 Порівняння існуючих аналогів

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

- безкоштовний функціонал – загальна оцінка сукупного функціоналу (сукупність вільного місця та можливість опрацювання файлів, що зберігаються), що надається користувачам на вільній основі;
- розподіл приватного та загального сховища – можливість роздільного збереження файлів для вільного розповсюдження та особистого зберігання;
- обмеження за форматом файлів – чи накладає сервіс обмеження на збереження файлів певного формату;
- опис та коментування матеріалів – чи є можливість залишити власні чи сторонні опис та коментарі до файлів.

З результатів порівняння можна зробити висновок, що в більшості випадків платформи достатньо обмежені за вільними профілями користування, тобто обмеження на вільний простір для завантаження файлів або час їх зберігання пов'язаний із додатковою оплатою цього функціоналу.

Також Dropbox та Google Drive розглядають зберігання файлів в рамках персональних просторів користувачів і при бажанні розповсюджувати файли необхідно тонко налаштовувати формат їх розповсюдження через налаштування доступу. Сервіс же Fex.net в цьому питанні більш гнучкий, адже запозичив багато від свого попередника – Ex.ua. Більш того, цей ресурс не має обмежень за форматами файлів, на відміну від Dropbox та Google Drive де зустрічаються обмеження на виконувани файли, певні формати архівів або веб-сторінки, наприклад. Проте усі існуючі ресурси мають додаткову проблему із коментуванням або описом файлів – десь цей механізм можна налаштувати виключно для певних форматів файлів, а десь його реалізація недостатньо комфортна.

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій порівняння	Dropbox	Google Drive	Fex.net	Застосунок, що розробляється
Безкоштовний функціонал	Істотні обмеження за об'ємом пам'яті	Обмеження за об'ємом пам'яті	Обмеження за часом зберігання	Відсутність обмежень
Розподіл приватного та загального сховища	Окремо регулюється через налаштування прав доступу	Окремо регулюється через налаштування прав доступу	Організовані й розподіл	Організовані й розподіл
Обмеження за форматом файлів	Присутні	Присутні	Відсутні	Відсутні
Опис та коментування матеріалів	Окремо регулюється через налаштування	Окремо регулюється через налаштування	Можливо	Можливо

Саме тому, можна зробити висновок, що враховуючи велику кількість існуючих файлообмінників із різним функціоналом та можливостями, вони

усі мають низку недоліків та незручностей. Тому актуальною задачею залишається розробка нового файлообміннику із відсутністю обмежень на вільне місце для зберігання, а також на формат файлів. Також новий сервіс буде відрізнятися розподілом загального сховища та приватного для більш комфортного розподілу файлів на обмін та виключно зберігання – такий підхід також дозволить налаштувати комфортну систему опису та коментування вільно розповсюджуваних файлів.

1.4 Постановка завдання

Метою проекту є реалізація файлообміннику, що дозволить зберігати на сервері файли у приватній та загальнодоступній формі, ділитися ними з іншими користувачами та отримувати посилання на завантажений файл. Також має бути передбачена можливість платного розширення місця на сервері, доступного для зареєстрованого користувача.

1.4.1 Межі проєкту

Файлообмінник – це віртуальний хостинг для збереження користувацьких файлів з метою мати до них доступ з будь-якого девайсу, що має доступ до мережі Інтернет.

Проект надаватиме користувачам доступ до завантаження та скачування файлів певного об'єму, обміну ними. Отже, першочергово слід реалізувати саме функцію завантаження та скачування файлів, створення профілю користувача [15], [16].

В системі за допомогою системи керування версіями (CMS) буде створено функції реєстрації, завантаження та скачування файлів, а також додатковий функціонал (обмін посиланнями на файли, коментування файлів і т. д.) [15], [16]. Такі функції як оплата більшого розміру сховища буде

реалізована готовим кодом, що пропонується сайтом інтернет-гаманця (WebMerchantinterface) [15], [16].

1.4.2 Функціональність системи

Система має підтримувати:

- реєстрація користувачів;
- завантаження файлів у загальнодоступне сховище;
- коментування файлів у загальнодоступному сховищі;
- завантаження файлів у приватне сховище для зареєстрованих користувачів;
- оплата більшого об'єму приватного сховища;
- форма зв'язку з адміністрацією сайту.

1.4.3 Вимоги до інтерфейсу

Інтерфейс системи повинен бути простим та інтуїтивно-зрозумілим для користувачів системи. Вся інформація сайту повинна бути чітко структурована і розділена на розділи.

1.4.4 Вимоги до продуктивності

Система повинна підтримувати до 250 завантажень в секунду. Система та всі її компоненти повинні працювати стабільно і з максимально можливою швидкістю, не нижче ніж 3 секунди на завантаження сторінки при нормальних умовах з боку клієнтського термінала.

Реєстрація користувача з занесенням його даних в базу даних зареєстрованих користувачів і відправленням йому електронного листа повинна займати не більше 5 хвилин. Система повинна підтримувати до 5 запитів на реєстрацію в секунду.

1.4.5 Вимоги до безпеки

Незареєстровані користувачі можуть мати доступ до файлів, збережених як «Загальнодоступні». Зареєстровані можуть завантажувати файли для постійного зберігання та ділитися посиланнями на них. Для безпечного зберігання особистих даних користувача, вони будуть збережені в зашифрованому вигляді.

1.4.6 Експлуатаційні вимоги

Для користування додатком, користувач повинен мати девайс з доступом до мережі Інтернет, обладнаний веб-браузером. Для створення власного сховища користувач має зареєструватися.

1.4.7 Нефункціональні вимоги до продукту

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

- IBM-сумісна серверна станція;
- встановлена та налагоджена серверна операційна система (ОС) – будь-яка з різновидів Unix або Windows Server;
- 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 ПК;
- процесор із тактовою частотою від 1,4GHz;
- обсяг оперативної пам'яті (ОЗУ) – від 3GB (для безперебійної роботи нових версій браузерів);

- підключення до мережі Internet;
- встановлений браузер.

1.3 Висновки за розділом 1

В першому розділі відбулося ознайомлення із проблемною областю віддаленого зберігання файлів з використанням файлового хостингу. Більш детально розглянуто проблеми та перспективи існуючих файлообмінників. На сонові виявлених недоліків сформовано функціональні вимоги до нового застосунку, що буде реалізовувати функції файлообміннику. Для подальшого обґрунтування проектних питань та рішень було сформовано перелік функціональних та не функціональних вимог до застосунку.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

В попередньому розділі відбулося ознайомлення із проблемною областю віддаленого зберігання файлів з використанням файлового хостингу. Більш детально розглянуто переваги та недоліки існуючих файлообмінників. На сонові виявлених недоліків сформовано функціональні вимоги до нового застосунку, що буде реалізовувати функції файлообміннику. Також було сформовано перелік функціональних та не функціональних вимог до застосунку.

Надалі попередньо виконані завдання дозволять визначитися із низкою технічних питань розробки: вибір технології та техніки розробки, принципи реалізації. В цьому розділі буде детально проаналізовано та обрано мову програмування для реалізації функціональної частини, програмні комплекси та засоби для оформлення ПЗ.

2.1 Вибір інструментарію розробки

2.1.1 Вибір мови програмування

PHP сьогодні використовується при розробці більшості сайтів і веб-застосунків. Ця мова була задумана авторами як простий і доступний інструмент для створення динамічних сторінок, і я повинен сказати, що їм це вдалося [17], [18].

Якщо подивитися на історію мови і перші версії, стає ясно, що мова втілила в собі всі кращі можливості існуючих мов програмування для настільних систем, наприклад, багато функцій нагадують поведінку функцій з мови C або C++... Згідно з деякими джерелами, на розробку PHP вплинули такі мови, як Perl, C, C++, Java, Tcl [17], [18].

Як працює PHP. Відомо, що файли сайту розміщуються на сервері, який здатний обробляти скрипти на певній мові, такій як PHP. Сам сервер, наприклад, Apache, написаний на системній мові програмування, наприклад

[17], [18], на С. тобто сервер являє собою просту програму на спеціально підготовленому комп'ютері, яка здатна приймати запити ззовні і відправляти відповіді клієнту, який запросив дані. Оскільки сервер, по суті, являє собою звичайну програму, найкраще було б написати код для нього на тій же системній мові, на якому написаний сам сервер. Але системні мови складні, тому код необхідно було спростити. Для таких цілей і був задуманий мову PHP [17], [18].

Весь написаний код на PHP в момент початку завантаження сторінки завантажується в пам'ять сервера, а код інтерпретується і аналізується. Якщо є помилки, робота зупиняється і видається повідомлення про помилку, якщо ні, код обробляється, перетворюється в машинні коди і виконується [17], [18].

PHP – це свого роду доповнення, аналіз і синтаксичний аналіз PHP-коду відбувається на більш низькому рівні [17], [18]. Як правило, при виконанні коду прості команди PHP переводяться в більш складні низькорівневі, для чого використовуються різні аналізатори (лексичні, синтаксичні і т.д.), використовується транслятор, який генерує коди операцій і команди - після цього вони виконуються віртуальною машиною і переводяться в машину, яка код [17], [18]. У PHP можна просто написати фразу з використанням echo, і ця фраза буде відображатися, але відобразити той же контент на екрані на мові нижчого рівня складніше, довелося б написати там багато підготовчого коду перед відправкою даних на висновок [17], [18].

Головною особливістю PHP є удавана простота і легкість розробки. З одного боку, це плюс, вам не потрібно стежити за типами змінних, вам не потрібно керувати виділенням пам'яті та виконувати інші рутинні операції [17], [18]. З іншого боку, ця простота породила безліч непрофесійних скриптів, які можуть бути небезпечними, складними в обслуговуванні і т.д. почати розробку PHP легко, але для того, щоб все зробити правильно, потрібен досвід [17], [18]. Код може бути написаний з використанням різних

парадигм програмування, будь то об'єктно-орієнтовне програмування або процедурний стиль, або інша парадигма. Добре написаний сценарій залежить від багатьох факторів. Дотримання певних правил для візуального оформлення коду є хорошою практикою-код повинен мати однакові відступи, узгоджене іменування змінних, функцій і т.д. крім зовнішнього дизайну, важлива логіка самої програми - необхідно оптимізувати написаний код, перевірити його на безпеку та інші процедури [17], [18].

Таким чином, розробка PHP може бути хорошої якості, незважаючи на простоту використання мови. Але на відміну від мов низького рівня, PHP не слід і не навчає правильним навичкам програмування, тому Ви повинні самостійно дотримуватися встановлених правил розробки відповідно до документації. Документація доступна на офіційному сайті PHP, в тому числі українською мовою [17], [18].

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

- швидкість роботи та продуктивність – швидкість та продуктивність роботи серверної частини застосунку;
- складність синтаксису – загальна складність синтаксису, що в подальшому пришвидшує розробку, підтримку та розвиток проєкту;
- підтримка – матеріали та спільнота, що допоможе під час розробки;
- програмні бібліотеки фреймворків та систем керування контентом (CMS) – механізми та конструкторські рішення, що спрощують та пришвидшують розробку.

З таблиці видно, що PHP вже довгий час є найбільш популярною мовою програмування для розробки веб-застосунків, саме тому має велику спільноту та найбільш оптимізовані сценарії підтримки на серверах, що робить її найоптимальнішим вибором для розробки.

Таблиця 2.1 – Порівняння мов програмування

Критерій	PHP	C# ASP.NET	Python
Швидкість роботи та продуктивність	++	+	±
Складність синтаксису	+	+	++
Підтримка	++	+	++
Програмні бібліотеки фреймворків та CMS	++	±	±

2.1.2 Система керування контентом

WordPress – це безкоштовна платформа для створення веб-сайтів з відкритим кодом. На більш технічному рівні WordPress – це CMS, написана на PHP, яка використовує базу даних MySQL. Мовою не гіків WordPress – це найпростіший і потужний конструктор блогів і веб-сайтів з існуючих на сьогоднішній день [19], [20].

WordPress-це чудова веб-платформа для різних веб-сайтів. WordPress-це універсальна CMS, від ведення блогів до електронної комерції, бізнес-сайтів і сайтів-портфоліо. Розроблений з урахуванням зручності використання і гнучкості, WordPress є відмінним рішенням як для великих, так і для невеликих веб-сайтів [19], [20].

Веб-сайт WordPress – це будь-який веб-сайт, який використовує WordPress як свою CMS. WordPress підтримує як внутрішню частину веб-сайту (інтерфейс, в якому користувач входить в систему, щоб вносити зміни або додавати новий контент), так і зовнішній інтерфейс (видима частина веб-сайту, яку ваші відвідувачі бачать в Інтернеті) [19], [20].

Ось лише кілька прикладів типів веб-сайтів, які можна створювати за допомогою WordPress [19], [20].

Блог – це особливий тип веб-сайту, призначений для обміну думками, фотографіями, оглядами, навчальними посібниками, рецептами та багато іншого. У блогах зазвичай спочатку відображається самий останній опублікований контент [19, 20].

Веб-сайт електронної комерції – веб-сайт електронної комерції дозволяє вам продавати товари або послуги онлайн і отримувати платежі через систему онлайн-платежів. Користувачі можуть завантажити та встановити плагін електронної комерції WordPress, щоб розширити функціональність WordPress за замовчуванням, щоб у користувача був інтернет-магазин на вашому сайті [19], [20].

Бізнес-сайт – багато підприємств виграють від присутності в Інтернеті у вигляді власного веб-сайту. Якщо користувацький бізнес потребує веб-сайту, щоб клієнти могли дізнатись про компанію та про те, що вона може запропонувати, WordPress – це чудовий варіант. Клієнти можуть зв'язатися з вами, запросити цінову пропозицію, записатися на прийом і багато іншого [19], [20]. Веб-сайт членства – веб-сайт членства дозволяє користувачам розміщувати контент за платним доступом або входом в обліковий запис. Щоб отримати доступ до сторінок або публікацій, користувачі повинні увійти в систему або оплатити контент. WordPress також може обробляти веб-сайти з членством за допомогою додаткових плагінів [19], [20].

