

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: **Розробка мобільного застосунку для роботи з електронними документами у середовищі ОС Android**

Виконав: студент групи 3157ст

Тимошенко П.А.

(підпис, ПІБ)

Керівник роботи:

доц. каф. ПЗАС, к.т.н. Суслов С.В.

(посада, науковий ступень вчене звання)

(підпис, ПІБ)

Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»
 «ЗАТВЕРДЖУЮ»

Гарант освітньої програми

_____ доц. Макарова Л.М.

(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти «бакалавр»

Студенту Тимошенко Павлу Андрійовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка мобільного застосунку для роботи з електронними документами у середовищі ОС Android. _____

Керівник роботи Сулов С.

Затверджені наказом ректора № 257-уч _____ від «17» березня 2023 р.

2. Термін подання роботи: 29.05.2023 р.

3. Вихідні дані по роботі: Дані для постановки задачі потрібно отримати шляхом аналізу застосовної області

4. Перелік питань, що належать до розробки (найменування розділів)

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів Мета та завдання кваліфікаційної роботи; Модель варіантів використання застосунку; Подання архітектури програмного забезпечення; Структура класів реалізації; Діаграми взаємодії реалізацій варіантів використання; Інтерфейс користувача; Висновки. _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 13.10.20 р. _____**КАЛЕНДАРНИЙ ПЛАН**

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	24.03.2023	
2.	Аналіз практичної задачі інженерії ПЗ	28.03.2023	
3.	Аналіз вимог до ПЗ	31.03.2023	
4.	Розробка архітектури ПЗ	27.04.2023	
5.	Розробка проекту ПЗ	06.05.2023	
6.	Реалізація та тестування ПЗ	13.05.2023	
7.	Підготовка розділу Результати розробки ПЗ	17.05.2023	
8.	Підготовка розділу з охорони праці	18.05.2023	
9.	Підготовка розділу ВИСНОВКИ	19.05.2023	
10.	Оформлення списку використаних джерел та додатків	20.05.2023	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	10.06.2023	
12.	Підготовка презентації та доповіді	12.06.2023	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	14.06.2023	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	24.06.2023	

* - а результатами переддипломної практики (ПП), яка була з 1.02.2023 по 19.03.2023 р.

Студент

(підпис)

Тимошенко П.А.

(ПІБ)

Керівник роботи

(підпис)

Суслов С.В.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена визначення сутності і стану практичної задачі інженерії програмного забезпечення з теми «Розробка мобільного застосунку для роботи з електронними документами у середовищі ОС Android».

Об'єктом кваліфікаційної роботи є процес розробки мобільного застосунку для роботи з електронними документами у середовищі ОС Android. У кваліфікаційній роботі проведено аналіз предметної галузі, проаналізовано існуючі програмні продукти та їх функціонал.

Програмне забезпечення розробляється за допомогою Android Studio на мові Java.

Кваліфікаційна робота викладена на 83 сторінках друкованого тексту, містить 30 рисунки, 12 таблиць, список використаних джерел із 9 найменувань та 5 додатки.

Кваліфікаційна робота виконана українською мовою.

ANNOTATION

The qualification work is devoted to determining the essence and state of the practical task of software engineering on the topic "Development of a mobile application for working with electronic documents in the Android OS environment."

The object of the qualification work is the process of developing a mobile application for working with electronic documents in the Android OS environment. In the qualification work, an analysis of the subject area was carried out, existing software products and their functionality were analyzed.

The software is developed using Android Studio in Java.

The qualification work is laid out on 83 pages of printed text, contains 30 figures, 12 tables, a list of used sources of 9 titles and 5 appendices.

The qualification work was completed in Ukrainian.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ЯКУ МАЄ РОЗВ'ЯЗУВАТИ КВАЛІФІКАЦІЙНА РОБОТА, ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ РОБОТИ З ЕЛЕКТРОННИМИ ДОКУМЕНТАМИ У СЕРЕДОВИЩІ ОС ANDROID.	10
1.1 Аналіз практичної задачі інженерії програмного забезпечення, яку має розв'язувати кваліфікаційна робота;	10
1.2 Аналіз сучасних застосунків для роботи з текстовими документами	10
1.3 Мета роботи	16
1.4 Постановка задачі на розробку мобільного застосунку	16
1.5 Обґрунтування створення мобільного застосунку.	17
1.6 Аналіз вимог	18
2. Розробка мобільного застосунку для роботи з електронними документами у середовищі ОС Android	20
2.1 Вибір архітектурного стилю	20
2.2 Вибір методології реалізації	20
2.3 Визначення архітектури застосунку	21
2.4 Модель варіантів використання застосунку	22
2.5 Структура класів	33
2.6 Специфікація класів	33
2.7 Діаграми взаємодії варіантів використання	35
2.8 Проект користувальницького інтерфейсу	38
2.9 Тестування основних функцій мобільного додатку	42
2.10 Випробування програмного забезпечення	45
4.Охорона праці	53
4.1 Основне положення	53
4.2 Заходи щодо безпечної праці при роботі з мобільним пристроєм	54

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ РОБОТИ З ЕЛЕКТРОНИМИ ДОКУМЕНТАМИ У СЕРЕДОВИЩІ ОС ANDROID	60
ДОДАТОК Б – Інструкція користувача	66
ДОДАТОК В – Опис програмного забезпечення	72
ДОДАТОК Г – Текст програми	75
ДОДАТОК Д - Програма та методика випробувань програмного забезпечення	80

ВСТУП

Електроні пристрої полегшили як професійну роботу так і повсякденне життя людей. Основними перевагами мобільних пристроїв є їх фінансова доступність та зручний інтерфейс, але виконання різноманітних задач призводить до обробки та зберігання великої кількості інформації. Саме робота з інформацією і є основною метою користування мобільними пристроями, у більшості людей свої вимоги та потреби до оброблення інформації. Для виконання цих вимог розробляються різноманітні мобільні застосунки для роботи з електронними документами. Існуючі застосунки мають свої недоліки, отже розробка мобільного застосунку для роботи з електронними документами є актуальною.

Для розробки нового мобільного застосунку потрібно визначити основні вимоги для користувачів шляхом дослідження їх потреб. Дослідження готової продукції також допоможе з формуванням основних вимог на базі їх функціоналу[1].

Об'єктом кваліфікаційної роботи є процес розробки мобільного застосунку для роботи з електронними документами у середовищі ОС Android. У кваліфікаційній роботі буде розглянуто повний ланцюжок розробки застосунку: від аналізу вимог та готових застосунків, до тестування готового застосунку.

Метою роботи є створення мобільного застосунку для роботи з електронними документами у середовищі ОС Android, що є практичною задачею інженерії програмного забезпечення, яка характеризується комплексністю та невизначеністю умов.

Для досягнення поставленої мети потрібно вирішити наступні завдання:

- провести дослідження предметної області;
- провести аналіз існуючих аналогів застосунків для роботи з електронними документами на базі Android;
- провести аналіз методів розробки на базі Android;
- створити проєкт програмного забезпечення;

- провести тестування програмного забезпечення;
- ознайомитися з питаннями охорони праці при розробці ПЗ та роботі з комп'ютером.

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ЯКУ МАЄ РОЗВ'ЯЗУВАТИ КВАЛІФІКАЦІЙНА РОБОТА, ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ РОБОТИ З ЕЛЕКТРОННИМИ ДОКУМЕНТАМИ У СЕРЕДОВИЩІ ОС ANDROID.

1.1 Аналіз практичної задачі інженерії програмного забезпечення, яку має розв'язувати кваліфікаційна робота;

Згідно аналізу задачі інженерії програмного забезпечення сформулюємо постановку задачі, яку має розв'язувати кваліфікаційна робота : необхідно розробити мобільний застосунок для роботи з електронними документами у середовищі ОС Android, яке задовольняє наступним функціональним вимогам: відкриття та зберігання, редагування, видалення, коригування текстового інтерфейсу та сортування текстових документів.

Мобільний пристрій має задовольняти мінімальним технічним вимогам: операційна система: Android версії 6.0 та пам'ять: 80 МБ для роботи з додатком

1.2 Аналіз сучасних застосунків для роботи з текстовими документами

Невід'ємною частиною життя людей є навчання. Люди навчаються на базі отриманої інформації. Для того щоб полегшити навчання, люди почали зберігати отриману інформацію на зовнішніх носіях: спочатку прості рисунки, а потім рукописні книги. Але такі джерела не можуть зберігатися довго, а кількість нової інформації лише зростала. Тому створювалися більш

безпечні джерела: від простого паперового листа до цілих бібліотек різноманітних книг.

У сучасному світі з виникненням електроніки основна задача зберігання інформації стала більш зручною. На електронних носіях можливо зберігати цілі бібліотеки інформації. Також електроніка допомагає більш швидко та надійно обмінюватися інформацією.

З появою мобільних пристроїв користувачі прагнуть отримати більш функціональні та зручні у використанні засоби доступу до інформації. Наразі існує багато мобільних застосунків, які виконують лише окремі потреби користувачів. Розробка зручного та функціонального застосунку для роботи з текстовими документами, що зможе задовольнити вимоги широкого кола користувачів є актуальною проблемою.

У цій кваліфікаційній роботі розглядається розробка застосунку для роботи з електронними документами у середовищі ОС Android. Дане завдання є актуальним, тому що в даний час багато користувачів надають перевагу використанню мобільних пристроїв.

Основною метою даної роботи є створення функціонального та зручного у використанні застосунку, який задовольнить найбільш важливі потреби користувачів для роботи з електронними документами.

Ключовими аспектом є виконання основних функцій роботи з електронними документами: зберігання, редагування, видалення, сортування та створення нових текстових документів.

Також важливий аспект застосунку – гнучкий та зрозумілий для користувача інтерфейс. Він буде допомагати користувачеві працювати з основним функціоналом.

Таким чином, дана кваліфікаційна робота спрямована на вирішення практичного завдання розробки застосунку для роботи з електронними документами з урахуванням особливостей даної предметної галузі. Розробка даного застосунку повинна забезпечити користувачам основні функції для роботи з електронними документами та зручний інтерфейс.

Для розуміння основних потреб користувачів потрібно проаналізувати досліджувану галузь, дослідити готові продукти, методи та інструменти розробки на середовищі ОС Android.

До найбільш популярних застосунків відповідного призначення можна визначити такі:

eBoox – застосунок для читання книг та електронних документів різних форматів, має комфортне завантаження у додаток злюбих директорій телефону або з Інтернету.

Основні переваги додатку:

- застосунок не містить реклами;
- оптимальна настройка для користувача.
- миттєве завантаження сторінок.
- просте та комфортне завантаження документів з карти пам'яті та Інтернету;
- комфортний інтерфейс.
- синхронізація файлів для Android – пристрою.
- служба підтримки користувача.

На рисунку 1.1 надано інтерфейс застосунку eBoox:



Рисунок 1.1 – Інтерфейс застосунку eBoox

Також у додатку є декілька каталогів книг на різних мовах для ознайомлення користувачів.

MyBook - мобільний застосунок з більш ніж 450000 книг або аудіокниг. Має книги усіх жанрів для усіх користувачів. Користувач має можливість читати і слухати книги 14 днів безкоштовно як онлайн, так і без Інтернету.

Основні переваги додатку:

- Зберігання книг і прочитання їх без Інтернету.
- Повна синхронізація акаунту між різними пристроями.
- Цитати, нотатки та рецензії книг.

Основним недоліком є платна підписка – при несплаті підписки користувач лишається майже усіх переваг.

На рисунку 1.2 надано інтерфейс застосунку MyBook:

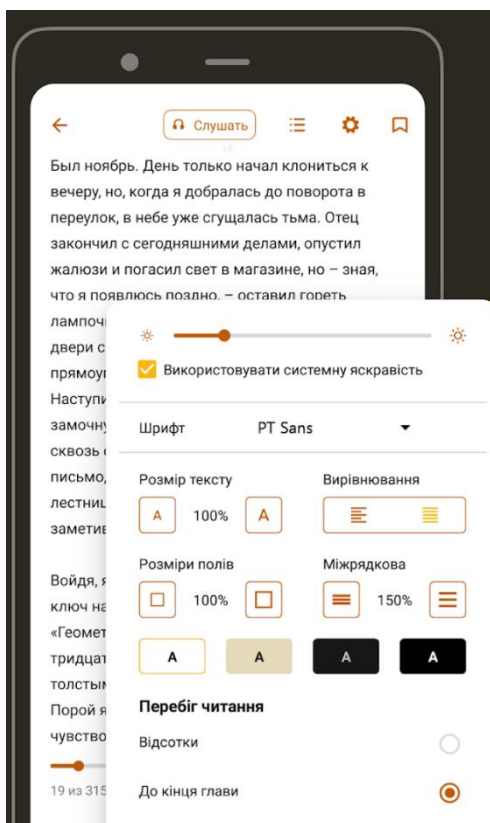


Рисунок 1.2 – Інтерфейс застосунку MyBook

Writer Plus – це програма для створення документів без зайвих зусиль та відволікання від традиційного текстового процесора. Writer Plus ідеально підходить для написання нотаток, романів, текстів, віршів, есе, чернеток на телефоні.

На рисунку 1.3 надано інтерфейс застосунку Writer Plus:

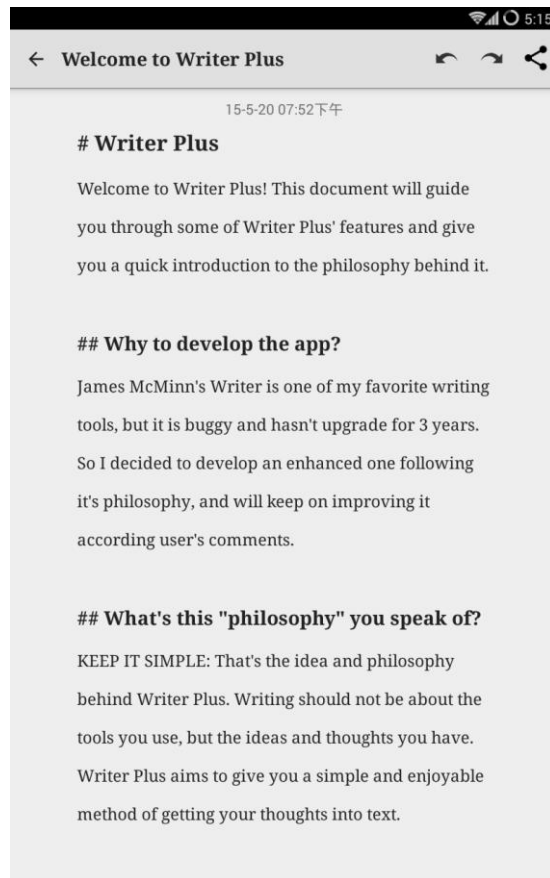


Рисунок 1.3 – Інтерфейс застосунку Writer Plus

Основні переваги додатку:

- відкриття, редагування та зберігання звичайних текстових файлів;
- створення папок для сортування текстових файлів;
- комбінаційні клавіші;
- нічний режим;
- режим обмеження використання ресурсів.

