

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: дистанційна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**ПРОЄКТУВАННЯ СИСТЕМИ ПОШУКУ ІНФОРМАЦІЇ НА
КОРПОРАТИВНОМУ САЙТІ**

здобувача Сергєєвої Ольги Володимирівни_____

Науковий керівник:

д.е.н., професор

_____Мозгаллі О.П.

Проект допущений до захисту

перед екзаменаційною комісією з

атестації здобувачів вищої освіти

завідувач кафедри:

к.е.н., доцент.

_____Тішков Б.О.

Київ 2024

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО_ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)

освітньо-професійної програми

Іванченко Г.Ф.

“ ” 2024 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

Тішков Б.О.

“ ” 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Сергєєвої Ольги Володимирівни

дистанційної форми навчання

на підготовку кваліфікаційного бакалаврського проекту
на тему: «Проектування системи пошуку інформації на корпоративному сайті»

Тему затверджено наказом ректора Університету від «11» 03 2024 р. № 531-ст.

Кваліфікаційний бакалаврський проект виконується на матеріалах наукових публікацій, інформації з мережі інтернет, внутрішньої інформації компанії, для якої розробляється система

План кваліфікаційного бакалаврського проекту

Розділ I Характеристика та аналіз предметної галузі

Розділ II Розробка вимог і моделювання системи

Розділ III Проектування та реалізація компонентів системи

Об'єкт дослідження: проектування системи корпоративного пошуку

Предмет дослідження: сукупність теоретичних, методичних і практичних підходів та технологій проектування системи корпоративного пошуку

Мета кваліфікаційного бакалаврського проекту: вдосконалення пошуку інформації на корпоративному сайті шляхом розроблення нової системи корпоративного пошуку

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі I дослідити теоретичні відомості щодо процесів пошуку внутрішньокорпоративної інформації, проаналізувати існуючі системи і технології пошуку інформації

У розділі II визначити вимоги (бізнес-вимоги, функціональні, нефункціональні) до системи. Розробити постановку задачі, описати її характеристики, вхідну, вихідну інформацію, спроектувати інформаційну модель задачі, розробити алгоритм функціонування системи, здійснити моделювання поведінки та структури системи.

У розділі III запроєктувати інформаційне, технічне та програмне забезпечення. Реалізувати запропоновані рішення у вигляді системи корпоративного пошуку.

Завдання підготував
науковий керівник _____

Мозгаллі Ольга Петрівна

“ 12 ” березня 2024 р.

Завдання одержав
студент _____

Сергєєва Ольга Володимирівна

“ 12 ” березня 2024 р.

Відгук

про кваліфікаційний бакалаврський проект
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Сергєєвої Ольги Володимирівни

на тему

«Проектування системи пошуку інформації на корпоративному сайті»

1. Актуальність теми: Тема кваліфікаційного бакалаврського проекту є актуальною, адже в сучасному світі інформація стає все більш цінним ресурсом, а корпоративні сайти – одним з основних каналів її доступу. Ефективна система пошуку інформації на сайті може значно покращити його зручність використання, економити час користувачів та підвищити загальну продуктивність роботи.

2. Позитивні риси кваліфікаційного бакалаврського проекту: В роботі було системно досліджено предметну область, зокрема було проведено ґрунтовний аналіз існуючих систем пошуку інформації на корпоративних сайтах, що дозволило йому виявити їхні переваги та недоліки. Авторка чітко окреслила мету проекту – розробити систему пошуку інформації на корпоративному сайті, а також визначила чіткі завдання, які необхідно виконати для її досягнення. В проекті було не лише розроблено теоретичну модель системи пошуку, але й здійснена її практична реалізація, що дозволило оцінити її працездатність та ефективність. Кваліфікаційний бакалаврський проект оформлений відповідно до вимог, текст чітко написаний, зрозумілий та логічно організований. Розроблена інформаційна система пошуку інформації на корпоративному сайті може мати практичне застосування у різних сферах діяльності.

3. Наявність самостійних розробок автора: Кваліфікаційний бакалаврський проект виконаний авторкою самостійно із застосуванням сучасного інструментарію. Авторка самостійно розробила алгоритм пошуку інформації, який враховує специфіку корпоративного сайту та його потреб. Також вона розробила інтерфейс користувача системи пошуку, який є зручним та інтуїтивно зрозумілим. Крім того, авторкою було проведено тестування розробленої системи пошуку та усунено виявлені помилки.

4. Цінність теоретичних висновків та практичних рекомендацій: Теоретичні висновки та практичні рекомендації, які представлені в кваліфікаційному бакалаврському проекті, можуть бути цінними для розробників корпоративних сайтів, які прагнуть покращити систему пошуку інформації на своїх ресурсах.

5. Наявність недоліків: Представлене проєктне рішення потребує подальшого вдосконалення й розвитку для використання у промисловій експлуатації. Також перед впровадженням в промислову експлуатацію необхідно буде провести тестування розробленої системи пошуку на проникнення, щоб оцінити її стійкість до можливих кібератак.

6. Загальна оцінка кваліфікаційного бакалаврського проєкту та його допущення до захисту перед ЕК: Загалом, представлений кваліфікаційний бакалаврський проєкт відповідає встановленим вимогам методичних вказівок щодо структури, обсягу та змісту, та рекомендується до захисту з **високою оцінкою**, а Сергєєвій Ользі Володимирівні може бути присвоєний освітньо-кваліфікаційний рівень «бакалавр» галузі знань 12 «Інформаційні технології» за спеціальністю 122 «Комп'ютерні науки».

Науковий керівник

(підпис)

д. е. н., Мозгаллі О.П.

“10” червня 2024 р.

Рецензія
на кваліфікаційний бакалаврський проект
Сергєєвої Ольги Володимирівни

тема

«Проектування системи пошуку інформації на корпоративному сайті»

Актуальність теми кваліфікаційного бакалаврського проекту і доцільність його розроблення. *Актуальність розробки інформаційної системи пошуку на корпоративному сайті обумовлена зростаючими обсягами даних, необхідністю оперативного доступу до релевантної інформації, підвищенням ефективності роботи персоналу та покращенням взаємодії з клієнтами. В умовах жорсткої конкуренції компанії потребують інструментів для швидкого знаходження потрібних даних, що сприяє прийняттю стратегічних рішень та зміцненню ринкових позицій. Корпоративні сайти стають основним каналом комунікації, тому інтеграція ефективної системи пошуку підвищує задоволеність користувачів, залучає нових клієнтів і автоматизує процеси збору та обробки інформації, знижуючи навантаження на персонал і мінімізуючи ризик помилок. Це робить розробку таких систем незамінною для сучасних організацій, які прагнуть оптимізувати свої інформаційні ресурси та підвищити конкурентоспроможність.*

Якість проведеного дослідження.

Кваліфікаційний бакалаврський проект виконаний авторкою самостійно із застосуванням сучасного інструментарію. Авторка самостійно розробила алгоритм пошуку інформації, який враховує специфіку корпоративного сайту та його потреб. Авторка розробила власну актуальну систему пошуку інформації на корпоративному сайті.

.

Позитивні риси кваліфікаційного бакалаврського проекту.

Авторка провела комплексний аналіз існуючих систем пошуку на корпоративному сайті. На основі здійсненого аналізу Сергєєвою О.В. було розроблено вимоги до системи, постановка задачі та здійснено моделювання системи. В проєкті було не лише розроблено теоретичну модель системи пошуку, але й здійснена її практична реалізація. Кваліфікаційний бакалаврський проєкт оформлений відповідно до вимог, текст чітко написаний, зрозумілий та логічно організований. Розроблена інформаційна система пошуку інформації на корпоративному сайті може мати практичне застосування у різних сферах діяльності.

Зауваження. Варто було більше уваги приділити обґрунтуванню вибору мови програмування РНР, оскільки на даний момент існують більш сучасні і прогресивні мови програмування та фреймворки.

Практична значимість висновків і рекомендацій.

Практична частина проекту містить рекомендації щодо вибору технологій, проектування системи та розробки інтерфейсу користувача. Результатом роботи є робочий програмний продукт, який може бути вдосконалений і розширений у майбутньому, а також адаптований для будь-якої компанії, яка має корпоративний сайт.

Місце роботи та посада рецензента:

ВСП «Фаховий коледж інформаційних систем і технологій КНЕУ імені
Вадима Гетьмана», завідувач відділенням техніко-інформаційних
спеціальностей

Науковий ступінь, учене звання
(за наявності)

_____ Андрій ПРАВДИВИЙ

Місце печатки організації, де працює рецензент

АНОТАЦІЯ

кваліфікаційного бакалаврського проекту
здобувача навчально-наукового інституту «Інститут інформаційних
технологій в економіці»
освітньо-професійної програми «Комп'ютерні науки»

Сергєєвої Ольги Володимирівни

на тему

«Проектування системи пошуку інформації на корпоративному сайті»

Київ: кафедра інформаційних систем в економіці, 2024

Бакалаврський дипломний проект присвячений актуальній проблемі автоматизації пошуку інформації на корпоративному сайті, яку пропонується розв'язувати та удосконалювати з використанням сучасних інформаційних технологій.

Кваліфікаційний бакалаврський проект складається з трьох розділів, логічно пов'язаних між собою. В першому розділі дана характеристика предметної галузі й об'єкта дослідження, наведено аналіз задач, що розв'язуються, описані основні процеси корпоративного пошуку, а також наводиться перелік існуючих програмних засобів даної предметної галузі. Другий розділ є проектним і присвячений обґрунтуванню методу проектування системи, розробленню її архітектури, виконанню постановки та розробленню алгоритму розв'язання задач. Визначені вимоги до системи, змодельована її поведінка та структура. Третій розділ – конструктивний. Тут наведено інформаційне, технічне та програмне забезпечення системи корпоративного пошуку. Висвітлені питання організації інформаційного забезпечення: розроблена структура інформаційного забезпечення. Обґрунтований комплекс технічних засобів, а також програмне забезпечення, що використовується для створення системи корпоративного пошуку. Наведено опис реалізованої системи. Висновки містять рекомендації щодо доцільності розроблення та впровадження системи корпоративного пошуку в компаніях.

РЕФЕРАТ

Кваліфікаційний бакалаврський проєкт містить 79 сторінок, 2 таблиці, 59 рисунків, список використаних джерел з 52 найменувань, 1 додатку.

«Проектування системи пошуку інформації на корпоративному сайті»

Об'єктом дослідження кваліфікаційного бакалаврського проєкту є проектування системи корпоративного пошуку

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів та технології проектування системи корпоративного пошуку.

Мета і завдання дослідження. Основною метою кваліфікаційного бакалаврського проєкту є вдосконалення пошуку інформації на корпоративному сайті шляхом розроблення нової системи корпоративного пошуку.

Відповідно до поставленої мети визначені такі *завдання*:

- дослідити предметну область,
- виконати аналіз існуючих систем пошуку інформації;
- визначити вимоги до системи: бізнес-вимоги, функціональні, нефункціональні
- розробити постановку задачі, описати її характеристику, вхідну, вихідну інформацію, спроектувати інформаційну модель задачі;
- розробити алгоритм функціонування системи,
- здійснити моделювання поведінки та структури системи.
- запроєктувати інформаційне, технічне та програмне забезпечення.
- реалізувати запропоновані рішення у вигляді системи пошуку на корпоративному сайті.

Теоретична, методична та практична значущість отриманих результатів. Під час дослідження було досліджено предметну область, проаналізовано існуючі системи пошуку. Розроблені проєктні рішення щодо проектування системи пошуку інформації на корпоративному сайті.

Практичні результати дослідження полягають у розробленні системи, яка здійснює пошук файлів і веб-сторінок за пошуковим запитом та зберігає статистику про найбільш поширені запити. Запропонована система може бути використана для будь-якого підприємства чи організації, яка має корпоративний веб-портал.

Рік виконання кваліфікаційного бакалаврського проєкту – 2024.

Рік захисту роботи – 2024.

Ключові слова: пошук, корпоративний сайт, інформаційна система, інформаційне забезпечення, програмне забезпечення, моделювання системи.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	5
1.1 Характеристика предметної галузі та об'єкта дослідження	5
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі	13
РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ СИСТЕМИ	21
2.1 Аналіз і специфікація вимог до інформаційної підсистеми	21
2.1 Бізнес-вимоги до системи	21
2.2 Функціональні вимоги до системи	23
2.3 Нефункціональні вимоги до системи	25
2.2 Постановка задачі та алгоритм розв'язання задачі	26
2.2.1 Постановка задачі	26
2.2.2 Математичне забезпечення та алгоритм функціонування системи	32
2.3 Моделювання інформаційної підсистеми	34
2.3.1 Моделювання поведінки системи	34
2.3.2 Моделювання структури системи	42
2.3.3 Розподіл вимог за компонентами системи	47
РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	54
3.1 Інформаційне забезпечення	54
3.2 Технічне забезпечення	59
3.3 Програмне забезпечення	61
3.4 Результати реалізації системи пошуку на корпоративному сайті	65
ВИСНОВКИ	73
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	75
ДОДАТКИ	80

ВСТУП

Актуальність теми. З поширенням корпоративних веб-сайтів як основного інструменту комунікації між компанією та її співробітниками, клієнтами, стейкхолдерами або партнерами, виникає проблема ефективного доступу до різноманітної інформації, яка зберігається на сайті. Недоцільно ручний пошук великої кількості інформації через структуровані меню або довгий перелік посилань.

Крім того, вимоги користувачів до швидкості, точності та релевантності результатів пошуку постійно зростають. Це зумовлено зростанням конкуренції на ринку, де користувачі очікують найкращого досвіду використання веб-ресурсів. Таким чином, розробка ефективної інформаційної системи пошуку стає необхідністю для підтримки високого рівня задоволеності користувачів та покращення їхньої взаємодії з корпоративним сайтом.

Об'єктом дослідження кваліфікаційного бакалаврського проєкту є проєктування системи корпоративного пошуку

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів та технології проєктування системи корпоративного пошуку.

Мета і завдання дослідження. Основною метою кваліфікаційного бакалаврського проєкту є вдосконалення пошуку інформації на корпоративному сайті шляхом розроблення нової системи корпоративного пошуку.

Відповідно до поставленої мети визначені такі *завдання*:

- дослідити предметну область,
- виконати аналіз існуючих систем пошуку інформації;
- визначити вимоги до системи: бізнес-вимоги, функціональні, нефункціональні
- розробити постановку задачі, описати її характеристики,

вхідну, вихідну інформацію, спроектувати інформаційну модель задачі;

- розробити алгоритм функціонування системи,
- здійснити моделювання поведінки та структури системи.
- запроєктувати інформаційне, технічне та програмне забезпечення.
- реалізувати запропоновані рішення у вигляді системи пошуку на корпоративному сайті.

Теоретична, методична та практична значущість отриманих результатів. Під час дослідження було досліджено предметну область, проаналізовано існуючі системи пошуку. Розроблені проєктні рішення щодо проєктування системи пошуку інформації на корпоративному світі.

Практичні результати дослідження полягають у розробленні системи, яка здійснює пошук файлів і веб-сторінок за пошуковим запитом та зберігає статистику про найбільш поширені запити. Зпропонована система може бути використана для будь-якого підприємства чи організації, яка має корпоративний веб-портал.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Корпоративний сайт – це веб-ресурс, що створюється для представлення бізнесу в Інтернеті та забезпечення комунікації з цільовою аудиторією.

Цей сайт може включати інформацію про компанію, її продукти та послуги, новини, блоги, контактні дані та інші ключові елементи.

Головною метою корпоративного сайту є підвищення впізнаваності бренду компанії, збільшення обсягу продажів та підтримка взаємодії з клієнтами. Він може допомогти компанії розширити свою діяльність на нові ринки та залучити нових клієнтів.

Однією з основних переваг корпоративного веб-сайту є його доступність. Він дає компанії можливість бути присутньою в Інтернеті цілодобово і забезпечує доступ до інформації про компанію, її продукти та послуги, незалежно від місця перебування користувача. Крім того, корпоративний веб-сайт може стати ефективним інструментом просування бренду, оскільки дозволяє залучити більше клієнтів за більш доступною ціною, ніж традиційна реклама [1].

У порівнянні зі звичайним сайтом, корпоративний веб-ресурс має більше функцій. Він не лише відображає інформацію про компанію та її діяльність, але й сприяє розвитку взаємовідносин з клієнтами, пошуку партнерів та організації внутрішньої та зовнішньої комунікації. Крім того, це ефективний рекламний інструмент, хоча він не має функції продажу, але зазвичай містить каталог.

На корпоративному веб-сайті можна знайти різноманітну інформацію, таку як профіль компанії, новини, контактні дані, форму для зворотного зв'язку, чати, каталоги або портфоліо, відгуки клієнтів, документи, вакансії та архів [2].

Більше половини (54%) інформаційних працівників у всьому світі відриваються від роботи кілька або більше разів на місяць, щоб витратити час на пошук або спроби отримати доступ до інформації, ідей і відповідей. Проблема: старі корпоративні пошукові системи на основі ключових слів минулого застаріли [3].

Пошукові системи в корпоративній мережі призначені для роботи з масивами текстових документів підприємства, що мають обсяги від декількох гігабайт до декількох десятків гігабайт. Такі програми розташовані на сервері локальної мережі, здійснюється з робочих станцій співробітників і мають бути реалізовані в мережному варіанті, при якому доступ до бази даних адмініструється правами доступу.

Джерелами інформації в корпоративних мережах виступають не тільки сайти і сторінки, що наповнюють ці сайти і які створені за єдиними стандартами, але і безліч інших баз даних і різних репозиторіїв структурованих і неструктурованих даних. Різноманітність джерел робить неможливим просте перенесення в корпоративне середовище звичайних пошукових запитів.

Існує кілька варіантів мережного виконання пошукової програми.

Перший і найпростіший – це можливість пошуку в мережному оточенні. Така програма може індексувати файли, що розташовані не тільки на локальному комп'ютері, але і на дисках інших робочих станцій, з'єднаних в локальну мережу. При цьому пошук може здійснюватися тільки з комп'ютера, на якому встановлена система і розташована база даних, включаючи пошуковий індекс. Багатокористувацький режим, при якому користувачі з своїх робочих місць можуть звернутися до бази даних за інформацією, не забезпечується.

Другий варіант – це пошукові системи, що працюють по Інтернет протоколу. В цьому випадку база даних і основна програма розташована на центральному сервері локальної мережі, а всі користувачі мають доступ до інформації з своїх комп'ютерів через стандартний Інтернет браузер. Тобто все відбувається так само, як і при пошуку в глобальному Інтернеті. Користувач,

працюючи в Інтранет мережі, для доступу до бази даних вводять адреси баз даних і далі шукає інформацію по стандартній схемі, із стандартним інтерфейсом пошукової системи. Природно, що пошукові програми, створені на основі пошукових Інтернет систем, в основному використовують Інтернет протоколи у випадку роботи у режимі багатокористувацький.

Наступний рівень – це програмні системи, що мають клієнт-серверну архітектуру з власною клієнтською частиною програми. Програма клієнт встановлюється на всіх робочих станціях мережі, а програма сервер забезпечує індексування інформації всієї мережі, створення бази даних на сервері та доступ до неї всіх користувачів. Такі системи складніші в розробці, але мають більше функціональних можливостей, чим системи, які використовують стандартний браузер. Наприклад, для розмежування доступу користувачів до різних видів корпоративної інформації використовуються системні засоби, а інтерфейс призначений для користувача можна зробити більш функціональним і зручним [3].

Мета створення інформаційної системи пошуку на корпоративному сайті полягає в забезпеченні швидкого, точного та зручного доступу користувачів до необхідної інформації, що зберігається на сайті. Основними цілями такої системи можуть бути наступні:

- покращення навігації користувачів: система повинна допомагати користувачам ефективно знаходити потрібну інформацію без зайвих зусиль;
- забезпечення релевантних результатів пошуку: система повинна виводити результати, які найкращим чином відповідають запитам користувачів, враховуючи семантику запиту, синоніми та контекст;
- оптимізація завантаження інформації: інформаційна система повинна працювати швидко і ефективно, незалежно від обсягу даних, що зберігаються на сайті;
- інтеграція з іншими системами: система повинна бути легко інтегрована з іншими корпоративними системами для автоматизації процесів оновлення та синхронізації інформації;

– підтримка аналітики: збір та аналіз даних про використання системи для подальшого вдосконалення її функціональності та відповідності потребам користувачів.

Процеси здійснення пошуку опишемо за допомогою методології об'єктного моделювання процесів (далі – ОРМ) в програмному середовищі «Orscat» [4].

ОРМ – це підхід до розробки програмного забезпечення, який базується на концепції об'єктно-орієнтованого програмування [5]. Методологія об'єктного процесу дозволяє розбити процес розробки програмного забезпечення на окремі етапи та кроки, кожен з яких має свої визначені цілі та завдання.

Ця методологія передбачає поділ проекту на об'єкти, які взаємодіють між собою. Кожен об'єкт має властивості та методи, які відповідають за обробку даних та взаємодію з іншими об'єктами. Даний підхід дозволяє зменшити залежність між окремими елементами програми, збільшити її стійкість до змін та розвивати програму більш ефективно.

Головною діаграмою в методології ОРМ є системна діаграма, на якій вказується головний процес інформаційної системи, його об'єкти та інші процеси (рис. 1.1).

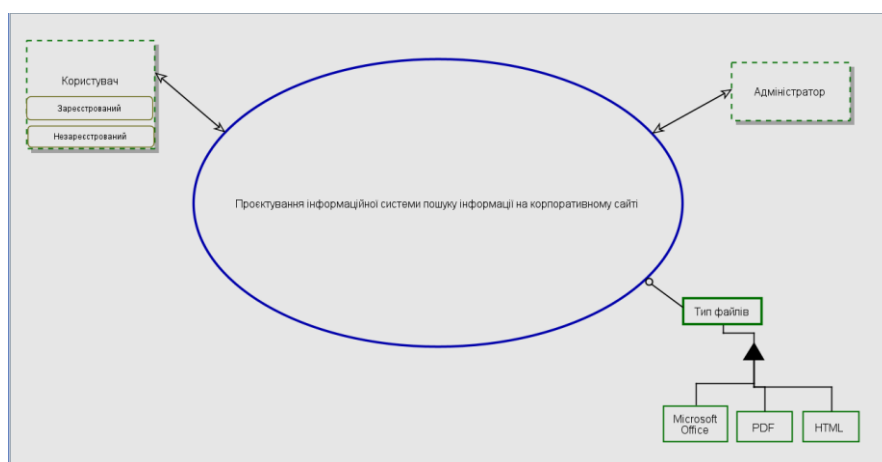


Рисунок 1.1 – Діаграма OPD верхнього рівня

Джерело: розроблено автором самостійно

Цей рівень моделі показує систему, осіб, об'єкти та їх стани. Серед осіб присутні «Користувач» що має стан «zareєстрований» та «nezareєстрований» а також «Адміністратор». Об'єкт «Тип файлів» який складається з «Microsoft Office», «PDF», «HTML» (рис. 1.2).

За допомогою unfold можна детально описати об'єкт, так на рисунку 1.2 показано розгортання об'єкту «Тип файлів».

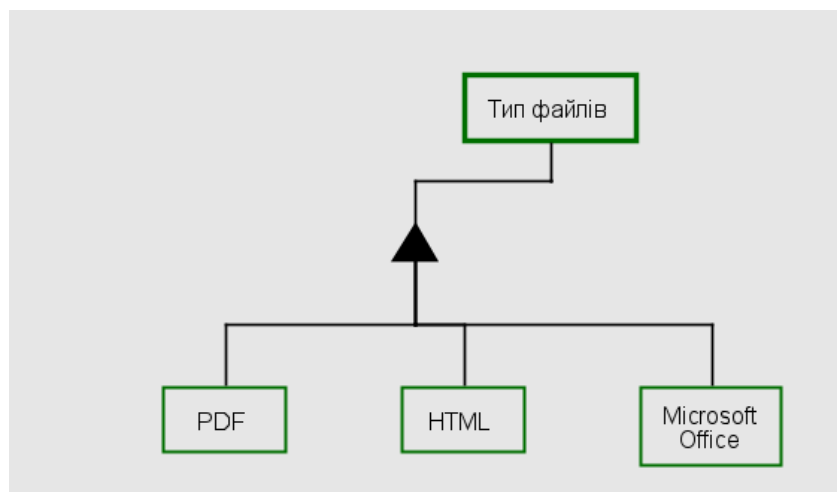


Рисунок 1.2 – Механізм розгортання

Джерело: розроблено автором самостійно

За допомогою in-zoom можна провести більш детальний огляд системи (рис. 1.3).

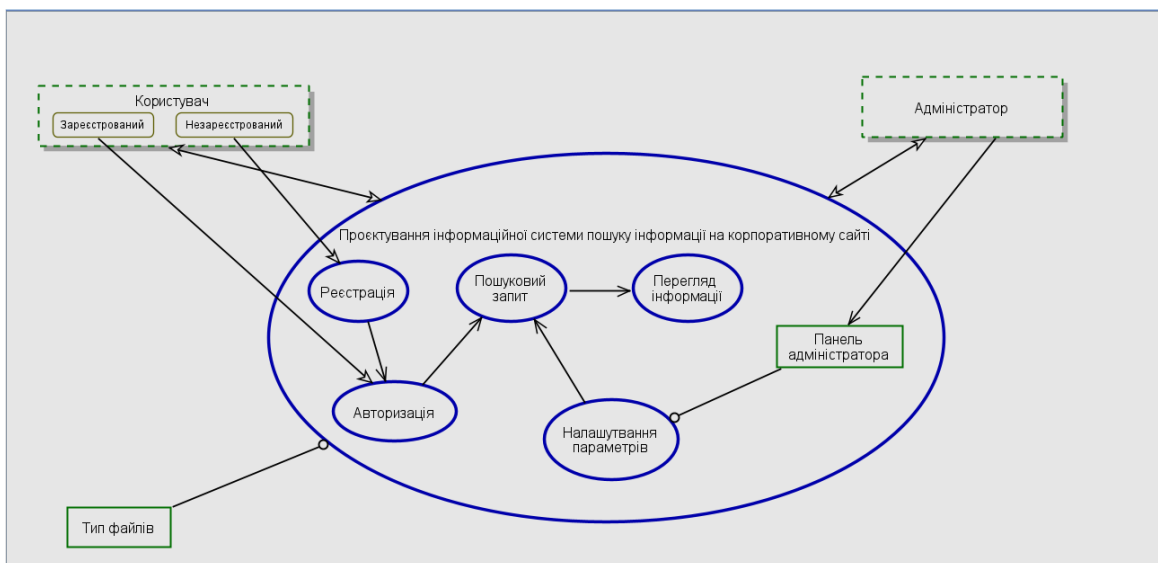


Рисунок 1.3 – Декомпозиція головного модуля

Джерело: розроблено автором самостійно

Користувач залежно від стану переходить до процесу Реєстрації/Авторизації, звідки він може формувати пошуковий запит та переглядати отриману інформацію. Адміністратор взаємодіє з об'єктом панель адміністратора для налаштування параметрів пошуку. На рисунку 1.4 – 1.8 зображені діаграми нижчого рівня (in-zoomed), в яких детально розписано основні процеси системи з відповідними зв'язками та об'єктами.

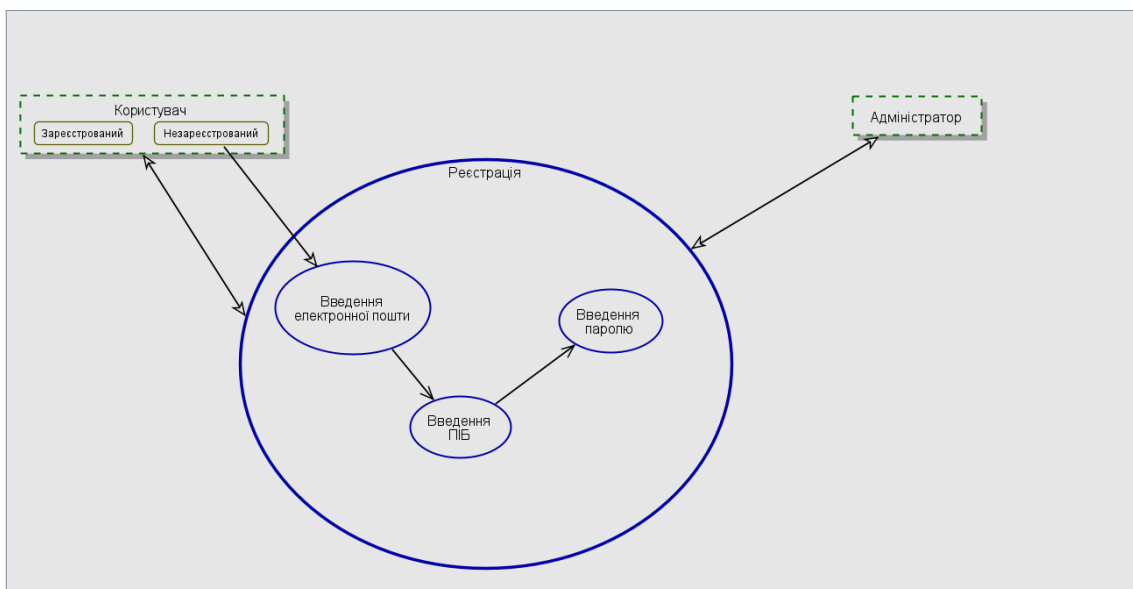


Рисунок 1.4 – Діаграма OPD нижнього рівня. Процес реєстрації

Джерело: розроблено автором самостійно

Процес реєстрації складається з процесу введення незареєстрованим користувачем електронної пошти, після чого вводиться ПІБ та придумується пароль.