Веб-сайт порт фоліо – сайт для демонстрації своєї роботи, навички дизайну та багато іншого за допомогою веб-сайту портфоліо, побудованого на WordPress [19], [20]. Веб-сайт форуму – веб-сайт форуму може бути корисним місцем для користувачів, щоб задавати питання або ділитися порадами [19], [20]. Веб-сайт заходу або ж хостинг заходу. WordPress дозволяє легко ділитися інформацією про вашу подію та продавати квитки [19], [20].

Веб-сайт електронного навчання – студенти можуть відвідувати онлайн-курси, відстежувати свій прогрес, завантажувати ресурси та багато іншого з веб-сайту електронного навчання. За допомогою спеціального плагіна, званого плагіном «WordPress LMS», ви можете пропонувати онлайн-курси з веб-сайту WordPress [19], [20].

2.1.3 Обґрунтування вибору CMS для розробки веб-ресурсу

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

Серед критеріїв порівняння будемо розглядати наступні:

- пошукова оптимізація – чи налагоджена в CMS пошукова оптимізація для популяризації ресурсу;
- адаптованість дизайну сайту – чи адаптовані дизайни (шаблони або теми) CMS до різноманітних форматів відображення (мобільні пристрої, планшети, тощо);
- підтримка системи керування базами даних (СКБД) – які СКБД підтримує фреймворк;
- підтримка розширень – наявність та простота застосування розширень для розширення функціональності застосунків без витрачання часу на розробку нових модулів;
- надійність та захищеність веб-сайту – чи надає CMS додаткові механізми для захисту інформації та гарантування надійності ресурсу.

З порівняння наведеного в таблиці можна зробити висновок, що використання WordPress найбільш оптимальним варіантом. Через її популярність більшість внутрішніх механізмів та процесів максимально оптимізовано під потреби розробників.

Таблиця 2.2 – Порівняння CMS

	WordPress	Joomla!	Drupal
Пошукова оптимізація	++	+	+
Адаптованість дизайну сайту	++	+	+
Підтримка СКБД	+	±	+
Підтримка розширень	++	+	+
Надійність та захищеність веб-сайту	++	+	+

2.2 Структурна схема розробки

На рисунку 2.1 наведено структурну схему ПЗ, що розробляється.

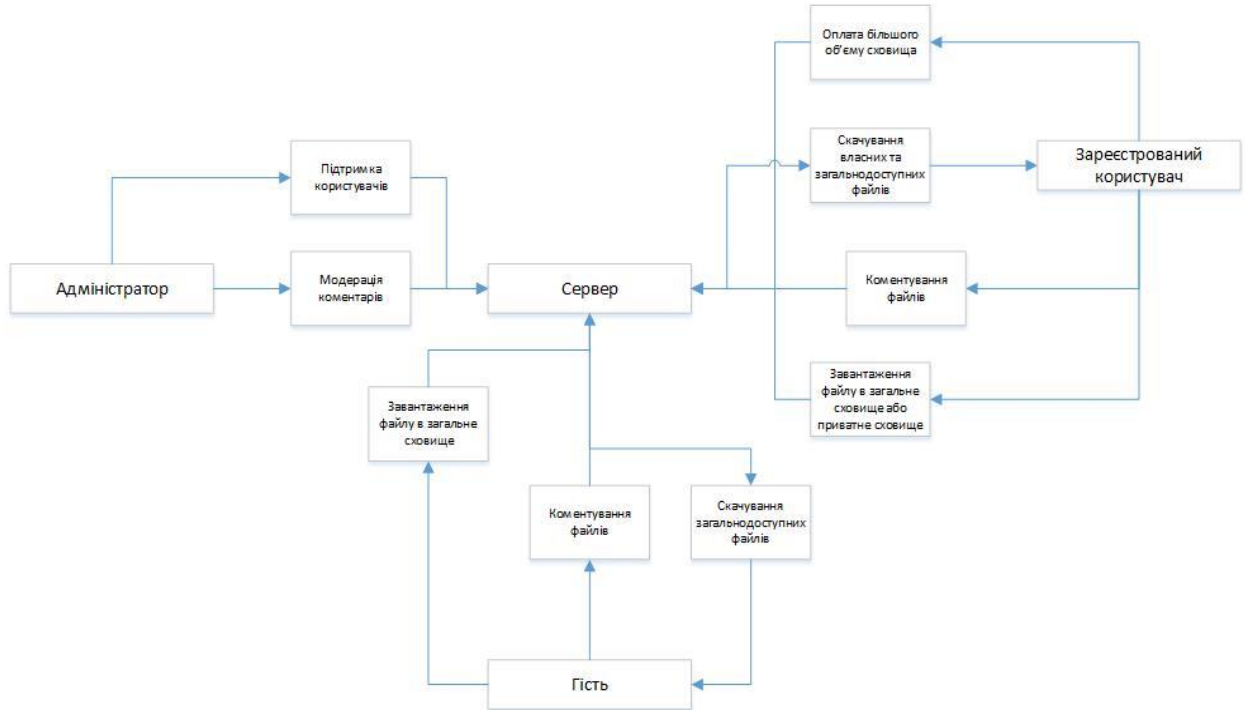


Рисунок 2.1 – Загальна схема системи за обраною темою

WordPress – це проста у встановленні та використанні система керування вмістом з відкритим кодом, яка широко використовується для створення веб-сайтів. Сфера застосування – від блогів до складних веб-сайтів. Вбудована система тем і плагінів в поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти [21].

Написана на мові програмування PHP з використанням бази даних MySQL. Сирцевий код поширюється на умовах ліцензії GNU General Public License [21].

Структура веб-додатків, створених на WordPress зображена на рисунку 2.2.

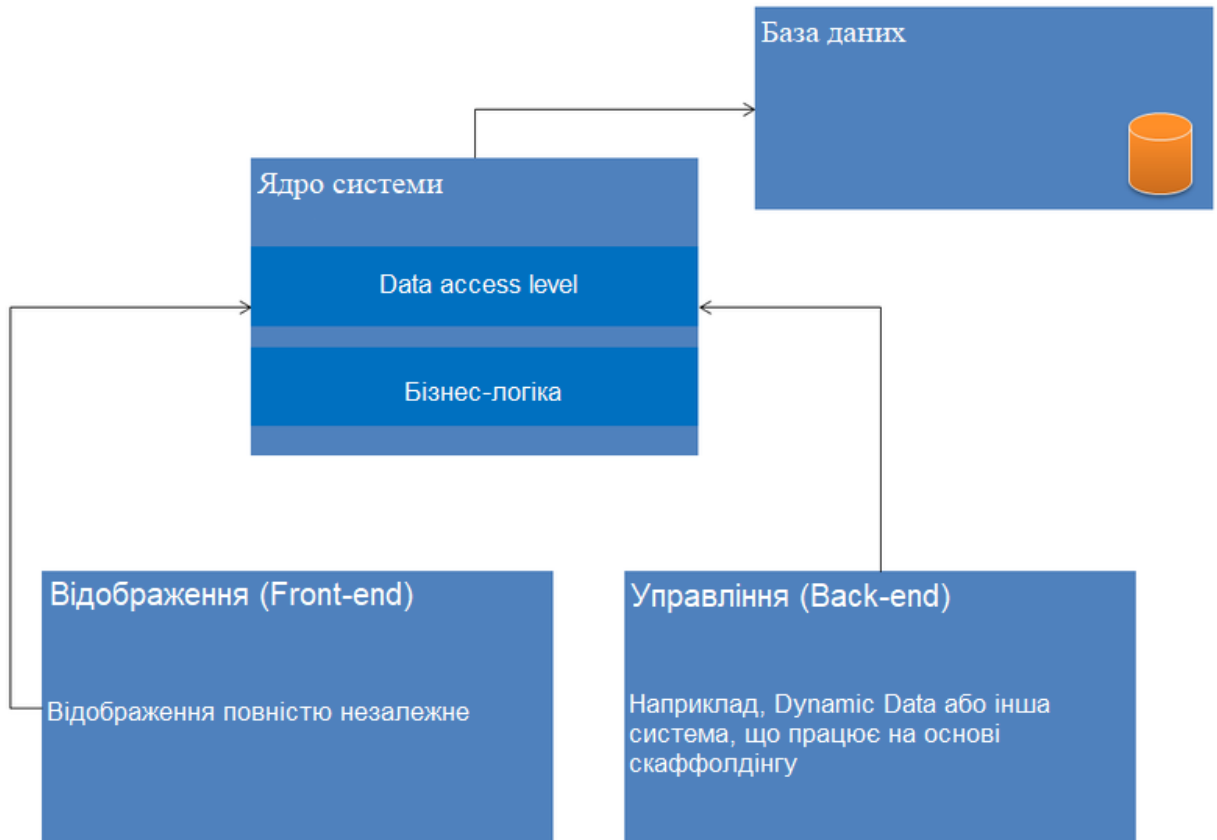


Рисунок 2.2 – Архітектура веб-застосунку

2.3 Висновки за розділом 2

В другому розділі було проведено детальне дослідження загальних технічних питань розробки. Так після аналізу та порівняння різних мов програмування для розробки веб-застосунків було обрано PHP через низку переваг. Для полегшення та пришвидшення розробки було обрано використовувати CMS, серед низки варіантів було обрано WordPress через ряд переваг в порівнянні із іншими CMS.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проектування моделей системи

Unified Modeling Language (UML) — це мова моделювання загального призначення[22]-[24]. Основна мета UML – визначити стандартний спосіб візуалізації того, як була розроблена система. Він дуже схожий на креслення, які використовуються в інших галузях техніки [22]-[24].

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, проектувати та аналізувати [22]-[24]. 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 [22]-[24].

Діаграма класів UML – це графічне позначення, яке використовується для побудови та візуалізації об'єктно-орієнтованих систем [22-24]. Діаграма класів на UML – це тип діаграми статичної структури, яка описує структуру системи, показуючи:

- класи;
- їх атрибути;
- операції (або методи);
- відносини між об'єктами [22]-[24].

Діаграма класів веб-застосунку у вигляді файлообміннику, що розробляється зображена на рис. 3.1.

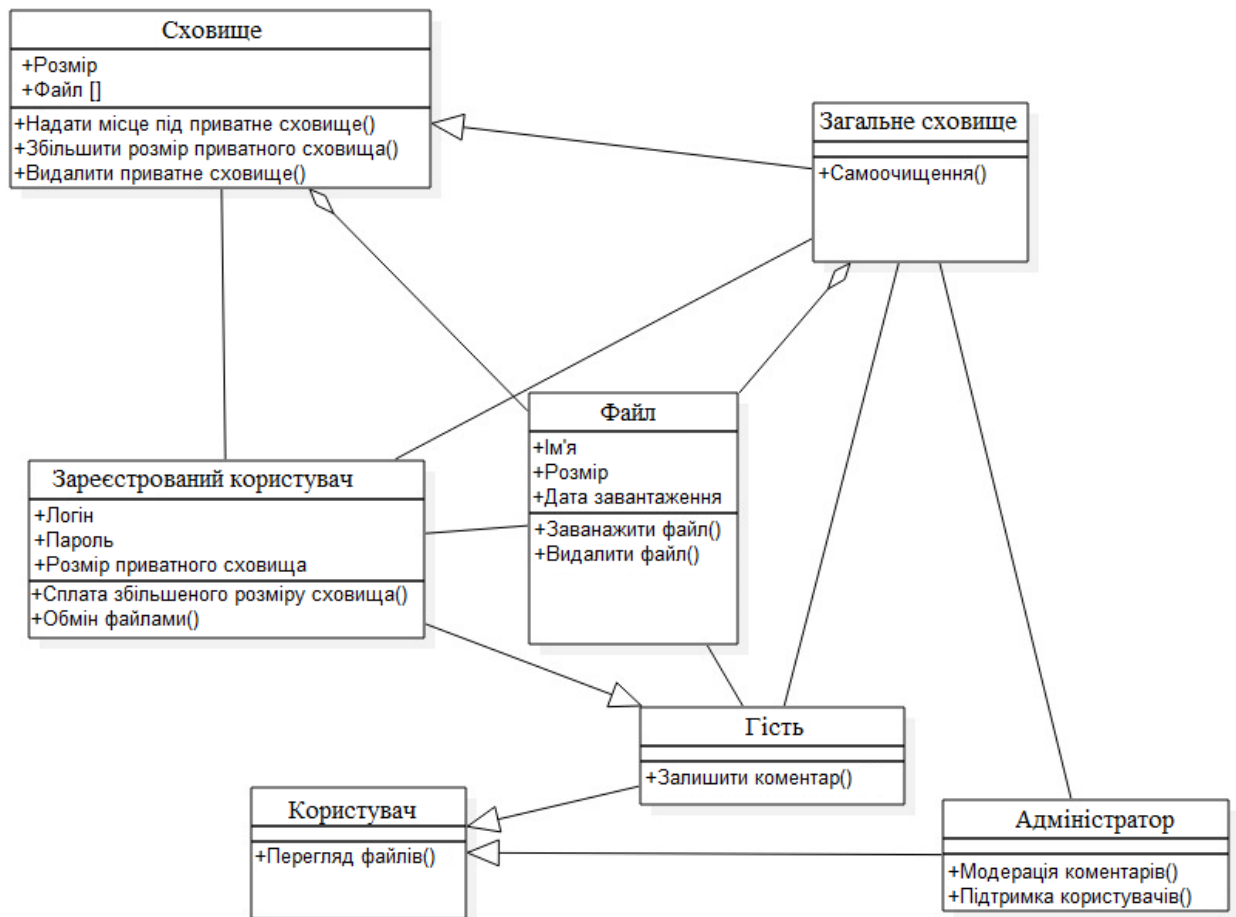


Рисунок 3.1 – Діаграма класів

Діаграми пакетів – це структурні діаграми, які використовуються для показу організації та розташування різних елементів моделі у вигляді пакетів [22]-[24]. 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 багатопов’язаної архітектури в будь-якому класифікаторі UML, наприклад, у програмній системі [22]-[24].

