

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**на тему:** «Система забезпечення документообігу для дипломного проектування із використання хмарних сервісів»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології проектування»

**Виконавець роботи:** студентка групи ІТ-62 Александренко Тетяна Володимирівна

**Кваліфікаційна робота бакалавра  
захищена на засіданні ЕК  
з оцінкою**

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 р.

Науковий керівник

\_\_\_\_\_

(підпис)

к.т.н., доц., Парфененко Ю. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

\_\_\_\_\_

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

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

Студент \_\_\_\_\_  
(підпис)

Суми-2020

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук  
Секція інформаційних технологій проектування  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ  
Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик  
« \_\_\_\_\_ » \_\_\_\_\_ 2020 р.

## **З А В Д А Н Н Я** **НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

*Александренко Тетяна Володимирівна*

**1 Тема роботи** Система забезпечення документообігу для дипломного проектування із використання хмарних сервісів

**керівник роботи** Парфененко Юлія Вікторівна, к.т.н., доцент

затверджені наказом по університету від « 14 » травня 2020 р. № 0576-III

**2 Строк подання студентом роботи** «1» червня 2020 р.

**3 Вхідні дані до роботи** документація з дипломного проектування

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** Аналіз предметної області, Проектування інформаційної системи, Розробка інформаційної системи

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** актуальність роботи, мета і задачі дипломного проекту, аналіз систем електронного документообігу, функціональні вимоги до інформаційної системи, структурно-функціональне моделювання процесу документообігу при дипломному проектуванні, засоби реалізації, демонстрація інформаційної системи

## 6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2019

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.1	Ініціювання проекту	до 22.03.20	
1.1.1	Збір даних	до 19.03.20	
1.1.2	Визначення недоліків	до 20.03.20	
1.1.3	Обмеження часових рамок	до 21.03.20	
1.1.4	Ідентифікація проекту	до 22.03.20	
1.2	Планування	до 25.04.19	
1.2.1	Аналіз роботи типових додатків	до 15.04.20	
1.2.2	Визначення вимог до додатку	до 25.03.20	
1.2.3	Визначення інструментарію для реалізації	до 16.04.20	
1.2.4	Планування змісту та структури робіт	до 17.04.20	
1.2.5	Розробка календарного плану	до 18.04.20	
1.2.6	Визначення ресурсів	до 20.04.20	
1.2.7	Визначення ризиків	до 21.04.20	
1.3	Реалізація	до 09.06.20	
1.3.1	Розробка дизайну системи	до 11.04.20	
1.3.2	Розробка UX дизайну	до 08.04.20	
1.3.3	Розробка UI дизайну	до 11.04.20	
1.3.4	Програмування	до 25.05.20	
1.3.5	Створення бази даних та підключення до сайту	до 03.05.20	
1.4	Завершення	до 09.06.20	
1.4.1	Створення документації	до 01.06.20	
1.4.2	Запуск і підтримка	до 06.06.20	
14.3	Архівація проекту	до 08.06.20	

Студент

\_\_\_\_\_

(підпис)

Александренко Т.В.

Керівник роботи

\_\_\_\_\_

(підпис)

к.т.н., доц. Парфененко Ю.В.

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Система забезпечення документообігу для дипломного проектування із використання хмарних сервісів».

Пояснювальна записка містить вступ, 3 розділи, висновки, додатки та список літератури, включає 89 сторінок, 2 таблиці, 40 ілюстрацій, 20 джерел.

В першому розділі проводиться огляд літературних джерел, присвячених вирішенню питань, близьких до поставленої мети дипломного проектування. Проводиться огляд існуючих програмних продуктів, що можуть бути використані для вирішення поставленої задачі. На основі проведеного аналізу предметної області деталізуються мета роботи та задачі роботи із зазначенням методів та технології вирішення задач.

Другий розділ призначений для опису поетапного проектування інформаційної системи. В даному розділі наведений структурно-функціональний аналіз проекту, описані основні принципи роботи розробленої інформаційної системи.

У третьому розділі наводиться архітектура системи. Описуються основні етапи створення програмного додатку, бази даних із зазначенням засобів реалізації. Наводяться приклади використання програмного продукту, що демонструють його працездатність.

Результатом проведеної роботи є розроблена інформаційна система, яка надає можливість більш зручної віддаленої роботи з студентами і можливість стежити за ходом дипломного проектування завідувачу кафедри.

Ключові слова: інформаційна система, документообіг, хмарні технології, дипломне проектування, завідувач кафедри, прикладний програмний інтерфейс(API).

# ЗМІСТ

ВСТУП .....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій .....	8
1.2 Аналіз програмних продуктів для автоматизації документообігу.....	9
1.3 Постановка задачі .....	13
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	15
2.1 Структурно-функціональне моделювання процесу документообігу при дипломному проектуванні .....	15
2.2 Моделювання варіантів використання.....	18
2.3 Проектування схеми бази даних .....	20
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	24
3.1 Архітектура інформаційної системи .....	24
3.2 Програмна реалізація .....	25
3.3 Використання програмного додатку .....	36
ВИСНОВКИ .....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	47
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ .....	49
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ .....	61
ДОДАТОК В. ЛІСТИНГ ПРОГРАМНОГО КОДУ .....	71

## ВСТУП

Нині вирішення проблеми електронного дипломного документообігу залишається в значній мірі технологічно недосконалим. Обмін документами, такими як пояснювальна записка, протокол перевірки роботи на плагіат, файли презентації, архів з програмним додатком між студентами, викладачами та секретарем екзаменаційної комісії здійснюється у месенжерах, електронною поштою, і надсилають або файл, або посилання на папку на гугл-диску. Такий підхід не відповідає вимогам до формалізованих технологічних процесів в проектуванні і в значній мірі ускладнює керованість даними (облік даних, зберігання, аналіз, пошук, повторне використання).

Світова практика демонструє, що все більше зарубіжних архівів переходить на зберігання електронних документів із застосуванням хмарних систем. При використанні хмарних технологій в навчально-методичній діяльності педагога можна привести багато прикладів використання – це організація планування, електронні щоденники, журнали, особисті кабінети для студентів і викладачів, тематичні форуми, конференції, дистанційні прийомні освітні установи. Застосування хмарних технологій сприяє вдосконаленню освітнього процесу, підвищенню ефективності педагогічної праці, поліпшення відмінності знань, умінь і навиків студентів.

Мета даної роботи полягає у розробці інформаційної системи, яка б дозволила надати можливість більш зручної віддаленої роботи з студентами і можливість стежити за ходом дипломного проектування завідувачу кафедри.

Для досягнення цієї мети треба вирішити наступні задачі:

- вивчити основні характеристики систем електронного документообігу для визначення основних вимог до інформаційної системи та планування робіт і на основі аналізу існуючих аналогів-систем розробити технічне завдання;

- обрати засоби реалізації для розроблення та розміщення системи в мережі Інтернет;
- спроектувати базу даних, розробити прототип інтерфейсу та сформувати моделі інформаційної системи;
- реалізувати інформаційну систему для забезпечення організаційного процесу дипломного проектування, підвищення якості індивідуальної роботи, заповнити контент та продемонструвати її роботу.

Практичне значення – забезпечення організаційного процесу дипломного проектування, скорочення часу на відповідність друкованої роботи вимогам стандарту, підвищення якості індивідуальної роботи і автоматизації складання і зміни списку дипломних робіт.

Результати роботи доповідалися на конференції «ІМА - 2020». За матеріалами роботи опубліковано тези [1].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень і публікацій

Огляд друкованих й електронних наукових видань показав, що питання організації процесу документообігу для дипломного проектування із використанням хмарних сервісів є актуальним і дискусійним.

Тематиці електронного документообігу присвячено чимало наукових досліджень. Публікації аналізують організацію структури систем, надають головні вимоги і подають найважливіші завдання систем електронного документообігу (СЕД). Формують абстрактні принципи побудови і роботи інтегрованої системи та бар'єри, які виникають у процесі їх установки [2 - 4].

В останні десятиліття застосування хмарних технологій набуло важливої актуальності в зв'язку з покращенням програмного забезпечення та персональних комп'ютерів [5 - 7]. Тому, застосування хмарних сервісів у навчальному процесі досліджувало багато науковців. Серед них Морзе Н.В. , що вивчав головні педагогічні концепції використання хмарних обчислень у навчальних методах навчання. Воронкін О. С. звернув увагу на побудову приватних навчальних середовищ на основі хмарних технологій. Також Триус Ю.В. вивчав хмарні технології для студента для його професійної підготовки. С.О. Семеріков переглядав хмарні засоби навчання комп'ютерного моделювання фізичних методів.

Тематиці документообігу при дипломному проектуванні присвячено методичні вказівки для студентів [8]. У цьому методичному посібнику автори, вперше спробували адаптувати систему реального документообігу до процесу навчального дипломного проектування в вузі.

Водночас варто зазначити, що систематичних досліджень процесу документообігу для дипломного проектування із використання хмарних сервісів в закладах вищої освіти недостатньо.



## 1.2 Аналіз програмних продуктів для автоматизації документообігу

На поточний час існує безліч СЕД. Порівняємо деякі системи, представлені на ринку: «Directum» (компанія «Directum») [9], «Escom.doc» (компанія «Евроменеджмент») [10], «Optima-WorkFlow» (компанія «UpScale Soft») [11], «Гран-Док» (ВАТ «Науково-виробниче підприємство «Граніт-Центр») [12], «PayDox. Електронний документообіг і колективна робота» (компанія «Paybot LLC») [13]. З технічної точки зору, предметна область, яку автоматизують засоби документообігу, вкрай проста. Якщо дивитися далі, будь-яка система документообігу - це лише сховище неструктурованих документів. Тому всі системи без винятку реалізують такі функції і можуть бути охарактеризовані за такими основними параметрами [14 - 15]:

1. можливості картки документа,
2. забезпечення безпеки,
3. можливість зберігання версій,
4. пошук по сховищу документів,
5. різні повідомлень в ході процесу обробки документа,
6. маршрути і завдання користувачам,
7. інтеграція з електронною поштою,
8. можливість архівації застарілих документів,
9. розподілена структура сховища,
10. зручний інтерфейс,
11. сканування і розпізнавання документів,

Для простоти порівняння кожної функції продуктів проставимо оцінки за п'ятибальною шкалою в табл. 1.1.

Таблиця 1.1 - Результати аналізу функціональних можливостей деяких систем електронного документообігу

<b>№ критерію оцінки функціональних вимог</b>	<b>Directum</b>	<b>Escom.doc</b>	<b>Optima-WorkFlow</b>	<b>PayDox</b>	<b>«Гран-Док»</b>
1.	5	5	5	4	4
2.	4	5	5	4	5
3.	5	4	5	5	4
4.	5	5	5	4	5
5.	3	4	5	4	4
6.	5	4	5	4	1
7.	5	3	4	4	4
8.	4	3	4	4	1
9.	5	-	4	2	3
10.	5	3	5	3	5
11.	5	3	3	2	3
<b>Підсумковий середній бал</b>	<b>4,6</b>	<b>3,5</b>	<b>4,5</b>	<b>3,6</b>	<b>3,5</b>

Крім функціональних можливостей розглянемо ціни розглянутих СЕД [16]. Більшість з них занадто дорогі, а процес їх впровадження виявляється складним та тривалим і окупається не скоро.

Максимальний бал набрала система «Directum. Це розвинена і самодостатня система. Володіє широкою функціональністю і простими принципами роботи. Перевагою продукту є розширена реалізація механізму електронного цифрового підпису. Висока масштабованість системи може забезпечити роботу великої кількості користувачів. Є розвинений інструментарій (IS-Builder), що дозволяє швидко адаптувати систему, інтегрувати її з іншими системами. Продукт порівняно дорогий і по співвідношенню «ціна-якість». Для вищого навчального закладу функціонал, швидше за все, виявиться надмірним.

Незначно менше балів у програми «Optima-WorkFlow». Документообіг - всього лише одне із застосувань даної системи. В цілому досить дорога система з урахуванням вартості користування. Підходить великим організаціям, які хочуть не тільки впровадити документообіг, а й вирішити інші завдання автоматизації інформаційних потоків. Впровадження системи вимагає досить широких знань, починаючи від структур баз даних. Потрібен штат сильних ІТ-фахівців. Тільки в цьому випадку покупка системи виправдає очікування.

«PayDox» - цікава і досить нова пропозиція на ринку. Занадто вартісний для WEB-додатку, особливо з огляду на технічну підтримку. Відсутність GUI-інтерфейсу значно скорочує можливості по інтеграції з клієнтським програмним забезпеченням. Обрати систему можна для простих випадків, коли документів мало і процеси нескладні. Програма «Escom.doc» позиціонується як готове рішення. А тому, містить мінімум можливостей щодо адаптації вбудованими засобами. Співробітникам доведеться пристосовуватися до цієї системи. Система підходить невеликим підприємствам, які готові перекроювати свої бізнес-процеси під можливості програми. І нарешті «Гран-Док», багато в чому порівнянна за своїми можливостями з «Escom.doc». Це просте рішення, яке в основному буде служити не чим іншим як централізованим сховищем документів.

Отже, було виконано порівняння досить популярних СЕД. У зв'язку з цим можна зробити висновок, що лідерство за кількістю згадок тих чи інших продуктів в засобах масової інформації і Інтернеті не завжди є репрезентативним відображенням якості і масштабу функціональних характеристик. Було виявлено ряд мінусів впровадження таких систем для підтримки дипломного проектування:

1. висока ціна ліцензій;
2. надмірний функціонал;
3. потрібен штат висококваліфікованих ІТ-фахівців;
4. додаткові витрати на впровадження систем;
5. навчання персоналу в рамках впровадження.

Тому через вище перелічені проблеми дані СЕД для поставленої задачі не підходять.

В Сумському державному університеті розроблена власна платформа «Міх» (рис.1.1). Вона інтегрує всі навчальні матеріали університету, до яких мають вільний доступ з будь-якого пристрою всі здобувачі вищої освіти, які зареєструвались на даній платформі [17].

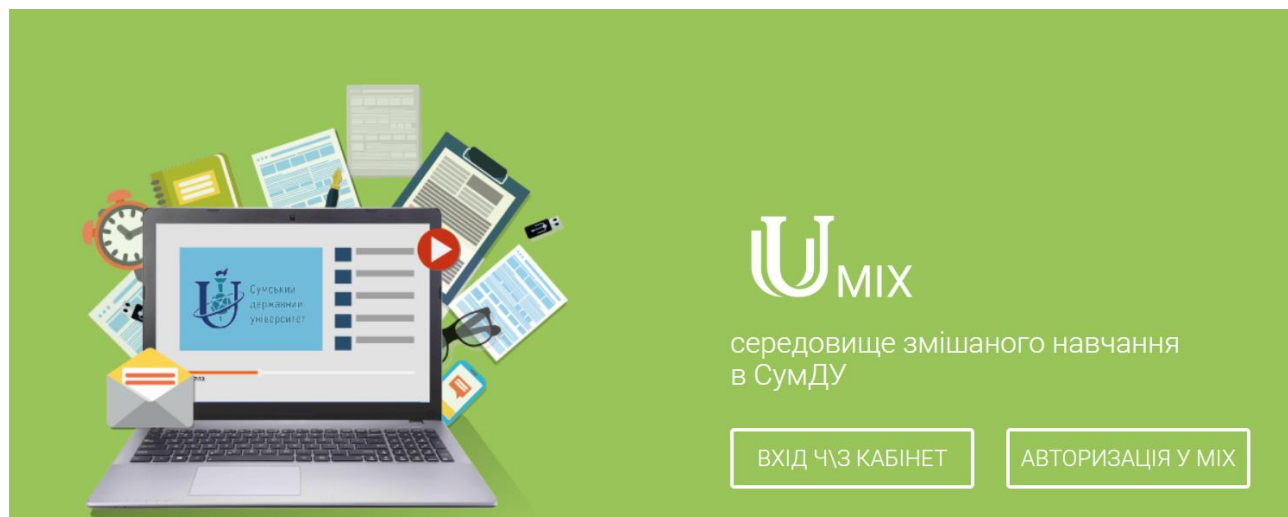


Рисунок 1.1 - Головна сторінка платформи «Міх»

Ресурсом «Міх» передбачено такі ролі для користувачів: адміністратор, студент та викладач (автор). Всі особи після реєстрації отримують автоматично роль студента. Для отримання ролі автора користувач звертається до навчально-методичної лабораторії електронного навчання організаційно-методичного центру технологій електронного навчання (ОМЦТЕН). Після цього у викладача з'явиться розділ «Розробка курсів», що дає йому змогу створювати дисципліни та імпортувати навчальні об'єкти. Таким чином автор може розробити курс «Дипломне проектування» і вести обмін необхідними документами зі студентом, але система не може враховувати повністю цикл дипломного проектування.

Тому було прийнято рішення про розроблення власної інформаційної системи, яка б враховувала повний життєвий цикл документообігу та формувала зведену таблицю по стану виконання робіт студентів. При створенні власної системи необхідно відмовитися від придбання нового дорогого серверного обладнання, яке необхідне для підтримки програми. Тому простим рішенням цієї проблеми є використання хмарних сховищ. Найбільш популярними є «Dropbox» та

«GoogleDrive». Функціональність цих сервісів може бути інтегрована в інформаційну систему через інтерфейс прикладного програмування (API).

### 1.3 Постановка задачі

На сьогоднішній день вирішення проблеми електронного дипломного документообігу залишається в значній мірі технологічно недосконалим. Обмін документами, такими як пояснювальна записка, протокол перевірки роботи на плагіат, файли презентації, архів з програмним додатком між студентами, викладачами та секретарем екзаменаційної комісії (ЕК) здійснюється у месенжерах, електронною поштою, і надсилають або файл, або посилання на папку на гугл-диску. Такий підхід не відповідає вимогам до формалізованих технологічних процесів в проектуванні і ускладнює керуваність даними. Також є потреба у відслідковуванні основних етапів документообігу секретарем ЕК та завідувачим секцією, тобто перегляд зведено статистики, яким студентам вже керівники відповіли, що їх робота є завершеною, і її можна надсилати на перевірку на плагіат.

Отже за результатами аналізу предметної області була сформульована мета дипломної роботи: розробити хмарний клієнт-серверний додаток забезпечення документообігу для дипломного проектування.

Для зберігання файлів використовувати хмарне сховище «Dropbox». Клієнтська частина інформаційної системи повинна створена з використанням web-технологій HTML, CSS та JavaScript [18]. Серверна частина та логіка системи повинні використовувати мову PHP, базу даних MySQL. Така серверна частина повинна визначати порядок взаємодії між користувачами, надавати доступ до хмарного середовища та його обчислювальних потужностей, сховищ зберігання даних, представляти доступ до публічних даних та методів API для інтеграції з клієнтською частиною, забезпечувати поєднання зовнішніх компонентів і модулів.

Використання інформаційної системи спрямоване на скорочення часу на відповідність друкованої роботи вимогам стандарту, підвищення якості індивідуальної роботи і автоматизації складання і зміни списку дипломних робіт. Система повинна надавати можливість більш зручної віддаленої роботи з студентами і можливість стежити за ходом дипломного проектування завідувачу кафедри. Результати виконання проекту можуть бути корисні для застосування на кафедрах університету.

Впровадження інформаційної системи передбачає такі результати:

- Завантаження документів формату .doc, .pdf та .ppt з фіксацією дати завантаження та версії документа в відповіді розділи.
- Зміна статусу документа, відповідно для кожної ролі власні статуси.
- Можливість перегляду контактних даних користувача.
- Перегляд зведеної статистики студентів, яких робота є завершеною.
- Доступ до зведеної таблиці з інформацією, в якому статусі знаходиться остання версія документа для кожного студента.

Повна та детальна інформація про вимоги представлена в технічному завданні, яке було розміщене у додатку А. Планування робіт представлено у додатку Б.

## 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Структурно-функціональне моделювання процесу документообігу при дипломному проектуванні

Стадія формування вимог до інформаційної системи є однією з найважливіших стадій, що передують процес проектування. Одним з етапів даної стадії є проведення обстеження об'єкта, що автоматизується і побудова моделей його діяльності. Існують два види моделей - модель «As Is» і модель «To Be»[19]. Кожна з них відображає як інформаційний, так і функціональний аспект діяльності організації.

Методологія структурного аналізу і проектування (SADT) розроблена для побудови функціональних моделей складних систем. На її основі і з ініціативи був створений стандарт IDEF0. Головний принцип методології IDEF0 - представлення системи у вигляді набору взаємопов'язаних блоків, що відображають функції, операції, процеси.

Структурно-функціональну модель організації процесу документообігу при дипломному проектуванні з використанням розробленого додатку було побудовано в програмному продукті Erwin Process Modeler. Контекстна діаграма моделі зображена на рис. 2.1. Ця діаграма описує модель процесу організації документообігу при дипломному проектуванні в узагальненому вигляді.

В центрі діаграми знаходиться основний блок діяльності. Для нашої задачі це організація процесу документообігу при дипломному проектуванні. Діаграма побудована узагальнено для різних типів документів, серед яких – пояснювальна записка, результат перевірки роботи на плагіат, файли презентації, архів з програмним додатком, відгук наукового керівника, рецензія на дипломну роботу і т. д.

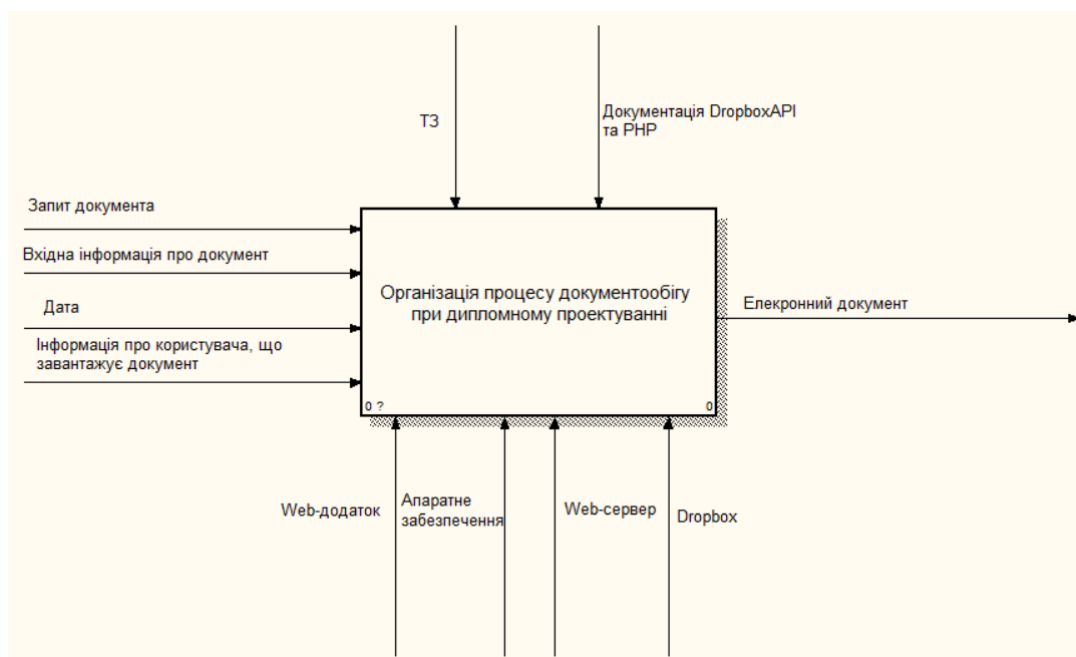


Рисунок 2.1 – Контекстна діаграма IDEF0

В ролі вхідних даних є:

- запит документа;
- вхідна інформація про документ;
- дата;
- інформація про користувача, що завантажує документ.

Основними механізмами за допомогою яких буде виконуватись процес є сам web-додаток, що виконує документообіг, апаратне забезпечення на якому та за допомогою якого виконуються обчислення, web-сервер, що очікує запити користувачів і виконує їх обробку та хмарне сховище Dropbox, в якому будуть зберігатися документи.

Організація процесу документообігу керується поставленим технічним завданням та документацією DropboxAPI та PHP за допомогою яких будується серверна частина додатку.

Контекстна діаграма декомпозується на діаграми нижчого рівня для того аби показати внутрішні процеси, що проходять в ній. Всі дуги та зв'язки з батьківської діаграми переносяться на діаграму декомпозиції.

Головна батьківська діаграма декомпозується на наступні блоки, вона представлена на рисунку 2.2:



- авторизація/реєстрація;
- робота з документами;
- перегляд звітної інформації.

