

File Management Application with AI Assistance

Aiman Hazriq Shazwan¹, Mohd Zanes Sahid^{1*}

*¹ Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: zanes@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2026.07.01.049>

Article Info

Received: 15 June 2025

Accepted: 22 June 2026

Available online: 30 June 2026

Keywords

File Manager System, Mobile
Application, AI-assisted

Abstract

With the increasing reliance on mobile devices for everyday tasks, file management has become a critical challenge, particularly for non-technical users who struggle with complex systems. This project aims to develop an AI-enhanced file management application tailored for Tiang Seri Sdn. Bhd., which automates tasks such as file naming, and categorization, and storage optimization. The application leverages AI to offer intelligent suggestions and simplify file organization, making it easier for users to manage their documents. The system was developed using Flutter, focusing on an intuitive interface and efficient workflow. Preliminary testing indicates that the app significantly improves file organization, reduces cognitive load, and enhances overall productivity. Future work will focus on further refining AI capabilities and expanding the app's features for broader business applications.

1. Introduction

The increasing integration of mobile devices into daily life has led to a revolution in how people interact with technology and manage data. Mobile applications have become essential tools, transforming the way users perform tasks ranging from communication to productivity and entertainment. As of 2023, it was reported that over 6.8 billion people globally use smartphones, making mobile applications an integral part of modern life [1]. Among these applications, file management apps have remained critical, allowing users to organize, retrieve, and manage their files efficiently. These apps enable users to work with a wide variety of file types, whether for business, educational, or personal use. However, despite their importance, many existing file management applications are still plagued by issues related to user experience (UX), navigation complexity, and poor adaptability to user behavior.

Traditional file management systems often require users to manually organize files, which can become time-consuming, particularly when dealing with a large number of documents. For instance, users often struggle with naming conventions, categorizing files, or retrieving important files quickly from a cluttered system. This challenge becomes even more prominent among non-technical users who may find it difficult to navigate complex systems. Moreover, many current applications fail to leverage modern technologies such as Artificial Intelligence (AI) to streamline these tasks, offering no intelligent suggestions or automation features. According to Nielsen and Molich [2], overly complex interfaces, especially those with unnecessary features, tend to frustrate users, making these systems inefficient and counterproductive.

This study seeks to address these challenges by developing an AI-enhanced mobile file management application that automates tasks such as file naming, categorization, and storage optimization. The application aims to improve the efficiency and usability of file management by integrating AI algorithms to suggest intelligent actions like archiving, deleting, or categorizing files based on usage patterns. By automating these processes, the

application is designed to significantly reduce the manual effort required to organize and retrieve files, thereby enhancing overall productivity. Specifically, the research questions guiding this study are: How can AI be utilized to automate and streamline file management tasks? What are the key features of an AI-enhanced file manager that can improve usability and efficiency for non-technical users?

The primary objective of this study is to develop an Android-based AI-powered file management system using the Flutter framework, ensuring that the application is intuitive and easy to use for individuals with minimal technical knowledge. This application will not only focus on automating file organization but will also offer functionalities such as AI-driven suggestions, cloud synchronization, and a user-friendly interface. By focusing on the needs of users like Shazwan Shamsuddin, Chief Operator at Tiang Seri Sdn. Bhd., the app aims to streamline file organization tasks that are currently burdensome due to lack of digital literacy and poor usability. The expected outcome is a practical, user-friendly solution that will reduce the cognitive load on users, improve productivity, and minimize the time spent searching for and organizing files. The project will also examine how AI can be effectively integrated into file management applications to benefit users with varying levels of technical proficiency.

This paper is structured as follows: Section 2 provides a detailed review of related work, covering existing file management systems and AI applications in this domain. Section 3 outlines the methodology used to develop the AI-enhanced file management system, including system architecture and design considerations. Lastly, Section 4 presents the results of testing the application.

2. Related Work

This section describes Mobile File Management Systems, AI in File Management Systems, and Study of Existing Systems.

2.1 Mobile File Management Systems

The rise of cross-platform development tools like Flutter has simplified the creation of efficient, user-friendly mobile applications that work seamlessly across different devices. Mobile file management apps often face challenges such as limited screen size, slower processing speeds, and difficulties managing large volumes of data on portable devices [3]. Flutter, an open-source framework for building natively compiled applications, has become popular for its speed and efficiency, enabling the creation of smooth user interfaces and the integration of AI models [4]. This makes Flutter an ideal choice for developing a file management system that is functional and accessible, especially for users like En. Wan, who may not have prior experience with complex digital systems.