Діаграма пакетів для веб-застосунку у вигляді файлообмінника, що розробляється на рис. 3.2.

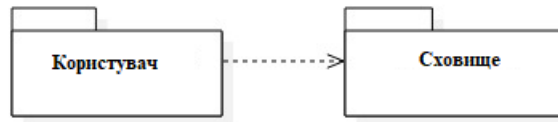


Рисунок 3.2 – Діаграма пакетів

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 [22]-[24]. Існує цілий підхід, пов’язаний з розробкою 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 [22]-[24].

Діаграми компонентів для веб-застосунку у вигляді файлообміннику, що розробляється, зображена на рис. 3.3.

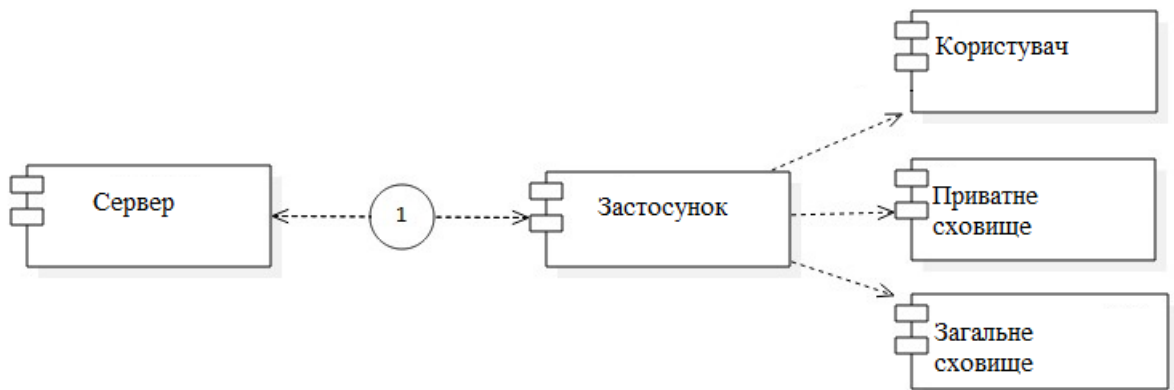


Рисунок 3.3 – Діаграма компонентів

3.2 Моделювання системи з точки зору прецедентів

Діаграма варіантів використання – це графічне зображення можливої взаємодії користувача з системою [22]-[24]. Діаграма варіантів використання показує різні варіанти використання та різні типи користувачів, яких має система, і часто супроводжується діаграмами інших типів. Варіанти використання представлені або колами, або еліпсами. Актори часто зображені у вигляді фігурок [22]-[24].

Прецедент. Завантаження файлу у загальнодоступне сховище (рис. 3.4).

Основна діюча особа. Гість/Зареєстрований користувач.

Основний позитивний сценарій:

- користувач заходить на сайт;
- заходить за загального сховища;
- вибирає пункт меню «Завантажити файл»;
- вибирає файл і заповнює форму;
- виводиться повідомлення про успішне завантаження.

Альтернативний потік АП1:

- користувач невірно заповнює дані у формі;
- вибирає файл зовеликого розміру;
- виводиться повідомлення про помилку.

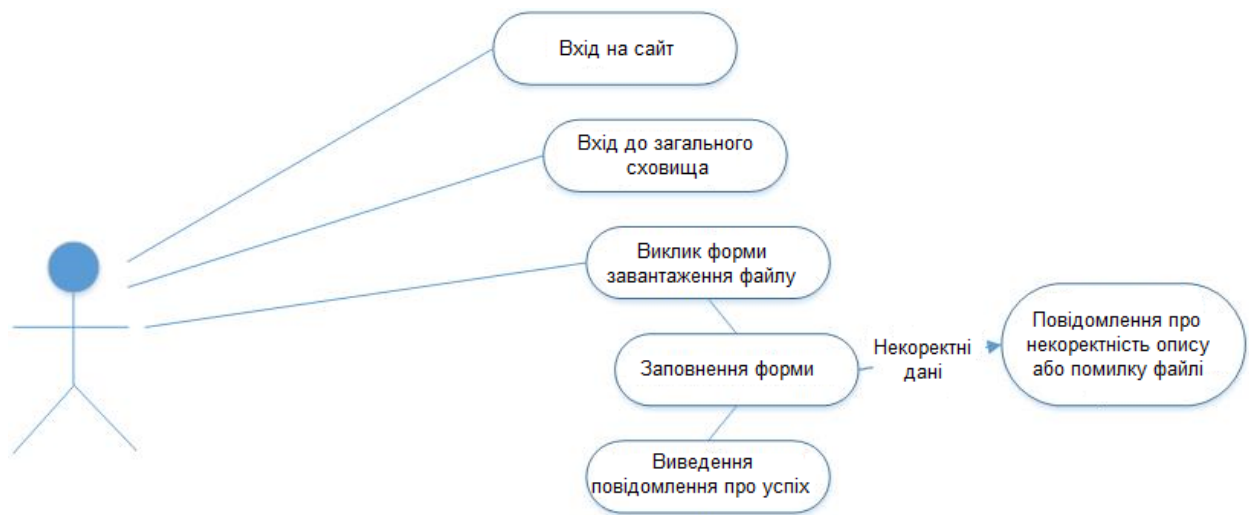


Рисунок 3.4 – Прецедент «Завантаження файлу у загальнодоступне сховище»

3.3 Фізичне проектування

Діаграма активностей (видів діяльності) – один із доступних видів діаграм, що підтримуються Flexberry Designer [22]-[24]. Вона, як і діаграма станів, відбиває динамічні аспекти поведінки системи. По суті, ця діаграма є блок-схемою, яка наочно показує, як потік управління переходить від однієї діяльності до іншої [22]-[24].

Діаграми діяльності для веб-застосунку у вигляді файлообміннику, що розробляється зображені на рис. 3.5–3.7.

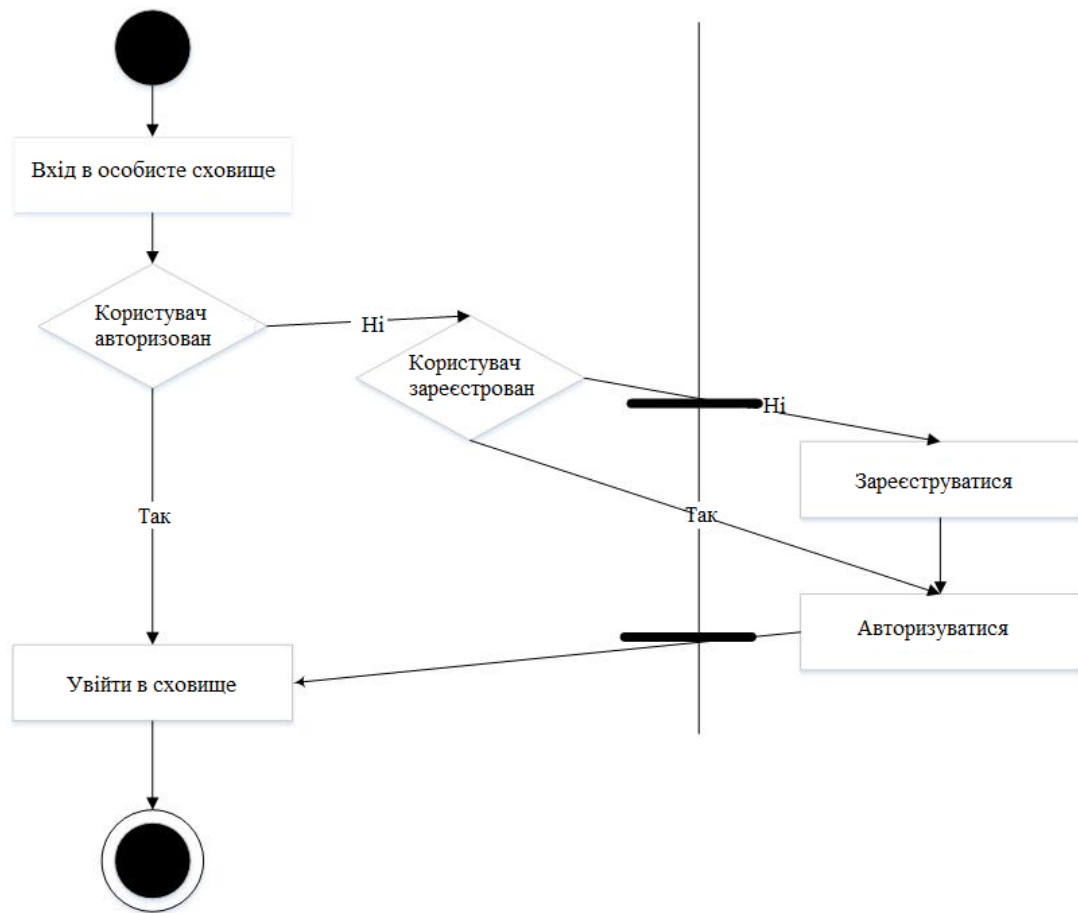


Рисунок 3.5 – Діаграма діяльності: Вхід до приватного сховища

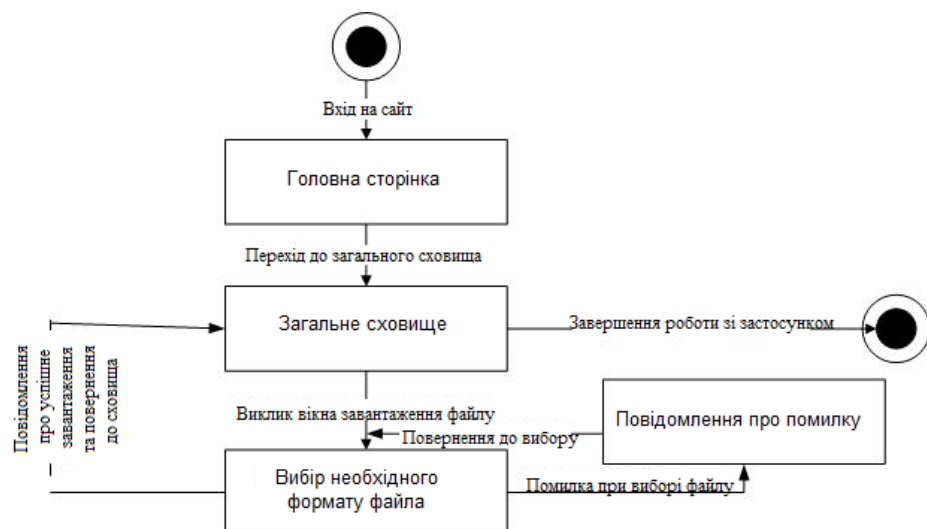


Рисунок 3.6 – Діаграма станів системи

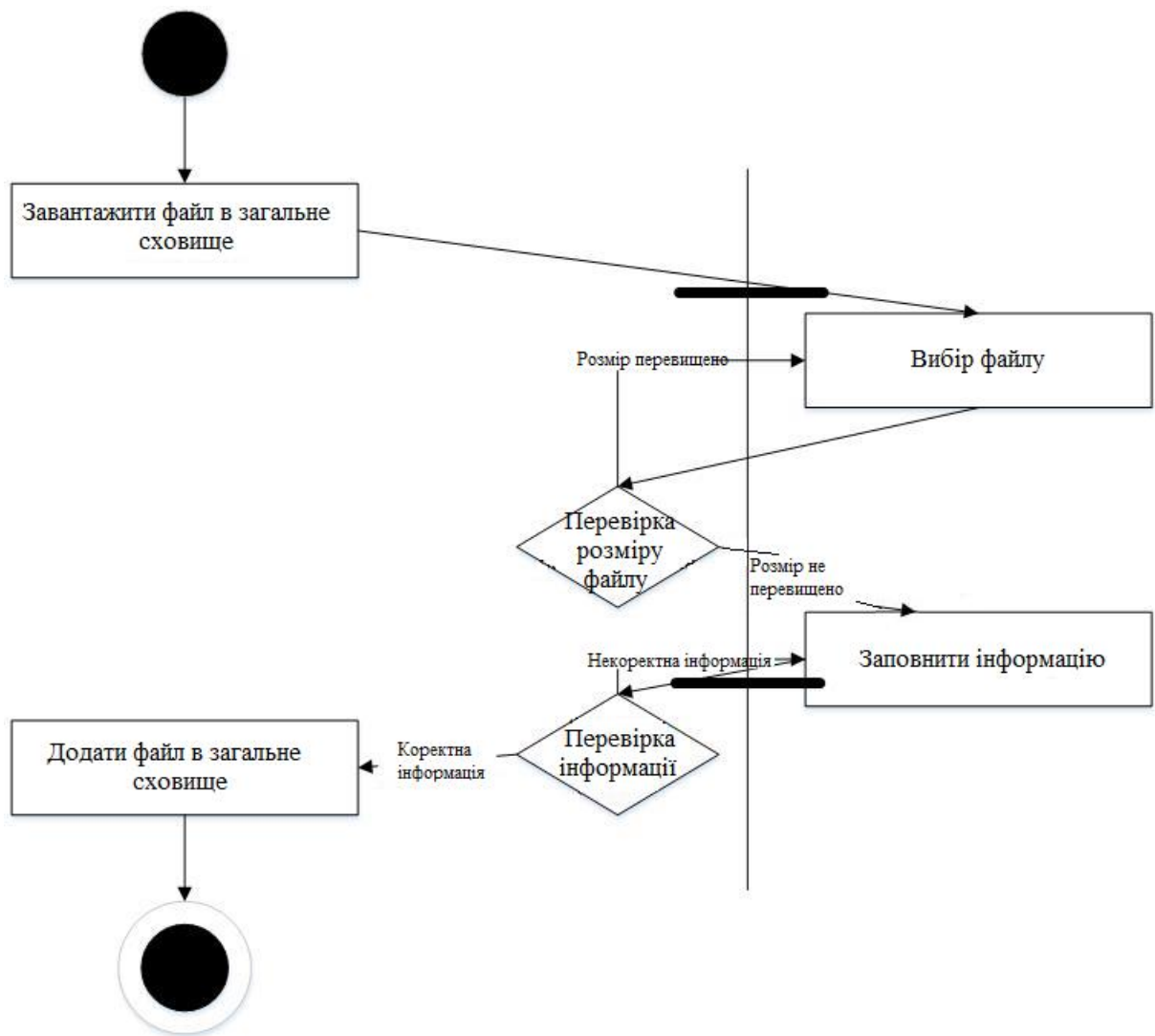


Рисунок 3.7 – Діаграма діяльності: Завантаження файлу

Діаграми кооперацій використовуються, щоб показати, як об'єкти взаємодіють, щоб виконувати поведінку конкретного варіанту використання або частини варіанту використання [22]-[24]. Поряд із діаграмами послідовності, кооперація використовується дизайнерами для визначення й уточнення ролей об'єктів, які виконують певний потік подій у варіанті використання [22]-[24]. Вони є основним джерелом інформації, яка використовується для визначення обов'язків класу та інтерфейсів [22]-[24].