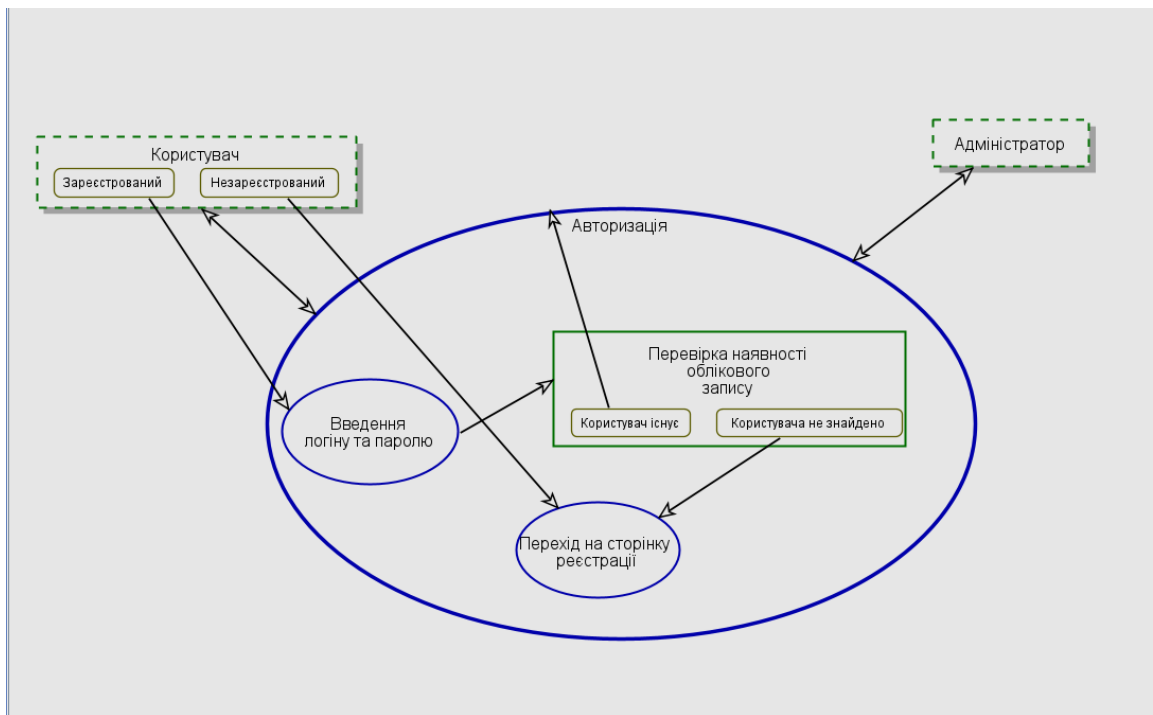


Рисунок 1.5 – Діаграма OPD нижнього рівня. Процес авторизації

Джерело: розроблено автором самостійно

Процес авторизації складається з процесу введення даних логіну та паролю після якого здійснюється перевірка наявності облікового запису і якщо користувач існує – проводиться успішна авторизація, в іншому випадку – користувача не знайдено, тоді відбувається перехід на сторінку реєстрації.

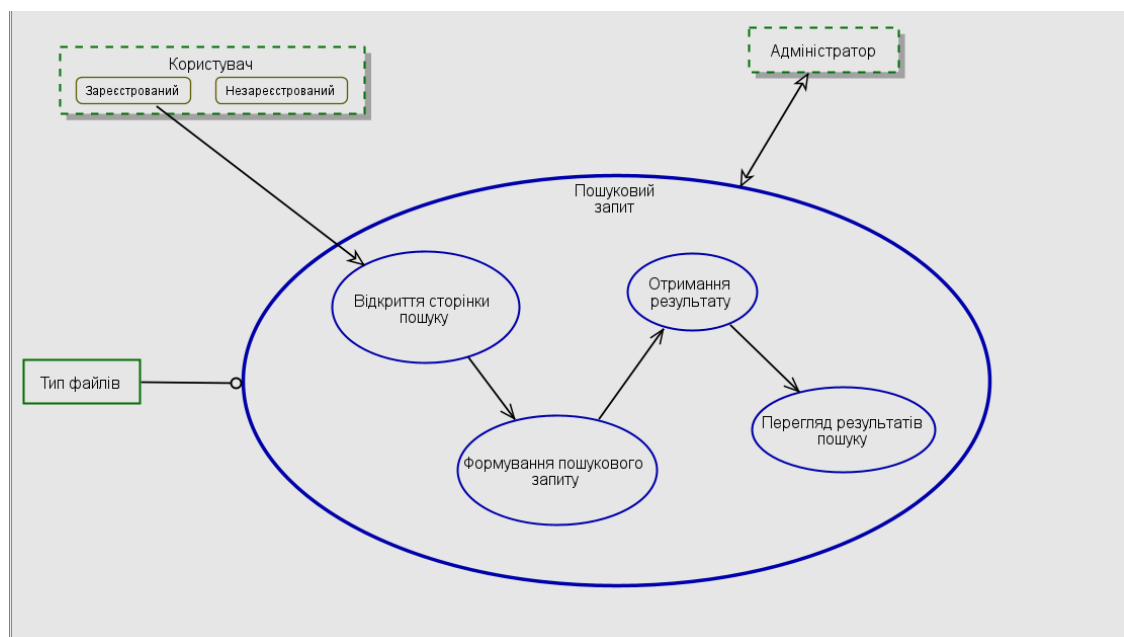


Рисунок 1.6 – Діаграма OPD нижнього рівня. Процес пошукового запиту

Джерело: розроблено автором самостійно

Процес пошукового запиту починається з відкриття користувачем сторінки пошуку де формується сам запит і отримуються результати які надалі можна переглянути.

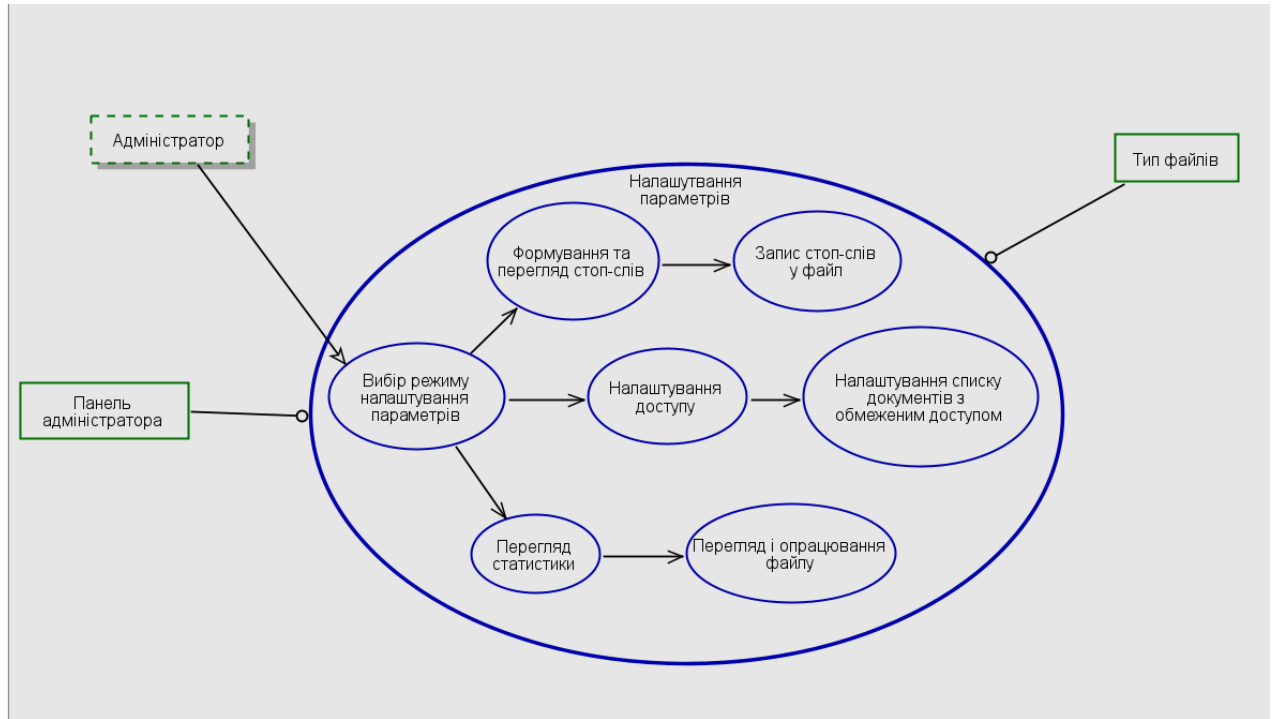


Рисунок 1.7 – Діаграма OPD нижнього рівня. Процес налаштування параметрів

Джерело: розроблено автором самостійно

Процес налаштування параметрів починається з вибору режиму налаштування адміністратором. Звідти він може: формувати та переглядати стоп-слова та записувати їх у відповідний файл, налаштовувати доступ до певних файлів та переглядати і працювати статистичні дані.

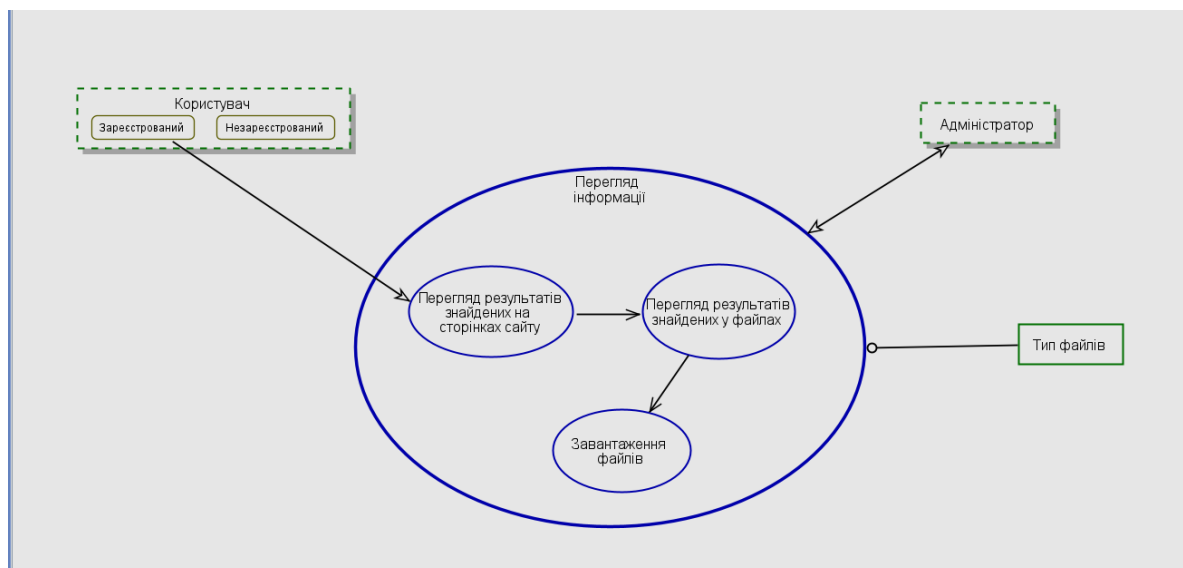


Рисунок 1.8 – Діаграма OPD нижнього рівня. Процес перегляду інформації

Джерело: розроблено автором самостійно

В процесі перегляду інформації, спочатку користувач відкриває список знайдених результатів. В системі встановлено пріоритет на сторінки сайтів над файлами, тому спочатку переглядає сторінки, а після чого переглядає результати у файлах з подальшою можливістю зберегти або переглянути результат на екрані.

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Серед багатьох популярних пошукових систем слід визначити найбільш відомі системи: dtSearch, Greenstone, Google Search Appliance, Google Custom Search, Google Desktop, Autonomy, RetrievalWare, METATEKA.

Далі розглянемо їх основні можливості.

Пошукова система **dtSearch** [6].

Програмне забезпечення dtSearch 7.0 є потужною пошуковою системою, спрямованою на знаходження інформації як у локальному, так і в мережевому середовищі. Завдяки своєму англomовному інтерфейсу, воно працює на

операційних системах Windows 9x/Me/NT/2000/XP/2003. Складається з основних модулів: dtSearch Desktop 7.0 - це головний інтерфейс програми, dtSearch Indexer - індексатор документів, dtSearch Index Library Manager - менеджер бібліотек індексів, dtSearch CD Wizard - індексатор даних, що знаходяться на CD. Основна функціональність включає створення загального індексу для кількох комп'ютерів в мережі, пошук інформації різних типів і мов, включаючи zip, rtf, pdf, html, xml, а також документи Microsoft Office (Word, Excel, PowerPoint) і WordPerfect. Програма підтримує кодування Unicode і надає різноманітні види пошуку, такі як морфологічний, фонетичний, пошук синонімів і слова з орфографічними помилками. Для лінгвістичного аналізу текстів dtSearch використовує засоби WordNet і має додаткові засоби підтримки української мови.

Система **Greenstone** [7]. Greenstone - це програмний продукт з відкритим вихідним кодом, призначений для створення "цифрових бібліотек". Вона надає широкі можливості для організації та керування колекціями документів. Система включає в себе інструменти для виконання пошуку з попереднім індексуванням документів у різних форматах, зокрема doc і pdf, а також архівних файлів. Основною функцією є створення каталогу документів, який автоматично конвертує їх у формат html. Після цього користувачі мають можливість отримати віддалений доступ до бібліотеки та інших ресурсів за допомогою веб-браузера.

Програмно-апаратний комплекс **Google Search Appliance** [8]. Програмно-апаратний комплекс Google Search Appliance [8] призначений для пошуку документів в корпоративних мережах. Пошуковий механізм комплексу оптимізований для роботи з більш ніж 200 типами файлів, включаючи html, pdf та doc. Він підтримує повнотекстовий пошук з врахуванням синонімів та працює з понад 50 мовами. Google Search Appliance також забезпечує можливість пошуку захищених даних на закритих серверах, але доступ до захищених документів можливий лише з відповідними дозволами доступу.

Google Desktop Search (GDE) - це безкоштовна локальна версія відомої пошукової системи Google. Однак як сама GDE, так і інші безкоштовні пошукові системи іноземного походження поки не ідеально підходять для текстових масивів на російській та українській мовах. Вони не враховують російську та українську морфологію, погано індексують тексти цих мов. Крім того, максимальний обсяг тексту, який може бути індексований в пошуковому індексі, у таких системах значно нижчий, ніж у висококласних пошукових системах. Наприклад, для GDE максимальний обсяг індексованого тексту обмежений 2 Гб.

Autonomy [9]. Технологія, розроблена компанією Autonomy для корпоративних систем, є інструментом для автоматизації управління інформаційними потоками. Основні наукові принципи Autonomy базуються на інформаційній теорії Клода Шеннона, байєсівських ймовірностях і нейронних мережах. Застосування концепції адаптивного моделювання ймовірності дозволяє системі Autonomy виявляти шаблони в тексті документів і автоматично визначати подібні шаблони в інших документах. Під час обробки шаблонів у документах система Autonomy виявляє кореляції між образами і встановлює закономірності серед великих обсягів документів, при цьому не застосовуючи жодних специфічних правил, включаючи лінгвістичні. Оскільки система не базується на заздалегідь визначених ключових словах, вона може працювати з будь-якою мовою.

Одним з продуктів Autonomy є пакет Portal-in-a-box, який, крім традиційних функцій агрегації інформації з різних джерел, має інструменти для вирішення проблеми систематизації неструктурованих даних, що виникає при побудові порталів. Угрупування документів за категоріями і створення їх метаописів часто потребує значних редакційних зусиль. У таких випадках Portal-in-a-box повністю автоматизує процеси категоризації інформації, її реферування та розстановки гіперпосилань.

Інформаційно-пошукова система **RetrievalWare** [10].

RetrievalWare є інструментом для повнотекстового і атрибутного пошуку. Система може обробляти різні типи документів у різних форматах та кодуваннях, включаючи тексти, електронні таблиці, бази даних, поштові повідомлення тощо - більше 200 різних форматів. RetrievalWare також має інструменти для налаштування підтримки специфічних форматів документів. Масштаб архіву може досягати терабайтів. Архітектура RetrievalWare дозволяє працювати як у корпоративній локальній мережі, так і через Інтернет. Серверна частина підтримує усі популярні серверні платформи, а клієнтська частина може бути запущена на будь-якому комп'ютері з графічним веб-браузером.

Система RetrievalWare може працювати в різних багатопроесорних і розподілених багатосерверних конфігураціях. Джерелами інформації можуть бути файли, системи управління базами даних (наприклад, MS SQL, ORACLE, Sybase), поштові системи (наприклад, Microsoft Exchange, Lotus Notes), системи управління документами (наприклад, Documentum EDMS, FileNET Panagon), вузли корпоративної мережі та Інтернет, а також електронний архів Excalibur File Room, який дозволяє організувати доступ до паперових документів.

Основу технології RetrievalWare становить адаптивне розпізнавання образів на основі нейронних мереж, яке дозволяє системі індексувати та організовувати бінарні образи документів. Використання цієї технології дозволяє здійснювати нечіткий пошук, бути мовно-незалежною та мінімізувати розмір індексних файлів. Також в системі реалізована підтримка російської морфології, включаючи семантичну мережу словника, що враховує близько 40 тисяч семантичних груп. Використання семантичних мереж дозволяє користувачам вводити запити на природній мові, а система автоматично шукає всі документи, контекст яких відповідає контексту запиту.

InfoStream [11].

Центральним елементом обробки контенту в системі InfoStream є повнотекстова інформаційно-пошукова система InfoReS. Технологія InfoStream дозволяє створювати повнотекстові бази даних, виконувати пошук інформації, формувати тематичні інформаційні канали, автоматично створювати рубрики

інформації, формувати дайджести, а також аналізувати взаємозв'язки понять у мережевих публікаціях через таблиці і гістограми.

За допомогою InfoStream можна обробляти дані у різних форматах, включаючи Microsoft WORD (версії 2000, 97, 6), rtf, pdf, а також всі текстові формати, такі як простий текст, html і xml. Системи, що базуються на InfoStream, в даний час підтримують операційні системи, такі як FreeBSD, Linux і Solaris.

Технології InfoStream дозволяють створити комплексну систему підтримки документального інформаційного сховища, що інтегрує інформаційно-пошукове середовище на базі веб-рішень. Цей комплекс надає доступ до електронних документів, розміщених на комп'ютерах у корпоративній мережі, з можливістю пошуку, навігації по комп'ютерах та каталогах, а також перегляду як оригінальних документів, так і їх текстових варіантів.

Система забезпечує можливість ефективного повнотекстового пошуку інформації за складними запитамі, що включають ключові слова, логічні та контекстні оператори, а також різноманітне ранжування результатів пошуку. Вона також дозволяє уточнювати результати за допомогою механізму "інформаційних портретів".

SearchInform Server є оптимальним інструментом для пошуку в корпоративній мережі. Пошук проводиться не лише за файлами у локальній мережі, а й за архівами поштових клієнтів, базами даних та іншими джерелами. Забезпечується швидкий доступ до інформації у всій мережі, навіть при великій кількості одночасних запитів. Унікальна технологія пошуку дозволяє легко виявляти документи-дублікати та скорочувати час пошуку, максимально точно визначаючи параметри запиту. Впровадження SearchInform не вимагає змін в існуючих бізнес-процесах і дозволяє економно використовувати інвестиції в інформаційну інфраструктуру компанії.

Пошукова система "МЕТАТЕКА" компанії "Мета" [12].

Пошукову систему "METATEKA" можна відзначити серед українських розробок у цій галузі. Вона доступна для операційних систем Windows NT/2000/XP/2003 та Linux/FreeBSD, і використовує універсальний синтаксис для індексації ресурсів. Структурно "METATEKA" виглядає стандартно: складається з трьох основних компонентів, включаючи пошуково-індексуєчий механізм у вигляді системного сервісу, і два CGI-скрипти для адміністрування та виконання пошуку. Для обробки файлів у спеціальних форматах (DOC, RTF, XLS, PDF, а в майбутньому - інші) існують підключені фільтри, а також словники для англійської, української та російської мов. Важливо зазначити, що "METATEKA" вирізняється серед подібних продуктів повноцінною підтримкою морфології словника (особливо для української мови), що разом з мовою запитів гарантує високу ефективність пошуку. У системі, крім стандартних опцій, таких як плюси, мінуси і лапки, підтримуються круглі дужки для групування слів і керування порядком виконання логічних операторів, а також фігурні дужки для пошуку виразів з урахуванням словозміни.

Ядро "METATEКИ" призначене для роботи з великими обсягами інформації, і може одночасно оновлювати індекс та обробляти запити користувачів. Технології пошуку інформації, які існують на ринку, спрямовані на вирішення основної задачі - пошук інформації у великих повнотекстових базах даних. Ці системи можуть включати різноманітні джерела текстової інформації, такі як енциклопедії, довідники, архіви періодичних видань, бібліотеки, корпоративні документи та інше. Після введення конкретного запиту система видає множину посилань. Далі вона обробляє кожне посилання, щоб знайти відповідні тексти, тобто система шукає не просто документи, а інформацію, що міститься в них.

Головна мета цих систем - це проведення пошуку інформації у великих повнотекстових обсягах [13]. У базі даних таких систем можуть бути різноманітні текстові джерела, включаючи енциклопедії, довідники, архіви періодичних видань, повні бібліотеки фахової літератури, архіви

корпоративних документів, спеціалізовані архіви, такі як історичні, патентні, судові, розшифровки розмов, протоколи та багато іншого. При отриманні конкретного запиту система надає набір посилань. Далі система аналізує кожне посилання та виводить відповідні тексти, що означає, що система шукає не лише документи, а й саму інформацію, що міститься у них.

Існуючі технології пошуку не є досить ефективними для знаходження інформації, що відповідає запиту, у великій кількості різноманітної інформації в глобальній мережі. Тому розвивається дворівневий процес пошуку: перший - це тематичний пошук і відбір даних у повнотекстовій базі даних, другий - це пошук у повнотекстовій базі даних. Усі пошукові системи Інтернету побудовані за принципом "одного кроку релевантності" (Single Shot Relevancy), тобто "релевантність з першого попадання": ми робимо один запит і одержуємо відповідні результати. Ми можемо вносити деякі зміни у запит, але спеціальних механізмів зворотного зв'язку не передбачено.

Підхід Single Shot Relevancy відповідає поняттю "пошук", що ґрунтується на ідеї наближеності, забезпечуючи при цьому достовірність. В корпоративному середовищі, де точність набуває не меншого значення, ключовим стає поняття "знаходження". Тут важливо знайти саме той документ, який користувач шукає і про який він вже має попередню інформацію.

Для підвищення точності пошуку за допомогою запитів (аналіз "drill down" - аналіз з підвищеною деталізацією) необхідно зробити процес пошуку "інтерактивним". Користувач повинен мати можливість коригувати свої запити з використанням зворотного зв'язку і поступово досягати необхідної йому точності.

Для ефективності такої процедури необхідно впровадити нові методи представлення результатів пошуку. Простий список сторінок з короткими анотаціями, як це роблять звичайні пошукові машини, вже недостатній. Результати можуть бути представлені у текстовій формі, з вказівкою зв'язків між окремими компонентами. Великі можливості також є у використанні графічних форматів для представлення результатів пошуку.

Особливу увагу слід звернути на обробку мультимедійних даних, оскільки наразі жодна пошукова система не здатна ефективно працювати з такими даними. Розв'язання цієї проблеми вимагає впровадження якісно нових підходів. Сучасні пошукові системи базуються на кількох ключових принципах: дані зберігаються у вигляді окремих документів, тому для збільшення продуктивності можна використовувати розподілену архітектуру; шляхом використання метаданих можна якимось чином визначити значення документа і, базуючись на моделях, досягти необхідної релевантності.

РОЗДІЛ 2

РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної підсистеми

2.1 Бізнес-вимоги до системи

Для визначення бізнес-вимог необхідно визначити стейкхолдерів та їх бізнес-цілі, до яких можна віднести бізнес-вимоги.

Визначені стейкхолдери інформаційної системи пошуку інформації на корпоративному сайті наступні:

- Верховне керівництво компанії. Включає у себе генерального директора, керівників відділів та інших осіб, які відповідають за стратегічне напрямку розвитку компанії. Вони зацікавлені у використанні системи для підтримки стратегічного прийняття рішень та досягнення бізнес-цілей.
- IT-відділ компанії. Цей відділ відповідає за технічну підтримку та розвиток інформаційних систем компанії. Вони зацікавлені у тому, щоб система корпоративного пошуку була технічно надійною, масштабованою та відповідала вимогам безпеки та інтеграції з існуючими системами.
- Користувачі системи. Це співробітники компанії, які використовують систему корпоративного пошуку для пошуку інформації у робочому процесі. Вони зацікавлені у тому, щоб система була зручною у використанні, швидкою та надійною.
- Адміністратори системи. Це особи, які відповідають за управління та підтримку системи корпоративного пошуку, включаючи налаштування, моніторинг та розвиток. Вони зацікавлені у простоті адміністрування та ефективності системи.
- Бізнес-аналітики. Це фахівці, які аналізують дані та звіти, щоб надати бізнес-відомості та рекомендації для управління та прийняття рішень.

Вони зацікавлені у точності та повноті інформації, яку надає система корпоративного пошуку.

– Зовнішні партнери та клієнти. Це особи або організації, які мають взаємодію з компанією як партнери або клієнти. Вони можуть бути зацікавлені у доступі до певної інформації через систему корпоративного пошуку або взаємодії з компанією через цю систему.

Бізнес-цілі:

- підвищення продуктивності співробітників;
- покращення процесів прийняття рішень;
- зменшення часу пошуку інформації;
- підвищення рівня задоволеності клієнтів;
- підвищення конкурентоспроможності компанії;
- покращення якості прийняття рішень.

На основі перелічених бізнес-цілей визначимо бізнес-вимоги, перелік яких покажемо на діаграмі бізнес-вимог (рис. 2.1), розроблених в середовищі Enterprise Architect [14].

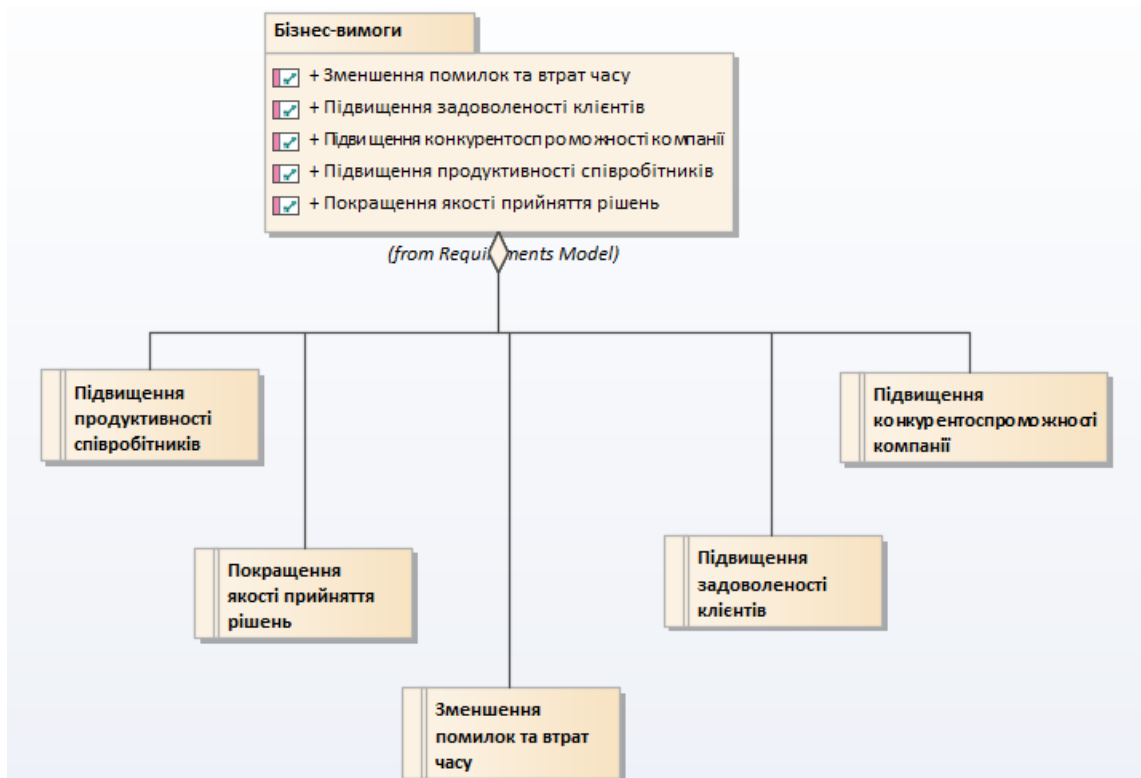


Рисунок 2.1 – Бізнес-вимоги до системи

Джерело: розроблено автором самостійно

2.2 Функціональні вимоги до системи

Функціональні вимоги визначають, як система має взаємодіяти з користувачем та які функції мають бути доступні. Вони описують функції, які система повинна виконувати, щоб задовольнити потреби користувачів.

Функціональні вимоги розділимо на вимоги для користувачів, адміністратора та системи, які показані на рис. 2.2-2.4. Зокрема, користувач в системі може здійснювати вхід в систему (авторизація), налаштування (зміна мови, налаштування інтерфейсу, параметрів відображення результатів), вводити пошуковий запит, переглядати результати пошуку. Адміністратор – налаштовувати систему і формувати кеш-файли. Система здійснює пошук за алгоритмом та генерує результати.