2.2 AI in File Management Systems

Artificial Intelligence (AI) has significantly enhanced file management systems by automating tasks, improving organization, and speeding up file retrieval. AI applications, such as file categorization, naming conventions, and data labeling, have proven effective in reducing manual effort and errors [5]. Additionally, machine learning algorithms improve search functions by understanding user preferences and predicting file-related needs. Natural Language Processing (NLP) has enabled intelligent search systems that interpret conversational queries, enhancing usability and productivity [6]. By integrating AI, businesses can provide a more personalized and efficient file management experience.

2.3 Study of Existing Systems

While some advanced file management systems offer automation and improved organization, they often require significant user expertise. With many of these systems, despite incorporating AI and machine learning, they still have steep learning curves. For example, Google Drive and Dropbox offer intelligent search features but do not automatically organize files based on user behavior or suggest optimal naming conventions. This section compares existing applications, such as Files by Google [7], FX File Explorer [8], and Solid Explorer [9], with the proposed AI-enhanced application.

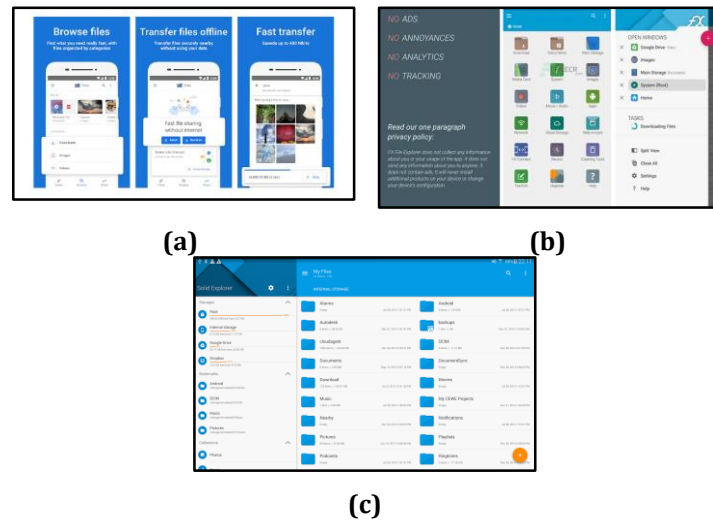


Fig. 1 Existing Systems (a) Files by Google; (b) FX File Explorer; (c) Solid Explorer

Table 1 Comparison between existing apps and the proposed application

Categories	Features	Files by Google	FX File Explorer	Solid Explorer	Proposed Application
User Interface (UI)	User Friendly	✓	✗	✓	✓
	Customizable Themes	✓	✓	✓	✓
	Complex	✗	✓	✓	✗
User Experience (UX)	Smart Suggestions	✗	✗	✗	✓
	Seamless File Navigation	✓	✗	✓	✓
Additional Functions	Automated File Naming and Organising	✗	✗	✗	✓
	AI-assisted Categorization	✗	✗	✗	✓
	Custom Folder Structures	✗	✓	✓	✓
Target Audiences	Casual Users	✓	✗	✗	✓
	Power Users	✗	✓	✓	✗
Price	Free	✓	✓	✗	✓
	Paid	✗	✓	✓	✗

3. Methodology/Framework

This chapter aims to clarify the adopted methodology, emphasizing the selected software development model. The chosen model is Agile Model.

3.1 Agile Model

The development of the AI-enhanced file management application will follow the Agile methodology, which emphasizes flexibility, iterative progress, and close collaboration with stakeholders. Agile's adaptability to changing requirements makes it ideal for this project, allowing for incremental improvements through multiple

1–2 week sprints. Each sprint will focus on tasks like UI design, AI algorithm integration, and mobile app development, followed by feedback from stakeholders such as En. Wan. This iterative approach ensures continuous refinement, early issue identification, and alignment with user needs, leading to a high-quality product [10].

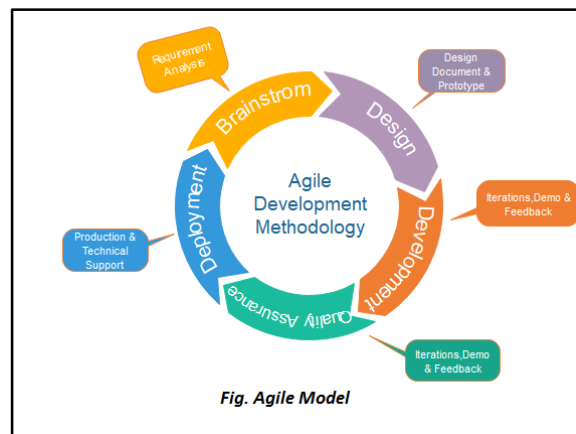


Fig. 1 *Agile Mode*

The testing phase is critical to ensuring the AI-enhanced file management application is functional, secure, and meets user requirements. This phase will involve both manual and automated testing to identify and resolve bugs, performance issues, and security vulnerabilities. Unit testing will be performed to test individual components, such as AI algorithms and mobile app features, to ensure they function correctly in isolation. Integration testing will assess the interaction between different system components, including the AI, mobile front-end, and back-end, ensuring seamless communication. User Acceptance Testing (UAT) will involve testing with En. Wan and other users to verify that the application meets their needs and is intuitive to use. Performance testing will evaluate the app's response times, load handling, and resource utilization, ensuring optimal efficiency. Finally, security testing will include vulnerability assessments to ensure the system is secure, particularly given its handling of sensitive documents.