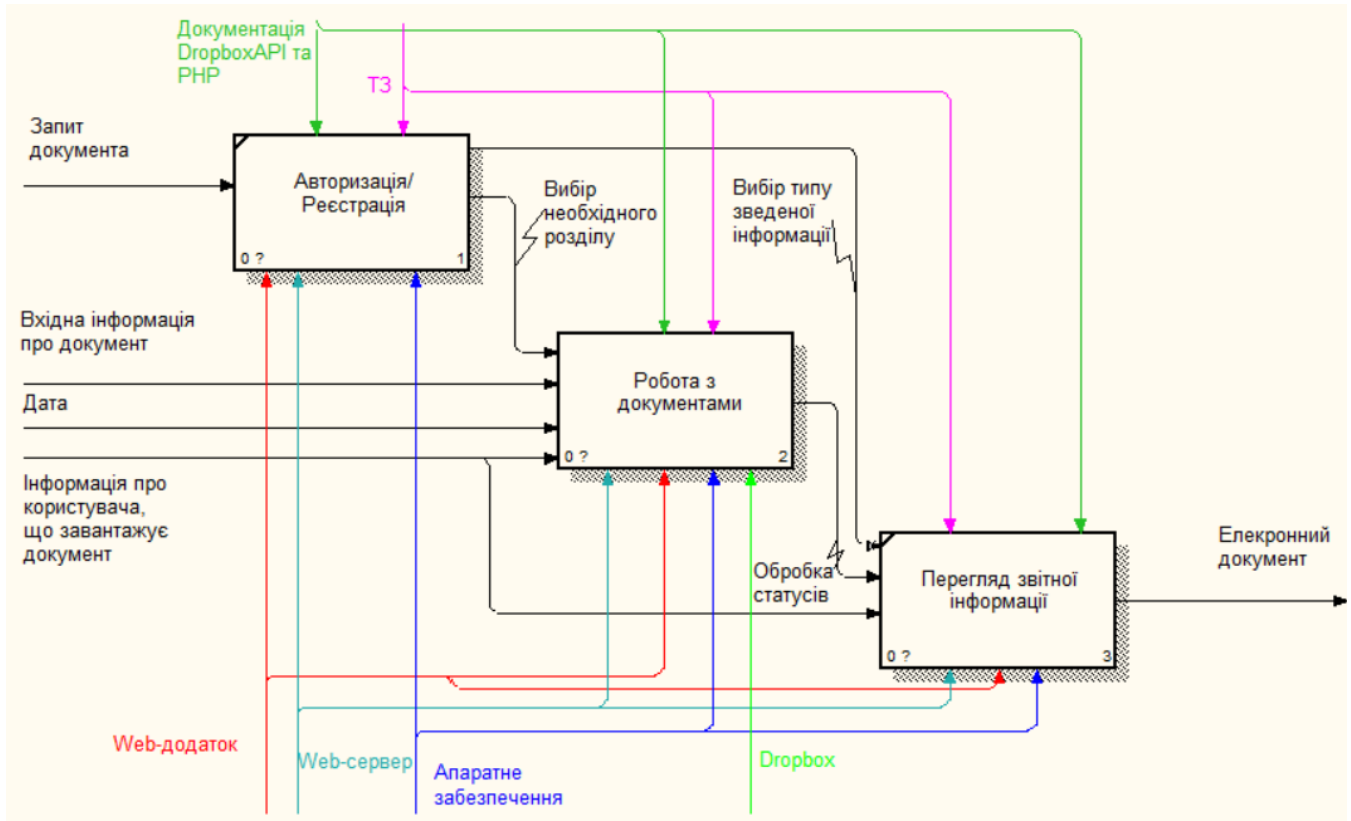


Рисунок 2.2 – Діаграма декомпозиції IDEF0 (перший рівень)

Вхідними даними до діяльності «авторизація/реєстрація» є запит документа. Для «роботи з документами» вхідними даними є: вхідна інформація про документ, дата, інформація про користувача, що завантажує документ. Вхідними даними до діяльності «перегляд звітної інформації» є інформація про користувача, що завантажує документ. Стрілками контролю для діяльностей «авторизація/реєстрація», «робота з документами» та «перегляд звітної інформації» – «Технічне завдання» та «Документація DropboxAPI та PHP». Стрілками механізмів для діяльностей «авторизація/реєстрація», «робота з документами» та «перегляд звітної інформації» – «Апаратне забезпечення», «Web-додаток» та «Web-сервер», а для діяльності «Робота з документами» – «Dropbox».

Для розуміння процесів що відбуваються в системі декомповано діяльність «Робота з документами». Діаграма декомпозиції показана на рис 2.3.

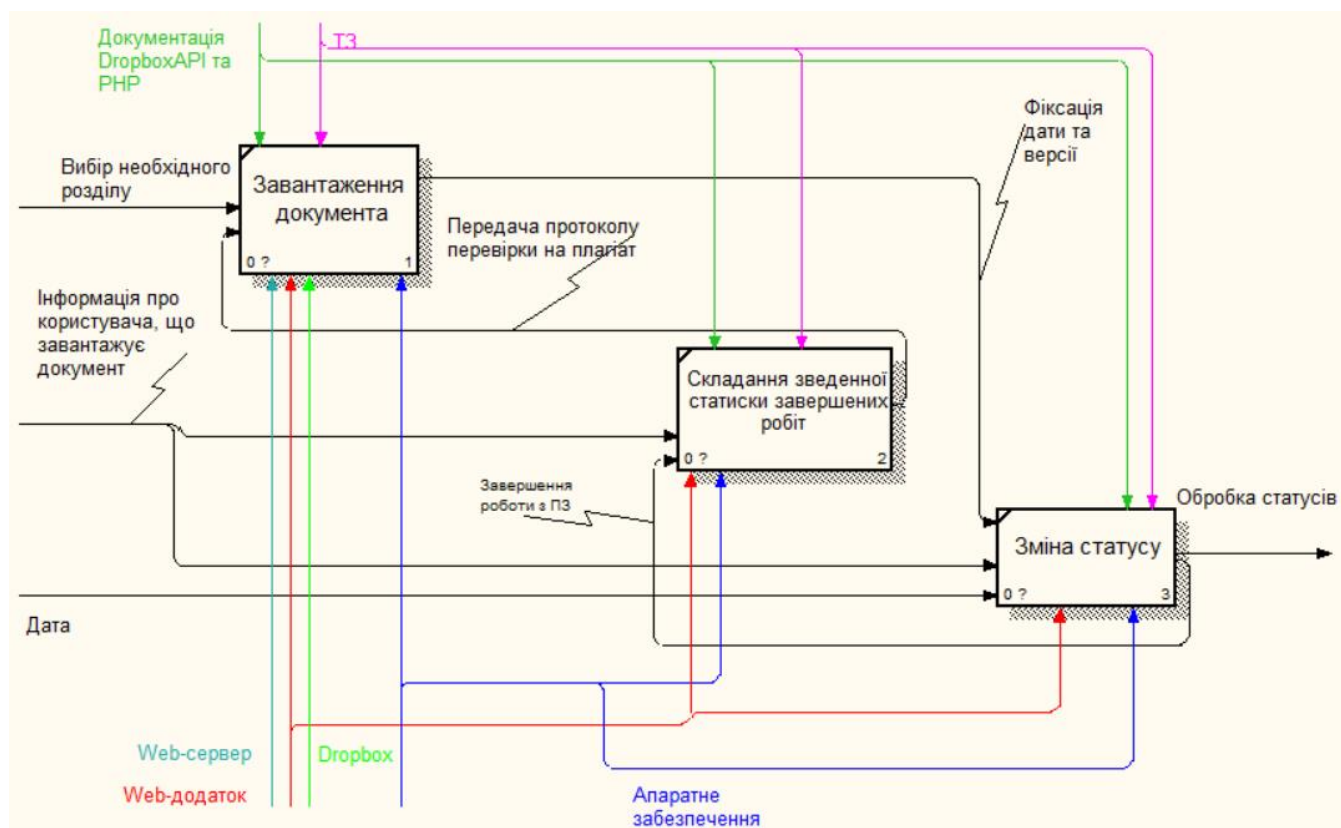


Рисунок 2.3 – Діаграма декомпозиції IDEF0 (другий рівень)

Було виконано аналіз всіх процесів, що проходять в системі та показано їх на діаграмах. Проведена декомпозиція для більш точного розуміння зв'язків між процесами.

## 2.2 Моделювання варіантів використання

Уніфікована мова моделювання (UML) є стандартним інструментом для створення схем програмного забезпечення [20]. За допомогою UML можна конструювати, візуалізувати, уточнити і документувати артефакти програмних систем. Діаграма в UML представляє набір елементів, що використовується для

візуалізації системи з різних точок зору. Одним з видів UML діаграм є діаграма варіантів використання.

Діаграма варіантів використання – це концептуальне уявлення системи в процесі її розробки. Вона складається з варіантів використання, акторів, та відносин між ними. Варіанти використання це описання послідовності дій які робить система у відповідь на діяльність користувачів, або інших програмних систем. Вони відображають функціональність системи.

Діаграму варіантів використання веб-додатку представлена на рис.2.4.

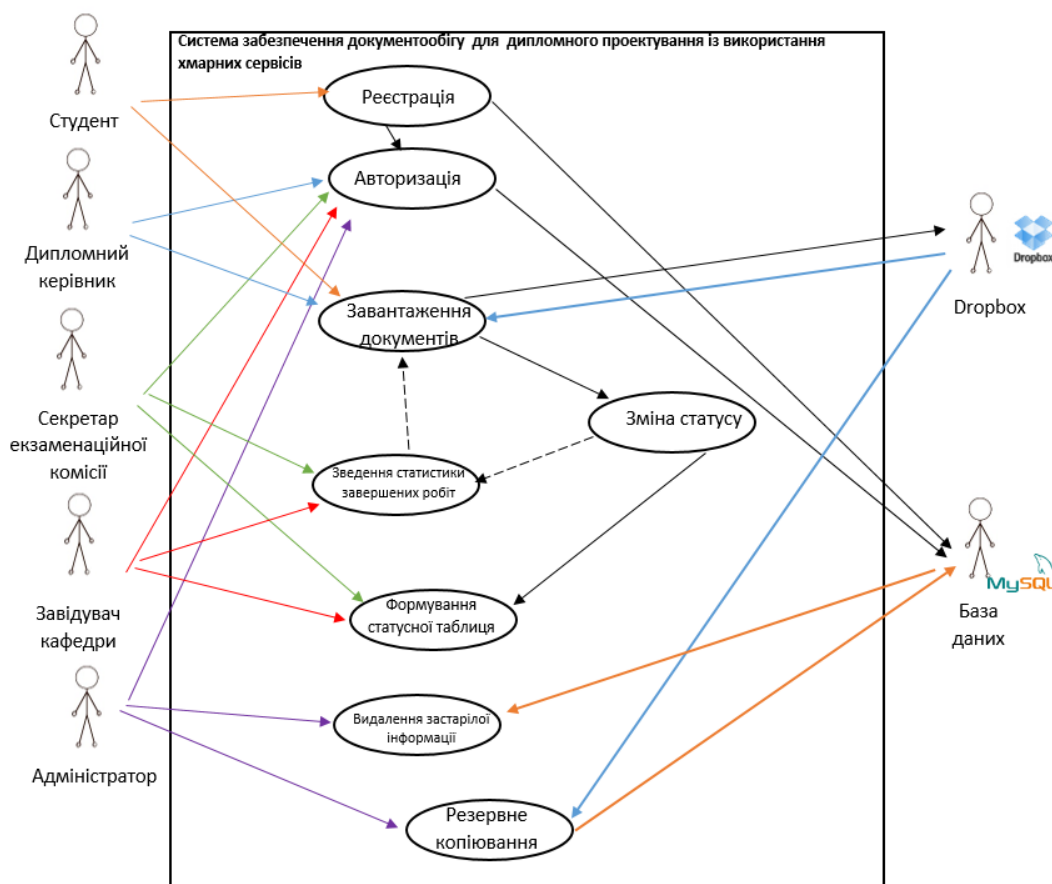


Рисунок 2.4 - Діаграма варіантів використання

Актори :

- студент – виконавець дипломної роботи;
- дипломний керівник – перевіряє дипломні роботи;
- секретар екзаменаційної комісії – надає протокол перевірки на плагіат;

- завідувач кафедри – перевіряє стан роботи студентів;
- адміністратор – редагує вміст сторінок для кожної ролі;
- Dropbox – хмарне сховище для зберігання документів;
- база даних – сховище для зберігання даних про користувачів.

Варіанти використання (ВВ) :

- 1) ВВ Авторизація - ВВ авторизація користувачів.
- 2) ВВ Реєстрація - ВВ реєстр студентів в системі.
- 3) ВВ Завантаження документів – ВВ завантаження документів (ПЗ, презентація, архів додатку, протокол перевірки на плагіат та інш.).
- 4) ВВ Зміна статусу – ВВ зміни статусу документів.
- 5) ВВ Статистика завершених робіт – ВВ зведеної статистики по стану зданих дипломних робіт
- 6) ВВ Статусна таблиця – ВВ зведеної таблиці з інформацією, в якому статусі знаходиться остання версія документа для кожного студента.
- 7) ВВ Резервне копіювання – ВВ резервного копіювання даних кожного місяця.
- 8) ВВ Видалення застарілої інформації – ВВ після закінчення терміну давності - знищення копій інформації і особистої інформації студентів.

### 2.3 Проектування схеми бази даних

У реальному проектуванні структури бази даних застосовується метод - семантичне моделювання. Семантичне моделювання являє собою моделювання структури даних, спираючись на зміст цих даних. Як інструмент семантичного моделювання використовуються різні варіанти діаграм сутність-зв'язок (ER - Entity-Relationship).

При розробці ER-моделей ми повинні отримати наступну інформацію про предметну область:

- сутності предметної області.
- атрибути сутностей.
- взаємозв'язки між сутностями.

ER-діаграми зручні тим, що процес виділення сутностей, атрибутів і зв'язків є ітераційним. Розробивши перший наближений варіант діаграм, ми уточнюємо їх, опитуючи експертів предметної області. При цьому документацією, в якій фіксуються результати бесід, є самі ER-діаграми.

Головними сутностями бази даних є:

- Користувачі.
- Групи.
- Підтвердження користувачів.
- Файли користувачів.
- Типи документів.
- Статус документів.
- Роль користувача.

Кожна із даних сутностей містить набір атрибутів, який представлений у табл. 2.1.

Таблиця 2.1 – Таблиця опису сутностей та їх атрибутів

Назва таблиці	Назва сутності	Назва поля	Опис атрибуту
confirm_users	Інформація про підтвердження користувачів	id_confirm	Id підтвердження користувача
		email	Пошта користувача
		token	Токен користувач з Dropbox
		date_registration	Дата реєстрації
groups	Інформація про групи	id_group	Id групи
		name_group	Найменування групи

Продовження таблиці 2.1 – Таблиця опису сутностей та їх атрибутів

Назва таблиці	Назва сутності	Назва поля	Опис атрибуту
users	Інформація про користувача	id_user	Id користувача
		id_group	Id групи
		role_id	Id ролі користувача
		id_confirm	Id підтвердження користувача
		first_name	Ім'я користувача
		last_name	Прізвище користувача
		middle_name	По-батькові користувача
		id_teacher	Id викладача
		theme	Тема диплома
		avatar	Шлях до фото профілю користувача
		email	Пошта користувача
		email_status	Статус підтвердження пошти (0 – не підтверджений, 1- підтверджений)
		password	Пароль користувача
		reset_password_token	Токен для зміни паролю
date_registration	Дата реєстрації користувача		
roles	Інформація про ролі користувачів	role_id	Id ролі
		role	Найменування ролі
files	Інформація про файли користувачів	id_file	Id файлу
		doc_id	Id документу
		id_user	Id користувача
		status_id	Id статусу
		name	Найменування файлу
		size	Розмір файлу
		date	Дата завантаження файлу
		version	Версія файлу
document	Інформація про тип документу	doc_id	Id документу
		name_doc	Найменування документу
status	Інформація про статус файлу	status_id	Id статусу
		role_id	Id ролі користувача
		name_status	Найменування статусу

На основі даних сутностей та їх атрибутів можна розробити діаграму типу «сутність-зв'язок, яка представлена на рис. 2.5.

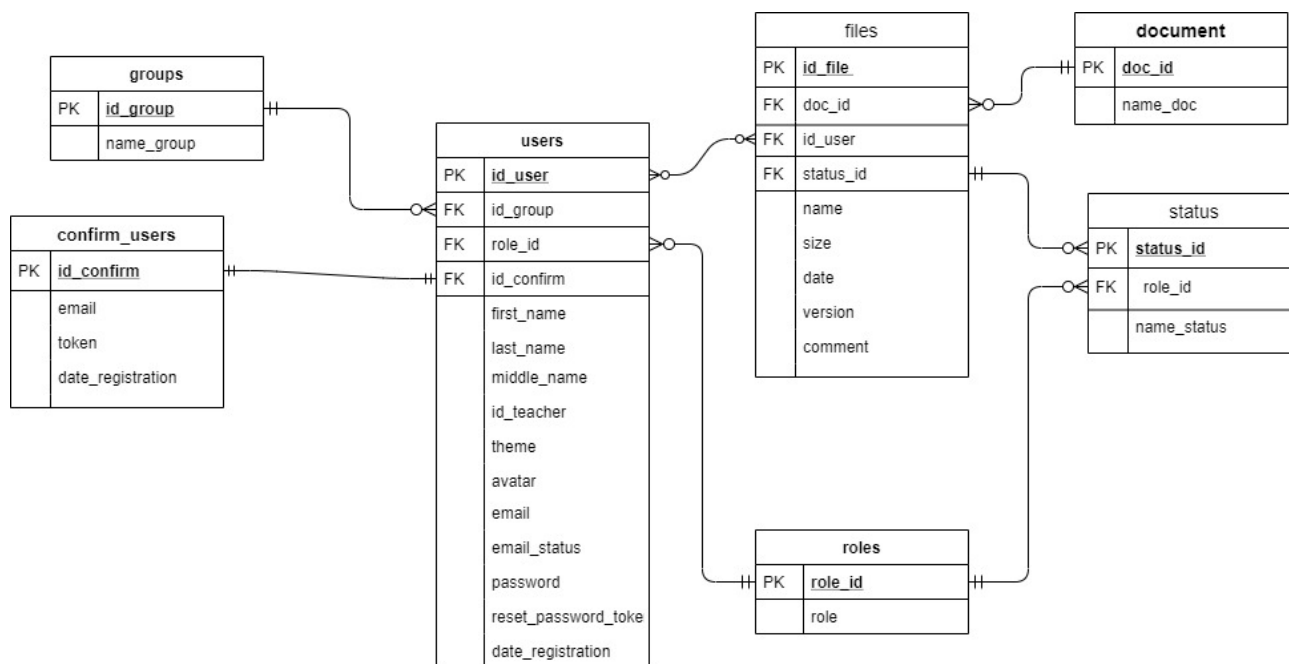


Рисунок 2.5 – Діаграма бази даних

На діаграмі представлені зв'язки між сутностями. Між таблицею з інформацією про користувача (users) встановлений зв'язок з таблицею що містить інформацію про його файли (files). Між ними встановлено зв'язок багато і тільки багато. Такі зв'язки були використані через те що декілька користувачів можуть зайти у додаток та можуть звернутися до талиці з файлами багато разів або не звернутися взагалі.

Між таблицями з інформацією про користувачів (users) та групи(groups) встановлено зв'язок один – багато через те що лише студент може обрати одну групу, але в групі може бути багато студентів.

Між таблицею з користувачами та таблицями що містять інформацію про підтвердження пошти користувача встановлено зв'язки один – один, тому що оди користувач має одне підтвердження своєї пошти.

## 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1 Архітектура інформаційної системи

Архітектура розробленого сайту є простою і інтуїтивно зручною і складається (як показано на рисунку 3.1) з :

- клієнтської частини;
- серверної частини.

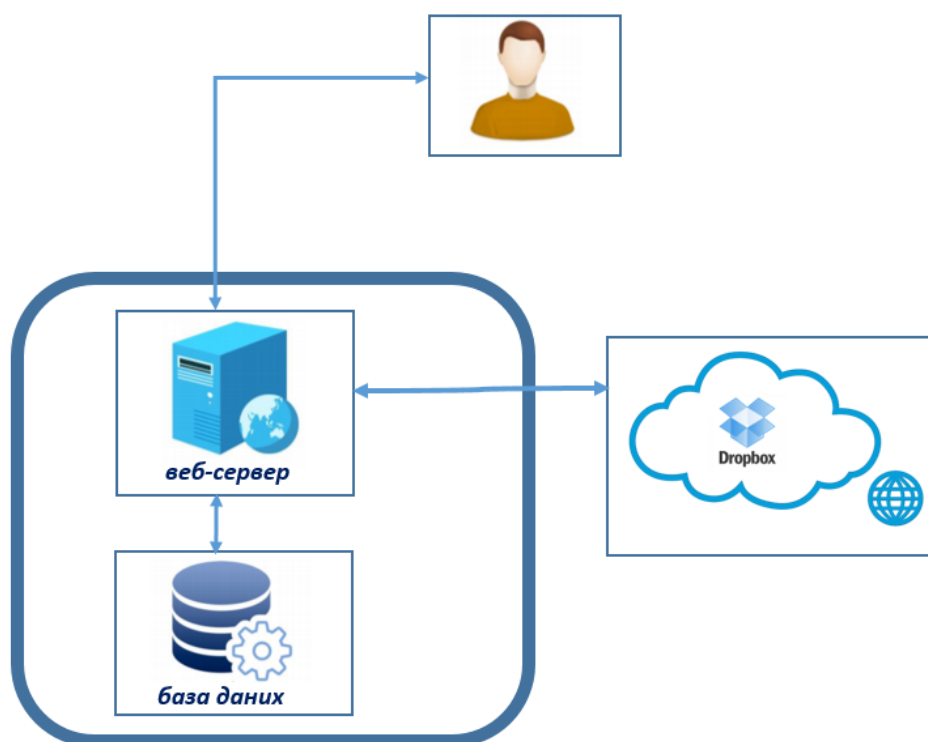


Рисунок 3.1 - Архітектура системи

Серверна частина архітектури сайту розглядається як взаємозв'язок операційної і програмної частини. В операційній частині розглядається середовище розробки сайту, а саме Brackets.

У програмній частині архітектури розглядається робота сайту в мережі Інтернет, взаємодія програмного забезпечення магазину і сервісів, що надаються власниками серверів, провайдерів. Так як після виготовлення сайту необхідно буде



його розмістити в мережі Інтернет, то доведеться зіткнутися з вибором місць розміщення, інакше кажучи, вибрати хостинг.

У клієнтській частині архітектури розроблено максимально зручна і доступна робота. Адміністрування містить інструменти управління сайтом. Так як розроблений сайт планується використовуватися як інформаційна система, то він включає в себе як загальні установки, так і спеціальні настройки.

Така серверна частина визначає порядок взаємодії між користувачами, надає доступ до хмарного середовища та його обчислювальних потужностей, сховищ зберігання даних, представляє доступ до публічних даних та методів API для інтеграції з клієнтською частиною, забезпечує поєднання зовнішніх компонентів і модулів.

### 3.2 Програмна реалізація

Перед початком розробки необхідно авторизуватися на сайті Dropbox. Для поставленої задачі було використано безкоштовну бібліотеку Dropbox SDK версії 2.0. Для використання бібліотеки необхідно вміти працювати з Composer та встановлений на комп'ютері PHP версії 7.0. Сама установка пакета для php робиться простою командою:

```
$ php composer require kunalvarma05/dropbox-php-sdk.
```

Отже, необхідно створити тестовий додаток на сайті Dropbox. Для цього переходимо за посиланням <https://www.dropbox.com/developers/apps> і натискаємо на кнопку Create App. Далі прямуємо запропонованим варіантам і заповнюємо необхідні поля. В результаті отримуємо App key, App secret і Access Token. А кож необхідно заповнити поле Redirect URIs. Сайт розташовується на локальному сервері за адресою localhost (після перенесення сайту на хост він зміниться). На рис. 3.2 та рис. 3.3 наведені пояснення.

Create a new app on the DBX Platform

### 1. Choose an API

<input checked="" type="radio"/> <b>Dropbox API</b> For apps that need to access files in Dropbox. <a href="#">Learn more</a>	<input type="radio"/> <b>Dropbox Business API</b> For apps that need access to Dropbox Business team info. <a href="#">Learn more</a>
----------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------

### 2. Choose the type of access you need

[Learn more about access types](#)

<input type="radio"/> App folder – Access to a single folder created specifically for your app.
<input checked="" type="radio"/> Full Dropbox – Access to all files and folders in a user's Dropbox.

### 3. Name your app

Create app

Рисунок 3.2 – Створення додатку в Dropbox

Settings	Branding	Analytics
Status	Development	<a href="#">Apply for production</a>
Development users	Only you	<a href="#">Enable additional users</a>
Permission type	Full Dropbox <a href="#">?</a>	
App key	tb5xrifyj8xi4ks	
App secret	98iz6k4tgs0vxu5	
OAuth 2	<b>Redirect URIs</b> <input type="text" value="http://localhost"/> <a href="#">×</a> <input type="text" value="https:// (http allowed for localhost)"/> <a href="#">Add</a>	
	<b>Allow implicit grant</b> <a href="#">?</a> <input type="text" value="Allow"/>	
	<b>Generated access token</b> <a href="#">?</a> <input type="button" value="Generate"/>	
Chooser/Saver domains	<input type="text" value="example.com"/> <a href="#">Add</a> <small>If using the Chooser or the Saver on a website, the domain of that site.</small>	
Webhooks	<b>Webhook URIs</b> <a href="#">?</a> <input type="text" value="https://"/> <a href="#">Add</a>	
Delete app	<input type="button" value="Delete app"/>	

Рисунок 3.3 – Налаштування додатку в Dropbox

Для зручності всю логіку додатку розміщено в файлі `dropbox.php`, код представлений в додатку В. Рекурсивна функція `makeTree ()` призначена для створення папок користувачів, в назву папки записується електронна пошта користувача (рис. 3.4). Масив містить такі поля як назва, розмір файлу, дата завантаження, яке заносяться до бази даних.

