

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка мобільного додатку для управління та організації бібліотеки CAD-файлів у форматах DWG і DXF

Виконав: студент

IV курсу, групи СН-43

спеціальності

122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Тимошенко Т.С

(прізвище та ініціали)

Керівник

(підпис)

Марценко С.В

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Тимошенку Тарасу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка мобільного додатку для управління та організації бібліотеки
CAD-файлів у форматах DWG і DXF

Керівник роботи Марценко Сергій Володимирович, канд. техн. наук, доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» травня 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 23 червня 2025 р.

3. Вихідні дані до роботи Перелік вихідних матеріалів та даних

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. Розділ 1. Аналіз предметної області та постановка завдання. 1.1 Аналіз предметної області. 1.2 Аналіз існуючих рішень для роботи з CAD-файлами на мобільних пристроях.

1.3 Постановка задачі розробки мобільного додатку. 1.4 Висновок до першого розділу

Розділ 2. Вибір технологій та інструментів розробки. 2.1 Вибір технологій та інструментів.

2.2 Архітектура та структура мобільного додатку. 2.3 Архітектура та структура мобільного додатку 2.4 Висновок до другого розділу. Розділ 3 Практична частина. реалізація та

тестування мобільного додатку. 3.1 Реалізація основного функціоналу додатку.

3.2 Робота з бібліотекою файлів: збереження, перегляд, управління. 3.3 Тестування

мобільного додатку. 3.4 Висновок до третього розділу. Розділ 4 Безпека життєдіяльності та охорона праці 4.1 Діяльність. Її види та розуміння в безпеці праці. 4.2 Загальні вимоги

безпеки з охорони праці для користувачів ПК 4.3 Висновок до четвертого розділу

Висновки. Перелік джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульний слайд, 2. Актуальність теми, 3. Мета та завдання роботи, 4. Аналіз предметної області, 5. Аналіз існуючих рішень, 6. Архітектура мобільного додатку, 7. Функціональні можливості додатку, 8. Технології реалізації, 9. Тестування та результати, 10. Висновки та перспективи розвитку.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин Віктор Степанович		

7. Дата видачі завдання 27 січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	07.05.2025	Виконано
2.	Підбір та опрацювання літературних джерел по темі кваліфікаційної роботи	08.05.2025–15.05.2025	Виконано
3.	Виконання дослідження щодо розробки мобільного додатку для файлів формату DWG та DXF	16.05.2025–22.05.2025	Виконано
	Розроблення додатку CADLIB		
4.	Оформлення розділу «АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ. »	23.05.2025–01.06.2025	Виконано
5.	Оформлення розділу «ВИБІР ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ РОЗРОБКИ»	23.05.2025–01.06.2025	Виконано
6.	Оформлення розділу «ПРАКТИЧНА ЧАСТИНА. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОБІЛЬНОГО ДОДАТКУ»	23.05.2025–01.06.2025	Виконано
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	02.06.2025–05.06.2025	Виконано
8.	Виконання завдання до підрозділу «Основи охорони праці»	02.06.2025–05.06.2025	Виконано
9.	Оформлення кваліфікаційної роботи	06.06.2025–10.06.2025	Виконано
10.	Нормоконтроль	24.06.2025–25.06.2025	Виконано
11.	Перевірка на плагіат	20.06.2025	Виконано
12.	Попередній захист кваліфікаційної роботи	25.06.2025	Виконано
13.	Захист кваліфікаційної роботи	28.06.2025	

Студент

(підпис)

Тимошенко Т.С

(прізвище та ініціали)

Керівник роботи

(підпис)

Марценко С.В

(прізвище та ініціали)

АНОТАЦІЯ

Розробка мобільного додатку для управління та організації бібліотеки CAD-файлів у форматах DWG і DXF // Кваліфікаційна робота освітнього рівня «Бакалавр» // Тимошенко Тарас Сергійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-43 // Тернопіль, 2025 // С. 61 , рис. – 5 , табл. – 1 , кресл. – 0 , додат. – 13 , бібліогр. – 0 .

Ключові слова: cad-файли, мобільний додаток, управління файлами, бібліотека файлів, dwg, dxf, android, flutter.

Кваліфікаційна робота присвячена дослідженню процесу розробки мобільного додатку для управління та організації бібліотеки CAD-файлів у форматах DWG і DXF на базі технології Flutter.

У першому розділі кваліфікаційної роботи описано сучасний стан розробки мобільних додатків для роботи з CAD-файлами. Висвітлено аналіз існуючих програмних рішень. Розглянуто основні технології, що використовуються для створення подібних систем. Проаналізовано вимоги до мобільних додатків, які забезпечують зручність управління CAD-файлами.

У другому розділі кваліфікаційної роботи подано обґрунтування вибору технологій та середовища розробки, зокрема Flutter та мови Dart. Досліджено архітектуру мобільного додатку та спроектовано його структуру. Подано опис інтерфейсу користувача, що забезпечує зручність роботи з бібліотекою файлів.

У третьому розділі кваліфікаційної роботи описано процес реалізації основного функціоналу мобільного додатку. Проаналізовано можливості збереження, перегляду та передачі CAD-файлів через вбудовані плеєри. Проведено тестування додатку та оцінено його відповідність вимогам до продуктивності, зручності використання та стабільності роботи.

Об'єкт дослідження: процес управління CAD-файлами у мобільних додатках.

Предмет дослідження: методи, інструменти та технології розробки мобільного додатку для організації бібліотеки CAD-файлів у форматах DWG і DXF.

ANNOTATION

Development of a Mobile App for Organizing and Managing a Library of CAD Files in PWG and DXF Formats // Qualification work of the educational level «Bachelor» // Tymoshenko Taras // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-43 // Ternopil, 2025 // P. 61 , fig. – 5 , tabl. – 1 , chair. – 0 , annexes. – 13 , references – 0 .

Keywords: cad files, mobile application, file management, file library, dwg, dxf, android, flutter.

The qualification work is devoted to the study of the process of developing a mobile application for managing and organizing a library of CAD files in DWG and DXF formats based on Flutter technology.

The first section of the qualification work describes the current state of development of mobile applications for working with CAD files. The analysis of existing software solutions is highlighted. The main technologies used to create such systems are considered. The requirements for mobile applications that provide convenience in managing CAD files are analyzed.

Another section of the qualification work provides a justification for the choice of technologies and development environment, in particular Flutter and the Dart language. The architecture of the mobile add-on is studied and its structure is designed. A description of the user interface is provided, which ensures the convenience of working with the file library.

The third section of the qualification work describes the process of implementing the main functional mobile application. The possibilities of saving, viewing and transferring CAD files through built-in players are analyzed. The application was tested and its compliance with the requirements for performance, usability and stability of work was assessed.

Object of research: the process of managing CAD files in mobile applications.

Subject of research: methods, tools and technologies for developing a mobile application for organizing a library of CAD files in DWG and DXF formats.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ТЕРМІНІВ

MIT (англ. Massachusetts Institute of Technology) – Массачусетський технологічний інститут.

CAD (англ. Computer–Aided Design) – автоматизоване проєктування.

DWG (англ. Drawing) – формат файлу креслень, розроблений Autodesk.

DXF (англ. Drawing Exchange Format) – відкритий формат обміну кресленнями.

API (англ. Application Programming Interface) – інтерфейс програмування застосунків.

UI (англ. User Interface) – користувацький інтерфейс.

UX (англ. User Experience) – досвід користувача.

SDK (англ. Software Development Kit) – комплект засобів розробки програмного забезпечення.

OS (англ. Operating System) – операційна система. RAM (англ. Random Access Memory) – оперативна пам'ять.

CPU (англ. Central Processing Unit) – центральний процесор.

HTTP (англ. HyperText Transfer Protocol) – протокол передачі гіпертексту.

JSON (англ. JavaScript Object Notation) – формат обміну даними.

REST (англ. Representational State Transfer) – стиль архітектури веб-сервісів.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	12
1.1 Аналіз предметної області.....	12
1.1.1 Характеристика форматів DWG і DXF	13
1.2 Аналіз існуючих рішень для роботи з CAD-файлами	14
1.2.1 Огляд популярних мобільних застосунків для роботи з CAD – файлами.....	16
1.3 Постановка задачі розробки мобільного додатку	17
1.3.1 Вимоги до функціоналу мобільного додатку.....	19
1.4 Висновок до першого розділу	20
РОЗДІЛ 2 ВИБІР ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ РОЗРОБКИ	22
2.1 Вибір технологій та інструментів розробки	22
2.1.1 Аналіз середовищ розробки	23
2.1.1 Обґрунтування вибору мов програмування та фреймворків.....	24
2.2 Архітектура та структура мобільного додатку.....	26
2.2.1 Вибір архітектурного шаблону для мобільного додатку	27
2.2.2 Опис основних компонентів та їх взаємодії.....	28
2.2.3 Структура проекту в середовищі Android Studio.....	29
2.3 Архітектура та структура мобільного додатку.....	31
2.3.1 Принципи побудови UX/UI для мобільних додатків	32
2.3.2 Схеми та екрани мобільного додатку	34
2.4 Висновок до другого розділу	35
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОБІЛЬНОГО ДОДАТКУ	37
3.1 Реалізація основного функціоналу додатку.....	37
3.3.1 Реалізація пошуку файлів у бібліотеці	39
3.3.2 Відображення дати додавання файлів	40
3.2 Робота з бібліотекою файлів: збереження, перегляд, управління.....	42

3.3 Тестування мобільного додатку.....	45
3.3.1 Методика функціонального тестування мобільного додатку	46
3.3.2 Методика функціонального тестування мобільного додатку	47
3.4 Висновок до третього розділу	48
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОХОРОНА ПРАЦІ	50
4.1 Діяльність. Її види та розуміння в безпеці праці.....	50
4.2 Загальні вимоги безпеки з охорони праці для користувачів ПК.....	52
4.3 Висновок до четвертого розділу	54
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ	58
ДОДАТКИ	

ВСТУП

В умовах глобалізації та стрімкого розвитку цифрових технологій особливої актуальності набуває питання оптимізації процесів зберігання, обробки та ефективного управління інженерною документацією. Це є надзвичайно важливим для таких галузей, як будівництво, архітектура та машинобудування, де ключовим елементом інформаційного обміну та збереження технічних даних є CAD-файли, зокрема у форматах DWG та DXF. З огляду на постійну потребу у мобільності та оперативному доступі до проєктної документації, сучасні користувачі дедалі більше орієнтуються на використання мобільних рішень, які не потребують прив'язки до стаціонарних комп'ютерів. У цьому контексті розробка мобільного додатку для ефективного управління та організації бібліотеки CAD-файлів є вкрай важливим і своєчасним напрямом сучасних досліджень у сфері інформаційних технологій та програмної інженерії.

Метою даної дипломної роботи є розробка мобільного додатку, який забезпечить підвищення зручності управління та доступу до CAD-файлів у зазначених форматах, а також створить ефективну систему їх локальної організації на мобільних пристроях. Для досягнення поставленої мети необхідно вирішити такі основні завдання: здійснити аналіз сучасного стану досліджень і наявних програмних рішень для роботи з CAD-файлами на мобільних пристроях; визначити вимоги до розробки мобільного додатку з урахуванням особливостей роботи з форматами DWG та DXF; спроектувати архітектуру та користувацький інтерфейс програмного продукту; реалізувати функціональні можливості щодо збереження, перегляду, організації та управління CAD-файлами; провести комплексне тестування розробленому додатку з метою перевірки його відповідності функціональним, технічним та ергономічним вимогам.

Практична цінність отриманих результатів полягає у створенні зручного та доступного мобільного інструменту, що забезпечує швидку організацію, перегляд і керування бібліотекою CAD-файлів без необхідності постійного підключення до мережі Інтернет. Запропоноване рішення має значний потенціал для застосування фахівцями у сферах будівництва, проєктування, архітектури та суміжних галузях, особливо у випадках, коли робота здійснюється в динамічних або виїзних умовах, де мобільність та автономність програмного забезпечення мають вирішальне значення.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.

1.1 Аналіз предметної області

У сучасному світі проєктно–конструкторська документація відіграє ключову роль у сферах будівництва, машинобудування, архітектури, енергетики та суміжних галузях. Основним інструментом для створення такої документації є системи автоматизованого проєктування (CAD), які дозволяють моделювати, редагувати та візуалізувати об'єкти у дво– та тривимірному просторі. Результати роботи в CAD – системах зазвичай зберігаються у спеціалізованих форматах файлів, серед яких найбільш поширеними є формати DWG та DXF [1].

Формат DWG, розроблений компанією Autodesk, є одним з основних для зберігання креслень, технічних схем та планів. Файли цього типу можуть містити велику кількість детальної інформації, включаючи шари, анотації, об'єкти різного рівня складності, а також векторну графіку. Формат DXF (Drawing Exchange Format) є відкритим стандартом обміну, який дозволяє передавати креслення між різними CAD – програмами, зберігаючи при цьому їхню структуру та властивості [1].

Традиційно робота з CAD – файлами здійснюється на персональних комп'ютерах або робочих станціях з потужним апаратним забезпеченням. Однак із розвитком мобільних технологій зростає потреба у використанні таких файлів безпосередньо на мобільних пристроях – смартфонах і планшетах. Це обумовлено необхідністю оперативного перегляду креслень на будівельних майданчиках, об'єктах монтажу, під час виїзних нарад або у процесі погодження проєктних рішень.

У мобільному середовищі користувачі стикаються з низкою обмежень, серед яких:

- обмежені апаратні ресурси пристроїв порівняно з традиційними комп'ютерами;
- необхідність зручного та швидкого доступу до файлів без додаткових налаштувань;
- потреба в автономній роботі без стабільного інтернет-з'єднання;
- важливість організації файлів у зрозумілу структуру, яка дозволяє швидко знаходити потрібні документи [3].