4. Result and Discussion

4.1 System Requirements

System requirements describes the detailed description of system functionalities that a system must possess in order to meet the user requirements. Table 2 shows the list of 5 functional requirements. Table 3 shows the non-functional requirement, Table 4 shows the user requirement.

Table 2 *Functional Requirement*

Function	Requirements/Functionalities
File Management	Allows the user to perform basic file operations.
AI File Assistance	Automatic file categorization, naming convention, and smart suggestions.
Search, Filter, and Indexing	Allows the user to search files based on certain criterias or keywords.
Notifications and Alerts	Notifies and alerts users.
Security	Request only necessary Android permissions for accessing local storage.

Table 3 *Non-functional Requirement*

Function	Requirements/Functionalities
----------	------------------------------

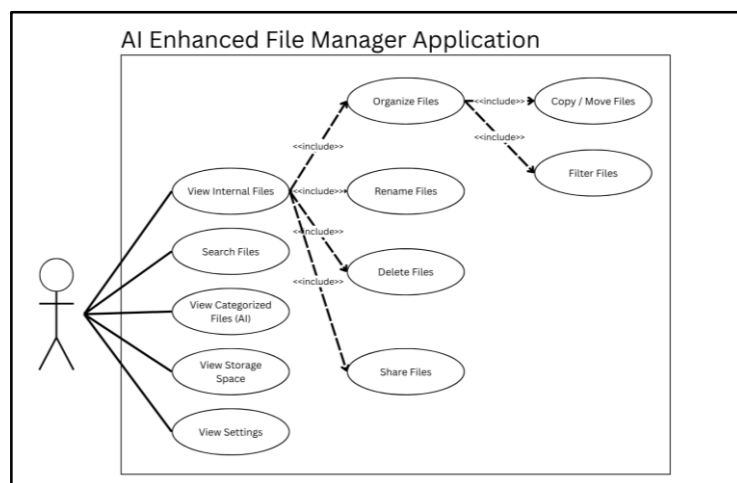
Performance	File operations should be completed in less than 5 seconds for typical file sizes.
Scalability	The app should be able to handle large volumes of files without performance degradation.
Reliability	File operations must be atomic, meaning they should either fully complete or leave the file system in a consistent state.
Usability	The app should be easy to use, with a smooth learning curve, especially for non-technical users.
Compatibility	The app should be compatible with a wide range of Android devices, from budget to high-end models.

Table 4 User Requirement

NO	Requirements
1	The application should provide an intuitive and easy-to-use interface suitable for non-technical users.
2	Users should be able to create customized file naming conventions, and the AI should automatically suggest or enforce these conventions for new files.
3	The application must provide a powerful search feature that allows the user to find files quickly based on various criteria (e.g., name, file type, content).
4	Users should be able to send files directly from within the app to other users, or through external platforms (e.g., email, cloud services).
5	Users should be notified when storage is nearing capacity and prompted to take appropriate actions.
6	The app should allow users to personalize settings.

4.2 System Analysis

Figure 2 illustrates the use-case diagram for the proposed application.

**Fig. 2** Use Case Diagram of Proposed Application

In Figure 2, the use case diagram includes 5 major use cases. This diagram describes the interactions within the system and how the various components work together to provide the full functionality of the proposed system.