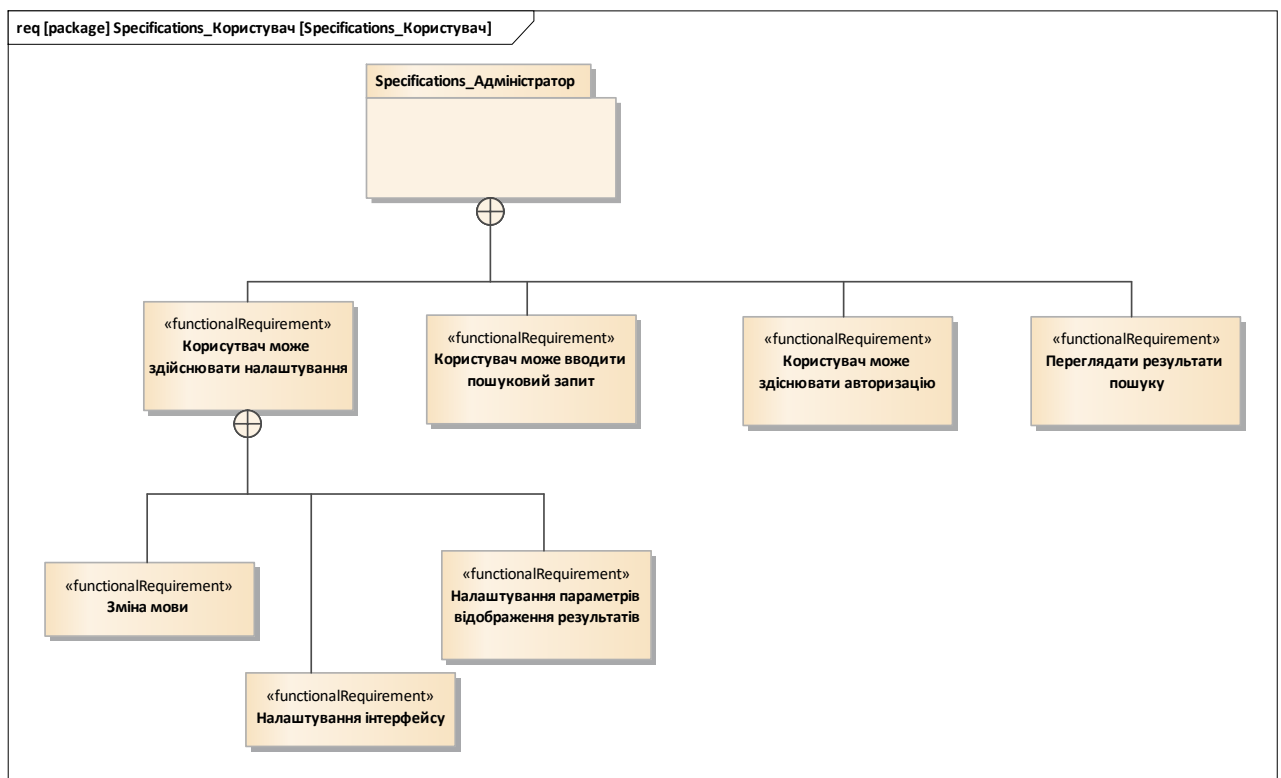


Рисунок 2.2 – Діаграма функціональних вимог для користувача

Джерело: розроблено автором самостійно

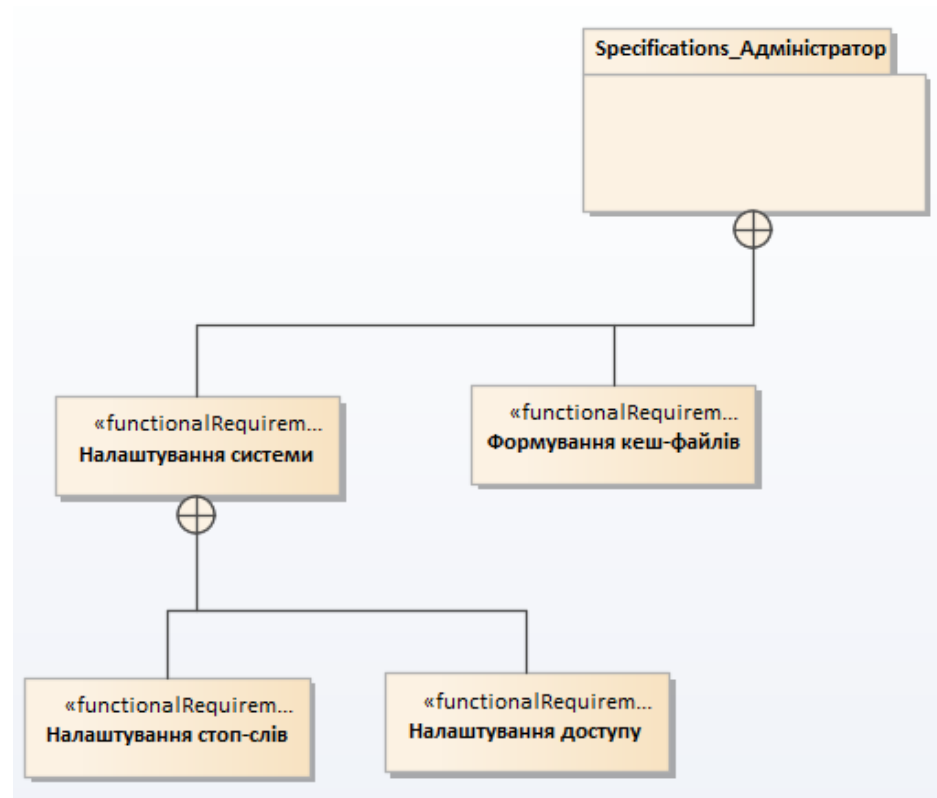


Рисунок 2.3 – Діаграма функціональних вимог для адміністратора

Джерело: розроблено автором самостійно

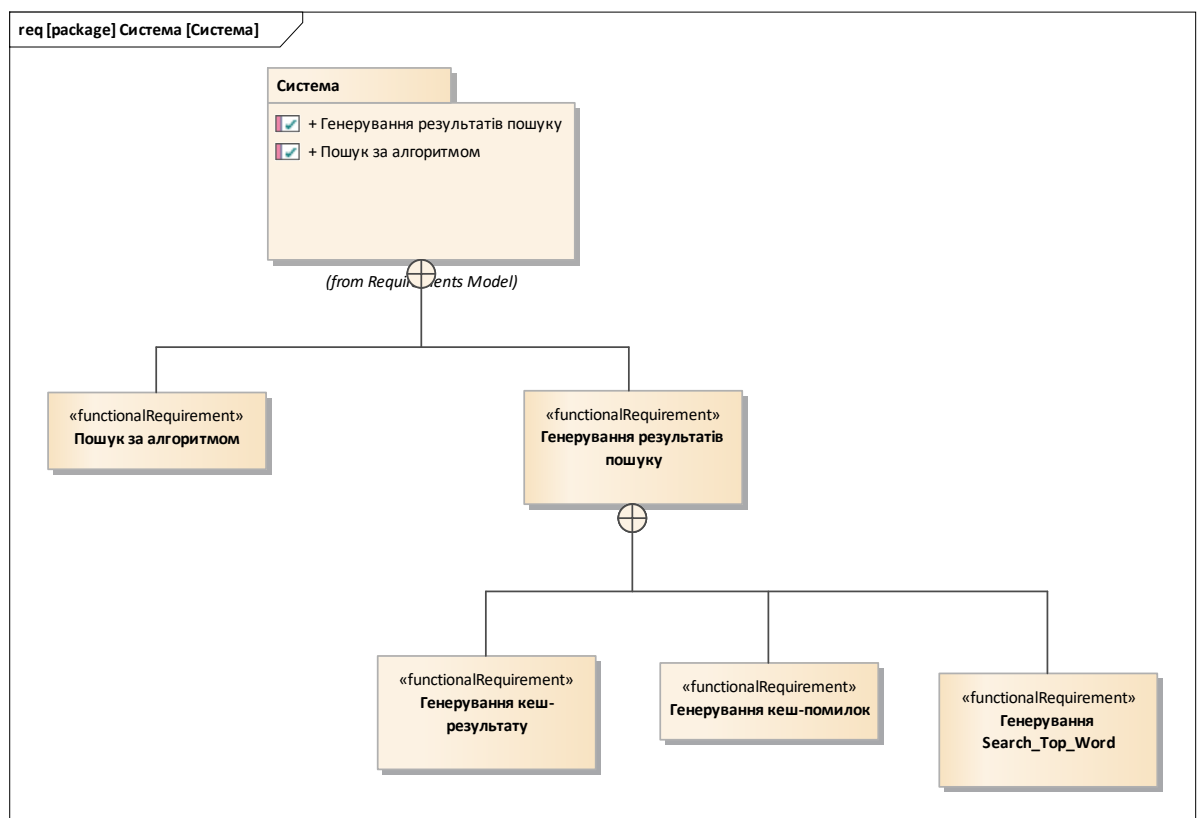


Рисунок 2.4 – Діаграма функціональних вимог для системи

Джерело: розроблено автором самостійно

2.3 Нефункціональні вимоги до системи

Нефункціональні вимоги описують обмеження та вимоги, не пов'язані безпосередньо з функціональністю системи. Вони описують властивості, які система повинна мати, але які не є функціональними. Діаграма нефункціональних вимог показана на рис. 2.5.

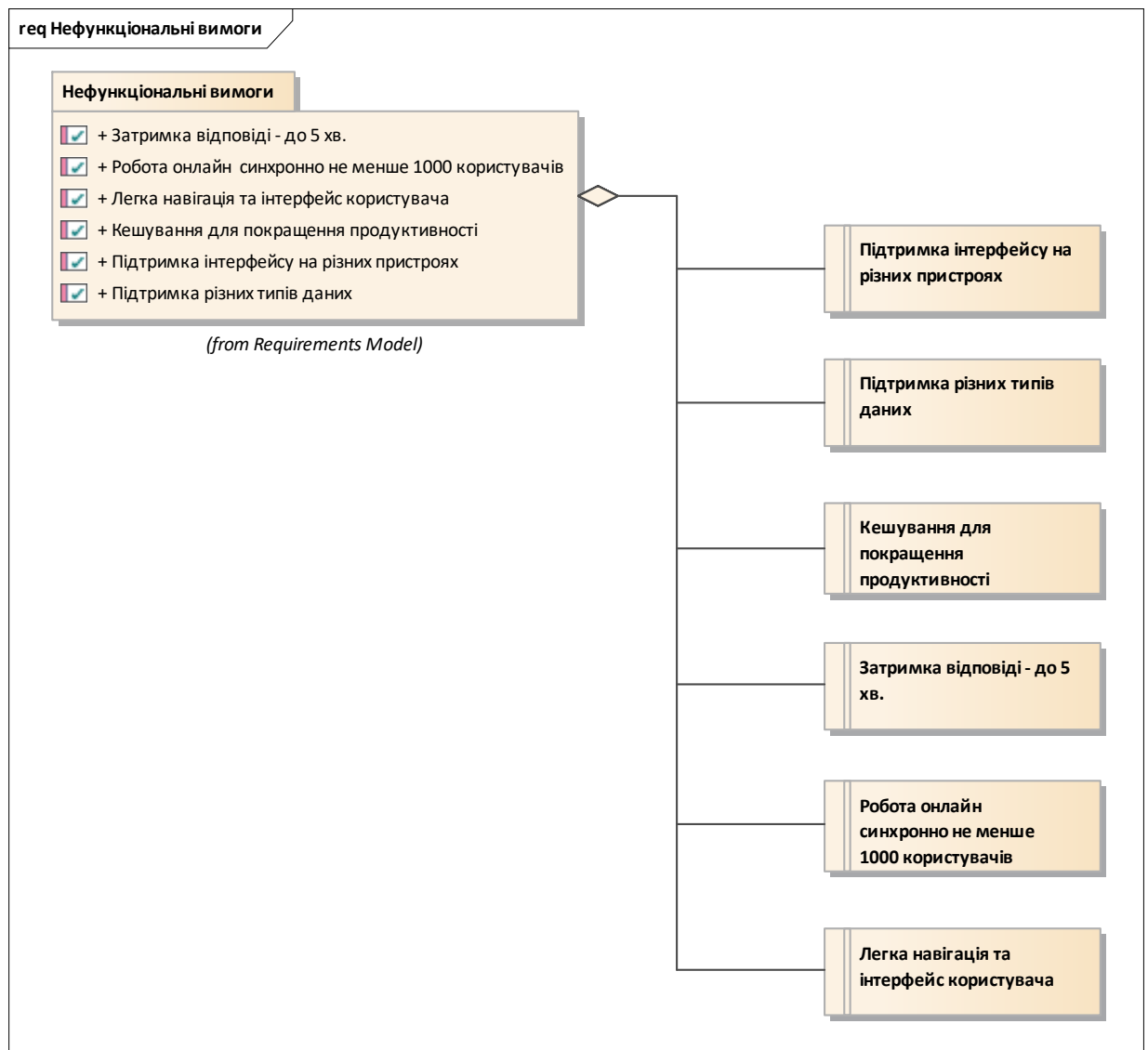


Рисунок 2.5 – Діаграма нефункціональних вимог

Джерело: розроблено автором самостійно

2.2 Постановка задачі та алгоритм розв'язання задачі

2.2.1 Постановка задачі

Характеристика задачі.

Предметна область даної задачі полягає в розробці модуля пошуку для корпоративного веб-сайту. Ця задача передбачає впровадження комплексного пошукового рішення, яке забезпечить користувачам зручний доступ до інформації на веб-сайті, забезпечуючи при цьому високу ефективність і точність пошуку.

Основним завданням цього модуля є надання користувачам зручного та ефективного інструменту для пошуку інформації на сайті. З урахуванням особливостей розглянутої предметної області, пошуковий модуль має забезпечувати швидкий доступ до інформації, яка міститься як на веб-сторінках, так і у документах різних форматів.

Метою проекту є створення модуля пошуку для веб-сайту. Основною функцією модуля є здійснення пошуку за допомогою ключових слів, які вводяться користувачем. При введенні ключового слова чи фрази модуль повинен здійснити перехід на сторінку з результатами пошуку.

Для реалізації модуля необхідно виконати наступні вимоги.

1. Реалізувати алгоритм пошуку, який буде шукати вказані ключові слова в тексті всіх файлів на сайті, включаючи файли у форматах PDF, DOC, XLS, DOCX, XLSX, PPT, PPTX.
2. Забезпечити можливість пошуку як на англійській, так і на німецькій мовах, оскільки основні мови сайту - англійська та німецька.
3. Реалізувати можливість для адміністратора сайту генерувати кеш результатів пошуку. Це необхідно для оновлення результатів після додавання нових файлів або внесення змін на сторінках сайту.

4. Забезпечити можливість встановлення правил індексації сторінок та документів. Деякі сторінки та документи можуть бути виключені з результатів пошуку і не індексуватися.

5. Результати пошуку повинні виводитися у відповідності до встановлених пріоритетів: спочатку - результати, знайдені на сторінках сайту, потім - у файлах. Також важливо врахувати мовні налаштування, спочатку виводити результати на обраній мові, а потім - на інших.

Використання модуля пошуку дозволить забезпечити ефективний та швидкий пошук інформації, навіть у великих обсягах даних.

Підвищення швидкості та точності пошуку дозволяє користувачам легко та швидко знаходити необхідну інформацію на веб-сайті. Ефективне використання пошукового модуля допоможе у підтримці актуальності та релевантності результатів пошуку шляхом швидкої індексації та оновлення інформації на веб-сайті. Крім того, автоматизований процес пошуку спрощує роботу адміністраторів, оскільки дозволяє легко керувати та налаштовувати параметри пошуку, такі як правила індексації та фільтрації. Таким чином, використання ЕОМ для вирішення даної задачі призводить до підвищення продуктивності та задоволення потреб користувачів в ефективному пошуку інформації.

Об'єкти, за управлінням яких розв'язується задача пошуку на корпоративному веб-сайті, включають:

- користувачі веб-сайту: потреби та вимоги користувачів визначають параметри та функціональні можливості пошукової системи.
- технічна підтримка: забезпечує консультації користувачам щодо використання пошукового інтерфейсу та вирішення технічних проблем.
- адміністрація веб-сайту: налагоджує та керує параметрами пошукової системи, встановлює правила індексації та фільтрації результатів.

- адміністратори баз даних: відповідають за збереження та оптимізацію баз даних, які використовуються для пошуку інформації.
- системні адміністратори: налагоджують та підтримують інфраструктуру, на якій працює веб-сайт та пошукова система.
- управління: визначає стратегічні напрямки розвитку та вирішує ключові питання щодо ресурсного забезпечення та ефективності пошукової системи.

Вихідна інформація призначена для забезпечення ефективного та зручного доступу користувачів до необхідної інформації після виконання пошукового запиту на корпоративному веб-сайті.

Крім основного призначення – надання інформації, який відповідає запиту користувача, вихідна інформація використовується для фільтрації та сортування результатів пошуку відповідно до вимог користувача, забезпечуючи точність та релевантність знайдених даних. Вона також допомагає у підтримці навігації, надаючи користувачам зручні інструменти для швидкого перегляду та переміщення між сторінками та документами.

Крім того, вихідна інформація може бути використана для аналізу тенденцій у запитах користувачів, оцінки ефективності пошукової системи та генерації звітів для адміністраторів. Це допомагає покращити якість та продуктивність процесу пошуку на веб-сайті, забезпечуючи задоволення потреб користувачів.

Розв'язання задачі пошуку та видача вихідної інформації на корпоративному веб-сайті можуть мати різні періодичності, в залежності від потреб користувачів. Однак, загальною практикою є проведення пошуку та видача результатів у реальному часі або з мінімальною затримкою.

Термін видачі вихідної інформації може варіюватися в залежності від обсягу і складності пошукового запиту, а також від технічних обмежень та часу, необхідного для обробки даних. Зазвичай, видача результатів пошуку на

корпоративному веб-сайті відбувається протягом декількох секунд до кількох хвилин після виконання запиту.

Автоматизоване розв'язання задачі пошуку на корпоративному веб-сайті може припинитися в наступних випадках.

Помилки в роботі програмного забезпечення. Якщо виникають помилки або збої в роботі програмного забезпечення, яке відповідає за автоматизований пошук, це може призвести до припинення процесу розв'язання задачі.

Виникнення технічних проблем. Технічні проблеми, такі як відмова сервера, проблеми з мережею або недостатній обсяг пам'яті, можуть призвести до припинення автоматизованого розв'язання задачі.

Відсутність даних у базі даних. Оскільки пошукова система не матиме необхідної інформації для надання користувачам релевантних результатів, це може стати серйозною перешкодою для автоматизованого розв'язання задачі пошуку.

Відсутність необхідних ресурсів. Якщо виникає нестача ресурсів, таких як обсяг пам'яті, потужність обчислювальних пристроїв або доступ до даних, це може призвести до припинення автоматизованого розв'язання задачі.

На рисунку 2.6 показана інформаційна модель задачі.

Під час щоденного функціонування системи пошуку, персонал відповідає за моніторинг та аналіз роботи системи, виявлення можливих проблем та вирішення їх. Технічні засоби автоматизовано забезпечують пошук інформації, оновлюють бази даних та виконують інші завдання, пов'язані з функціонуванням системи. У випадку відмови або збою, персонал проводить аналіз причин та вживає заходів для відновлення роботи системи, в той час як технічні засоби автоматично виконують процес відновлення та резервного копіювання даних.

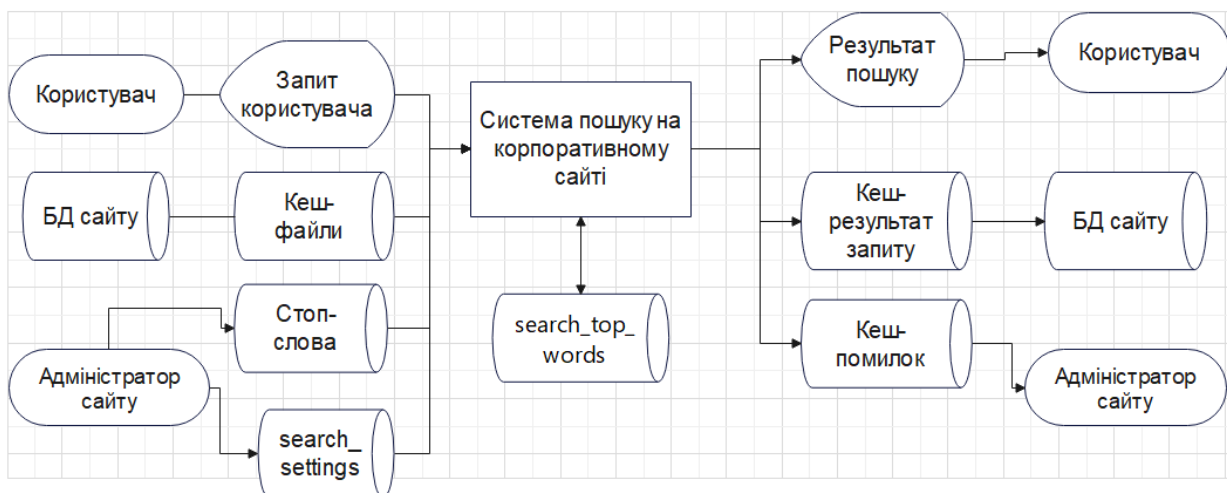


Рисунок 2.6 – Інформаційна модель задачі

Джерело: розроблено автором самостійно

Вихідна інформація.

Вихідна інформація призначена для надання користувачам точних, релевантних та швидких результатів після виконання пошукового запиту на корпоративному веб-сайті. Її головна мета - забезпечити зручний та ефективний доступ до необхідної інформації без зайвих зусиль користувачів. Вихідна інформація включає в себе результати пошуку, які повинні бути чітко представлені та зрозумілі, щоб користувачі могли швидко знайти потрібну інформацію без зайвих зусиль.

Перелік і опис вихідних повідомлень описаний у таблиці 2.1

Таблиця 2.1 – Перелік і опис вихідних повідомлень

№ з/п	Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність видання	Термін видання і допустимий час затримки	Користувачі інформації
1	Результат пошуку по сайту	Result	Екранна форма	За вимогою	У реальному часі з мінімальною затримкою	Користувач
2	Результат пошуку кешований	Cache_Result	Файл	За вимогою	У реальному часі з мінімальною затримкою при формуванні запиту	БД сайту
3	Кеш-файл	Cache_error	Файл	За вимогою	У реальному	Адміністр

	ПОМИЛОК та технічних збоїв				часі з мінімальною затримкою при формуванні запиту	ратор сайту
4	Статистика запитів	Search_top_w ords	Файл БД	За вимогою	У реальному часі з мінімальною затримкою при формуванні запиту	Адмініст ратор сайту

Джерело: розроблено автором самостійно

Вхідна інформація.

Вхідна інформація відіграє ключову роль у роботі інформаційної системи пошуку корпоративного сайту. Запит користувача (Request_search) є основним елементом, який виникає при введенні запиту користувачем через спеціалізовану екранну форму. Цей запит визначає пошукові параметри і ініціює пошуковий процес в системі.

Кеш-файли бази даних сайту (Cache_files) забезпечують швидкий доступ до попередньо збережених копій документів сайту. Ці файли автоматично оновлюються в реальному часі з мінімальною затримкою, що дозволяє забезпечити ефективність пошуку.

Список неважливих слів (стоп-слів) (Stop_words) використовується для виключення з пошуку слів, які не мають значущості або які можуть спричинити непотрібні результати.

Налаштування доступу до документів (search_settings) визначають параметри доступу до різних документів сайту, що допомагає забезпечити оптимальний рівень безпеки та ефективності системи.

Статистика запитів (Search_top_words) слугує для збору та аналізу даних про популярність та частоту запитів користувачів, що допомагає адміністратору в покращенні алгоритмів пошуку.

Отже, вхідна інформація отримується різними способами: запити користувачів формуються через екранну форму на сайті; кеш-файли бази даних

автоматично оновлюються зі змінами на сайті; список стоп-слів та налаштування доступу вводяться та оновлюються адміністратором сайту; а статистика запитів збирається автоматично при формуванні запитів користувачів.

Перелік і опис вхідних повідомлень показаний в таблиці 2.2.

Таблиця 2.2 – Перелік вхідних повідомлень

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
Запит користувача	Request_search	Екранна форма	У реальному часі з мінімальною затримкою	Користувач
Кеш-файли бази даних сайту	Cache_files	Файл	У реальному часі з мінімальною затримкою	БД документів сайту
Список неважливих слів (стоп-слів)	Stop_words	Файл	За потребою	Адміністратор сайту
Налаштування доступу до документів	Search_settings	Файл	За потребою	Адміністратор сайту
Статистика запитів	Search_top_words	Файл БД	У реальному часі з мінімальною затримкою при формуванні запиту	Адміністратор сайту

Джерело: розроблено автором самостійно

2.2.2 Математичне забезпечення та алгоритм функціонування системи

При розв'язанні задачі автоматизованим способом ведеться статистика пошукових запитів, яку за допомогою формули можна представити таким чином:

$$K_s = \sum_i KD_{si}, \text{ де}$$

K_s – Кількість документів, що відповідають пошуковому запиту s .

KD_{si} – i -документ, який відповідає пошуковому запиту s .

На рисунку 2.7 показаний алгоритм роботи адміністратора сайту під час налаштування параметрів.

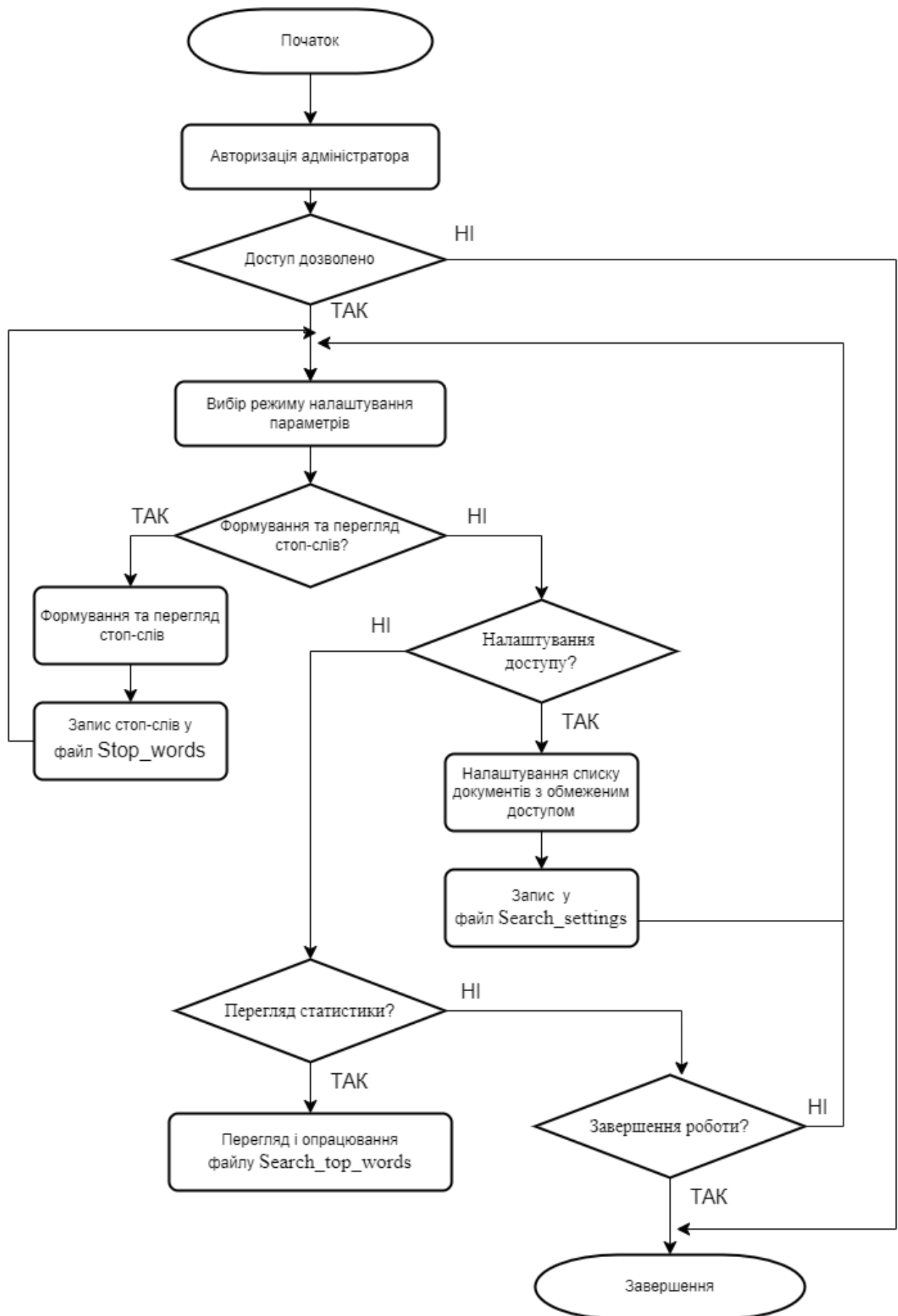


Рисунок 2.7 – Алгоритм роботи з системою користувача з правами доступу адміністратора

Джерело: розроблено автором самостійно

2.3 Моделювання інформаційної підсистеми

2.3.1 Моделювання поведінки системи

Основними функціями, які можна виконувати в інформаційній системі пошуку на корпоративному сайті є: створення пошукового запиту, переглядати результат запиту, здійснювати налаштування, здійснювати пошук.

Моделювання виконаємо в середовищі Enterprise Architect [14], яка підтримує мову SysML. Моделювання здійснено на основі дослідження предметної області та літературних джерел [16-18].

На рисунку 2.8 показано діаграму варіантів використання інформаційної системи пошуку на корпоративному сайті.