Зазначені особливості формують вимоги до програмного забезпечення, яке повинно бути оптимізованим для мобільних пристроїв, підтримувати ефективну роботу з великими файлами та забезпечувати простий і доступний інтерфейс [4].

У зв'язку з цим особливо актуальною є розробка мобільного додатку, який забезпечить локальне зберігання, перегляд та організацію CAD-файлів у форматах DWG та DXF без обов'язкової синхронізації з хмарними сервісами або сторонніми ресурсами. Це дозволить значно підвищити мобільність, продуктивність та оперативність роботи фахівців, які використовують інженерну документацію у своїй професійній діяльності.

1.1.1 Характеристика форматів DWG і DXF

Формати DWG і DXF є основними стандартами для зберігання та обміну креслярською документацією в системах автоматизованого проєктування (CAD). DWG (від англ. Drawing) – це власницький бінарний формат, розроблений компанією Autodesk, який широко використовується для збереження дво- та тривимірних креслень, технічних схем, планів та інших графічних матеріалів. Формат DWG відомий своєю здатністю зберігати складні структури даних, які включають багаторівневі шари, анотації, блоки, текстові та графічні об'єкти, а також метадані, що робить його універсальним для різних сфер інженерії і архітектури. Його підтримують більшість провідних CAD –

програм, що гарантує сумісність і стабільність роботи з файлами цього формату.

DXF (Drawing Exchange Format), створений також компанією Autodesk, є відкритим текстовим форматом, розробленим для забезпечення обміну кресленнями між різними CAD – системами [1]. На відміну від DWG, DXF має більш просту структуру, яка розшифровується людиною, що полегшує її аналіз і конвертацію, але водночас це може призводити до збільшення розміру файлу. Основною перевагою DXF є забезпечення сумісності між програмним забезпеченням, що робить його популярним вибором для обміну документацією між організаціями та фахівцями, які використовують різні CAD– системи. Крім того, DXF підтримує більшість графічних і структурних елементів креслень, що дозволяє зберігати їхню цілісність при передачі.

Обидва формати є надзвичайно важливими в таких галузях, як будівництво, машинобудування, промисловий дизайн та енергетика, оскільки забезпечують універсальність та гнучкість у роботі з проєктною документацією. Однак для мобільних додатків, що працюють із цими форматами, існують певні виклики. DWG – файли часто мають великий розмір і складну структуру, що вимагає оптимізації обробки на пристроях з обмеженими ресурсами. DXF, попри відкритість формату, також потребує адаптації для швидкого перегляду та обробки на мобільних платформах. Таким чином, розробка ефективного мобільного рішення повинна враховувати ці технічні особливості, забезпечуючи високу продуктивність, зручність використання та автономність у роботі без постійного інтернет-з'єднання.

1.2 Аналіз існуючих рішень для роботи з CAD–файлами

На сьогоднішній день ринок мобільних застосунків пропонує ряд рішень для перегляду та базової обробки CAD – файлів, зокрема у форматах DWG та DXF. Серед найбільш відомих мобільних програм варто відзначити такі

продукти, як AutoCAD Mobile, DWG FastView, A360 Viewer, а також ряд менш популярних, але функціональних рішень від сторонніх розробників [5].

AutoCAD Mobile є одним із найбільш повнофункціональних мобільних застосунків для роботи з CAD – файлами. Ця програма підтримує перегляд, базове редагування, вимірювання об'єктів, а також синхронізацію з хмарними сервісами. Однак серед її основних недоліків можна виокремити складність інтерфейсу для нових користувачів, обмеження безкоштовної версії, а також залежність від стабільного інтернет – з'єднання для доступу до всього спектра функцій [6].

DWG FastView – додаток, який дозволяє переглядати та коментувати CAD – файли на мобільних пристроях. Його перевагами є простий інтерфейс та можливість автономної роботи, що є важливою характеристикою для використання у виїзних умовах. Проте цей додаток має обмежену функціональність щодо редагування файлів, що зменшує його застосування у складніших робочих сценаріях [7].

A360 Viewer – хмарне рішення для перегляду CAD – файлів, яке забезпечує доступ до креслень з будь – якої точки світу за умов наявності інтернет – з'єднання. Серед переваг цього сервісу можна зазначити швидку інтеграцію з іншими продуктами Autodesk та простоту спільного доступу до файлів. Водночас основним недоліком залишається повна залежність від хмарної інфраструктури, що обмежує його використання в умовах відсутності стабільного доступу до мережі.

Окрім перелічених застосунків, існує значна кількість рішень від менших розробників, однак більшість з них мають низький рівень оптимізації, обмежену підтримку великих файлів, а також не забезпечують необхідної швидкодії при роботі з об'ємною графічною інформацією.

Таким чином, розробка мобільного додатку, який забезпечить зручне локальне зберігання, перегляд, організацію та управління CAD – файлами у форматах DWG і DXF без необхідності постійного підключення до мережі

Інтернет, є актуальною задачею, яка потребує практичної реалізації та подальшого дослідження[8].

1.2.1 Огляд популярних мобільних застосунків для роботи з CAD – файлами

Сучасний ринок мобільних застосунків пропонує декілька популярних рішень для роботи з CAD – файлами у форматах DWG та DXF, кожне з яких має свої переваги та недоліки. Найпоширенішим і найбільш функціональним є AutoCAD Mobile від Autodesk – офіційний мобільний додаток, який надає користувачам можливість перегляду, редагування, анотації креслень, а також синхронізації з хмарними сервісами. AutoCAD Mobile підтримує більшість функцій повноцінної десктопної версії, що робить його незамінним інструментом для професіоналів. Однак цей додаток має складний інтерфейс для новачків і є частково платним, при цьому безкоштовна версія обмежена у функціональності. Крім того, ефективне використання AutoCAD Mobile часто вимагає стабільного інтернет-з'єднання для доступу до хмарних сервісів, що ускладнює роботу в умовах поганого зв'язку або його відсутності[5].

Іншим варіантом є DWG FastView, який позиціонується як більш легкий та простий у використанні додаток, орієнтований на перегляд і базову обробку CAD – файлів. Головною перевагою DWG FastView є можливість автономної роботи без підключення до інтернету, що робить його привабливим для інженерів і будівельників, які часто працюють у польових умовах. Водночас функціонал програми є обмеженим, особливо в частині редагування, що не дозволяє виконувати складні операції без використання додаткових інструментів[6].

A360 Viewer – це рішення від Autodesk, яке пропонує хмарний перегляд CAD-файлів з можливістю спільної роботи над кресленнями. Цей додаток зручний для командної роботи, оскільки забезпечує доступ до файлів з будь-якого пристрою, що підключений до Інтернету. Однак його повна залежність

від мережевого з'єднання обмежує використання в автономних або виїзних умовах[7].

Окрім великих гравців, на ринку також присутні численні менш відомі програми, які пропонують базовий перегляд CAD – файлів. Вони часто мають обмежену підтримку форматів, недостатню оптимізацію та погану інтеграцію з файловою системою мобільних пристроїв. Ці фактори знижують їхню практичну цінність для професіоналів, які потребують швидкого та надійного доступу до технічної документації у польових умовах.

З огляду на вищевикладене, виникає потреба у розробці мобільного додатку, який би поєднував переваги автономності, простоти використання та розширеного функціоналу для роботи з CAD – файлами без залежності від інтернету, забезпечуючи ефективне управління локально збереженими документами[8].

1.3 Постановка задачі розробки мобільного додатку

Аналіз наявних програмних рішень для роботи з CAD – файлами на мобільних пристроях засвідчує наявність низки обмежень, серед яких особливої уваги заслуговують такі: необхідність постійного інтернет-з'єднання, відсутність функцій локального управління файлами, обмежені можливості організації персональної бібліотеки та недостатня оптимізація для швидкої роботи з великими файлами. Ці недоліки створюють суттєві перешкоди для професіоналів, які працюють у виїзних умовах або на об'єктах, де підключення до мережі є нестабільним чи відсутнім.

З огляду на це, виникає необхідність у створенні мобільного додатку, який би дозволяв користувачам виконувати повний цикл дій з CAD – файлами без потреби у зовнішніх сервісах, забезпечував швидкий локальний доступ до файлів, мав зручну систему їхньої організації та підтримував інтеграцію з файловою структурою мобільного пристрою[9].

Основними вимогами до розроблюваного програмного забезпечення є:

- підтримка форматів DWG та DXF для перегляду та організації файлів;
- можливість завантаження, перегляду та збереження файлів без підключення до мережі Інтернет;
- створення зручної бібліотеки файлів із можливістю сортування, пошуку та редагування атрибутів файлів;
- забезпечення простого та інтуїтивно зрозумілого інтерфейсу користувача, що дозволить швидко опанувати додаток без необхідності проходження спеціального навчання;
- реалізація ефективного та оптимізованого механізму роботи з великими файлами з метою забезпечення високої швидкодії додатка.

Завдання, які необхідно вирішити у межах цієї роботи, можна сформулювати таким чином:

- провести аналіз предметної області та оцінити наявні рішення для роботи з CAD – файлами на мобільних пристроях;
- розробити архітектуру мобільного додатку, що відповідатиме зазначеним вимогам;
- реалізувати функціонал для локального збереження, перегляду та управління файлами у форматах DWG та DXF;
- створити користувацький інтерфейс, орієнтований на зручність та швидкість доступу до основних функцій;
- здійснити тестування програмного продукту та оцінити його відповідність технічним та ергономічним вимогам.

Таким чином, результатом виконання даної роботи має стати мобільний додаток, який забезпечуватиме автономну, зручну та ефективну роботу з CAD – файлами у мобільному середовищі [10].

1.3.1 Вимоги до функціоналу мобільного додатку

Враховуючи результати аналізу предметної області та існуючих мобільних рішень для роботи з CAD – файлами, було сформульовано комплекс вимог до розроблюваного мобільного додатку. Перш за все, додаток має забезпечувати підтримку форматів DWG та DXF, що є ключовими для інженерної та архітектурної документації. Це означає не лише можливість відкриття та перегляду файлів, а й реалізацію базових операцій із файлами, включаючи завантаження, збереження, організацію, сортування та видалення.

Особливу увагу слід приділити автономній роботі програми, оскільки багато користувачів будуть використовувати додаток у польових умовах, де відсутній або нестабільний доступ до мережі Інтернет. Таким чином, функціонал має бути орієнтований на локальне збереження файлів із можливістю роботи без підключення до хмарних сервісів. Це передбачає розробку ефективних механізмів кешування, оптимізації обробки великих файлів та організації зручної структури бібліотеки.

Інтерфейс користувача повинен бути інтуїтивно зрозумілим, мінімалістичним, але водночас функціональним, щоб користувачі без спеціальної підготовки могли швидко освоїти роботу з додатком. Важливо реалізувати пошук, фільтрацію та сортування файлів за різними критеріями (назва, дата, тип файлу тощо), а також можливість додавання анотацій чи базового редагування для оперативної роботи.

Враховуючи складність роботи з великими CAD – файлами, програмне забезпечення повинно бути оптимізованим для мінімізації споживання ресурсів мобільного пристрою – оперативної пам'яті, процесора та дискового простору, що забезпечить плавний перегляд та взаємодію з документами без затримок.

До додаткових вимог можна віднести інтеграцію з файловою системою мобільного пристрою, що полегшить завантаження файлів у додаток та їх подальше експортування або обмін. Окрім цього, бажано передбачити

можливість резервного копіювання локальної бібліотеки для запобігання втраті даних [11].

Отже, мобільний додаток повинен стати потужним, надійним інструментом для інженерів, архітекторів і будівельників, забезпечуючи ефективну та автономну роботу з CAD – файлами, що сприятиме підвищенню продуктивності та якості виконання професійних завдань у різних умовах.

1.4 Висновок до першого розділу

У першому розділі було проведено аналіз предметної області, що підтвердив актуальність розробки мобільного додатку для управління та організації бібліотеки CAD – файлів у форматах DWG та DXF. Розглянуто особливості роботи з CAD – документацією, зокрема специфіку зберігання та перегляду таких файлів на мобільних пристроях, що обумовлює необхідність створення зручних, швидкодійних і автономних програмних рішень.

Аналіз існуючих програмних продуктів показав, що хоча на ринку представлені різноманітні інструменти для роботи з CAD – файлами, більшість із них вимагають стабільного доступу до мережі Інтернет або мають обмежений функціонал у безкоштовних версіях. Це значно знижує їхню практичну цінність у реальних умовах використання, особливо для фахівців, які працюють на виїздах або в зонах зі слабким покриттям мобільного зв'язку.

На основі проведеного аналізу сформульовано постановку задачі, яка передбачає розробку мобільного додатку із функціоналом, що забезпечує:

- локальне збереження та структурування CAD – файлів;
- перегляд файлів без підключення до Інтернету;
- інтуїтивно зрозумілий та зручний інтерфейс користувача;
- базові інструменти для організації файлів: додавання, перегляд, сортування та видалення.

Таким чином, у наступному розділі буде розглянуто архітектуру мобільного додатку, вибір технологій, інструментів реалізації, а також розроблені вимоги до інтерфейсу та функціоналу програмного продукту [12].

РОЗДІЛ 2 ВИБІР ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ РОЗРОБКИ

2.1 Вибір технологій та інструментів розробки

Розробка мобільного додатку для управління та організації бібліотеки CAD – файлів у форматах DWG і DXF вимагає застосування сучасних, ефективних і зручних у роботі технологій, які забезпечать високу продуктивність, кросплатформність та підтримку всього необхідного функціоналу. Аналізуючи доступні варіанти, було прийнято рішення використовувати кросплатформний фреймворк Flutter, який дозволяє створювати додатки для операційних систем Android і iOS з єдиної кодової бази. Це значно скорочує час розробки та спрощує подальше супроводження проєкту.

Flutter – це відкритий фреймворк, розроблений компанією Google, що відзначається високою продуктивністю та гнучкістю інтерфейсу користувача завдяки використанню власного движка рендерингу. Однією з основних переваг Flutter є функція “hot reload”, що дозволяє миттєво бачити зміни у коді без повторної компіляції, що значно пришвидшує процес розробки та тестування [13].