Функція `printTree ()` призначена для формування посилань на обраний файл користувача для його завантаження.

```
if ($node['type'] == 'folder') {
$url = 'https://www.dropbox.com/home' . $node['path'];
}
if ($node['type'] == 'file') {
$url = 'https://www.dropbox.com/preview' . $node['path'] . '?role=personal';
}
```

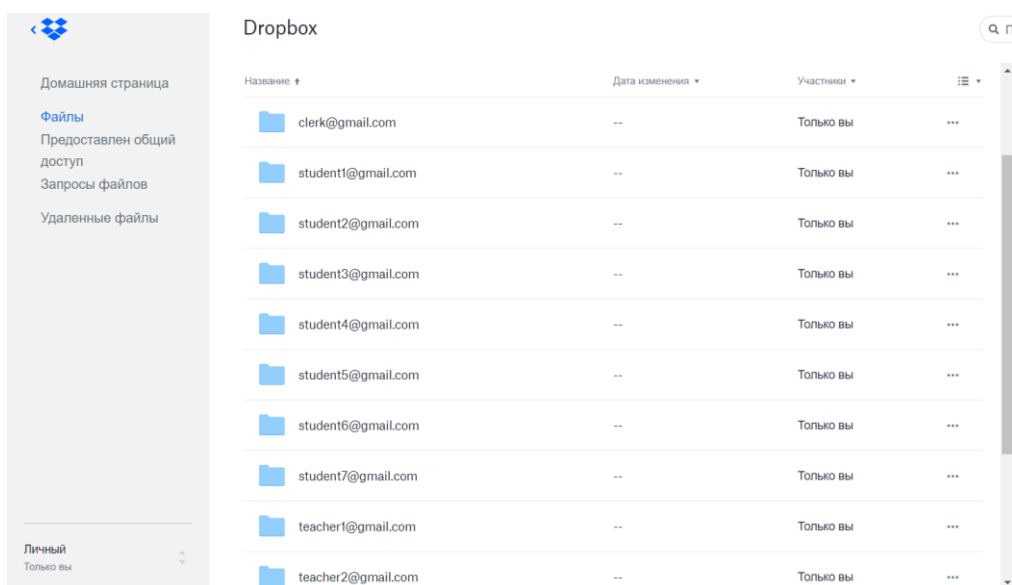


Рисунок 3.5 – Створенні папки користувачів в Dropbox

Далі необхідно реалізувати реєстрацію користувачів, тобто в першу чергу потрібна База Даних. Нам потрібно створити таблицю `users` в якій будемо додавати наших зареєстрованих користувачів. Процес створення таблиці зображений на рис. 3.6 .

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
id	INT		None			<input type="checkbox"/>	PRIMARY
first_name	VARCHAR	255	None	utf8_unicode_ci		<input type="checkbox"/>	--
last_name	VARCHAR	255	None	utf8_unicode_ci		<input type="checkbox"/>	--
email	VARCHAR	100	None	utf8_unicode_ci		<input type="checkbox"/>	UNIQUE
email_status	TINYINT	1	None			<input type="checkbox"/>	--
password	VARCHAR	100	None	utf8_unicode_ci		<input type="checkbox"/>	--

Рисунок 3.6 – Створення таблиці users

Е-mail ми будемо використовувати в якості логіна, тому це поле повинна бути унікальною, тобто мати індекс UNIQUE. Всі поля типу VARCHAR повинні мати значення за замовчуванням NULL і кодування utf8mb4\_unicode\_ci.

Тепер перейдемо до форми реєстрації, яка знаходиться в файлі form\_register.php. В цілях безпеки додаємо поле для введення капчі.

```
 <br>
```

Будь-яка форма потребує перевірки на валідність введених даних, як на стороні клієнта (за допомогою JavaScript, jQuery), так і на стороні сервера. Особливу уважність приділено полю Email. Дуже важливо щоб введена поштова адреса була дійсною. Для даного поля input, ми задали тип email (type = "email"), це застерігає від неправильних форматів. Але, цього недостатньо, тому що через інспектор коду, якого надає нам браузер, можна легко змінити значення атрибута type з email на text, і все, наша перевірка буде вже недійсна(рис. 3.7).

#### Форма реєстрації

Имя:

Фамилия:  173 x 21

Email:

Пароль:

Рисунок 3.7 – Перевірка поля input на його тип

І в такому випадку, необхідно зробити більш надійну перевірку. Для цього, скористаємося бібліотекою jQuery від JavaScript. За допомогою даної бібліотеки, перевіряємо введений поштовий адрес на валідність. Якщо користувач ввів неправильний Email, то ми виводимо йому помилку про це і дезактивуємо кнопку відправки форми. Якщо все добре, то прибираємо помилку і активуємо кнопку відправки форми. І так, з перевіркою форми клієнтської частини закінчено. Тепер можемо відправити її на сервер, де також зробимо пару перевірок і додамо дані в БД. Форму відправляємо на обробку до файлу register.php, через метод POST. Назва даного файлу обробника, зазначено в значення атрибуту action. А метод відправки зазначено в значення атрибуту method.

Якщо в запиті на додавання користувача в БД сталася помилка, додаємо повідомлення про цю помилку в сесію і повертаємо користувача на сторінку реєстрації. В випадку успішної реєстрації, тобто після успішного додавання користувача в таблицю users, необхідно отримати унікальний token з Dropbox, і разом з адресою електронної пошти та датою реєстрації додати користувача в таблицю confirm\_users. Після того як додали дані в таблицю confirm\_users, перевіряємо якщо раптом сталася помилка в запиті, то додаємо в сесію повідомлення про цю помилку, інакше відправляємо користувачу на зазначену пошту лист, з посиланням, для підтвердження пошти. А на сайті виводимо повідомлення про те, що реєстрація пройшла успішно і необхідно підтвердити введений адресу електронної пошти (рис. 3.8).

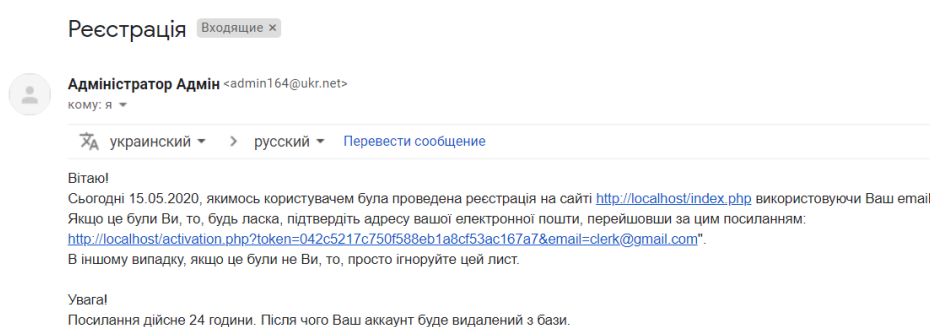


Рисунок 3.8 – Підтвердження реєстрації

Після натискання на посилання в листі, перейдемо в файл `activation.php`, в якому повинні закінчити з підтвердженням. Тобто, в цьому файлі необхідно, видалити користувача який підтвердив свою пошту, з таблиці `confirm_users`, а в таблиці `users`, поміняти статус пошти з 0 на 1. Далі користувач може перейти до авторизації.

Форма авторизації знаходиться у нас в файлі `form_auth.php`. Вона має поля для введення ПОШТОВОЇ адреси, пароля і капчи. Скрипт для перевірки формату поштової адреси і довжини пароля знаходиться в файлі `header.php`, тому він буде діяти і на поля з цієї форми. Запуск сесії також відбувається в файлі `header.php`, тому в файлі `form_auth.php` сесію запускати не потрібно, тому що отримаємо помилку. Після усунення помилок кнопка увійти стає активною, і користувач зможе відправити форму на сервер, де вона буде оброблятися. В значення атрибута `action` у форми авторизації вказано файл `auth.php`, це означає, що форма буде оброблятися саме в цьому файлі. І так, відкриваємо файл `auth.php` і пишемо код для обробки форми авторизації. Перше що потрібно зробити це запустити сесію і підключити файл `dbconnect.php` для з'єднання з БД.

Далі нам потрібно оголосити осередки для зберігання повідомлень в глобальному масиві `$_SESSION`. Далі потрібно перевірити чи була натиснута кнопка відправки форми (кнопка увійти). Тобто перевіряти не зайшов зловмисник безпосередньо в файл `auth.php`. Після всіх перевірок користувач може зайти на власну сторінку відповідно від його ролі. Також було реалізовано відновлення пароля користувача. На сторінку відновлення пароля, просимо користувача ввести свою поштову адресу. При обробці даної форми перевіriamo, чи зареєстрований користувач з таким поштовою адресою, якщо так, то на вказану поштову адресу відправимо лист з посиланням, яка буде ввести на сторінку установки нового пароля, інакше, якщо такий користувач не зареєстрований то, введемо повідомлення про помилку .

Для розмежування ролей використовувалася таблиця `roles` з Бази Даних.

```
if ($row["id_role"] == 1){
    header("HTTP/1.1 301 Moved Permanently");
```

```

header("Location:index.php");}
else if ($row["id_role"] == 2){
    header("HTTP/1.1 301 Moved Permanently");
header("Location:admin.php");}
else if($row["id_role"] == 3 || $row["id_role"] == 5){
    header("HTTP/1.1 301 Moved Permanently");
header("Location:teacher.php");}
else if($row["id_role"] == 4 ){
    header("HTTP/1.1 301 Moved Permanently");
header("Location:clerk.php"); }
exit();
}

```

Таким чином, розроблена система розмежовує права використання на типи користувачів «Студент», «Дипломний керівник», «Секретар екзаменаційної комісії», «Завідувач кафедри», «Адміністратор», і має інструмент адміністрування сайтом.

Робота студента і керівника відбувається в спільній хмарі. Студент завантажує необхідний документ причому фіксується дата завантаження та версія документа. Дата завантаження вікується за допомогою функції pow() PHP. Для версії документа необхідно реалізувати логічний механізм, а саме:

1. Студент вперше завантажує необхідний документ – версія 1.0.
2. В свою чергу керівник перевіряє документ і змінює статус (відхилено – наступна версія буде – 1.1 , прийнято – версія змінюється на 2.0 , робота завершена – закінчення фіксації версії) та залишає короткий коментар.
3. Секретар екзаменаційної комісії може призначити власний статус (прийнято – наступна версія не змінюється, перевірено й допущено до захисту – закінчення фіксації версії, перевірено й потребує доопрацювання – наступна версія 2.1, відхилено – версія змінюється на 1.1).

Для реалізації системи було виконано функціонал збереження файлу з Dropbox. Але через деякий час з'ясувалося, що зі скачуванням є одна проблема. Так як для тестування системи використовувався браузер Google Chrome, скачування файлів відбувалася - Google Chrome запитував: відкрити або завантажити. А Internet

Explorer - за замовчуванням відкривав документи. А для рядового користувача скачати цей файл виявиться досить важко.

Така проблема виникала і з різними типами файлів (doc, pdf, xls, jpg, gif і т.д.) повинна запускатись, яку асоціюють з цим типом файлу.

Було реалізовано функція яка вирішує дану проблему:

```
if (isset($_GET["file_id"])){
    $id = $_GET["file_id"];
    $file1 = $mysqli->query("SELECT * FROM `files` where id_file = '". $id. "'");
    $rowf = $file1->fetch_assoc();
    $filepath = 'Files/' . $rowf["name"];
    if(file_exists($filepath)){
        header("Content-Type: application/octet-stream");
        header("Content-Description: File Transfer");
        header("Content-Disposition: attachment; filename=" .
basename($filepath));
        header("Expires: 0");
        header("Cache-Control: must-revalidate");
        header("Pragma:public");
        header("Content-Length:" . filesize("Files/" . $rowf["name"]));
        readfile("Files/" . $rowf["name"]);
        exit;}}}
```

Даний скрипт допомагає управляти збереження файлів з системи. Не допускати браузерам запускати асоційовану програму з даним файлом за умовчанням (doc, pdf і т.д.), а завжди питати у користувача що потрібно робити (відкрити або зберегти).

Реалізовано відслідковування основних етапів документообігу Секретарем екзаменаційної комісії та завідувачем кафедри зі своїх сторінок, а саме перегляд зведеної статистики, яким студентам вже керівники відповіли, що їх робота є завершеною, і чи можна її надсилати на перевірку на плагіат.

Для того, щоб реалізувати даний механізм необхідно виконувати запити с Базы Даних для кожного стовпця власне свій. Кожен запис необхідно сортувати за збільшення версії документа DESC.

```
<td>
<?php
```



```

$pz = $mysqli->query("SELECT `status`.name_status from `status` JOIN `files` ON
`status`.id = `files`.status_id WHERE `files`.stud_id = ".$array["id"]." and `files`.sec_id
= 1 order by `files`.version DESC LIMIT 1");
while ($PZ = $pz->fetch_assoc()): ?>
    <?php echo $PZ["name_status"];?><?php
endwhile;?>
</td>

```

Також Секретарем екзаменаційної комісії та Завідувачем кафедри зі своїх сторінок мають доступ до зведеної таблиці з інформацією, в якому статусі знаходиться остання версія документа для кожного студента. Даний механізм працює аналогічно зі статусною таблицею – через запити до Баз Даних. Всі дані заносяться в масив і для розробки статистики підраховується кількість студентів, яких робота або завершена, або ні.

```

<td>
<?php
$pz = $mysqli->query("SELECT `status`.name_status from `status` JOIN `files` ON
`status`.id = `files`.status_id WHERE `files`.stud_id = ".$array["id"]." and `files`.sec_id
= 1 order by `files`.version DESC LIMIT 1");
while ($PZ = $pz->fetch_assoc()): ?>
    <?php if ($PZ["name_status"] == "Робота завершена"){
        $y++; echo $PZ["name_status"]; }
    else { $n++; echo "Робота не завершена"; }
    ?>
    <?php endwhile;?>
</td></tr>
<?php endwhile;?>
</tbody>
</table><p></p>
<?php echo "Кількість студентів, що завершили роботу: " . $y . "."; ?><p></p>
<?php echo "Кількість студентів, що не завершили роботу: " . $n . ".";?>

```

Тепер створимо просту діаграму для відображення статистики, використовуючи Google Chart. Google Chart Tools API - це багатофункціональний набір інструментів для візуалізації даних. За допомогою нього можна відносно легко будувати графіки та діаграми на сайті.

Функціонал Google Chart Tools API включає в себе:

- динамічні піктограми;
- карти;

- циферблати і дисплеї;
- формули;
- QR-коди;
- можливість створювати свої інструменти візуалізації і використовувати сторонні.

```
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);
function drawChart() {
    var data = google.visualization.arrayToDataTable([
        ['Завершена робота', 'Кількість'],
        ['Так', <?php echo $y ;?> ],
        ['Ні', <?php echo $n ;?>]]);
```

Щоб зробити діаграму 3D, додаємо параметр is3D: true.

```
var options = { title: 'Quantity of colors', is3D: true }
```

Визивається функція по id .

```
<script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);
    function drawChart() {
        var data = google.visualization.arrayToDataTable([
            ['Завершена робота', 'Кількість'],
            ['Так', <?php echo $y ;?> ],
            ['Ні', <?php echo $n ;?>]]);
        var options = {
            title: 'Статистика студентів'
        };
        var chart = new
google.visualization.PieChart(document.getElementById('piechart'));
        chart.draw(data, options);
    }
```

```
</script>
```

```
<div id="piechart" style="width: 900px; height: 500px;"></div>
```

В свою чергу Адміністратор додає нових користувачів з ролі Секретар екзаменаційної комісії, Завідувач кафедри та Дипломний керівник. Даний механізм аналогічний реєстрації студентів, але Адміністратор повинен власноруч відіслати листа кожному зареєстрованому ним користувачу. Також Адміністратор може додавати нові розділи ( документи) та групи, видаляти та редагувати користувачів. Даний механізм виконується за допомогою запитів з Бази Даних - даний механізм був представлений вище.

Також система автоматично, використовуючи PHP, виконує резервне копіювання баз даних MySQL на DropBox за допомогою кінцевих точок DropBox REST. Даний метод вимагає більше PHP-коду у файлі резервного копіювання, але він не вимагає PHP SDK, запропонований DropBox, і є стабільнішим, ніж SDK.

Все, що вам потрібно зробити, це створити файл PHP на сервері під назвою `dropbox_sqlbackup.php`. Цей файл PHP буде використовувати API REST, наданий DropBox, і повністю уникати використання будь-якого SDK.

Спочатку додаємо маркер доступу до верхньої частини файлу.

```
$accessToken = 'YOUR_DROPBOX_ACCESS_TOKEN';
```

По-друге, додати код, який створить стислий файл резервної копії нашої бази даних MySQL перед завантаженням його на сервер. Цей код буде працювати за винятком функції `DropboxUploadLargeFile ()`, яка буде визначена пізніше.

По-третє, додати функції, які фактично завантажують файл. Тут визначена функція `Dropbox Upload Large File ()`.

Резервна копія повинна створюватися через рівні проміжки часу. Необхідно використовувати планувальник задач CRON, щоб запускати його щомісяця. Якщо виникають якісь проблеми, необхідно перевірити `/ var / mail / your_user` на наявність будь-яких повідомлень про помилки, які можуть бути створені завданням CRON.

### 3.3 Використання програмного додатку

Розроблена система розмежовує права використання на типи користувачів «Студент», «Дипломний керівник», «Секретар екзаменаційної комісії», «Завідувач кафедри», «Адміністратор», і має інструмент адміністрування сайтом.рис3.0

Система дозволяє в онлайн-режимі забезпечувати роботу зі студентом. Студенти реєструються в системі, обирають дипломного керівника, групу та заносять необхідні дані (рис. 3.9).

Рисунок 3.9 – Реєстрація

Після вдалої реєстрації, виводиться повідомлення(рис. 3.10). Далі студент повинен зайти до електронної пошти, яку він вказав, для підтвердження реєстрації(рис.3.11).

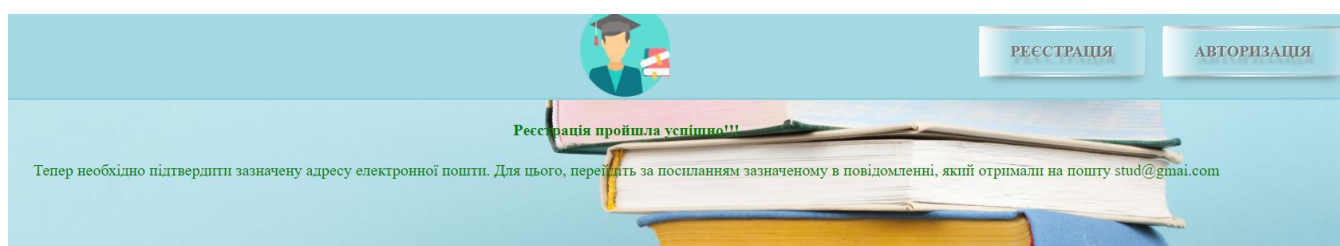


Рисунок 3.10 – Успішна реєстрація

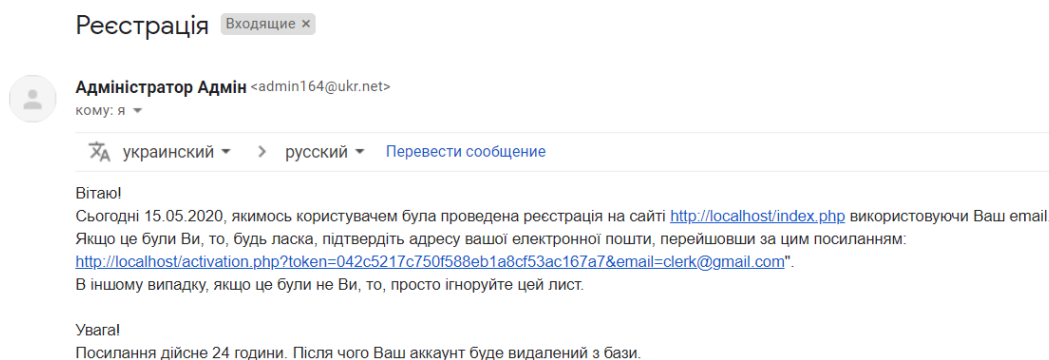


Рисунок 3.11 – Отримання листа для підтвердження реєстрації

Перейшовши за посилання в листі отримуємо повідомлення про успішну реєстрацію(3.12).

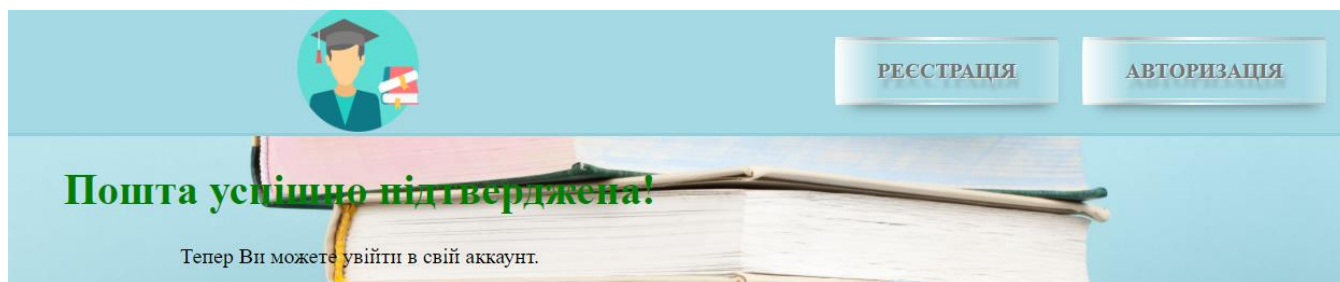


Рисунок 3.12 – Повідомлення про успішну реєстрацію

Тепер студент може авторизуватися, ввівши логін та пароль, форма авторизації представлена на рисунку 3.13.

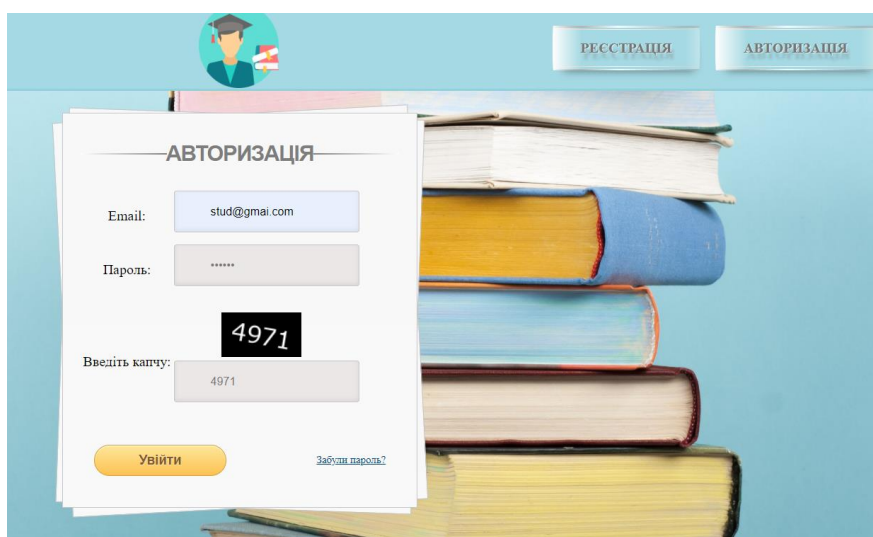


Рисунок 3.13 – Форма авторизації

Якщо студент забув свій пароль, він може змінити його на новий, натиснувши кнопку «Забули пароль?». В формі необхідно вести пошту, на якій зареєстрований студент(рис. 3.14). Посилання на сторінку установки нового пароля, був відправлений на зазначений E-mail(рис. 3.15).

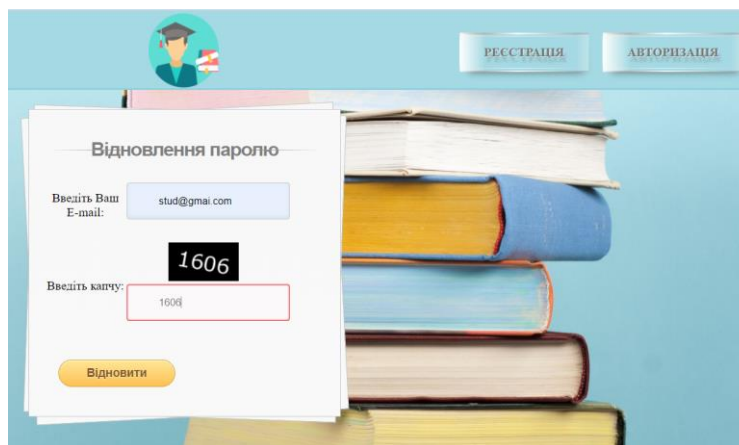


Рисунок 3.14 – Форма відновлення паролю

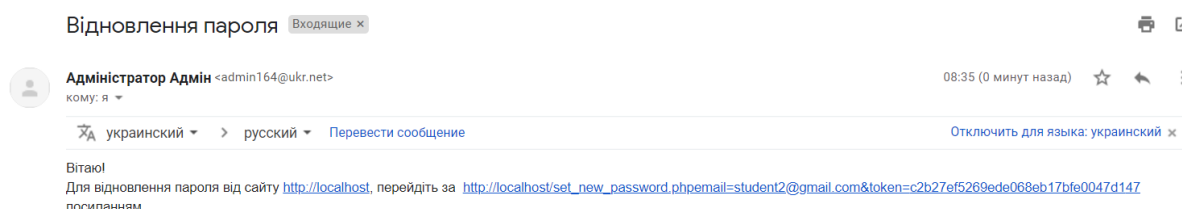


Рисунок 3.15 – Форма відновлення паролю

Після переходу за посиланням з листа, відкривається форма введення нового паролю(рис. 3.16). Вводимо новий пароль і натиснути кнопку «Змінити пароль», після чого зявиться повідомлення (рис. 3.17).