Діаграми кооперацій для веб-застосунку у вигляді файлообміннику, що розробляється зображені на рис. 3.8 та рис. 3.9.



Рисунок 3.8 – Діаграма кооперації. Завантаження файлу

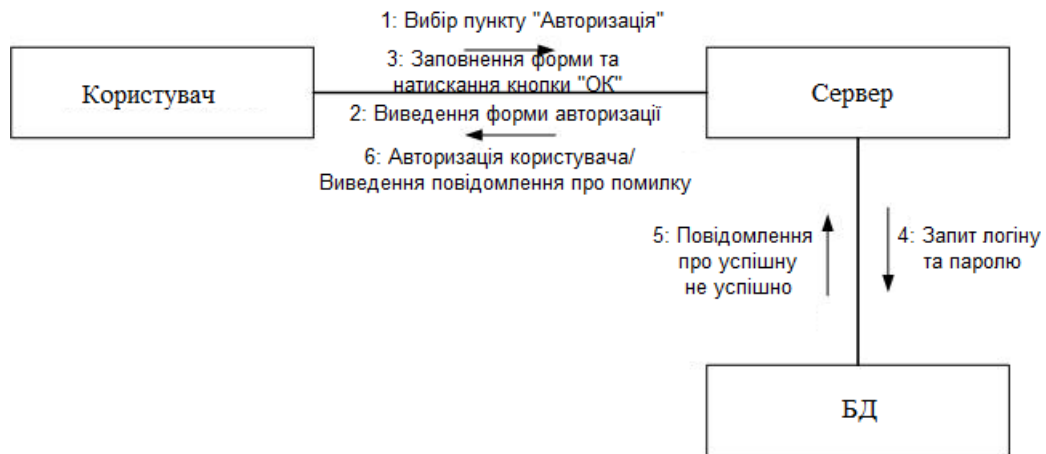


Рисунок 3.9 – Діаграма кооперації. Авторизація користувача

Діаграми послідовностей для веб-застосунку у вигляді файлообміннику, що розробляється зображені на рис. 3.10 та рис. 3.11.



Рисунок 3.10 – Діаграма кооперації. Завантаження файлу

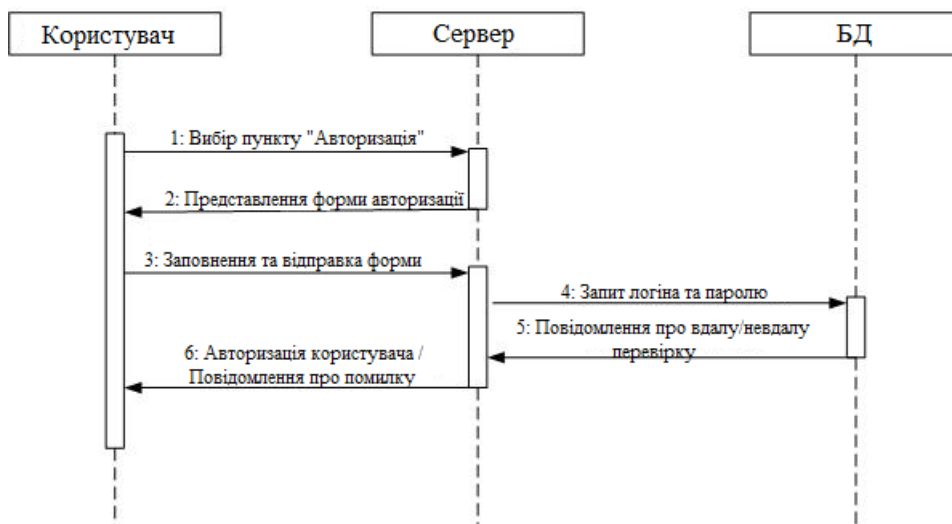


Рисунок 3.11 – Діаграма кооперації. Авторизація користувача

3.4 Проектування інтерфейсу

Спроектований інтерфейс веб-застосунку, враховуючи вимоги до інтерфейсу замовника та ергономічні показники інтерфейсу, зображений на рис. 3.12 – 3.18.

Ласково просимо до файлообміннику!

Вперше тут? Натисніть "Гостьовий режим" для початку ознайомлення

Гостьовий режим

Хочете мати власне сховище? Створіть свій власний акаунт

Створити акаунт

Авторизація

Логін: Пароль:

Рисунок 3.12 – Головна сторінка для неавторизованого користувача

Ім'я_користувача(Вийти)Гість(Авторизуватися)

Новини

Загальне сховище

Приватне сховище

Підтримка

Рисунок 3.13 – Головне меню сайту

Ім'я_користувача(Вийти)

Пошук

Завантажити файл

Список файлів

До власного сховища

На головну

Рисунок 3.14 – Загальне сховище

Ім'я_користувача(Вийти)

Назад

Інформація про файл

Ім'я:

Тип:

Розмір:

Дата завантаження:

Кількість завантажень:

Статус:

Час зберігання:

Завантажити

Коментарі

Список коментарів:

Текст коментаря...

Розмістити

До власного сховища

На головну

Рисунок 3.15 – Сторінка файлу

The screenshot shows a web interface for file uploading. At the top right, there is a user status indicator: "Ім'я_користувача(Вийти)". On the left side, there are three buttons: "Назад", "До власного сховища", and "На головну". The main area is titled "Форма завантаження" and contains a large text input field labeled "Поле для файлу". To the right of this field is a button labeled "Обрати". Below the input field, there is a label "Обрати категорію:" followed by a dropdown menu. At the bottom right of the main area is a button labeled "Завантажити".

Рисунок 3.16 – Форма завантаження файлу

The screenshot shows a web interface for private storage. At the top right, there is a user status indicator: "Ім'я_користувача(Вийти)". On the left side, there are four buttons: "Завантажити", "Створити папку", "Поділитися", and "Видалити". The main area is titled "Файлова система сховища" and contains a large empty space for file management. At the bottom left, there is a button labeled "На головну". At the bottom center, there is a button labeled "Повідомлення". At the bottom right, there is a button labeled "Збільшити місце у сховищі".

Рисунок 3.17 – Приватне сховище

Ім'я_користувача(Вийти)

Форма зворотнього зв'язку

Назад

ПІБ

Категорія питань

Усі питання

З відповідями

F.A.Q.

Відправити

Рисунок 3.18 – Форма зворотнього зв'язку

3.5 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

Веб-застосунок «Файлообмінник» призначений для зручного зберігання і доступу користувача до необхідних йому файлів.

Головною умовою для функціонування системи є наявність сервера, на якому встановлені Apache, PHP5 та MySQL.

4.2 Умови виконання програми

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

- IBM-сумісна серверна станція;
- встановлена та налагоджена серверна ОС – будь-яка з різновидів Unix або Windows Server;
- інтерпретатор PHP5;
- MySQL-сервер для керування користувачами та організації доступу до БД.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

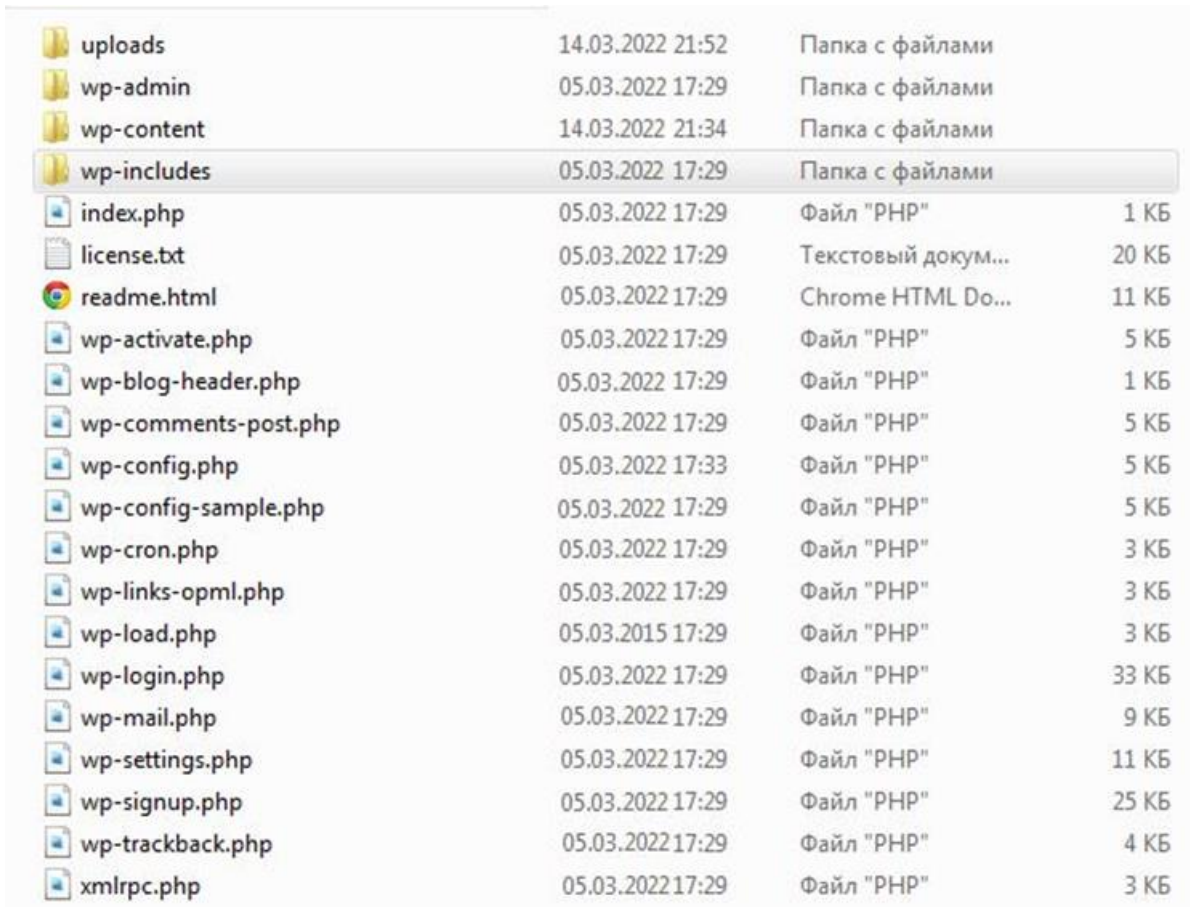
- IBM-сумісний ПК;
- процесор із тактовою частотою від 1,4GHz;
- обсяг ОЗУ – від 3GB (для безперебійної роботи нових версій браузерів);
- підключення до мережі Internet;
- встановлений браузер.

4.3 Інструкція по експлуатації програми

4.3.1 Характеристики програми

Дане програмне забезпечення являється веб-застосунок, розміщеним на сервері, тобто працює цілодобово, поки сервер запущений та знаходиться в працездатному стані.

Проект має наступну файлову структуру (рис 4.1):



uploads	14.03.2022 21:52	Папка с файлами	
wp-admin	05.03.2022 17:29	Папка с файлами	
wp-content	14.03.2022 21:34	Папка с файлами	
wp-includes	05.03.2022 17:29	Папка с файлами	
index.php	05.03.2022 17:29	Файл "PHP"	1 КБ
license.txt	05.03.2022 17:29	Текстовый докум...	20 КБ
readme.html	05.03.2022 17:29	Chrome HTML Do...	11 КБ
wp-activate.php	05.03.2022 17:29	Файл "PHP"	5 КБ
wp-blog-header.php	05.03.2022 17:29	Файл "PHP"	1 КБ
wp-comments-post.php	05.03.2022 17:29	Файл "PHP"	5 КБ
wp-config.php	05.03.2022 17:33	Файл "PHP"	5 КБ
wp-config-sample.php	05.03.2022 17:29	Файл "PHP"	5 КБ
wp-cron.php	05.03.2022 17:29	Файл "PHP"	3 КБ
wp-links-opml.php	05.03.2022 17:29	Файл "PHP"	3 КБ
wp-load.php	05.03.2015 17:29	Файл "PHP"	3 КБ
wp-login.php	05.03.2022 17:29	Файл "PHP"	33 КБ
wp-mail.php	05.03.2022 17:29	Файл "PHP"	9 КБ
wp-settings.php	05.03.2022 17:29	Файл "PHP"	11 КБ
wp-signup.php	05.03.2022 17:29	Файл "PHP"	25 КБ
wp-trackback.php	05.03.2022 17:29	Файл "PHP"	4 КБ
xmlrpc.php	05.03.2022 17:29	Файл "PHP"	3 КБ

Рисунок 4.1 – Файлова структура проекту

У кореневому каталозі зберігаються файли конфігурацій та налаштувань [19]-[21].