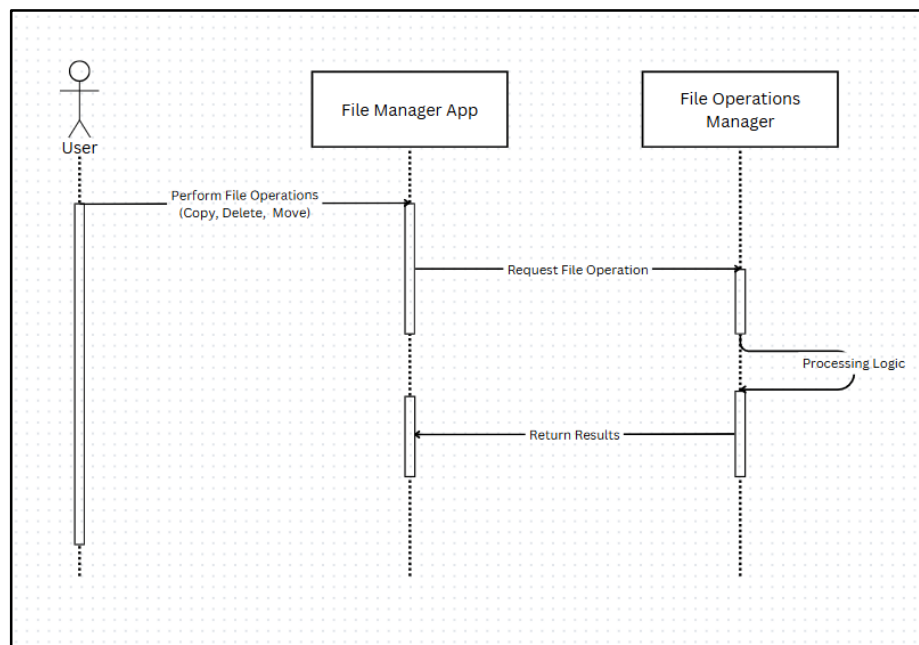


Fig. 3 Sequence Diagram for File Operations

In Figure 3, the sequence diagram for file operations is presented. The process initiates with the user performing file operation tasks such as copying, moving or deleting files. Next, the file manager app will send the user's request to the file operations manager. After processing the needed logic for the file operation, the file operations manager will return the results to the file manager app and display it to the user.

In Appendix A, the sequence diagram for file organization is presented. The process initiates with the user requesting to organize files. Next, the file manager app will send the user's request to categorize files to the file operations manager. The file operations manager will process the request via AI Engine as it performs AI Categorization. The AI helps the user by labeling similar image file patterns into their distinctive categories. After processing, the categorized files are returned along with a smart suggestion if the user wishes to accept or reject the tip after reviewing the suggestion.

In Appendix B, the sequence diagram for file searching is presented. It begins with the user entering a search query into the file manager app. Next, the file manager app parses the search query for processing to the search engine. Once the search query has been processed, the search engine will return the results.

Appendix C depicts the user activity diagram. The user launches the application allowing the user to perform basic file management tasks, operations, searching files, viewing AI-categorized files and manipulate settings.

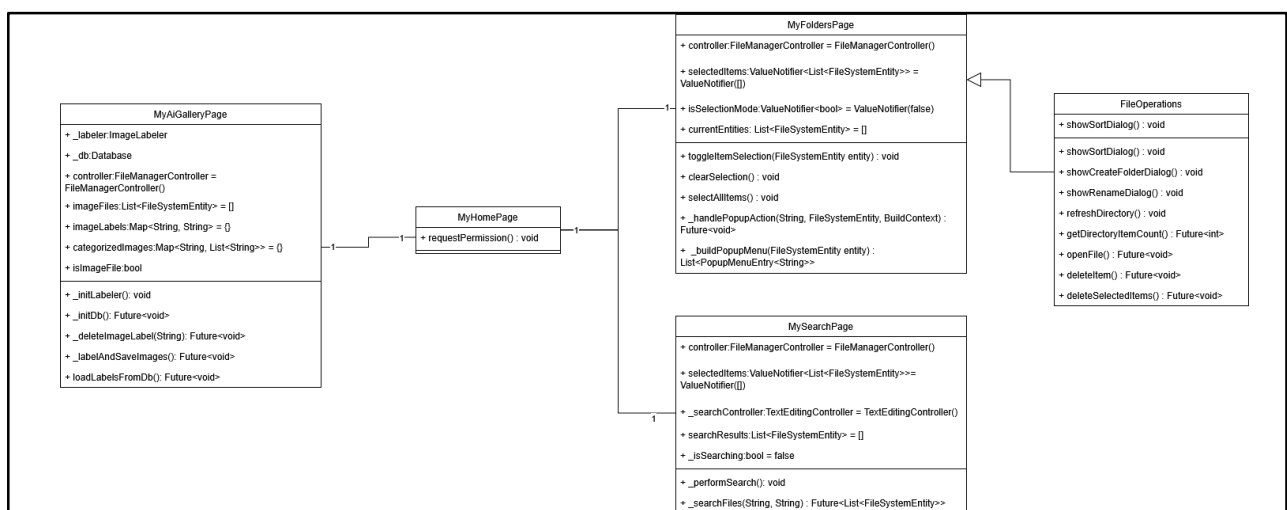


Fig. 4 Class Diagram of Proposed Application

The class diagram is a static diagram that depicts the main classes, variables, methods and the relationships between them in a system. The diagram consists of 5 classes: MyHomePage, MyFoldersPage, MySearchPage,, FilesOperations, and MyAiGalleryPage.

4.3 System Design

The system design depicts the general system architecture that is used to demonstrate the system's overall function. It helps in comprehending the application's design and how it is constructed. Figure 5 depicts the proposed application's constructed design. IT is made up of the following components: the User, User Interface (Android-Based), Core Application Logic, Android File System, AI Engine and External Integration.