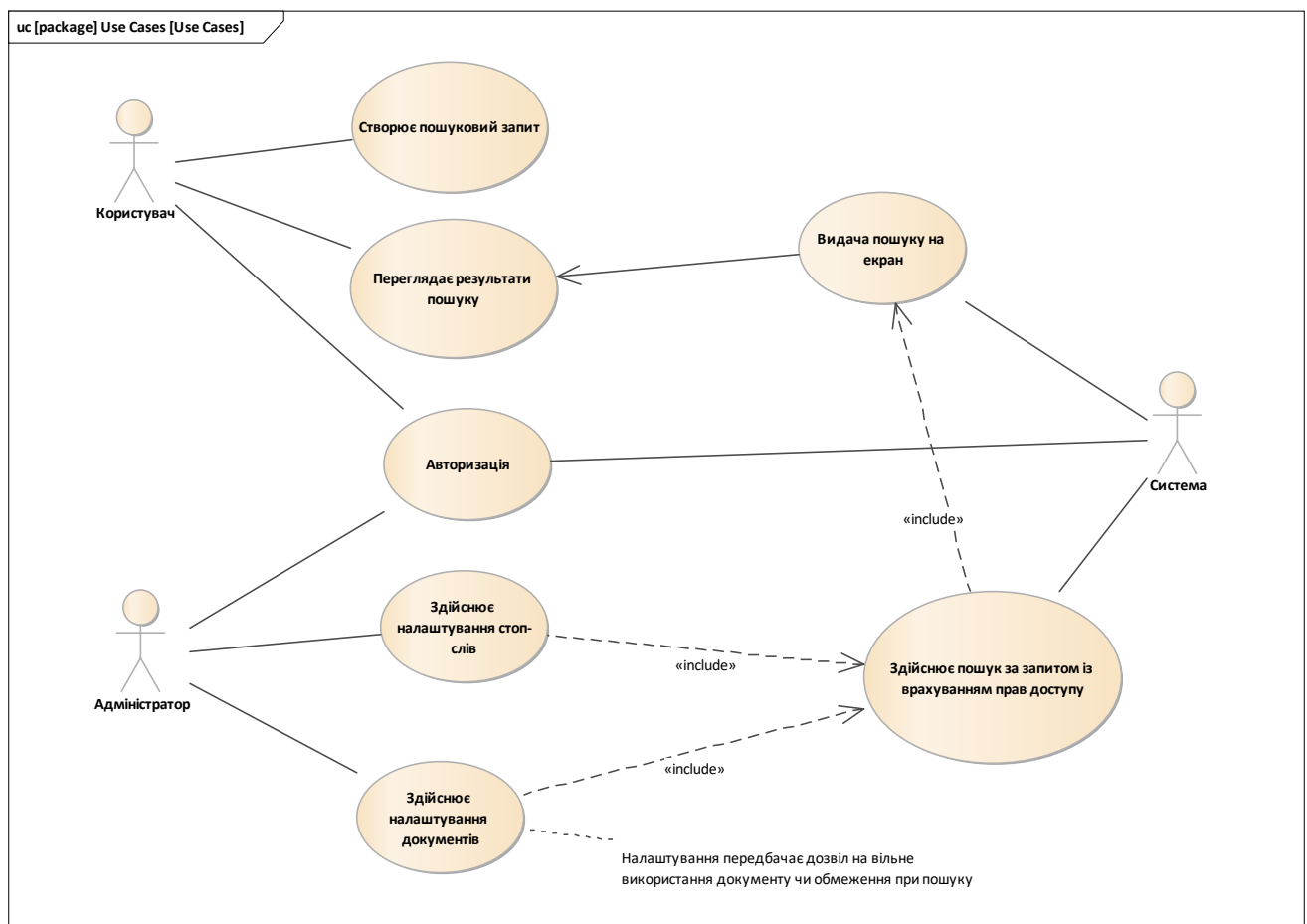


Рисунок 2.8 – Діаграма варіантів використання

Джерело: розроблено автором самостійно

Розглянемо сценарії для деяких варіантів використання. На рис. 2.9 показаний сценарій прецедента «Авторизація». Якщо логін чи пароль під час авторизації вказаний не вірно, розглядаємо альтернативний сценарій (рис. 2.10, 2.11).

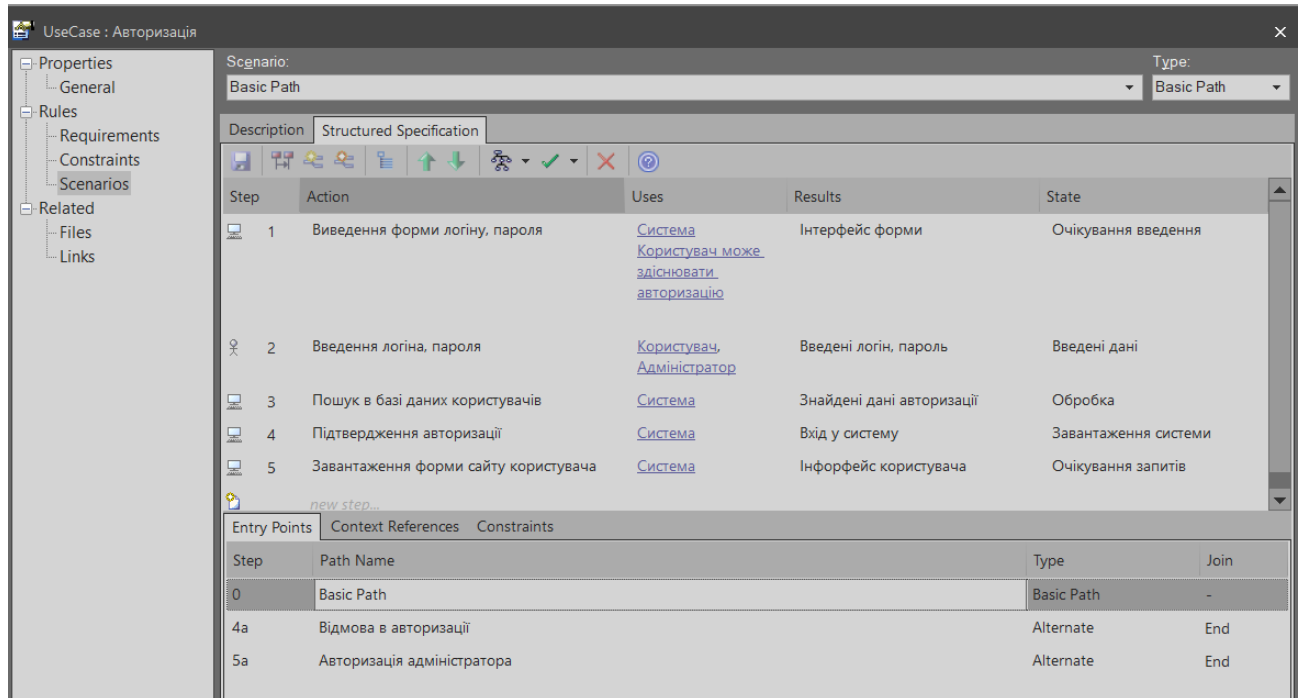


Рисунок 2.9 – Сценарій прецедента «Авторизація»

Джерело: розроблено автором самостійно

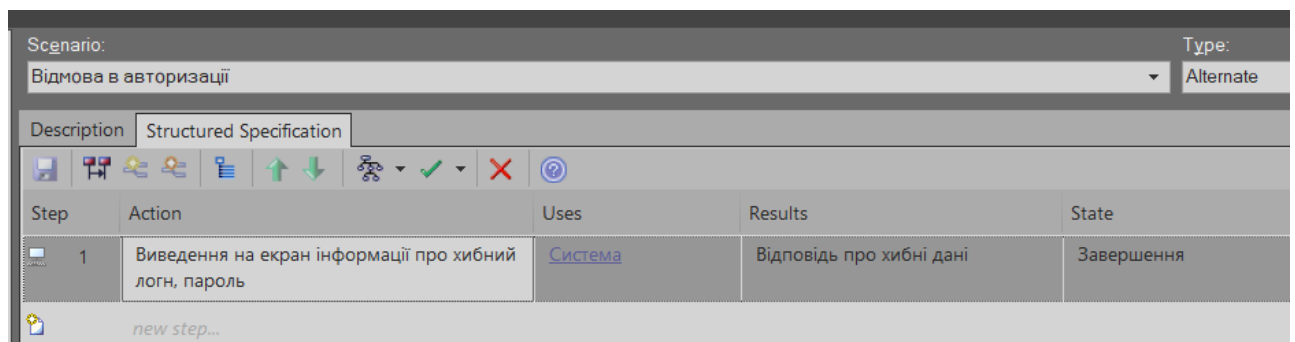


Рисунок 2.10 – Альтернативна дія для «Підтвердження авторизації» у сценарії прецедента «Авторизація»

Джерело: розроблено автором самостійно

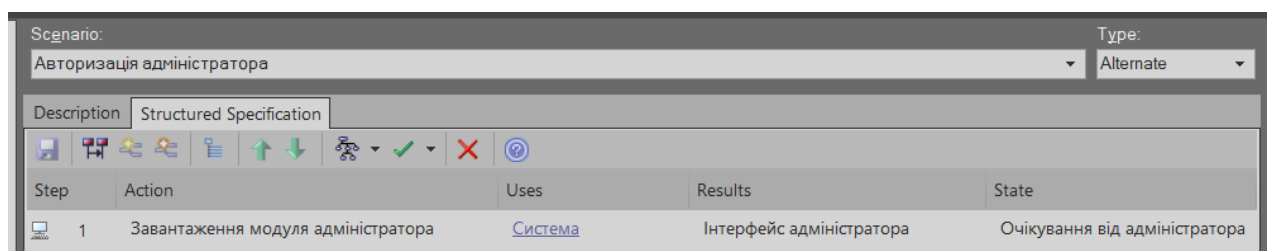


Рисунок 2.11 – Альтернативна дія для «Авторизація адміністратора» у

сценарії прецедента «Авторизація»

Джерело: розроблено автором самостійно

На рис. 2.12 показаний сценарій прецедента, коли система здійснює пошук, враховуючи вказані попередньо права доступу. На рис. 2.13 показано додатковий процес, який виникає у разі виявлення помилок. В такому разі буде створено додатковий файл помилок.

Step	Action	Uses	Results	State
1	Отримання пошукового запиту	СИСТЕМА	Запит для обробки	Передача запиту для обробки
2	Очищення запиту від слів-слів і перетворений для запиту	СИСТЕМА	Оброблений запит для пошуку	Передача запиту для пошуку
3	Передача кеш-файлу до алгоритму пошуку	СИСТЕМА	Переданий кеш-файл	Передача кеш-файлу
4	Пошук за алгоритмом	СИСТЕМА	Результати пошуку	Передача результатів для обробки
5	Формування результату алгоритму	СИСТЕМА	Результати пошуку	Формування кеш-файлів
6	Формування кеш-файлу	СИСТЕМА	Кеш-файл	Підготовка для видачі
7	Формування результатів на екран	СИСТЕМА	Список результатів	Очікування відповіді користувача

Step	Path Name	Type	Join
0	Basic Path	Basic Path	-
5a	При наявності помилок	Exception	6

Рисунок 2.12 – Сценарій прецедента «Здійснює пошук за запитом із врахуванням прав доступу»

Джерело: розроблено автором самостійно

Step	Action	Uses	Results	State
1	Формування кеш-файлу помилок	СИСТЕМА	Кеш-файл	Передача кеш-файлу

Рисунок 2.13 – Додаткова дія у разі наявності помилок

Джерело: розроблено автором самостійно

На рисунку 2.14 показана діаграма діяльності для роботи користувача із системою.

На рисунку 2.15 показана діаграма діяльності для прецедента «Авторизація», а на рисунку 2.16 – діаграма станів.

На рисунку 2.17 – діаграма діяльності для прецедента «Пошук за запитом», а на рис. 2.18 – відповідна діаграма станів.

На рисунку 2.19 показана діаграма послідовності для роботи із системою.

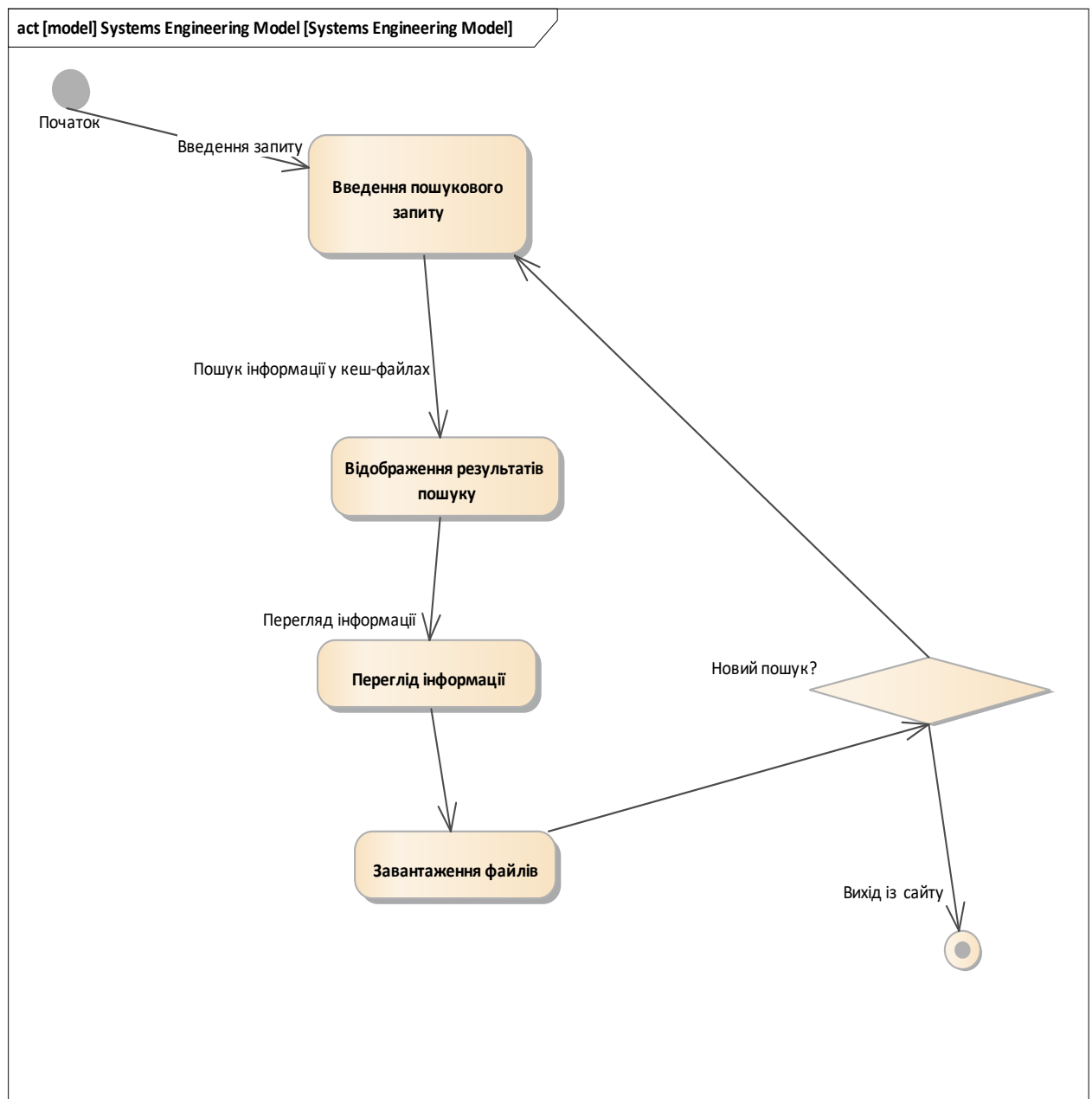


Рисунок 2.14 – Діаграма діяльності роботи користувача із системою

Джерело: розроблено автором самостійно

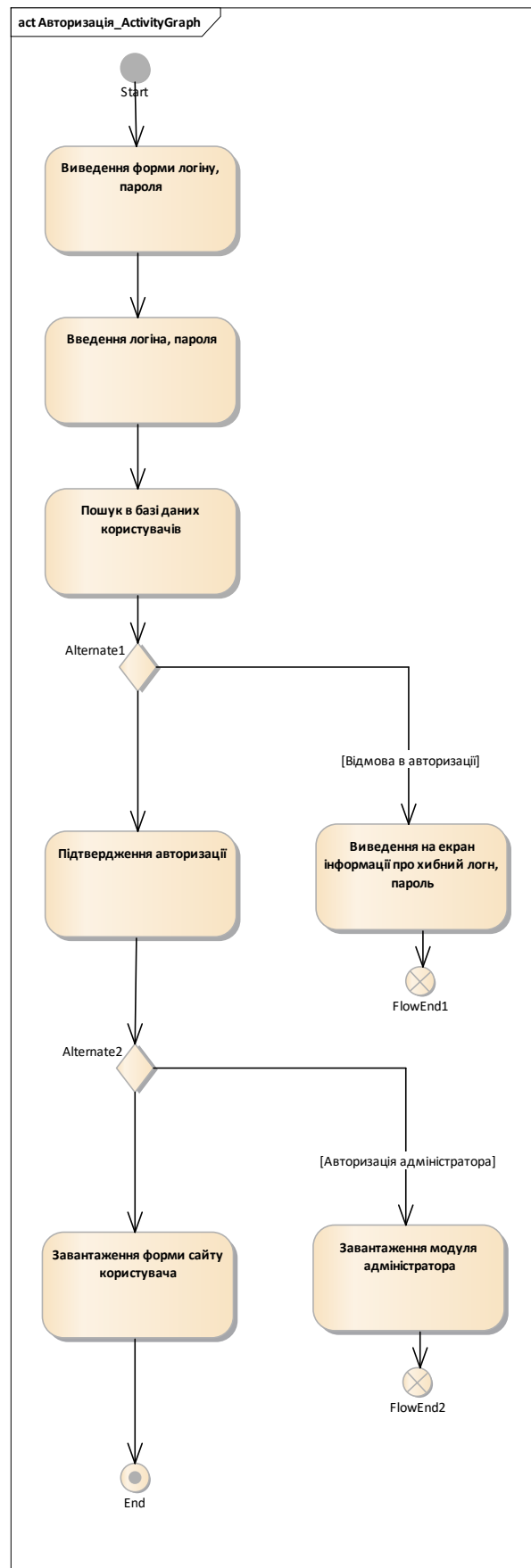


Рисунок 2.15 – Діаграма діяльності для прецедента «Авторизація»

Джерело: розроблено автором самостійно

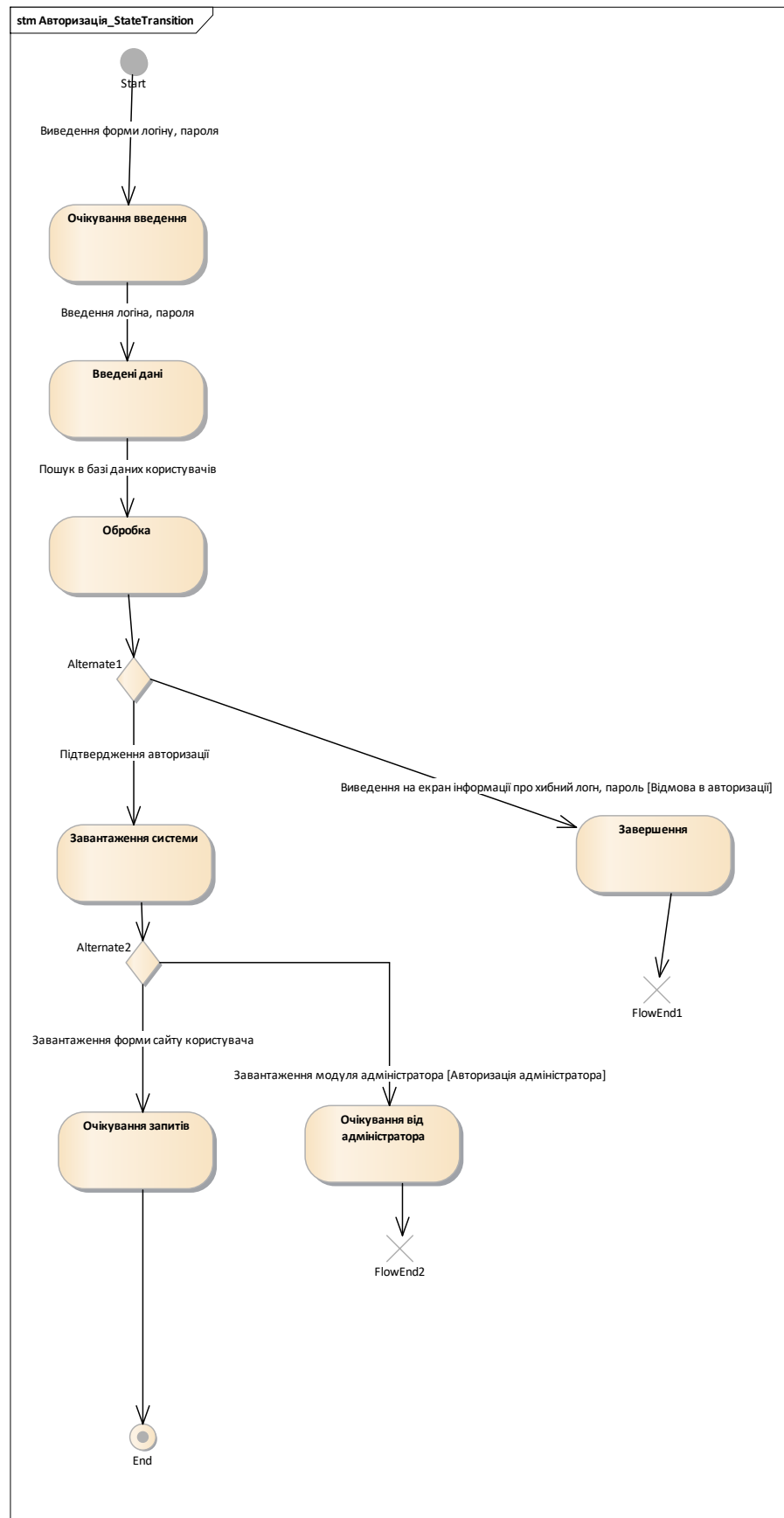


Рисунок 2.16 – Діаграма станів для прецедента «Авторизація»

Джерело: розроблено автором самостійно

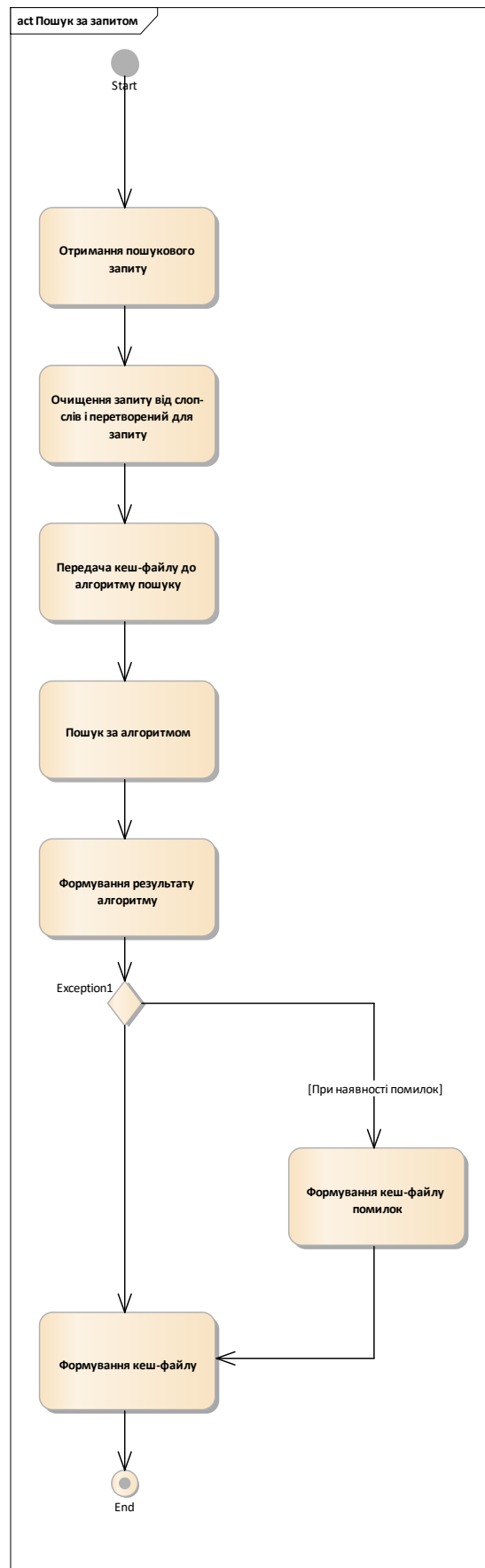


Рисунок 2.17 – Діаграма діяльності для прецедента «Пошук за запитом»

Джерело: розроблено автором самостійно

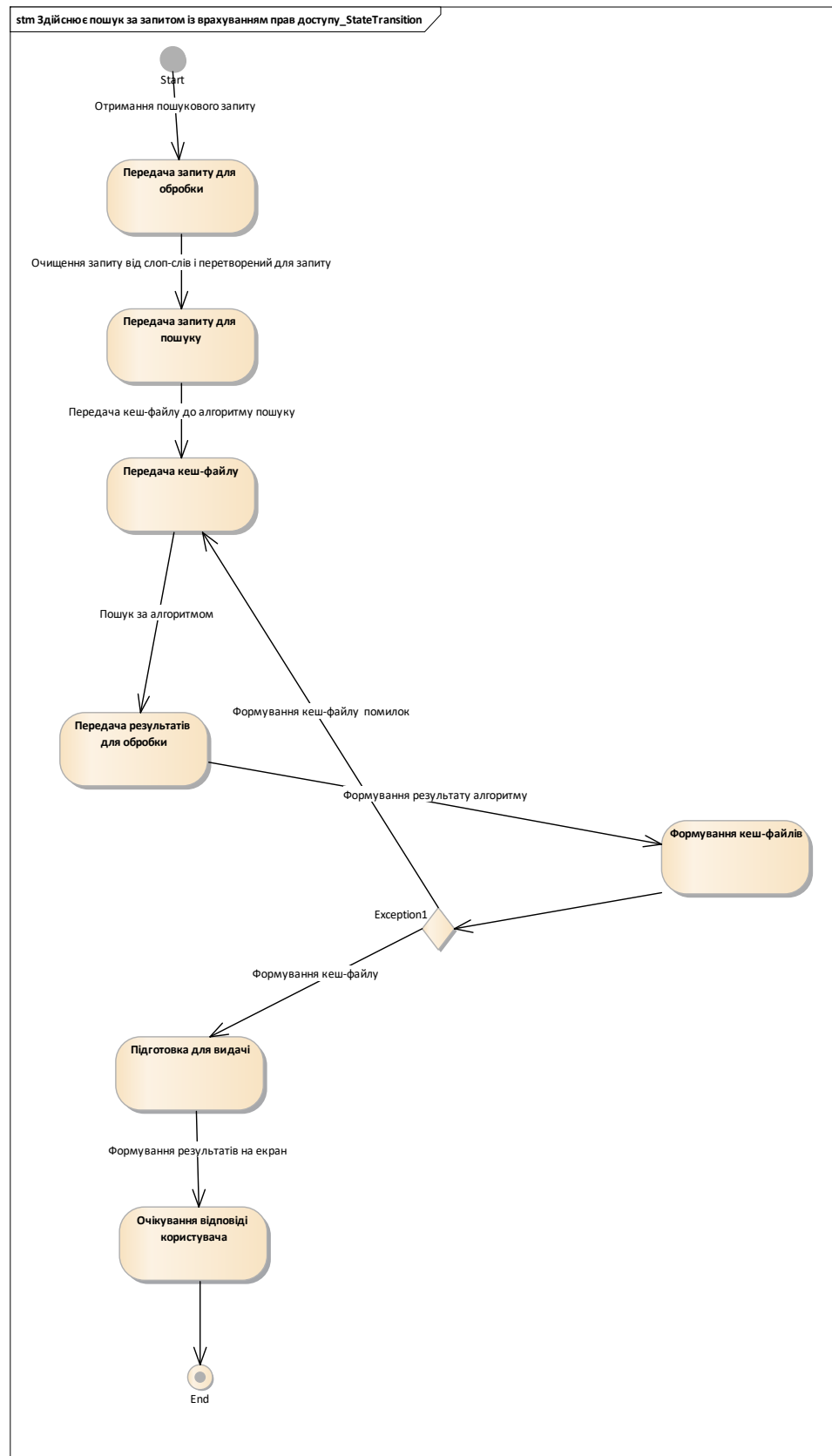


Рисунок 2.18 – Діаграма станів для прецедента «Пошук за запитом»

Джерело: розроблено автором самостійно

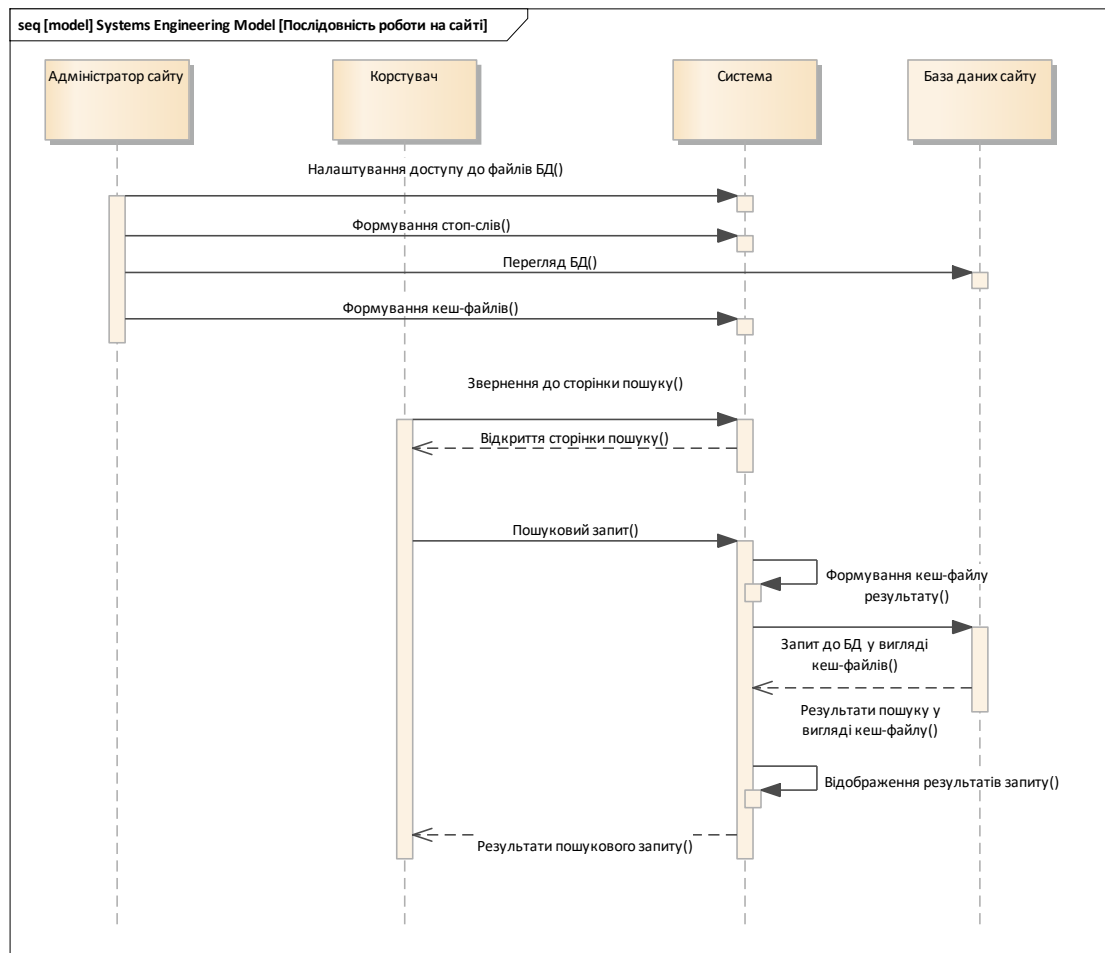


Рисунок 2.19 – Діаграма послідовності роботи у системі.