У каталозі wp-admin зберігаються файли, необхідні для функціонування консолі Wordpress, за допомогою якої ми працюємо над проектом. Не підлягає редагуванню [19]-[21].

Каталог `wp-includes` містить ядро CMS і не підлягає будь-якому редагуванню.

Каталог `wp-content` цікавить нас найбільше, оскільки саме в ньому зберігаються всі використовуємі сайтом медіафайли, теми, плагіни, мовні пакети та ін. Вміст цього каталогу можна вільно редагувати, не опасаючись пошкодження працездатності CMS [19]-[21].

4.3.2 Звернення до програми

Для виклику додатку слід набрати у адресній сторінці браузера «myfile.com». Для керування роботою додатку використовуються посилання та кнопки [19]-[21].

4.3.3 Вхідні й вихідні дані

Початковими даними програми є дані про користувача, що пройшов реєстрацію, файл та атрибути. Вихідними даними є: виведені на екран користувача сховища, та файли, що користувач завантажує зі сховищ даного веб-застосунку.

4.3.4 Виконання програми

Для запуску застосунку потрібно виконати наступні дії:

- запустити встановлений веб-браузер;
- у адресному рядку набрати «myfile.com» після чого з'явиться головна сторінка застосунку.

Для користування загальнодоступним сховищем слід:

- запустити веб-додаток;
- на боковій панелі (зліва) знайти пункт «Загальне сховище» і перейти до нього за посиланням;

- якщо користувач хоче додати файл, слід натиснути кнопку «AddFile», що знаходиться над списком завантажених файлів, та вибрати необхідний файл з файлової системи комп'ютера, після чого підтвердити свій вибір;

- для скачування файлу достатньо натиснути один раз на його назву.

Для створення власного сховища, користувач має створити свій власний обліковий запис (акант). Для цього:

- на головній сторінці вибрати пункт «створити обліковий запис»;
- у відкритому полі для реєстрації слід заповнити необхідні поля з даними і підтвердити правильність інформації, після чого, якщо поля з даними були заповнені правильно, обліковий запис буде створено.

Якщо у користувача виникає питання стосовно додатку, він може зв'язатися з адміністрацією сайту таким чином:

- на боковій панелі (зліва) вибрати пункт «Зворотній зв'язок»;
- у відкритому полі для зворотного зв'язку слід заповнити необхідні поля з даними і підтвердити правильність інформації, після чого повідомлення буде відправлено.

4.3.5 Розширення функціональності веб-застосунку за допомогою модулів

Profile Builder [19]-[21]. Даний плагін потрібен для роботи з акаунтами. Він дозволяє створювати та редагувати користувацькі акаунти, додає сторінку реєстрації та форми входу до свого облікового запису, а також дозволяє обмежувати доступ до певного контенту незареєстрованим користувачам [19]-[21].

Цей плагін являється одним з ключових при створенні проекту [19]-[21] (рис. 4.2).



Рисунок 4.2 – Profile Builder

File away [19]-[21]. Даний плагін потрібен для роботи з файлами. Він надає можливість завантажувати файли на сервер, а також створювати інтерфейси для відображення завантажених файлів. Він надає велику кількість можливостей для взаємодії з завантаженими файлами [19]-[21].

Цей плагін являється одним з ключових при створенні проекту [19]-[21] (рис. 4.3).



Рисунок 4.3 – Fileaway

Сарча. Існують чотири основні критерії оцінки ергономічності веб-додатку: ефективність, точність, остаточна пам'ять, емоціональна реакція [19]-[21].

Представлений веб-додаток у достатній мірі відповідає цим критеріям, оскільки являється простим і інтуїтивно зрозумілим [19]-[21] (рис. 4.4).

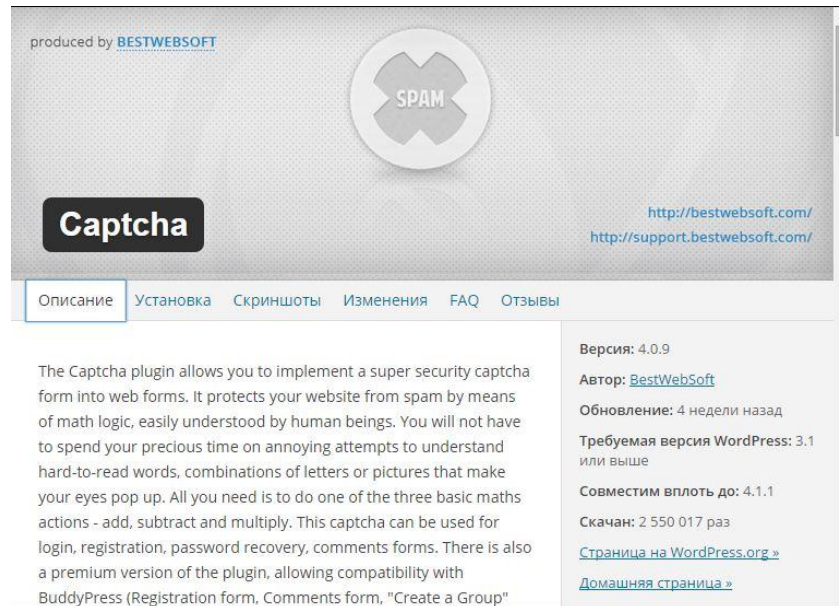


Рисунок 4.4 – Сарча

4.4 Методика та результати тестування

4.4.1 Аналіз реалізованих функціональних вимог

В ході реалізації програмного продукту, було реалізовано наступні функції, передбачені в попередніх планах розробки [22]:

- загальне сховище з можливістю завантаження, скачування та коментування;
- можливість реєстрації для нових користувачів;
- приватне сховище для зареєстрованого користувача;
- модерація коментарів;
- форма зв'язку з адміністрацією сайту.

4.4.2 Ергономічність веб-застосунку

Існують чотири основні критерії оцінки ергономічності веб-додатку: ефективність, точність, остаточна пам'ять, емоціональна реакція [22].

Представлений веб-додаток у достатній мірі відповідає цим критеріям, оскільки являється простим і інтуїтивно зрозумілим.

Приклади основних сторінок застосунку продемонстровані на рис. 4.5 – 4.8.

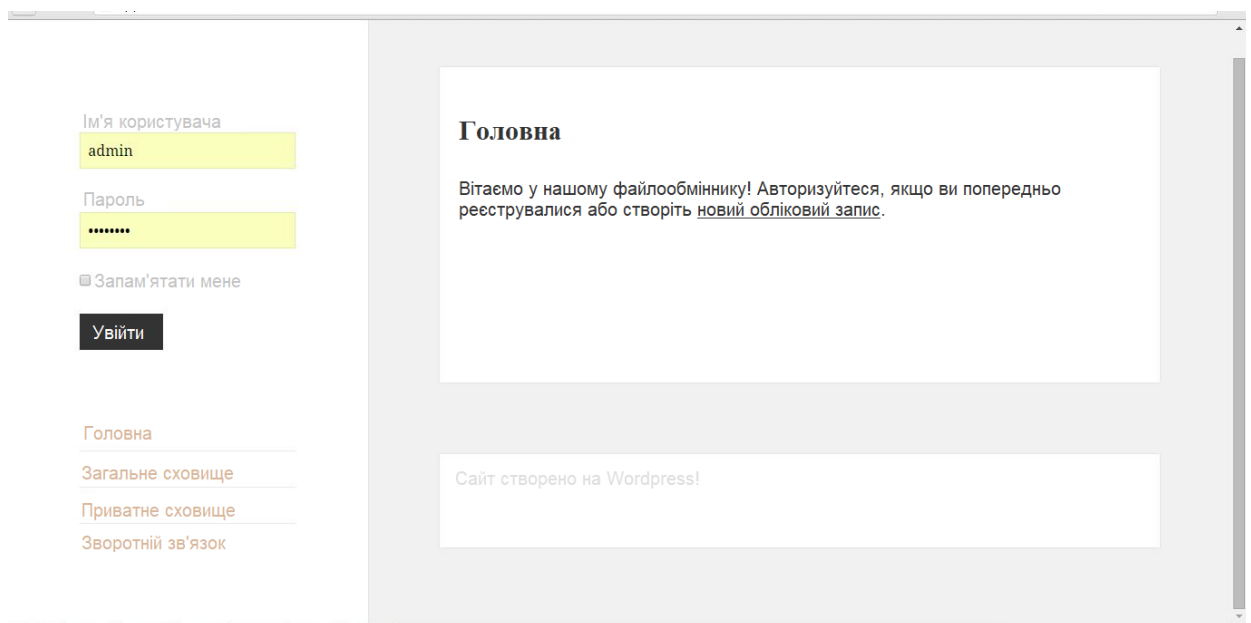


Рисунок 4.5 – Головна сторінка

Як видно зі знімку з екрану, робочу область поділено на дві частини: бокову панель, що служить для зручної навігації по сайту та управління профілем, та область контенту, де відображається вміст сторінки [22].

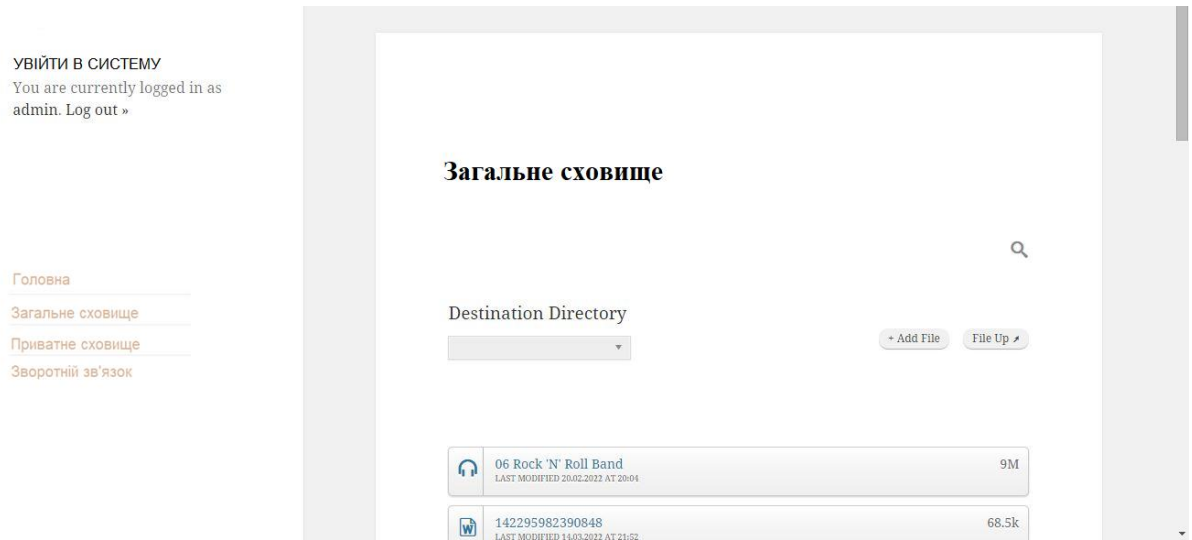
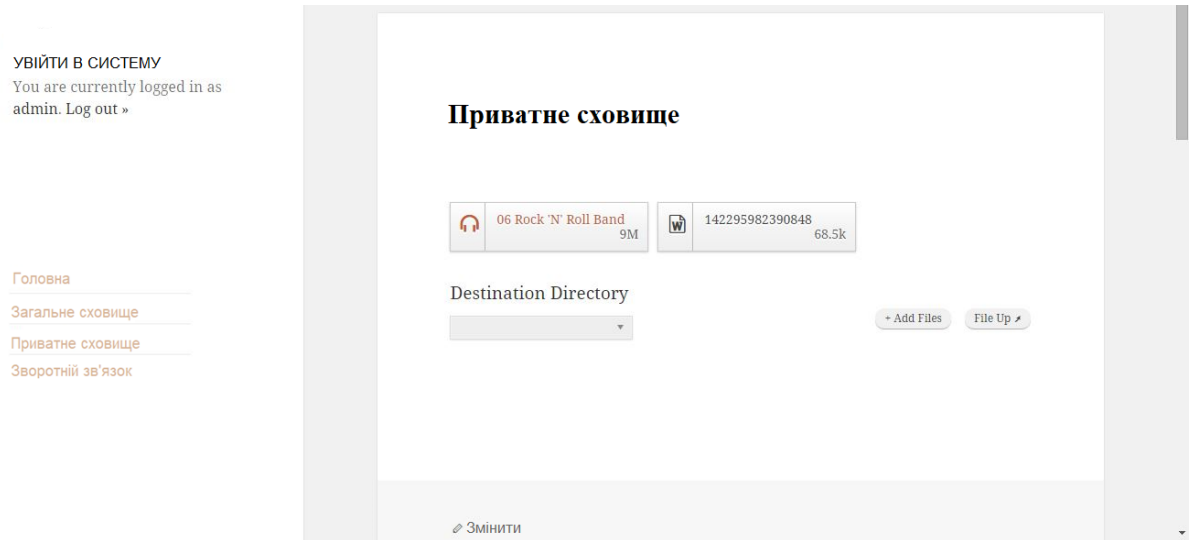


Рисунок 4.6 – Загальне сховище



Риунок 4.7 – Приватне сховище

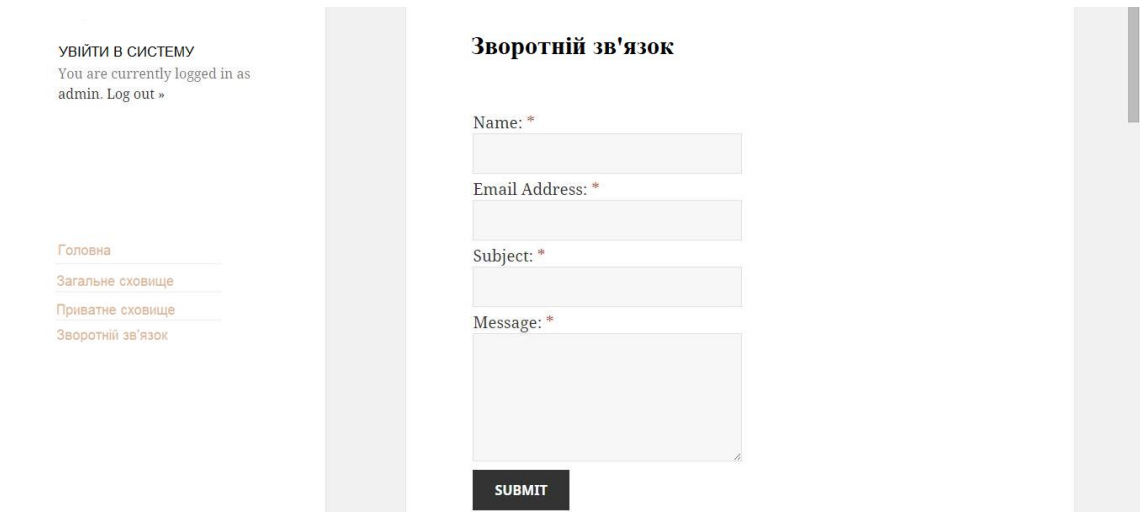


Рисунок 4.8 – Форма зворотного зв'язку

4.4.3 Тестування безпеки веб-додатку

Існують три основні вимоги до безпеки веб-додатків [22]:

- конфіденційність – укриття певних ресурсів від деякої групи користувачів. Оскільки доступ до приватних сховищ мають тільки їх володарі, а доступ до консолі Wordpress має тільки адміністратор, можна вважати, що додаток відповідає даному критерію;
- цілісність – дані сайту можуть змінюватися тільки певною групою осіб (адміністратори) і зміна якихось важливих даних неможлива як для незареєстрованих так і для зареєстрованих користувачів;
- доступність – ресурси мають бути доступні авторизованому користувачу, внутрішньому об'єкту або пристрою.