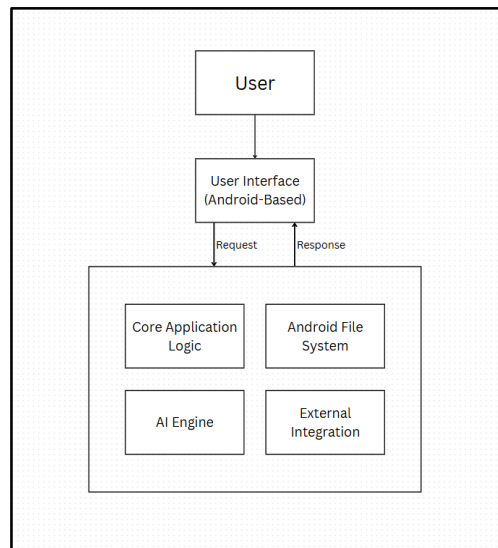


Fig. 5 System Architecture Of Proposed Application

4.4 Implementation

The implementation showcases the layout and interface of the system across pages.

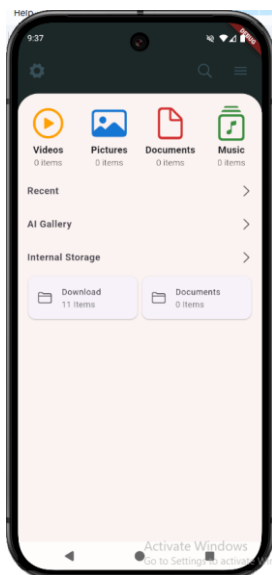


Fig. 6 Home Page

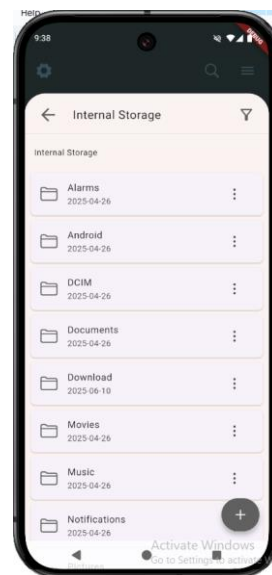


Fig. 7 Internal Storage Page

When the user first launches the app, the home page is presented as shown in Figure 6. The home page contains, from top to bottom, the settings, search function, disk space info, type of files by category, recent files page, AI gallery page, internal storage page and instant access folders. Figure 7 depicts how the internal storage page is displayed.

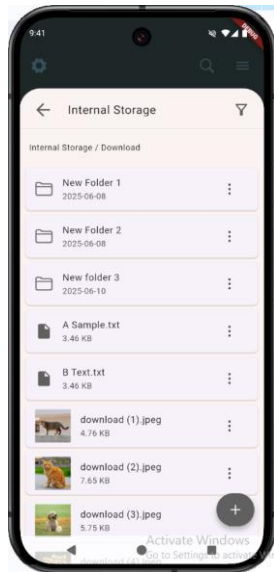


Fig. 7 Navigating into a Directory

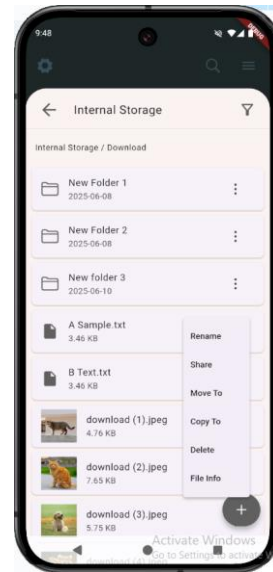


Fig. 8 File Operations

Figure 7 shows navigation into a directory, in this case, the Download folder. Figure 8 depicts the available file operation functions: 'Rename', 'Share', 'Move To', 'Copy To', 'Delete' and 'File Info'.

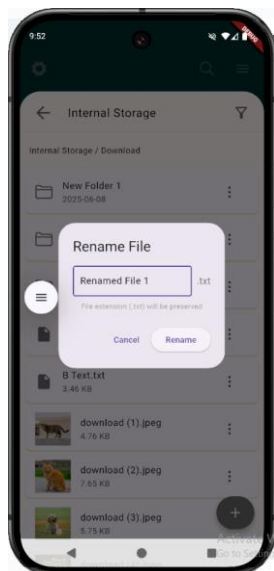


Fig. 9 Rename Function

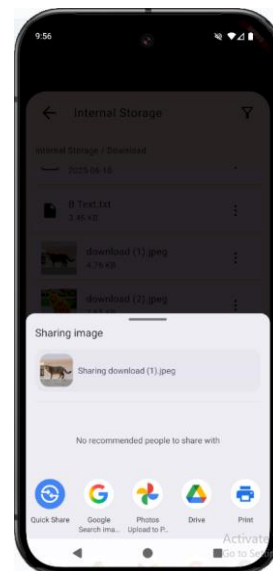


Fig. 10 Share / Export Function

Figure 9 depicts what would be displayed if the rename function is executed. Figure 10 is shown how the share function is depicted with multiple sharing choices: 'Share to Google Drive', 'Print', etc..