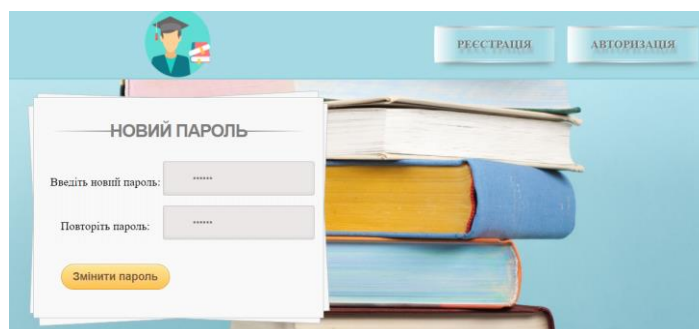


Рисунок 3.16 – Форма нового паролю

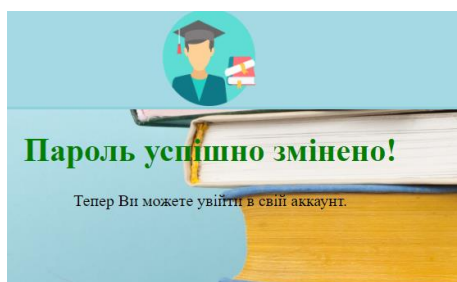


Рисунок 3.17 – Повідомлення про успішну зміну паролю

Після вдалої авторизації користувач потрапляє на власну сторінку(рис. 3.18). Зліва студент має інформацію про його дипломного керівника та меню вибору документу.

У процесі дипломного проектування ведеться обмін декількома документами:

- пояснювальна записка (ПЗ)(рис. 3.19) до диплому у текстовому форматі між студентом і керівником, між студентом та секретарем екзаменаційної комісії (ЕК);
- протокол перевірки роботи на плагіат(рис. 3.20) між секретарем ЕК та студентом;
- файли презентації(рис. 3.21) між студентом та керівником;
- пояснювальна записка, архів з програмним додатком(рис. 3.22), інша супровідна документація(рис. 3.23) - між студентом, керівником та секретарем ЕК - по завершенню роботи.

Студент завантажує необхідний документ, причому фіксується дата завантаження та версія документа.



Рисунок 3.18 – Головна сторінка студента

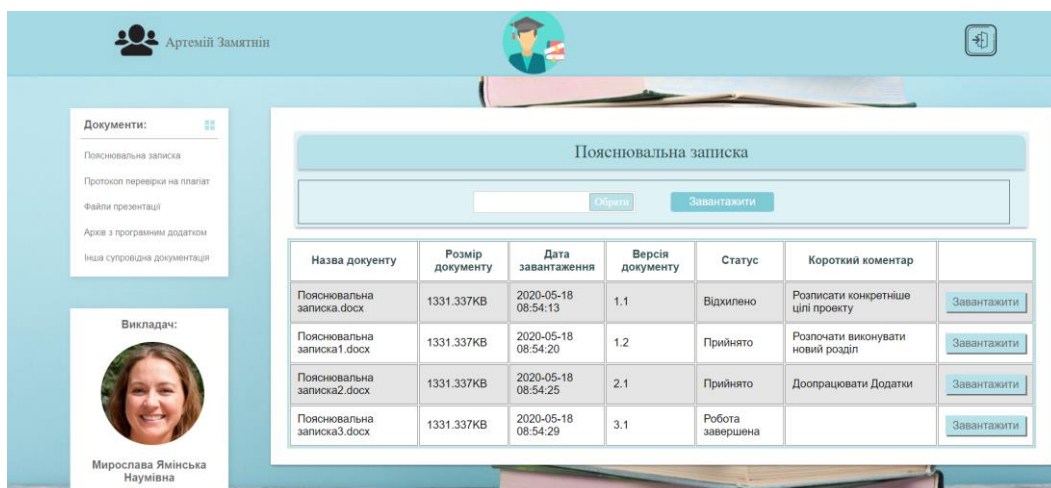


Рисунок 3.19 – Форма для обміну пояснювальної записки

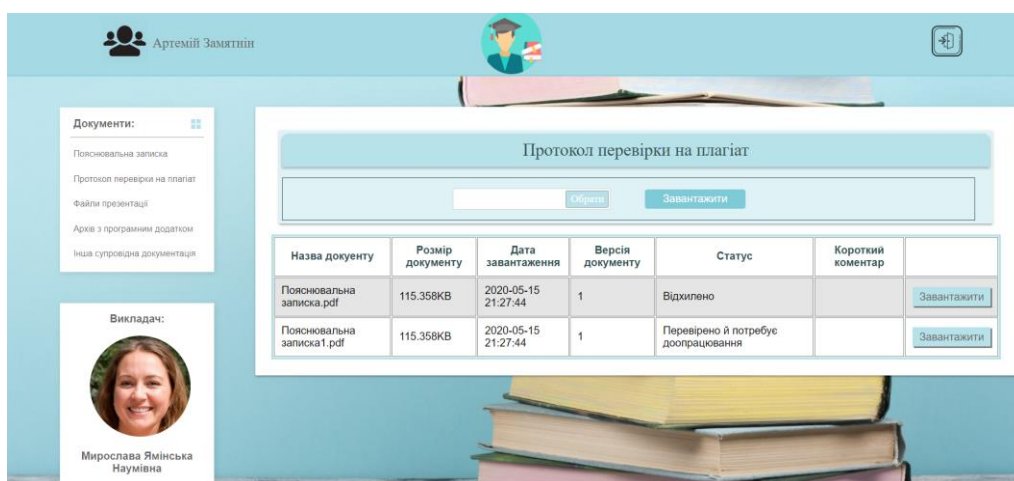


Рисунок 3.20 – Форма для обміну пояснювальної записки для перевірки на плагіат

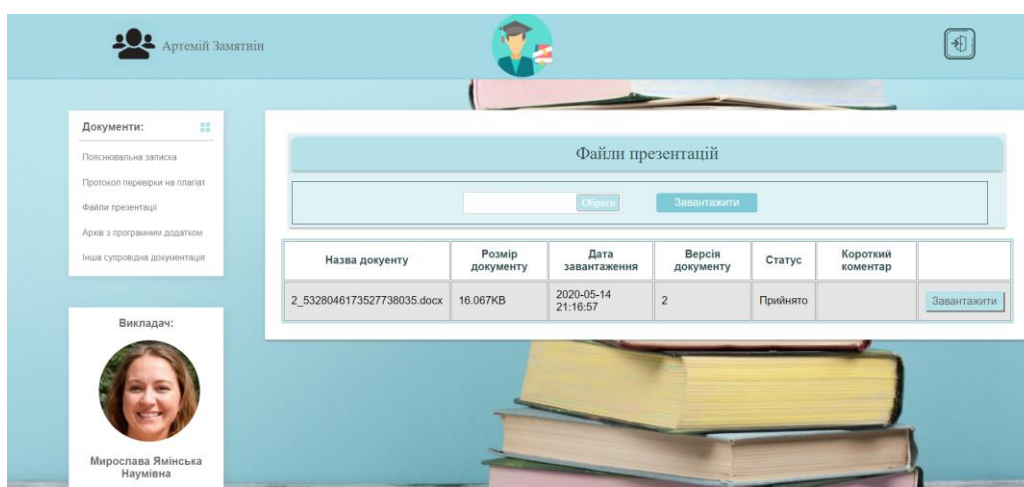


Рисунок 3.21 – Форма для обміну файлами презентації



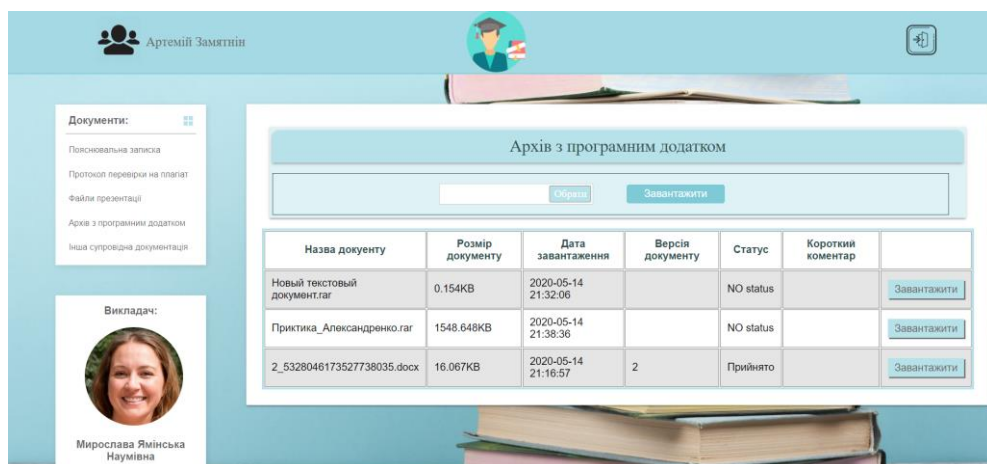


Рисунок 3.22 – Форма для обміну архіву додатку

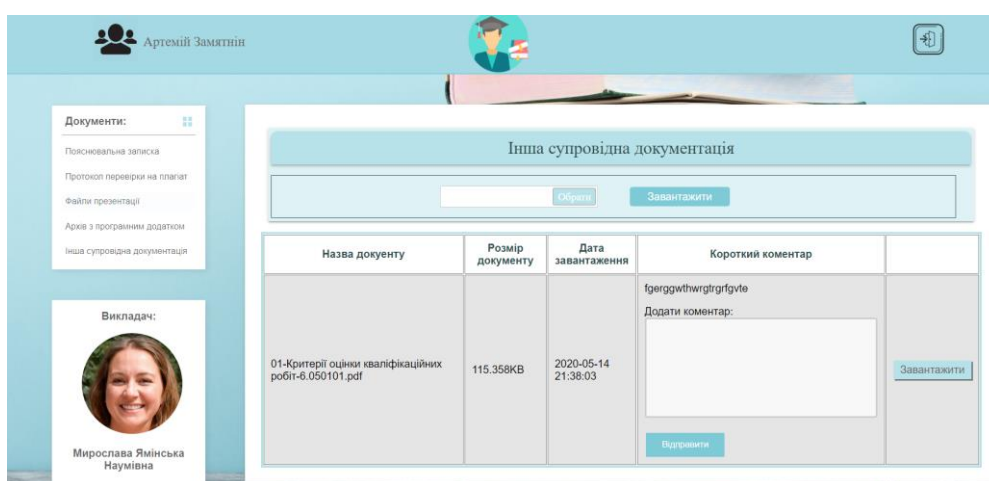


Рисунок 3.23 – Форма для обміну різною документацією

Також студент має форму виводу інформації про себе, представлений на рисунку 3.24.

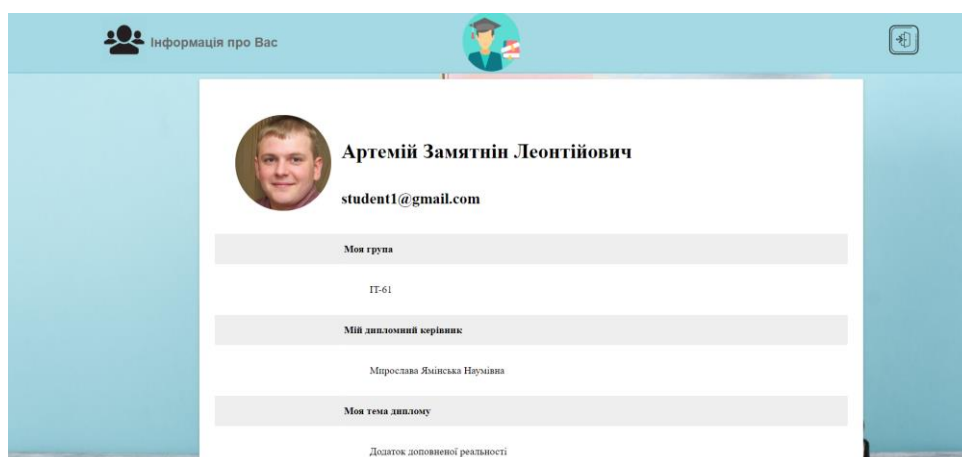


Рисунок 3.24 – Форма виводу контактної інформації студента

Після отримання листа з паролем, Дипломний керівник авторизується в системі. Зліва виведена інформація про його студентів (рис. 3.25). Натиснувши на ім'я студента, Дипломний керівник переходить до обміну документами. Керівник перевіряє документи і змінює статус (відхилено, прийнято, робота завершена) та залишає короткий коментар (рис. 3.26).



Рисунок 3.25 – Головна сторінка дипломного керівника

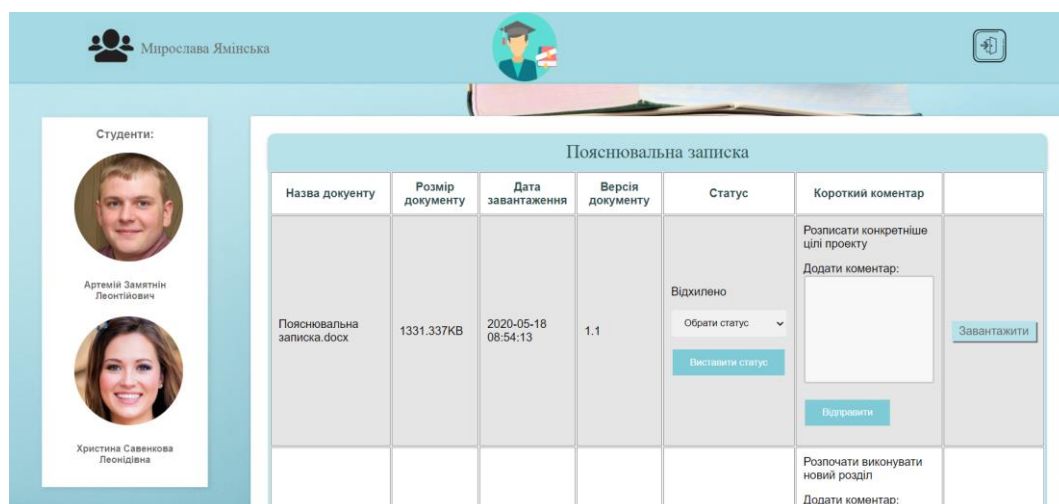


Рисунок 3.26 – Вивід документів студента

Також Дипломний керівник має доступ до власної контактної інформації (рис. 3.27).

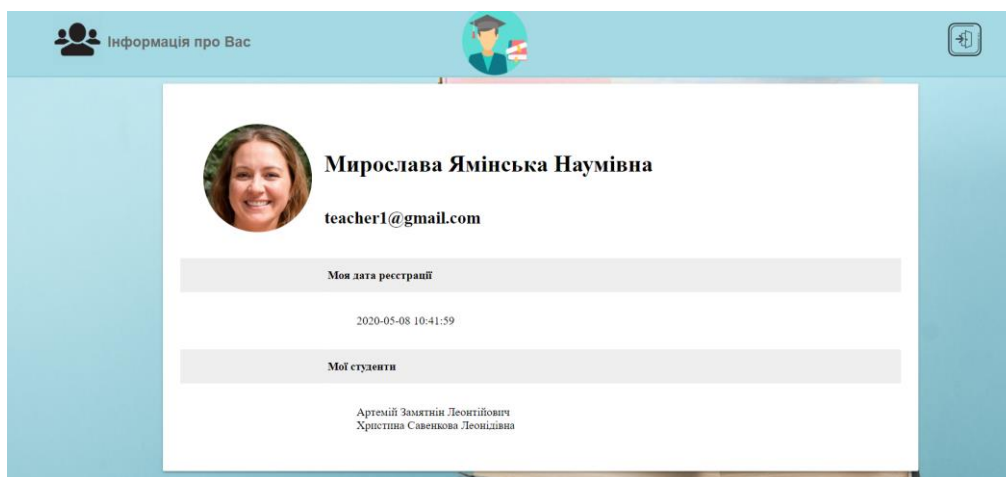


Рисунок 3.27 – Контактна інформація дипломного керівника

Завідуючий кафедри також отримує пароль і авторизується в системі. Він має власних студентів, їх документи організовані так само як у дипломного керівника(рис. 3.28).



Рисунок 3.28 – Головна сторінка завідуючого кафедри

Реалізовано відслідковування основних етапів документообігу Секретарем екзаменаційної комісії та Завідувачем кафедри, а саме перегляд зведеної статистики, яким студентам вже керівники відповіли, що їх робота є завершеною, і чи можна її надсилати на перевірку на плагіат(рис. 3.29).



ФІО	Пояснювальна записка	Протокол перевірки на плагіат	Файли презентації	Архів з програмним додатком
Артемій Замітнін Леонітович	Робота завершена	Відхилено	Прийнято	Прийнято
Галина Лачкова Гаврилівна	Відхилено			Робота завершена
Захар Ключников Остапович	Відхилено			Відхилено
Яна Крупина Володимирівна	Прийнято		Прийнято	Робота завершена
Тихон Типалой Єрофійович	Робота завершена	Перевірено й потребує доопрацювання	Прийнято	Відхилено
Марина Заріца Богданівна	Відхилено			Робота завершена
Діна П'яніха Єлісеївна	Прийнято			Прийнято

Рисунок 3.29 – Статусна таблиця

Також Секретарем екзаменаційної комісії та Завідувачем кафедри зі своїх сторінок мають доступ до зведеної таблиці з інформацією, в якому статусі знаходиться остання версія документа для кожного студента(рис. 3.30 – 3.31).



ФІО	Статус пояснювальної записки
Артемій Замітнін Леонітович	Робота завершена
Галина Лачкова Гаврилівна	Робота не завершена
Захар Ключников Остапович	Робота не завершена
Яна Крупина Володимирівна	Робота не завершена
Тихон Типалой Єрофійович	Робота завершена
Марина Заріца Богданівна	Робота не завершена
Діна П'яніха Єлісеївна	Робота не завершена

Рисунок 3.30 – Таблиця статусів пояснювальної записки

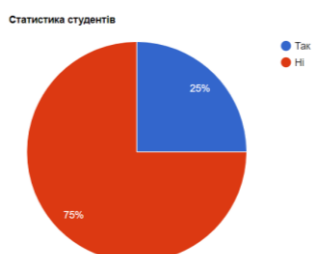


Рисунок 3.31 – Статистика студентів

Крім того, Секретар екзаменаційної комісії зі своєї сторінки має доступ до документів студентів для їх обміну(рис. 3.32 – 3.33).

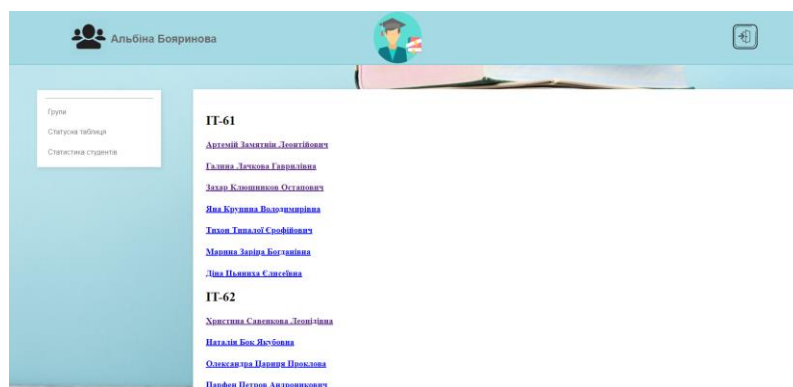


Рисунок 3.32 – Список студентів по групах

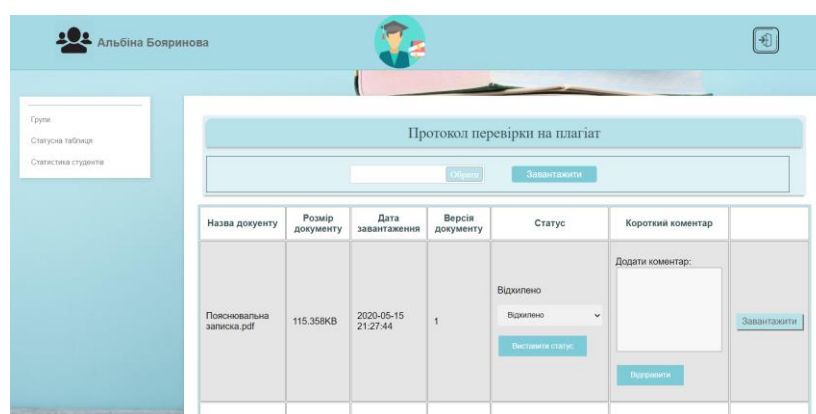


Рисунок 3.33 – Форма виводу документів обраного студента

В свою чергу Адміністратор додає нових користувачів з ролі секретар екзаменаційної комісії, Завідувач кафедри та Дипломний керівник(рис.3.34). Даний механізм аналогічний реєстрації студентів, але Адміністратор повинен власноруч відіслати листа кожному зареєстрованому ним користувачу. Також Адміністратор може додавати нові розділи ( документи) та групи, видаляти та редагувати користувачів.

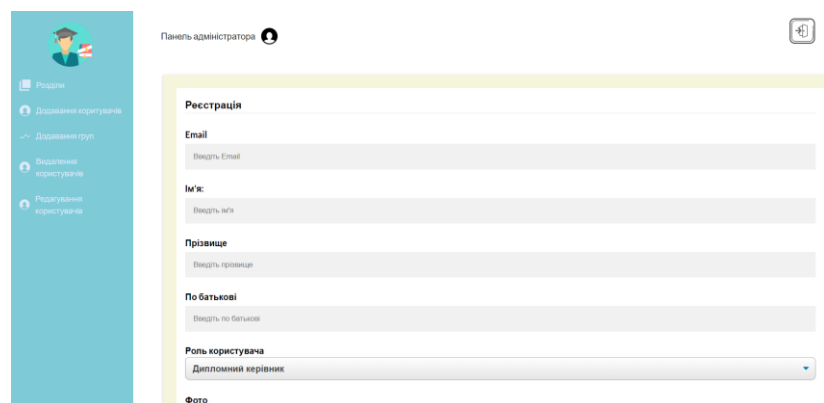


Рисунок 3.34 – Форма реєстрації користувачів через Адміністратора

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи бакалавра розроблено інформаційну систему забезпечення документообігу для дипломного проектування із використання хмарних сервісів.

У аналізі предметної області був розглянутий процес організації документообігу при дипломному проектуванні. Виконаний огляд існуючих СЕД та платформи університету показав, що потрібно створити нову інформаційну систему, через відсутність готових рішень у даній тематиці, які б задовольняли поставленим задачам.

Після прийняття рішення по розробці інформаційної системи у постановці задачі було детальніше розглянуто мету та задачі програмного продукту, створено схему роботи та складено технічне завдання на виконання робіт.

Для забезпечення досягнення поставленої в роботі мети виконано планування робіт, визначено календарний графік та проведено аналіз можливих ризиків. Процес роботи програми описаний за допомогою функціональних діаграм нотації IDEF0, а також моделі варіантів використання .

По створеним архітектурі було виконано реалізацію інформаційної системи. Клієнтська частина інформаційної системи була створена з використанням web технологій HTML, CSS та JavaScript. Серверна частина та логіка системи використовують мову PHP, базу даних MySQL. Результатом реалізації стала інформаційна система, що відповідає всім функціональним вимогам які висувалися в технічному завданні. Продемонстровано використання системи, що демонструють її працездатність.

Створена інформаційна система підтримки процесу дипломного проектування розроблена з метою скорочення часу на відповідність друкованої роботи вимогам стандарту, підвищення якості індивідуальної роботи і автоматизації складання і зміни списку дипломних робіт. Система надає можливість більш зручної віддаленої роботи з студентами і можливість стежити за ходом дипломного проектування завідувачу кафедри.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Александренко Т.В. Система забезпечення документообігу для дипломного проектування із використання хмарних сервісів. / Александренко Т.В., Парфененко Ю.В // Інформатика, математика, автоматика : матеріали та програма науково-технічної конференції, м. Суми, 2020 р. – Суми : СумДУ, 2020.
2. ЕЛЕКТРОННИЙ ДОКУМЕНТООБІГ, ТЕНДЕНЦІЇ ТА ПЕРСПЕКТИВИ/ укладач: Величкевич М.Б., Мітрофан Н.В., Кунанець Н.Е. – Львів : Національний університет «Львівський політехнік», 2010. – С. 44 - 53 .
3. Ткачук Г. І. Використання електронної системи документообігу у ВНЗ / Г. І. Ткачук, С.А. Постова // Магістратура в умовах євроінтеграційних процесів вищої школи. – Житомир: ЖДУ, 2014. – С. 254.
4. Гречко А.В. Основи електронного документообігу: навч. посіб. / Київський національний торговельно- економічний ун-т. – К., 2006. – 156 с.
5. Морзе Н. В. Педагогічні аспекти використання хмарних обчислень / Н. В. Морзе, О. Г. Кузьмінська // Інформаційні технології в освіті. – 2011. – № 9. – С. 20–29
6. Воронкін О. С. "Хмарні" обчислення як основа формування персональних навчальних середовищ / О. С. Воронкін // Збірник наукових праць : матеріали другої міжнародної науково-практичної конференції FOSS Lviv 2012, Львів, 26–28 квітня 2012 р. – Львів, 2012. – С. – 143–146.
7. П'ять вимірів хмарних обчислень: лекція голови Microsoft С. Балмера для студентів КПІ та інших ВНЗ [Електронний ресурс]. – Режим доступу : <http://www.microsoft.com/ukraine/events/ballmer-studentslecture2010/default.aspx>
8. Документообіг в дипломному проектуванні, методичні вказівки / укладач: А. Д. Хасбієва, А. Б. Шмідт - Санкт-Петербург: Санкт-Петербурзький державний архітектурно-будівельний університет, 2013. – 43 с.