Даний проєкт відповідає всім вищезазначеним критеріям безпеки.

4.4.4 Тестування стабільності

Даний веб-додаток повністю відповідає всім критеріям стабільності:

- взаємодія між внутрішнім модулями;
- правильне відображення сторінок та їх елементів;
- деструктивне тестування.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра виконано аналіз проблеми реалізації та подальшої підтримки інтернет-сервісу із функціями файлообміннику. Для цього було виконано наступні завдання:

- проаналізовано проблемну область віддаленого зберігання та розповсюдження цифрових даних;
- досліджено існуючі аналогічні рішення;
- спроектовано архітектуру веб-застосунку;
- розроблено веб-застосунок;
- проведено тестування.

Розроблена структура веб-застосунку була реалізована із використанням CMS Wordpress.

Використали модулі розширення функціональності.

Був розроблений сценарій та проведено тестування створеного веб-застосунку за різними напрямками та сценаріями. 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. Бондаренко Т. С. Хмарні технології в соціально-педагогічних системах [Текст] : навч. посіб. / Тетяна Бондаренко ; Укр. інж.-пед. акад. – Харків : Друкарня Мадрид, 2020. – 191 с.
2. Мар'єнко М. В. Наукові платформи та хмарні сервіси, їх місце у системі наукової освіти вчителя [Текст] / М. В. Мар'єнко // Фізико-математична освіта. – 2019. – Вип. 4. – С. 93-99.
3. Гуржій А. М. Хмарні ресурси та сервіси для підготовки майбутніх фахівців з інформаційних технологій: критерії добору, приклади використання [Текст] / А. М. Гуржій, О. Г. Глазунова, Т. В. Волошина, В. І. Корольчук, О. В. Якобчук // Інформаційні технології в освіті. – 2019. – Вип. 3. – С. 7-28.
4. Соловйов А. В. Використання хмарних сервісів GoogleDrive та Telegram при підготовці майбутніх фахівців засобами наскрізного моделювання / А. В. Соловйов // Фіз.-мат. освіта. – 2018. – № 3. – С. 89-93.
5. Shcrabets F. P. The systems of backup power supply based on renewable energy sources for mobile facilities [Текст] / F. P. Shcrabets, P. Yu. Krasovskyi, V. V. Berdnyk // Науковий вісник Національного гірничого університету. – 2017. – № 2. – С. 81-86.
6. Tikhonova N. V. Significance of youth and sports in Ukraine backup for the education of the younger generation [Текст] / N. V. Tikhonova // Pedagogics, psychology, medical-biological problems of physical training and sports. – 2014. – № 9. – С. 60-64.
7. Білодід Н. М. Особливості використання хмарного сховища Dropbox для обміну даними у розподіленій базі даних "1С : Підприємство" [Текст] / Н. М. Білодід, О. В. Власенко, І. А. Оринчак // Вісник Житомирського державного технологічного університету. Серія : Технічні науки. – 2015. – № 1. – С. 84-91.

8. Медведовская О. Г. Программный инструментарий облачного сервиса Dropbox [Текст] / О. Г. Медведовская, В. В. Яценко // Наукові записки [Центральноукраїнського державного педагогічного університету імені Володимира Винниченка]. Серія : Педагогічні науки. – 2018. – Вип. 168. – С. 156-159.
9. Сервіс Dropbox [Електронний ресурс]. – Режим доступу : <https://www.dropbox.com/>
10. Атаян Б. Г. Безопасность резервного копирования данных в облачной системе хранения / Б. Г. Атаян, Т. А. Багдасарян // Безпека інформації. – 2016. – 22, № 2. – С. 119-122.
11. Полупан Ю. В. Один із способів доступу до GoogleDrive без використання авторизації [Текст] / Ю. В. Полупан, Є. В. Полупан // Вісник Східноукраїнського національного університету імені Володимира Даля. – 2018. – № 3. – С. 99-102
12. Сервіс GoogleDrive [Електронний ресурс]. – Режим доступу : https://www.google.com/intl/ua_ua/drive/
13. Сервіс Fex.net [Електронний ресурс]. – Режим доступу : <https://fex.net/uk/>
14. Український хмарний сервіс FEX.NET запустив нову версію платформи, в якій змінив підхід до зберігання даних і "вбив" старі посилання на файли [Електронний ресурс]. – Режим доступу : <https://itc.ua/news/ukrainskiy-oblachnyiy-servis-fex-net-zapustil-novuyu-versiyu-platformyi-v-kotoroy-izmenil-podhod-k-hraneniyu-dannyih-i-ubil-staryie-ssylki-na-faylyi/>
15. Проектування та розробка Web-сторінок. Тема: Основи створення HTML-документів [Текст] : завдання та метод. вказівки до лаб. робіт та самост. роботи для магістрів спец. 8.050201 "Менеджмент організацій" / Укоопспілка, Львів. комерц. акад. ; [уклад. Ладик С. Р., Пенцак Г. С.]. - Л. : Вид-во Львів. комерц. акад., 2011. – 32 с.

16. Єгорова І. М. Проектування та розробка Web-документів : [навч. посіб.] / І. М. Єгорова; Харків. нац. ун-т радіоелектроніки. – Харків : Точка, 2018. – 264 с.
17. Meloni J.C. Sams teach yourself PHP, MySQL and Apache [Text] : all in one / Julie C. Meloni. – 5th ed. – Indianapolis, Indiana : Sams, 2012. – XVI, 653 p.
18. Самойленко Д. М. Створення безпечних шаруватих структур мовою PHP/Secure layers construction on PHP / Д. М. Самойленко // Вісн. Нац. ун-ту "Львів. політехніка". – 2016. – № 852. – С. 93-98.
19. Михалюк Н. Основні функції та можливості системи керування вмістом веб-сайту WordPress [Текст] / Н. Михалюк // Науковий вісник Східноєвропейського національного університету імені Лесі Українки. Міжнародні відносини. – 2018. – № 5. – С. 80-85.
20. Багнюк Н. В. Аналіз та порівняння можливостей розробки веб-ресурсів засобами PHP-фреймворку Laravel і CMSWordPress [Текст] / Н. В. Багнюк, В. М. Мельник, О. В. Казмірчук // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2017. – № 26. – С. 10-15.
21. Мешечко С. С. Методы и способы защиты CMSWordPress [Текст] / С. С. Мешечко, В. Я. Певнев, В. А. Погорелов // Системи управління, навігації та зв'язку. – 2017. – Вип. 1. – С. 23-25.
22. Постіл С. Д. UML. Уніфікована мова моделювання інформаційних систем [Текст] : навч. посіб. / С. Д. Постіл. – Ірпінь : Ун-т держ. фіск. служби України, 2019. – 321 с.
23. Богдан І. Верифікація моделей об'єктно-орієнтованих програм: перевірка на несуперечливість та узгодженість [Текст] / І. Богдан // Техн. науки та технології. – 2017. – № 2. – С. 110-115.
24. Гудзовата О. О. Використання уніфікованої мови візуального моделювання UML (Unified Modeling Language) як інструменту підтримки проектування інформаційних систем [Текст] / О. О. Гудзовата, В. І. Костирко, І. В. Артищук // Підприємництво і торгівля. – 2019. – Вип. 24. – С. 108-114.

25. Коваленко О. В. Імітаційна модель технології тестування безпеки на основі положень теорії масштабування / О. В. Коваленко // Безпека інформації. – 2018. – 24, № 2. – С. 110-117.

ДОДАТОК А
Технічне завдання

Вступ

Застосунок, що розроблюється, має назву «Файлообмінник». У вік комп'ютерних технологій людина зазвичай має більше, ніж один електронний девайс, тому виникає потреба в зручному доступі до цих файлів з будь-якого місця в будь-який час.

Веб-застосунок «Файлообмінник» призначений для зручного зберігання і доступу користувача до необхідних йому файлів.

Головною умовою для функціонування системи є наявність сервера, на якому встановлені Apache, PHP5 та MySQL.

A.1 Підстави для розробки

Розробка виконується на основі завдання на випускню дипломну роботу бакалавра, яке було визначене за темою «Програмна реалізація системи файлообмінника» і затверджене наказом № 102 від 27 квітня 2022 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

«Файлообмінник» можна віднести до категорії «Веб-сервісів». Він створений для тимчасового або постійного зберігання файлів з метою зручного доступу до них з будь-якого девайсу, що має доступ до мережі Інтернет.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Система має підтримувати:

- реєстрація користувачів;

- завантаження файлів у загальнодоступне сховище;
- коментування файлів у загальнодоступному сховищі;
- завантаження файлів у приватне сховище для зареєстрованих користувачів;
- оплата більшого об'єму приватного сховища;
- форма зв'язку з адміністрацією сайту.

A.3.2 Вимоги до надійності

Система повинна запускатися з сервера, також повинна бути забезпечена збережуваність інформації при збоях та її захист від несанкціонованого доступу, а так само на всіх етапах збору, обробки та зберігання за рахунок створення страхових копій бази даних та застосування системи паролів при організації доступу до інформації.

Для захищення особових даних кожен користувач мусить мати свій логін та пароль.

A.3.3 Умови експлуатації

Для користування додатком, користувач повинен мати девайс з доступом до мережі Інтернет, обладнаний веб-браузером. Для створення власного сховища користувач має зареєструватися.

A.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 станція;
- встановлена та налагоджена серверна ОС – будь-яка з різновидів Unix або Windows Server;

- 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 1,4GHz;
- обсяг ОЗУ – від 3GB (для безперебійної роботи нових версій браузерів);
- підключення до мережі Internet;
- встановлений браузер.

A.3.6 Вимоги до маркування та упакування

Перша версія матиме маркування 1.0, наступні версії матимуть маркування 1.1, 2.0 і т.д.

A.3.7 Вимоги до транспортування та збереження

Допускається транспортування програмного виробу в транспортній тарі усіма видами транспорту. При транспортуванні та зберіганні програмного виробу має бути передбачений захист від попадання пилу і атмосферних опадів.

A.3.8 Вимоги до програмної документації

До складу програмної документації повинні входити:

- документ опису вимог;
- технічне завдання;
- опис програми.

A.4 Стадії та етапи розробки

Розробка всього проекту складається з 4х етапів:

- аналіз даних. Він включає в себе два під етапи: збір та аналіз даних і складання технічного завдання;
- підготовка до розробки. Він включає в себе такі підетапи: розробка архітектури проекту та підбирання готових компонентів-модулів для системи;
- розробка програмного продукту;
- фінальне тестування системи та здача замовнику готового додатку з усіма необхідними документами.

A.5 Порядок контролю та приймання системи

Для контролю та приймання системи використовується широко розповсюджена методологія керування проектами SCRUM. Тобто весь проект розбивається на декілька модулів, при завершенні розробки певного модуля проект представляє собою завершену повноцінно працюючу систему.

Коли програмне забезпечення повністю розроблене та протестоване, воно передається замовнику на флеш-носії, двома архівами з розширенням zip, які містять у собі: перший – файли готового проекту та експортовану БД, другий – всю необхідну документацію.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Застосунок, що розроблюється, має назву «Файлообмінник». У вік комп'ютерних технологій людина зазвичай має більше, ніж один електронний девайс, тому виникає потреба в зручному доступі до цих файлів з будь-якого місця в будь-який час.

Веб-застосунок «Файлообмінник» призначений для зручного зберігання і доступу користувача до необхідних йому файлів.

Головною умовою для функціонування системи є наявність сервера, на якому встановлені Apache, PHP5 та MySQL.

Б.2 Функціональне призначення

Система має підтримувати:

- реєстрація користувачів;
- завантаження файлів у загальнодоступне сховище;
- коментування файлів у загальнодоступному сховищі;
- завантаження файлів у приватне сховище для зареєстрованих користувачів;
- оплата більшого об'єму приватного сховища;
- форма зв'язку з адміністрацією сайту.