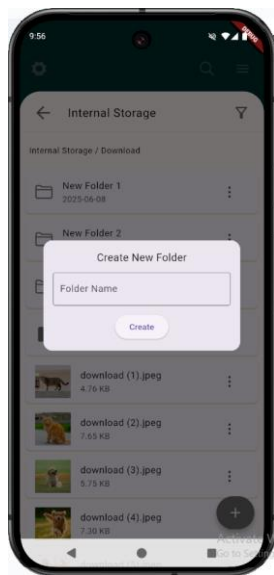


Fig. 11 *Create New Folder Function*

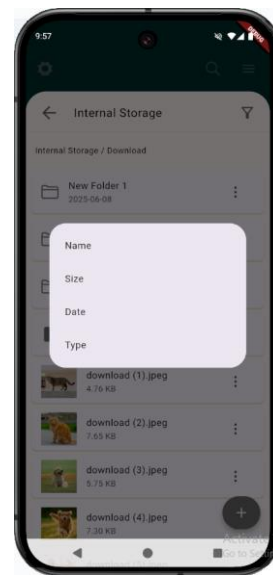


Fig. 12 *Filter Function*

Figure 11 depicts what would be displayed if the user wants to create a new folder inside a directory. Figure 12 shows a list of sorting options by name, size, date and type if the user uses the filter function.

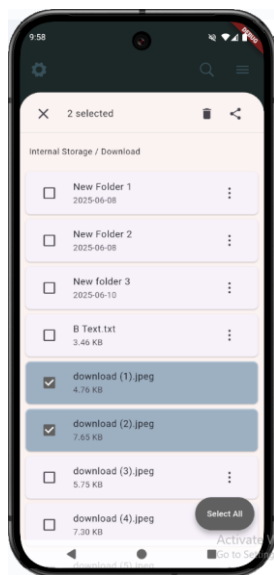


Fig. 13 *Multi-Selection*



Fig. 14 *Image Preview*

Figure 13 is how the multi-selection tool is depicted enabling for batch operations. Figure 14 depicts an image preview tool for image files.

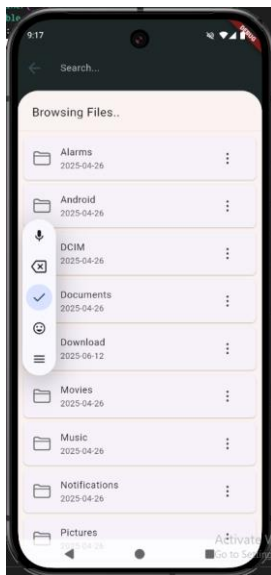


Fig. 15 Search Function

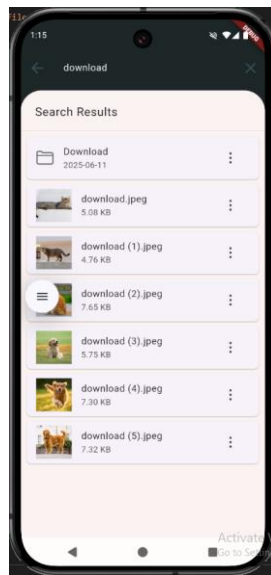


Fig. 16 Search Results

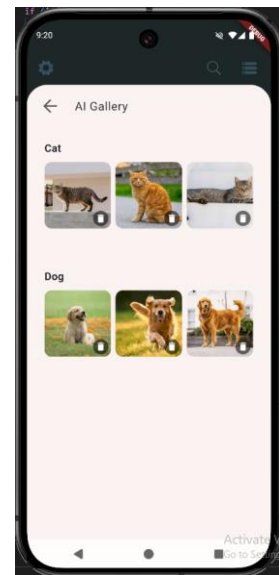


Fig. 17 AI Gallery

Figure 15 depicts the user using the search function. Search results are depicted as seen in Figure 16. In Figure 17, the AI Gallery is depicted recognizing and labeling image files into categories while saving all the data into a local embed database.