1.3 Мета роботи

Мета роботи: створення мобільного застосунку для роботи з електронними документами у середовищі ОС Android, що є практичною задачею інженерії програмного забезпечення, яка характеризується комплексністю та невизначеністю умов.

1.4 Постановка задачі на розробку мобільного застосунку

При виконанні аналізу предметної області було прийнято рішення розробки мобільного текстового редактору. Розроблений додаток має відповідати наступним вимогам:

- Відкриття та зберігання.
- Редагування.
- Видалення.
- Коригування текстового інтерфейсу – для того щоб користувачеві було більш зручно користуватися додатком, потрібно надати йому можливість налаштовувати відображення текстового документу.

Користувач матиме можливість змінювати:

- Цвітову схему.
- Шрифт.
- Розмір шрифту.
- Жирність шрифту.
- Міжрядковий інтервал.

- Сортування текстових документів – для того щоб користувачеві було більш зручно знаходити потрібні текстові документи, потрібно надати функцію сортування текстових документів по таким критеріям:
 - Назва текстового документу.
 - Формат документу.
 - Дата останнього відкриття документу.
 - Дата додавання документу до застосунку.

1.5 Обґрунтування створення мобільного застосунку.

При огляді наведених вимог до мобільного додатку та створених готових продуктів були виявлена доцільність розробки мобільного застосунку

- Робота застосунку без використання Інтернету – мобільний застосунок лише використовує внутрішню пам'ять мобільного пристрою.
- Робота з більшістю популярних текстових форматів – мобільний застосунок працює з doc, docx, pdf, txt, word і т.д.
- Гнучкий тестовий інтерфейс – користувач має змогу вносити свої корективи відображення текстового документу.
- Підтримка української мови
- Безкоштовність – мобільний застосунок буде безкоштовним.
- Підтримка нових версій мобільних систем – мобільний застосунок буде орієнтований на роботу з новітніми версіями Android.

1.6 Аналіз вимог

Новий застосунок повинен:

- Відкривати текстовий документ – повинен вміти відкривати текстові документи для подальшої роботи з ними.
- Редагування текстових документів – повинен вміти вносити корективи до відкритого текстового документу. До редагування відноситься:
 - Додавання тексту – додавання певної кількості тексту до текстового документу.
 - Видалення тексту – видалення певної кількості тексту з текстового документу.
 - Зміна розміру шрифту.
 - Зміна жирності шрифту.
 - Зміна стилю шрифту.
 - Зміна кольорових схеми текстового документу – зміна кольору шрифту і заднього плану.
 - Зміна міжрядкового інтервалу.
- Зберігати текстовий документ – повинен зберігати внесені корективи у новий або в уже відкритий текстовий документ.
- Видаляти текстові документи – повинен видаляти текстові документи з пам'яті мобільного пристрою.
- Створювати нові текстові документи – повинен створювати нові текстові документи у пам'яті мобільного пристрою для подальшої роботи з ним.
- Сортувати текстові документи – для більш зрозумілого використання користувачем мобільного додатку потрібно створити можливість орієнтуватися в великому обсязі текстових документів. Тому мобільний додаток буде сортувати текстові документи по:

- Назві текстового документу – для більш легкої орієнтації по текстовим документам потрібна функція запиту по назві текстових документів.
- Формату документу – мобільний додаток має можливість відкривати текстові документи у найбільш популярних текстових форматах – docx, txt, pdf і т.д. Тому потрібна можливість сортування по текстовим форматам.
- Дата останнього відкриття документу – для того щоб користувач міг більш легко відкривати текстові документи, які він нещодавно або давно відкривав, потрібно створити функцію сортування по даті останнього відкриття.
- Дата додавання документу - для того щоб користувач міг більш легко відкривати текстові документи, які він нещодавно або давно додав до додатку, потрібно реалізувати функцію сортування по даті додавання документу.

2. Розробка мобільного застосунку для роботи з електронними документами у середовищі ОС Android

2.1 Вибір архітектурного стилю

Для подальшого моделювання та розробки мобільного застосунку потрібно вибрати архітектуру проектування. Після аналізу вимог до мобільного додатку була вибрана архітектура обробки повідомлень.

Архітектури потоків даних характеризується як:

- Інформація і діяльність визначається через механізм повідомлень.
- Використовується коли схема взаємодії і обміну інформації в системі є складною і невідомою при проектуванні.
- Базується на тому, що компоненти реєструють свій зв'язок з певними подіями.
- Коли ці події відбуваються, компоненти отримують повідомлення.

2.2 Вибір методології реалізації

Для подальшої розробки мобільного застосунку потрібно вибрати методологію реалізації ПЗ. При огляді наведеної вище інформації можна виділити такі моменти:

- Можливість повторно використовувати створенні методи.
- Деякі методи будуть залежати від інших методів.

Тому при огляді вище наведених пунктів, можна зробити висновок, що самою ефективною методологією реалізації буде об'єктно-орієнтована методологія.

2.3 Визначення архітектури застосунку

Проаналізувавши вимоги було вирішено розробити UML діаграму компонентів для представлення роботи компонентів програми та їх взаємодію.[3]

На рисунку 2.1 надано UML діаграму компонентів:

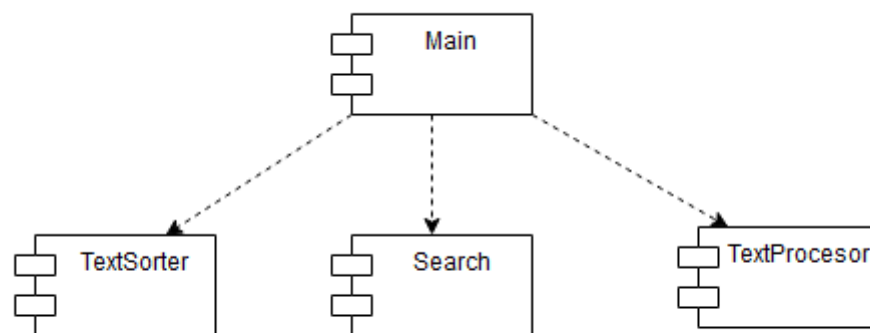


Рисунок 2.1 – UML діаграма компонентів

Main – основний компонент, який з'єднує усі інші. Представляє собою інтерфейс програми, з якої користувач виходить на потрібні йому функції.

TextSorter – представляє собою компонент усіх типів сортування текстових документів.

TextProcesor – представляє собою компонент усіх типів роботи з текстовим документом.

Search – компонент пошуку текстового документа за назвою.

2.4 Модель варіантів використання застосунку

Проаналізувавши вимоги було вирішено розробити модель варіантів використання застосунку для більш чіткого представлення роботи мобільного застосунку.

На рисунку 2.2 надано варіанти використання застосунку:



Рисунок 2.2 – Варіанти використання застосунку

Специфікація варіанту використання «Перегляд існуючих текстових документів»

Назва варіанту використання «Перегляд існуючих текстових документів»

Короткий опис: користувач програми ініціює цей об'єкт та надає можливість перегляду існуючих текстових документів.

Потоки подій

Основний потік

Функція варіанту використання набуває своєї дії після того як користувач вибере функцію «Перегляд текстових документів» у головному меню додатку.

1. Система починає пошук усіх текстових документів, які знаходяться у сховищі.

2. Система виводить на екран усі текстові документи які змогла знайти.

Альтернативні потоки

Система не змогла знайти текстові документи: якщо у сховищі не має доступних текстових документів, та система виводить помилку «Не має доступних текстових документів»

Спеціальні вимоги: немає спеціальних вимог.

Передумови: користувач має надати програмі доступ до сховища мобільного додатку.

Постумови: вивід списку доступних текстових документів.

Додаткові зауваження: не має додаткових зауважень.

Специфікація варіанту використання «Редагування текстового документа»

Назва варіанту використання «Редагування текстового документа»

Короткий опис: користувач програми ініціює цей об'єкт та надає можливість редагування уже створеного текстового документа.

Потоки подій

Основний потік

Функція варіанту використання набуває своєї дії після того як користувач відкриє текстовий документ, вибравши його зі списку знайдених текстових документів.

1. Система відображає текстовий документ користувачеві завдяки створеному інтерфейсу і надає інструменти для роботи з ним.

2. Користувач має змогу видаляти або додавати новий текст до текстового документа.

3. Користувач має змогу змінювати відображення тексту (зміна розміру, жирності, стилю шрифту, зміна кольорової схеми текстового документа, зміна міжрядкового інтервалу) після вибору функції «Коригування тексту».

4. Система змінює текст, що відображається користувачеві згідно його змін.

5. Користувач зберігає зміни у текстовому документі шляхом вибору функції «Зберегти».

6. Система зберігає змінений текстовий документ шляхом заміни попередню версію на уже змінений на носії.

Альтернативні потоки

Користувач не хоче зберігати змінений текстовий документ: користувач вибирає функцію «Закрити» і закриває текстовий документ без зберігання.

Спеціальні вимоги: немає спеціальних вимог.

Передумови: вибір текстового документа зі списку доступних текстових документів.

Постумови: відредагований текстовий документ зберігається на носії.

Додаткові зауваження: не має додаткових зауважень.

На рисунку 2.3 надано діаграму діяльності варіанту використання «Редагування текстового документа»:

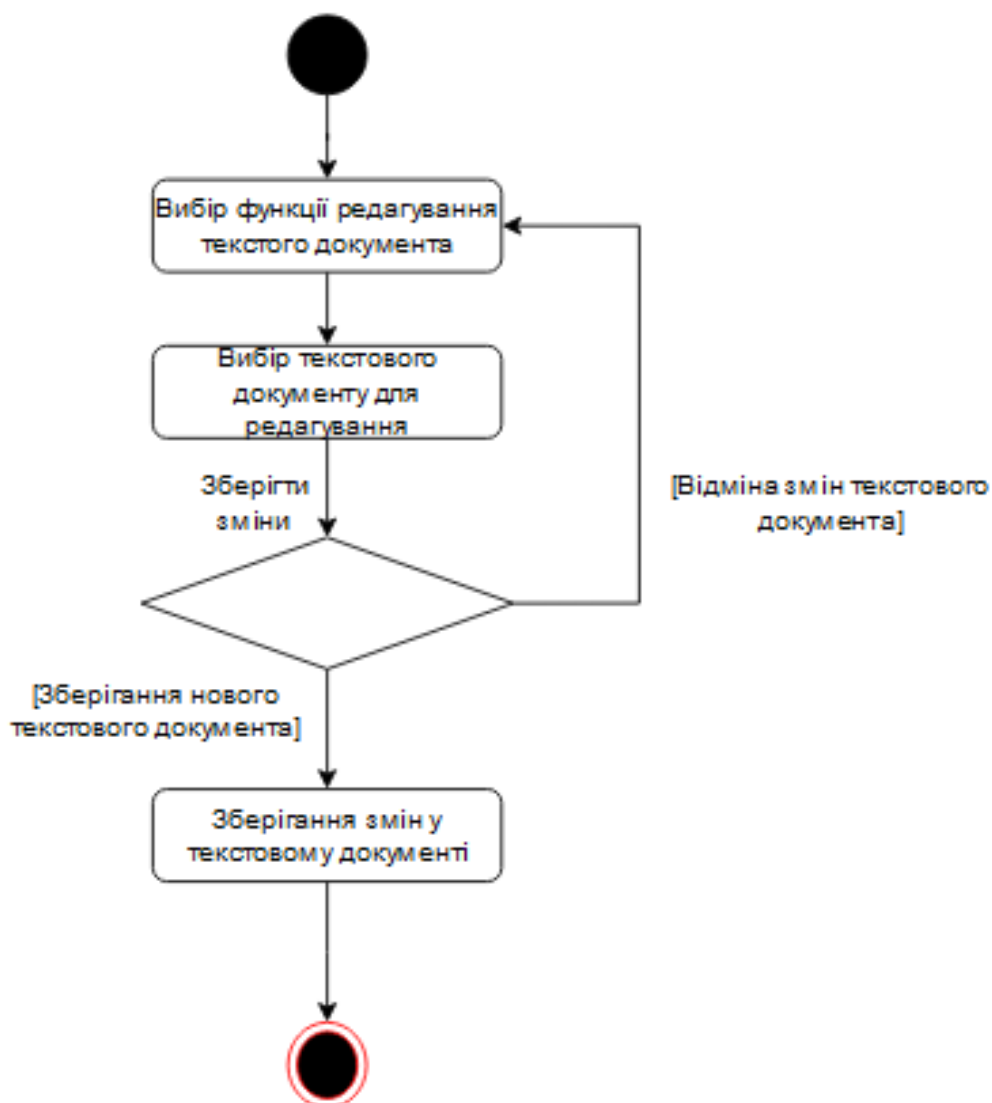


Рисунок 2.3 – Діаграма діяльності сценарію варіанту використання «Редагування текстового документа»

Специфікація варіанту використання «Видалення текстових документів»

Назва варіанту використання «Видалення текстових документів»

Короткий опис: користувач програми ініціює цей об'єкт та надає можливість видалити уже існуючий текстовий документ.

Потоки подій

Основний потік

Функція варіанту використання набуває своєї дії після того як користувач відкриє текстовий документ, вибравши його зі списку знайдених текстових документів.

1. Користувач має змогу вибрати функцію видалення текстового документа.

2. Система попереджує користувача про те що обраний текстовий документ буде видалений і надає йому можливість видалити або не видалити текстовий документ.

3. Користувач підтверджує видалення текстового документа.

4. Система видаляє вибраний текстовий документ з носія.

Альтернативні потоки

Користувач не погоджується з видаленням текстового документа:
користувач не погоджується з видаленням текстового документа і система не видаляє його.

Спеціальні вимоги: немає спеціальних вимог.

Передумови: вибір текстового документа зі списку доступних текстових документів.

Постумови: текстовий документ видаляється з носія.

Додаткові зауваження: не має додаткових зауважень.