9. Електронний документообіг Directum, система електронного документообігу (СЕД) та управління бізнес процесами: веб-сайт. URL: <https://www.directum.ru/> (дата звернення: 24.04.2020).
10. Євроменеджмент - консалтингові послуги: веб-сайт. URL: <http://www.emd.ru/>(дата звернення: 24.04.2020).
11. UpScale Soft (ГК "ОПТiМА") (Корпоративний блог): веб-сайт. URL: <https://club.cnews.ru/UpScale> (дата звернення: 24.04.2020).
12. Гран-Док :: Система Електронного Документообігу - Головна: веб-сайт. URL: <http://grandoc.ru/> (дата звернення: 24.04.2020).
- 13.Електронний документообіг і спільна робота): веб-сайт. URL: <http://www.paydox.ru/> (дата звернення: 24.04.2020).
- 14.Системи електронного документообігу: керівництво для покупця : веб-сайт. URL: <https://www.intalev.ru/library/articles/article.php?ID=24475> (дата звернення: 20.04.2020).
15. Впровадження систем електронного документообігу: проблеми та рішення : веб-сайт. URL: [http://www.iteam.ru/publications/it/section\\_64/article\\_2687/](http://www.iteam.ru/publications/it/section_64/article_2687/) (дата звернення: 21.04.2020).
- 16.Системи електронного документообігу. Enterprise Content Management. Управління корпоративною інформацією: веб-сайт. URL: <http://www.tadviser.ru/index.php/СЭД/> (дата звернення: 22.04.2020).
17. АКТУАЛЬНІ ПИТАННЯ ОСВІТИ. Збірник наукових праць/ укладач: О. О. Базиль, Ю. А. Кравченко, О. С. Соколов – Суми : Сумський державний університет, 2017. – С. 124 - 129.
18. Сенченко П.В. Обґрунтування доцільності застосування Web-орієнтованих технологій при розробці систем електронного документообігу // Проблеми теорії і практики управління. - 2011. - № 6. - С. 33-37.
19. Вендров А.М. Проектування програмного забезпечення економічних інформаційних систем: Підручник: Фінанси і статистика, 2002.- 352 с.
20. Ларман К. Применение UML и шаблонов проектирования [Електронний ресурс]. Режим доступу –Z:\ТПСПП\Література.



## **ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ**

### **1. Призначення й мета створення інформаційної системи**

#### **1.1 Призначення інформаційної системи**

Інформаційна система повинна представляти хмарний клієнт-серверний додаток забезпечення документообігу для дипломного проектування.

#### **1.2 Мета створення інформаційної системи**

Автоматизація організаційного процесу дипломного проектування.

#### **1.3 Цільова аудиторія**

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

1. Студенти.
2. Викладачі-керівники дипломних робіт.
3. Голова та члени екзаменаційної комісії
4. Секретар екзаменаційної комісії.
5. Завідувач кафедри.

### **2. Вимоги до інформаційної системи**

#### **2.1 Вимоги до інформаційної системи в цілому**

##### **2.1.1 Вимоги до структури й функціонування інформаційної системи**

Інформаційна система повинна бути реалізована у вигляді хмарного клієнт-серверного додатку, доступного в мережі Інтернет. Він повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

##### **2.1.2 Вимоги до персоналу**

Для підтримки додатку й експлуатації веб-інтерфейсу системи керування сайтом від персоналу не повинно вимагатися спеціальних технічних навичок, знання

технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером.

### **2.1.3 Вимоги до збереженні інформації**

У системі керування сайтом повинен бути передбачений механізм резервного копіювання структури й умісту бази даних. Процедура резервного копіювання повинна проводитися співробітником, відповідальним за підтримку сайту, не рідше 1 рази в місяць.

### **2.1.4 Вимоги до розмежування доступу**

Користувачів сайту можна розділити на 5 груп відповідно до прав доступу:

1. Студент.
2. Дипломний керівник.
3. Секретар екзаменаційної комісії.
4. Завідувач кафедри.
5. Адміністратор.

## **2.2 Вимоги до функцій, виконуваних сайтом**

### **2.2.1 Основні вимоги**

#### **2.2.1.1 Структура сайту**

Додаток повинен складатися з наступних модулів

- Вікно авторизації – початкова сторінка з полями авторизації;
- Головна – відрізняється відсутністю або наявністю деяких розділів для кожного типу користувача (ролі).

#### **2.2.1.2 Навігація**

Користувацький інтерфейс сайту повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і сторінок. Навігаційні елементи повинні забезпечувати

однозначне розуміння користувачем їх змісту: посилання на сторінки повинні бути мати заголовок, умовні позначки відповідати загальноприйнятим. Графічні елементи навігації повинні бути мати альтернативний підпис.

Система повинна забезпечувати навігацію по всіх доступних користувачеві ресурсах і відображати відповідну інформацію.

### **2.2.1.3 Наповнення сайту (контент)**

Модулі додатку повинні формуватися програмним шляхом на підставі інформації з бази даних, яка зберігається в Dropbox.

Модифікація вмісту розділів повинна здійснюватися за допомогою адміністраторського веб-інтерфейсу (системи керування сайтом), який без застосування спеціальних навичок програмування ( без використання програмування й спеціального кодування або форматування) повинен передбачати можливість редагування інформаційного вмісту сторінок сайту.

### **2.2.1.4 Система навігації (карта сайту)**

Взаємозв'язок між модулями додатку представлено на рис. А1.

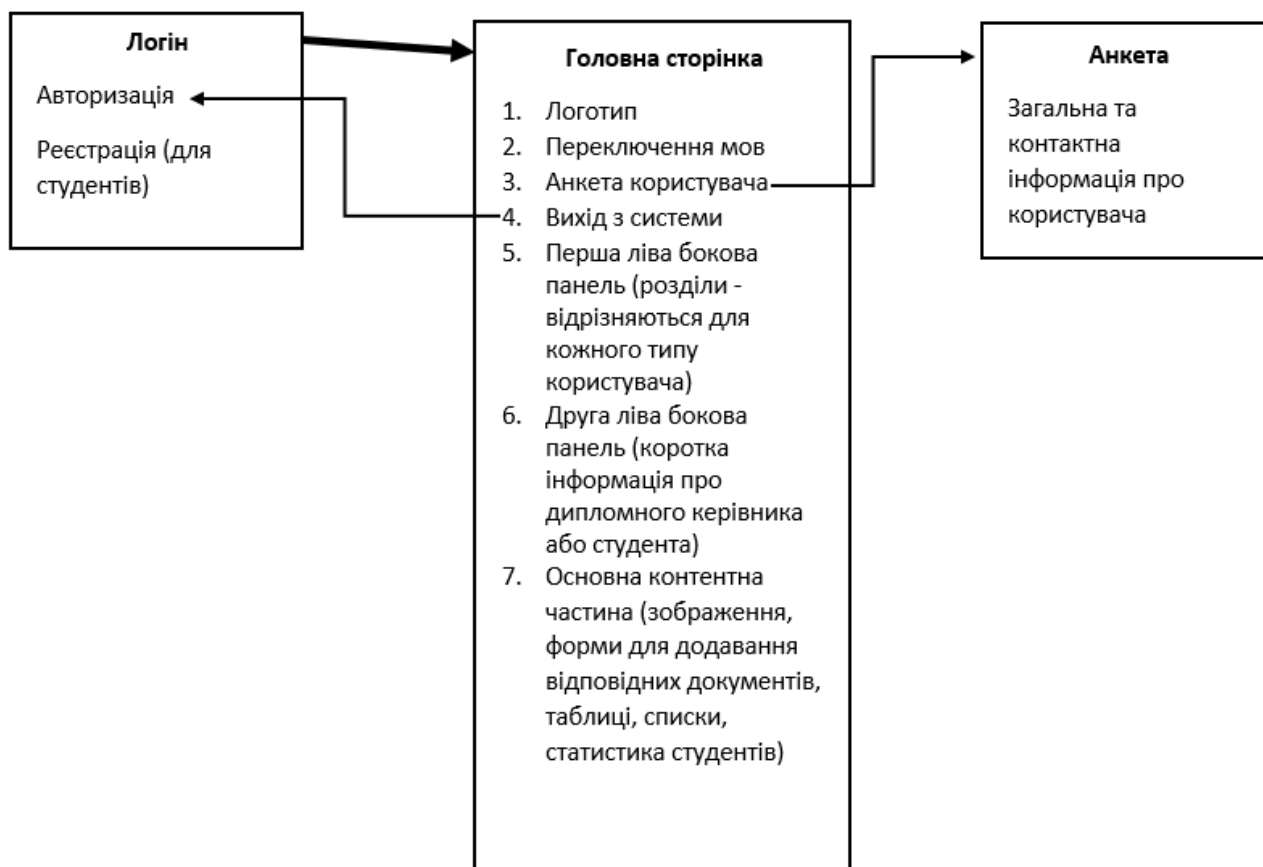


Рисунок А.1 – Взаємозв'язок між модулями додатку

## 2.2.2 Вимоги до функціональних можливостей

### 2.2.2.1 Вимоги користувача

Таблиця А.1 – Вимоги користувача

ID	Потреба користувача	Група користувачів
ВК-0100	Студент повинен реєструватися в системі особисто.	Студент
ВК-0200	Студент повинен мати доступ до власної хмари, де завантажуватиме необхідні версії документів та змінюватиме їх статуси.	Студент
ВК-0300	Дипломний керівник повинен мати доступ до хмар своїх студентів та змінювати статуси їх документів, залишати короткий коментар, завантажувати власні документи.	Дипломний керівник

## Продовження таблиці А.1 – Вимоги користувача

<b>ID</b>	<b>Потреба користувача</b>	<b>Група користувачів</b>
ВК-0400	Секретар екзаменаційної комісії повинен призначити власний статус до фінальної версії документа студента.	Секретар екзаменаційної комісії
ВК-0500	Завідувач кафедри та секретар екзаменаційної комісії повинні бачити зведену статистику по стану зданих дипломних робіт	Завідувач кафедри та секретар екзаменаційної комісії
ВК-0600	Завідувач кафедри та секретар екзаменаційної комісії повинні бачити зі своїх сторінок зведену таблицю з інформацією, в якому статусі знаходиться остання версія документа для кожного студента.	Завідувач кафедри та секретар екзаменаційної комісії
ВК-0700	Адміністратор повинен додавати користувачів із правами Дипломний керівник, Секретар екзаменаційної комісії, Завідувач кафедри	Адміністратор
ВК-0800	Адміністратор повинен додавати й видаляти розділи сайту.	Адміністратор

**2.2.2.2 Системні вимоги**

Таблиця А.2 – Опис системних вимог

<b>ID</b>	<b>Системна вимога</b>	<b>Пріоритет</b>	<b>Опис</b>
СВ-0100	Реєстрація студента	М	Система повинна автоматично призначати дипломного керівника відповідно до бажаної тематики дипломної роботи яку вказав студент при реєстрації, та завантаженості керівників.
СВ-0200	Можливості студента	М	Система повинна надавати доступ студенту до власної хмари, в якій студент завантажуватиме необхідні документи doc, pdf, ppt та zip форматів, причому система повинна фіксувати дату завантаження та версію документа.

## Продовження таблиці А.2 – Опис системних вимог

ID	Системна вимога	Пріоритет	Опис
СВ-0300	Статус документа	М	Для документів, обмін якими ведеться між користувачами, система повинна змінювати їх статус, причому різні користувачі мають різні варіанти статусу : 1) студент - на доопрацюванні, завершено; 2) дипломний керівник - у процесі перевірки, відправлено на доопрацювання, вважаю роботу завершеною, відхиллено; 3) секретар ЕК - прийнято, очікує перевірки на плагіат, перевірено й допущено до захисту, перевірено й потребує доопрацювання, відхиллено. При зміні статусу система повинна надавати можливість залишати короткі коментарі.
СВ-0400	Можливості дипломного керівника	М	Система повинна надавати доступ дипломному керівнику до хмар своїх студентів, завантажувати власні документи doc, pdf, ppt та zip форматів в розділ «Інша документація» для студентів.
СВ-0500	Статистика завершеної роботи	М	Система повинна відображати для завідувача кафедри та секретаря екзаменаційної комісії на сторінці зведену статистику, яким студентам вже керівники відповіли, що їх робота є завершеною, і чи можна її надсилати на перевірку на плагіат.
СВ-0600	Статусна таблиця	М	Система повинна відображати для завідувача кафедри та секретаря екзаменаційної комісії на сторінці таблицю з прізвищами студентів та датою надсилання роботи, якщо робота надіслана із затримкою, відмічати жовтим кольором, якщо робота ще не надіслана, червоним, цю таблицю можна буде завантажити як pdf файл.
СВ-0700	Реєстрація	М	Система повинна надавати можливість адміністратору реєструвати дипломних керівників, секретаря екзаменаційної комісії, завідувача кафедри та висилати на їх поштові скриньки унікальні логін і пароль.

## Продовження таблиці А.2 – Опис системних вимог

ІД	Системна вимога	Пріоритет	Опис
СВ-0800	Вистежування застарілої інформації	М	Система повинна надавати можливість адміністратору вистежувати терміни зберігання та знищення документів після закінчення терміну давності, а також знищення копій інформації і особистої інформації студентів.
СВ-0900	Адміністративна частина сайту	М	Система повинна надавати можливість адміністратору додавати, редагувати й видаляти вміст статичних і динамічних сторінок.

## 2.2.2.3 Загальні вимоги

Стиль додатку можна описати як сучасний, діловий. У якості основних кольорів додатку рекомендується використовувати пастельні відтінки.

Оформлення не повинне заземляти інформативність: хоча на сайті не повинно бути досить багато графіки, він повинен бути зручний користувачам у плані навігації.

Розташування елементів вікна авторизації та реєстрації схематично показано на рис. А.2 – А.3.

The image shows a web interface for a system login. At the top, there is a header bar with the word 'DIPLOMA' on the left and a small Ukrainian flag on the right. Below the header is a large green rectangular area containing a white-bordered box. Inside this box, the text 'Login to the system' is centered. Below this text are two input fields: 'Login:' followed by a text input field, and 'Password:' followed by a password input field. At the bottom of the box are two blue buttons: 'Sign in' on the left and 'Sign up' on the right.

Рисунок А.2 – Макет вікна авторизації

The screenshot shows a registration form within a window titled "DIPLOMA". The form is titled "Registration in the system" and contains the following elements:

- Fields for "Surname", "Name", and "Patronymic" arranged horizontally.
- Fields for "Birthday" and "Email" arranged horizontally below the first row.
- Two buttons at the bottom: "Sign up" (light blue) and "Cancel" (dark red).

Рисунок А.3 – Макет вікна реєстрації

Розташування елементів головної сторінки (з усіма розділами без розподілу по ролям) схематично показано на рис. А.4.

The screenshot shows the main page of the "DIPLOMA" system. The page layout includes:

- Header: "DIPLOMA" on the left and "Questionnaire Exit" with a Ukrainian flag icon on the right.
- Sub-header: "Diploma topic ...."
- Left sidebar menu:
  - Group
  - Explanatory note**
  - Presentation
  - Application archive
  - Other documents
  - Plagiarism
  - Student statistics
  - Status table
  - Graduate supervisor or student
  - Profile picture placeholder
  - Name Surname
- Main content area:
  - "File" button above a large blue bar.
  - "Title" section containing:
    - Text input: "Microsoft Word Document.doc"
    - Text: "Date and version"
    - Status dropdown: "Status"
  - Two additional large blue bars below the title section.

Рисунок А.4 – Макет головної сторінки

Розташування елементів вікна адміністратора схематично показано на рис. А.5.



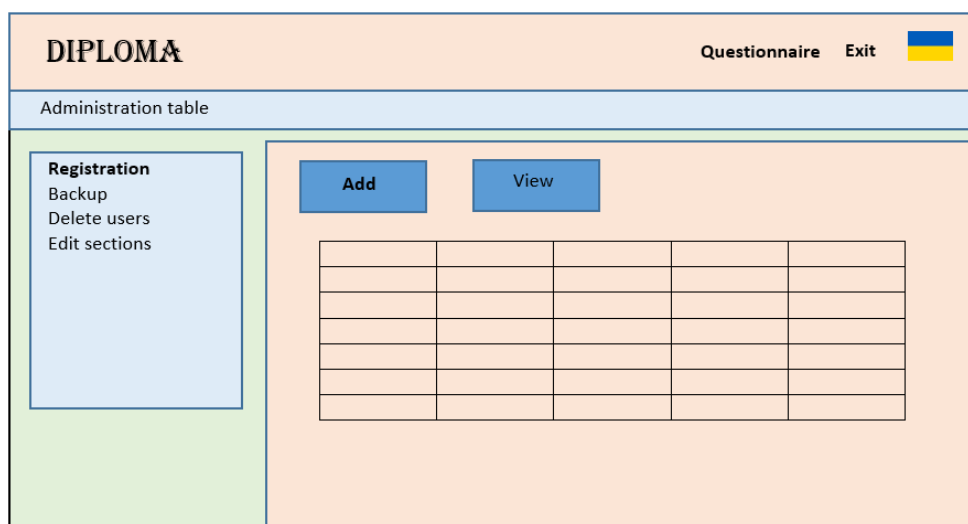


Рисунок А.5 – Макет вікна адміністратора

#### 2.2.2.4 Типові навігаційні й інформаційні елементи

- Шапка сайту.
- Ліве бокове меню.
- Основне поле контенту

#### 2.2.2.5 Шапка сайту

Шапка сайту повинна містити логотип і назву сайту. Логотип є посиланням на головну сторінку сайту.

#### 2.2.2.6 Ліве бокове меню

Бокове меню повинне розташовуватися в лівій частині вікна ( залежно від затвердженого дизайн-макета) і містити посилання на розділи, зв'язані тематично з поточним, якщо такі існують для обраного розділу першого рівня.

#### 2.2.2.7 Основне поле контенту

Основне поле контенту повинне розташовуватися в центрі сторінки. У цьому полі відображається основний зміст обраного розділу. Стильове оформлення матеріалів і їх елементів (зображень, форм, таблиць і т.п.) повинне бути єдиним для всього додатку.

## **2.3 Вимоги до видів забезпечення**

### **2.3.1 Вимоги до інформаційного забезпечення**

У вигляді інформаційного забезпечення виступають UML діаграми, діаграми класів, діаграми варіантів використання які будуть створенні під час планування проекту.

Реалізація додатку відбувається з використанням:

- Dropbox API
- HTML
- CSS
- JavaScript
- PHP 5.6
- MySQL 5.6

### **2.3.2 Вимоги до лінгвістичного забезпечення**

Сайт повинен бути виконаний українською мовою.

### **2.3.3 Вимоги до програмного забезпечення**

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;

Включена підтримка javascript і cookies.

### **2.3.4 Вимоги до апаратного забезпечення**

Апаратне забезпечення серверної частини повинне задовольняти наступним вимогам:

- Для зберігання файлів використовувати хмарне сховище «Dropbox».
- Доступ до мережі Інтернет.

Апаратне забезпечення клієнтської частини повинне забезпечувати підтримку програмного забезпечення клієнтської частини, зазначеного в п. 2.3.3.

### 3. Склад і зміст робіт зі створення додатку

Докладний опис етапів роботи зі створення додатку наведено в табл. А.3.

Таблиця А.3 – Етапи створення сайту

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Проектування: Створення юзкейс-діаграм, структури БД, проектування класів.	5 дні
2	Розробка Frontend: Створення дизайну web-додатку, подавання інформації в привабливому вигляді, приємному і зручному для користувачів.	3 дні
3	Використання хмарного сховища даних Розробка програмної оболонки до хмарного сховища Dropbox API	2 дні
4	Розробка модуля авторизації: Розроблення та реалізація вікна авторизації користувача	2 дні
5	Розробка модуля реєстрація: Розроблення та реалізація вікна авторизації користувача	2 дні
6	Розробка модуля розділів: Розроблення та реалізація лівої бокової панелі додатку для кожної ролі	3 дні
7	Головний вивід (основна контентна частина): Реалізація виводу структурованих даних у центральній частині сайту по натисканню на кнопки розділів лівої бокової панелі	10 днів

## Продовження таблиці А.3 – Етапи створення сайту

№	Склад і зміст робіт	Строк розробки (у робочих днях)
8	Розробка модуля адміністратора: Розроблення та реалізація вікна адміністратора	2 дні
9	Завершення роботи: Проведення стилістичних виправлень додатку, перевірка (тестування) реалізованого функціоналу	2 дні
	<b>Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту</b>	32 дні

**4. Вимоги до складу й змісту робіт із введення сайту в експлуатацію**

Для створення умов функціонування, при яких гарантується відповідність створюваного сайту вимогам сьогодення ТЗ і можливість його ефективної роботи, в організації Замовника повинен бути проведений певний комплекс заходів, а саме забезпечити виконання вимог до програмного ти апаратного забезпечення.

## ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ

### 1. Ідентифікація ідеї проекту

Метою інформаційної системи дипломного проекту є забезпечення організаційного процесу дипломного проектування.

Дипломний проект призначений для скорочення часу на відповідність друкованої роботи вимогам стандарту, підвищення якості індивідуальної роботи і автоматизації складання і зміни списку дипломних робіт. Система надає можливість більш зручної віддаленої роботи з студентами і можливість стежити за ходом дипломного проектування завідувачу кафедри.

### 2. Деталізація мети методом SMART

Продуктом дипломного проекту є хмарний клієнт-серверний додаток забезпечення документообігу для дипломного проектування. Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити хмарний клієнт-серверний додаток забезпечення документообігу для дипломного проектування.
Measurable (вимірювана)	Результатом роботи проекту є оцінка замовника.
Relevant (реалістична)	У наявності є всі необхідні технічні та програмні засоби. Розробники достатньо кваліфіковані для виконання поставлених задач.

## Продовження таблиці Б.1 – Деталізація мети методом SMART

Achievable (досяжна)	Клієнтська частина інформаційної системи здійснюється з використанням web технологій HTML, CSS та JavaScript. Серверна частина та логіка системи використовують мову PHP, базу даних MySQL, для зберігання файлів використовується хмарне сховище «Dropbox».
Time-framed (обмежена у часі)	Ціль має часове обмеження. Робота повинна бути виконана у терміни, що були оговорені замовником та виконавцем проекту. Проект повинен бути виконаний згідно з календарним планом.

### 3. Описання фази розробки ІТ – проекту

#### 3.1 Планування змісту структури робіт ІТ – проекту (WBS)

Основним інструментом для планування змісту структури робіт служить WBS діаграма – графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов’язані з продуктом проекту. Побудуємо структуру WBS, у якій детально опишемо роботи, які потрібно виконати на кожному етапі створення проекту. Виконаємо декомпозицію робіт для даного проекту. Діаграма WBS зображена на рис. Б.7.

#### 3.2 Планування структури організації, для впровадження готового проекту

Після побудови WBS розробляють організаційну структуру виконавців. Організаційна структура представляє собою графічне відображення учасників проекту та їх відповідальних осіб, які задіяні в реалізації проекту. Основними

учасниками розробки проекту виступають Парфененко Ю. В. у ролі куратора проекту, Александренко Т. В. у ролі розробника.

Таким чином на основі побудованої раніше WBS діаграми(рис. Б.7) будуюмо OBS- структуру проекту, яка зображена на рис. Б.8.

### 3.3 Розробка діаграми Ганта

Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі, а також, проекту в цілому з урахуванням вихідних та святкових днів, будують календарний графік робіт, що називається Діаграмою Ганта. Діаграма Ганта зображена на рис. Б.2 – Б.5.