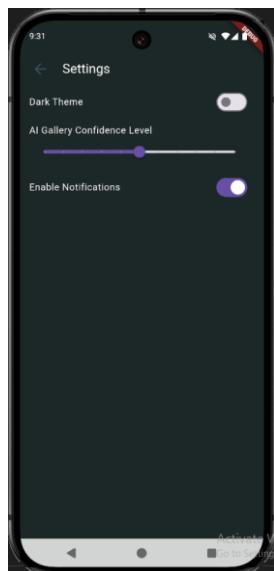


Fig. 18 Settings Page

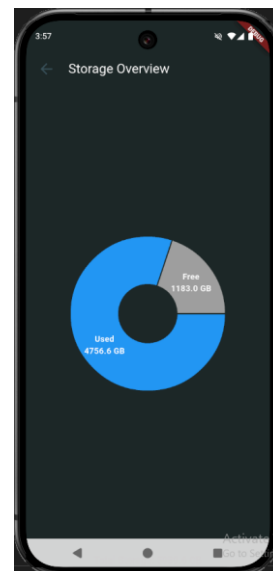


Fig. 19 Storage Visualization

Figure 18 depicts the settings page and its various options. Figure 19 depicts the storage space used and left.

4.5 Testing

Testing is carried out to ensure functionality expectations are met. There are two types of tests, functional testing and user acceptance testing.

4.6 Functional Testing

Table 5 *Functional Testing*

Test Case ID	Test Case	Expected Outcome	Actual Outcome	Pass / Fail
TC-01	Users perform basic file management operations such as creating folders, renaming, deleting, copying, and moving file items.	File management operations are performed successfully.	As expected	Pass
TC-02	Users are able to filter files based on name, size, date and type.	The filtering function is performed successfully.	As expected	Pass
TC-03	Users are able to share files to external platforms such as email, Google Drive, and Whatsapp.	The sharing function is performed successfully.	As expected	Pass
TC-04	Users are able to utilize the search function and find specific file items.	The search function locates the correct specified file item.	As expected	Pass
TC-05	Users are able to see categorized image files by the AI assistance and can rename them to suit their preferences.	Image files are identified or labeled with 95% consistency and categorized.	As expected	Pass
TC-06	Users are introduced to smart suggestions by the AI assistance.	Smart suggestions operating throughout app usage.	Not achieved	Fail
TC-07	Users are notified by the system about their internal storage capacity.	The system pushed notifications regarding storage capacity by the percentage.	As expected	Pass
TC-08	Users are requested by the system for necessary Android permissions.	The system requests and is granted permission successfully.	As expected	Pass
TC-09	Users are able to personalize the system's settings.	Personalizing themes.	As expected	Pass

4.7 User Acceptance Testing

In Appendix D, the user acceptance testing form is presented.

Conclusion

In conclusion, the development of an AI-enhanced file management application aims to address the common challenges faced by En. Wan in organizing, retrieving, and managing files efficiently. By leveraging the power of Artificial Intelligence, the proposed system offers intelligent automation for file categorization, with custom labeling all while maintaining a user-friendly interface suitable for both technical and non-technical users.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

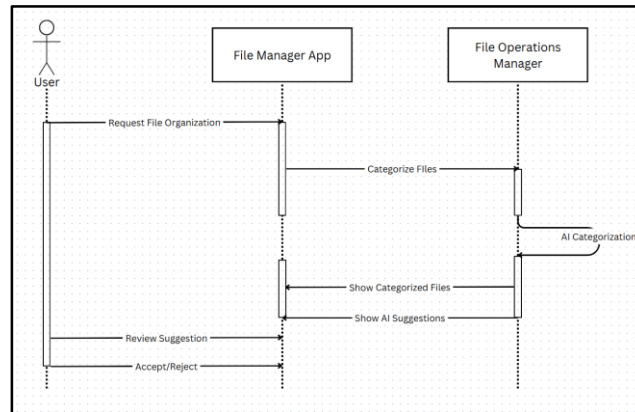
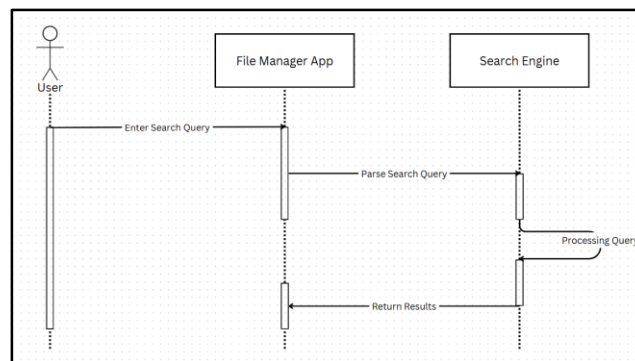
Author Contribution