На рисунку 2.4 надано діаграму діяльності варіанту використання «Видалення текстового документа»:

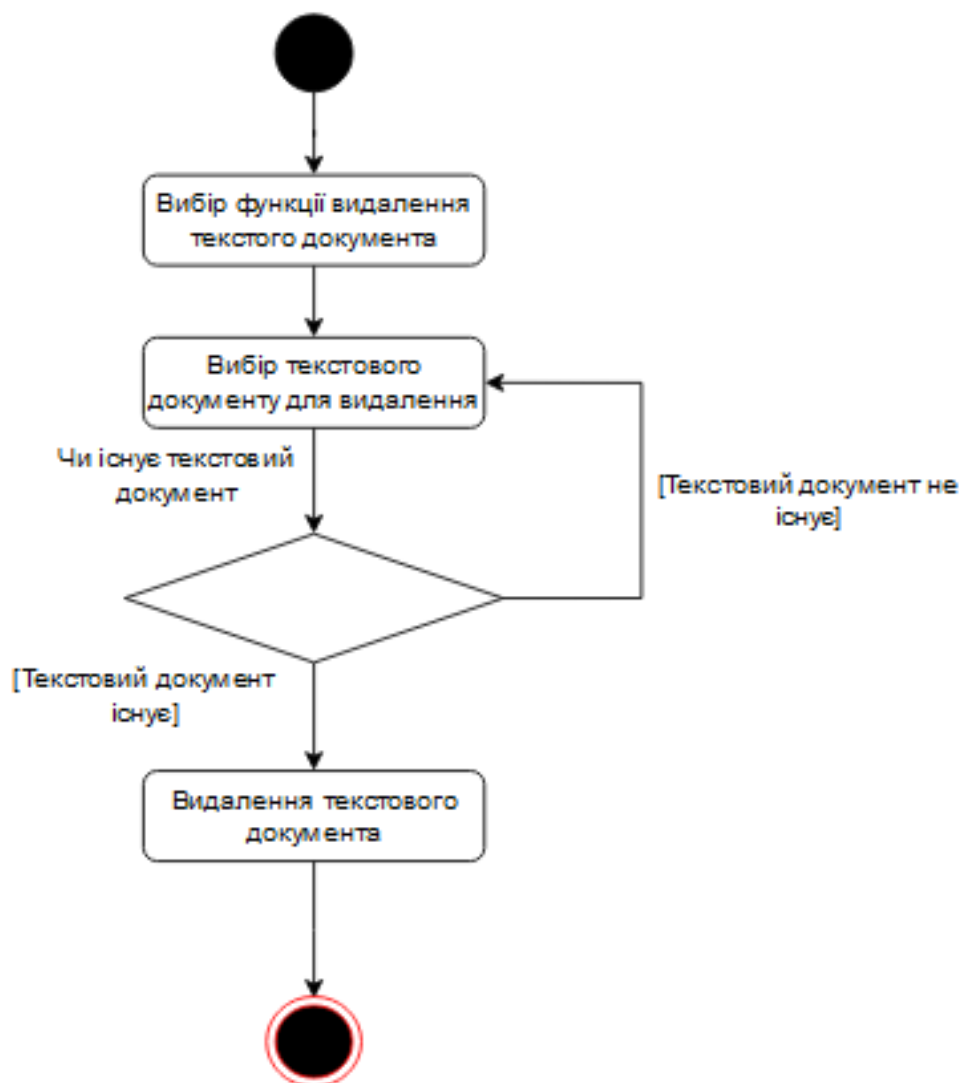


Рисунок 2.4 – Діаграма діяльності сценарію варіанту використання «Видалення текстового документа»

Специфікація варіанту використання «Створення текстового документа»

Назва варіанту використання «Створення текстового документа»

Короткий опис: користувач програми ініціює цей об'єкт та надає можливість створення нового текстового документа.

Потоки подій

Основний потік

Функція варіанту використання набуває своєї дії після того як користувач вибере функцію «Створення текстового документа» у головному меню програми.

1. Користувач повинен заповнити бланк для створюваного текстового документа (назва нового документа, формат нового документа).
2. Система звіряє назву та формат нового текстового документа та уже існуючих.
3. Якщо новий текстовий документ не схож на уже створені, то система створює новий текстовий документ на носії.

Альтернативні потоки

Користувач хоче створити схожий текстовий документ: якщо після перевірки система знаходить схожі текстові документи з новим, то система виводить помилку «Ідентичний текстовий документ».

Спеціальні вимоги: немає спеціальних вимог.

Передумови: вибір функції «Створення текстового документа».

Постумови: створює новий текстовий документ на носії.

Додаткові зауваження: не має додаткових зауважень.

На рисунку 2.5 надано діаграму діяльності варіанту використання «Створення текстового документа»:

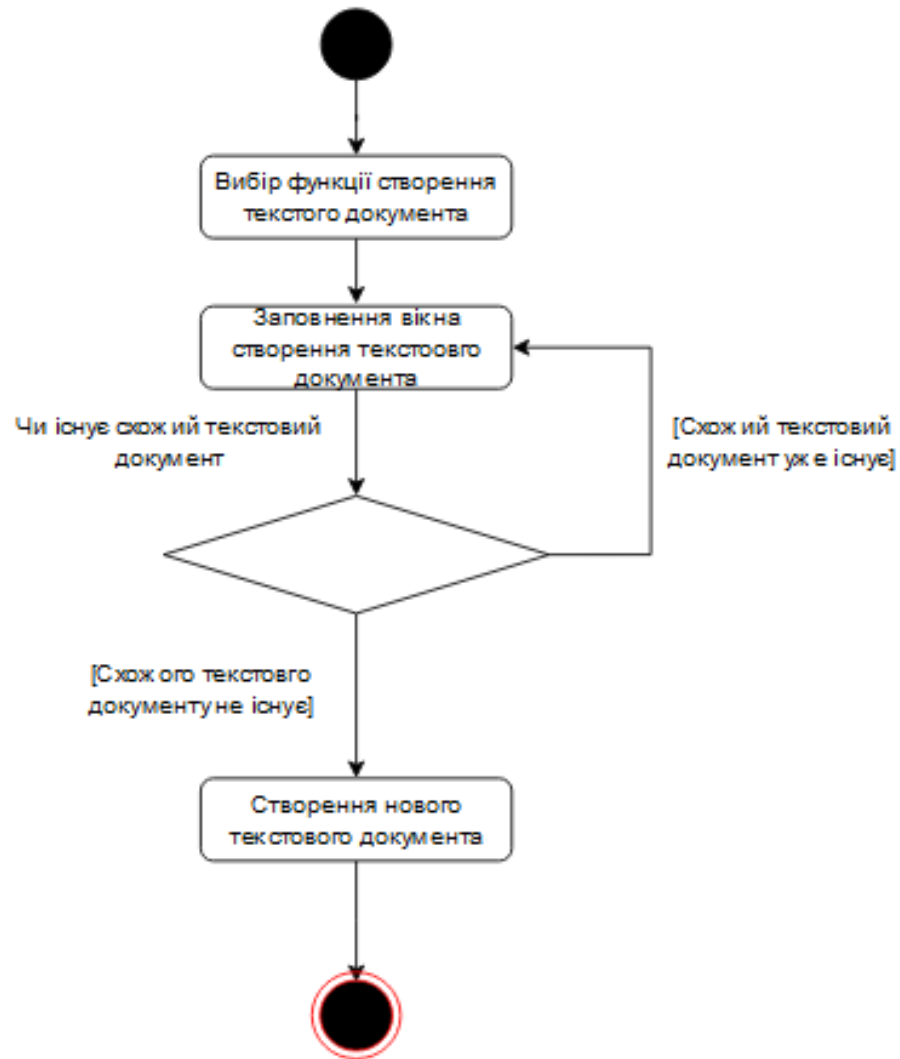


Рисунок 2.5 – Діаграма діяльності сценарію варіанту використання «створення текстового документу»

Специфікація варіанту використання «Сортування текстових документів»

Назва варіанту використання «Сортування текстових документів»

Короткий опис: користувач програми ініціює цей об'єкт та надає можливість вибору типу сортування текстових документів.

Потоки подій

Основний потік

Функція варіанту використання набуває своєї дії після того як користувач вибере функцію «Сортування текстових документів» у головному меню програми.

1. Користувач повинен вибрати тип сортування текстових документів у відповідному бланку типів сортування.

2. Система виводить бланк вибраного типу сортування.

Альтернативні потоки

Користувач хоче сортувати текстові документи: користувач закриває бланк типів сортування.

Спеціальні вимоги: немає спеціальних вимог.

Передумови: Вибір функції «Сортування текстових документів»

Постумови: виводить бланк вибраного типу сортування.

Додаткові зауваження: не має додаткових зауважень.

На рисунку 2.6 надано діаграму діяльності варіанту використання «Сортування текстових документів»:

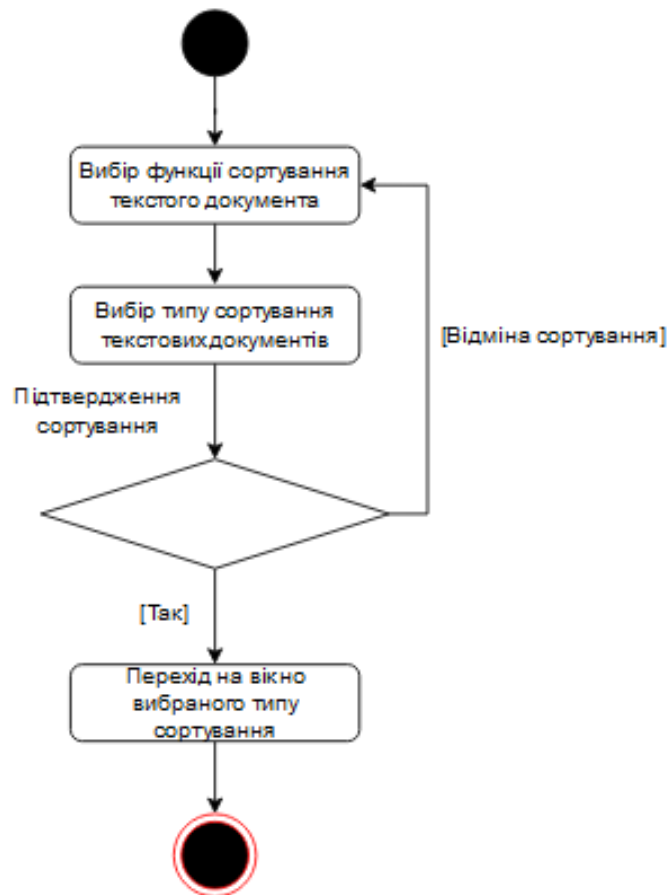


Рисунок 2.6 – Діаграма діяльності сценарію варіанту використання «Сортування текстових документів»

Специфікація варіанту використання «Пошук по назві»

Назва варіанту використання «Пошук по назві»

Короткий опис: користувач програми ініціює цей об'єкт та надає можливість знайти текстовий документі по назві.

Потоки подій

Основний потік

Функція варіанту використання набуває своєї дії після того як користувач вибере функцію «Пошук по назві» у бланку типів сортування.

1. Користувач вводить назву текстового документа.

2. Система виводить текстовий документ, назва якого співпадає з введеною назвою.

Альтернативні потоки

Система не знайшла схожу назву текстового документа: якщо після пошуку система не знаходить співпадаючий текстовий документ, то система виводить помилку «Не має шуканого текстового документа».

Спеціальні вимоги: немає спеціальних вимог.

Передумови: вибір функції «Пошук по назві».

Постумови: виводить шуканий текстовий документ.

Додаткові зауваження: не має додаткових зауважень.

На рисунку 2.7 надано діаграму діяльності варіанту використання «Пошук по назві»:

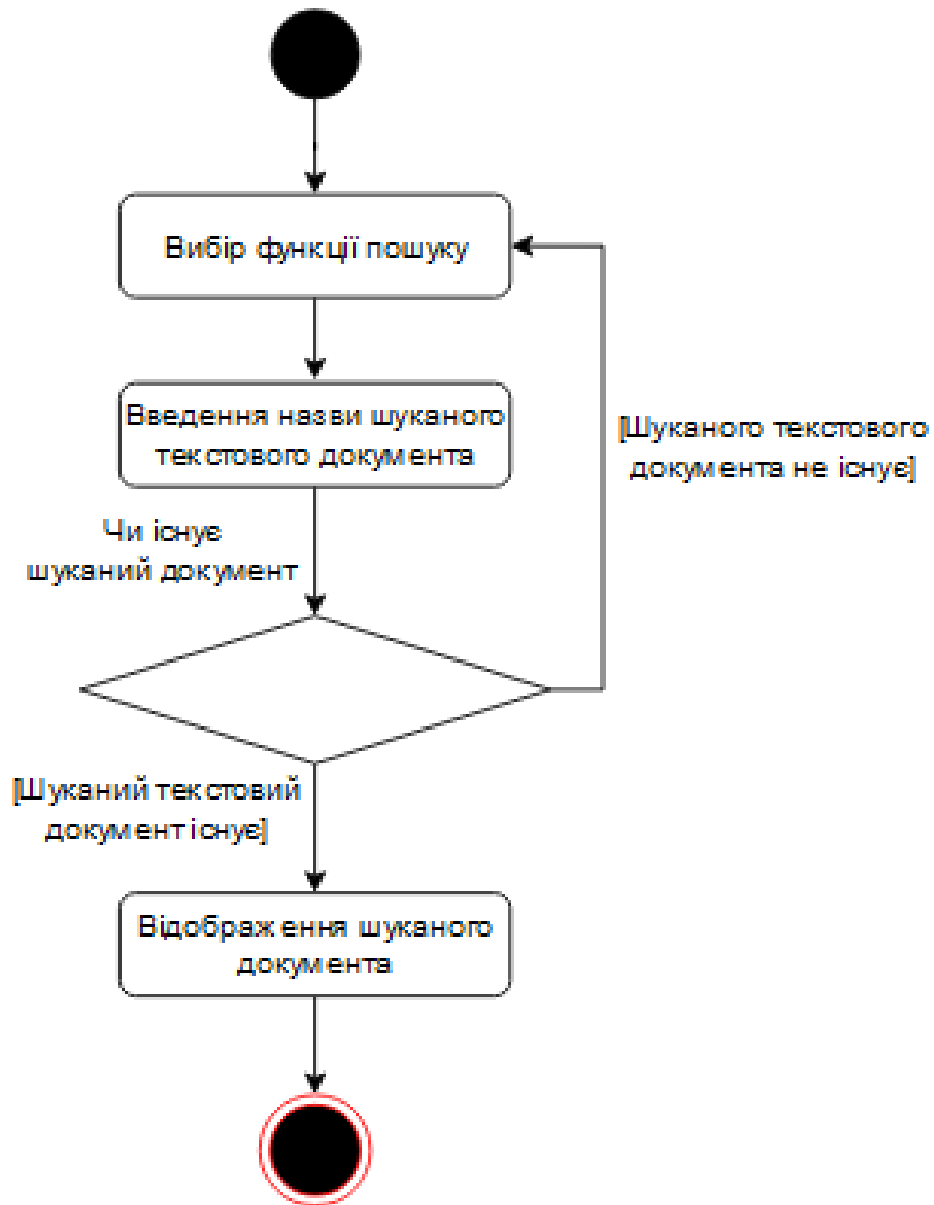


Рисунок 2.7 – Діаграма діяльності сценарію варіанту використання «Пошук по назві»

Таким чином, була створена модель варіантів використання та її специфікації для мобільного застосунку.