Джерело: розроблено автором самостійно

2.3.2 Моделювання структури системи

Структуру системи покажемо за допомогою діаграм визначення блоків (рис. 2.20), компонентів системи (рис. 2.21), розгортання (рис. 2.22) та діаграми класів (рис. 2.23, 2.24).

Діаграма визначення блоків (або BDD, за англійською аббревіатурою Block Definition Diagram) - це один із типів діаграм UML, що служить для зображення функціональної структури системи та її взаємодії з зовнішнім середовищем. На діаграмі BDD система розглядається як сукупність блоків, які відображають її функції, компоненти або підсистеми. Була розроблена діаграма структури системи, яка включає підсистеми (рис. 2.20).

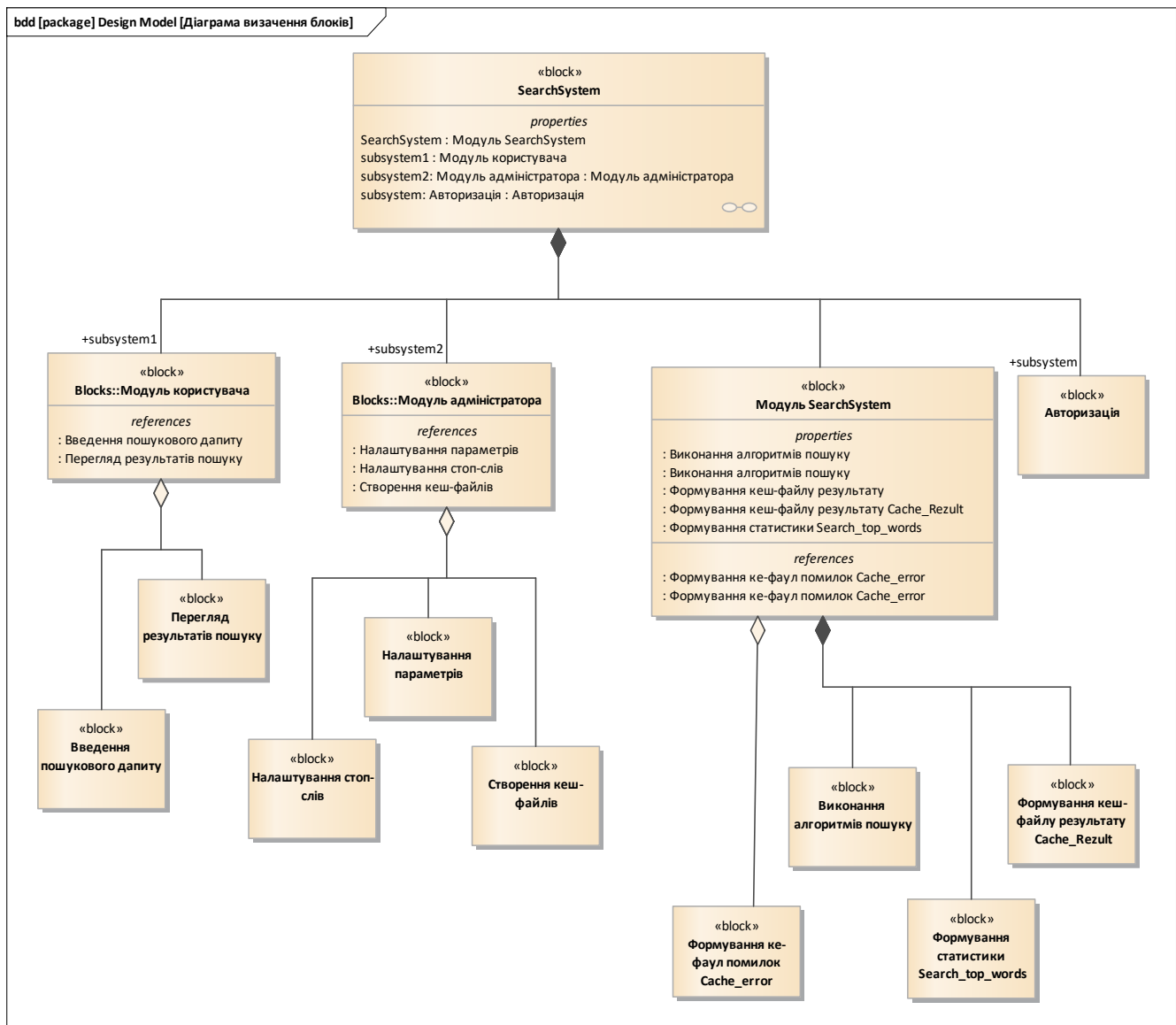


Рисунок 2.20 – Діаграма визначення блоків

Джерело: розроблено автором самостійно

Компонентна діаграма UML допомагає користувачам зрозуміти структуру системи. На рисунку 2.21 показана діаграма компонентів інформаційної системи корпоративного пошуку.

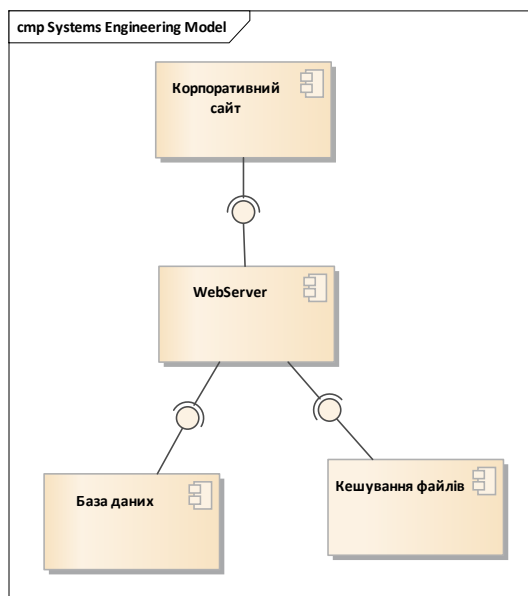


Рисунок 2.21 – Діаграма компонентів

Джерело: розроблено автором самостійно

Діаграма розгортання відтворює візуальне представлення інфраструктури, на яку буде встановлено програмний додаток: структуру системи та розміщення компонентів на її вузлах, а також маршрути передачі даних між цими вузлами. Ця діаграма сприяє ефективнішому організації компонентів, що впливає на продуктивність системи, а також допомагає вирішити додаткові завдання, наприклад, пов'язані з безпекою. У складі діаграми розгортання відображаються графічні представлення процесорів, пристроїв, процесів та зв'язків між ними (рис. 2.22).

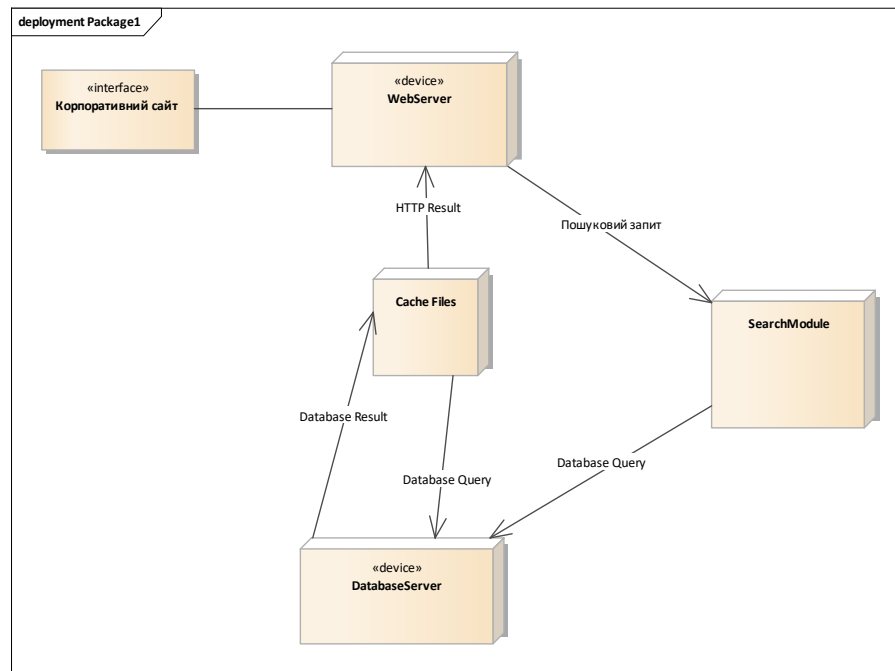


Рисунок 2.22 – Діаграма розгортання

Джерело: розроблено автором самостійно

Діаграма класів – це один із типів діаграм UML, який дозволяє описати класи системи та їхню взаємодію. Розроблена діаграма класів представлена у вигляді класів-сутностей (рис. 2.23) та граничних класів (рис. 2.24).

Пропонуємо наступні класи:

- Request_search – безпосередньо пошуковий запит;
- Search_settings – налаштування системи;
- Stop_words – перелік стоп-слів, які виключаємо під час виконання алгоритму пошуку;
- Cache_Rezult – результат запиту, включає адреси, де знаходяться файли та сторінки),
- Cache_error – створюється кеш-файл у разі виникнення помилок
- Cache_files – клас, який створює кеш-файли
- Search_top_words – список найбільш поширених пошукових слів.

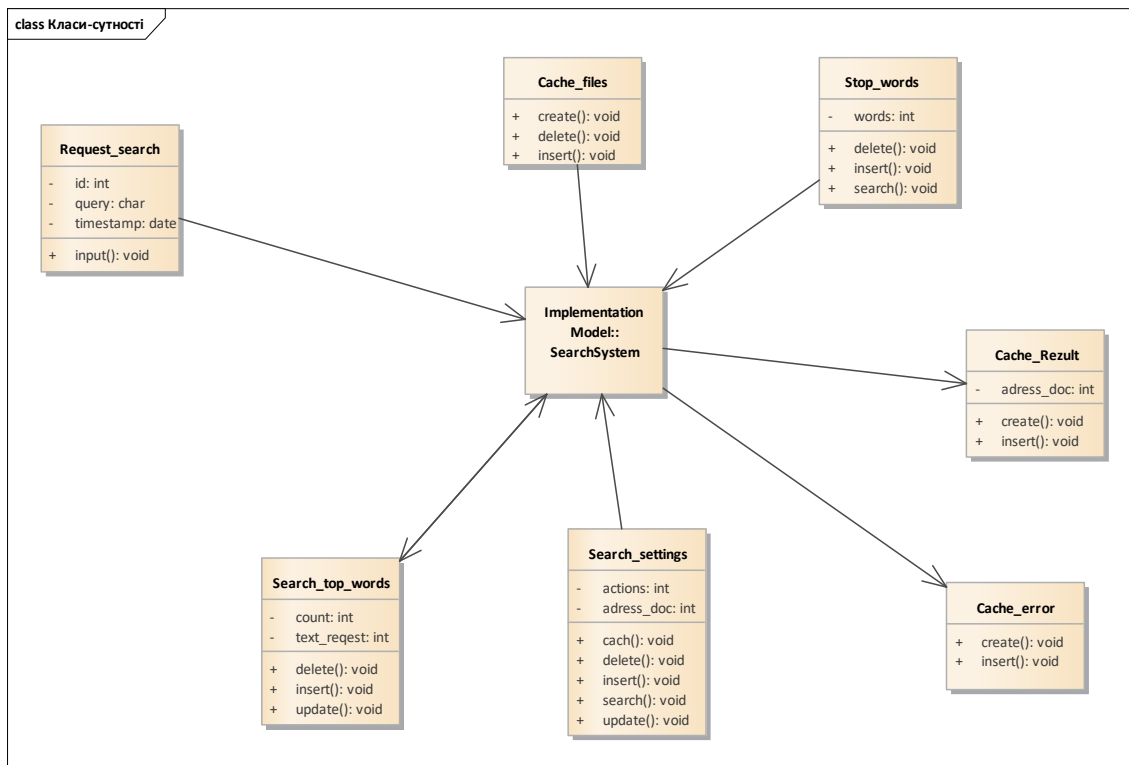


Рисунок 2.23 – Діаграма класів-сутностей

Джерело: розроблено автором самостійно

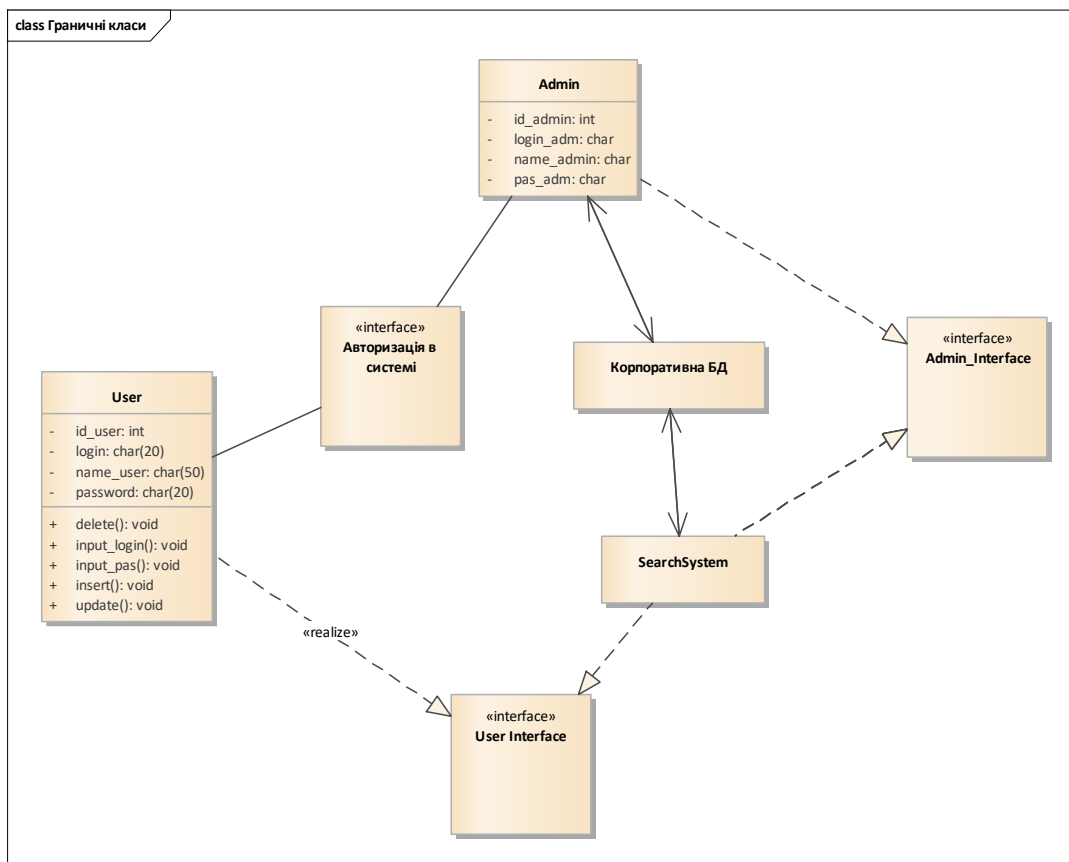


Рисунок 2.24 – Діаграма граничних класів

Джерело: розроблено автором самостійно

2.3.3 Розподіл вимог за компонентами системи.

Для розроблюваної системи було розроблено тест-кейси, у яких передбачено можливі помилки та дії користувача.

На рисунках 2.25-2.26 показані тест-кейси і тестові сценарії варіанту використання «Авторизація».

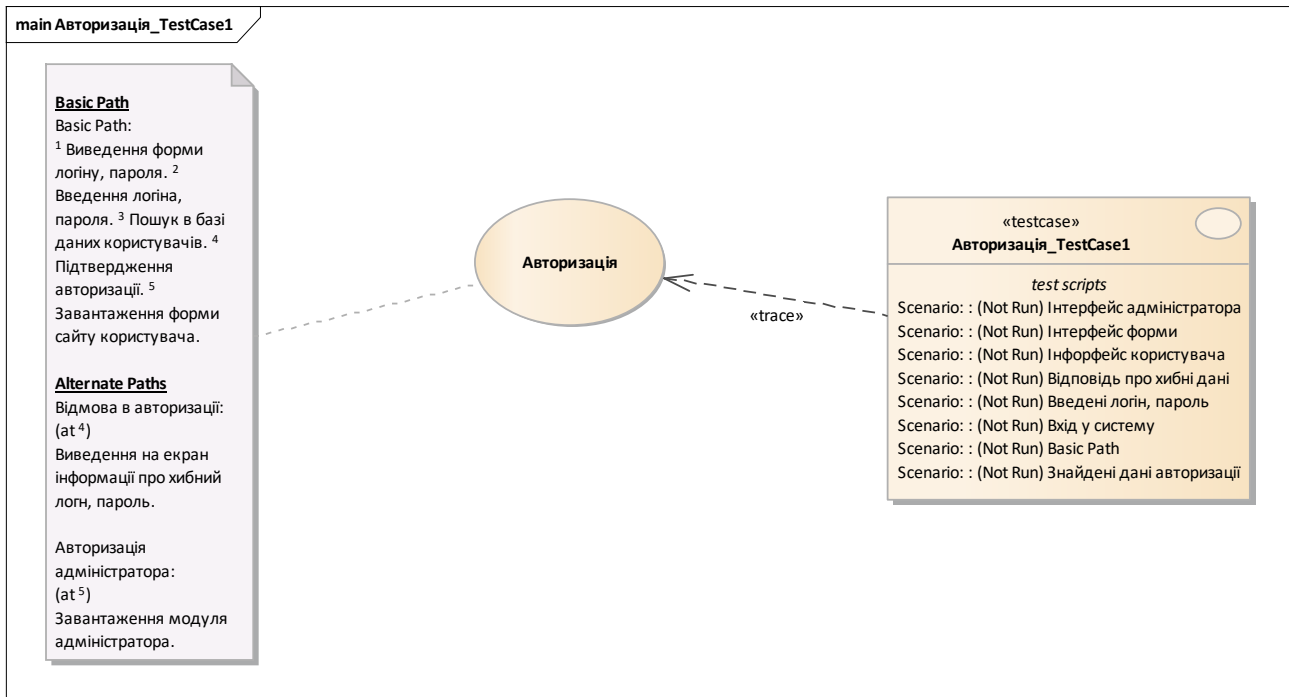


Рисунок 2.25 – Тестові сценарії для дії «Авторизація»

Джерело: розроблено автором самостійно

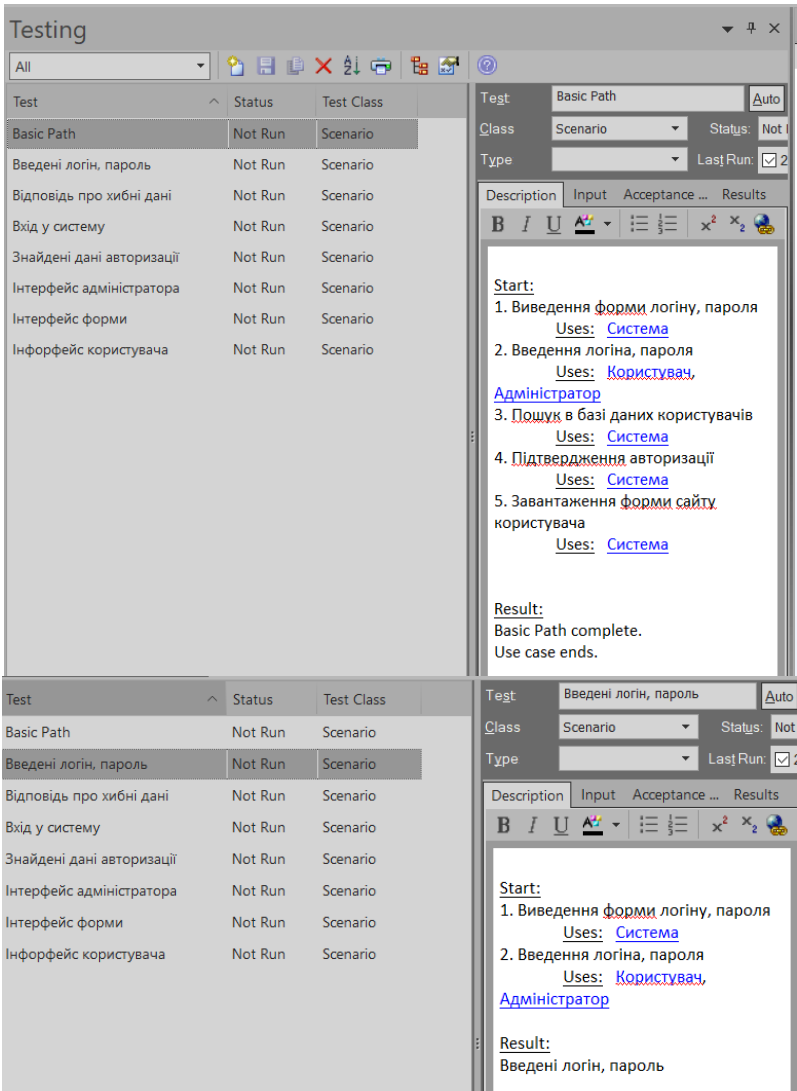


Рисунок 2.26 – Тест-кейс «Авторизація»

Джерело: розроблено автором самостійно

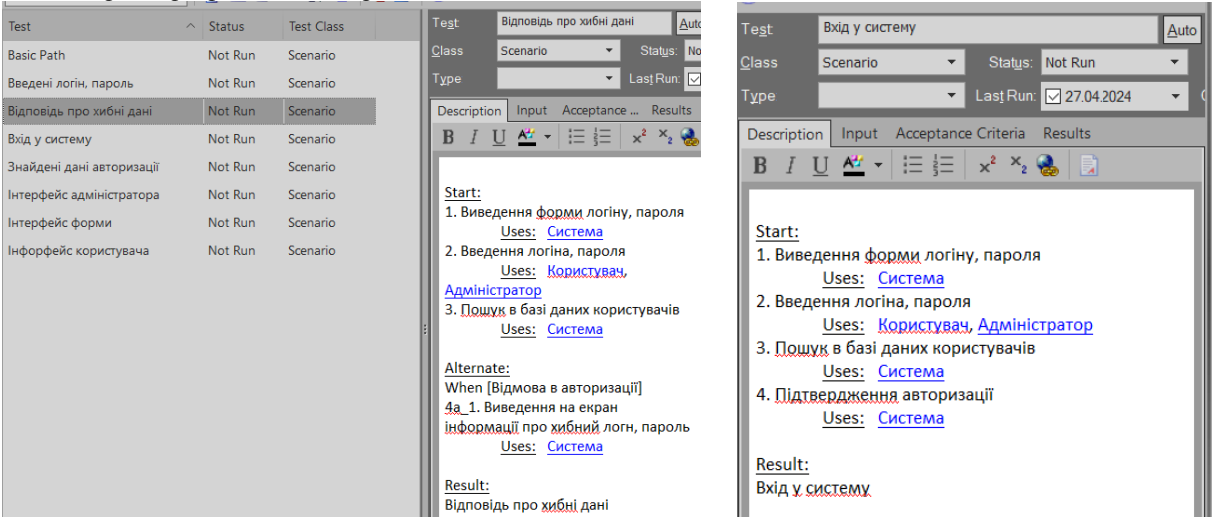


Рисунок 2.26 – Тестовий сценарій «Авторизація» (продовження)

Джерело: розроблено автором самостійно

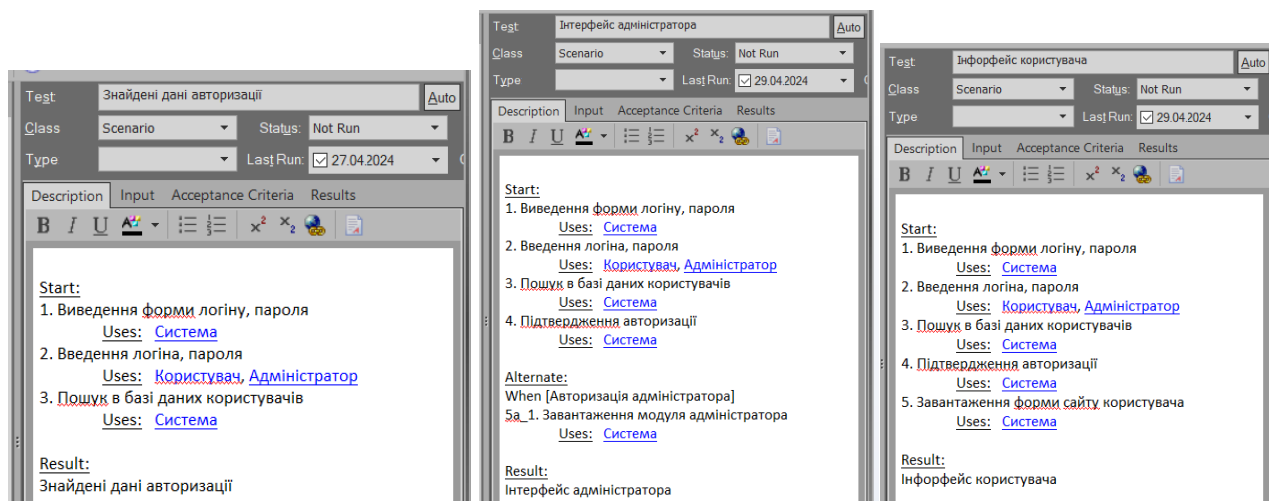


Рисунок 2.26 – Тестові сценарії «Авторизація» (закінчення)

Джерело: розроблено автором самостійно

На рисунках 2.27-2.31 показані тест-кейси і тестові сценарії варіанту використання «Авторизація».