Мовою програмування, на якій базується Flutter, є Dart. Dart є сучасною об'єктноорієнтованою мовою зі статичною типізацією, що поєднує простоту синтаксису та потужні засоби асинхронного програмування. Вона оптимізована для розробки швидких і ефективних клієнтських додатків. Dart дозволяє ефективно працювати з файловою системою, асинхронними операціями, а також підтримує інтеграцію з великою кількістю бібліотек і плагінів.

Серед інтегрованих середовищ розробки (IDE) для Flutter найбільш зручним та функціональним є Android Studio. Воно пропонує потужні інструменти налагодження, редактор коду з підсвічуванням синтаксису, емулятори мобільних пристроїв, а також інтеграцію з Flutter SDK.

Використання Android Studio дозволяє організувати ефективний робочий процес та швидко виявляти помилки[14].

Для реалізації функціоналу, пов'язаного зі збереженням, вибором та переглядом CAD – файлів, у проєкті будуть застосовані такі популярні бібліотеки Flutter, як `file_picker` для вибору файлів із пам'яті пристрою, `path_provider` для доступу до директорій додатка, `open_file` для відкриття файлів зовнішніми програмами, а також `shared_preferences` для збереження налаштувань користувача та даних між сесіями. За потреби може бути використаний SDK або сторонні плагіни для роботи з форматом DWG/DXF.

Обраний набір технологій та інструментів відповідає вимогам до мобільного додатку, забезпечує необхідну гнучкість, продуктивність і зручність у підтримці, що є ключовим для успішної реалізації проєкту.

2.1.1 Аналіз середовищ розробки

Dart – це об'єктноорієнтована мова програмування зі статичною типізацією, розроблена компанією Google для створення високопродуктивних клієнтських застосунків. Вона використовується як основна мова для фреймворку Flutter, що дає змогу розробляти кросплатформні мобільні додатки з єдиним кодом для Android та iOS [1].

Однією з ключових особливостей Dart є проста та зрозуміла синтаксична структура, що значно полегшує процес навчання та прискорює розробку. Dart підтримує асинхронне програмування на основі `Future` та `Stream`, що дозволяє ефективно виконувати операції вводу–виводу, зокрема роботу з файлами та мережею, не блокуючи основний потік виконання.

Мова має вбудовану підтримку сучасних парадигм програмування, таких як функціональне програмування, із можливістю використання лямбда–виразів, генераторів та розширень методів. Статична типізація в Dart допомагає виявляти помилки на етапі компіляції, що підвищує надійність коду.

Dart компілюється у нативний код, що забезпечує високу продуктивність додатків. При цьому, для вебзастосунка Dart може трансформуватися у JavaScript, що демонструє гнучкість мови.

Для роботи з Flutter Dart має потужну екосистему пакетів, що включає бібліотеки для роботи з файлами, базами даних, мережею та іншими ресурсами. Така інтеграція полегшує створення комплексних додатків з широким функціоналом.

2.1.1 Обґрунтування вибору мов програмування та фреймворків

Для розробки мобільного додатку було обрано фреймворк Flutter, який на сьогодні є одним із найперспективніших інструментів для створення кросплатформних мобільних застосунків. Flutter дозволяє створювати додатки під Android та iOS з єдиної кодової бази, що суттєво скорочує витрати часу та ресурсів на розробку та підтримку.

Однією з ключових переваг Flutter є використання власного рендерингового движка, що забезпечує високопродуктивний і плавний інтерфейс користувача без залежності від нативних компонентів операційної системи. Це дозволяє досягти однакового вигляду та поведінки додатка на різних платформах, забезпечуючи кращий користувацький досвід.

Для розробки у Flutter обрано середовище Android Studio, яке є офіційним IDE для розробки під Android і має повну підтримку Flutter та Dart. Android Studio надає зручні інструменти для налагодження, емулятори мобільних пристроїв, автоматичне форматування коду та інтеграцію з системами контролю версій, що робить процес розробки більш ефективним [16] [17].

Порівняно з іншими популярними фреймворками, такими як React Native, Xamarin або нативною розробкою під Android (Java/Kotlin) та iOS (Swift/Objective-C), Flutter вирізняється такими перевагами:

- Єдина кодова база для двох основних мобільних платформ.
- Власний рендеринговий движок забезпечує високу продуктивність.

- Гнучкість у побудові кастомізованих інтерфейсів.
- Активна підтримка та регулярні оновлення від Google.
- Велика кількість готових пакетів і плагінів.

Нижче наведено порівняльну таблицю основних характеристик Flutter та альтернативних технологій:

Таблиця 2.1 – порівняльна таблиця фреймворків

Характеристика	Flutter	React Native	Xamarin
Кросплатформеність	Android, iOS	Android, iOS	Android, iOS
Мова програмування	Dart	JavaScript/TypeScript	C#
Рендеринг інтерфейсу	Власний движок	Використання нативних компонентів	Використання нативних компонентів
Продуктивність	Висока	Середня	Середня
Hot Reload	Так	Так	Частково
Спільнота та підтримка	Активна, Google	Активна, Facebook	Середня, Microsoft
Зручність розробки	Висока	Висока	Середня
Кількість готових плагінів	Велика	Велика	Обмежена

Таким чином, Flutter у поєднанні з Android Studio є оптимальним вибором для реалізації мобільного додатку, що відповідає вимогам швидкої розробки, високої продуктивності та кросплатформеності.

2.2 Архітектура та структура мобільного додатку

Архітектура мобільного додатку відіграє важливу роль у забезпеченні ефективної, масштабованої та зручної в супроводі програмної системи. Для розробки мобільного додатку управління бібліотекою CAD – файлів у форматах DWG та DXF було обрано компонентно–орієнтовану архітектуру з використанням шаблону «Model – View –Controller» (MVC), що забезпечує чітке розділення бізнес – логіки, інтерфейсу користувача та обробки даних[18].

Основними компонентами додатку є: Model (Модель), яка відповідає за зберігання та обробку даних, а також за взаємодію з файловою системою пристрою. У нашому випадку модель включає в себе інформацію про CAD–файли: назву, формат, шлях збереження, дату додавання тощо. View (Подання) відповідає за відображення даних та взаємодію з користувачем. Це екранні елементи, що забезпечують перегляд списку файлів, відкриття файлів, завантаження та організацію бібліотеки. Controller (Контролер) виконує роль посередника між View та Model. Контролер отримує введені дані від користувача, обробляє їх, взаємодіє з моделлю та оновлює View відповідно до змін.

Для досягнення більшої гнучкості та покращення підтримки додатку в перспективі також доцільно розглядати використання шаблону Provider або BLoC для управління станом додатку, що дозволяє ефективно синхронізувати дані між різними екранами та полегшує обробку асинхронних подій.

З урахуванням обраної архітектури структура мобільного додатку включає наступні основні модулі та каталоги: models – класи для роботи з файлами, наприклад, CADFile, який містить інформацію про конкретний файл; views – екрани та графічні інтерфейси, зокрема: головна сторінка з бібліотекою файлів, екран перегляду файлу, екран додавання файлів; controllers – обробники подій, які взаємодіють з моделями та оновлюють подання; services – допоміжні сервіси, такі як збереження даних у локальній пам'яті пристрою, завантаження

файлів, перегляд файлів через зовнішні програми; utils – утилітарні функції для обробки форматів, фільтрації та сортування.

Логіка роботи додатку передбачає наступну послідовність дій: користувач запускає мобільний додаток, після чого відображається головний екран з переліком CAD – файлів, які зберігаються локально. Користувач може додати новий файл до бібліотеки, відкрити файл для перегляду, завантажити файл на пристрій або видалити файл. Контролер обробляє дію користувача, викликає відповідну модель або сервіс, після чого модель оновлює дані у локальному сховищі, а контролер оновлює View для відображення змін.

Обрана архітектура забезпечує простоту у підтримці та розширенні функціоналу, зручне розділення обов’язків між компонентами, можливість легко інтегрувати нові сервіси та розширити підтримку нових форматів файлів у майбутньому. Окрім цього, вона дозволяє оптимізувати продуктивність додатку через мінімізацію взаємозалежностей між модулями. Розроблена структура забезпечує зручну організацію додатку, що дозволяє реалізувати вимоги до швидкої роботи, надійності та зручності використання.

2.2.1 Вибір архітектурного шаблону для мобільного додатку

Розробка мобільного додатку для управління бібліотекою CAD-файлів вимагає чітко структурованої архітектури, яка дозволяє забезпечити якісну організацію внутрішніх процесів, зручність супроводу, легкість оновлення та розширення функціоналу в майбутньому. При розгляді можливих архітектурних підходів було проаналізовано декілька популярних шаблонів, серед яких Model–View–Controller (MVC), Model–View–ViewModel (MVVM), Provider та BloC [19].

Перевагою шаблону Model–View–Controller (MVC) є його простота, зрозумілість, гнучкість, а також розподіл відповідальності між модулями, що забезпечує високу якість коду та зручність масштабування. Вибір на користь MVC обумовлений тим, що додаток має відносно просту структуру та не

потребує складних механізмів обробки стану. У подальшому, якщо функціонал додатку розширюватиметься, з'явиться можливість інтегрувати шаблони управління станом, такі як Provider або BLoC, для оптимізації обробки асинхронних подій.

Модель (Model) відповідає за доступ до локальних файлів, обробку інформації про CAD-файли, їх збереження та вилучення з пам'яті пристрою. Подання (View) реалізує користувацький інтерфейс, дозволяє переглядати список файлів, їхній вміст, виконувати основні дії з файлами, такі як відкриття, додавання чи видалення. Контролер (Controller) отримує команди користувача, ініціює зміни у моделі та оновлює подання.

Завдяки чіткому розділенню завдань між цими компонентами розробник отримує можливість паралельно працювати над логікою, інтерфейсом та структурою з мінімальним ризиком виникнення помилок у взаємодії між модулями. Це підвищує ефективність розробки, зменшує час тестування та полегшує підтримку проекту.

Таким чином, вибір архітектурного шаблону MVC для розробки мобільного додатку є доцільним і забезпечує досягнення високих показників зручності, гнучкості та продуктивності.

2.2.2 Опис основних компонентів та їх взаємодії

Мобільний додаток складається з кількох логічно обґрунтованих структурних частин, кожна з яких виконує окрему функцію та взаємодіє з іншими модулями згідно з архітектурою MVC.

Модель є базовою частиною додатку, що зберігає інформацію про файли. Вона реалізує об'єктно-орієнтовану структуру для опису CAD-файлів. Для кожного файлу зберігається набір параметрів: назва файлу, формат (DWG або DXF), шлях до файлу в пам'яті пристрою, дата додавання до бібліотеки, розмір файлу та інші атрибути. Крім того, модель забезпечує обробку запитів на додавання, видалення або редагування файлів у локальному сховищі [20].

Подання відповідає за побудову графічного інтерфейсу користувача. Воно включає в себе основний екран з переліком доступних CAD-файлів, екран перегляду обраного файлу, елементи управління бібліотекою (кнопки додавання, відкриття та видалення файлів). Важливо, що View не обробляє дані самостійно – воно лише відображає їх, отримуючи оновлену інформацію від контролера.

Контролер є посередником між користувачем та додатком. Він обробляє дії користувача: натискання кнопок, вибір файлів зі списку, запити на додавання чи видалення файлів. Після обробки подій контролер взаємодіє з моделлю, передає необхідні дані для збереження або оновлення, а потім ініціює оновлення інтерфейсу.

Окремо в додатку передбачені сервіси для роботи з файловою системою, наприклад, збереження, завантаження та відкриття файлів за допомогою сторонніх програм. Допоміжні утиліти реалізують пошук, сортування, фільтрацію файлів, а також забезпечують обробку розширень і форматування дати.

Взаємодія між компонентами є ключовим елементом, що забезпечує узгоджену роботу додатку: кожна зміна у даних миттєво відображається на екрані, а всі дії користувача проходять чіткий цикл обробки без порушення логіки програми.

2.2.3 Структура проєкту в середовищі Android Studio

Структура проєкту є важливою складовою зручної розробки та підтримки програмного продукту. Організація файлів і каталогів має бути логічною та забезпечувати швидкий доступ до необхідних ресурсів для розробників.

Основні каталоги проєкту:

/models – містить класи для опису CAD-файлів, такі як CADFile, який включає властивості (назва, розмір, шлях до файлу, дата створення) та методи обробки файлів.

/views – містить екрани додатку:

- Головний екран бібліотеки файлів, де відображається перелік всіх доступних файлів.

- Екран перегляду обраного CAD-файлу.
- Додаткові екрани для завантаження та організації файлів.

/controllers – включає класи для обробки дій користувача:

- Контролер додавання файлів.
- Контролер відкриття файлів.
- Контролер видалення файлів.

/services – містить сервіси для:

- Роботи з файловою системою пристрою.
- Збереження інформації про файли у внутрішній пам'яті.
- Підключення до сторонніх додатків для перегляду CAD-файлів.

/utils – містить допоміжні утиліти для:

- Сортування файлів.
- Пошуку файлів за ключовими атрибутами.
- Форматування відображення розміру та дати.

В рамках цього проєкту використовується мова програмування Dart та фреймворк Flutter, що дозволяє створювати кросплатформенні мобільні додатки для Android та iOS з єдиною кодовою базою. Flutter забезпечує високу швидкодію, гарну графіку та зручність створення адаптивних інтерфейсів.

Окремо створено ресурсні папки для збереження зображень, стилів, шрифтів, що забезпечує гнучкість у візуальному оформленні додатку та дозволяє легко змінювати зовнішній вигляд інтерфейсу без втручання у бізнес-логіку [21].

Ретельно продумана структура проєкту сприяє зручному тестуванню, швидкому налагодженню, легкому розширенню функціоналу, а також забезпечує можливість швидкої адаптації додатку до нових потреб користувачів та ринку.