2.5 Структура класів

На основі розробленої специфікації варіантів використання та діаграм діяльності розроблено діаграму класів подану на рисунку 2.8.[4]

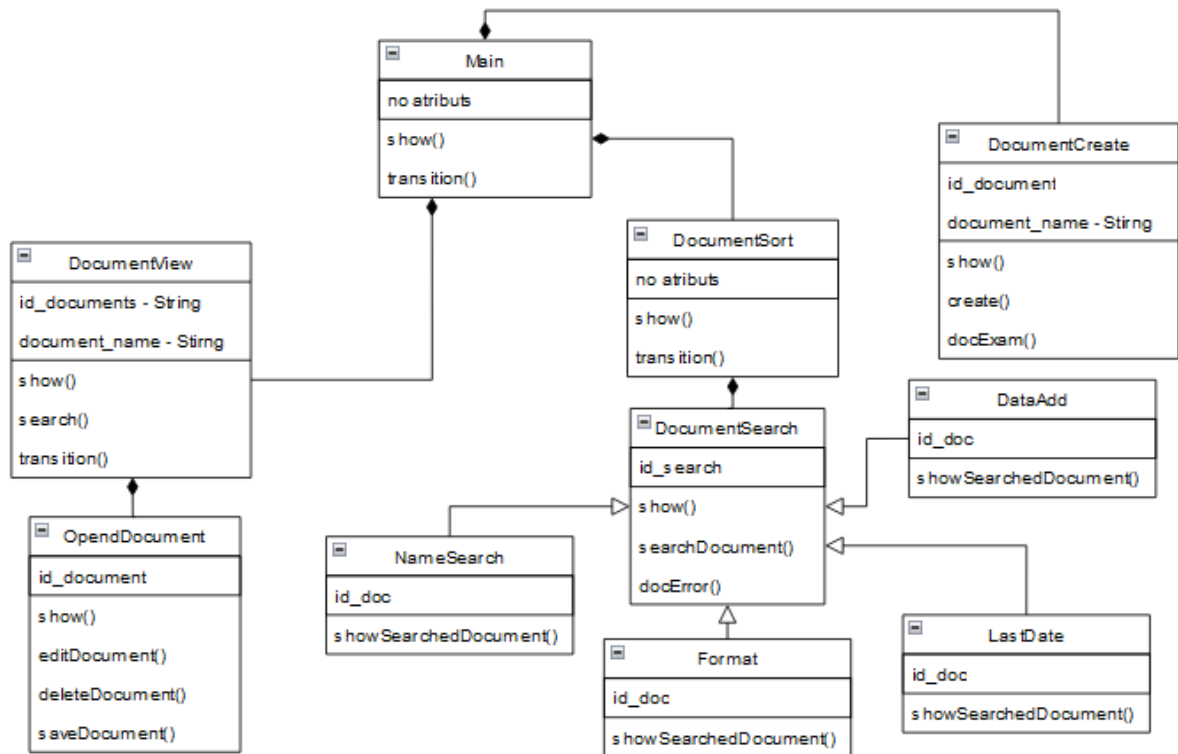


Рисунок 2.8 – Структура класів

Таким чином, була розроблена діаграма класів мобільного застосунку.

2.6 Специфікація класів

Клас «Main»

Відповідальність: Основний клас застосунку.

Функції класу:

show() – функція демонстрування інтерфейсу програми для користувача.

transition() – функція переходу на інтерфейс обраної функції

користувачем.

Клас «DocumentView»

Відповідальність: клас відповідальний за показ усіх доступних текстових документів.

Функції класу:

show() – функція демонстрування інтерфейсу програми для користувача.

search() – функція пошуку усіх доступних текстових документів та відображення їх на інтерфейсі.

transition() – функція переходу на обраний текстовий документ для подальшої взаємодії з ним.

Клас «OpendDocument»

Відповідальність: клас відповідальний за взаємодію з обраним текстовим документом

Функції класу:

show() – функція демонстрування інтерфейсу програми для користувача.

editDocument() – функція редагування текстового документа.

deleteDocument() – функція видалення текстового документа.

saveDocument() – функція зберігання внесених змін до текстового документа.

Клас «DocumentCreate»

Відповідальність: клас відповідальний за створення нових текстових документів

Функції класу:

show() – функція демонстрування інтерфейсу програми для користувача.

create() – функція створення нового текстового документа.

docExam() – функція перевірки створюваного та уже існуючих текстових документів на схожість.

Клас «DocumentSort»

Відповідальність: клас відповідальний за вибір типу сортування текстових документів

Функції класу:

show() – функція демонстрування інтерфейсу програми для користувача.

transition() – функція переходу на інтерфейс обраного типу сортування.

Клас «DocumentSearch»

Відповідальність: клас відповідальний за пошук та сортування текстових документів за обраним типом сортування.

Функції класу:

show() – функція демонстрування інтерфейсу програми для користувача.

searchDocument() – функція пошуку та сортування текстових документів за обраним типом сортування.

docError() – функція виводу помилки, якщо та сталася при виконанні сортування.

Клас «NameSearch»

Відповідальність: клас відповідальний за відображення результату сортування по назві.

Функції класу:

showSearchedDocument() – функція демонстрування результату сортування.

2.7 Діаграми взаємодії варіантів використання

На основі діаграм діяльності побудовані діаграми взаємодії варіантів використання.[\[5\]](#)

На рисунку 2.9 зображена діаграма взаємодії варіантів використання «Редагування текстових документів»

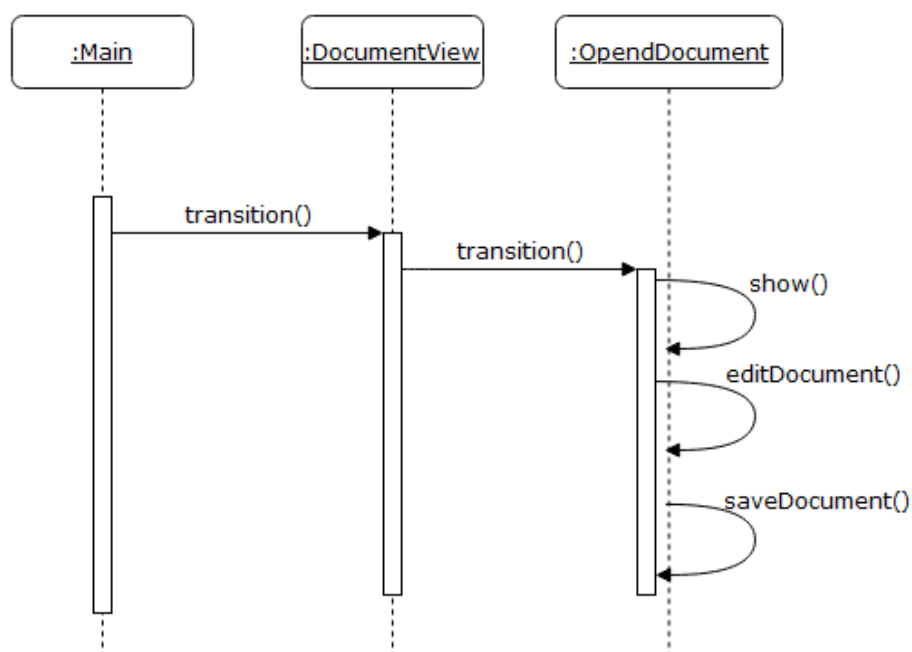


Рисунок 2.9 – Діаграма взаємодії варіантів використання «Редагування текстових документів»

На рисунку 2.10 зображена діаграма взаємодії варіантів використання «Видалення текстових документів»

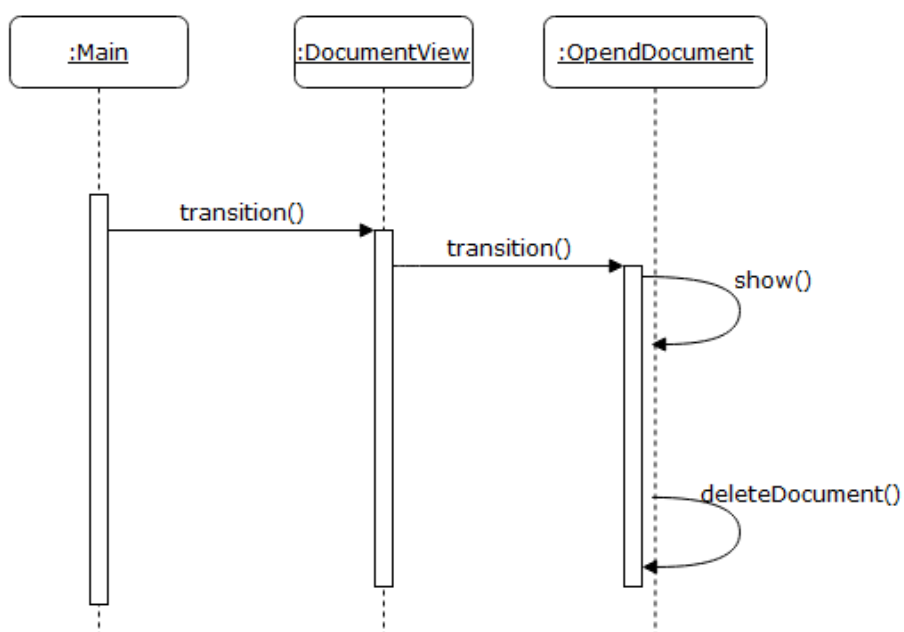


Рисунок 2.10 – Діаграма взаємодії варіантів використання «Видалення текстових документів»

На рисунку 2.11 зображена діаграма взаємодії варіантів використання «Створення текстових документів»

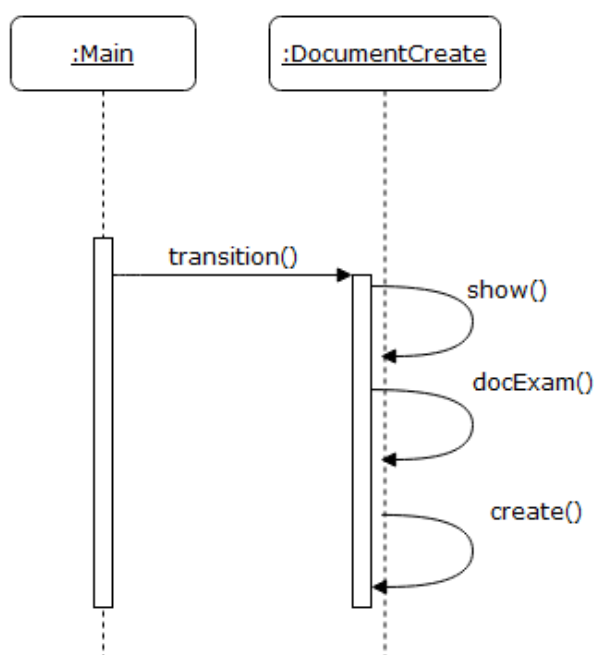


Рисунок 2.11 – Діаграма взаємодії варіантів використання «Створення текстових документів»

На рисунку 2.12 зображена діаграма взаємодії варіантів використання «Пошук по назві»

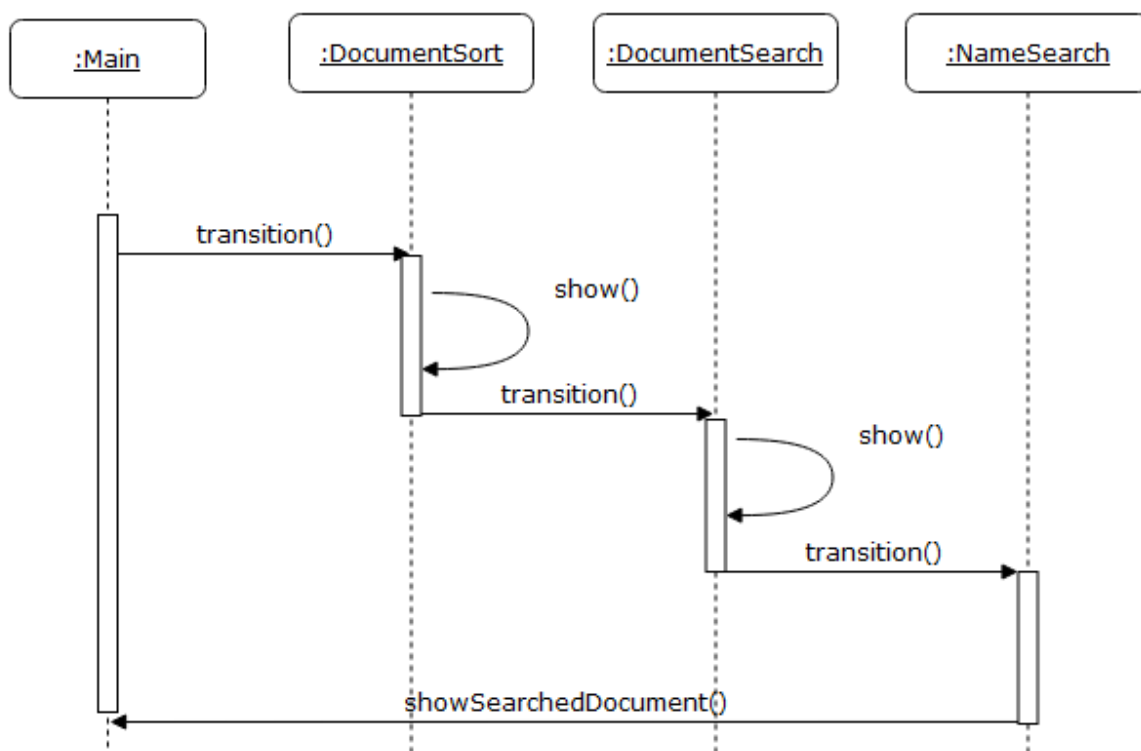


Рисунок 2.12 – Діаграма взаємодії варіантів використання «Пошук по назві»

Таким чином, були розроблені діаграми взаємодії варіантів використання мобільного застосунку.