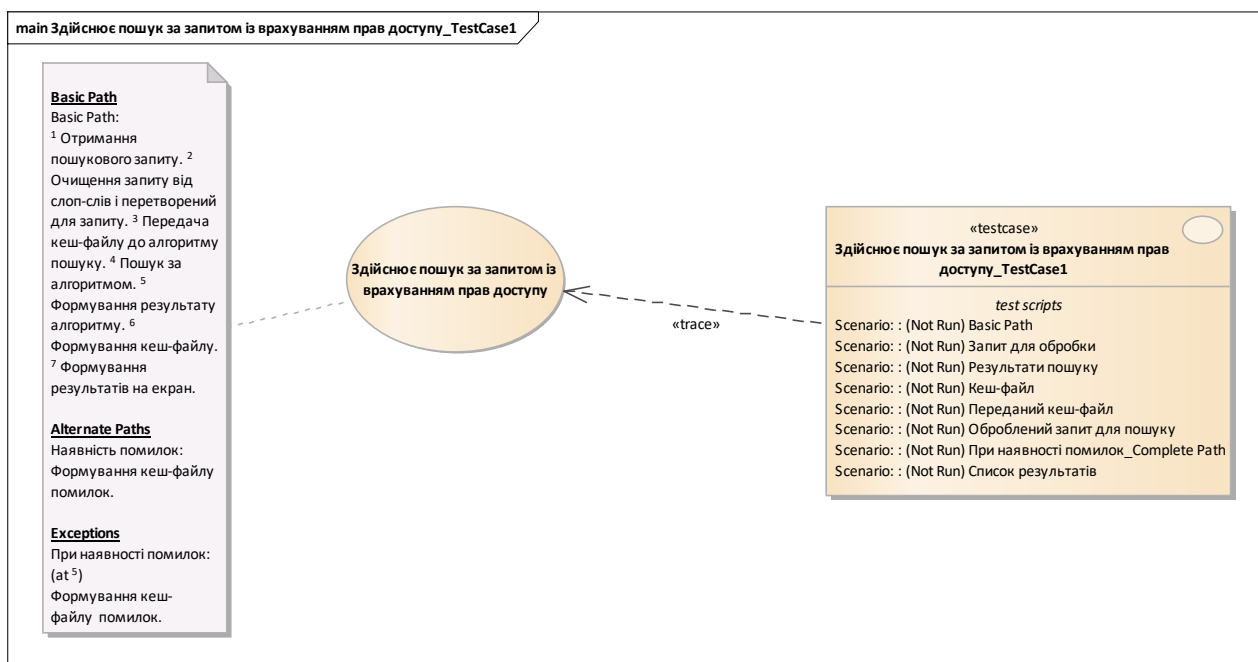


Рисунок 2.27 – Тест-кейс для прецеденту «Пошук»

Джерело: розроблено автором самостійно

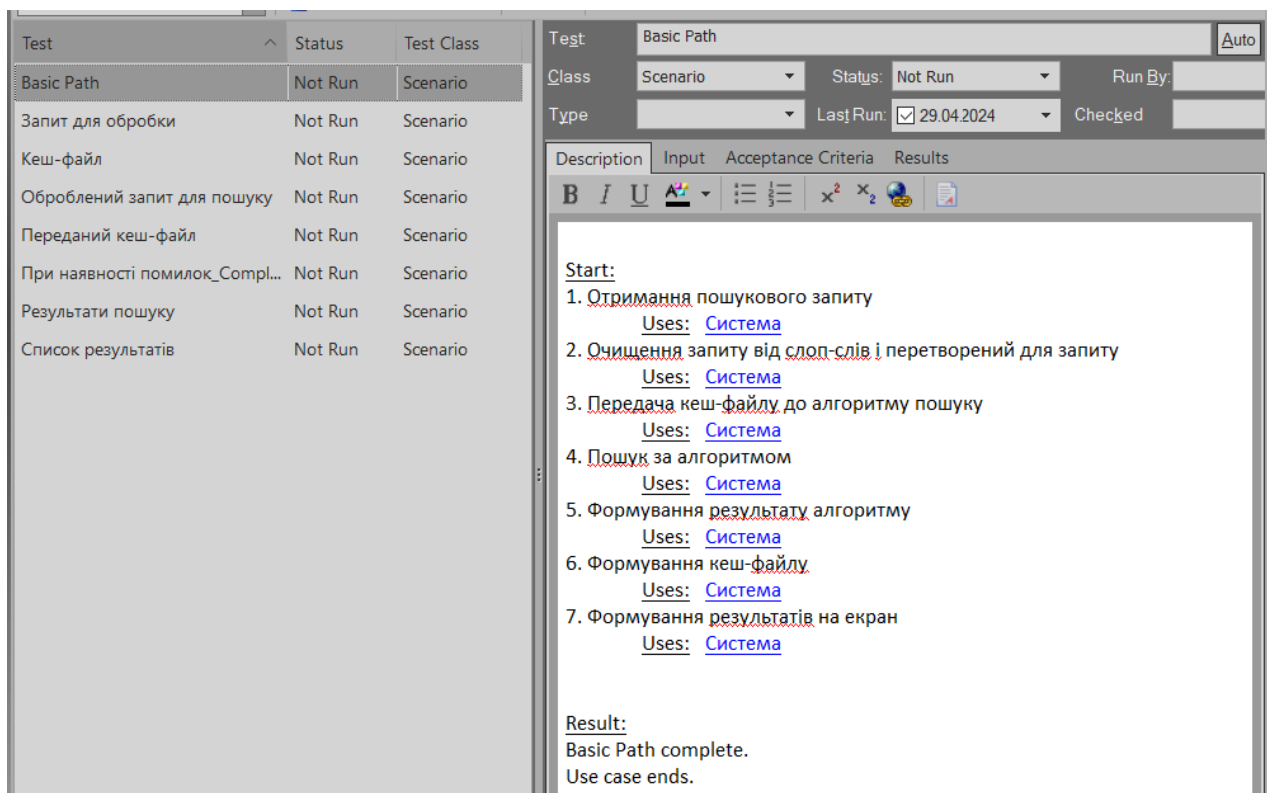


Рисунок 2.28 – Тестовий сценарій «Пошук»

Джерело: розроблено автором самостійно

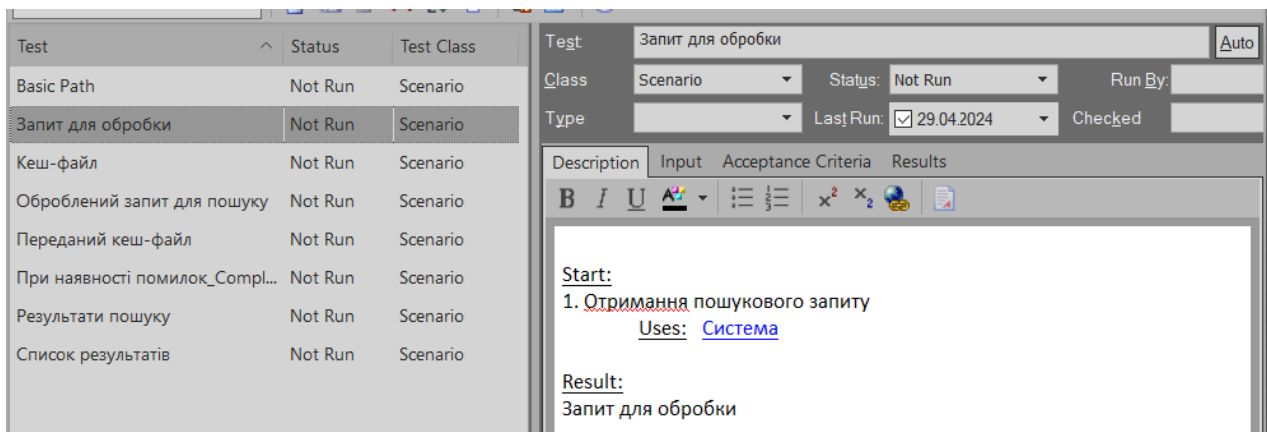


Рисунок 2.29 – Тестовий сценарій «Запит для обробки»

Джерело: розроблено автором самостійно

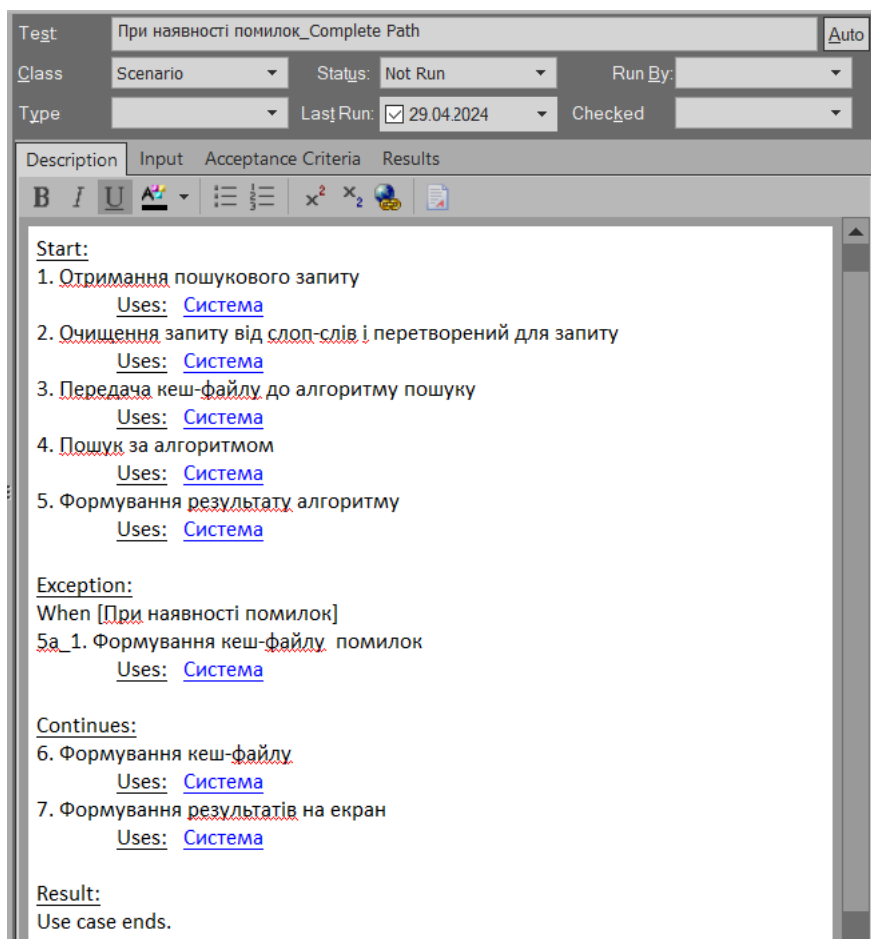


Рисунок 2.30 – Тестовий сценарій «Введення хибної інформації»

Джерело: розроблено автором самостійно

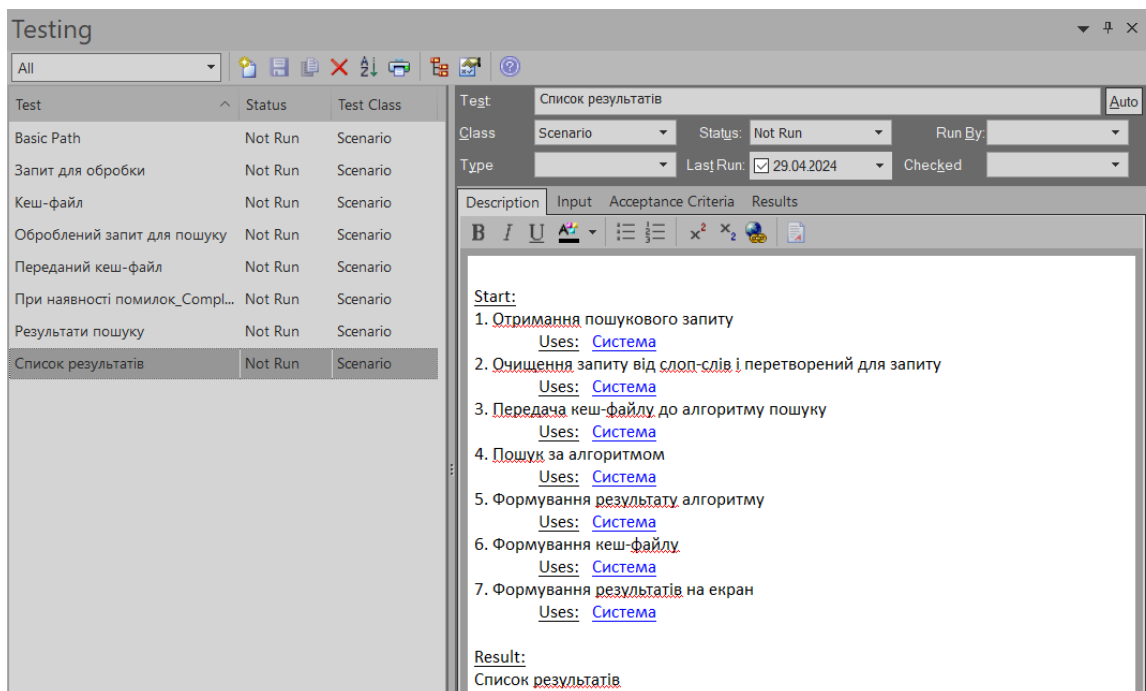


Рисунок 2.31 – Тестовий сценарій «Список результатів»

Джерело: розроблено автором самостійно

Діаграма трасування дозволяє відобразити вимоги, прецеденти, варіанти тестування та елементи користувацького інтерфейсу, дозволяючи відобразити зв'язок між ними.

На рисунку 2.32 показана діаграма трасування вимог для системи.

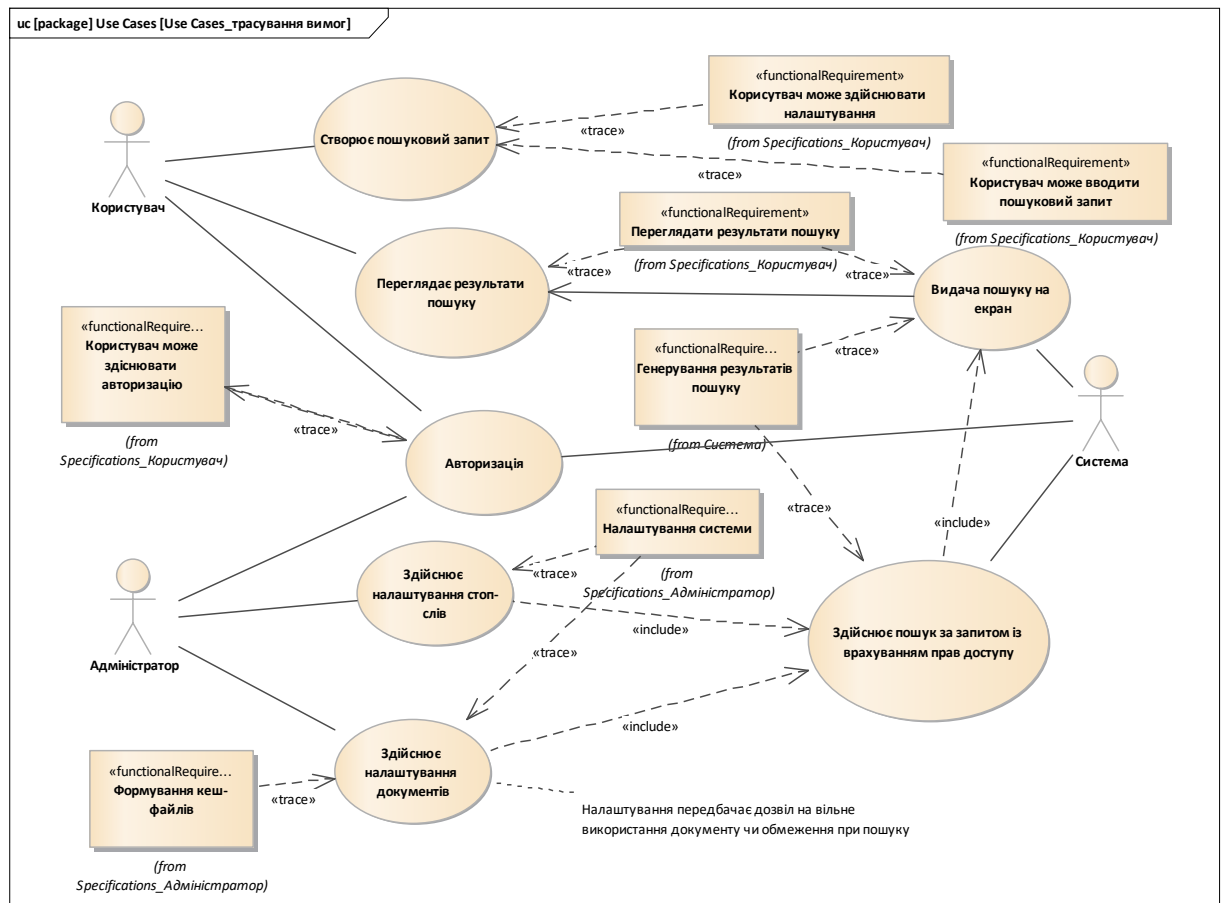


Рисунок 2.32 – Діаграма трасування вимог

Джерело: розроблено автором самостійно

Матриця взаємозв'язків вимог і прецедентів, яка показує, що вимоги системи відповідають варіантам використання, показана на рис. 2.33.

РОЗДІЛ 3

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

Інформаційне забезпечення системи пошуку на корпоративному сайті включає в себе методичні та інструктивні матеріали, позамашинну інформаційну базу та внутрішньомашинну інформаційну базу. Схема інформаційного забезпечення показана на рисунку 3.1

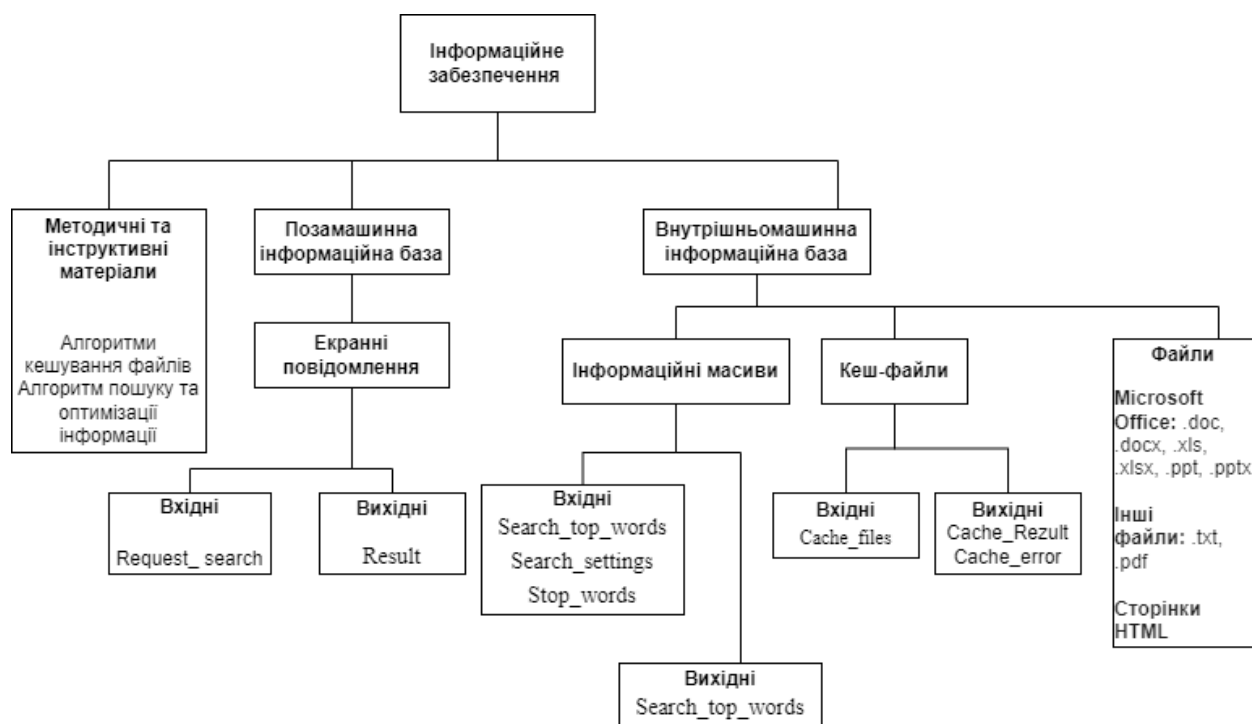


Рисунок 3.1 – Схема інформаційного забезпечення

Джерело: розроблено автором самостійно

До складу **методичних та інструктивних матеріалів** відносять Алгоритми кешування файлів та Алгоритм пошуку та оптимізації інформації. Методичні та інструктивні матеріали відіграють важливу роль у визначенні та підтримці оптимальних стратегій роботи системи.

Алгоритми кешування файлів допомагають оптимізувати швидкість пошуку, зберігаючи копії часто використовуваних документів або сторінок в

кеш-файлах. Це дозволяє зменшити навантаження на базу даних та прискорити відгук системи на запити користувачів. Методичні матеріали з алгоритмів кешування надають рекомендації щодо вибору оптимальних стратегій кешування, враховуючи специфіку сайту та потреби користувачів.

Алгоритм пошуку та оптимізації інформації визначає, як система обробляє запити користувачів, вибирає та ранжує результати пошуку. Ці алгоритми базуються на ряді факторів, таких як релевантність документів, популярність запитів, частота оновлення контенту тощо. Методичні та інструктивні матеріали надають рекомендації щодо використання ефективних алгоритмів пошуку, оптимізації та адаптації їх до конкретних потреб корпоративного сайту.

Позамашинна інформація включає в себе екранні повідомлення. Зокрема, вхідним повідомленням є екранна форма з запитом для пошуку інформації (Request_search), а вихідним – відображений на сайті результат пошуку (Result).

Внутрішньомашинна інформаційна база включає в себе:

- масиви бази даних;
- кеш-файли;
- файли і сторінки HTML, що містять інформацію для пошуку.

Масиви БД поділяються на вхідні та вихідні.

Вхідні інформаційні масиви зберігають наступну інформацію: список неважливих слів (Stop_words), налаштуваннях доступу до документів (search_settings) та статистику запитів (Search_top_words). Ця інформація використовується системою для обробки та аналізу запитів користувачів, швидкого доступу до даних та покращення якості пошуку.

Вихідним інформаційним масивом є файл статистики запитів (Search_top_words).

В системі пошуку на корпоративному сайті для швидкого пошуку інформації файли зберігаються у вигляді кеш-файлів. Вони допомагають

зменшити навантаження на сервер, прискорюючи відгук системи на запити користувачів.

Дані документів, по яким здійснюється пошук, зберігаються у вхідних кеш-файлах (Cache_files). Вони дозволяють прискорити обробку запитів користувачів, зберігаючи структуровані дані в пам'яті системи.

Вихідні кеш-файли (Cache_Rezult) містять кешовані результати пошуку, які були згенеровані після обробки вхідних запитів користувачів. Ці файли дозволяють швидко надавати користувачам раніше згенеровані результати без повторної обробки запиту, що допомагає знизити навантаження на систему та покращити швидкість відгуку.

Кеш-файли помилок (Cache_error) зберігають інформацію про помилки, які виникають під час обробки запитів. Це дозволяє системі швидко виявляти та вирішувати проблеми без повторної обробки запиту та забезпечує кращу стабільність та надійність роботи системи.

Загальна концепція кешування в системі пошуку базується на зберіганні та використанні заздалегідь підготовлених даних для оптимізації роботи системи та підвищення задоволення користувачів.

Корпоративна інформація зберігається у файлах і на сторінках сайту, і є основним об'єктом пошуку інформації. Це наступні документи:

- файли Microsoft Office (.doc, .docx, .xls, .xlsx, .ppt, .pptx) — це документи, таблиці та презентації, які часто використовуються для зберігання корпоративної інформації;
- інші текстові файли (.txt, .pdf) також важливі для системи пошуку. Текстові файли (.txt) зазвичай легко індексуються, оскільки вони не містять складних форматів. Файли PDF (.pdf), як правило, потребують спеціалізованих інструментів для індексації, оскільки вони можуть включати графіку, таблиці та інші об'єкти;
- HTML-сторінки — основний формат веб-сторінок, що використовується в Інтернеті. Індексція HTML-коду дозволяє системі пошуку

ефективно знаходити та відображати результати пошуку користувачам у зручному для них форматі.

Проектування форм первинних документів, машинограм та відеокадрів.

Для моделювання користувацького інтерфейсу запропонована ієрархія екранних форм (рис. 3.2).

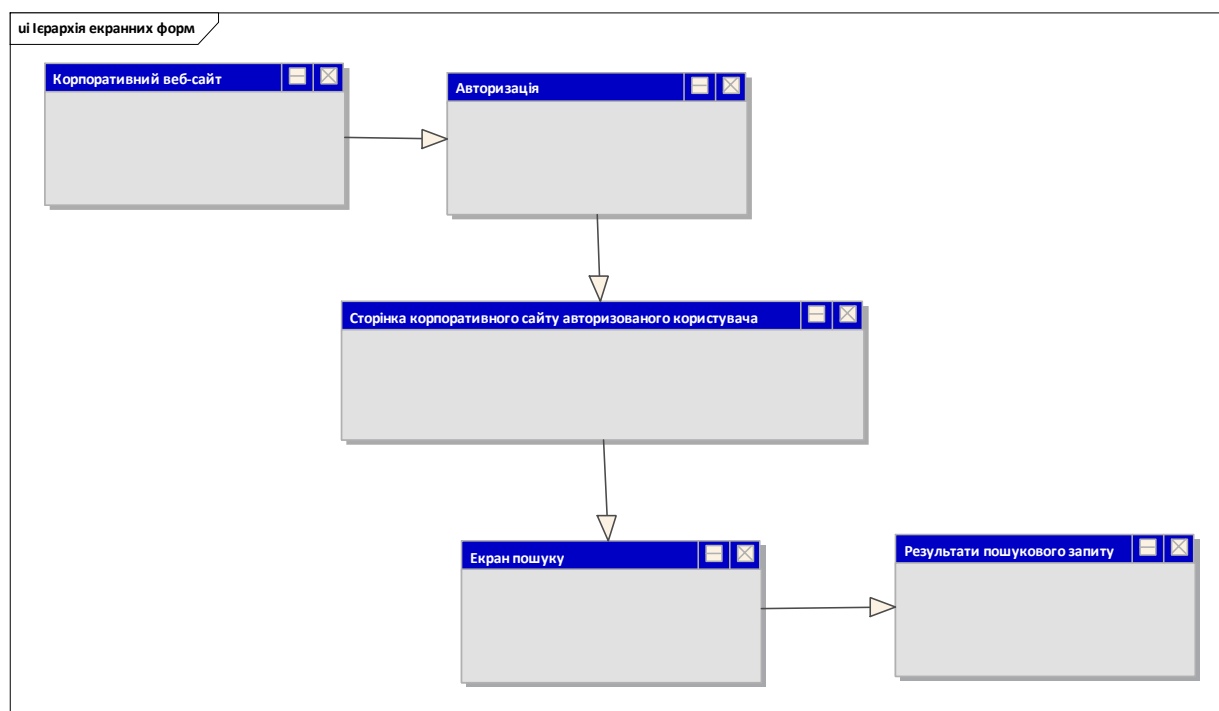


Рисунок 3.2 – Ієрархія екранних форм

Джерело: розроблено автором самостійно

Складові інтерфейсу системи корпоративного пошуку включають наступні елементи:

- пошукове поле: елемент введення, де користувач може ввести свій пошуковий запит;
- кнопка "Пошук": елемент керування, який використовується для запуску пошуку після введення пошукового запиту;
- фільтри пошуку: елементи керування, які дозволяють користувачеві фільтрувати результати пошуку за різними параметрами, такими як тип документа, дата, автор тощо;

- список результатів пошуку: відображення знайдених документів або ресурсів, які відповідають пошуковому запиту користувача;
- детальна інформація про результати: можливість перегляду детальної інформації про кожен знайдений документ, таку як заголовок, опис, автор, дата тощо;
- навігаційні елементи: елементи керування, які дозволяють користувачеві переходити між різними сторінками результатів пошуку або різними розділами системи;
- повідомлення про помилки або немає результатів: відображення повідомлень користувачеві в разі, якщо виникла помилка під час пошуку або якщо результатів не знайдено;
- елементи керування вибором мови або налаштувань: можливість користувача вибирати мову інтерфейсу або налаштовувати інші параметри системи.

Прототипи інтерфейсних форм показані на рис. 3.13-3.14.

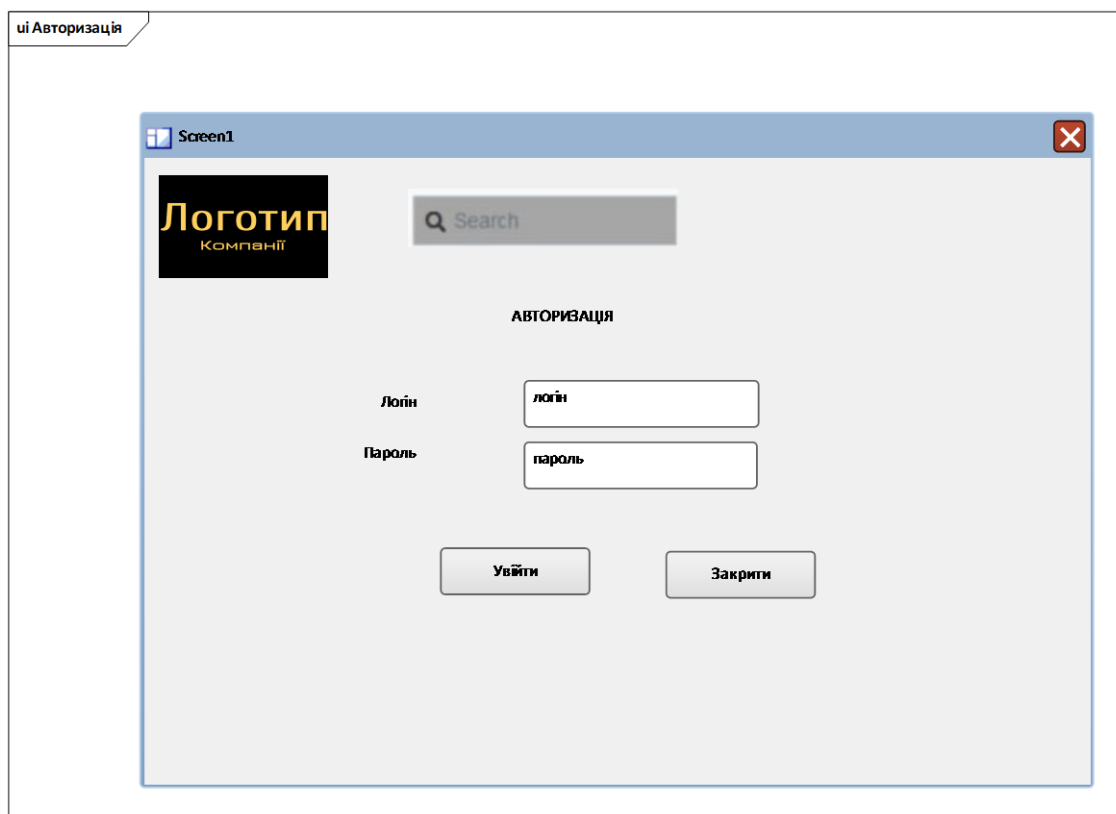


Рисунок 3.3 – Екранна форма «Авторизація»

Джерело: розроблено автором самостійно

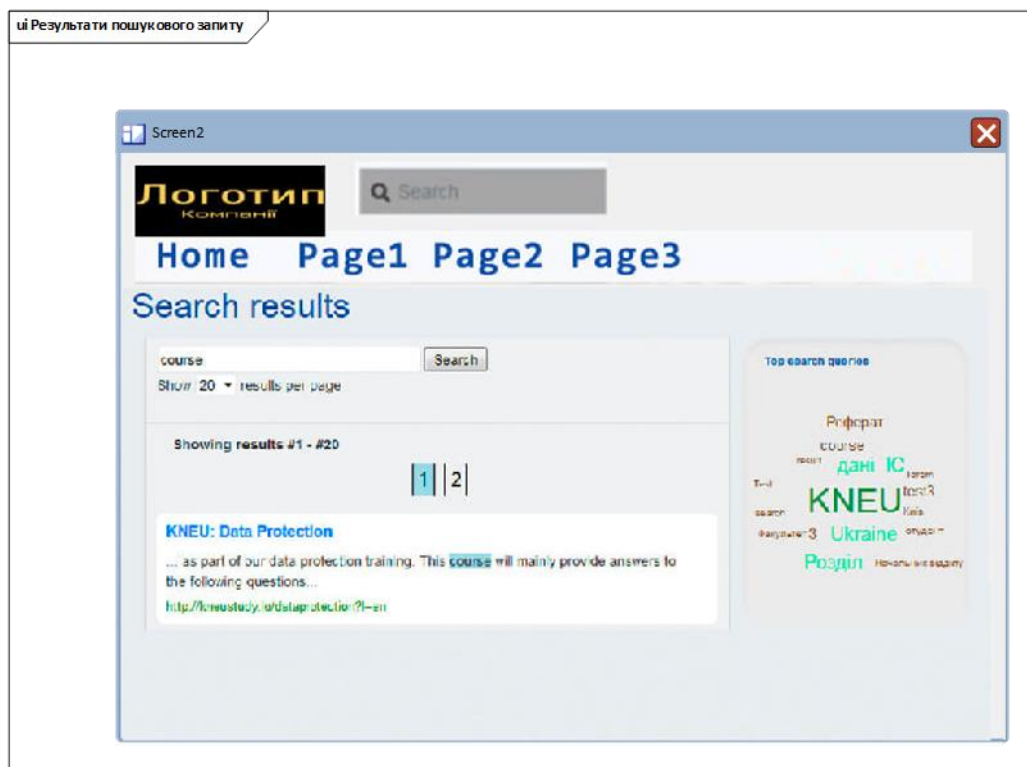


Рисунок 3.4 – Екранна форма «Результати пошуку»

Джерело: розроблено автором самостійно

3.2 Технічне забезпечення

Загальні положення та схема автоматизації.