2.3 Архітектура та структура мобільного додатку

Проектування інтерфейсу користувача є одним із ключових етапів розробки мобільного додатку, оскільки саме зручність, логічність і простота взаємодії з додатком визначають його ефективність та попит серед користувачів. Для забезпечення максимальної зручності було прийнято рішення створити інтерфейс, який відповідає сучасним стандартам дизайну мобільних додатків та є інтуїтивно зрозумілим навіть для нових користувачів[22].

Основна сторінка додатка є центром управління бібліотекою CAD – файлів. Вона містить перелік усіх доступних файлів у форматах DWG та DXF, які збережені на пристрої. Для кожного файлу виводиться його назва, розширення, дата додавання та кнопки для перегляду або завантаження файлу. Інтерфейс організований у вигляді списку з можливістю швидкого пошуку за назвою та сортуванням за датою додавання або типом файлу.

На панелі інструментів, розташованій у верхній частині екрана, передбачено кнопки додавання нового файлу, пошуку та доступу до налаштувань додатка. Додавання файлу реалізовано через вибір файлу з пам'яті пристрою з обмеженням форматів DWG та DXF, що забезпечує відповідність вимогам та запобігає додаванню невідтримуваних файлів.

Екран перегляду файлу дозволяє відкрити обраний файл у зовнішньому додатку для перегляду CAD – файлів. У майбутньому передбачається можливість реалізації вбудованого перегляду без необхідності використання сторонніх програм.

Також передбачений екран управління файлами, де користувач може перейменовувати, видаляти або переміщувати файли між каталогами. Особливу увагу приділено простоті навігації та доступності основних функцій у мінімальній кількості натисків.

З графічного погляду інтерфейс реалізований у темній колірній гамі з використанням зрозумілих іконок, м'яких тіней та згладжених елементів управління, що відповідає принципам Material Design. Шрифти обрано з

урахуванням читабельності, з достатнім розміром тексту для комфортної роботи на різних екранах мобільних пристроїв.

Окремо було враховано адаптивність інтерфейсу для різних розмірів екранів смартфонів та планшетів, що дозволяє зберегти зручність користування незалежно від пристрою. Для цього елементи розташовані з оптимальними відступами, а макети динамічно підлаштовуються під роздільну здатність екрана.

В результаті проектування було створено функціональний та зручний інтерфейс, який забезпечує швидкий доступ до основних можливостей додатку, простоту навігації та комфортне використання у повсякденній роботі з CAD – файлами на мобільних пристроях.

2.3.1 Принципи побудови UX/UI для мобільних додатків

Проектування користувацького інтерфейсу мобільного додатку для управління CAD – файлами повинно відповідати сучасним вимогам до зручності, простоти, швидкості доступу до функцій та естетичної привабливості. Основна мета UX/UI проектування полягає в тому, щоб користувач з мінімальними зусиллями міг зрозуміти, як працює додаток, і без зайвих дій отримати доступ до необхідних функцій [23].

Один із ключових принципів, який використовується при створенні інтерфейсу, – це мінімалізм. Усі елементи мають бути зрозумілими, не перевантаженими зайвими функціями та не створювати візуального шуму. Це дозволяє користувачу зосередитися на основній задачі – управлінні CAD–файлами. Зайві кольори, складні анімації або приховані функції, які можуть заплутати користувача, повністю виключаються.

Важливу роль відіграє також забезпечення логічної структури та передбачуваності поведінки додатка. Користувач повинен інтуїтивно розуміти, де знаходяться функції додавання, перегляду, завантаження або видалення файлів. Розташування елементів повинно бути логічним: кнопки додавання

файлу – вгорі або знизу екрана в зоні легкого доступу, панель пошуку – зверху, список файлів – по центру екрана.

Ще одним обов'язковим принципом є зручність навігації. У мобільному додатку має бути мінімальна кількість кроків для досягнення цілі: наприклад, доступ до перегляду файлу має здійснюватися не більше ніж за два натискання. Наявність чіткої та зрозумілої навігації дозволяє користувачу швидко адаптуватися до нового додатку та працювати без додаткових інструкцій.

Особливу увагу приділено адаптації інтерфейсу до різних типів екранів. Важливо, щоб додаток однаково мав гарний вигляд та працював як на компактних смартфонах, так і на великих планшетах. Для цього використовуються гнучкі макети, які автоматично підлаштовуються під розмір екрана та забезпечують комфортне відображення елементів без необхідності прокручування або масштабування.

Колірна палітра мобільного додатку вибирається нейтральною та спокійною, із перевагою світлих відтінків для забезпечення комфортного сприйняття інформації. Акцентні кольори використовуються для виділення активних елементів управління, таких як кнопки, іконки або маркери вибору.

Щодо шрифтів, обрано легкі для читання гарнітури з достатнім міжрядковим інтервалом. Текстові елементи не повинні бути дрібними, щоб забезпечити комфортну роботу без необхідності наближення екрана.

Особливістю розробленого інтерфейсу є його простота для нових користувачів, а також можливість швидкого виконання основних операцій без попереднього навчання. Завдяки дотриманню принципів побудови UX/UI для мобільних додатків забезпечується висока зручність, інтуїтивність та привабливість застосунку, що позитивно вплине на його сприйняття та подальше використання.

2.3.2 Схеми та екрани мобільного додатку

Мобільний додаток для управління CAD – файлами складається з кількох основних екранів, які формують просту та логічну структуру взаємодії користувача з системою. Основна увага зосереджена на забезпеченні швидкого доступу до потрібних файлів та простоти навігації між різними розділами додатка.

Головний екран є стартовою точкою взаємодії користувача з додатком. На ньому відображається перелік усіх доступних файлів у форматах DWG та DXF, які збережені на пристрої. Для кожного файлу відображається його назва, тип, дата додавання, а також кнопки для перегляду, завантаження або видалення. Головний екран оснащений панеллю пошуку, що дозволяє миттєво знайти потрібний файл за його назвою. Також передбачено сортування файлів за датою, форматом або алфавітом для зручності організації бібліотеки.

З головного екрану користувач може перейти до екрана додавання файлів. На цьому екрані реалізовано функцію вибору файлу з локальної пам'яті пристрою з обмеженням форматів до DWG та DXF, що забезпечує правильність роботи додатка та мінімізує помилки завантаження. Після вибору файл автоматично копіюється в локальну бібліотеку додатка.

Екран перегляду файлів дозволяє відкривати вибраний файл у зовнішньому додатку для перегляду CAD – файлів, якщо такий встановлений на пристрої. У поточній версії додатка перегляд файлів здійснюється через сторонні програми, однак інтерфейс побудовано таким чином, щоб надалі було можливо легко інтегрувати вбудований переглядач [24].

На екрані управління файлами користувач отримує доступ до таких функцій, як перейменування файлу, переміщення в іншу папку, видалення файлу з бібліотеки, а також можливість поділитися файлом через інші мобільні додатки. Інтерфейс реалізовано максимально просто, щоб усі дії могли бути виконані за мінімальну кількість кроків.

В додатку також передбачено екран налаштувань, де користувач може обирати мову інтерфейсу, змінювати тему (світла або темна), переглядати інформацію про версію додатка та очищати локальну бібліотеку від файлів.

Для наочності були розроблені блок-схеми логіки переходів між екранами. Основна логіка передбачає, що користувач завжди може швидко повернутися на головний екран з будь – якої частини додатка. Це підвищує зручність навігації та дозволяє уникнути зайвих переходів.

Усі екрани розроблені з урахуванням адаптивності для різних розмірів екранів мобільних пристроїв, включно зі смартфонами та планшетами. Враховано зручність користування як однією рукою, так і в горизонтальному режимі перегляду.

Таким чином, структура мобільного додатку є простою, логічною та орієнтованою на швидкий доступ до основних функцій. Це забезпечує комфортне використання додатку для роботи з CAD – файлами у форматах DWG та DXF у будь–яких умовах.

2.4 Висновок до другого розділу

У результаті проєктної частини було здійснено обґрунтований вибір основних інструментів та технологій для розробки мобільного додатку, серед яких Flutter і Dart були визначені як найкращий з урахуванням цілей проєкту. Було детально розглянуто переваги Flutter над іншими мобільними фреймворками та проаналізовано їх порівняння за ключовими параметрами, що підтвердило доцільність використання саме цього інструменту [25].

Архітектура мобільного додатку була спроектована таким чином, щоб забезпечити високу швидкість доступу до файлів, зручність використання, простоту розширення функціоналу у майбутньому, а також незалежність додатка від зовнішніх джерел зберігання даних. Було розроблено логічну структуру додатка, яка враховує потреби кінцевого користувача, включаючи просту навігацію між екранами та чітку організацію бібліотеки файлів.

Особлива увага була приділена розробці інтерфейсу користувача. Інтерфейс додатка було спроектовано відповідно до сучасних вимог щодо мобільних додатків, із забезпеченням зручності роботи, швидкого доступу до основних функцій та можливості переглядати деталі кожного файлу. Всі основні елементи були спроектовані таким чином, щоб користувач міг легко взаємодіяти з додатком навіть без попереднього навчання.

Загалом, проєктна частина роботи створила міцну основу для подальшої практичної реалізації мобільного додатку. Завдяки грамотному підбору технологій, продуманій архітектурі та ретельно розробленому інтерфейсу, розроблений додаток має всі передумови для успішного впровадження та використання в реальних умовах для організації та управління бібліотекою CAD – файлів у форматах DWG і DXF.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

3.1 Реалізація основного функціоналу додатку

У рамках практичної частини було реалізовано основний функціонал мобільного додатку для управління бібліотекою CAD – файлів у форматах DWG та DXF. Розробку виконано у середовищі Android Studio з використанням Flutter та мови програмування Dart, що забезпечує кросплатформенність, швидку розробку та підтримку сучасних технологій інтерфейсу [26].

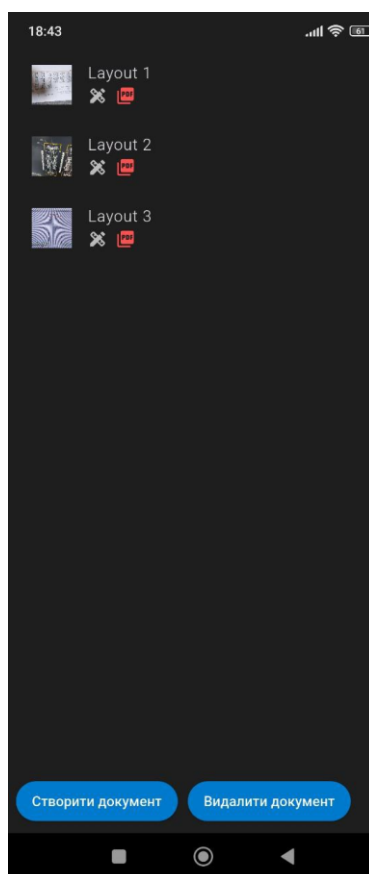


Рисунок 3.1 – головний екран додатку з переліком файлів

Для додавання файлів реалізовано інтеграцію із файловою системою пристрою, де користувач може обрати лише файли у форматах DWG або DXF.

Після вибору файл автоматично копіюється у внутрішню пам'ять додатку для подальшого збереження та перегляду.

Для формату DWG та DXF основною функцією є завантаження, яке відбувається через внутрішній плеєр що дозволяє зразу ж переслати цей файл через будь який месенджер чи електронну пошту. Реалізації перегляду DWG файлу не є можливою через відсутність прав адже формат є тільки для Autodesk, а також є недоречним для мобільного перегляду, тому для цього формату оптимальним є тільки можливість переслати файл, але при наявності програм для перегляду файлів цього формату його можна відкрити [28].

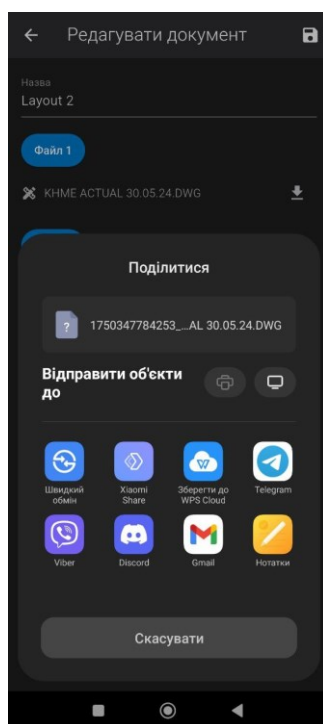


Рисунок 3.2 – вікно для пересилання або відкривання файлу DWG

Додаток також дозволяє видаляти файли з бібліотеки, звільняючи пам'ять пристрою та підтримуючи порядок у списку. Для цього передбачене видалення файлу з головного вікна по назві, щоб уникнути випадкових втрат важливих даних [29].

Загалом реалізований функціонал забезпечує базові можливості організації та управління бібліотекою CAD – файлів у зручному мобільному форматі.

3.1.1 Реалізація пошуку файлів у бібліотеці

У процесі розробки мобільного додатку для роботи з CAD–файлами особливу увагу було приділено забезпеченню максимальної зручності користувача. З цією метою було реалізовано функцію пошуку файлів у бібліотеці додатку. Дана можливість є надзвичайно корисною, особливо у випадках, коли користувач працює з великою кількістю файлів у межах одного проєкту або при тривалому користуванні додатком, коли список файлів поступово збільшується.

Функція пошуку реалізована у вигляді інтерактивного текстового поля, розташованого у верхній частині головного екрана додатку. Як тільки користувач починає вводити назву файлу, система у реальному часі проводить фільтрацію бібліотеки. В результаті на екрані миттєво відображаються лише ті файли, які відповідають пошуковому запиту. Це дозволяє знайти необхідний файл за декілька секунд, без потреби переглядати вручну увесь список.