2.8 Проект користувальницького інтерфейсу

Для більш детального зрозуміння як користувач буде взаємодіяти з мобільним додатком будуть розроблені інтерфейсні форми. Інтерфейсна форма відкритого текстового документа представлен на рисунку 2.13:

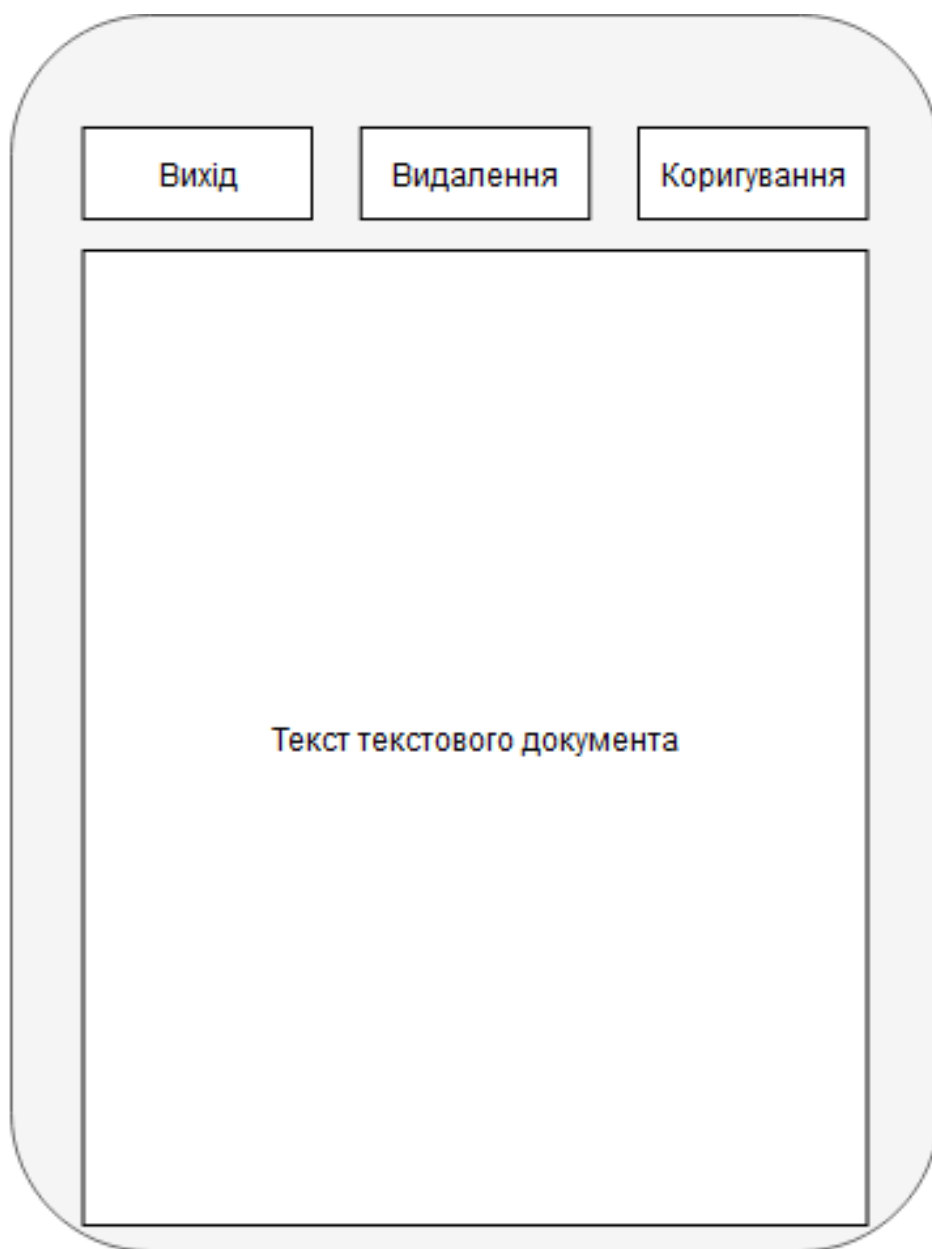
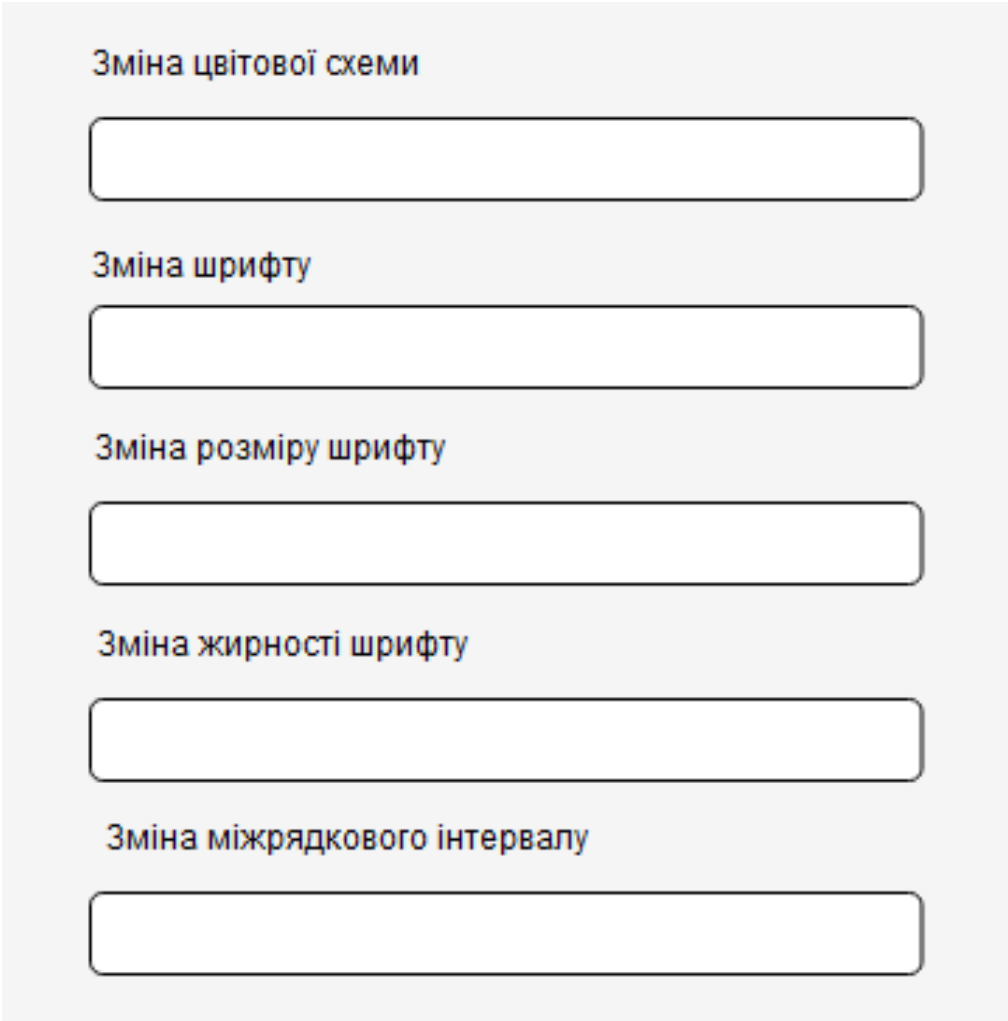


Рисунок 2.12 – Інтерфейсна форма відкритого текстового документа

Інтерфейс відкритого текстового документу має 4 пункти – кнопки «Вихід», «Видалення», «Коригування» та поле, на якому буде відображатися текстовий документ. Кнопка «Вихід» виконує функцію закриття та переходу до основного меню програми. Перед виходом користувачеві буде представлено вибір «зберігати» або «не зберігати» зміни до текстового документа. Кнопка «Видалення» виконує функцію видалення текстового документа з сховища мобільного пристрою. Кнопка «Коригування» визиває діалогове вікно з функціями для зміни відображення текстового документа.

Інтерфейсна форма зміни відображення текстового документа представлено на рисунку 2.13:



Зміна цвітової схеми

Зміна шрифту

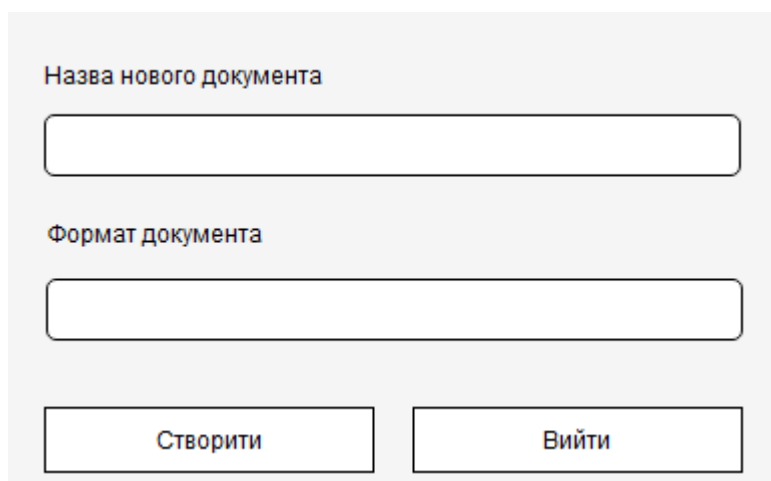
Зміна розміру шрифту

Зміна жирності шрифту

Зміна міжрядкового інтервалу

Рисунок 2.13 – Інтерфейсна форма зміни відображення текстового документа

Функції створення нового текстового документа буде реалізована через інтерфейсну форму створення текстового документа. Інтерфейсна форма створення текстового документа представлено на рисунку 2.14:



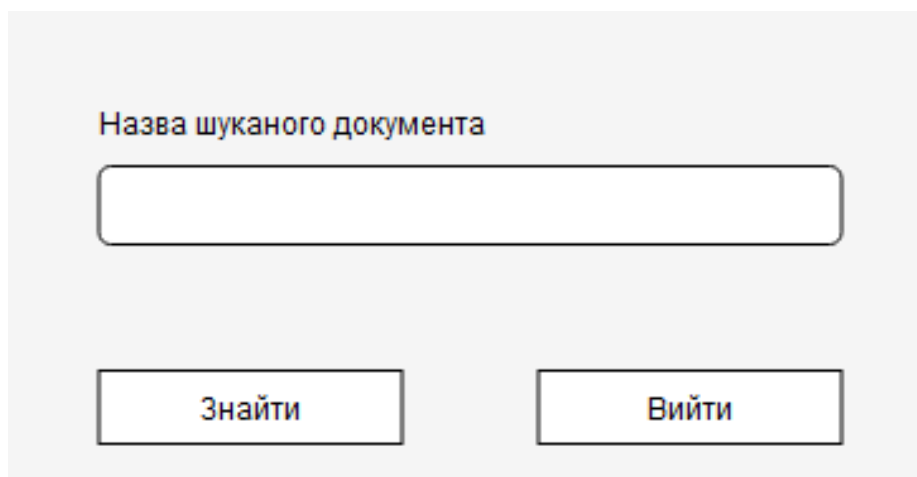
Назва нового документа

Формат документа

Створити Вийти

Рисунок 2.14 – Інтерфейсна форма створення текстового документа

Для функції пошуку текстового документа по назві буде створена інтерфейсна форма пошуку текстового документа по назві. Інтерфейсна форма пошуку текстового документа по назві представлена на рисунку 2.15:



Назва шуканого документа

Знайти Вийти

Рисунок 2.15 – Інтерфейсна форма пошуку текстового документа по назві

Для функції сортування текстових документів треба створити інтерфейсну форму сортування текстових документів. Інтерфейсна форма сортування текстових документів представлена на рисунку 2.16:

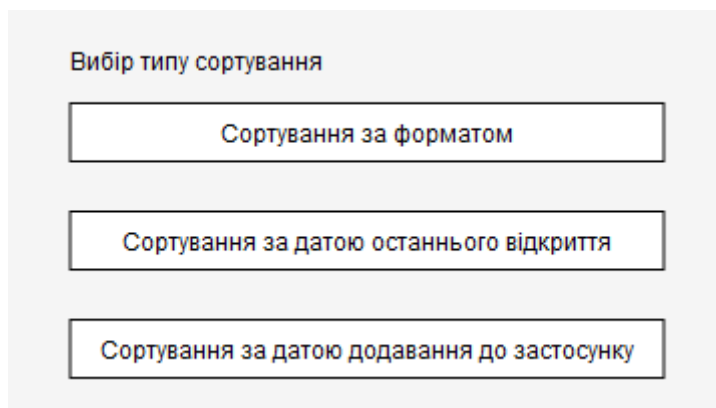


Рисунок 2.16 – . Інтерфейсна форма сортування текстових документів
Таким чином, були створені інтерфейсні форми мобільного доадатку.

2.9 Тестування основних функцій мобільного додатку

Для тестування функцій використано методи « чорної скриньки». Нижче наведено тестові приклади за методом еквівалентного розбиття.

Тестування функції «Створення нового документу»

Клас еквівалентності наведено у таблиці 2.1

Таблиця 2.1 - Класи еквівалентності функції «Створення нового документа»

Вхідні умови	Правильний клас еквівалентності	Неправильний клас еквівалентності
Назва документа	1. Назва кирилицею, латиницею та числа	2. Символи відмінні від кирилиці, латиниці та чисел
Формат документа	3. Обрані формати документів	4. Будь-які інші формати документів

Тести з правильних класів еквівалентності наведено в таблиці 2.2

Таблиця 2.2 – Тести з правильних класів еквівалентності

№ Тесту	№ обраного класу еквівалентності	Вхідні данні	Результати
1	1	РоботаТимошенко	В результаті тестування створюється новий текстовий документ
	3	docx	

Тести з неправильних класів еквівалентності наведено в таблиці 2.3

Таблиця 2.3 – Тести з неправильних класів еквівалентності

№ Тесту	№ обраного класу еквівалентності	Вхідні данні	Результати
1	2	Робота+Тимошенко	В результаті тестування з'являється помилка «Хибна назва текстового документа»
2	4	png	В результаті тестування з'являється помилка «Хибний формат текстового документа»

Тестування функції «Пошук текстового документа»

Клас еквівалентності наведено у таблиці 2.4

Таблиця 2.4 - Класи еквівалентності функції «Пошук текстового документа»

Вхідні умови	Правильний клас еквівалентності	Неправильний клас еквівалентності
Назва документа	1. Назва кирилицею, латиницею та числа	2. Символи відмінні від кирилиці, латиниці та чисел

Тести з правильних класів еквівалентності наведено в таблиці 2.5

Таблиця 2.5 – Тести з правильних класів еквівалентності

№ Тесту	№ обраного класу еквівалентності	Вхідні данні	Результати
1	1	РоботаТимошенко	В результаті тестування шукається текстовий документ

Тести з неправильних класів еквівалентності наведено в таблиці 2.6

Таблиця 2.6 – Тести з неправильних класів еквівалентності

№ Тесту	№ обраного класу еквівалентності	Вхідні данні	Результати
1	2	Робота\Тимошенко	В результаті тестування з'являється помилка «Хибна назва шуканого документа»

Тестування функції «Сортування за форматом»

Клас еквівалентності наведено у таблиці 2.7

Таблиця 2.7 - Класи еквівалентності функції «Сортування за форматом»

Вхідні умови	Правильний клас еквівалентності	Неправильний клас еквівалентності
Формат документа	1. Обрані формати документів	2. Будь-які інші формати документів

Тести з правильних класів еквівалентності наведено в таблиці 2.8

Таблиця 2.8 – Тести з правильних класів еквівалентності

№ Тесту	№ обраного класу еквівалентності	Вхідні данні	Результати
1	1	pdf	В результаті тестування сортуються документи відповідного формату

Тести з неправильних класів еквівалентності наведено в таблиці 2.9

Таблиця 2.9 – Тести з неправильних класів еквівалентності

№ Тесту	№ обраного класу еквівалентності	Вхідні данні	Результати
1	2	jpg	В результаті тестування з'являється помилка «Невірний формат»

Завдяки методам «чорної скриньки» було проведено тестування функцій мобільного застосунку.