Для функціонування системи пошуку на корпоративному сайті необхідним є наявність веб-сервера та веб-додатку, на якому і знаходиться система пошуку.

Пошук у мережі здійснюється за умови авторизації користувача.

В межах компанії організована внутрішня мережа Ethernet, яка поєднує між собою комп'ютери та інші пристрої компанії.

Структура комплексу технічних засобів.

Для роботи з системою необхідно наявність комп'ютера будь-якої конфігурації чи іншого пристрою (смартфона, планшета), під'єданого до мережі інтернет.

Схема мережі передачі даних.

Схема мережі передачі даних показана на рис. 3.5.

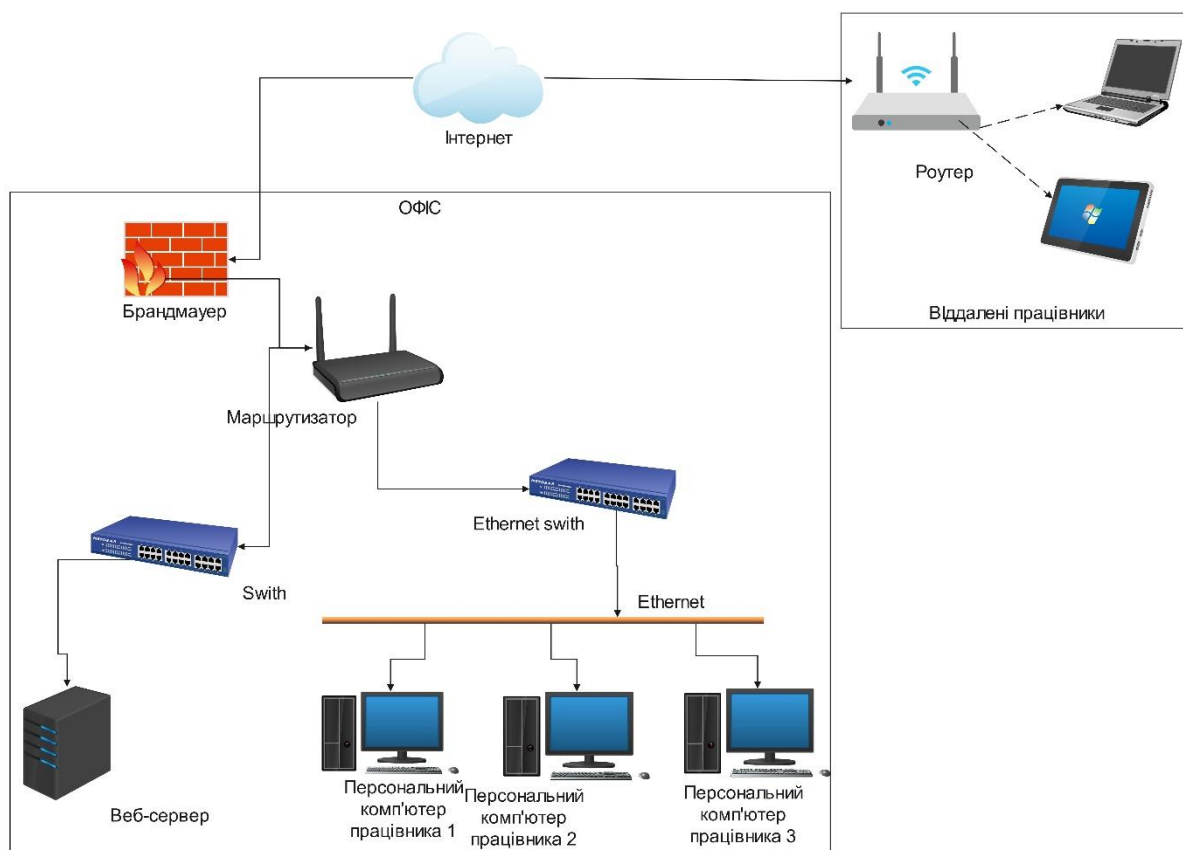


Рисунок 3.4 – Схема мережі передачі даних

Джерело: розроблено автором самостійно

Інфраструктура корпоративної мережі, представлена на рисунку 3.4, забезпечує надійну і ефективну роботу системи пошуку на корпоративному сайті шляхом з'єднання різних компонентів.

Мережа під'єднана до Інтернету, який є джерелом зовнішнього доступу до ресурсів компанії. Інтернет-з'єднання проходить через брандмауер, який виступає першою лінією захисту мережі, контролюючи вхідний та вихідний трафік та запобігаючи несанкціонованому доступу.

З брандмауера трафік надходить до маршрутизатора. Маршрутизатор керує потоками даних між Інтернетом і внутрішньою мережею компанії, розподіляючи їх до відповідних сегментів мережі. Від маршрутизатора трафік надходить до двох комутаторів (Ethernet switch). Один комутатор підключений до маршрутизатора, а інший забезпечує зв'язок з основним сегментом Ethernet внутрішньої мережі.

Комутатори об'єднують різні пристрої в мережі, дозволяючи їм ефективно обмінюватися даними. Вони забезпечують високошвидкісний обмін даними між усіма підключеними пристроями, такими як веб-сервер і персональні комп'ютери працівників.

Веб-сервер підключений до комутатора і є основним сервером, на якому розміщений корпоративний сайт та система пошуку. Він обробляє запити від користувачів як з внутрішньої мережі, так і з Інтернету, забезпечуючи доступ до веб-сторінок і баз даних.

Персональні комп'ютери працівників також підключені до мережі через Ethernet-комутатор. Це дозволяє працівникам доступ до корпоративного сайту, системи пошуку та інших внутрішніх ресурсів. Таким чином, комп'ютери працівників мають надійне і швидке з'єднання для виконання своїх завдань.

У разі віддаленої роботи, працівники можуть увійти на корпоративний сайт через мережу інтернет. Це дає працівникам гнучкість у використанні різних пристроїв для роботи, забезпечуючи зручність і мобільність [19].

3.3 Програмне забезпечення

Система надає клієнту та адміністратору доступ на веб-ресурс через браузер, але клієнт має свій функціонал, а адміністратор розширений. Клієнт має доступ до певних даних ІС, але це обмежено рівнем доступу до системи. Адміністрування та безпека відповідають за керування користувацькими даними та забезпеченням базового захисту даних бази даних. Опис архітектури проекту в рамках системи зосереджується на взаємозв'язках між основними компонентами, що спільно формують архітектуру системи, показану на рис. 3.5.

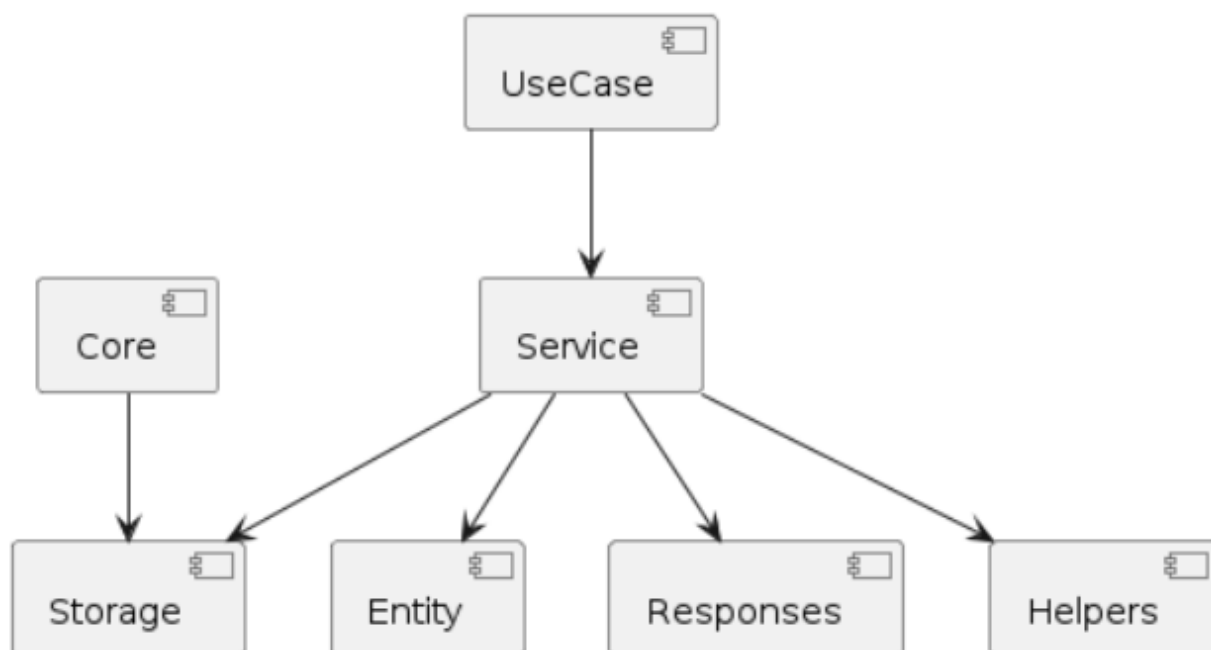


Рисунок 3.5 – Архітектура системи пошуку

Джерело: розроблено автором самостійно

Опишемо основні компоненти системи.

Core – основний компонент, який формує ядро системи. Він включає класи для взаємодії з базою даних та інші ключові елементи, без яких функціонування системи стає неможливим.

UseCase служить для моделювання сценаріїв використання, в яких реалізована бізнес-логіка. Він виступає в ролі джерела розпоряджень для компонентів типу Service, ініціюючи їх дії та взаємодію.

Компоненти типу Service функціонують як виконавчі механізми, які забезпечують реалізацію сценаріїв, визначених в UseCase. Вони взаємодіють зі сховищем для отримання або зберігання даних.

Компоненти Storage є базовими компонентами, які відповідають за зберігання даних. Сховища не здатні обробляти логіку системи; вони просто зберігають або віддають дані у відповідному форматі.

Компоненти Entity слугують класами-обгортками для різних сутностей системи. Вони допомагають у стандартизації об'єктів, що використовуються в системі, роблячи їх структурно схожими.

Компоненти **Responses** спеціалізуються на формуванні відповідей на запити, які надходять ззовні на **UseCase**, забезпечуючи їх стандартизацію.

Helpers – компоненти, які представляють собою бібліотеки, що містять додаткові інструменти або функції, які можуть бути використані в інших частинах системи.

Основні компоненти типу **UseCase** і **Service** взаємодіють між собою із залученням **Storage** для зберігання та отримання даних. Компоненти **Entity** та **Responses** спрощують структуру даних, забезпечуючи стандартизацію об'єктів та відповідей. **Helpers** служать як додаткові інструменти для підтримки основних функцій системи.

Для розроблення системи було обрано мову програмування **PHP**, середовище розробки – **PHPStorm**.

Для реалізації системи були використані наступні бібліотеки:

1. **PDO** - надає єдиний інтерфейс для роботи з базами даних, забезпечуючи підтримку для багатьох СУБД, таких як **MySQL**, **PostgreSQL**, **SQLite** тощо [20];
2. **Igbinary** - альтернатива стандартному серіалізатору **PHP**, що забезпечує більш компактне зберігання серіалізованих даних, зменшуючи розмір серіалізованих об'єктів [21];
3. **cURL** - дозволяє здійснювати **HTTP**-запити з **PHP**, підтримуючи різні протоколи, такі як **HTTP**, **HTTPS**, **FTP** тощо, що робить його ідеальним для взаємодії з веб-сервісами [22];
4. **libxml** - базова бібліотека для роботи з **XML** у **PHP**, надаючи інтерфейси для обробки та аналізу **XML**-документів [23];
5. **DOM** - забезпечує об'єктно-орієнтований інтерфейс для роботи з **XML**-документами, дозволяючи створювати, маніпулювати та змінювати **XML**-структури [24] ;
6. **iconv** - забезпечує функціонал для конвертації наборів символів, дозволяючи змінювати кодування тексту, що корисно при роботі з багатомовними додатками [25];

7. **smalot pdfparser** - бібліотека для аналізу та витягання тексту з PDF-документів [26] ;

- **symfony polyfill-mbstring** - забезпечує функції багатобайтових рядків для середовищ, де ці функції відсутні [27] ;
- **Zlib** - використовується для стиснення та розпаковування даних, що особливо корисно при роботі з архівами та потоками даних [28];

8. **phpoffice phpword** - дозволяє створювати, редагувати та зберігати документи Microsoft Word [29] ;

- **json** - підтримка JSON (JavaScript Object Notation) для кодування та декодування даних у форматі JSON [30] ;

- **xml** - базова підтримка роботи з XML-документами [31];

- **laminas laminas-escaper** - надає функції для безпечного екранування даних для використання в HTML, XML, JSON тощо [32] ;

- **ctype** - надає функції для перевірки символів на основі їх типів, таких як алфавітні, числові тощо [33] ;

- **mbstring** - забезпечує функції для роботи з багатобайтовими рядками, необхідними для обробки тексту у різних кодуваннях [34];

9. **phpoffice phpspreadsheet** - дозволяє створювати, редагувати та зберігати електронні таблиці у різних форматах [35] ;

- **fileinfo** - надає функції для отримання інформації про файли, включаючи їх тип і MIME-тип [36];

- **GD** - бібліотека для роботи з графікою, що дозволяє створювати, змінювати і обробляти зображення [37];

- **SimpleXML** - надає простий спосіб роботи з XML-документами, дозволяючи легко аналізувати та витягувати дані [38];

- **xml** - базова підтримка роботи з XML-документами [39] ;

- **xmlreader** - надає спосіб для ефективного, потокового читання XML-документів [40];

- **xmlwriter** - дозволяє створювати XML-документи з використанням потокового інтерфейсу [41] ;

- **zip** - надає функції для роботи з ZIP-архівами, дозволяючи створювати, читати та модифікувати архіви [42];
- **zlib** - використовується для стиснення та розпаковування даних, що особливо корисно при роботі з архівами та потоками даних [43];
- **ezyang htmlpurifier** - бібліотека для очищення HTML від шкідливого коду, забезпечуючи безпечний вивід [44];
- **maennchen zipstream-php** - дозволяє створювати ZIP-архіви на льоту і передавати їх через HTTP [45] ;
- **markbaker/complex** - бібліотека для роботи з комплексними числами [46];
- **markbaker/matrix** - бібліотека для роботи з матрицями [47];
- **psr/http-client** - інтерфейси для реалізації HTTP-клієнтів [48];
- <https://github.com/php-fig/http-message> - інтерфейси для HTTP-повідомлень [49];
- **psr/http-factory** - інтерфейси для створення HTTP-запитів та відповідей [50];
- <https://github.com/php-fig/http-factory> [51]
- **psr/simple-cache** - стандартний інтерфейс для простих кеш-систем [52].

3.4 Результати реалізації системи пошуку на корпоративному сайті

Фрагмент розробленої системи зображено на рис. 3.6-3.9. На рис. 3.6 показана головна сторінка. Сторінка з полем для пошукового запиту розміщена по центру, біля логотипу компанії (рис. 3.7).

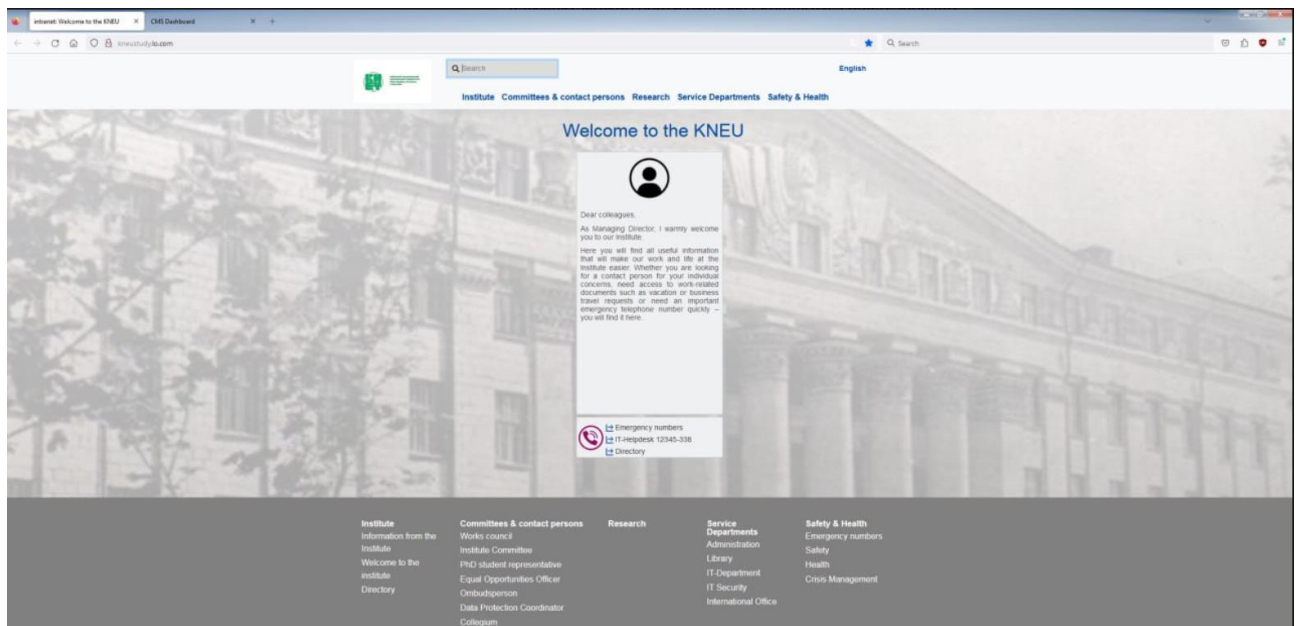


Рисунок 3.6 – Головна сторінка

Джерело: розроблено автором самостійно

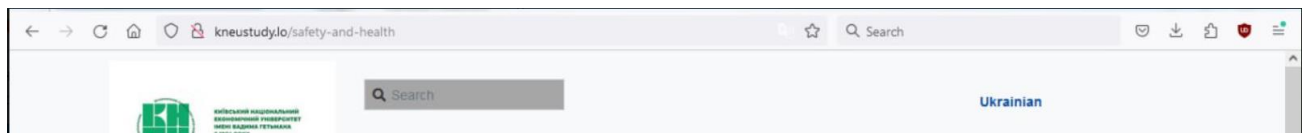


Рисунок 3.7 – Сторінка з пошуком

Джерело: розроблено автором самостійно

На рис. 3.8-3.11 показані результати пошуку за запитом.

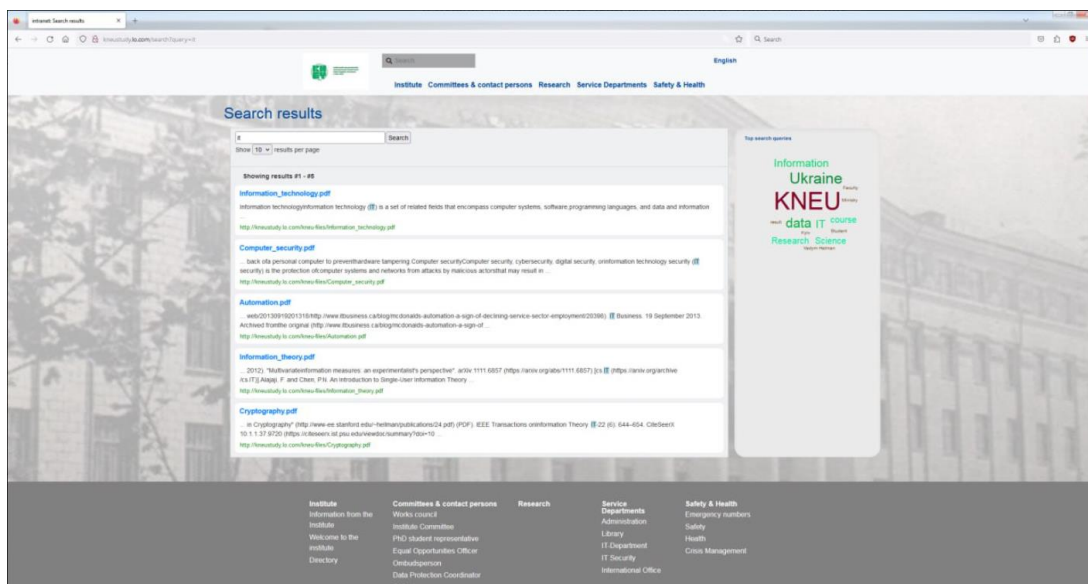


Рисунок 3.8 – Результати пошуку. Відображення файлів

Джерело: розроблено автором самостійно

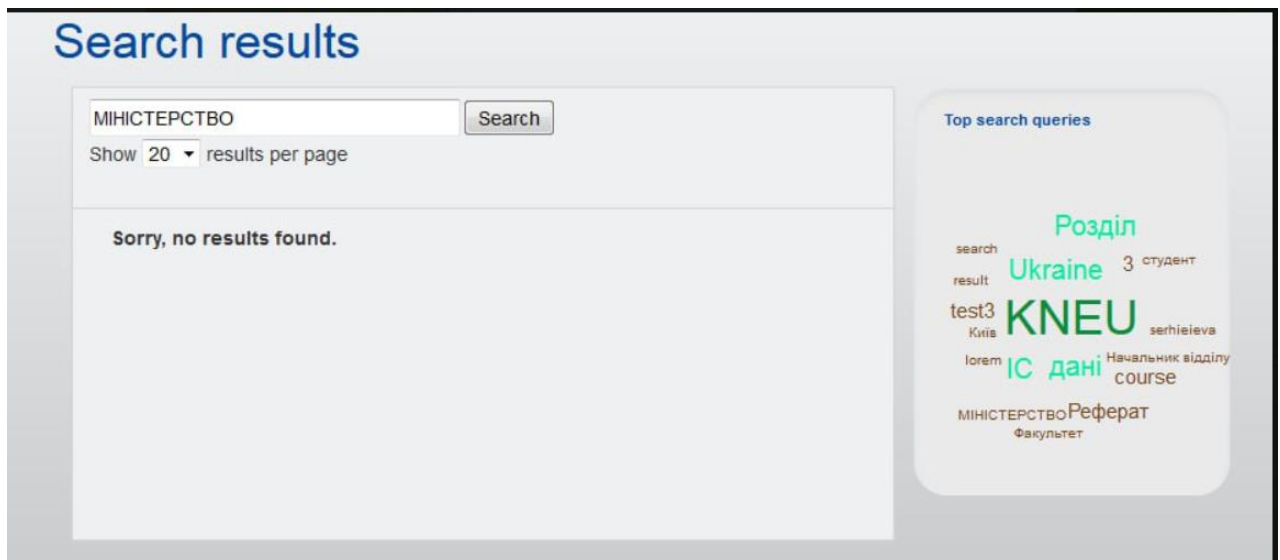


Рисунок 3.9 – Відображення інформації у випадку відсутності результатів
Джерело: розроблено автором самостійно

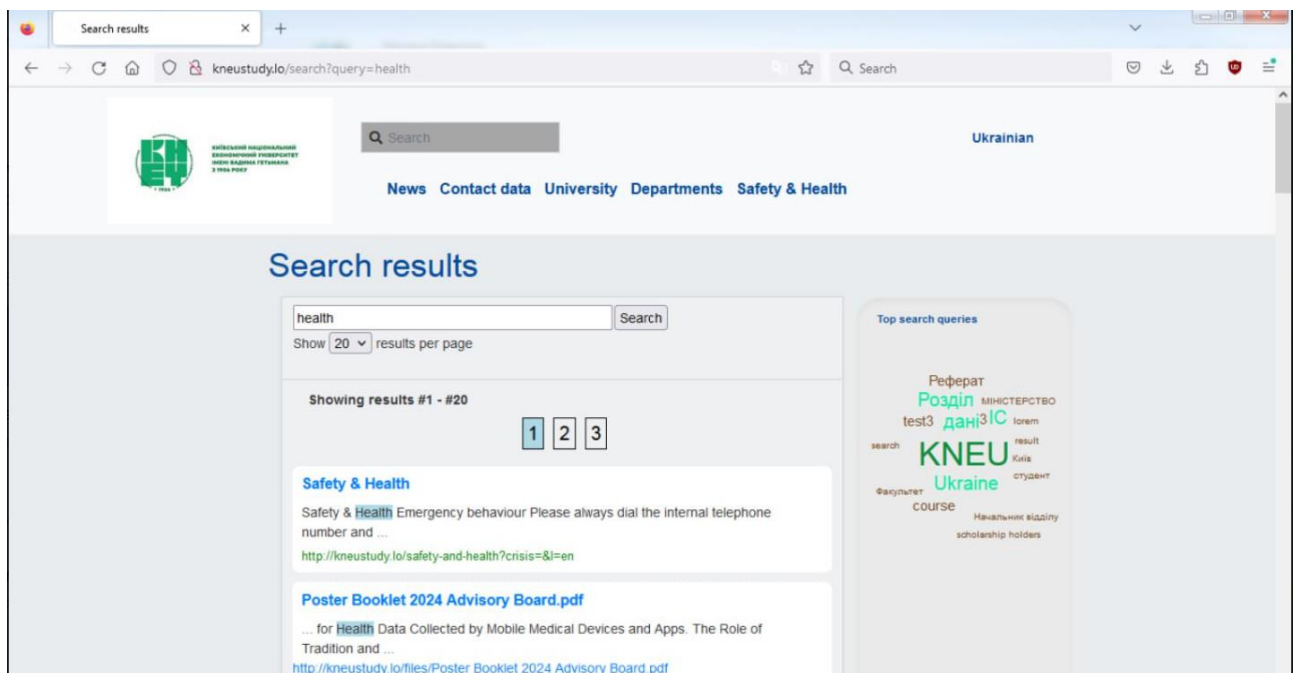


Рисунок 3.10 – Результати пошуку. Результати знайдених веб-сторінок
Джерело: розроблено автором самостійно

Розглянемо серверну частину реалізованої системи.

На рис. 3.11 зображена файлова структура проекту. Безпосередньо, сайт знаходиться в папці webroot.

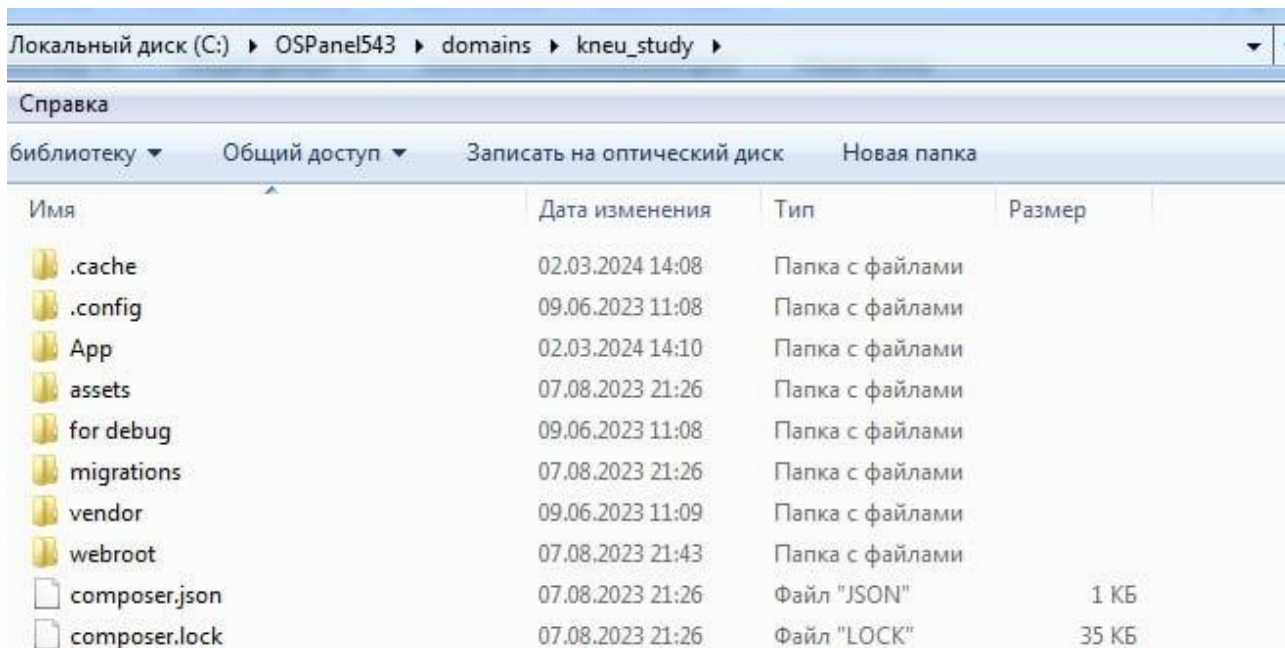


Рисунок 3.11 – Файлова структура проєкту.

Джерело: розроблено автором самостійно

В папці .cache (рис. 3.12) знаходиться згенерований кеш:

- кеш файлів та сторінок розташований у папці content (рис. 3.13);
- кеш запитів зберігається у папці queries (рис. 3.14).