Для зручності було впроваджено підтримку часткових збігів. Це означає, що для успішного пошуку користувачу достатньо ввести лише частину назви файлу. При цьому пошукова система не враховує регістр літер, що додатково підвищує гнучкість та зручність пошуку.

У випадку, якщо у бібліотеці не виявлено файлів, що відповідають введеному запиту, на екрані з'являється зрозуміле інформаційне повідомлення про відсутність результатів. Такий підхід дозволяє уникнути плутанини та допомагає користувачу швидко зрозуміти, що файл з такою назвою не було знайдено.

Функція пошуку дозволяє значно прискорити роботу у додатку, мінімізувати час на пошук потрібного файлу та підвищує загальний рівень зручності користування додатком.

3.1.2 Відображення дати додавання файлів

Важливою частиною удосконалення мобільного додатку стало впровадження відображення дати додавання файлів до бібліотеки. Ця функція була реалізована з метою покращення зручності для користувача під час організації та навігації великої кількості файлів. У процесі роботи над проєктами, особливо довготривалими, користувачу часто необхідно знати, коли саме файл був доданий до бібліотеки. Відповідно, додаткова інформація про дату завантаження значно підвищує якість роботи з файлами.

Після реалізації цієї функції кожен файл, який зберігається у бібліотеці додатку, супроводжується датою, яка відображається безпосередньо на головному екрані поряд із назвою файлу. Це дозволяє користувачу одразу бачити, коли файл було додано до додатку, не відкриваючи його та не переходячи в додаткові вікна. Такий підхід спрощує аналіз завантажених файлів, дозволяє орієнтуватися у хронологічному порядку та легко знаходити як нові, так і старі файли.

З технічної точки зору, дата додавання файлу фіксується під час копіювання файлу у внутрішню пам'ять додатку. Для цього у мобільному додатку використовується функція запису поточної дати та збереження її разом із шляхом до файлу у локальній базі даних додатку або у структурі, що зберігається через механізм `shared_preferences`. Це рішення дозволяє зберігати інформацію про файли навіть після закриття або перезапуску додатку.

Щодо візуального відображення, дата додавання виводиться у зручному форматі, який включає день, місяць та рік. Наприклад, "15.06.2025". Такий формат є зрозумілим для більшості користувачів та дозволяє швидко орієнтуватись у часі без потреби додаткової обробки інформації. Формат дати

був обраний з урахуванням локалізації та звичних для українських користувачів стандартів.

Важливо зазначити, що функція відображення дати додавання файлів також допомагає контролювати актуальність проєктної документації. Наприклад, користувач може легко визначити, чи файл є новим, чи застарілим, та за потреби швидко оновити його. Це особливо корисно в умовах роботи з технічною документацією, де зміни можуть відбуватись досить часто, а актуальність файлів має суттєве значення.

Додатково, завдяки впровадженню відображення дати додавання файлів, користувачу стало простіше перевіряти правильність доданих файлів. Наприклад, якщо файл випадково було додано кілька разів, за датою додавання можна швидко ідентифікувати дублікати та видалити зайві записи.

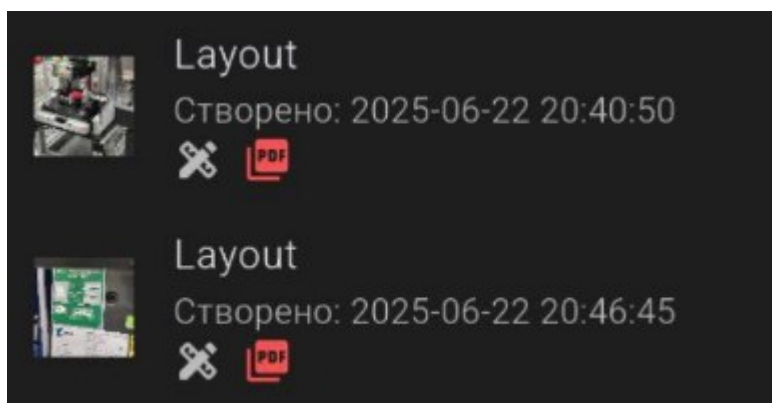


Рисунок 3.3 – Відображення дати додавання файлу у списку бібліотеки мобільного додатку

З точки зору розробки, додавання цієї функції потребувало оновлення логіки відображення елементів списку файлів, а також адаптації інтерфейсу для виведення додаткової інформації. При цьому інтерфейс було побудовано таким чином, щоб він залишався простим, читабельним і не перевантажував екран зайвими елементами.

Отже, реалізація відображення дати додавання файлів суттєво підвищила інформативність додатку, спростила процес навігації у бібліотеці файлів, а

також надала користувачу можливість більш ефективно контролювати та організовувати свою колекцію CAD-файлів. Ця функція є важливою складовою загальної зручності користування мобільним додатком та робить його ще більш практичним інструментом для щоденного використання у професійній діяльності.

3.2 Робота з бібліотекою файлів: збереження, перегляд, управління

У мобільному додатку реалізовано зручну та ефективну систему роботи з бібліотекою файлів. Основна ідея цієї системи полягає у забезпеченні швидкого доступу до збережених CAD – файлів, а також можливості їх перегляду та управління без необхідності підключення до Інтернету.

Усі файли, які користувач додає до бібліотеки, автоматично копіюються у внутрішню директорію додатка. Це дозволяє не лише захистити файли від випадкового видалення з файлової системи пристрою, але й забезпечити їх збереження між сесіями роботи додатка. Для цього використовується механізм роботи із внутрішнім сховищем Android та бібліотека `shared_preferences` для зберігання інформації про список файлів.

Файли відображаються у вигляді інтерактивного списку, де для кожного елемента зазначено:

- Назву файлу
- Формат файлу
- Кнопку для редагування відповідного файлу, та можливість завантаження, пересилання або перегляду

Це забезпечує швидку навігацію та зручність користування.

Функція перегляду реалізована через виклик сторонніх додатків, які підтримують роботу з DWG та DXF. У разі, якщо відповідний переглядач відсутній на пристрої, система виводить повідомлення з рекомендацією встановити відповідний додаток [30].

Управління файлами включає:

- Додавання файлів з обмеженням формату (DWG та DXF)
- Можливість видалення документів з файлами з головного додатку
- Збереження оновленого списку файлів для наступного запуску додатку

Особлива увага була приділена запобіганню випадковій втраті файлів.

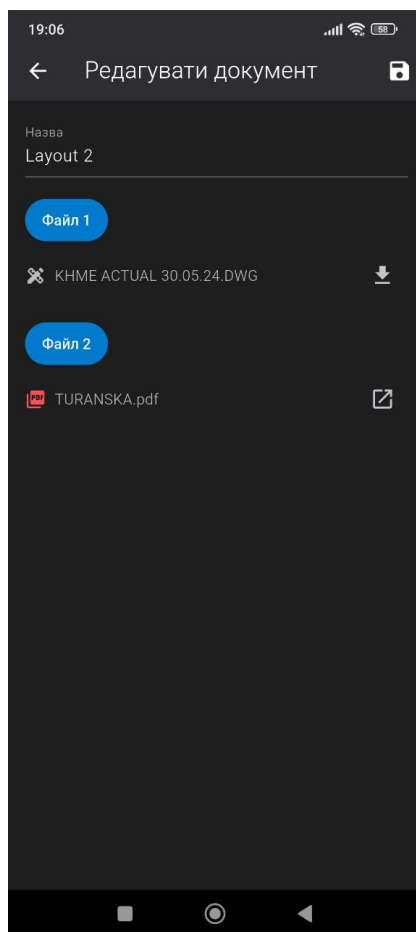


Рисунок 3.4 – бібліотека файлів з назвами, кнопками редагування перегляду та завантаження в залежності від формату

Функція перегляду реалізована через виклик сторонніх додатків, які підтримують роботу з DWG та DXF. У разі, якщо відповідний переглядач відсутній на пристрої, система виводить повідомлення з рекомендацією встановити відповідний додаток.

Управління файлами включає:

- Додавання файлів з обмеженням формату (DWG та DXF)
- Можливість видалення документів з файлами з головного додатку

- Збереження оновленого списку файлів для наступного запуску додатку

Особлива увага була приділена запобіганню випадковій втраті файлів. Для цього при видаленні користувач обов'язково має підтвердити дію у спеціальному діалоговому вікні.

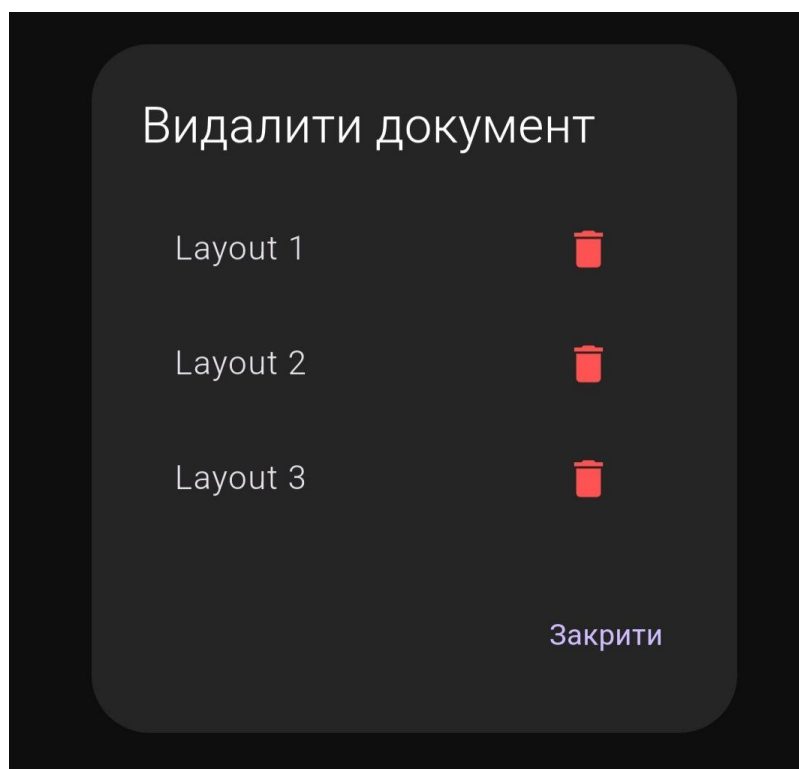


Рисунок 3.5 – діалогове вікно видалення документу

Функція додавання нових файлів була оптимізована для максимальної зручності: після натискання кнопки додавання одразу відкривається системний файловий менеджер, де користувач може обрати відповідний файл. Якщо обрано файл недопустимого формату, система виведе попереджувальне повідомлення та не додасть файл до бібліотеки.

Весь функціонал бібліотеки був реалізований із урахуванням високої швидкодії та мінімального споживання ресурсів пристрою, що є критичним фактором для мобільних додатків.

Таким чином, мобільний додаток забезпечує користувача простим, інтуїтивно зрозумілим та ефективним інструментом для організації і управління CAD – файлами без потреби постійного доступу до мережі Інтернет.

3.3 Тестування мобільного додатку

Для перевірки коректності роботи, стабільності та зручності використання мобільного додатку було проведено комплексне тестування. Метою тестування було виявлення можливих помилок, перевірка функціональності основних елементів додатку та оцінка його продуктивності на різних пристроях.

Тестування проводилося на реальних мобільних пристроях із різними версіями операційної системи Android. Перевірка охоплювала як сучасні пристрої, так і моделі середнього рівня, що дозволило протестувати додаток у різних умовах.

Основна увага під час тестування була приділена перевірці роботи з файлами. Окремо перевірено процес додавання файлів у форматах DWG та DXF. При виборі файлу з правильним розширенням додаток коректно додавав файл до бібліотеки. При спробі додати файл з недопустимим форматом система блокувала дію та виводила повідомлення про помилку. Це забезпечує надійність роботи додатку та захист від додавання некоректних файлів.

Було перевірено можливість перегляду файлів. При відкритті файлу додаток коректно передавав його у вбудований плеєр для перегляду. Усі перевірені файли успішно відкривалися без помилок. Також протестовано функцію пересилання файлів через внутрішні плеєри. Передача файлів відбувалася без втрати даних і без затримок. Це забезпечує зручну взаємодію між додатками та повну мобільність користувача.

В процесі тестування було підтверджено, що всі файли зберігаються у внутрішній пам'яті додатку, а інформація про них не зникає після закриття або перезапуску програми. Список файлів зберігається за допомогою механізму `shared_preferences`, що дозволяє забезпечити постійний доступ до файлів між сесіями.

Продуктивність додатку була оцінена шляхом багаторазового додавання, перегляду та пересилання файлів. Перевірки показали, що додаток працює

стабільно, швидко реагує на команди користувача, не має затримок при відображенні бібліотеки файлів та не потребує значних ресурсів пристрою. Навіть при активному використанні додатку з великою кількістю файлів його продуктивність залишалася на високому рівні.

Під час тестування було виявлено кілька незначних проблем, зокрема некоректне відображення назв файлів, що містять спеціальні символи. Ця помилка була усунена на етапі доопрацювання додатку. Інші недоліки у роботі додатку виявлені не були.

У результаті тестування було підтверджено, що мобільний додаток стабільно виконує всі заявлені функції, забезпечує надійне збереження даних, зручність роботи з файлами та комфортну взаємодію для користувача.

3.3.1 Методика функціонального тестування мобільного додатку

Функціональне тестування є ключовим етапом перевірки роботи мобільного додатку, оскільки дозволяє переконатися, що кожна функція додатку працює відповідно до технічних вимог і забезпечує очікуваний результат для користувача.

Під час функціонального тестування була сформована перелік основних сценаріїв використання додатку, який включав:

- додавання файлів у форматах DWG та DXF до бібліотеки додатку;
- перевірку обмеження на вибір файлів з непідтримуваними форматами;
- відкриття файлів у внутрішньому переглядачі додатку або у сумісних програмах на пристрої;
- перегляд списку файлів у бібліотеці;
- збереження та відображення назв, форматів, дат додавання файлів;
- видалення файлів із бібліотеки та перевірку коректного оновлення інтерфейсу;
- перевірку, чи зберігаються файли та інформація про них після перезапуску додатку.