Б.3 Опис логічної структури

На рисунку Б.1 наведено структурну схему ПЗ, що розробляється.

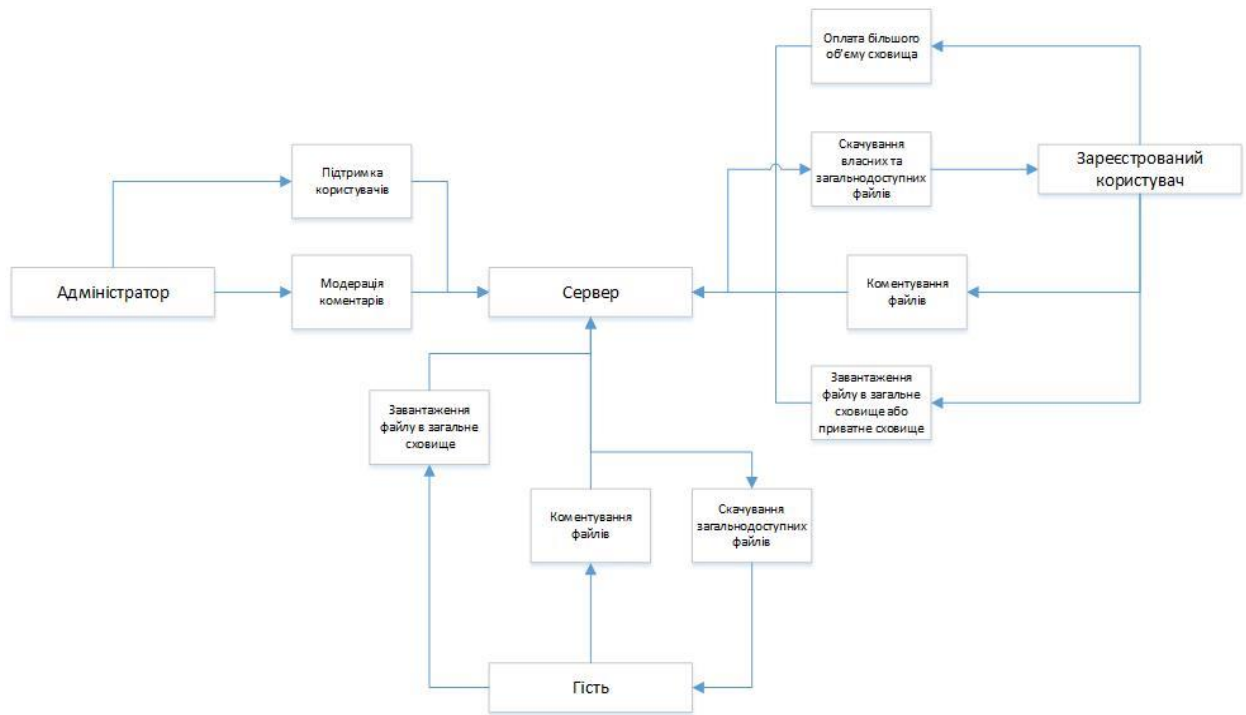


Рисунок Б.1 – Загальна схема системи за обраною темою

WordPress – це проста у встановленні та використанні система керування вмістом з відкритим кодом, яка широко використовується для створення веб-сайтів. Сфера застосування – від блогів до складних веб-сайтів. Вбудована система тем і плагінів в поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти.

Написана на мові програмування PHP з використанням бази даних MySQL. Сирцевий код поширюється на умовах ліцензії GNU General Public License.

Структура веб-додатків, створених на WordPress зображена на рисунку Б.2.

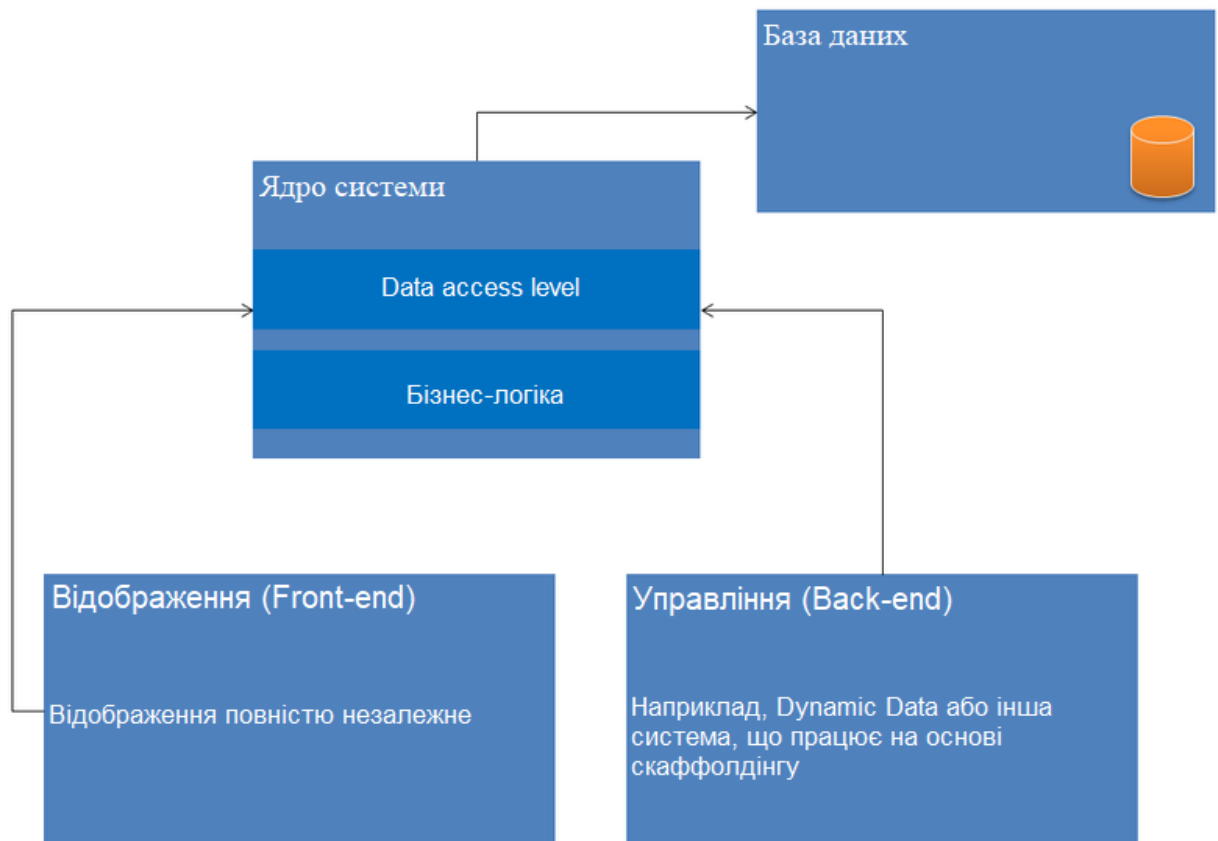


Рисунок Б.2 – Архітектура веб-застосунку

Б.4 Використані технічні засоби

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

- IBM-сумісна серверна станція;
- встановлена та налагоджена серверна ОС – будь-яка з різновидів Unix або Windows Server;
- 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 1,4GHz;
- обсяг ОЗУ – від 3GB (для безперебійної роботи нових версій браузерів);
- підключення до мережі Internet;
- встановлений браузер.

Б.5 Виклик та завантаження

Для виклику додатку слід набрати у адресній сторінці браузера «myfile.com». Для керування роботою додатку використовуються посилання та кнопки.

Б.6 Вхідні та вихідні дані

Початковими даними програми є дані про користувача, що пройшов реєстрацію, файл та атрибути. Вихідними даними є: виведені на екран користувача сховища, та файли, що користувач завантажує зі сховищ даного веб-застосунку.

ДОДАТОК В
Текст програми

```

require_once 'node.namespace.php';

use node\DB as DB;
use node\Settings as Settings;

class Upload
{

    public static string $f_n;
    public static mixed $f_e;
    public static string $f_m;
    public static string $hsa1;
    public static string $n_N;
    public static string $n_N_FULL;
    public static mixed $IP;

    public static string $f_s;
    public static string $t_f;

    public static function reFiles($files): array
    {
        $res = [];
        $files = self::diverArr($files);

        foreach ($files as $file) {
            self::$f_n = strip_tags($file['name']);
            self::$f_s = $file['size'];
            self::$t_f = $file['tmp_name'];
            self::$hsa1 = hsa1_file(self::$t_f);
        }
    }
}

```

```

        $res[] = [self::$f_n, self::$f_s, self::$t_f, self::$hsa1];
    }
    return $res;
}

```

```

public static function diverArr($files): array
{
    $res = [];

    foreach ($files as $key1 => $val1) {
        foreach ($val1 as $key2 => $val2) {
            $res[$key2][$key1] = $val2;
        }
    }
    return $res;
}

```

```

/**
 * @throws Exception
 */
public static function uploadFile(): array
{
    Settings::loadConfig();
    self::fileInfo();

    if (Settings::$BLACKLIST_DB) {
        DB::checkFileBlacklist();
    }

    if (Settings::$FILTER_MODE) {

```

```

        self::chckMBL();
        if(!is_null(self::$f_e)){
            self::chck_Ext_MBL();
        }
    }

    if (Settings::$ant_dp) {
        DB::antidp();
    }

    if (!Settings::$ant_dp) {
        self::generateName();
    }

    if (!is_dir(Settings::$FILES_ROOT)) {
        throw new Exception('File storage path not accessible.', 500);
    }

    if (!move_uploaded_file(self::$t_f, Settings::$FILES_ROOT .
self::$n_N_FULL)) {
        throw new Exception('Failed to move file to destination', 500);
    }

    if (!chmod(Settings::$FILES_ROOT . self::$n_N_FULL, 0644)) {
        throw new Exception('Failed to change file permissions', 500);
    }

    DB::newIntoDB();

    if (Settings::$SSL) {

```

```

        $preURL = 'https://';
    } else {
        $preURL = 'http://';
    }

    return [
        'hash' => self::$hsa1,
        'name' => self::$f_n,
        'url'   => $preURL . Settings::$URL . '/' .
rawurlencode(self::$n_N_FULL),
        'size' => self::$f_s
    ];
}

class resp
{
    private mixed $type;

    public function error($code, $desc)
    {
        $resp = null;

        switch ($this->type) {
            case 'csv':
                $resp = $this->csvError($desc);
                break;
            case 'html':
                $resp = $this->htmlError($code, $desc);
                break;

```

```

        case 'json':
            $resp = $this->jsonError($code, $desc);
            break;
        case 'text':
            $resp = $this->txtErr($code, $desc);
            break;
    }
    http_resp_code($code);
    echo $resp;
}

private static function csvError($descript): string
{
    return "error" . "\r\n" . "\"$descript\"" . "\r\n";
}

private static function htmlError($code, $descript): string
{
    return '<p>ERROR: ( ' . $code . ' ) ' . $descript . '</p>';
}

private static function jsonError($code, $descript): bool|string
{
    return json_encode([
        'success' => false,
        'errorcode' => $code,
        'descript' => $descript,
    ], JSON_PRETTY_PRINT);
}

```

```

private static function txtErr($code, $descript): string
{
    return 'ERROR: ( ' . $code . ' ) ' . $descript;
}

public function send($files)
{
    $resp = null;

    switch ($this->type) {
        case 'csv':
            $resp = $this->csvSuccess($files);
            break;
        case 'html':
            $resp = $this->htmlSuccess($files);
            break;
        case 'json':
            $resp = $this->jsonSuccess($files);
            break;
        case 'text':
            $resp = $this->textSuccess($files);
            break;
    }

    http_resp_code(200); // "200 OK". Success.
    echo $resp;
}

private static function csvSuccess($files): string
{

```

```

$result = "name","url","hash","size" . "\r\n";
foreach ($files as $file) {
    $result .= "" . $file['name'] . "" . ',' .
        "" . $file['url'] . "" . ',' .
        "" . $file['hash'] . "" . ',' .
        "" . $file['size'] . "" . "\r\n";
}

return $result;
}

private static function htmlSuccess($files): string
{
    $result = "";

    foreach ($files as $file) {
        $result .= '<a href=' . $file['url'] . '>' . $file['url'] . '</a><br>';
    }

    return $result;
}

private static function jsonSuccess($files): bool|string
{
    return json_encode([
        'success' => true,
        'files' => $files,
    ], JSON_PRETTY_PRINT);
}

```

ДОДАТОК Д
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»

Програмна реалізація системи файлообмінника

Котов Олександр Олександрович

ст. групи КНТ-119сп

Коцур Михайло Ігорович

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес програмної реалізації файлообміннику.

Предмет дослідження – методи реалізації файлообміннику.

Метою роботи є розроблення системи файлообміннику для віддаленого зберігання та поширення цифрових даних за допомогою мережі Інтернет.

Рисунок Д.2 – Слайд 2

Постановка завдання

Метою проекту є реалізація файлообміннику, що дозволить зберігати на сервері файли у приватній та загальнодоступній формі, ділитися ними з іншими користувачами та отримувати посилання на завантажений файл. Також має бути передбачена можливість платного розширення місця на сервері, доступного для зареєстрованого користувача.