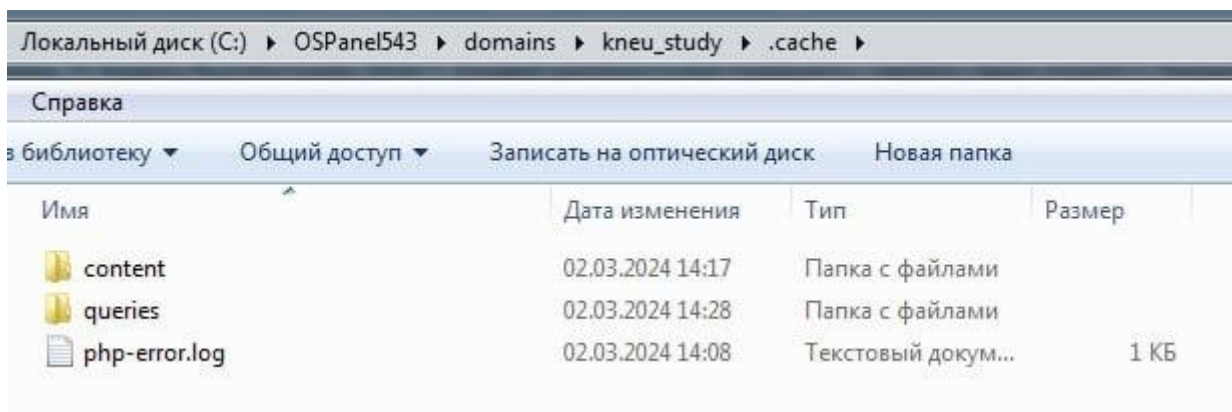


Рисунок 3.12 – Файлова структура згенерованих кеш-файлів та сторінок

Джерело: розроблено автором самостійно

На рис. 3.13 показані папки, кожна з яких стосується конкретного файлу чи сторінки, які отримані в результаті пошуку. Для швидкого пошуку та унікального результату (деякі файли на сайті можуть мати однакові назви, але різний зміст) назва кеш-папки є хешем від розміщення файлу та його назви.

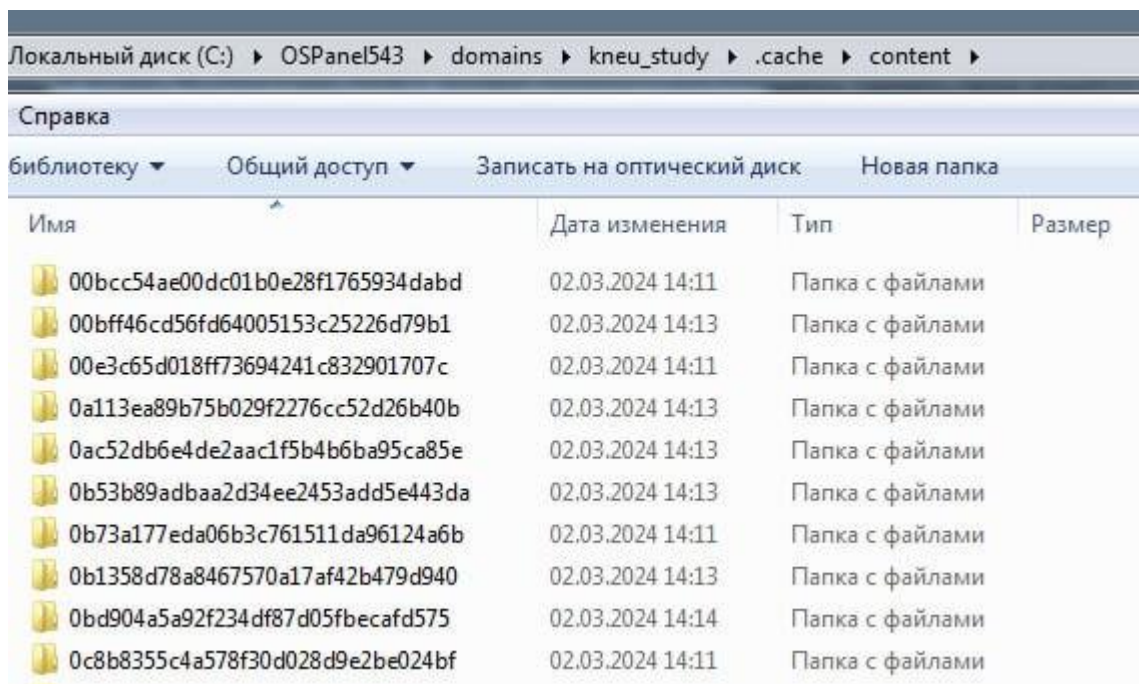


Рисунок 3.13 – Результат знайдених файлів і сторінок, розміщених у файлах

Джерело: розроблено автором самостійно

В цій папці лежать два бінарних файли - cache та bin. Перший файл необхідний для швидкого пошуку, другий - для виводу результату на екран.

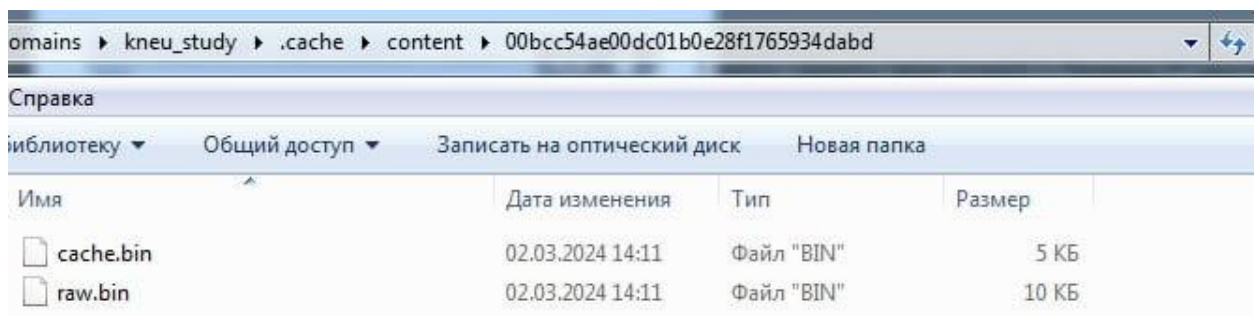


Рисунок 3.14 – Кеш файлів та сторінок, знайдених на сайті

Джерело: розроблено автором самостійно

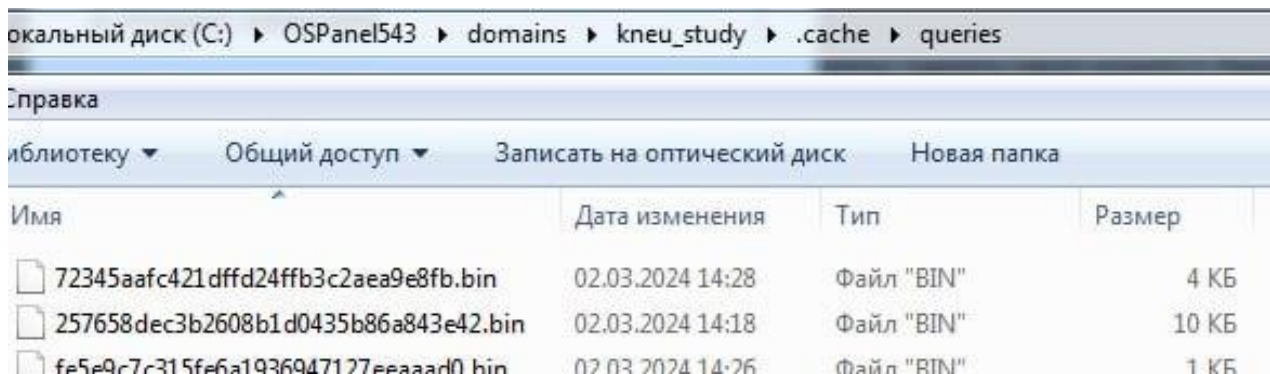


Рисунок 3.15 – Кеш запитів

Джерело: розроблено автором самостійно

Інформація про найпоширеніші запити, яка зберігається у файлі search_top_words, показана на рис. 3.16.

text	weight
test	1
KNEU	173
Ukraine	89
МІНІСТЕРСТВО	3
test test 2	1
lorem	2
test3	32
111	1
222	1
3	35
course	37
Факультет	2
Завідувач	1
Начальник відділу	4
Київ	6
IC	55
студент	3
test 4	1
Реферат	37
Розділ	59
дані	85
область застосування	1

Рисунок 3.15 – Інформація про пошукові запити з їх частотою

Джерело: розроблено автором самостійно

На рис. 3.16 – 3.18 показана адміністративна панель налаштування пошуку.

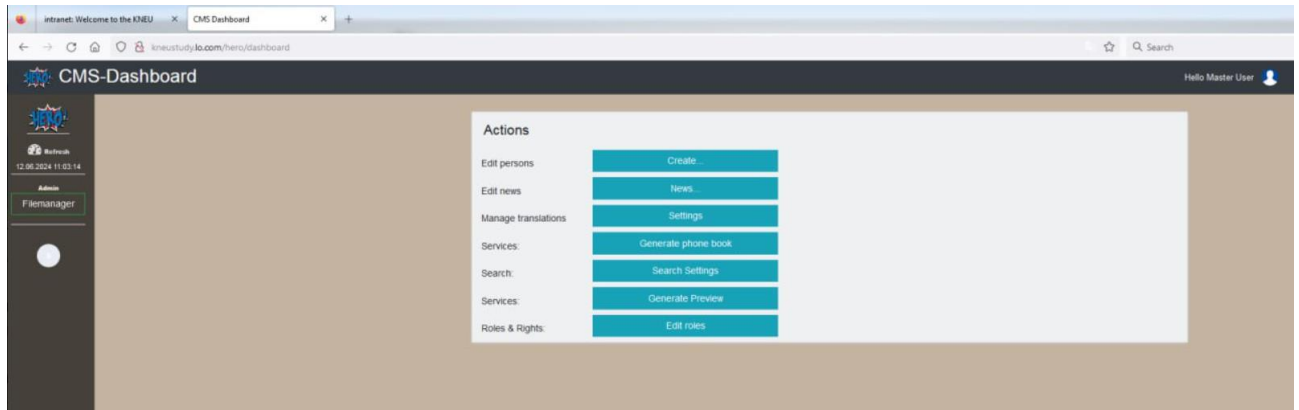


Рисунок 3.16 - Панель адміністрування сайту. Доступ до налаштування модуля пошуку.

Джерело: розроблено автором самостійно

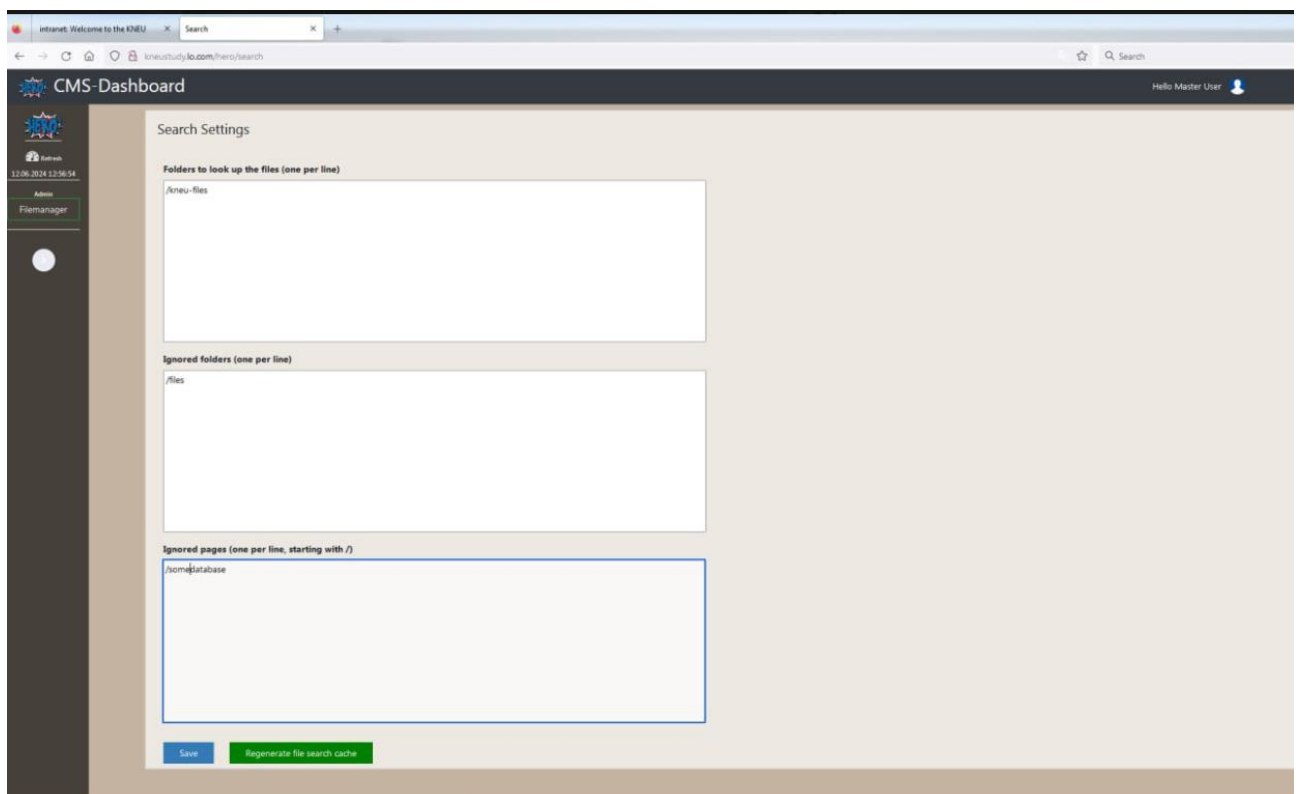


Рисунок 3.17 - Поля налаштування пошуку, кнопки збереження та перегенерування кешу.

Джерело: розроблено автором самостійно

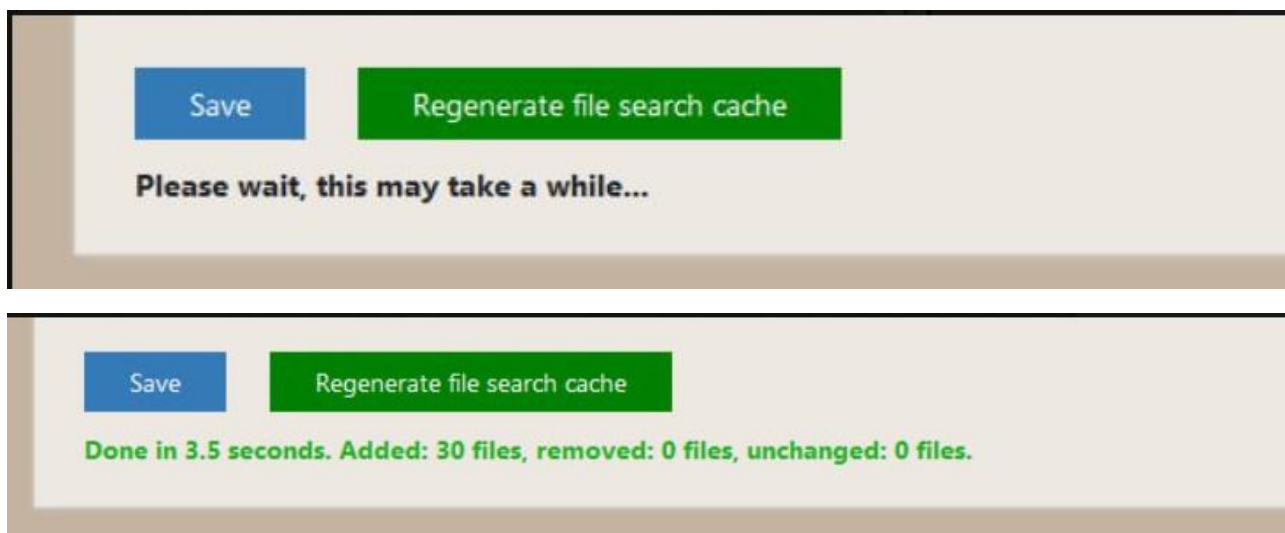


Рисунок 3.18 - Процес та результат генерування кешу.

Джерело: розроблено автором самостійно

ВИСНОВКИ

В результаті виконання випускного бакалаврського проєкту було досліджено теоретичні і практичні питання щодо проєктування систем пошуку інформації на корпоративному сайті.

Пошукові системи в корпоративній мережі призначені для роботи з масивами текстових документів інтернет-сторінок підприємства. які програми сервері локальної мережі, здійснюється з робочих станцій співробітників і мають бути реалізовані в мережному варіанті, при якому доступ до бази даних адмініструється правами доступу. Оскільки такому пошуку притаманні великі обсяги інформації та обмеженість в правах доступу, це вимагає створення спеціальних пошукових програм, які вбудовані в веб-сайт підприємства.

В роботі визначені вимоги до системи, зокрема, визначені бізнес цілі: підвищення продуктивності співробітників, покращення процесів прийняття рішень, зменшення часу пошуку інформації, підвищення рівня задоволеності клієнтів, підвищення конкурентоспроможності компанії, покращення якості прийняття рішень. Функціональні вимоги запропоновано розглядати в розрізі користувачів, адміністратора і системи.

Зокрема, користувач в системі може здійснювати вхід в систему (авторизація), налаштування (зміна мови, налаштування інтерфейсу, параметрів відображення результатів), вводити пошуковий запит, переглядати результати пошуку. Адміністратор – налаштовувати систему і формувати кеш-файли. Система здійснює пошук за алгоритмом та генерує результати.

Описана характеристика системи, розроблена інформаційна модель та алгоритм роботи із системою.

Змодельована поведінка системи, яка показує, які можливості має система, розглянуті основні блоки, представлені у вигляді діаграми визначення блоків. Зокрема, система має модуль SearchSystem, який є основним у системі, а також модулі авторизація, модуль користувача, модуль адміністратора.

Запропоновано такі класи: Request_search (безпосередньо пошуковий запит, адреси, де знаходяться файли та сторінки), Search_settings (налаштування системи), Stop_words (перелік стоп-слів), Cache_Rezult (результат запиту), Cache_error (помилки, якщо вони виникають), Cache_files – (створює кеш файли), Search_top_words (список найбільш поширених пошукових слів).

Запропоновано інформаційне, технічне і програмне забезпечення.

Для розроблення системи було обрано мову програмування PHP, середовище розробки – PHPStorm. Результати реалізації програми показані у вигляді екранних форм.

Запропонована система може бути використана для будь-якого підприємства чи організації, яка має корпоративний веб-портал.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сурков Д. Корпоративний веб-сайт: що треба врахувати для збільшення продажів [Електронний ресурс] / Д. Сурков. – 2023. – Режим доступу до ресурсу: <https://shop-express.ua/ukr/blog/creating-website-for-corporation/>. (дата звернення 20.04.2024).
2. Miller E. Cognitive Search Is The AI Version Of Enterprise Search [Електронний ресурс] / Emily Miller. – 2017. – Режим доступу до ресурсу: <https://www.forrester.com/blogs/17-06-12-cognitive-search-is-the-ai-version-of-enterprise-search/>. (дата звернення 20.04.2024).
3. Корпоративний сайт: від аналізу до готового рішення [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://fractus.com.ua/uk/blog/korporativnij-sajt-vid-analizu-do-gotovogo-rishennya/>.
4. OPCAT 4.0 [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://opcat.software.informer.com/>.
5. Object-Process Methodology [Електронний ресурс] – Режим доступу до ресурсу: <https://www.iso.org/obp/ui/#iso:std:iso:pas:19450:ed-1:v1:en> (дата звернення 29.05.2024).
6. DtSearch [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.dtsearch.com/>.
7. About Greenstone [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.greenstone.org/>.
8. Google.com [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <http://www.google.com>
9. Autonomy. Making EVs accessible and affordable [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.autonomy.com/>.
10. Convera [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://convera.com/>.

11. Infostream [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <http://infostream.com.ua>
12. Meta.ua [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <http://www.meta.ua>
13. Андон П.І., Деречкий В.А. Процесори пошуку та аналізу природномовної текстової інформації в аналітичних системах – 2001.– № 3-4. – С. 144–163.
14. Enterprise Architect [Електронний ресурс] – Режим доступу до ресурсу: <https://sparxsystems.com/>. (дата звернення 20.04.2024).
15. SysML [Електронний ресурс] – Режим доступу до ресурсу: <http://www.omg.sysml.org/index.htm>. (дата звернення 20.04.2024).
16. UML - мова графічного опису для об'єктного моделювання [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://allref.com.ua/uk/skachaty/UML_-_mova_grafichnogo_opisu_dlya_obyektnogo_modelyuvannya (дата звернення 20.04.2024).
17. Каграманова Ю. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти [Електронний ресурс] / Ю. Каграманова. – 2022. – Режим доступу до ресурсу: https://dou.ua/forums/topic/40575/?__cf_chl_tk=0hhQ.SuMqF02cl4CE8z0X5Jzus2WP3ToLFHiICvrJSM-1681047323-0-gaNycGzNC7sOPCAT [Електронний ресурс] – Режим доступу до ресурсу: <http://esml.iem.technion.ac.il/opcat-installation/> . (дата звернення 20.04.2024).
18. UML [Електронний ресурс] – Режим доступу до ресурсу: <http://www.uml-diagrams.org/>. (дата звернення 20.04.2024).
19. Організація комп'ютерних мереж [Електронний ресурс] : підручник: для студ. спеціальності 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського ; Ю. А. Тарнавський, І. М. Кузьменко. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 259 с.119.

20. PDO. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/pdo.installation.php>
21. Igbinary. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.igbinary.php>
22. URL. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.curl.php>
23. libxml. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.libxml.php>
24. DOM. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/intro.dom.php>
25. iconv. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.iconv.php>
26. smalot pdfparser. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://github.com/smalot/pdfparser> :
27. symfony polyfill-mbstring. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://github.com/symfony/polyfill-mbstring>
28. Zlib. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.zlib.php>
29. phoffice phpword. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://github.com/PHPOffice/PHPWord>
30. json. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.json.php>
31. xml. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.xml.php>
32. laminas laminas-escaper. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://github.com/laminas/laminas-escaper>
33. ctype. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.ctype.php>
34. mbstring. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://www.php.net/manual/en/book.mbstring.php>

35. phpoffice phpspreadsheet. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://github.com/PHPOffice/PhpSpreadsheet>
36. fileinfo. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.fileinfo.php>
37. GD. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.image.php>
38. SimpleXML. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.simplexml.php>
39. xml. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.xml.php>
40. xmlreader. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.xmlreader.php>
41. xmlwriter. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.xmlwriter.php>
42. zip. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.zip.php>
43. zlib. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.php.net/manual/en/book.zlib.php>
44. ezyang htmlpurifier. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://github.com/ezyang/htmlpurifier>
45. maennchen zipstream-php. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://github.com/maennchen/ZipStream-PHP>
46. markbaker/complex. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://github.com/MarkBaker/PHPComplex>
47. markbaker/matrix. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://github.com/MarkBaker/PHPMatrix>
48. http-повідомлення. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://github.com/php-fig/http-message>.
49. psr/http-client. [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://github.com/php-fig/http-client>

50. http-message . [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://github.com/php-fig/http-message>
51. http-factory . [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://github.com/php-fig/http-factory>
52. psr/simple-cache. [Электронный ресурс]. – 2024. – Режим доступа до ресурсу: <https://github.com/php-fig/simple-cache>

ДОДАТКИ

Фрагменти коду програми

Програмний код для формування кеш-файлу:

```
<?php

namespace App\UseCases;

ini_set('log_errors', 1);
ini_set('error_log', $_SERVER['DOCUMENT_ROOT'] . '/../.cache/php-error.log');

use App\Core\DBException;
use App\Helpers\FilesystemHelper;
use App\Helpers\IgnoredWordsHelper;
use App\Helpers\WebHelper;
use App\Responses\CacheResponse;
use App\Responses\Response;
use App\Services\Cache\CacheInfoService;
use App\Services\Cache\CacheService;

/** This class contains Cache scenarios */
class CacheUC {

    /** @var CacheService Cache service */
    protected CacheService $cacheService;

    /** @var CacheInfoService Cache info service (included/excluded dirs or pages) */
    protected CacheInfoService $cacheInfoService;

    public function __construct() {
        $this->cacheService = new CacheService();
        $this->cacheInfoService = new CacheInfoService();
    }

    /**      * This function builds a complete cache of valid files and pages
     * @return CacheResponse
```

```

* @throws DBException
*/

public function buildCache(): CacheResponse {
    $ignoredWords = IgnoredWordsHelper::load();

    $includedDirs = $this->includedDirs();
    $excludedDirs = $this->excludedDirs();
    $excludedPages = $this->excludedPages();

    $this->cacheService->prepareCaching($ignoredWords);
    $this->cacheService->cacheFiles($includedDirs, $excludedDirs);
    $this->cacheService->cachePages($excludedPages);
    list ($added, $removed, $unchanged, $time) = $this->cacheService->finish();

    return CacheResponse::ok($added, $removed, $unchanged, $time);
}

/**
 * Stores all the settings related to cache
 * @param string[] $includedDirs array of included directories
 * @param string[] $excludedDirs array of excluded directories
 * @param string[] $excludedPages array of excluded pages
 * @return Response
 * @throws DBException
 */

public function setDirs(
    array $includedDirs,
    array $excludedDirs,
    array $excludedPages
): Response {

    $includedDirs = FilesystemHelper::onlyExistingDirs($includedDirs);
    $excludedDirs = FilesystemHelper::onlyExistingDirs($excludedDirs);

```

```

if ($includedDirs === null || $excludedDirs === null) {
    return Response::withError(Response::INVALID_INPUT);
}

$ch = curl_init();
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_HEADER, true);

$domain = WebHelper::domain();
$numExcludedPages = count($excludedPages);

for ($i = 0; $i < $numExcludedPages; $i++) {
    if (!WebHelper::isLocalHTMLPage($ch, $domain, $excludedPages[$i])) {
        return Response::withError(Response::INVALID_INPUT);
    }
}

$this->cacheInfoService->putIncludedDirs($includedDirs);
$this->cacheInfoService->putExcludedDirs($excludedDirs);
$this->cacheInfoService->putExcludedPages($excludedPages);

return Response::ok();
}

/**
 * @return string[] Array of included directories
 * @throws DBException
 */
public function includedDirs(): array {
    return $this->cacheInfoService->includedDirs();
}

/**

```

```
* @return string[] Array of excluded directories
* @throws DBException
*/
public function excludedDirs(): array {
    return $this->cacheInfoService->excludedDirs();
}

/**
* @return string[] Array of excluded pages
* @throws DBException
*/
public function excludedPages(): array {
    return $this->cacheInfoService->excludedPages();
}
}
```

Програмний код для виконання пошуку.

```
<?php
namespace App\UseCases;

ini_set('log_errors', 1);
ini_set('error_log', $_SERVER['DOCUMENT_ROOT'] . '/../.cache/php-error.log');

use App\Core\DBException;
use App\Entities\Search\SearchQuery;
use App\Entities\Search\TopQueries;
use App\Entities\Search\SearchCache;
use App\Helpers\IgnoredWordsHelper;
use App\Helpers\LangHelper;
use App\Responses\Response;
use App\Responses\SearchResponse;
use App\Services\Search\SearchInfoService;
use App\Services\Search\SearchService;

/** This class contains Search scenarios */
class SearchUC {
    /** @var int Max words in a search query */
    protected const WORD_LIMIT = 20;

    /** @var int Max chars in a search query */
    protected const CHAR_LIMIT = 128;

    /** @var SearchService Search service */
    protected SearchService $searchService;

    /** @var SearchInfoService Search meta information service */
    protected SearchInfoService $searchInfoService;
```

```

public function __construct() {
    $this->searchService = new SearchService();
    $this->searchInfoService = new SearchInfoService();
}

/**
 * Performs search by the given parameters
 * @param string $queryStr Raw search query string
 * @param int $page Page number
 * @param int $perPage Results per page
 * @return SearchResponse An object containing all the information about results
 * @throws DBException
 */
public function search(string $queryStr, int $page, int $perPage): SearchResponse {
    if (mb_strlen($queryStr) > self::CHAR_LIMIT) {
        return SearchResponse::withError(Response::INVALID_INPUT);
    }

    $query = new SearchQuery($queryStr, IgnoredWordsHelper::load());
    $queryStr = $query->str();

    if (empty($queryStr)) {
        return SearchResponse::withError(Response::EMPTY_INPUT);
    }

    if ($query->isUnspecified()) {
        return SearchResponse::withError(Response::UNSPECIFIED_INPUT);
    }

    if ($query->numWordsExact() > self::WORD_LIMIT) {
        return SearchResponse::withError(Response::INVALID_INPUT);
    }

    $searchCache = $this->searchService->search($query);
    if ($query->onlyExact()) {

```

```

        $searchCache->removeNonExact();
    }

    if ($searchCache->numResults() == 0) {
        return SearchResponse::withError(Response::EMPTY_RESPONSE);
    }

    $this->searchInfoService->incQueryWeight($queryStr);

    return self::buildSearchResponse($searchCache, $page, $perPage);
}

/**
 * Returns top search queries by weight
 * @param int $num Number of top search queries
 * @return TopQueries
 * @throws DBException
 */
public function topQueries(int $num): TopQueries {
    return $this->searchInfoService->topQueries($num);
}

/** This function prepares search response based on search results and pagination
 * @param SearchCache $searchCache Search cache object which contains the results
 * @param int $page Page number
 * @param int $perPage Results per page
 * @return SearchResponse
 */
protected static function buildSearchResponse(
    SearchCache $searchCache,
    int $page,
    int $perPage
): SearchResponse {
    $searchCache->sort(LangHelper::lang());

```

```
$numResults = $searchCache->numResults();  
if ($numResults <= $perPage) {  
    return SearchResponse::ok($searchCache->results(), 1, 1, 1, $numResults);  
}  
  
$maxPage = intdiv(($numResults - 1), $perPage) + 1;  
  
if ($page < 1 || $page > $maxPage) {  
    $page = 1;  
}  
$resultsArr = $searchCache->paginate($page, $perPage);  
  
$firstIndex = ($page - 1)*$perPage + 1;  
$lastIndex = $firstIndex + count($resultsArr) - 1;  
  
return SearchResponse::ok($resultsArr, $page, $maxPage, $firstIndex, $lastIndex);  
}  
}
```