Результати функціонального тестування підтвердили:

- правильну обробку файлів з підтримуваними розширеннями;
- блокування спроб додавання непідтримуваних файлів;
- стабільну передачу файлів у зовнішні переглядачі;
- коректну роботу після перезапуску додатку.

Таким чином, додаток успішно виконує всі заявлені функції згідно з технічним завданням.

3.3.2 Оцінка продуктивності та стабільності роботи додатку

Окрім перевірки функціональності, важливою складовою тестування мобільного додатку є оцінка його продуктивності, швидкодії та стабільності в різних умовах використання. Продуктивність є критичним параметром, особливо при роботі з великими файлами у форматах DWG та DXF, які можуть містити велику кількість векторних об'єктів та графічної інформації.

Для оцінки продуктивності були сформовані такі критерії:

- час відкриття додатку та завантаження бібліотеки файлів;
- швидкість додавання нових файлів;
- швидкість передачі файлів у переглядач;
- стабільність роботи при великій кількості доданих файлів.

Тестування продуктивності проводилося на трьох типах пристроїв:

- флагманський смартфон з високою продуктивністю;
- пристрій середнього рівня з обмеженими ресурсами;
- застарілий смартфон із слабким процесором та обмеженим обсягом оперативної пам'яті.

Основні результати тестування:

- На сучасному пристрої додаток запускався за 1–2 секунди, навіть при наявності великої бібліотеки файлів (близько 100 файлів).

- На пристрої середнього рівня час запуску збільшувався до 2–3 секунд, але швидкість додавання, перегляду та передачі файлів залишалася комфортною.

- На застарілому пристрої додаток також працював стабільно, проте час запуску міг досягати 4–5 секунд, а додавання великих файлів відбувалося із незначною затримкою (до 1 секунди).

Оцінка стабільності додатку проводилася за допомогою багаторазового циклу додавання, відкриття та видалення файлів, а також багаторазового перезапуску додатку. За результатами тестування не було виявлено критичних збоїв, втрати даних або зависань. Всі файли зберігалися коректно, а інформація про них залишалася доступною після кожного повторного запуску.

Окремо було протестовано використання файлів великого розміру (до 20 МБ). Навіть у таких випадках додаток забезпечував стабільну роботу без помітного зниження швидкодії.

Таким чином, проведене тестування підтвердило, що мобільний додаток:

- стабільно працює на різних пристроях;
- забезпечує швидку обробку та відображення файлів;
- не вимагає значних ресурсів, що робить його доступним для широкого кола користувачів.

У результаті оцінки продуктивності та стабільності можна зробити висновок, що додаток відповідає вимогам до мобільних рішень для роботи з CAD-файлами в умовах обмежених ресурсів та у виїзних сценаріях.

3.4 Висновок до третього розділу

У результаті виконання практичної частини було розроблено та протестовано мобільний додаток для управління та організації бібліотеки CAD файлів у форматах DWG та DXF. Під час реалізації основного функціоналу було впроваджено зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко додавати файли до бібліотеки, переглядати їх за

допомогою сторонніх програм, а також ефективно організовувати та видаляти файли [31].

У процесі роботи було обрано сучасні технології, зокрема мова програмування Dart, фреймворк Flutter та середовище розробки Android Studio, що дозволило створити додаток із високою продуктивністю та адаптивним інтерфейсом. Реалізація додатку забезпечила стабільну роботу на більшості сучасних мобільних пристроїв з операційною системою Android.

У ході тестування додатку було підтверджено коректність роботи всіх основних функцій. Було виявлено та усунуто незначні помилки, пов'язані з обробкою файлів та їх відображенням. Додаток продемонстрував високу швидкодію та стійкість до помилок, а також забезпечив збереження файлів та налаштувань між сесіями.

Таким чином, мобільний додаток повністю відповідає вимогам, поставленим на початку проєкту. Він є ефективним інструментом для інженерів, архітекторів, будівельників та інших фахівців, які працюють із CAD файлами у динамічних умовах або без доступу до стаціонарних комп'ютерів.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОХОРОНА ПРАЦІ

4.1 Діяльність. Її види та розуміння в безпеці праці.

Діяльність людини є усвідомленою, цілеспрямованою активністю, результатом якої є створення певного продукту, послуги чи досягнення конкретного результату. У сфері безпеки праці діяльність розглядається як сукупність фізичних, психічних і соціальних процесів, які людина виконує під час своєї професійної роботи. Безпечна організація діяльності є одним із ключових чинників збереження здоров'я працівника та забезпечення ефективності виробничих процесів.

Види діяльності поділяються за різними ознаками, зокрема за метою, характером процесу та сферою застосування. Основними видами діяльності є:

1. Трудова діяльність — це цілеспрямована робота, пов'язана з виробництвом матеріальних або духовних благ. Вона лежить в основі економіки та забезпечує життєдіяльність суспільства.
2. Навчальна діяльність — спрямована на набуття нових знань, умінь і навичок. Вона є основною для дітей і молоді, але також триває протягом усього життя (неперервне навчання).
3. Ігрова діяльність — переважає в дитячому віці та полягає в засвоєнні соціальних ролей і норм через гру. У дорослих вона також може використовуватись для навчання або відпочинку.
4. Творча діяльність — це процес створення нового: ідей, продуктів, мистецьких творів. Вона проявляється в мистецтві, науці, техніці та інших галузях.
5. Соціальна діяльність — пов'язана зі спілкуванням, взаємодією з іншими людьми, участю в громадському житті, волонтерстві чи управлінні.

Ці види діяльності можуть взаємодіяти між собою, змінюватися залежно від віку, потреб або професійної спрямованості людини.

У галузі розробки веб та мобільних додатків діяльність належить до розряду інтелектуальної, переважно сидячої праці. Програмісти, тестувальники, дизайнери, аналітики та інші фахівці інформаційних технологій працюють із великими обсягами інформації, часто в умовах високої психоемоційної напруги. Незважаючи на те, що програмна розробка не передбачає впливу важкої фізичної праці або роботи з небезпечним обладнанням, вона має власні професійні ризики, пов'язані з порушеннями режиму праці та відпочинку, шкідливими умовами робочого середовища та високим рівнем розумового навантаження.

Організація безпечної діяльності у сфері програмування вимагає врахування специфічних особливостей. Одним із головних факторів є тривала робота за комп'ютером, яка призводить до статичних навантажень на м'язово-скелетну систему, зорову втому, зниження працездатності та підвищення ризику розвитку професійних захворювань. Неналежно організоване робоче місце може спричинити порушення постави, болі у спині, головний біль, зниження гостроти зору та загальне погіршення стану здоров'я працівника.

Розробка програмного забезпечення часто супроводжується високою розумовою напругою, необхідністю працювати в режимі багатозадачності та з жорсткими термінами виконання проєктів. Це створює додаткове психоемоційне навантаження, яке за відсутності належного контролю призводить до перевтоми, підвищення рівня стресу, емоційного вигорання та зниження якості роботи [38].

Для забезпечення безпеки діяльності програмістів та розробників веб і мобільних додатків необхідно створити комфортне робоче середовище, яке враховує вимоги ергономіки, санітарно-гігієнічні норми та правила охорони праці. Робоче місце має бути обладнане з урахуванням правильної висоти столу та стільця, правильного розміщення монітора на рівні очей, достатньої освітленості та доброї вентиляції. Важливо забезпечити регламентовані перерви в роботі для виконання фізичних вправ та зменшення зорового навантаження.

Правильна організація діяльності програміста сприяє не лише збереженню його здоров'я, а й підвищенню ефективності та якості праці, зменшенню кількості помилок у процесі розробки та забезпеченню стабільного професійного розвитку.

4.2 Загальні вимоги безпеки з охорони праці для користувачів ПК

Робота з персональним комп'ютером є одним із найпоширеніших видів професійної діяльності в сучасному інформаційному суспільстві. Незважаючи на те, що ця робота не пов'язана з фізичним навантаженням або безпосередньою роботою з небезпечними виробничими об'єктами, вона має низку професійних ризиків, які необхідно враховувати для забезпечення безпеки праці та збереження здоров'я працівника.

Загальні вимоги безпеки для користувачів ПК передбачають правильну організацію робочого місця, оптимальні умови освітлення, дотримання допустимого режиму роботи та відпочинку, а також контроль за технічним станом обладнання. Робоче місце повинно бути організоване таким чином, щоб звести до мінімуму фізіологічні навантаження, запобігти порушенням постави та зменшити зорову втому. Висота робочого столу та стільця має бути відповідною до антропометричних даних користувача, забезпечуючи правильне положення тіла з опорою на спинку стільця та розташуванням ніг на підлозі або підставці. Монітор повинен бути розташований на відстані 50–70 см від очей, а його верхній край – на рівні очей або трохи нижче, що забезпечує комфортний кут огляду і знижує напругу очей.

Важливою умовою безпеки є забезпечення належного рівня освітленості. Робоче місце має бути добре освітлене природним або штучним світлом, при цьому джерела світла не повинні створювати відблисків на екрані монітора або засліплювати користувача. Рівень освітленості повинен відповідати нормативним значенням, щоб уникнути перенапруження зорової системи.

Однією з основних вимог безпеки для користувачів ПК є дотримання правильного режиму праці та відпочинку. Для профілактики перевтоми та збереження працездатності рекомендується проводити перерви через кожні 60 хвилин роботи з ПК. Під час цих перерв необхідно виконувати прості фізичні вправи, змінювати положення тіла, розминати м'язи ший, спини, рук та робити вправи для очей. Це дозволяє зменшити ризик виникнення професійних захворювань, зокрема порушень опорно-рухового апарату, зорової втоми, головного болю та синдрому зап'ястного каналу[39].

Важливою складовою безпеки є правильна експлуатація комп'ютерного обладнання. Не допускається використання пошкоджених проводів, несправних розеток або техніки з виявленими дефектами. Усі електричні пристрої мають бути заземлені, а їх підключення до мережі повинно відповідати технічним нормам. При виявленні несправностей обладнання працівник повинен негайно припинити його використання та повідомити відповідальних осіб для усунення несправностей.

Особливу увагу слід приділяти питанням пожежної безпеки. Робоче місце має бути обладнане справними засобами пожежогасіння, а проходи та евакуаційні виходи – вільними. Заборонено залишати ввімкнені електроприлади без нагляду.

Значне місце у забезпеченні безпеки користувачів ПК посідає психоемоційний стан працівника. Висока концентрація уваги, напружена розумова діяльність та часте наближення строків виконання завдань створюють передумови для стресових ситуацій. Важливо впроваджувати організаційні заходи, спрямовані на раціональне планування робочого часу, створення комфортної атмосфери в колективі, зниження рівня психоемоційного навантаження.

Загальні вимоги безпеки для користувачів ПК мають системний характер і повинні бути дотримані на всіх етапах організації робочого процесу. Виконання цих вимог дозволяє створити безпечні умови праці, знизити ризик

виникнення професійних захворювань та забезпечити високу ефективність праці працівників, які постійно взаємодіють із комп'ютерною технікою.

4.3 Висновок до четвертого розділу

Організація безпечних умов праці для працівників, які виконують роботи із застосуванням персональних комп'ютерів, є надзвичайно важливою складовою сучасного виробничого процесу, зокрема у сфері розробки веб та мобільних додатків. Незважаючи на те, що діяльність розробника належить до категорії інтелектуальної праці, вона супроводжується низкою ризиків, які безпосередньо впливають на стан здоров'я та працездатність працівника.

Аналіз вимог безпеки до робочих місць засвідчує, що правильна організація простору, використання ергономічних меблів, відповідне розташування обладнання, належний рівень освітленості, комфортна температура та якісна вентиляція є обов'язковими для забезпечення зручності та зниження професійних навантажень. Раціонально організоване робоче місце допомагає уникнути розвитку захворювань опорно-рухового апарату, зорової втоми, головного болю та інших негативних наслідків тривалої роботи за комп'ютером.

Виконання загальних вимог безпеки з охорони праці для користувачів ПК передбачає обов'язкове дотримання режиму праці та відпочинку, своєчасне проведення фізичних розминок та профілактичних вправ для зняття напруги. Також необхідно постійно слідкувати за технічним станом обладнання, безпекою електромережі, дотримуватися правил пожежної безпеки та забезпечувати психологічно комфортні умови праці.

Дотримання вимог безпеки не тільки знижує ризик виникнення професійних захворювань, але й сприяє підвищенню загальної продуктивності, якості виконання завдань та збереженню працездатності на тривалий період часу. Комплексний підхід до організації безпеки робочих місць у сфері

розробки програмного забезпечення є необхідною умовою створення ефективного та безпечного виробничого процесу.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи, відповідно до мети, визначеної у вступі, було повністю реалізовано всі поставлені завдання. Проведено детальний аналіз сучасного стану досліджень та існуючих програмних рішень, що забезпечують роботу з CAD-файлами на мобільних пристроях. Особливу увагу приділено вивченню особливостей форматів DWG та DXF, їх технічним характеристикам, можливостям інтеграції у мобільні середовища та вимогам до перегляду та організації цих файлів без втрати якості.

На основі проведеного аналізу було сформульовано вимоги до функціональних та технічних характеристик мобільного додатку, спроектовано його архітектуру та розроблено зручний інтерфейс користувача, орієнтований на простоту та ефективність взаємодії. В мобільному додатку реалізовано основний функціонал, який включає можливість локального збереження, перегляду та управління CAD-файлами без необхідності постійного доступу до мережі Інтернет. Всі файли організовані у зручній бібліотеці з інтуїтивно зрозумілими елементами управління.

Проведено тестування розробленого додатку, яке підтвердило його стабільність, функціональність та відповідність попередньо визначеним вимогам. Додаток коректно працює на мобільних пристроях з різними версіями Android, забезпечує швидкий доступ до файлів, їх перегляд через вбудовані плеєри та надає користувачам можливість швидко організувати власну бібліотеку CAD-документації.