2.10 Випробування програмного забезпечення

Фахівцем з експлуатації виконується перевірка з виконання вимог до документації мобільного застосунку. Перевірка здійснюється шляхом порівняння переліку програмної документації (технічне завдання, текст програми, опис програми, програма та методика випробувань, інструкція користувача) з комплектність програмної документації, представленої розробником.

Випробування вважається успішним у разі відповідності між вимогами до функціональних характеристик і послідовністю виконаних дій

Згідно з вимог до функціональної характеристик застосунку виконується перевірка працездатності продукту

В процесі тестування була перевірена функціональність за темою «Розробка мобільного застосунку для роботи з електронними документами у середовищі ОС Android»

У таблиці 2.10 вказано перелік випробувань основних функціональних можливостей.

Таблиця 2.10 – Методика випробування

№	Дія	Очікуваний результат	Результат перевірки	Зауваження
1	2	3	4	5
1	Випробування функції редагування текстового документу	Застосунок повинен змінити текстовий документ і зберегти	Виконано	
2	Випробування функції видалення текстового документа	Застосунок повинен видалити текстовий документ	Виконано	
3	Випробування функції створення нового текстового документа	Застосунок повинен створити новий текстовий документ	Виконано	
4	Випробування функції пошуку по назві текстового документа	Застосунок повинен перевірити схожість назв текстових документів і знайти схожі	Виконано	

Продовження таблиці 2.10

1	2	3	4	5
5	Випробування функції сортування по формату	Застосунок повинен знайти текстові документи з відповідного формату	Виконано	
6	Випробування функції сортування за датою додавання до застосунку	Застосунок повинен відсортувати текстові документи за датою додавання їх до застосунку	Виконано	
7	Випробування функції сортування за датою останнього відкриття	Застосунок повинен відсортувати текстові документи за датою останнього відкриття	Виконано	

Проведено випробування мобільного застосунку для роботи з електронними документами у середовищі ОС Android.

3. Результати розробки мобільного додатку для роботи з електронними документами у середовищі ОС Android

Мобільний застосунок для роботи з електронними документами у середовищі ОС Android є результатом кваліфікаційної роботи. Для досягнення поставленої задачі, було виконано такі дії: аналіз поставленої задачі, розгляд готових застосунків, формування мети та вимог до мобільного застосунку.

При розробці мобільного застосунку було обрано архітектурний стиль, побудована модель варіантів використання, модель класів, діаграма взаємодії варіантів використання для основних функцій застосунку.

Для перевірки правильності роботи мобільного застосунку були проведені тести основних функцій за методологією «чорної скриньки» та випробування основних функцій програми.

У результаті розроблено мобільний додаток, що вирішує наступні функції:

- редагування текстових документів;
- видалення текстових документів;
- створення нових текстових документів;
- пошук по назві текстових документів;
- сортування по формату;
- сортування по даті останнього відкриття;
- сортування по даті додавання до додатку;

Деякий функціонал розробленого мобільного застосунку представлений на рисунках 3.1-3.4



Рисунок 3.1 – Інтерфейс перегляду документів

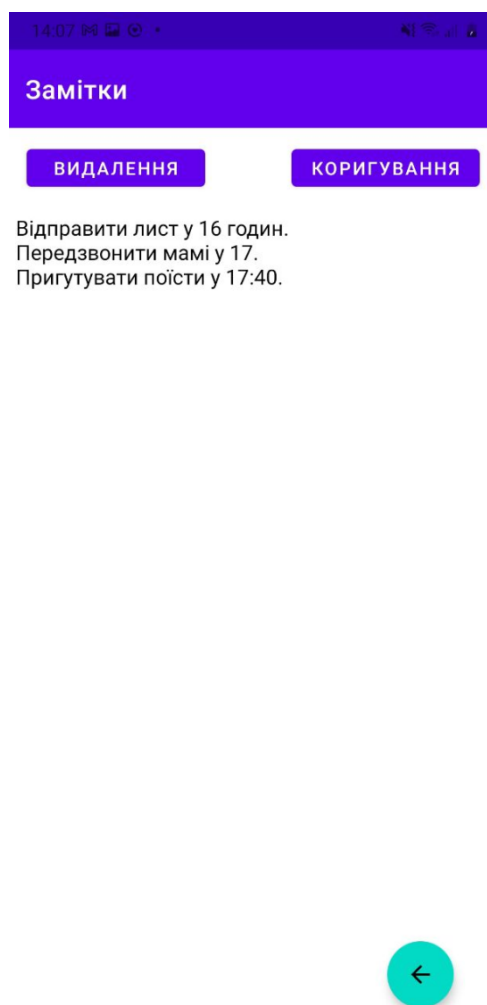


Рисунок 3.2 – Інтерфейс відкритого текстового документа

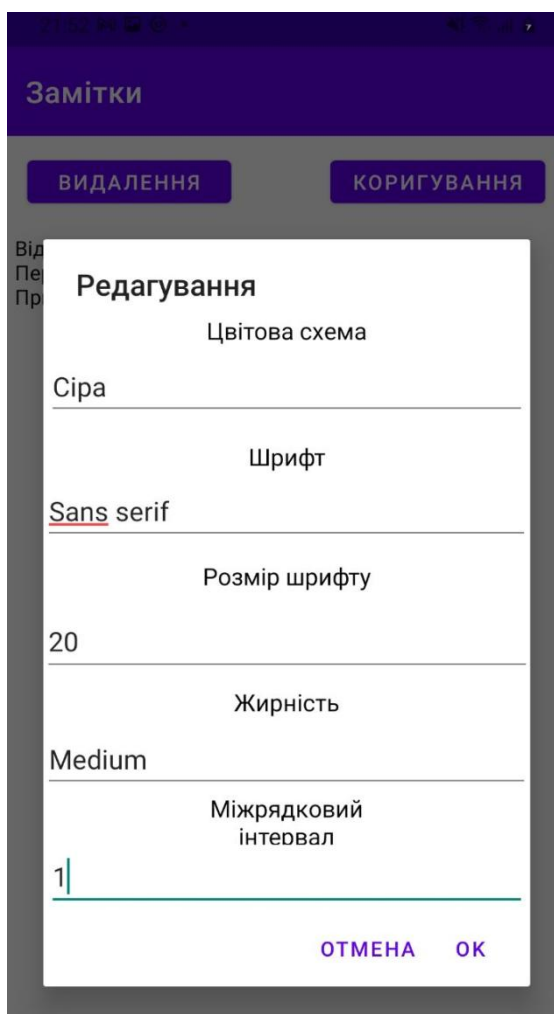


Рисунок 3.3 – Інтерфейс редагування текстового документа

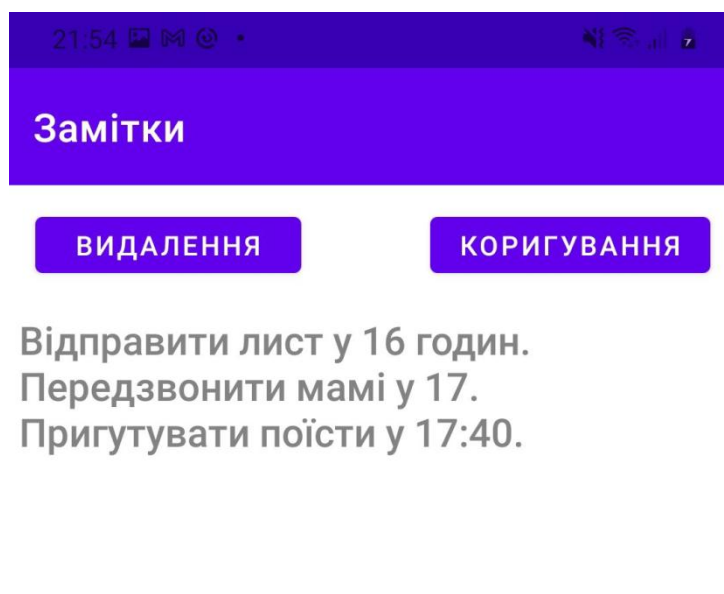


Рисунок 3.4 – Редагований текстовий документ

Більш детально результат роботи надано у додатку Б

Для розробки мобільного застосунку були використані наступні інструменти:

- Java – багатоплатформна об'єктно-орієнтована мова програмування.
- Android Studio – інтегроване середовище для розробки для платформи

Android.

Також у кваліфікаційній роботі представлені:

- технічне завдання (Додаток А);
- інструкція користувача (Додаток Б)
- опис програмного забезпечення (Додаток В);
- вихідні тексти програмних модулів (Додаток Г);
- програма та методика випробувань мобільного застосунку (Додаток Д).

4. Охорона праці

4.1 Основне положення

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Головною метою охорони праці - створення на кожному робочому місці безпечних умов праці, безпечної експлуатації обладнання, зменшення або повна нейтралізація дії шкідливих і небезпечних виробничих факторів на організм людини і, як наслідок, зниження виробничого травматизму та професійних захворювань.

Державна політика в галузі охорони праці базується на принципах:

- пріоритету життя і здоров'я працівників, повної відповідальності роботодавця за створення належних, безпечних і здорових умов праці;
- підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництв, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці;
- комплексного розв'язання завдань охорони праці на основі загальнодержавної, галузевих, регіональних програм з цього питання та з урахуванням інших напрямів економічної і соціальної політики, досягнень в галузі науки і техніки та охорони довкілля;
- соціального захисту працівників, повного відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;
- використання економічних методів управління охороною праці, участі держави у фінансуванні заходів щодо охорони праці, залучення

доброчинних внесків та інших надходжень на ці цілі, отримання яких не суперечить законодавству;[8]

4.2 Заходи щодо безпечної праці при роботі з мобільним пристроєм

2.2.1 Користування телефоном

Користуйтеся телефоном обережно. Акумулятор чи сам пристрій можна пошкодити, якщо його розібрати, кинути, зігнути, проколоти, підпалити чи розбити. Не користуйтеся пристроєм із тріснутим екраном. Пошкоджений пристрій може спричинити перенагрівання акумулятора або травмування. Уникайте контакту телефона з рідинами, оскільки це може спричинити коротке замикання й перегрівання. Якщо телефон намокне, не намагайтеся висушити його за допомогою зовнішнього джерела тепла.

Телефон найкраще працює за температури від 0° до 35 °С. Його потрібно зберігати за температури від -20° до 45 °С. Не залишайте телефон у місцях, де температура може перевищувати 45 °С, зокрема на панелі приладів автомобіля чи поряд із вентиляційними отворами опалювальних приладів, оскільки це може пошкодити телефон або спричинити перегрівання акумулятора чи навіть пожежу. Тримайте телефон подалі від джерел тепла та прямих сонячних променів. Якщо пристрій перегріється, відключіть його від джерела живлення (якщо його під'єднано), покладіть у холодніше місце та не використовуйте, доки він не охолоне.

Якщо телефон не працює належним чином або має пошкодження, не користуйтеся ним і зв'яжіться зі службою обслуговування клієнтів.[9]

2.2.2 Заряджання телефону

Переконайтеся, що адаптер живлення й телефон добре провітрюються під час роботи або заряджання. Використання пошкоджених кабелів або адаптера живлення чи заряджання у вологому приміщенні може спричинити займання, ураження електричним струмом, травмування або пошкодження телефона чи іншого майна. Заряджаючи телефон, переконайтеся, що зарядний пристрій увімкнено в розетку поряд із телефоном і до нього легко

отримати доступ. Під час заряджання тримайте пристрій подалі від прямих сонячних променів.

Заряджайте телефон лише за допомогою адаптера та кабелю живлення, які постачаються в комплекті пристрою. Використання несумісних аксесуарів може спричинити займання, ураження електричним струмом, травмування або пошкодження телефону чи аксесуарів.

Відключаючи адаптер від мережі живлення, не можна тягнути за кабель. Не скручуйте та не стискайте кабель USB, а також не вставляйте конектор у порт силою.

2.2.3 Тривалий вплив тепла

Під час звичайного використання телефон і зарядний пристрій нагріваються, але не перевищують прийнятних температурних обмежень і стандартів. Уникайте тривалого прямого та непрямого контакту зі шкірою під час використання чи заряджання пристрою, оскільки тривалий вплив гарячих поверхонь на шкіру може спричинити дискомфорт або опіки. Не спіть на пристрої чи адаптері живлення та не накривайте їх ковдрою або подушкою. Пам'ятайте про це, якщо через фізичний стан можете не одразу відчувати вплив тепла на шкіру.

Пристрій більше нагрівається під час заряджання, коли ввімкнено ліхтар або певні функції (як-от запис відео чи в режимі віртуальної реальності). У цих режимах користуйтеся пристроєм обережніше.

2.2.4 Захист органів слуху

Тривале прослуховування гучних звуків (зокрема музики) може спричинити втрату слуху. Щоб не пошкодити слух, не вмикайте звук надто гучно на тривалий час. Через тривалий вплив гучного аудіоконтенту та фонового шуму інші гучні звуки можуть здаватися тихішими, ніж насправді. Перш ніж користуватися навушниками, перевірте рівень гучності.

Обмеження щодо умов використання

Щоб запобігти пошкодженню деталей і внутрішніх схем телефона, не використовуйте й не зберігайте його та аксесуари в запилених, задимлених,

вологих і забруднених місцях, а також поблизу сильних магнітних полів. Тримайте його подалі від джерел тепла та прямих сонячних променів. Не залишайте телефон у салоні транспортного засобу або в місцях, де температура може перевищувати 45 °С, зокрема на панелі приладів автомобіля, підвіконні, поряд із вентиляційними отворами опалювальних приладів чи за склом, що перебуває під прямими сонячними променями або сильним ультрафіолетовим випромінюванням упродовж тривалого часу. Це може пошкодити телефон, спричинити перегрівання акумулятора або навіть пожежу чи вибух.