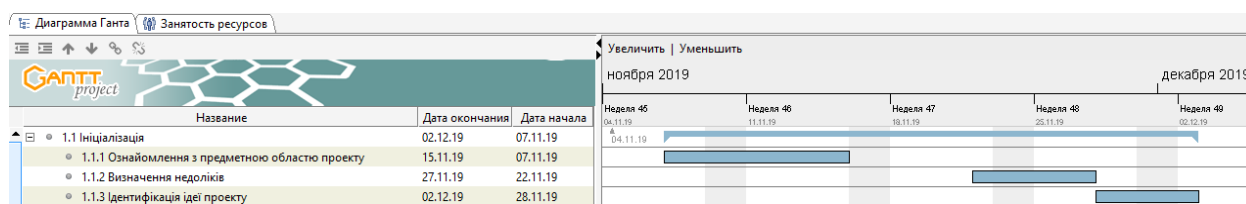


Рисунок Б.1 – Діаграма Ганта

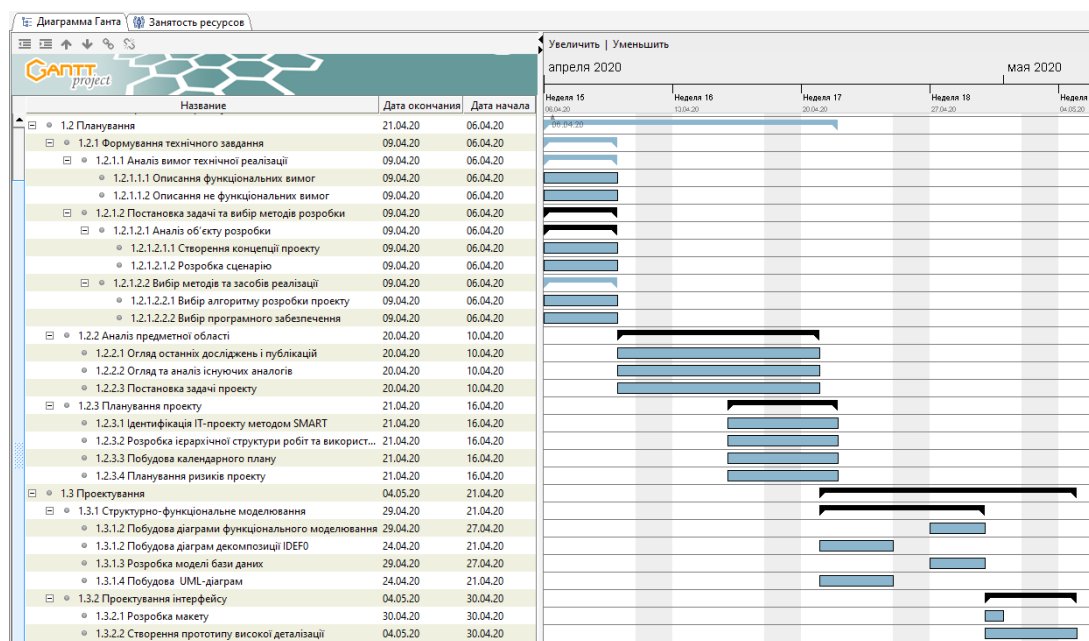


Рисунок Б.2 – Продовження діаграми Ганта

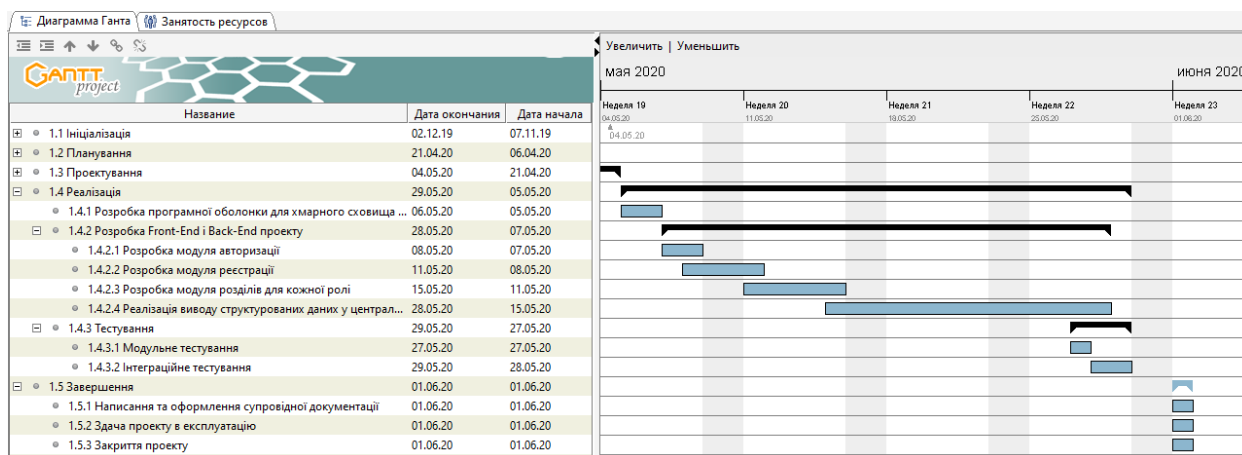


Рисунок Б.3 – Продовження діаграми Ганта

Назва	Дата початку	Дата закінчення
1.1 Ініціалізація	07.11.19	02.12.19
1.1.1 Ознайомлення з предметною областю проекту	07.11.19	15.11.19
1.1.2 Визначення недоліків	22.11.19	27.11.19
1.1.3 Ідентифікація ідей проекту	28.11.19	02.12.19
1.2 Планування	06.04.20	21.04.20
1.2.1 Формування технічного завдання	06.04.20	09.04.20
1.2.1.1 Аналіз вимог технічної реалізації	06.04.20	09.04.20
1.2.1.1.1 Описання функціональних вимог	06.04.20	09.04.20
1.2.1.1.2 Описання не функціональних вимог	06.04.20	09.04.20
1.2.1.2 Постановка задачі та вибір методів розробки	06.04.20	09.04.20
1.2.1.2.1 Аналіз об'єкту розробки	06.04.20	09.04.20
1.2.1.2.1.1 Створення концепції проекту	06.04.20	09.04.20
1.2.1.2.1.2 Розробка сценарію	06.04.20	09.04.20
1.2.1.2.2 Вибір методів та засобів реалізації	06.04.20	09.04.20
1.2.1.2.2.1 Вибір алгоритму розробки проекту	06.04.20	09.04.20
1.2.1.2.2.2 Вибір програмного забезпечення	06.04.20	09.04.20
1.2.2 Аналіз предметної області	10.04.20	20.04.20
1.2.2.1 Огляд останніх досліджень і публікацій	10.04.20	20.04.20
1.2.2.2 Огляд та аналіз існуючих аналогів	10.04.20	20.04.20
1.2.2.3 Постановка задачі проекту	10.04.20	20.04.20
1.2.3 Планування проекту	16.04.20	21.04.20
1.2.3.1 Ідентифікація IT-проекту методом SMART	16.04.20	21.04.20
1.2.3.2 Розробка ієрархічної структури робіт та використання WBS\OBS	16.04.20	21.04.20
1.2.3.3 Побудова календарного плану	16.04.20	21.04.20
1.2.3.4 Планування ризиків проекту	16.04.20	21.04.20
1.3 Проектування	21.04.20	04.05.20
1.3.1 Структурно-функціональне моделювання	21.04.20	29.04.20
1.3.1.2 Побудова діаграми функціонального моделювання	21.04.20	29.04.20
1.3.1.2 Побудова діаграм декомпозиції IDEF0	21.04.20	24.04.20
1.3.1.3 Розробка моделі бази даних	27.04.20	29.04.20
1.3.1.4 Побудова UML-діаграм	21.04.20	24.04.20
1.3.2 Проектування інтерфейсу	30.04.20	04.05.20
1.3.2.1 Розробка макету	30.04.20	30.04.20
1.3.2.2 Створення прототипу високої деталізації	30.04.20	04.05.20

Рисунок Б.4 – Список робіт для побудови діаграми Ганта



☐ ● 1.4 Реалізація	29.05.20	05.05.20
● 1.4.1 Розробка програмної оболонки для хмарного сховища Dropbox API	06.05.20	05.05.20
☐ ● 1.4.2 Розробка Front-End і Back-End проекту	28.05.20	07.05.20
● 1.4.2.1 Розробка модуля авторизації	08.05.20	07.05.20
● 1.4.2.2 Розробка модуля реєстрації	11.05.20	08.05.20
● 1.4.2.3 Розробка модуля розділів для кожної ролі	15.05.20	11.05.20
● 1.4.2.4 Реалізація виводу структурованих даних у центральній частині сайту	28.05.20	15.05.20
☐ ● 1.4.3 Тестування	29.05.20	27.05.20
● 1.4.3.1 Модульне тестування	27.05.20	27.05.20
● 1.4.3.2 Інтеграційне тестування	29.05.20	28.05.20
☐ ● 1.5 Завершення	01.06.20	01.06.20
● 1.5.1 Написання та оформлення супровідної документації	01.06.20	01.06.20
● 1.5.2 Задача проекту в експлуатацію	01.06.20	01.06.20
● 1.5.3 Закриття проекту	01.06.20	01.06.20

Рисунок Б.5 – Продовження списку робіт для побудови діаграми Ганта

### 3.4 Управління ризиками проекту

Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків. У табл. Б.2 знаходиться класифікація ризиків за показниками ймовірності виникнення ризику та величині втрат.

Далі виконаємо планування реагування на ризики — це розробка методів і технологій зниження негативного впливу ризиків на проект. Визначимо ефективність розробки реагування на проект, визначимо чи будуть наслідки впливу ризику на проект позитивними або негативним. Оцінюємо ризики за показниками, що знаходяться в табл. Б.2. На основі оцінки будуємо матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рис. Б.5.

Таблиця Б.2 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

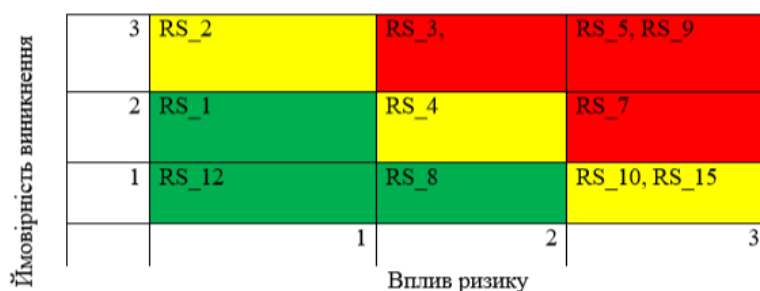


Рисунок Б.5 – Матриця ймовірності виникнення ризиків та впливу ризику

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. Б.3. Оцінка ймовірності виникнення, впливу і рангу ризику наведена у таблиці Б.4.

Таблиця Б.3 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номер)
1	Прийнятні	$1 \leq R \leq 2$	1,8,11,12,13
2	Виправдані	$3 \leq R \leq 4$	2,4,6,10,15
3	Недопустимі	$6 \leq R \leq 9$	3,5,7,9,14

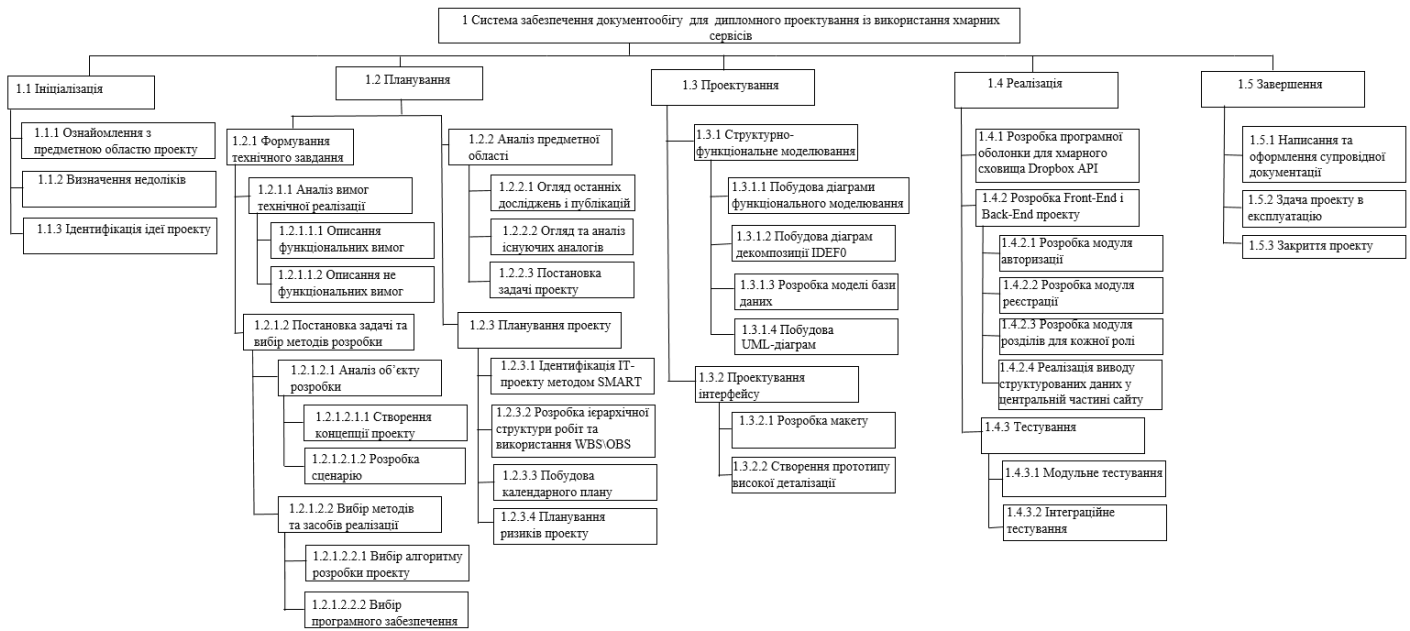


Рисунок Б.7 – WBS. Структура робіт проекту

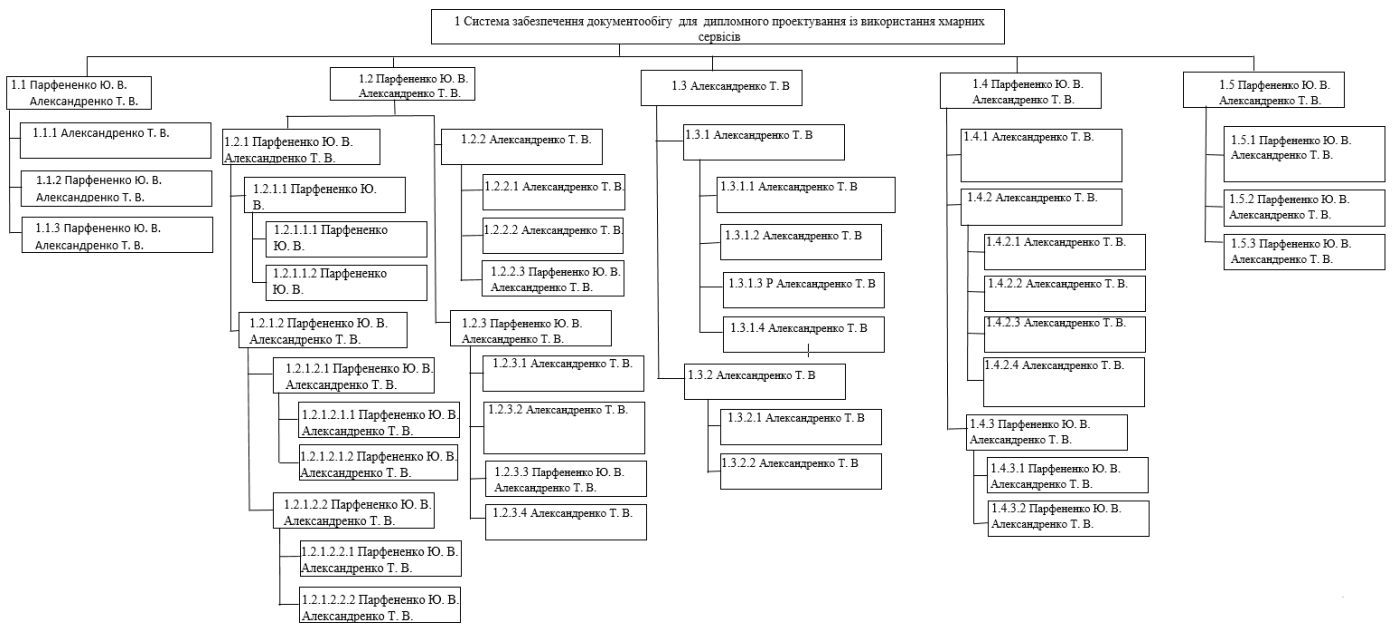


Рисунок Б.8 – Організаційна структура проекту (OBS)

Таблиця Б.4 – Оцінка ймовірності виникнення, впливу і рангу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Непорозуміння між розробником та замовником	Низька	Середній	2	1. Дотримуватися ділового етикету спілкування. 2. Створити комфортні умови для співпраці	Попередження	Потрібно в'яснити, що саме стало причиною непорозуміння обговорити її та створити здорову атмосферу в колективі.
RS_2	Відкритий	Поява альтернативного продукту	Низька	Високій	3	1. Провести аналіз ринку на наявність альтернативних продуктів	Прийняття	
RS_3	Відкритий	Нечітке завдання на розробку	Середня	Високій	6	1. Ясно і однозначно обговорити із замовником усі види вимог і задач програмного продукту.	Попередження	Уважно та чітко окреслити те, що було виконано невірно та зробити правки
RS_4	Відкритий	Низька кваліфікація розробників в проекту	Середня	Середній	4	1. Підвищити кваліфікацію персоналу. 2. Переглянути онлайн-ресурси для підвищення рівня знань.	Пом'якшення	Врахувати час на підготовку працівників. Видати літературу, переглянути онлайн-уроки.

Продовження таблиці Б.4– Оцінка ймовірності виникнення, впливу і рангу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_5	Відкритий	Неоптимальний розподіл часу	Висока	Високій	9	Провести аналіз актуальності найважливіших процесів та робіт, розрахувати час необхідний для їхнього виконання. Створити календарний план з урахуванням резервного строку виправлення помилок і чітко його дотримуватися	Пом'якшення	Змінити порядок пріоритетів в робіт. Знайти способи оптимізації роботи із вже існуючою розстановкою. Обговорити варіанти внесення правок до термінів реалізації із замовником.
RS_6	Відкритий	Не вірна оцінка масштабів проекту	Низька	Середній	2	Провести детальний аналіз проекту. Визначити основні етапи проекту, розподілити час на їх виконання. Проаналізувати масштаби проекту на основі додаткових джерел.	Пом'якшення	Переоцінка масштабів проекту. Перебудова стратегії реалізації проекту.

Продовження таблиці Б.4– Оцінка ймовірності виникнення, впливу і рангу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_7	Відкритий	Помилки розробки	Висока	Високим	9	На етапі розробки тісно співпрацювати із замовником та на певних етапах демонструвати поточні результати.	Пом'якшення	Здійснювати проміжний контроль результатів в ході виконання проекту.
RS_8	Відкритий	Збої в роботі програмного забезпечення	Низька	Високим	3	Залучити спеціаліста для усунення збоїв.	Попередження	Замінити програмне забезпечення.

## ДОДАТОК В. ЛІСТИНГ ПРОГРАМНОГО КОДУ

### *Dropbox.php*

```

<?php require_once("dbconnect.php");
require("vendor/autoload.php");
use Kunnu\Dropbox\{
    Dropbox,
    DropboxApp,
    DropboxFile};
ob_start();
$app = new DropboxApp(
    'v7gaslt29ri8kme',
    'gk3yct7j1nycn3s',
    'k4FsGpRWTkAAAAAFAF3hVRk66Lg0HHJZK-arPEI5qdDHxLTCs3sTbr5y_htEJ');
$redirect_uri = "http://{$_SERVER['HTTP_HOST']}";
$dropbox = new Dropbox($app);
$authHelper = $dropbox->getAuthHelper();
if (isset($_REQUEST['code']) && isset($_REQUEST['state'])) {
    $code = $_REQUEST['code'];
    $state = $_REQUEST['state'];
    $accessToken = $authHelper->getAccessToken($code, $state, $redirect_uri);
    $_SESSION['accessToken'] = $accessToken->getToken();
    header('Location:' . filter_var($redirect_uri, FILTER_SANITIZE_URL));
} elseif (!isset($_SESSION['accessToken'])) {
    header('Location:' . filter_var($authHelper->getAuthUrl($redirect_uri, FILTER_SANITIZE_URL));
}
}
$tree = makeTree($dropbox, "");
echo '<style type="text/css">
    ul > li {
        list-style: none;
    }
    ul > li > img {
        margin-right: 3px;
    }
    .icon-new {
        display: inline-block;
        width: 16px;
        height: 16px;
    }
    .icon-upload{
        display: inline-block;
        width: 16px;
        height: 16px;
    }
    .icon-delete {
        display: inline-block;
        width: 16px;
        height: 16px;
    }
</style>';
printTree($tree);
if (isset($_GET['action'])) {
    switch ($_GET['action']) {
        // Создание новой папки
        case 'newdir':
            if (isset($_GET['pid'])) {
                include 'form-create-dir.php';
                if ($_POST) {
                    if (!empty($_POST['newFolder'])) {
                        $suffixPath =
                            urldecode(base64_decode(htmlspecialchars($_GET['pid'])));
                        $dirName = htmlspecialchars($_POST['newFolder']);
                        $folder = $dropbox->createFolder("$suffixPath/$dirName");
                    }
                }
            }
        }
    }
}

```

```

                                header('Location:'. filter_var($redirect_uri,
FILTER_SANITIZE_URL));} }}
                                case 'upload':
                                    if (isset($_GET['fid'])) {
                                        include 'form-upload-file.php';
                                        if (isset($_FILES['Files']) && is_uploaded_file($_FILES['Files']['tmp_name'][0])
&& ($_FILES['Files']['error'][0] == 0)) {
                                            $files = normalizeFilesArray($_FILES);
                                            $affixPath = urldecode(base64_decode(htmlspecialchars($_GET['fid'])));
                                            foreach ($files as $file) {
                                                $file = $dropbox->upload(
DropboxFile::createByStream("$affixPath/{ $file['name']}", file_get_contents($file['tmp_name']),
"$affixPath/{ $file['name']}",
['autorename' => true]);}
                                            header('Location:'. filter_var($redirect_uri,
FILTER_SANITIZE_URL));} }break;
                                case 'delete':
                                    if (isset($_GET['fid'])) {
                                        $deletedFolder = $dropbox-
>delete(urldecode(base64_decode(htmlspecialchars($_GET['fid']))));
                                        header('Location:'. filter_var($redirect_uri, FILTER_SANITIZE_URL));}
                                        break;}}
                                ob_end_flush();
                                function makeTree(Dropbox &$dropbox, $path){
                                    $branch = [];
                                    $listFolderContents = $dropbox->listFolder($path);
                                    $items = $listFolderContents->getItems();
                                    $listFiles = $items->toArray();
                                    foreach ($listFiles as $k => $item) {
                                        $branch[$k] = [
                                            'id' => substr($item->id, 3),
                                            'name' => $item->name,
                                            'path' => $item->path_display,
                                            'type' => $item->getData()['tag'];
                                        if (isset($item->getData()['server_modified'])) {
                                            $branch[$k]['created'] = $item->getData()['server_modified'];}
                                        if ($item->size) {
                                            $branch[$k]['size'] = $item->size;}
                                        if ($item->getData()['tag'] == 'folder') {
                                            $branch[$k]['children'] = makeTree($dropbox, $item->path_lower);}
                                        return $branch;}
                                function printTree($tree){
                                    if (!is_null($tree) && count($tree) > 0) {
                                        echo '<ul>';
                                        foreach ($tree as $kn => $node) {
                                            echo '<li>';
                                            if ($node['type'] == 'folder') {
                                                //if($query == $node['path']){
                                                    $url = 'https://www.dropbox.com/home' . $node['path'];}
                                                if ($node['type'] == 'file') {
                                                    $url = 'https://www.dropbox.com/preview' . $node['path'] . '?role=personal';}
                                                echo '<a href="' . $url . '">';
                                                if ($node['type'] == 'folder') {
                                                    echo '<b>' . $node['name'] . '</b>';
                                                } else {
                                                    echo $node['name'];}
                                                echo '</a> ';
                                                if ($node['type'] == 'file') {
                                                    echo date('d.m.Y H:i:s', strtotime($node['created']));}
                                                if ($node['type'] == 'folder') {
                                                    echo ' <a href="?action=newdir&pid=' . base64_encode(urlencode($node['path'])) . '"
title="Новая папка"><i class="icon-new"></i></a>';}
                                                    echo ' <a href="?action=upload&fid=' . base64_encode(urlencode($node['path'])) . '"
title="Загрузить файл(ы)"><i class="icon-upload"></i></a>';

```



```

                echo ' <a href="?action=delete&fid=' . base64_encode(urlencode($node['path'])) . "'
title="Удалить файл(ы)" onclick="return confirm('\Вы действительно хотите удалить данный файл?');"><i
class="icon-delete"></i></a>';
                if (isset($node['children'])) printTree($node['children']);
                echo '</li>;'
            }
        }
    }
}
function normalizeFilesArray($files = []){
    $result = [];
    foreach ($files as $file) {
        if (!is_array($file['name'])) {
            $result[] = $file;
            continue;
        }
        foreach ($file['name'] as $idx => $name) {
            $result[$idx] = [
                'name' => $name,
                'type' => $file['type'][$idx],
                'tmp_name' => $file['tmp_name'][$idx],
                'error' => $file['error'][$idx],
                'size' => $file['size'][$idx];
            ];
        }
    }
    return $result;
}
function varDumper($data){
    echo '<pre>';
    print_r($data);
    echo '</pre>';
}

```

### *dropbox\_sqlbackup.php*

```

$accessToken = 'YOUR_DROPBOX_ACCESS_TOKEN';
$tmpDir = "/home/php/temp_dir/"; // Location of your temp directory
$user = "DB_USER"; // Username for database
$password = "DB_PASSWORD"; // Password for database
$unix_socket = "/opt/mariadb-data/mariadb.sock";
$dbName = "DB_NAME"; // Database name to backup
$dbHost = "DB_HOST"; // Hostname or IP address where database resides (localhost or xxx.xxx.xxx.xxx)
$prefix = "backup_db_"; // The compressed file will have this prefix
// Create the database backup file
$sqlFile = $tmpDir.$prefix.date('Y_m_d_H:i:s').".sql"; // Name of the .sql file we create for the database
$backupFilename = $prefix.date('Y_m_d_H:i:s').".tgz"; // Name of the .tgz backup file we upload to Dropbox
$backupFile = $tmpDir.$backupFilename; // Path to the backup file in the temp directory
// Shell commands for generating the .sql and .tgz files
// $createBackup = "mysqldump -h ".$dbHost." -u ".$user." --password=".$password." ".$dbName." --&gt;
".$sqlFile; // Shell command to create the .sql file for MySQL
// $createBackup = "mysqldump -h ".$dbHost." -u ".$user." --password=".$password." ".$dbName." --
socket=".$unix_socket." --&gt; ".$sqlFile; // Shell command to create the .sql file for MariaDB
// $createZip = "tar -cvzf $backupFile $sqlFile"; // Shell command to create the backup file we upload to Dropbox
try {
    exec($createBackup); // Run shell command to create the .sql file
    exec($createZip); // Run shell command to create the .tgz file
} catch (Exception $e) {
    echo "Failed to create CRM DB Backup or ZIP: " . $e-&gt;getMessage() . "\n";
}
// Upload the file to Dropbox
$file_path = $backupFile; // The path to the file on the local machine
$dbbox_file_path = "/DROPBOX_DIRECTORY/"; // The path in DropBox where we will upload the file
DropboxUploadLargeFile($file_path, $dbbox_file_path);
// Delete the temporary files
try {
    unlink($sqlFile);
    unlink($backupFile);
} catch (Exception $e) {
    echo "Failed to unlink DB Backup temporary file: " . $e-&gt;getMessage() . "\n";
}
// Functions for uploading files of any size