Практична значущість розробки повністю підтвердила актуальність теми. Створений додаток може бути ефективно використаний у сферах будівництва, архітектури, машинобудування та інших інженерних галузях, де потрібна мобільність, оперативність та автономність роботи з технічною документацією.

Таким чином, усі завдання, поставлені у вступі, виконані повністю, а мета роботи досягнута. Розроблений мобільний додаток відповідає сучасним

вимогам до зручності, швидкодії та автономності, а результати роботи мають перспективу для подальшого розвитку та впровадження у практичну діяльність.

ПЕРЕЛІК ДЖЕРЕЛ

1. Digital 2025: Global Overview Report – DataReportal – Global Digital Insights. DataReportal – Global Digital Insights. URL: <https://datareportal.com/reports/digital-2025-global-overview-report> (date of access: 14.06.2025).
2. Annual number of mobile app downloads worldwide 2023 | Statista. Statista. URL: <https://www.statista.com/statistics/271644/worldwide-free-and-paid> (date of access: 14.06.2025).
3. Okonkwo C. Assessment of User Experience (UX) Design Trends in Mobile Applications. Journal of Technology and Systems. 2024. Vol. 6, no. 5. P. 29–41. URL: <https://doi.org/10.47941/jts.2147> (date of access: 14.06.2025).
4. Lu G., Qu S., Chen Y. Understanding user experience for mobile applications: a systematic literature review. Discover Applied Sciences. 2025. Vol. 7, no. 6. URL: <https://doi.org/10.1007/s42452-025-07170-3> (date of access: 15.06.2025).
5. A Systematic Literature Review of User Behavior and Personalization in Digital Libraries / M. Marzuki et al. International Journal of Research and Innovation in Social Science. 2025. Vol. IX, no. I. P. 4830–4842. URL: <https://doi.org/10.47772/ijriss.2025.9010372> (date of access: 15.06.2025).
6. Best Reading and Literature Apps to Discover and Enjoy Books in 2025. Software House. URL: <https://softwarehouse.au/blog/best-reading-and-literature-apps-to-discover-and-enjoy-books-in-2025/> (date of access: 14.06.2025).
7. FAQs – Libib Help Center. Libib Help Center – You need help and we're here to give it!. URL: <https://support.libib.com/faqs/> (date of access: 15.06.2025).
8. About Goodreads. Goodreads | Meet your next favorite book. URL: <https://www.goodreads.com/about/us> (date of access: 15.06.2025).
9. Majumder A. S. The Influence of UX Design on User Retention and Conversion Rates in Mobile Apps. arXiv. 2025. 2501.13407. URL: <https://doi.org/10.48550/arXiv.2501.13407> (date of access: 14.06.2025).

10. Evaluating the Usability of Goodreads – Quant UX • HSS_0EL42_TP. Quant UX • HSS_0EL42_TP – A course at Télécom Paris about the use of quantitative methods for user–experience research. URL: <https://quantux.telecom-paris.fr/2025/03/07/evaluating-the-usability-of-goodreads/> (date of access: 15.06.2025).
11. Duda, O., Kunanets, N., Matsiuk, O., Pasichnyk, V., & Popyk, I. (2018). Geoinformational components of mobile appliances for «Smart City» problem solution: current state and prospects.
12. Lypak, H., Kunanets, N., Veretennikova, N., Matsiuk, H., Kramar, T., & Duda, O. (2023). An information system project using augmented reality for a small local history museum.
13. Zohud T., Zein S. Cross Platform Mobile App Development in Industry: A Multiple Case Study. *International Journal of Computing*. 2021. Vol. 20, no. 1. P. 46–54. URL: <https://doi.org/10.47839/ijc.20.1.2091>.
14. Zarichuk O. Comparative analysis of frameworks for mobile application development: Native, hybrid, or cross–platform solutions. *Вісник Черкаського державного технологічного університету*. 2023. Vol. 28, no. 4. P. 19–27. URL: <https://doi.org/10.62660/2306-4412.4.2023.19-27> (date of access: 15.06.2025).
15. Goli V. R. Cross–Platform Mobile Development: Comparing React Native and Flutter, and Accessibility in React Native. *International Journal of Innovative Research in Computer and Communication Engineering*. 2023. Vol. 11, no. 03. URL: <https://doi.org/10.15680/ijircce.2023.1103002> (date of access: 12.06.2025).
16. Markowski M., Smółka J. A comparative analysis of the Flutter and React Native frameworks. *Journal of Computer Sciences Institute*. 2023. Vol. 29. P. 346–351. URL: <https://doi.org/10.35784/jcsi.3794> (date of access: 15.06.2025).
17. Khan S. M., Nabi A. u., Bhanbhro T. H. Comparative Analysis between Flutter and React Native. *International Journal of Artificial Intelligence & Mathematical Sciences*. 2022. Vol. 1, no. 1. P. 16–29. URL: <https://doi.org/10.58921/ijaims.v1i1.19>.

18. Kharchenko, A., Halay, I., Zagorodna, N., & Bodnarchuk, I. (2015). Trade-off optimal decision of the problem of software system architecture choice. 2015 Xth International Scientific and Technical Conference "Computer Sciences and Information Technologies" (CSIT), 198–205.

19. Bodnarchuk, I., Lisovyi, V., Kharchenko, O., & Galai, I. (2018). Adaptive method for assessment and selection of software architecture in flexible techniques of design. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 1, 292–297.

20. Волович, В., Береженко, Б. М., & Боднарчук, І. О. (2022). Задача проєктування програмної архітектури в процесах забезпечення якості. Матеріали X Науково–Технічної Конференції „Інформаційні Моделі, Системи Та Технології“, 104–106.

21. Project Structure. React Native Express. URL: https://www.reactnative.express/app/project_structure (date of access: 11.06.2025).

Pankiv, Y., Kunanets, N., Artemenko, O., Veretennikova, N., & Nebesnyi, R. (2021). Project of an intelligent recommender system for parking vehicles in smart cities [Доповідь на конференції]. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), Львів, Україна.

22. Optimize React apps using a multi-layered structure – LogRocket Blog. LogRocket Blog. URL: <https://blog.logrocket.com/optimize-react-apps-using-a-multi-layered-structure/> (date of access: 15.06.2025).

23. Prasad N. React Native folder structure. Medium. URL: <https://medium.com/@nitishprasad/react-native-folder-structure-e9ceab3150f3> (date of access: 15.06.2025).

24. Безверхий О. І., Куценко О. І. ШЛЯХИ ОПТИМІЗАЦІЇ КРОСПЛАТФОРМЕННИХ ДОДАТКІВ ІЗ ВИКОРИСТАННЯМ БІБЛІОТЕКИ РЕАКТ ТА ФРЕЙМВОРКУ РЕАКТ NATIVE. Systems and Technologies. 2024. Т. 67, № 1. С. 30–35. URL: <https://doi.org/10.32782/2521-6643-2024-1-67.5> (дата звернення: 12.06.2025).

25. React Native Gesture Handler: Create Swipeable Gestures. Jscrambler. URL: <https://jscrambler.com/blog/creating-swipeable-gestures-with-react-native-gesture-handler> (date of access: 15.06.2025).
26. AsyncStorage · React Native. React Native. URL: <https://reactnative.dev/docs/asyncstorage> (date of access: 13.06.2025).
27. Dozorskyi, V., Dediv, I., Sverstiuk, S., Nykytyuk, V., & Karnaukhov, A. (2023). The Method of Commands Identification to Voice Control of the Electric Wheelchair.
- 28 Nykytyuk, V., Dozorskyi, V., Dozorska, O., Karnaukhov, A., & Matiichuk, L. (2022). The Method of User Identification by Speech Signal.
29. Харченко, О., Яцишин, В., & Боднарчук, І. (2013). Експертна система проектування архітектури програмного забезпечення. Комп'ютерні Технології Друкарства, (29), 10–26.
30. Боднарчук, І., Харченко, О., Хоміцький, Б., & Шимчук, Г. (2019). Проектування архітектури програмних систем в проектах з гнучкими методами управління. Матеріали XXI Наукової Конференції Тернопільського Національного Технічного Університету Імені Івана Пулюя, 46–48.
31. Goli V. R. REACT NATIVE EVOLUTION, NATIVE MODULES, AND BEST PRACTICES. INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY. 2021. Vol. 12, no. 2. P. 73–85. URL: https://doi.org/10.34218/ijcet_12_02_009 (date of access: 15.06.2025).
32. React Native vs Native Development: Pros and Cons. AWS Consulting Partner | Gen AI | Product Engineering. URL: <https://www.brilworks.com/blog/react-native-vs-native-development/> (date of access: 15.06.2025).
33. Expo Documentation – Expo CLI overview. Expo. URL: <https://docs.expo.dev/workflow/expo-cli/> (дата звернення: 14.06.2025).
34. Держспецзв'язку затвердила Методичні рекомендації щодо реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі. Інтернет журнал Кібербез. URL: <https://cybersec.net.ua/normatyvni-dokumenty/568-derzhspetsvziazku-zatverdyla-metodychni-rekomendatsii->

shchodo-reahuvannia-subiektamy-zabezpechennia-kiberbezpeky-na-rizni-vydy-podii-u-kiberprostori.html (дата звернення: 16.06.2025).

35. Мобільна безпека: Поради та стратегії для захисту даних. MindScope. URL: <https://mindscope.biz.ua/bezpeka-v-mobilnyh-dodatkah-na-shho-zvernuty-uvagu/> (дата звернення: 16.06.2025).

36. Goodman S. P. J. Approaches to inducing mental fatigue: A systematic review and meta-analysis of (neuro)physiologic indices. Behavior Research Methods. 2025. Vol. 57, no. 3. P. 857–875. URL: <https://doi.org/10.3758/s13428-025-02620-7>.

37. Albulescu P. "Give me a break!" A systematic review and meta-analysis on the efficacy of micro-breaks for increasing well-being and performance. PLOS ONE. 2022. Vol. 17, no. 8. URL: <https://doi.org/10.1371/journal.pone.0272460>.

38. Безпека життєдіяльності : Навч. посіб. / Ю. С. Скобло та ін. Харків : ХПІ. 424 с.

39. Алби Березюк О. В., Лемешев М. С. Безпека життєдіяльності : Навч. посіб. Вінниця : ВНТУ. 204 с.

ДОДАТКИ

Код мобільного додатка

```
import 'dart:convert';
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:file_picker/file_picker.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:path_provider/path_provider.dart';
import 'package:open_file/open_file.dart';
import 'package:share_plus/share_plus.dart';
import 'file_viewer_screen.dart';

void main() {
  runApp(const MyApp());
}

class MyFile {
  final String name;
  final String path;

  MyFile({required this.name, required this.path});

  factory MyFile.fromJson(Map<String, dynamic> json) {
    return MyFile(
      name: json['fileName'],
      path: json['filePath'],
    );
  }

  Map<String, dynamic> toJson() => {
    'fileName': name,
    'filePath': path,
  };
}

class Document {
  final String title;
  final MyFile? file1;
  final MyFile? file2;
  final MyFile? iconFile;

  Document({
    required this.title,
    this.file1,
    this.file2,
    this.iconFile,
  });

  factory Document.fromJson(Map<String, dynamic> json) {
    return Document(
      title: json['title'],
```

```

        file1:          json['file1']          !=          null          ?
MyFile.fromJson(json['file1']) : null,
        file2:          json['file2']          !=          null          ?
MyFile.fromJson(json['file2']) : null,
        iconFile:       json['iconFile']       !=          null          ?
MyFile.fromJson(json['iconFile']) : null,
    );
}

Map<String, dynamic> toJson() => {
    'title': title,
    'file1': file1?.toJson(),
    'file2': file2?.toJson(),
    'iconFile': iconFile?.toJson(),
};
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'CADLIB',
            debugShowCheckedModeBanner: false,
            theme: ThemeData.dark().copyWith(
                scaffoldBackgroundColor: const Color(0xFF1E1E1E),
                appBarTheme: const AppBarTheme(
                    backgroundColor: Color(0xFF252526),
                    foregroundColor: Colors.white,
                    elevation: 1,
                ),
                elevatedButtonTheme: ElevatedButtonThemeData(
                    style: ElevatedButton.styleFrom(
                        backgroundColor: const Color(0xFF007ACC),
                        foregroundColor: Colors.white,
                        textStyle:          const          TextStyle(fontWeight:
FontWeight.bold),
                        padding:          const          EdgeInsets.symmetric(horizontal: 16,
vertical: 12),
                    ),
                ),
                inputDecorationTheme: const InputDecorationTheme(
                    labelStyle: TextStyle(color: Colors.white70),
                    enabledBorder: UnderlineInputBorder(
                        borderSide: BorderSide(color: Colors.white38),
                    ),
                    focusedBorder: UnderlineInputBorder(
                        borderSide: BorderSide(color: Color(0xFF007ACC)),
                    ),
                ),
                textTheme: const TextTheme(
                    bodyMedium: TextStyle(color: Colors.white),

```

```

        ),
    ),
    home: const Scaffold(
        body: SafeArea(child: ButtonScreen()),
    ),
);
}
}

class ButtonScreen extends StatefulWidget {
    const ButtonScreen({super.key});

    @override
    State<ButtonScreen> createState() => _ButtonScreenState();
}

class _ButtonScreenState extends State<ButtonScreen> {
    final List<Document> _documents = [];

    @override
    void initState() {
        super.initState();
        _loadDocuments();
    }

    Future<void> _loadDocuments() async {
        final prefs = await SharedPreferences.getInstance();
        final docs = prefs.getStringList('documents') ?? [];
        setState(() {
            _documents.clear();
            _documents.addAll(docs.map((s) =>
Document.fromJson(jsonDecode(s))));
        });
    }

    Future<void> _saveDocuments() async {
        final prefs = await SharedPreferences.getInstance();
        final asJson = _documents.map((d) =>
jsonEncode(d.toJson())).toList();
        await prefs.setStringList('documents', asJson);
    }

    Future<MyFile?> _copyFileToAppDir(PlatformFile pickedFile) async
    {
        if (pickedFile.path == null) return null;

        final appDir = await getApplicationDocumentsDirectory();
        final fileName =
'${DateTime.now().millisecondsSinceEpoch}_${pickedFile.name}';
        final savedFile = File('${appDir.path}/${fileName}');
        try {
            await File(pickedFile.path!).copy(savedFile.path);
            return MyFile(name: pickedFile.name, path: savedFile.path);
        }
    }
}