Не слід використовувати, зберігати чи транспортувати телефон там, де зберігаються легкозаймисті або вибухові речовини. Не користуйтеся телефоном під час проведення підривних робіт, а також у потенційно вибухонебезпечних зонах, як-от на заправних станціях, складах пального, нижній палубі кораблів і станціях транспортування та зберігання пального й хімічних речовин, де повітря містить хімічні речовини та частки (наприклад, промисловий пил чи дрібні частинки металу). Іскри в таких зонах можуть призвести до травм або навіть смерті в результаті вибуху чи займання.

2.2.5 Відволікання

Пристрій може відволікати вас під час виконання певних дій, призводячи до ситуацій, небезпечних для вас чи інших осіб. Щоб знизити ризик аварій, не користуйтеся пристроєм за кермом, під час їзди на велосипеді, роботи з обладнанням або в інших випадках, коли це може мати серйозні наслідки. Дотримуйтеся місцевих законів щодо використання мобільних пристроїв, навушників і захисного обладнання.

Вплив радіочастот на роботу інших пристроїв

Не використовуйте бездротові технології (наприклад, мобільний Інтернет або Wi-Fi), коли це заборонено. Пристрій відповідає вимогам щодо радіочастотного випромінювання, однак бездротові пристрої можуть негативно впливати на інше електронне обладнання. Наприклад, під час польоту або перед посадкою в літак користуйтеся бездротовим пристроєм

лише відповідно до вказівок авіакомпанії. Використання бездротового пристрою в літаку може порушити роботу бездротових мереж і становити загрозу для керування літаком, а також вважатися незаконним. Ви можете користуватися пристроєм у режимі польоту.

2.2.6 Магнітні поля

Не кладіть носії, які містять магніти або є чутливими до магнетизму (як-от кредитні картки, банківські картки, аудіо-, відеокасети чи магнітні пристрої пам'яті), поблизу цього продукту, його зарядного пристрою чи кабелю. Це може спричинити втрату інформації, що на них зберігається. Чутливі до магнітів пристрої з даними потрібно тримати щонайменше за 5 см від телефону.

Висновки

У результаті виконання кваліфікаційної роботи було розроблено мобільний застосунок для роботи з електронними документами у середовищі ОС Android, який дозволяє користувачеві працювати з електронними документами.

У даній кваліфікаційній роботі була розв'язана практична задача інженерії програмного забезпечення, яка характеризується комплексністю та невизначеністю умов, а саме мобільний застосунок для роботи з електронними документами у середовищі ОС Android.

Під час роботи над кваліфікаційною роботою було пройдено етапи:

– детального дослідження предметної галузі, аналізу готових рішень та постановки задачі та обґрунтування створення мобільного застосунку;

- розробка мобільного додатку складається з моделі варіантів використання, моделі класів, діаграм взаємодії та випробувань програмного забезпечення;

- результатів розробки програмного забезпечення

- дослідження розділу з охорони праці;

Самостійно опрацьовано основні компоненти додатків під ОС Android, життєві принципи роботи, створення інтерфейсу, робота з файлами, зберігання та динамічне відтворення даних на екрані додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Android studio: переваги та особливості URL: <https://qagroup.com.ua/publications/android-studio-perevagy-ta-osoblyvosti/> (дата звернення: 13.05.2023).
2. 10 причин почему ваш проект должен использовать Dojo Toolkit URL: <https://habr.com/ru/post/189576/> (дата звернення: 17.05.2023).
3. Створення концептуальної моделі. URL: http://ni.biz.ua/3/3_6/3_65278_tema--sozдание-kontseptualnoy-modeli.html (дата звернення: 01.06.2023).
4. Діаграма класів. URL: <https://studopedia.org/10-134736.html> (дата звернення: 02.06.2021).
5. Діаграми послідовності. URL: <http://um.co.ua/1/1-5/1-53877.html> (дата звернення: 05.10.2021).
6. Java. URL: <https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:Java> (дата звернення: 07.06.2023).
7. Литвин В.В., Пасічник В.В., Шаховська Н.Б. Проектування інформаційних систем. – Львів: «Магнолія 2006», 2021. –380 с.
8. Про охорону праці URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення: 07.06.2023).
9. Посібник з техніки безпеки та нормативних вимог URL: <https://support.google.com/pixelphone/answer/7486047?hl=uk> (дата звернення: 13.06.2023).

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ РОБОТИ З ЕЛЕКТРОНИМИ ДОКУМЕНТАМИ У СЕРЕДОВИЩІ ОС ANDROID

Вступ

Повна назва програмного продукту: IBook

Сфера застосування: мобільний телефон користувача

1. Підстави для розробки

Підставою для розробки даного програмного забезпечення є наказ на затвердження тем кваліфікаційних робіт бакалаврів зі спеціальності 121 «Інженерія програмного забезпечення» в Національному університеті кораблебудування ім. адмірала Макарова.

2. Призначення розробки

2.1 Функціональне призначення

Для відвідувача сайту програма надає наступні можливості: перегляду товарів за категоріями з можливістю отримання детальної інформації про товар на відповідній сторінці та оформлення замовлення товару.

Для користувача застосунку додаток надає наступні можливості: відкриття, редагування, видалення та створення нових текстових документів, сортування текстових документів по назві, по формату, по даті останнього відкриття і по даті додавання до додатку.

2.2 Експлуатаційне призначення

Програма має експлуатуватися на мобільному пристрої користувача.

3. Вимоги до програми або програмного виробу

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Функції користувача

Відкриття текстового документа – вміння відкривати текстовий документ для подальшої роботи з ними.

Редагування текстових документів – вміння вносити корективи до відкритого текстового документа. До редагування відноситься:

Додавання тексту – додавання певної кількості тексту до текстового документа.

Видалення тексту – видалення певної кількості тексту з текстового документа.

- Зміна розміру шрифту.
- Зміна жирності шрифту.
- Зміна стилю шрифту.
- Зміна кольорових схеми текстового документа – зміна кольору шрифту і заднього плану.
- Зміна міжрядкового інтервалу.

Зберігання тестового документа – вміння зберігати внесені корективи у новий або в уже відкритий текстові документи.

Видалення текстового документа – вміння віддаляти текстовий документ з пам'яті мобільного пристрою.

Створення нового текстового документа – вміння створювати новий текстовий документ у пам'яті мобільного пристрою для подальшої роботи з ним.

Сортування текстових документів – для більш зрозумілого використання користувачем мобільного додатку створена можливість орієнтуватися в великому обсязі текстових документів завдяки сортуванню по:

- Назва текстового документу – для більш легкої орієнтації по текстовим документам створена функція запиту по назві текстових документів.
- Формату документу – мобільний додаток має можливість відкривати текстові документи у найбільш популярних текстових форматах – docx, txt, pdf і т.д.. Тому є можливість сортування по текстовим форматам.
- Дата останнього відкриття документу – для того щоб користувач міг більш легко відкривати текстові документи які він нещодавно або давно відкривав, створена функція сортування по даті останнього відкриття.
- Дата додавання документу - для того щоб користувач міг більш легко відкривати текстові документи які він нещодавно або давно додав до додатку, створена функція сортування по даті додавання документу.

3.1.2 Вимоги до організації вхідних та вихідних

Текстові документи збережені на мобільному пристрої користувача. Для того, щоб додаток міг працювати з пам'яттю мобільного пристрою, користувач потрібен надати йому на це дозволення.

3.2 Вимоги до надійності

Надійне (стійке) функціонування програми має бути забезпечене виконанням сукупності організаційно-технічних заходів, а також захист від некоректних дій користувача

Відмови через некоректні дії користувача можливі унаслідок закриття програми, в цьому випадку користувачеві показується додаткове вікно, чи впевнений він що хоче закрити програму;

3.3 Умови експлуатації

3.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні відповідати вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

3.3.2 Вимоги до чисельності та кваліфікації користувачів

Користувач повинен мати базові навички роботи з мобільним пристроєм.

3.4 Вимоги до складу і параметрів технічних засобів

Мобільний пристрій відповідати наступному складу технічних засобів з мінімальними параметрами:

- Операційна система: Android версії 6.0 або більш пізніші версії
- Пам'ять: 80 МБ для роботи з додатком

3.5 Спеціальні вимоги

Застосунко повинен забезпечувати взаємодію з користувачем за допомогою графічного інтерфейсу користувача, розробленого відповідно до рекомендацій компанії-виробника операційної системи.

4. Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- програму та методику випробувань;
- керівництво користувача;
- опис програми;
- код програми.

5. Техніко-економічні показники

У даній кваліфікаційній роботі техніко-економічні показники не розраховуються.

6. Порядок контролю та приймання

Для контролю приймання має бути надано опис застосунку, а також програму і методику випробувань.

Якщо застосунок не пройшов випробування, виконавець зобов'язаний виправити помилки і недоліки в строк, не більш ніж за 1 тиждень від дня випробування.

7. Стадії та етапи розробки

Стадії та етапи розробки представлені у таблиці А.1

Таблиця А.1 – Стадії та етапи розробки програмного забезпечення

Стадії розробки	Етапи роботи	Термін виконання	
1 Аналіз практичної задачі	1.1 Аналіз практичної задачі		
	1.2 Аналіз сучасних застосунків		
	1.3 Постанова задачі		
	1.4 Обґрунтування створення застосунку		
	1.5 Аналіз вимог		
2 Розробка мобільного застосунку	2.1 Вибір методології реалізації		
	2.2 Визначення архітектури застосунку		
	2.3 Модель варіантів використання		
	2.4 Структура та специфікація класів		
	2.5 Діаграми взаємодії		

Продовження таблиці А.1:

	2.6 Розробка інтерфейсних форм		
	2.7 Тестування основних функцій		
	2.8 Випробування програмного забезпечення		
3 Результати розробки	3 Результати розробки мобільного застосунку		

ДОДАТОК Б – Інструкція користувача

Для роботи з мобільним застосунком потрібно встановити на мобільний пристрій застосунок «IBook».

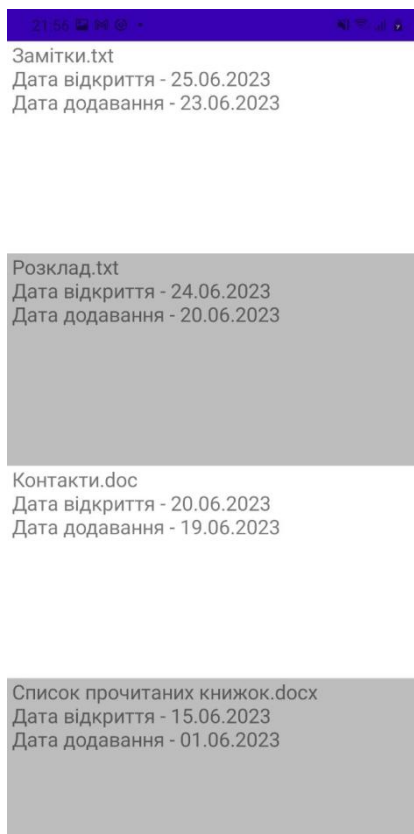
Запуск застосунку здійснюється за допомогою натискання іконки на робочому столі мобільного пристрою.

Після запуску перед користувачем з'являється головне меню, зображена на рисунку Б.1



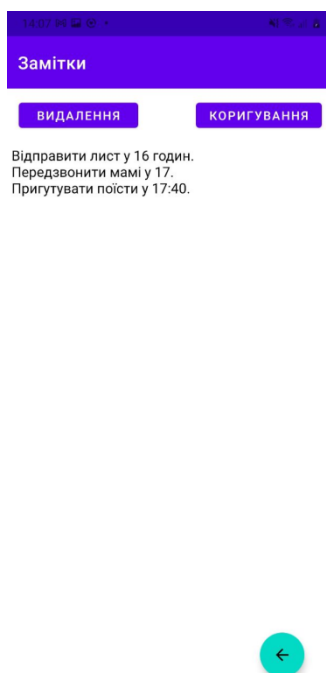
Б.1 – Форма головного меню

Після натискання функції «Перегляд існуючих документів» здійснюється перехід на вікно списку існуючих документів, яке надано на рисунку Б.2:



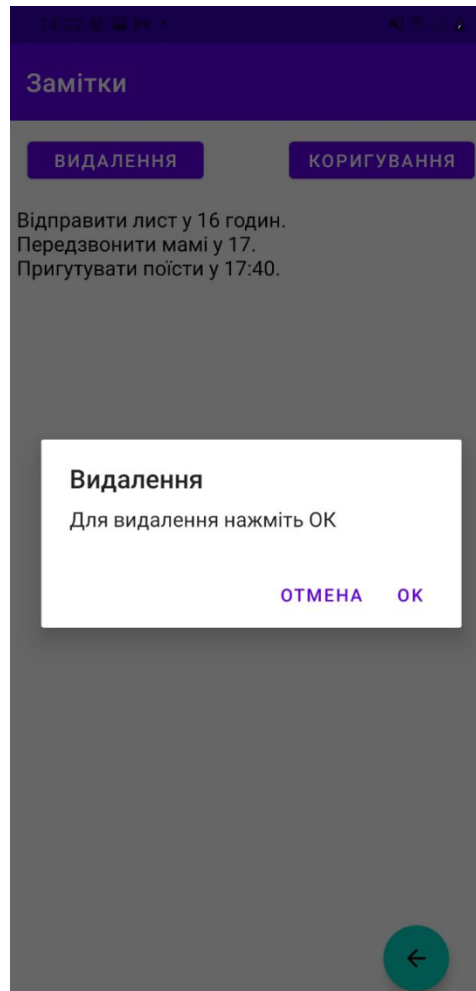
Б.2 – Форма списку існуючих документів

Після вибору документу здійснюється перехід на вікно редагування текстового документа, яке надано на рисунку Б.3:



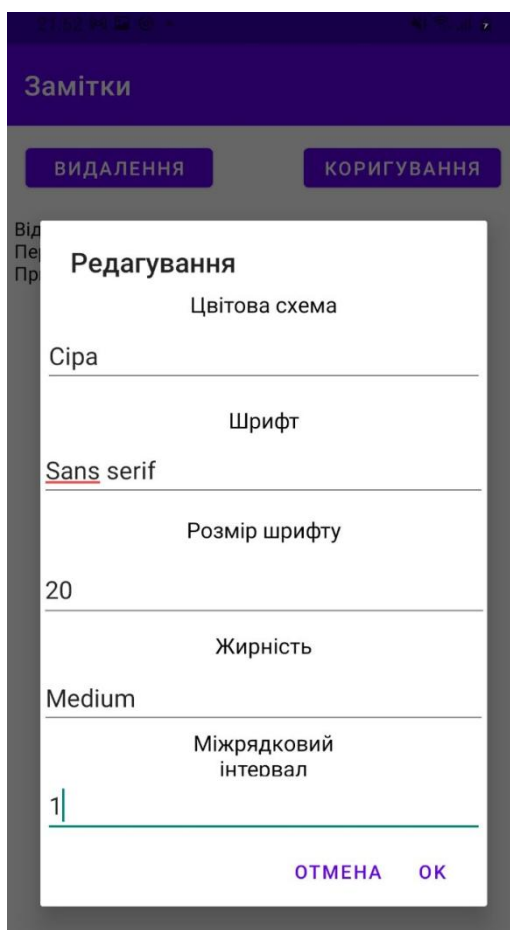
Б.3 – Форма редагування текстового документа