3

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Dropbox	Google Drive	Fex.net	Застосунок, що розробляється
Безкоштовний функціонал	Істотні обмеження за об'ємом пам'яті	Обмеження за об'ємом пам'яті	Обмеження за часом зберігання	Відсутність обмежень
Розподіл приватного та загального сховища	Окрема регулюється через налаштування прав доступу	Окрема регулюється через налаштування прав доступу	Організований розподіл	Організований розподіл
Обмеження за форматом файлів	Присутні	Присутні	Відсутні	Відсутні
Опис та коментування матеріалів	Окрема регулюється через налаштування	Окрема регулюється через налаштування	Можливо	Можливо

4

Рисунок Д.4 – Слайд 4

Порівняння мов програмування			
Критерій	PHP	C# ASP.NET	Python
Швидкість роботи та продуктивність	++	+	±
Складність синтаксису	+	+	++
Підтримка	++	+	++
Програмні бібліотеки фреймворків та CMS	++	±	±

5

Рисунок Д.5 – Слайд 5

Порівняння CMS			
	WordPress	Joomla!	Drupal
Пошукова оптимізація	++	+	+
Адаптованість дизайну сайту	++	+	+
Підтримка СКБД	+	±	+
Підтримка розширень	++	+	+
Надійність та захищеність веб-сайту	++	+	+

6

Рисунок Д.6 – Слайд 6

Проектування моделей системи

```

classDiagram
    class Сховище {
        +Розмір
        +Файл []
        +Повідомити про приватне сховище()
        +Збільшити розмір приватного сховища()
        +Віддати приватне сховище()
    }
    class Зареєстрований_користувач {
        +Пити
        +Пароль
        +Розмір приватного сховища
        +Сповістити про збільшення розміру сховища()
        +Обрати файли()
    }
    class Файл {
        +Ім'я
        +Розмір
        +Дата завантаження
        +Завантажити файл()
        +Віддати файл()
    }
    class Гість {
        +Захистити коментар()
    }
    class Користувач {
        +Ім'я
        +Пароль
    }
    class Зарегистрированное_хранилище {
        +Самозашифрування()
    }
    class Администратор {
        +Модифікувати коментар()
        +Підтримка користувачів()
    }

    Користувач <|-- Гість
    Користувач <|-- Зареєстрований_користувач
    Зарегистрированное_хранилище <|-- Сховище
    Сховище --> Зареєстрований_користувач
    Зареєстрований_користувач --> Файл
    Файл --> Гість
    Гість --> Зареєстрований_користувач
    Зареєстрований_користувач --> Зарегистрированное_хранилище
    Зареєстрований_користувач --> Администратор
    Файл --> Администратор
    Гість --> Администратор
  
```

Рисунок Д.8 – Слайд 8



Рисунок Д.9 – Слайд 9

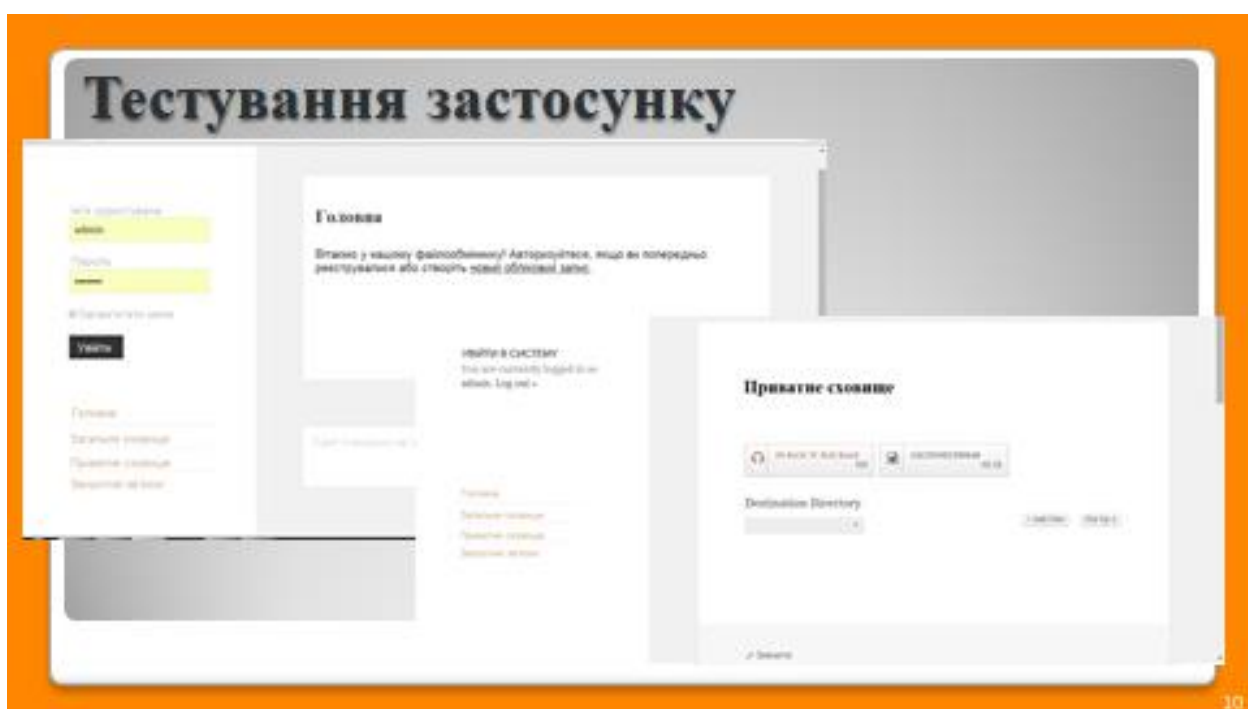


Рисунок Д.10 – Слайд 10

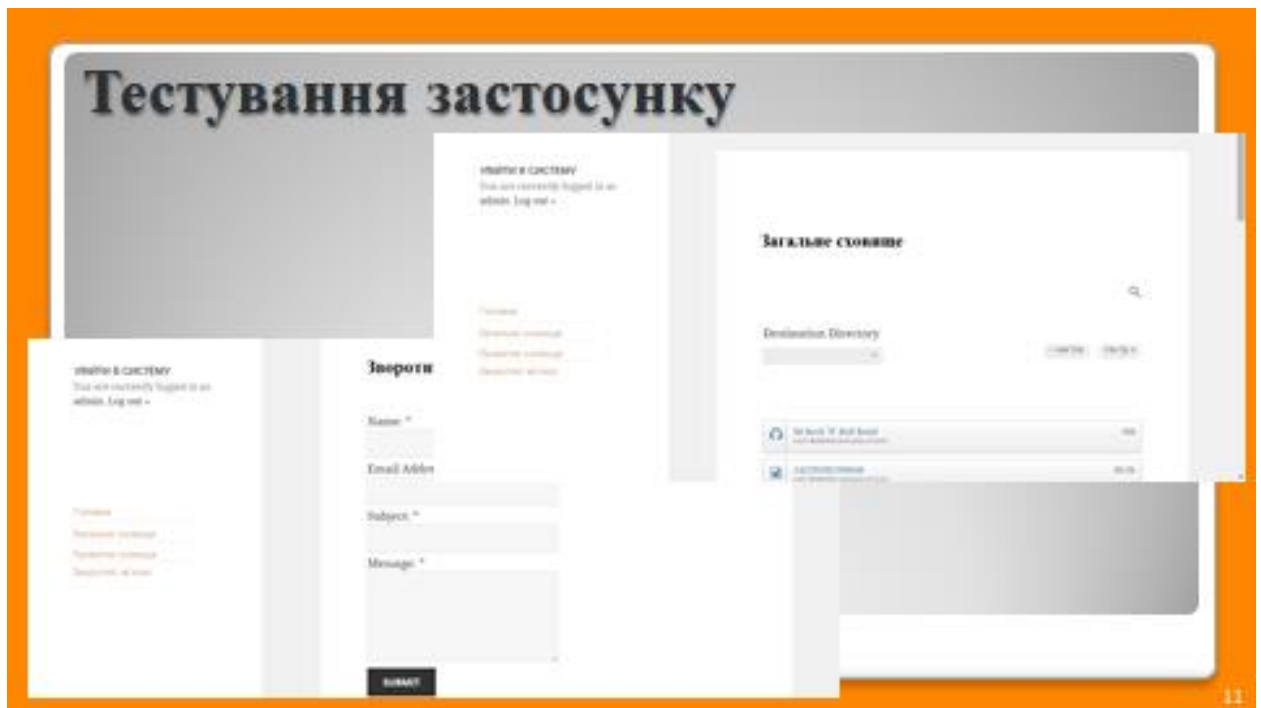


Рисунок Д.11 – Слайд 11

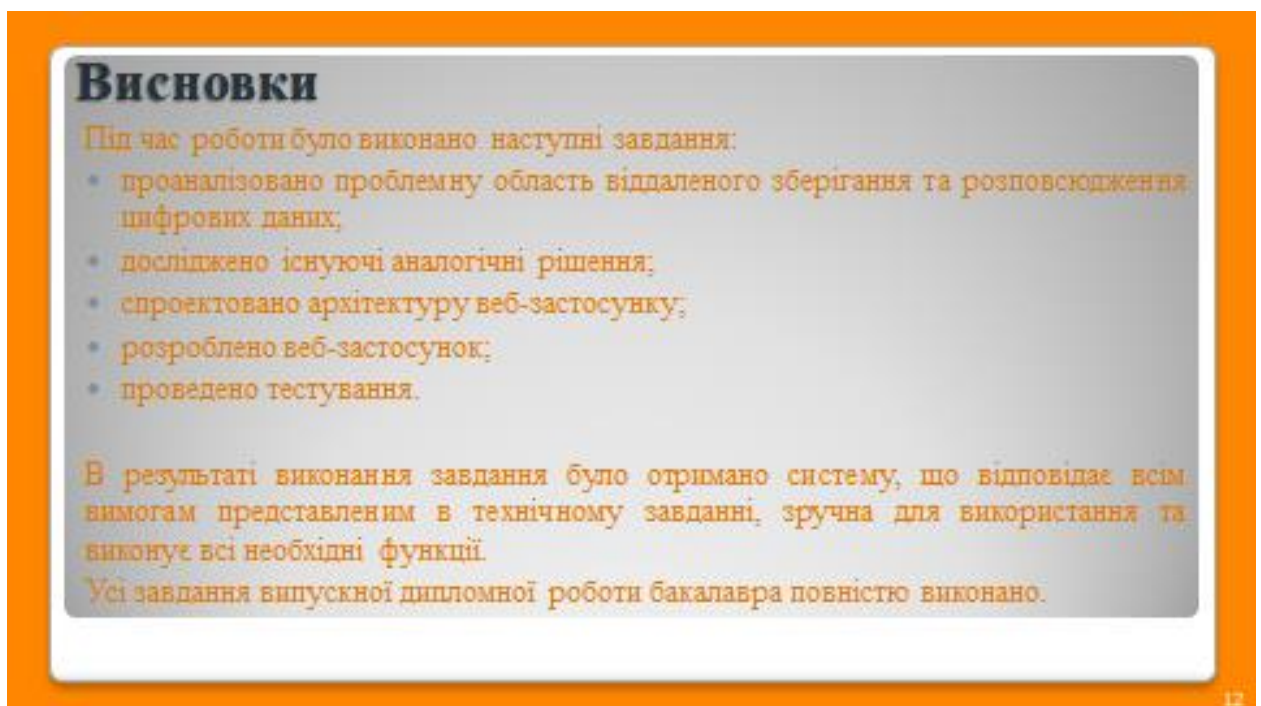


Рисунок Д.12 – Слайд 12

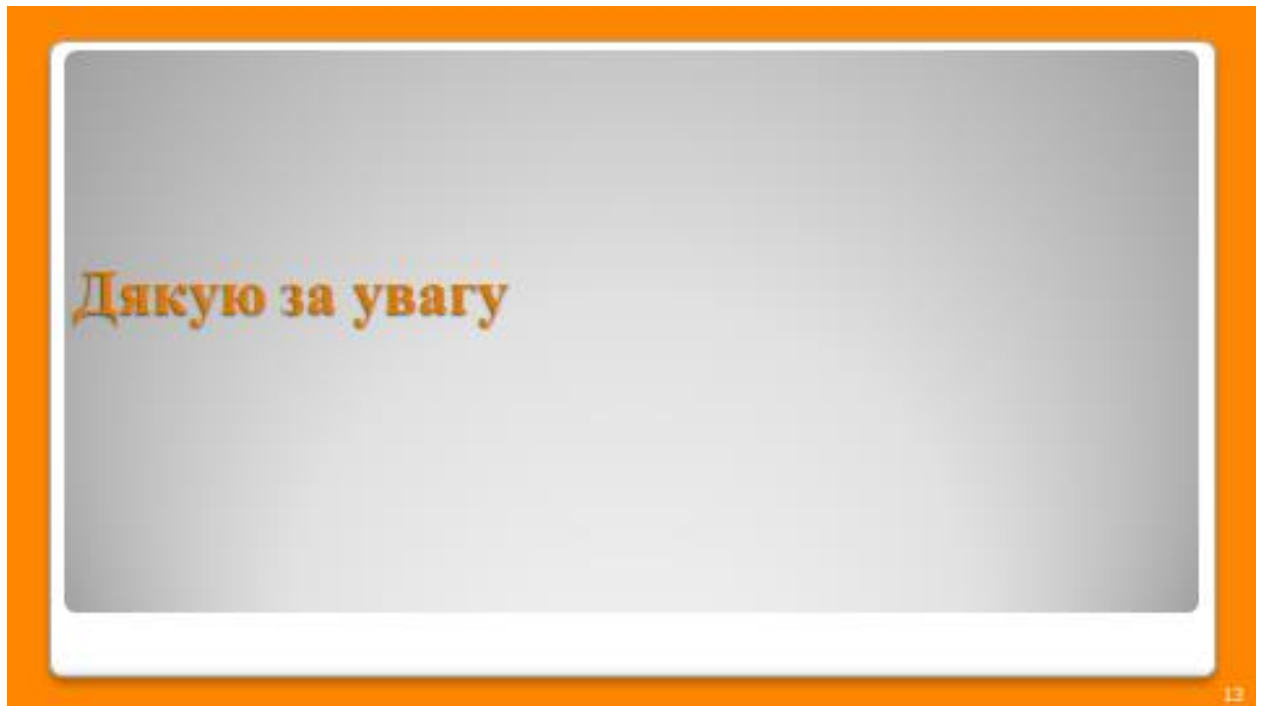


Рисунок Д.13 – Слайд 13