```

```

function DropboxUploadLargeFile($file_path, $dbox_file_path) { // Large is &gt; 150MB
    $session_ID = GetUploadSessionID(); // Get the session ID that we will use for uploading the file
    $filesize = filesize($file_path); // Get the file size
    $file_name = basename($file_path); // Get the name of the file (i.e. database.db, compressed.zip,
excel.csv, etc.)
    UploadFileInChunks($session_ID, $file_path); // Upload the file
    FinishUploadingChunks($session_ID, $filesize, $file_name, $dbox_file_path); // Finish uploading the file}
function GetUploadSessionID() {
    global $accessToken; // Get the global access token value
    $session_ID = ""; // Declare our session ID and initialize it to an empty string
    $dropbox_endpoint_url = "https://content.dropboxapi.com/2/files/upload_session/start"; // Dropbox url for
retrieving a session ID
    $arguments = '{
        "close": false } ';
    // Encode the url parameters
    $url = $dropbox_endpoint_url . "?authorization=" . urlencode("Bearer " . $accessToken) . "&arg=" .
urlencode($arguments);
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/octet-stream', ));
    curl_setopt($ch, CURLOPT_POST, true); // Tell CURL to use HTTP POST
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); // Tell CURL that we want the data to be
returned in a way that can be assigned to our $response variable
    $response = curl_exec($ch); // Obtain response
    curl_close($ch); // Close the CURL connection
    $session_ID = json_decode($response, true)['session_id']; // Parse the JSON encoded array to get the
session ID
    return ($session_ID); // Return the session ID }
function UploadFileInChunks($session_ID, $file_path) {
    $chunk_size = 1024 * 1024 * 140; // 140 MB (1024B = 1KB, 1024KB = 1MB)
    $file = fopen($file_path, "r"); // Open the file as read-only
    $loop_count = 0;
    while (!feof($file)) { // While not at the end of file, upload the current chunk
        $buffer = fread($file, $chunk_size); // Read the current chunk of the file into a buffer
        $offset = $loop_count * $chunk_size; // Define the offset of the chunk
        UploadFileChunk($session_ID, $offset, $buffer); // Upload the current chunk
        $loop_count++; // Increment the loop count }
    fclose($file); // Close the file }
function UploadFileChunk($session_ID, $offset, $file_chunk) {
    global $accessToken; // Get the global access token value
    $dropbox_endpoint_url = "https://content.dropboxapi.com/2/files/upload_session/append_v2"; // Dropbox
upload endpoint
    // The arguments require the session ID and the chunk offset
    $arguments = ' {
        "cursor": {
            "session_id": ".$session_ID.",
            "offset": '.$offset.' },
        "close": false } ';
    // Encode the url parameters
    $url = $dropbox_endpoint_url . "?authorization=" . urlencode("Bearer " . $accessToken) . "&arg=" .
urlencode($arguments);
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/octet-stream', ));
    curl_setopt($ch, CURLOPT_POST, true); // Tell curl to use HTTP POST
    curl_setopt($ch, CURLOPT_POSTFIELDS, $file_chunk); // Write the file chunk into the post stream
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); // Tell CURL that we want the data to be
returned in a way that can be assigned to our $response variable
    $response = curl_exec($ch); // Obtain response
    curl_close($ch); // Close the CURL connection }
function FinishUploadingChunks($session_ID, $offset, $file_name, $dbox_file_path) {
    global $accessToken; // Get the global access token value
    $dropbox_endpoint_url = "https://content.dropboxapi.com/2/files/upload_session/finish"; // Dropbox finish

```

```

upload endpoint
    // The arguments require the session ID and the chunk size
    $arguments = '{
        "cursor": {
            "session_id": ".$session_ID.",
            "offset": '.$offset.',
        }
        "commit": {
            "path": ".$dbox_file_path.$file_name.",
            "mode": "add" } } ';
    // Encode the url parameters
    $url = $dropbox_endpoint_url . "?authorization=" . urlencode("Bearer " . $accessToken) . "&arg=" .
urlencode($arguments);
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/octet-stream', ));
    curl_setopt($ch, CURLOPT_POST, true); // Tell curl to use HTTP POST
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); // Tell CURL that we want the data to be returned in
a way that can be assigned to our $response variable
    $response = curl_exec($ch); // Obtain response
    curl_close($ch); // Close the CURL connection }
function DropboxUploadFile($file_path, $dbox_file_path) { // &lt; 150MB
global $accessToken;
    $dropbox_endpoint_url = "https://content.dropboxapi.com/2/files/upload"; // Dropbox upload endpoint
    $file_name = basename($file_path);
    $arguments = ' {
        "path": ".$dbox_file_path.$file_name.",
        "mode": "add" } ';
    $url = $dropbox_endpoint_url . "?authorization=" . urlencode("Bearer " . $accessToken) . "&arg=" .
urlencode($arguments);
    $file_to_upload = fopen($file_path, 'rb'); // Open the file
    $filesize = filesize($file_path); // Get the file size
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/octet-stream',));
    curl_setopt($ch, CURLOPT_POST, true); // Tell curl to use HTTP POST
    curl_setopt($ch, CURLOPT_POSTFIELDS, fread($file_to_upload, $filesize)); // Write the file into the
post stream
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch); // Obtain response
    curl_close($ch);
    fclose($file_to_upload); // Close the file }

```

### *dbconnect.php*

```

<?php
    header('Content-Type: text/html; charset=utf-8');
    $server = "localhost";
    $username = "root";
    $password = "";
    $database = "start";
    $mysqli = new mysqli($server, $username, $password, $database);
    if (mysqli_connect_errno()) {
        echo "<p><strong>Помилка підключення до БД</strong> ".mysqli_connect_error()."</p>";
        exit(); }
    $mysqli->set_charset('utf8');
    $address_site = "http://localhost/index.php";
    $email_admin = "admin@sozdatisite.ru"?>

```

### *form\_auth.php*

```

<?php require_once("header.php");      ?>
<script type="text/javascript">
$(document).ready(function(){
    "use strict";
    var pattern = /^[a-z0-9][a-z0-9\._-]*[a-z0-9]*@[([a-z0-9]+([a-z0-9-]*[a-z0-9]+)*\.)+[a-z]+/i;
    var mail = $('input[name=email]');
    mail.blur(function(){
        if(mail.val() != ""){
            if(mail.val().search(pattern) == 0){
                $('#valid_email_message').text("");
                $('input[type=submit]').attr('disabled', false);
            }else{
                $('#valid_email_message').text('Не правильний Email');
                $('input[type=submit]').attr('disabled', true);}
        }else{
            $('#valid_email_message').text('Введіть Ваш email');});
    var password = $('input[name=password]');
    password.blur(function(){
        if(password.val() != ""){
            if(password.val().length < 6){
                $('#valid_password_message').text('Мінімальна довжина пароля 6 символів');
                $('input[type=submit]').attr('disabled', true);
            }else{
                $('#valid_password_message').text("");
                $('input[type=submit]').attr('disabled', false);
            }
        }else{
            $('#valid_password_message').text('Введіть пароль');}); });
</script>
<div class="block_for_messages">
<?php
if(isset($_SESSION["error_messages"]) && !empty($_SESSION["error_messages"])){
    echo $_SESSION["error_messages"];
    unset($_SESSION["error_messages"]);
}
if(isset($_SESSION["success_messages"]) && !empty($_SESSION["success_messages"])){
    echo $_SESSION["success_messages"];
    unset($_SESSION["success_messages"]); } ?></div>
<?php
if(!isset($_SESSION["email"]) && !isset($_SESSION["password"])){ ?>
<div class="container1">
<section id="content">
<form action="auth.php" method="post" name="form_auth" >
<h1>АВТОРИЗАЦІЯ</h1>
<table><tr><td> Email: </td><td>
        <input type="email" name="email" required="required" /><br />
        <span id="valid_email_message" class="message_error"></span></td></tr><tr>
<td> Пароль: </td><td>
        <input type="password" name="password" placeholder="мінімум 6 символів" required="required" /><br />
        <span id="valid_password_message" class="message_error"></span></td></tr>
<tr><td> Введіть капчу: </td><td><p>
         <br />
        <input type="text" name="captcha" placeholder="Код перевірки" /> </p></td></tr> </table><div>
        <input type="submit" name="btn_submit_auth" value="Увійти" />
        <a href="reset_password.php">Забули пароль?</a></div></form>
</section><!-- content --></div>
<?php }else{ ?>
<div id="authorized">
<h2>Ви вже авторизовані!</h2></div>
<?php }?>

```

### auth.php

```

<?php
session_start();

```



```

        exit("<p><strong>Помилка!</strong> Збій при видаленні простроченого аккаунта. Код помилки:
".$mysqli->errno."</p>");}
        $result_query_select = $mysqli->query("SELECT * FROM `users` WHERE email = '". $email.'" AND password
= '". $password."'");
        if(!$result_query_select){
            $_SESSION["error_messages"] .= "<p class='message_error' >Помилка запиту на виборі користувача з
БД</p>";
            header("HTTP/1.1 301 Moved Permanently");
            header("Location:form_auth.php");
            exit();
        }else{
            if($result_query_select->num_rows == 1){
                while(($row = $result_query_select->fetch_assoc()) !=false){
                    if($row["email_status"] == 0){
                        $_SESSION["error_messages"] = "<p class='message_error' >Ви зареєстровані, але, Ваш поштовий
адрес не підтверджений. Для підтвердження пошти перейдіть за посиланням з листа, яку отримали після
реєстрації.</p>
                        <p><strong>Увага!</strong> Посилання для підтвердження пошти, дійсна 24 години з моменту
реєстрації. Якщо Ви не підтвердите Ваш email протягом цього часу, то Ваш аккаунт буде видалений.</p>";
                        header("HTTP/1.1 301 Moved Permanently");
                        header("Location:form_auth.php");
                        exit();
                    }else{
                        $_SESSION['email'] = $email;
                        $_SESSION['password'] = $password;
                        if ($row["id_role"] == 1){
                            header("HTTP/1.1 301 Moved Permanently");
                            header("Location:index.php");}
                        else if ($row["id_role"] == 2){
                            header("HTTP/1.1 301 Moved Permanently");
                            header("Location:admin.php");}
                        else if($row["id_role"] == 3 || $row["id_role"] == 5){
                            header("HTTP/1.1 301 Moved Permanently");
                            header("Location:teacher.php");}
                        else if($row["id_role"] == 4){
                            header("HTTP/1.1 301 Moved Permanently");
                            header("Location:clerk.php");}exit();} }
                    }else{
                        $_SESSION["error_messages"] .= "<p class='message_error' >Неправильний логін і / або пароль</p>";
                        header("HTTP/1.1 301 Moved Permanently");
                        header("Location:form_auth.php");exit();} }
                }else{
                    exit("<p><strong>Помилка!</strong> Відсутній код перевірки, тобто код капчі. Ви можете перейти на <a
href='".$address_site."> головну сторінку </a>.</p>");}

            }else{
                exit("<p><strong>Помилка!</strong> Ви зайшли на цю сторінку безпосередньо, тому немає даних для
обробки. Ви можете перейти на <a href='".$address_site."> головну сторінку </a>.</p>");
            }
        }

```

### register.php

```

<?php
session_start();
require_once("dbconnect.php");
$_SESSION["error_messages"] = "";
$_SESSION["success_messages"] = "";
if(isset($_POST["btn_submit_register"]) && !empty($_POST["btn_submit_register"])){
    $im = 'Avatars/' . time() . $_FILES['avatar']['name'];
    if (!move_uploaded_file($_FILES['avatar']['tmp_name'], $im )){
        $error_message = "<p class='message_error'><strong>Помилка!</strong>Фото не завантажилось</p>";
        $_SESSION["error_messages"] = $error_message;
        header("HTTP/1.1 301 Moved Permanently");
        header("Location:form_register.php");}
}

```

```

$id_group = $_POST["id_group"];
$id_teacher = $_POST["id_teacher"];
$captcha = trim($_POST["captcha"]);
if(isset($_POST["captcha"]) && !empty($captcha)){
    if(($_SESSION["rand"] != $captcha) && ($_SESSION["rand"] != "")){
        $error_message = "<p class='message_error'><strong>Помилка!</strong>Неправильна капча</p>";
        $_SESSION["error_messages"] = $error_message;
        header("HTTP/1.1 301 Moved Permanently");
        header("Location:form_register.php");
        exit();}
    if(isset($_POST["first_name"])){
        $first_name = trim($_POST["first_name"]);
        if(!empty($first_name)){
            $first_name = htmlspecialchars($first_name, ENT_QUOTES);
        }else{
            $_SESSION["error_messages"] .= "<p class='message_error'>Вкажіть Ваше ім'я</p>";
            header("HTTP/1.1 301 Moved Permanently");
            header("Location:form_register.php");
            exit();}
        }else{
            $_SESSION["error_messages"] .= "<p class='message_error'>Відсутня поле з ім'ям</p>";
            header("HTTP/1.1 301 Moved Permanently");
            header("Location:form_register.php");
            exit();}
        if(isset($_POST["last_name"])){
            $last_name = trim($_POST["last_name"]);
            if(!empty($last_name)){
                $last_name = htmlspecialchars($last_name, ENT_QUOTES);
            }else{
                $_SESSION["error_messages"] .= "<p class='message_error' >Вкажіть Ваше прізвище</p>";
                header("HTTP/1.1 301 Moved Permanently");
                header("Location:form_register.php");
                exit();}
            }else{
                $_SESSION["error_messages"] .= "<p class='message_error' >Відсутнє поле з прізвищем</p>";
                header("HTTP/1.1 301 Moved Permanently");
                header("Location:form_register.php");
                exit();}
            if(isset($_POST["middle_name"])){
                $middle_name = trim($_POST["middle_name"]);
                if(!empty($middle_name)){
                    $middle_name = htmlspecialchars($middle_name, ENT_QUOTES);
                }else{
                    $_SESSION["error_messages"] .= "<p class='message_error' >Вкажіть Ваше ім'я по батькові</p>";
                    header("HTTP/1.1 301 Moved Permanently");
                    header("Location:form_register.php");
                    exit();
                }
            }else{
                $_SESSION["error_messages"] .= "<p class='message_error' >Відсутнє поле з по батькові</p>";
                header("HTTP/1.1 301 Moved Permanently");
                header("Location:form_register.php");
                exit();}
                if(isset($_POST["theme"])){
                    $theme = trim($_POST["theme"]);
                    if(!empty($theme)){
                        $theme = htmlspecialchars($theme, ENT_QUOTES);
                    }else{
                        $_SESSION["error_messages"] .= "<p class='message_error' >Вкажіть Вашу тему</p>";
                        header("HTTP/1.1 301 Moved Permanently");
                        header("Location:form_register.php");
                        exit();}
                    }else{
                        $_SESSION["error_messages"] .= "<p class='message_error' >Відсутнє поле з темою</p>";
                        header("HTTP/1.1 301 Moved Permanently");

```

```

header("Location:form_register.php");
exit();}
if(isset($_POST["email"])){
$email = trim($_POST["email"]);
if(!empty($email)){
$email = htmlspecialchars($email, ENT_QUOTES);
$reg_email = "/^[a-z0-9][a-z0-9\._-]*[a-z0-9]*@[a-z0-9]+([a-z0-9-]*[a-z0-9]+)*\.[a-z]+/i";
if( !preg_match($reg_email, $email)){
$SESSION["error_messages"] .= "<p class='message_error' >Ви ввели неправильний email</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");
exit();}
$result_query = $mysqli->query("SELECT `email` FROM `users` WHERE `email`='". $email. "'");
if($result_query->num_rows == 1){

if(($row = $result_query->fetch_assoc()) != false){
$SESSION["error_messages"] .= "<p class='message_error' >Користувач з такою поштовою адресою вже зареєстрований</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");
}else{
$SESSION["error_messages"] .= "<p class='message_error' >Помилка в запиті до БД</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");}
$result_query->close();
exit();}
$result_query->close();
}else{
$SESSION["error_messages"] .= "<p class='message_error' >Вкажіть Ваш email</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");
exit();}
}else{
$SESSION["error_messages"] .= "<p class='message_error' >Відсутнє поле для введення Email</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");exit();}
if(isset($_POST["password"])){
$password = trim($_POST["password"]);
if(isset($_POST["confirm_password"])){
$confirm_password = trim($_POST["confirm_password"]);
if($confirm_password != $password){
$SESSION["error_messages"] .= "<p class='message_error' >Паролі не співпадають</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");
exit();}
}else{
$SESSION["error_messages"] .= "<p class='message_error' >Відсутнє поле для повторення пароля</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");
exit();}
if(!empty($password)){
$password = htmlspecialchars($password, ENT_QUOTES);
$password = md5($password."top_secret");
}else{
$SESSION["error_messages"] .= "<p class='message_error' >Вкажіть Ваш пароль</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");exit();}
}else{
$SESSION["error_messages"] .= "<p class='message_error' >Відсутнє поле для введення пароля</p>";
header("HTTP/1.1 301 Moved Permanently");
header("Location:form_register.php");exit();}
$query_delete_users = $mysqli->query("DELETE FROM `users` WHERE `email_status` = 0 AND `date_registration` < ( NOW() - INTERVAL 1 DAY )");
if(!$query_delete_users){

```



```

        exit("<p><strong>Помилка!</strong> Збій при видаленні простроченого аккаунта. Код
помилки: ".$mysqli->errno."</p>");
    }
    $result_query_insert = $mysqli->query("INSERT INTO `users` (first_name, last_name,middle_name, id_group,
    id_teacher,theme, avatar, email, password, date_registration) VALUES ('".$first_name."', " ."$last_name.",
    " ."$middle_name.", " ."$id_group.", " ."$id_teacher.", " ."$theme.", " ."$im.", " ."$email.", " ."$password.", NOW());");
    if(!$result_query_insert){
        $_SESSION["error_messages"] .= "<p class='message_error' >Помилка запиту на додавання користувача в
БД</p>";
        header("HTTP/1.1 301 Moved Permanently");
        header("Location:form_register.php");
        exit();}else{
        $query_delete_confirm_users = $mysqli->query("DELETE FROM `confirm_users` WHERE
`date_registration` < ( NOW() - INTERVAL 1 DAY)");
        if(!$query_delete_confirm_users){
            exit("<p><strong>Помилка!</strong> Збій при видаленні простроченого аккаунта (confirm). Код
помилки: ".$mysqli->errno."</p>");}
        $token=md5($email.time());
        $query_insert_confirm = $mysqli->query("INSERT INTO `confirm_users` (email, token, date_registration)
VALUES ('".$email."', " ."$token.", NOW()) ");
        if(!$query_insert_confirm){
            $_SESSION["error_messages"] .= "<p class='message_error' >Помилка запиту на додавання користувача
в БД (confirm)</p>";
            header("HTTP/1.1 301 Moved Permanently");
            header("Location:form_register.php");
            exit();
        }else{
            $subject = "Підтвердження пошти на сайті ".$SERVER['HTTP_HOST'];
            $subject = "=?utf-8?B?".base64_encode($subject)."=?";
            $message = 'Вітаю! <br/> <br/> Сьогодні '.date("d.m.Y", time()).', якимось користувачем була проведена
реєстрація на сайті <a href=".'.$address_site.'">.'.$SERVER['HTTP_HOST'].</a> використовуючи Ваш email. Якщо
це були Ви, то, будь ласка, підтвердіть адресу вашої електронної пошти, перейшовши за цим посиланням: <a
href="http://localhost/activation.php?token='.$token.'&email='.$email.'"></a> <br/> <br/> В іншому випадку, якщо це
були не Ви, то, просто ігноруйте цей лист. <br/> <br/> <strong>Увага!</strong> Посилання дійсне 24 години.
Після чого Ваш аккаунт буде видалений з бази.';
            $headers = "FROM: $email_admin\r\nReply-to: $email_admin\r\nContent-type: text/html; charset=utf-
8\r\n";
            if(mail($email, $subject, $message, $headers)){
                $_SESSION["success_messages"] = "<h4 class='success message'><strong>Реєстрація пройшла
успішно!!!</strong></h4><p class='success message'> Тепер необхідно підтвердити зазначену адресу електронної
пошти. Для цього, перейдіть за посиланням зазначеному в повідомленні, який отримали на пошту ".$email."
</p>";
                header("HTTP/1.1 301 Moved Permanently");
                header("Location:form_register.php?hidden_form=1");
                exit();
            }else{
                $_SESSION["error_messages"] .= "<p class='message_error' >Помилка при відправленні листа з
посиланням підтвердження, на пошту ".$email."</p>"; }
                $result_query_insert->close();
                $query_insert_confirm->close(); } }

        $mysqli->close();
        header("HTTP/1.1 301 Moved Permanently");
        header("Location:form_register.php");
        exit(); }else{
        exit("<p><strong>Помилка!</strong> Відсутній код перевірки, тобто код капчі. Ви можете перейти на <a
href=".$address_site.'"> головну сторінку </a>.</p>");} }else{
        exit("<p><strong>Помилка!</strong> Ви зайшли на цю сторінку безпосередньо, тому немає даних для
обробки. Ви можете перейти на <a href=".$address_site.'"> головну сторінку </a>.</p>"); }?>

```

### reset\_password.php

```

<?phprequire_once("header.php");?>
<script type="text/javascript">

```

```

$(document).ready(function(){
    "use strict";
    var pattern = /^[a-z0-9][a-z0-9\._-]*[a-z0-9]*@[a-z0-9]+([a-z0-9]*[a-z0-9]+)*\.[a-z]+/i;
    var mail = $('input[name=email]');
    mail.blur(function(){
        if(mail.val() != ""){
            if(mail.val().search(pattern) == 0){
                $('#valid_email_message').text("");
                $('input[type=submit]').attr('disabled', false);
            }else{
                $('#valid_email_message').text('Не правильний Email');
                $('input[type=submit]').attr('disabled', true);} }else{
                $('#valid_email_message').text('Введіть Ваш email');});});</script>
<div class="block_for_messages">
    <?php
        if(isset($_SESSION["error_messages"]) && !empty($_SESSION["error_messages"])){
            echo $_SESSION["error_messages"];
            unset($_SESSION["error_messages"]);}
        if(isset($_SESSION["success_messages"]) && !empty($_SESSION["success_messages"])){
            echo $_SESSION["success_messages"];
            unset($_SESSION["success_messages"]);} ?></div>
    <?php
        if(!isset($_SESSION["email"]) && !isset($_SESSION["password"])) {
            if(!isset($_GET["hidden_form"])){ ?>
                <div class="container1">
                    <section id="content">
                        <p class="text_center message_error" id="valid_email_message"></p>
                        <form action="send_link_reset_password.php" method="post" name="form_request_email" >
                            <h1>Відновлення паролю</h1>
                            <table><tr><td> Введіть Ваш <br />E-mail: </td><td>
                                <input type="email" name="email" placeholder="" ></td></tr><tr>
                                    <td> Введіть капчу: </td><td><p>
                                         <br />
                                        <input type="text" name="captcha" placeholder="Код перевірки" /></p></td></tr><tr>
                                            <td colspan="2" class="text_center">
                                                <input type="submit" name="send" value="Відновити"></td></tr></table>
                            </form>
                        </section></div>
                    <?php }}else{ ?>
                        <div id="authorized">
                            <h2>Ви вже авторизовані!</h2>
                        </div>
                    <?php }?>

```

### header.php

```

<?php session_start();?>
<!DOCTYPE html>
<html>
<head>
    <title>Виконання дипломної роботи</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" type="text/css" href="css/style.css">
    <link rel="icon" href="Media/graduated.png" sizes="16x16" type="image/png">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</body> <header> <div class="container">
    <a href="index.php"></a>
    <?php
        if(!isset($_SESSION['email']) && !isset($_SESSION['password'])){ ?>
            <a href="form_register.php" class="button9" >РЕЄСТРАЦІЯ</a><div><div >
                <a href="form_auth.php" class="button10" >АВТОРИЗАЦІЯ</a></div> </div>
        <?php }else{ ?> <div >
            <a href="account.php"></a>
            <a href="logout.php"></a> </div><?php }?>