При натисканні кнопки «Видалення» визивається вікно підтвердження видалення текстового документа, яке надано на рисунку Б.4:



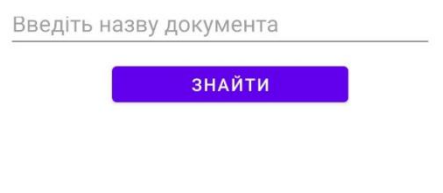
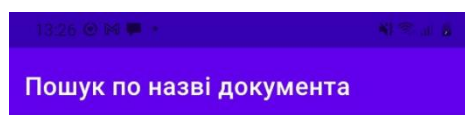
Б.4 – Форма видалення текстового документа

При натисканні кнопки «Редагування» визивається вікно редагування текстового документа, яке надано на рисунку Б.5:



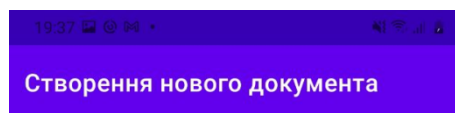
Б.5 – Форма редагування текстового документа

Для пошуку документів по назвам потрібно повернутися до головного меню та вибрати функцію «Пошук по назві», яка визиває вікно пошуку по назві, яке надано на рисунку Б.6:



Б.6 – Форма пошуку по назві

Для створення нового текстового документа потрібно повернутися до головного меню та вибрати функцію «Створення нового документа», яка визиває вікно створення нового документа, яке надано на рисунку Б.7:



Розклад

txt|

СТВОРИТИ

A form for creating a new document. It features a label "Розклад" (Schedule) above a text input field. The input field contains the text "txt" followed by a cursor. Below the input field is a purple button with the text "СТВОРИТИ" (CREATE).

Б.7 – Форма створення нового документа

ДОДАТОК В – Опис програмного забезпечення

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування програмного забезпечення – «IBook», що розробляється на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова. Позначення програми – «Робота з електронними документами».

1.2 Програмне забезпечення, необхідне для функціонування програми

Система повинна задовольняти вказаним вимогам на мобільного телефона наступної мінімальної комплекції:

- Процесор: 8 ядер, 2 ГГц, 64 бита;
- Оперативна пам'ять: 4 ГБ;
- 80 МВ вільного місця.

Для повноцінного функціонування системи необхідно використовувати ОС Android від версії 6.0 і вище.

1.3 Мови програмування

Для розробки програмного забезпечення обрали бібліотеку для розробки програм з графічним інтерфейсом на мові Java. Модулі, реалізовані на інших мовах програмування не використовуються.

2 Функціональне призначення

2.1 Класи розв'язуваних завдань та призначення програми

Функціональним призначенням програмного забезпечення «IBook» для Android» є автоматизація наступних процесів:

- редагування текстового документа;
- видалення текстового документа;
- створення нового текстового документа;
- пошук по назві текстового документа;

- сортування по формату;
- сортування дати останнього відкриття;
- сортування дати додавання до документа;

3 Опис логічної структури

3.1 Структура програми

Програмне забезпечення «IBook» має дві складові частини: клієнтська частина – це графічний інтерфейс та основні модулі програми - програмне забезпечення, що забезпечує зберігання даних та їх видачу в момент запиту.

Логічна структура програми представлена взаємодією основних модулів програми: модулями програми можна представити варіанти використання на діаграмі варіантів використання

3.2 Алгоритм програми

Робота програмного забезпечення «IBook» представлена взаємодією між клієнтською частиною та основними модулями програми. Алгоритм роботи програми є наступним:

1. Користувач взаємодіє з програмним додатком.
2. Програмний додаток звертається до сховища мобільного пристрою та отримує потрібні йому данні.
3. Програмний додаток відображає отриманий результат.

3.3 Використовувані методи

Програмне забезпечення – «IBook» розроблено мовою програмування Java, що є об'єктно-орієнтованою.

4 Технічні засоби, що використовуються

Програмне забезпечення – «IBook» потребує від мобільного телефону, на якому він буде встановлений, наступних характеристик, які треба розглядати як мінімальні:

- Процесор: 8 ядер, 2 ГГц, 64 бита;
- Оперативна пам'ять: 4 ГБ;
- 80 МВ вільного місця.

Для повноцінного функціонування системи необхідно використовувати ОС Android від версії 6.0 і вище.

5 Виклик та завантаження

Виклик та завантаження програмного забезпечення «IBook» відбувається запуском відповідно названого виконуваного файлу – IBook.jar. Запуск програмного додатку «IBook» іншими способами неможливий.

6 Вхідні дані

Вхідні дані: введення даних здійснюється за допомогою пристроїв введення даних (мобільний телефон, планшет). Дані до нотатку або вводяться вручну.

7 Вихідні дані

Вихідні дані: виведення даних здійснюється на дисплей мобільного телефону.

ДОДАТОК Г – Текст програми

В Android Studio весь код заходиться у класах, які розташовані в певному місці проекту.

Лістинг Г.1 – Клас «OpendDocument»

```

package com.example.dip;

import android.content.ContentValues;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class OpendDocument extends AppCompatActivity {

    private final static String FILE_NAME = getExternalPath();
    EditText Doc;
    Button Delete;
    Button Redact;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_opend_document);
        Doc = findViewById(R.id.editDoc);
        Delete = findViewById(R.id.buttonDelete);
        Redact = findViewById(R.id.buttonRedact);
    }

    public void saveText(View view){

        FileOutputStream fos = null;
        try {
            EditText textBox = findViewById(R.id.editor);
            String text = textBox.getText().toString();

            fos = openFileOutput(FILE_NAME, MODE_PRIVATE);
            fos.write(text.getBytes());
            Toast.makeText(this, "Файл сохранен",
Toast.LENGTH_SHORT).show();
        }
        catch(IOException ex) {

            Toast.makeText(this, ex.getMessage(),
Toast.LENGTH_SHORT).show();
        }
        finally{
            try{

```

```

        if(fos!=null)
            fos.close();
    }
    catch(IOException ex){
        Toast.makeText(this, ex.getMessage(),
Toast.LENGTH_SHORT).show();
    }
}
}
private void saveFile(String fileName) {
    try {
        OutputStream outputStream = openFileOutput(fileName,
0);
        OutputStreamWriter osw = new
OutputStreamWriter(outputStream);
        osw.write(mEditText.getText().toString());
        osw.close();
    } catch (Throwable t) {
        Toast.makeText(getApplicationContext(),
            "Exception: " + t.toString(),
            Toast.LENGTH_LONG).show();
    }
}

public void Delete(View view) {
    RedactDialogFragment dialog = new RedactDialogFragment();
    dialog.show(getSupportFragmentManager(), "redact");
}

public void Redact(View view) {
    RedactDialogFragment dialog = new RedactDialogFragment();
    dialog.show(getSupportFragmentManager(), "redact");
}
public void Exit(View view) {
    Intent intent = new Intent(this, MainActivity.class);
    startActivity(intent);
}
}
}

```

ЛІСТИНГ Г.2 – Клас «RedactDialogFragment»

```

package com.example.bd.activities;

import android.app.AlertDialog;
import android.app.Dialog;
import android.os.Bundle;
import androidx.fragment.app.DialogFragment;
import androidx.annotation.NonNull;

import com.example.bd.R;

public class CustomDialogFragment extends DialogFragment {

    @NonNull
    public Dialog onCreateDialog(Bundle savedInstanceState) {

```

```

        AlertDialog.Builder builder=new
AlertDialog.Builder(getActivity());
        return builder
            .setTitle("Редагування")
            .setView(R.layout.dialog)
            .setPositiveButton("OK", null)
            .setNegativeButton("Отмена", null)
            .create();
    }
    public void onPositiveButtonClick(View view) {
        SaveRedact redact = saveRedact(R.layout.dialog);
    }
}

```

Лістинг Г.3 – Клас «CreateDocument»

```

package com.example.dip;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class CreateDocument extends AppCompatActivity {

    private Toolbar mToolbar;
    Dialog dialog;
    private EditText mEditText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mToolbar = findViewById(R.id.toolbar);
        setTitle("Створення нового документа");
        setSupportActionBar(mToolbar);

    }
    public void onClick(View view) {
        String name = currentName.getText().toString();
        String form = currentForm.getText().toString();
        Create create = new Create(name,form);
        Intent intent = new Intent(this, MainActivity.class);
        startActivity(intent);
    }

}

```

Лістинг Г.4 – Клас «DocumentView»

```

package com.example.dip;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

```

```

import android.os.Bundle;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.ScrollView;
import android.widget.TextView;

public class DocumentVeiw extends AppCompatActivity {
    List<TextView> textViewList = new ArrayList<>();
    private Toolbar mToolbar;
    public String OpenedDoc;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mToolbar = findViewById(R.id.toolbar);
        setTitle("Список документів");
        setSupportActionBar(mToolbar);
        LinearLayout linearLayout = new LinearLayout(this);
        linearLayout.setOrientation(LinearLayout.VERTICAL);
        doc = DocScan.getDocs();
        num = DocScan.Num();
        DateLast = Doc DocScan.getDateL();
        DateAdd = Doc DocScan.getDateL();
        for(int i = 1; i < num + 1; i++){
            int id = getResources().getIdentifier("textView" + i, "id",
this.getPackageName());
            TextView textViewItem = (TextView)findViewById(id);
            textViewItem.setText(doc + DateLast + DateAdd);
            if(i%2 ==0){
                textViewItem.setBackgroundColor(0xFFBDBDBD);
            }
            textViewItem.setTextSize(18);
            linearLayout.addView(textViewItem, new
LinearLayout.LayoutParams
                (LinearLayout.LayoutParams.MATCH_PARENT, 0, 1));
        }
        public void Open(View view) {
            OpenedDoc = doc;
            Intent intent = new Intent(this, OpendDocument.class);
            startActivity(intent);
        }
        public void Exit(View view) {
            Intent intent = new Intent(this, MainActivity.class);
            startActivity(intent);
        }
    }
}

```

Лістинг Г.5 – Клас «Main»

```

package com.example.dip;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

```

```
import com.example.bd.R;

public class Main extends AppCompatActivity {

    private Toolbar mToolbar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mToolbar = findViewById(R.id.toolbar);
        setTitle("Головне меню");
        setSupportActionBar(mToolbar);
    }

    public void add(View view) {
        Intent intent = new Intent(this, CreateDoc.class);
        startActivity(intent);
    }

    public void search(View view) {
        Intent intent = new Intent(this, SearchDoc.class);
        startActivity(intent);
    }

    public void sort(View view) {
        Intent intent = new Intent(this, SortDoc.class);
        startActivity(intent);
    }

    public void list(View view) {
        Intent intent = new Intent(this, ListDoc.class);
        startActivity(intent);
    }
}
```

ДОДАТОК Д - Програма та методика випробувань програмного забезпечення

1 Об'єкт випробування

Об'єктом випробування є програмне забезпечення «IBook», що розробляється на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова.

2 Мета випробування

Метою проведення випробування є перевірка працездатності даного мобільного застосунку, перевірка відповідності характеристик та вимог розробленого застосунку, викладених у технічному завданні, приведення прикладу роботи та отримання відповідних навичок.

2 Вимоги до додатка

При проведенні випробувань функціональні характеристики (можливості) програми підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до функціональних характеристик» технічного завдання.

4 Вимоги до програмної документації

Програмна документація повинна включати в себе наступні документи:

1. Текст програми.
2. Опис програмного забезпечення.
3. Програма та методика випробувань.
4. Інструкція користувача.

5 Програма та методика випробувань програмного забезпечення

5 Засоби і порядок випробувань

5.1 Технічні засоби, що використовуються під час випробувань

Склад технічних засобів, що використовувалися під час випробувань:

- Процесор: 8 ядер, 2 ГГц, 64 бита;
- Оперативна пам'ять: 4 ГБ;
- 80 МВ вільного місця.

5.2 Програмні засоби, що використовуються під час випробувань

Склад програмних засобів, що використовувалися під час випробувань:

- операційна система ОС Android версії 6 і вище;
- Java Runtime Environment (JRE 1.8).

5.3 Порядок проведення випробувань

Порядок проведення випробувань складається з перевірки вимог до функціональних характеристик програмного продукту та перевірки вимог до програмної документації.

6 Методи випробувань

6.1 Методика проведення перевірки вимог до програмної документації

Перевірка дотримання вимог програмної документації на програмний продукт виконується візуально представником служби, відповідальної за експлуатацію. У ході перевірки зіставляється склад і комплектність програмної документації, представленої розробником, з переліком програмної документації, наведеним у пункті «Склад програмної документації, пропонованої на випробування» цього документа. Перевірка вважається завершеною у випадку відповідності складу та комплектності програмної документації, представленої розробником, переліком програмної документації, наведеному у зазначеному вище пункті.

6.2 Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» Додатку А – Технічне завдання. Перевірка вважається завершеною у разі відповідності складу і послідовності

виконаних дій пункту «Вимоги до функціональних характеристик». В процесі тестування була перевірена функціональність за темою «IBook». У табл. Д.1, що наведена нижче, вказано перелік випробувань основних функціональних можливостей

Таблиця Д.1 - Методика випробування

№	Дія	Очікуваний результат	Результат перевірки	Зауваження
1	2	3	4	5
1	Випробування функції редагування текстового документу	Застосунок повинен змінити текстовий документ і зберегти	Виконано	
2	Випробування функції видалення текстового документа	Застосунок повинен видалити текстовий документ	Виконано	
3	Випробування функції створення нового текстового документа	Застосунок повинен створити новий текстовий документ	Виконано	
4	Випробування функції пошуку по назві текстового документа	Застосунок повинен перевірити схожість назв текстових документів і знайти схожі	Виконано	

Продовження таблиці Д.1

1	2	3	4	5
5	Випробування функції сортування по формату	Застосунок повинен знайти текстові документи з відповідного формату	Виконано	
6	Випробування функції сортування за датою додавання до застосунку	Застосунок повинен відсортувати текстові документи за датою додавання їх до застосунку	Виконано	
7	Випробування функції сортування за датою останнього відкриття	Застосунок повинен відсортувати текстові документи за датою останнього відкриття	Виконано	