```

```

    } catch (e) {
        debugPrint('Помилка копіювання файлу: $e');
        return null;
    }
}

Icon _getIconForFile(String fileName) {
    final ext = fileName.toLowerCase();
    if (ext.endsWith('.dwg') || ext.endsWith('.dxf')) {
        return const Icon(Icons.design_services, color:
Colors.white70, size: 20);
    }
    if (ext.endsWith('.pdf')) {
        return const Icon(Icons.picture_as_pdf, color:
Colors.redAccent, size: 20);
    }
    if (ext.endsWith('.jpg') ||
        ext.endsWith('.jpeg') ||
        ext.endsWith('.png') ||
        ext.endsWith('.gif') ||
        ext.endsWith('.bmp') ||
        ext.endsWith('.webp')) {
        return const Icon(Icons.image, color:
Colors.lightGreenAccent, size: 20);
    }
    return const Icon(Icons.insert_drive_file, color:
Colors.white70, size: 20);
}

Future<void> _addDocument() async {
    final titleCtrl = TextEditingController();
    MyFile? file1;
    MyFile? file2;
    MyFile? iconFile;
    bool isIconSelected = false;

    await showDialog(
        context: context,
        builder: (ctx) => StatefulBuilder(
            builder: (ctx, setSt) => AlertDialog(
                backgroundColor: const Color(0xFF252526),
                title: const Text('Створити документ'),
                content: SingleChildScrollView(
                    child: Column(
                        mainAxisAlignment: MainAxisAlignment.min,
                        children: [
                            TextField(
                                controller: titleCtrl,
                                style: const TextStyle(color: Colors.white),
                                decoration: const InputDecoration(labelText:
'Назва'),
                            ),
                            const SizedBox(height: 10),
                        ],
                    ),
                ),
            ),
        ),
    );
}

```

```

        ElevatedButton(
          onPressed: () async {
            final res = await
FilePicker.platform.pickFiles(
              type: FileType.any,
            );
            if (res != null && res.files.isNotEmpty) {
              final copied = await
_copyFileToAppDir(res.files.first);
              if (copied != null) {
                setSt(() => file1 = copied);
              }
            }
          },
          child: const Text('Обрати файл 1'),
        ),
        if (file1 != null)
        Padding(
          padding: const EdgeInsets.only(top: 4),
          child: Row(
            children: [
              _getIconForFile(file1!.name),
              const SizedBox(width: 8),
              Expanded(
                child: Text(
                  'Вибрано: ${file1!.name}',
                  style: const TextStyle(color:
Colors.white70),
                  overflow: TextOverflow.ellipsis,
                ),
              ),
            ],
          ),
        ),
        const SizedBox(height: 10),
        ElevatedButton(
          onPressed: () async {
            final res = await
FilePicker.platform.pickFiles(
              type: FileType.any,
            );
            if (res != null && res.files.isNotEmpty) {
              final copied = await
_copyFileToAppDir(res.files.first);
              if (copied != null) {
                setSt(() => file2 = copied);
              }
            }
          },
          child: const Text('Обрати файл 2'),
        ),
        if (file2 != null)
        Padding(

```

```

padding: const EdgeInsets.only(top: 4),
child: Row(
  children: [
    _getIconForFile(file2!.name),
    const SizedBox(width: 8),
    Expanded(
      child: Text(
        'Вибрано: ${file2!.name}',
        style: const TextStyle(color:
Colors.white70),
        overflow: TextOverflow.ellipsis,
      ),
    ),
  ],
),
const SizedBox(height: 10),
ElevatedButton(
  onPressed: isIconSelected
    ? null
    : () async {
        final res = await
FilePicker.platform.pickFiles(
          type: FileType.image,
        );
        if (res != null && res.files.isNotEmpty) {
          final copied = await
_copyFileToAppDir(res.files.first);
          if (copied != null) {
            setSt(() {
              iconFile = copied;
              isIconSelected = true;
            });
          }
        }
      },
  child: const Text('Обрати файл іконки'),
),
if (iconFile != null)
  Padding(
    padding: const EdgeInsets.only(top: 4),
    child: Row(
      children: [
        _getIconForFile(iconFile!.name),
        const SizedBox(width: 8),
        Expanded(
          child: Text(
            'Вибрано: ${iconFile!.name}',
            style: const TextStyle(color:
Colors.white70),
            overflow: TextOverflow.ellipsis,
          ),
        ),
      ],
    ),
  ),

```

```

        ],
      ),
    ),
  ],
),
actions: [
  TextButton(
    onPressed: () => Navigator.pop(ctx),
    child: const Text('Скасувати'),
  ),
  TextButton(
    onPressed: () {
      final title = titleCtrl.text.trim();
      if (title.isEmpty) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text('Введіть
назву'))),
        );
      }
      return;
    }
    setState(() {
      _documents.add(Document(
        title: title,
        file1: file1,
        file2: file2,
        iconFile: iconFile,
      ));
    });
    _saveDocuments();
    Navigator.pop(ctx);
  },
  child: const Text('Додати'),
),
],
),
),
);
}

Future<void> _deleteDocument() async {
  if (_documents.isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Немає документів для
видалення'))),
    );
    return;
  }
  await showDialog(
    context: context,
    builder: (ctx) => AlertDialog(
      backgroundColor: const Color(0xFF252526),
      title: const Text('Видалити документ'),

```

```

        content: SizedBox(
          width: double.maxFinite,
          child: ListView.builder(
            shrinkWrap: true,
            itemCount: _documents.length,
            itemBuilder: (ctx, i) => ListTile(
              title: Text(_documents[i].title),
              trailing: IconButton(
                icon: const Icon(Icons.delete, color:
Colors.redAccent),
                onPressed: () {
                  setState(() {
                    _documents.removeAt(i);
                    _saveDocuments();
                  });
                  Navigator.pop(ctx);
                },
              ),
            ),
          ),
        ),
      ),
    ),
    actions: [
      TextButton(
        onPressed: () => Navigator.pop(ctx),
        child: const Text('Закрити'),
      ),
    ],
  ),
);
}

```

```

void _openDocument(Document doc) {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (_) => EditDocumentScreen(
        document: doc,
        onSave: (updatedDoc) {
          setState(() {
            final index = _documents.indexOf(doc);
            if (index != -1) {
              _documents[index] = updatedDoc;
              _saveDocuments();
            }
          });
        },
      ),
    ),
  );
}

```

```

Widget _buildDocumentItem(Document doc) {
  return ListTile(

```

```

        leading: doc.iconFile != null
            ? SizedBox(
                width: 40,
                height: 40,
                child: Image.file(
                    File(doc.iconFile!.path),
                    fit: BoxFit.cover,
                    errorBuilder: (context, error, stackTrace) =>
                        _getIconForFile(
                            doc.file1?.name ?? doc.file2?.name ?? 'default'),
                ),
            )
        : _getIconForFile(doc.file1?.name ?? doc.file2?.name ??
            'default'),
        title: Text(doc.title),
        subtitle: Text.rich(
            TextSpan(
                children: [
                    if (doc.file1 != null)
                        WidgetSpan(
                            child: Padding(
                                padding: const EdgeInsets.only(right: 8),
                                child: _getIconForFile(doc.file1!.name),
                            ),
                        ),
                    if (doc.file2 != null)
                        WidgetSpan(
                            child: _getIconForFile(doc.file2!.name),
                        ),
                ],
            ),
        ),
        onTap: () => _openDocument(doc),
    );
}

```

```

@override
Widget build(BuildContext context) {
    return Container(
        padding: const EdgeInsets.all(8),
        color: const Color(0xFF1E1E1E),
        child: Column(
            children: [
                Expanded(
                    child: _documents.isEmpty
                        ? const Center(
                            child: Text(
                                'Немає документів',
                                style: TextStyle(color: Colors.white70),
                            ),
                        ),
                ),
                : ListView.builder(
                    itemCount: _documents.length,

```

```

        itemBuilder: (context, i) =>
        _buildDocumentItem(_documents[i]),
      ),
    ),
    const SizedBox(height: 10),
    Row(
      children: [
        ElevatedButton(
          onPressed: _addDocument,
          child: const Text('Створити документ'),
        ),
        const SizedBox(width: 10),
        ElevatedButton(
          onPressed: _deleteDocument,
          child: const Text('Видалити документ'),
        ),
      ],
    ),
  ],
),
);
}
}

```

```

class EditDocumentScreen extends StatefulWidget {
  final Document document;
  final Function(Document) onSave;

  const EditDocumentScreen({
    super.key,
    required this.document,
    required this.onSave,
  });

  @override
  State<EditDocumentScreen> createState() =>
  _EditDocumentScreenState();
}

```

```

class _EditDocumentScreenState extends State<EditDocumentScreen> {
  late TextEditingController _titleController;
  MyFile? _file1;
  MyFile? _file2;
  MyFile? _iconFile;

  @override
  void initState() {
    super.initState();
    _titleController = TextEditingController(text:
    widget.document.title);
    _file1 = widget.document.file1;
    _file2 = widget.document.file2;
    _iconFile = widget.document.iconFile;
  }
}

```

```

    }

    Future<MyFile?> _copyFileToAppDir(PlatformFile pickedFile) async
    {
        if (pickedFile.path == null) return null;

        final appDir = await getApplicationDocumentsDirectory();
        final fileName = ... =
        '${DateTime.now().millisecondsSinceEpoch}_${pickedFile.name}';
        final savedFile = File('${appDir.path}/${fileName}');
        try {
            await File(pickedFile.path!).copy(savedFile.path);
            return MyFile(name: pickedFile.name, path: savedFile.path);
        } catch (e) {
            debugPrint('Помилка копіювання файлу: $e');
            return null;
        }
    }

    Icon _getIconForFile(String fileName) {
        final ext = fileName.toLowerCase();
        if (ext.endsWith('.dwg') || ext.endsWith('.dxf')) {
            return const Icon(Icons.design_services, color:
Colors.white70, size: 20);
        }
        if (ext.endsWith('.pdf')) {
            return const Icon(Icons.picture_as_pdf, color:
Colors.redAccent, size: 20);
        }
        if (ext.endsWith('.jpg') ||
            ext.endsWith('.jpeg') ||
            ext.endsWith('.png') ||
            ext.endsWith('.gif') ||
            ext.endsWith('.bmp') ||
            ext.endsWith('.webp')) {
            return const Icon(Icons.image, color:
Colors.lightGreenAccent, size: 20);
        }
        return const Icon(Icons.insert_drive_file, color:
Colors.white70, size: 20);
    }

    Future<void> _pickFile1() async {
        final res = await FilePicker.platform.pickFiles(
            type: FileType.any,
        );
        if (res != null && res.files.isNotEmpty) {
            final copied = await _copyFileToAppDir(res.files.first);
            if (copied != null) {
                setState(() => _file1 = copied);
            }
        }
    }
}

```

```

Future<void> _pickFile2() async {
  final res = await FilePicker.platform.pickFiles(
    type: FileType.any,
  );
  if (res != null && res.files.isNotEmpty) {
    final copied = await _copyFileToAppDir(res.files.first);
    if (copied != null) {
      setState(() => _file2 = copied);
    }
  }
}

void _save() {
  final title = _titleController.text.trim();
  if (title.isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Введіть назву')),
    );
    return;
  }
  widget.onSave(Document(
    title: title,
    file1: _file1,
    file2: _file2,
    iconFile: _iconFile,
  ));
  Navigator.pop(context);
}

void _downloadFile(MyFile? file) {
  if (file == null) return;
  Share.shareXFiles([XFile(file.path, name: file.name)]);
}

void _openFileInViewer(MyFile? file) {
  if (file == null) return;
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (_) => FileViewerScreen(file: file),
    ),
  );
}

Widget _buildFileRow(String label, MyFile? file, VoidCallback
onPick,
  {bool isFile1 = false, bool isFile2 = false}) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      ElevatedButton(
        onPressed: onPick,

```

```

        child: Text(label),
      ),
    if (file != null)
      Padding(
        padding: const EdgeInsets.only(top: 4),
        child: Row(
          children: [
            _getIconForFile(file.name),
            const SizedBox(width: 8),
            Expanded(
              child: Text(
                file.name,
                style: const TextStyle(color: Colors.white70),
                overflow: TextOverflow.ellipsis,
              ),
            ),
            IconButton(
              icon: Icon(isFile1 ? Icons.download :
Icons.open_in_new),
              onPressed: isFile1
                ? () => _downloadFile(file)
                : () => _openFileInViewer(file),
            ),
          ],
        ),
      ),
    ],
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Редагувати документ'),
      actions: [
        IconButton(
          icon: const Icon(Icons.save),
          onPressed: _save,
        ),
      ],
    ),
    body: Container(
      color: const Color(0xFF1E1E1E),
      padding: const EdgeInsets.all(16),
      child: ListView(
        children: [
          TextField(
            controller: _titleController,
            style: const TextStyle(color: Colors.white),
            decoration: const InputDecoration(labelText:
'Назва'),
          ),

```

```
        const SizedBox(height: 16),
        _buildFileRow('Файл 1', _file1, _pickFile1, isFile1:
true),
        const SizedBox(height: 16),
        _buildFileRow('Файл 2', _file2, _pickFile2, isFile2:
true),
      ],
    ),
  ),
);
}
}
```