```

```

<div class="clear"></div>
</div>
</header>
</body></html>

```

## *index.php*

```

<?php session_start();require_once("dbconnect.php");?>
<!DOCTYPE html>
<html>
<head>
<title>Виконання дипломної роботи</title>
<meta charset="utf-8" />
<link rel="stylesheet" type="text/css" href="css/styles1.css">
<link rel="icon" href="Media/graduated.png" sizes="16x16" type="image/png">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head><body>
<header> <div class="container">
<a href="index.php"></a>
<?php
if(!isset($_SESSION['email']) && !isset($_SESSION['password'])){ ?>
<div class="reg">
<a href="form_register.php" class="button9" >РЕЄСТРАЦІЯ</a><div>
<div >
<a href="form_auth.php" class="button10" >АВТОРИЗАЦІЯ</a></div> </div> </div>
<?php
}else{ ?> <div >
<a href="account.php"></a>
<?php
$email = $_SESSION['email'];
$password = $_SESSION['password'];
$users = $mysqli->query("SELECT * FROM `users` WHERE email = '". $email.'" AND password =
"'. $password.'"");
$row = $users->fetch_assoc();?>
<p class="Name">
<?php
print_r($row["first_name"]);?>
<?php
print_r($row["last_name"]);?></p>
<a href="logout.php"></a> </div>
<div class="dropbox">
<?php include 'dropbox.php';?></div>
<div class="category-wrap">
<h3>Документи:</h3><ul>
<li><a href="pz.php">Пояснювальна записка</a></li>
<li><a href="protokol.php">Протокол перевірки на плагіат</a></li>
<li><a href="present.php">Файли презентації</a></li>
<li><a href="archive.php">Архів з програмним додатком</a></li>
<li><a href="other.php">Інша супровідна документація</a></li></ul></div>
<div class="category-wrap">
<h2 >Викладач:</h2>
<?php
$teacher_info = $mysqli->query("SELECT * FROM `users` WHERE id IN (select id_teacher from `users` where
id_teacher is NOT NULL and email = '". $email.'"");
$row1 = $teacher_info->fetch_assoc();?><p></p>
<img src = "<?php print_r($row1["avatar"]) ?>" class="round"><p></p><h2 >
<?php
print_r($row1["first_name"]);?>
<?php
print_r($row1["last_name"]);?>
<?php
print_r($row1["middle_name"]);?></h2> </div><?php } ?>
<div class="clear"></div>
</div>

```

```
</header>
</body></html>
```

### account.php

```
<?php session_start();require_once("dbconnect.php");?>
<!DOCTYPE html>
<html>
<head>
  <title>Виконання дипломної роботи</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="css/acc.css">
  <link rel="icon" href="Media/graduated.png" sizes="16x16" type="image/png">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head><body>
<header> <div class="container">
  <?php
  if(!isset($_SESSION['email']) && !isset($_SESSION['password'])){  ?>
<div class="reg">
  <a href="form_register.php" class="button9" >РЕЄСТРАЦІЯ</a><div><div >
  <a href="form_auth.php" class="button10" >АВТОРИЗАЦІЯ</a>
  </div> </div></div>

  <?php
  }else{
  ?> <?php
  $email = $_SESSION['email'];
  $password = $_SESSION['password'];
  $users = $mysqli->query("SELECT * FROM `users` WHERE email = '". $email.'" AND password =
  '". $password."'");
  $row = $users->fetch_assoc();?>
  <?php if($row["id_role"]==1){ ?>
  <a href="index.php"></a><?php }
  elseif ($row["id_role"]==3 || $row["id_role"]==5){ ?>
  <a href="teacher.php"></a><?php }
  else if ($row["id_role"]==4){ ?>
  <a href="clerk.php"></a><?php }?>
  <div >
  <a href="account.php"></a>
  <p class="Name"> Інформація про Вас
  </p>
  <a href="logout.php"></a> </div>
  <div class="main">
  <div class="row">
  <div class="col s12 m4 l3 center-align" style="padding: 2rem">
  " />
  <!--<p class="center-align"><a href="#change">Изменить</a></p-->
  </div>
  <div class="col s12 m8 l9">
  <h1 style="float: left; margin-top:-15%; margin-left: 20%; "><?php print_r($row["first_name"]);?> <?php
  print_r($row["last_name"]);?> <?php print_r($row["middle_name"]);?></h1>
  <h2 style="float: left; margin-top:-7%; margin-left: 20%; "><?php print_r($row["email"]);?></h2>
  </div>
  </div>
  <table>
  <!-- generic account info --><?php if($row["id_role"] == 1){ ?><tr><td class="shadow-td"></td>
  <td class="shadow-td" colspan="2"><b>Моя група</b></td></tr><tr>
  <td width="20%"></td><td><ul>
  <?php
  $group_info = $mysqli->query("SELECT * FROM `groups` WHERE id_group IN (select id_group from `users` where
  email = '". $email.'" )");
  $row1 = $group_info->fetch_assoc();?>
  <li><?php print_r($row1["name_group"]);?></li> </ul> </td> </tr> <tr>
  <td class="shadow-td"></td>
```

```

<td class="shadow-td" colspan="2" ><b>Мій дипломний керівник</b></td></tr> <tr><td
width="20%"></td><td><ul><?php
$teacher_info = $mysqli->query("SELECT * FROM `users` WHERE id IN (select id_teacher from `users` where
id_teacher is NOT NULL and email = ".$email."");
$row2 = $teacher_info->fetch_assoc();?>
<li><?php
print_r($row2["first_name"]);?><?php
print_r($row2["last_name"]);?>
<?php
print_r($row2["middle_name"]);?></li></ul></td></tr><tr>
<td class="shadow-td"></td>
<td class="shadow-td" colspan="2" ><b>Моя тема диплому</b></td></tr><tr>
<td width="20%"></td>
<td>
<ul><li><?php
print_r($row["theme"]);?></li></ul> </td> </tr><tr>
<td class="shadow-td"></td>
<td class="shadow-td" colspan="2" ><b>Моя дата реєстрації</b></td></tr>
<tr><td width="20%"></td><td><ul><li><?php
print_r($row["date_registration"]);?> </li></ul> </td>
</tr> <?php }
else if ($row["id_role"] == 3 || $row["id_role"] == 5) { ?> <tr>
<td class="shadow-td"></td>
<td class="shadow-td" colspan="2" ><b>Моя дата реєстрації</b></td></tr>
<tr><td width="20%"></td><td><ul>
<li><?php
print_r($row["date_registration"]);?></li></ul></td></tr><tr>
<td class="shadow-td"></td>
<td class="shadow-td" colspan="2" ><b>Мої студенти</b></td></tr><tr>
<td width="20%"></td><td><ul>
<?php
$student_info = $mysqli->query("SELECT * FROM `users` WHERE id_teacher IN (select id from `users` where
email = ".$email."");?>
<?php
while($row1 = $student_info->fetch_assoc()): ?>
<li><?php
print_r($row1["first_name"]);?>
<?php print_r($row1["last_name"]);?>
<?php
print_r($row1["middle_name"]);?></li>
<?php endwhile; ?></ul> </td>
</tr> <?php
} else if ($row["id_role"] == 4) { ?> <tr>
<td class="shadow-td"></td>
<td class="shadow-td" colspan="2" ><b>Моя дата реєстрації</b></td></tr><tr>
<td width="20%"></td><td><ul> <li><?php
print_r($row["date_registration"]);?> </li></ul></td></tr><?php } ?></table></div>
<?php } ?>
<div class="clear"></div>
</div>
</header>
</body></html>

```

## admin.php

```

<link rel="stylesheet" type="text/css" href="css/admin1.css">
<link rel="icon" href="Media/graduated.png" sizes="16x16" type="image/png">
<script src="css/admin.js"></script>
<svg style="display:none;">
<symbol id="down" viewBox="0 0 16 16">
<polygon points="3.81 4.38 8 8.57 12.19 4.38 13.71 5.91 8 11.62 2.29 5.91 3.81 4.38" /></symbol>
<symbol id="users" viewBox="0 0 16 16">
<path d="M8,0a8,8,0,1,0,8,8A8,8,0,0,0,8,0ZM8,15a7,7,0,0,1-5.19-2.32,2.71,2.71,0,0,1,1.7-1,13.11,13.11,0,0,0,1.29-
.28,2.32,2.32,0,0,0,.94-.34,1.17,1.17,0,0,0-.27-

```

```

.7h0A3.61,3.61,0,0,1.5,15,7.49,3.18,3.18,0,0,1,8,4.07a3.18,3.18,0,0,1,2.86,3.42,3.6,3.6,0,0,1-
1.32,2.88h0a1.13,1.13,0,0,0-
.27.69,2.68,2.68,0,0,0,.93,31,10.81,10.81,0,0,0,1.28,23,2.63,2.63,0,0,1,1.78,1A7,7,0,0,1,8,15Z" /></symbol>
<symbol id="collection" viewBox="0 0 16 16">
  <rect width="7" height="7" />
  <rect y="9" width="7" height="7" />
  <rect x="9" width="7" height="7" />
  <rect x="9" y="9" width="7" height="7" /></symbol>
<symbol id="charts" viewBox="0 0 16 16">
  <polygon points="0.64 7.38 -0.02 6.63 2.55 4.38 4.57 5.93 9.25 0.78 12.97 4.37 15.37 2.31 16.02 3.07 12.94 5.72 9.29
2.21 4.69 7.29 2.59 5.67 0.64 7.38" />
  <rect y="9" width="2" height="7" />
  <rect x="12" y="8" width="2" height="8" />
  <rect x="8" y="6" width="2" height="10" />
  <rect x="4" y="11" width="2" height="5" /></symbol>
<symbol id="comments" viewBox="0 0 16 16">
  <path d="M0,16.13V2H15V13H5.24ZM1,3V14.37L5,12h9V3Z"/><rect x="3" y="5" width="9" height="1"/><rect
x="3" y="7" width="7" height="1"/><rect x="3" y="9" width="5" height="1"/></symbol>
<symbol id="pages" viewBox="0 0 16 16">
  <rect x="4" width="12" height="12" transform="translate(20 12) rotate(-180)"/><polygon points="2 14 2 2 0 2 0 14 0
16 2 16 14 16 14 2 14"/> </symbol>
<symbol id="appearance" viewBox="0 0 16 16">
  <path d="M3,0V7A2,2,0,0,0,5,9H6v5a2,2,0,0,0,4,0V9h1a2,2,0,0,0,2-2V0Zm9,7a1,1,0,0,1-1,1H9v6a1,1,0,0,1-
2,0V8H5A1,1,0,0,1,4,7V6h8ZM4,5V1H6V4H7V1H9V4h1V1h2V5Z"/></symbol>
<symbol id="trends" viewBox="0 0 16 16">
  <polygon points="0.64 11.85 -0.02 11.1 2.55 8.85 4.57 10.4 9.25 5.25 12.97 8.84 15.37 6.79 16.02 7.54 12.94 10.2
9.29 6.68 4.69 11.76 2.59 10.14 0.64 11.85"/></symbol>
<symbol id="settings" viewBox="0 0 16 16">
  <rect x="9.78" y="5.34" width="1" height="7.97"/><polygon points="7.79 6.07 10.28 1.75 12.77 6.07 7.79
6.07"/><rect x="4.16" y="1.75" width="1" height="7.97"/><polygon points="7.15 8.99 4.66 13.31 2.16 8.99 7.15
8.99"/><rect x="1.28" width="1" height="4.97"/><polygon points="3.28 4.53 1.78 7.13 0.28 4.53 3.28 4.53"/><rect
x="12.84" y="11.03" width="1" height="4.97"/><polygon points="11.85 11.47 13.34 8.88 14.84 11.47 11.85
11.47"/></symbol>
<symbol id="options" viewBox="0 0 16 16">
  <path d="M8,11a3,3,0,1,1,3-3A3,3,0,0,1,8,11ZM8,6a2,2,0,1,0,2,2A2,2,0,0,0,8,6Z"/><path d="M8.5,16h-
1A1.5,1.5,0,0,1,6,14.5v-.85a5.91,5.91,0,0,1,-.58-.24l-.66A1.54,1.54,0,0,1,2.7,14L2,13.3a1.5,1.5,0,0,1,0-2.12l.6-
.6A5.91,5.91,0,0,1,2.35,10H1.5A1.5,1.5,0,0,1,0,8.5v-1A1.5,1.5,0,0,1,1.5,6h.85a5.91,5.91,0,0,1,.24-
.58L2,4.82A1.5,1.5,0,0,1,2.2,7.2A1.54,1.54,0,0,1,4.82,2l.66A5.91,5.91,0,0,1,6,2.35V1.5A1.5,1.5,0,0,1,7.5,0h1A1.5,
1.5,0,0,1,10,1.5v.85a5.91,5.91,0,0,1,-.58-.24l-.66A1.54,1.54,0,0,1,13.3,2L14,2.7a1.5,1.5,0,0,1,0-2.12l-
.66A5.91,5.91,0,0,1,.24.58h.85A1.5,1.5,0,0,1,16,7.5v1A1.5,1.5,0,0,1,14.5,10h-.85a5.91,5.91,0,0,1-
.24.58l.66a1.5,1.5,0,0,1,0-2.12L13.3,14a1.54,1.54,0,0,1-2.12,0l-.66A5.91,5.91,0,0,1-
.58.24v.85A1.5,1.5,0,0,1,8.5,16ZM5.23,12.18l.33.18a4.94,4.94,0,0,0,1.07.44l.36.1V14.5a.5,5,0,0,0,.5.5h1a.5,5,0,0,0,.5-
.5V12.91l.36-.1a4.94,4.94,0,0,0,1.07-.44l.33-.18,1.12,1.12a.51,5.1,0,0,0,.71,0l.71-.71a.5,5,0,0,0,-.71l-1.12-1.12.18-
.33a4.94,4.94,0,0,0,.44-1.07l.1-.36H14.5a.5,5,0,0,0,.5-.5v-1a.5,5,0,0,0,-.5H12.91l-.1-.36a4.94,4.94,0,0,0,-.44-1.07l-.18-
.33L13.3,4.11a.5,5,0,0,0,-.71L12.6,2.7a.51,5.1,0,0,0,-.71,0L10.77,3.82l-.33-.18a4.94,4.94,0,0,0-1.07-
.44L9,3.09V1.5A.5,5,0,0,0,8.5,1h-1a.5,5,0,0,0,-.5V3.09l-.36.1a4.94,4.94,0,0,0-1.07.44l-.33.18L4.11,2.7a.51,5.1,0,0,0-
.71,0L2.7,3.4a.5,5,0,0,0,.71L3.82,5.23l-.18.33a4.94,4.94,0,0,0-.44,1.07L3.09,7H1.5a.5,5,0,0,0-
.5.5v1a.5,5,0,0,0,.5H3.09l.1.36a4.94,4.94,0,0,0,.44,1.07l.18.33L2.7,11.89a.5,5,0,0,0,7.11.71a.51,5.1,0,0,0,.71,0Z"/>
</symbol>
<symbol id="collapse" viewBox="0 0 16 16">
  <polygon points="11.62 3.81 7.43 8 11.62 12.19 10.09 13.71 4.38 8 10.09 2.29 11.62 3.81"/></symbol>
<symbol id="search" viewBox="0 0 16 16">
  <path d="M6.57,1A5.57,5.57,0,1,1,1,6.57,5.57,5.57,0,0,1,6.57,1m0-
1a6.57,6.57,0,1,0,6.57,6.57A6.57,6.57,0,0,0,6.57,0Z"/><rect x="11.84" y="9.87" width="2" height="5.93"
transform="translate(-5.32 12.84) rotate(-45)"/>
</symbol></svg><header class="page-header"><nav>
  <a href="admin.php"></a>
  <button class="toggle-mob-menu" aria-expanded="false" aria-label="open menu">
    <svg width="20" height="20" aria-hidden="true">
      <use xlink:href="#down"/></use></svg></button><ul class="admin-menu">
        <li class="menu-heading"></li><li>
          <a href="new_section.php"><svg>
            <use xlink:href="#pages"/></use></svg><span>Розділи</span></a></li><li>
          <a href="reg_role.php"><svg>
            <use xlink:href="#users"/></use></svg>

```

```

    <span>Додавання коритувачів</span></a> </li> </li>
<a href="#0"> <svg>
    <use xlink:href="#trends"></use> </svg>
    <span>Додавання груп</span> </a> </li> </li>
<a href="#0"> <svg>
    <use xlink:href="#users"></use> </svg>
    <span>Видалення користувачів</span> </a> </li> </li>
<a href="#0"> <svg>
    <use xlink:href="#users"></use> </svg>
    <span>Редагування користувачів</span> </a> </li> </ul> </nav></header>
<section class="page-content">
<section class="search-and-user">
<div class="admin-profile">
    <span class="greeting">Панель адміністратора</span>
    <div class="notifications"> <svg>
        <use xlink:href="#users"></use> </svg> </div> </div> </section>
    <a href="logout.php"></a>
</section>

```

## func.php

```

<?php
function LoadDataUsers(){
    include('dbconnect.php');
    $email = $_SESSION['email'];
    $password = $_SESSION['password'];
    $query = $mysqli->query("SELECT * FROM `users` WHERE id_teacher IN (select id from `users` where email =
    '". $email. "')"); $array = $query->fetch_assoc(); do { ?> <p></p>
        <img src = "<?php print_r($array["avatar"]) ?>" class="round"> <p></p>
        <h2 ><?php echo "<a href='\"user.php?id=\".$array['id'].\"'>\".$array["first_name"]." \".$array["last_name"]."
    ".$array["middle_name"]." </a>";?></h2><?php }
    while( $array = $query->fetch_assoc());?>

```

## document.php

```

<?php include("index.php");?>
<link rel="stylesheet" type="text/css" href="css/tables.css">
<div class="main"> <?php
    $section = $mysqli->query("SELECT * FROM `document` where id = 1");
    $sec = $section->fetch_assoc();$stud_id = $row["id"];$sec_id = $sec["id"];?>
<div class="section">
    <p class = "title"><?php echo $sec["name_doc"]; ?></p>
    <form class = "form" action="pz.php" method="post" enctype="multipart/form-data"> <p></p>
        <table class = "upload"><tr> <td>
            <div class="fileform">
                <div id="fileformlabel"></div>
                <div class="selectbutton">Обрати</div>
                <input id="upload" type="file" name="studentfile"></div></td>
            <td><button class="button" style="vertical-align:middle" type="submit" name="save"><span>Завантажити
        </span></button></td></tr></table> <?php
        if (isset($_POST["save"])){
            $filename = $_FILES["studentfile"]["name"];
            $destination = 'Files/' . $filename;
            $extension = pathinfo($filename,PATHINFO_EXTENSION);
            $file = $_FILES["studentfile"]["tmp_name"];
            $size = $_FILES["studentfile"]["size"];
            if (!in_array($extension,['doc','docx','docm','dot','dotm','dotx'])){
                echo "Your file must be .doc, .docx, .docm, .dot, .dotm, .dotx!"; } else {
                if(move_uploaded_file($file,$destination)){
                    $sql = $mysqli->query("INSERT INTO `files` (name,size,date,sec_id,stud_id)
VALUES('".$filename."','".$size."',NOW(),".$sec_id."','".$stud_id.")"); if ($sql){
                        echo "file uploaded succesfully"; } else{
                            echo "failed"; } } } ?> </form> </div> <?php

```

```

$file0 = $mysqli->query("SELECT * FROM `files` where stud_id = ".$stud_id." and sec_id = ".$sec_id."");?>
<table class="table_price"> <tr>
  <th>Назва документу</th>
  <th>Розмір документу</th>
  <th>Дата завантаження</th>
  <th>Версія документу</th>
  <th>Статус</th>
  <th>Короткий коментар</th>
</tr></table> <?php
if (isset($_GET["file_id"])){
  $id = $_GET["file_id"];
  $file1 = $mysqli->query("SELECT * FROM `files` where id_file = ".$id." ");
  $rowf = $file1->fetch_assoc();
  $filepath = 'Files/' . $rowf["name"];
  if(file_exists($filepath)){
    header("Content-Type: application/octet-stream");
    header("Content-Description: File Transfer");
    header("Content-Disposition: attachment; filename=" . basename($filepath));
    header("Expires: 0");
    header("Cache-Control: must-revalidate");
    header("Pragma:public");
    header("Content-Length:" . filesize("Files/" . $rowf["name"]));
    readfile("Files/" . $rowf["name"]); exit; } } ?>
<?php while($row5 = $file0->fetch_assoc()): ?> <tr>
  <td><?php echo $row5["name"];?></td>
  <td><?php echo $row5["size"] / 1000 . "KB";?></td>
  <td><?php echo $row5["date"];?></td>
  <td><?php echo $row5["version"];?></td> <td> <?php
  $status = $mysqli->query("SELECT * FROM `status` where id IN (select status_id from `files` where
id_file = ".$row5["id_file"].") ");
  if (!$st= mysqli_fetch_array($status)){ echo "NO status";}else{
  echo $st["name_status"];} ?> </td>
  <td><?php echo $row5["comment"];?></td> <td>
  <a href = "pz.php?file_id=<?php echo $row5["id_file"]?>">Завантажити</a>
  </td> </tr> <?php endwhile; ?> </table> </div>

```

### statistic.php

```

<?php include("clerk.php");?>
<link rel="stylesheet" type="text/css" href="css/stat.css">
<div class="main"> <?php $y = 0; $n = 0;
  $users = $mysqli->query("SELECT * FROM `groups`");
  while ($row = $users->fetch_assoc()): ?>
  <h2><?php echo $row["name_group"];?></h2> <table > <tr>
  <th>ФІО</th>
  <th>Статус пояснювальної записки</th> </tr> <?php
  $query = $mysqli->query("SELECT * FROM `users` where id_role = 1 and id_group = ".$row["id_group"]."");?>
  <tbody><?php while( $array = $query->fetch_assoc()):?> <tr> <td>
  <?php echo $array["first_name"]." ".$array["last_name"]." ".$array["middle_name"];?> </td> <td>
  <?php
  $pz = $mysqli->query("SELECT `status`.name_status from `status` JOIN `files` ON `status`.id = `files`.status_id
WHERE `files`.stud_id = ".$array["id"]." and `files`.sec_id = 1 order by `files`.version DESC LIMIT 1");
  while ($pz = $pz->fetch_assoc()): ?>
  <?php if ($pz["name_status"] == "Робота завершена"){
    $y++; echo $pz["name_status"]; }
  else { $n++; echo "Робота не завершена"; } ?> <?php endwhile;?> </td></tr>
  <?php endwhile;?> </tbody> </table><p></p>
  <?php echo "Кількість студентів, що завершили роботу: " . $y . ".";?><p></p>
  <?php echo "Кількість студентів, що не завершили роботу: " . $n . ".";?> <?php endwhile;?>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);

```



```
function drawChart() {
    var data = google.visualization.arrayToDataTable([
        ['Завершена робота', 'Кількість'],      ['Так', <?php echo $y ;?> ],
        ['Ні', <?php echo $n ;?>]];      var options = {
        title: 'Статистика студентів'      };
    var chart = new google.visualization.PieChart(document.getElementById('piechart'));
    chart.draw(data, options);      } </script>
<div id="piechart" style="width: 900px; height: 500px;"></div>
<p>Всього студентів: <?php echo $y+$n;?>.</p> </div> <!-->
```

### *status\_table.php*

```
<?php include("clerk.php");      require_once("funcgroups.php");?>
<link rel="stylesheet" type="text/css" href="css/stat.css">
<div class="main">      <?php
    $users = $mysqli->query("SELECT * FROM `groups`");
    while ($row = $users->fetch_assoc()):      ?>
<h2><?php echo $row["name_group"]?></h2>      <table >      <tr>
    <th>ФІО</th>
    <th>Пояснювальна записка</th>
    <th>Протокол перевірки на плагіат</th>
    <th>Файли презентації</th>
    <th>Архів з програмним додатком</th>      </tr>      <?php
        $query = $mysqli->query("SELECT * FROM `users` where id_role = 1 and id_group =
        ".$row["id_group"]."");?>
    <tbody><?php while( $array = $query->fetch_assoc()):?>      <tr>      <td>
        <?php echo $array["first_name"]." ".$array["last_name"]." ".$array["middle_name"];?>
    </td>      <td>      <?php
        $pz = $mysqli->query("SELECT `status`.name_status from `status` JOIN `files` ON `status`.id = `files`.status_id
        WHERE `files`.stud_id = ".$array["id"]." and `files`.sec_id = 1 order by `files`.version DESC LIMIT 1");
        while ($pz = $pz->fetch_assoc()):      ?>      <?php echo $pz["name_status"];?><?php
        endwhile;?>      </td>      <td><?php
        $protocol = $mysqli->query("SELECT `status`.name_status from `status` JOIN `files` ON `status`.id =
        `files`.status_id WHERE `files`.stud_id = ".$array["id"]." and `files`.sec_id = 2 order by `files`.version DESC LIMIT
        1");
        while ($pr = $protocol->fetch_assoc()):      ?>
            <?php echo $pr["name_status"];?><?php
        endwhile;?>      </td>      <td>      <?php
        $present = $mysqli->query("SELECT `status`.name_status from `status` JOIN `files` ON `status`.id =
        `files`.status_id WHERE `files`.stud_id = ".$array["id"]." and `files`.sec_id = 3 order by `files`.version DESC LIMIT
        1");
        while ($pres = $present->fetch_assoc()):      ?>
            <?php echo $pres["name_status"];?><?php
        endwhile;?>      </td>      <td>      <?php
        $archieive = $mysqli->query("SELECT `status`.name_status from `status` JOIN `files` ON `status`.id =
        `files`.status_id WHERE `files`.stud_id = ".$array["id"]." and `files`.sec_id = 4 order by `files`.version DESC LIMIT
        1");
        while ($ar = $archieive ->fetch_assoc()):      ?>
            <?php echo $ar["name_status"];?><?php
        endwhile;?>      </td>      </tr>
        <?php endwhile;?>      </tbody>      </table>      <?php endwhile;      ?></div>
```