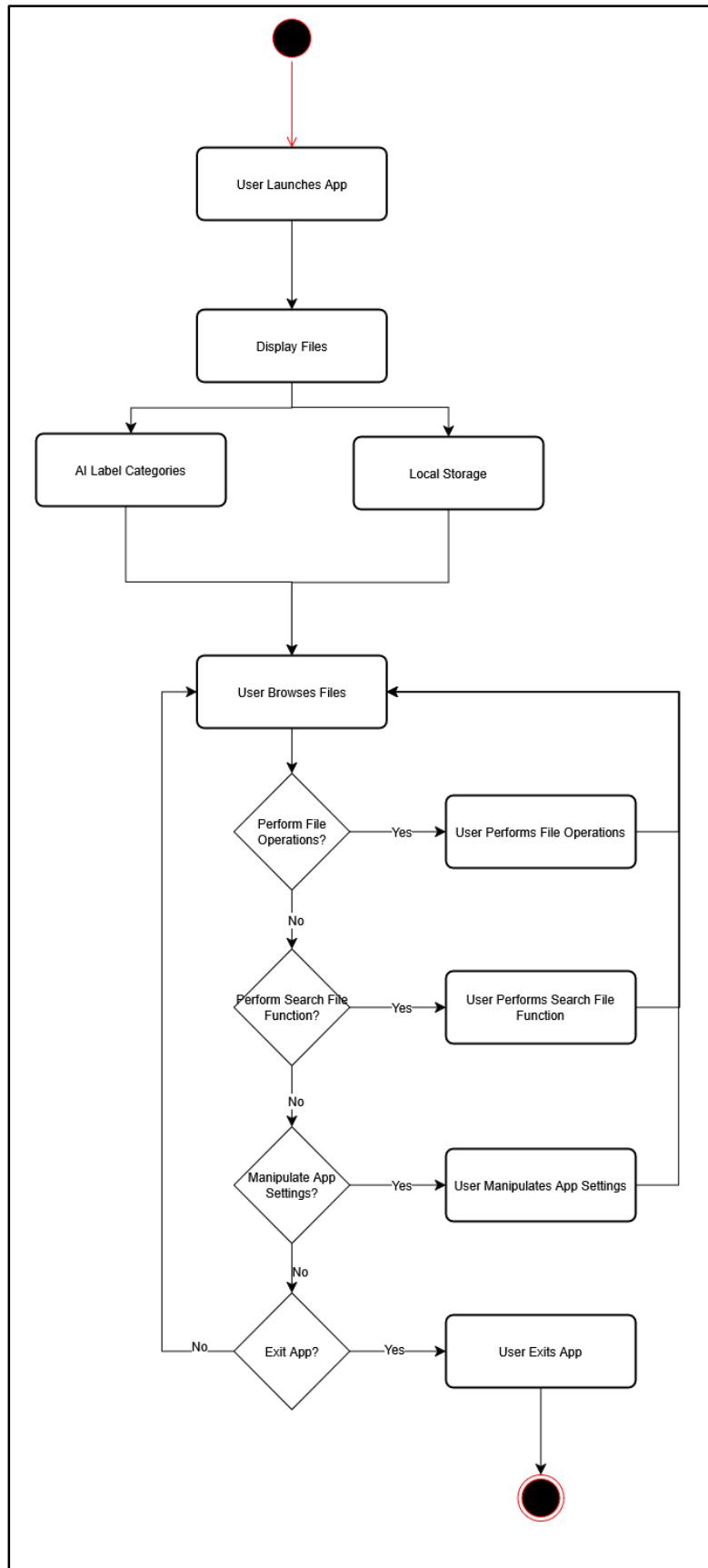
This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

*The authors confirm contribution to the paper as follows: **study conception and design:** Aiman Hazriq Bin Shazwan, Mohd Zanes Bin Zahid; **data collection:** Aiman Hazriq Bin Shazwan; **analysis and interpretation of results:** Aiman Hazriq Bin Shazwan, Mohd Zanes Bin Zahid; **draft manuscript preparation:** Aiman Hazriq Bin Shazwan, Mohd Zanes Bin Zahid. All authors reviewed the results and approved the final version of the manuscript.*

References

- [1] Statista, "Number of smartphone users worldwide from 2016 to 2023," Statista, 2023. [Online]. Available: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.
- [2] J. Nielsen and H. Molich, "Heuristic evaluation of user interfaces," ACM SIGCHI Bulletin, vol. 23, no. 1, pp. 249-251, 1990.
- [3] Kaur, A., et al., "Challenges in Mobile File Management Applications," Journal of Mobile Technology, vol. 15, no. 2, pp. 45-56, 2019.
- [4] Choudhury, S., "Overview of Flutter for Cross-Platform Development," International Journal of Mobile Computing, vol. 22, no. 4, pp. 123-130, 2020.
- [5] Z. Zhang et al., "AI-based file categorization and data labeling in file management systems," Journal of Artificial Intelligence Applications, vol. 18, no. 4, pp. 234-245, 2020.
- [6] L. Hu et al., "Intelligent file search using Natural Language Processing," International Journal of Computer Science and Applications, vol. 26, no. 3, pp. 112-124, 2018.
- [7] Google, "Files by Google," Google Play, 2024. [Online]. Available: <https://play.google.com/store/apps/details?id=com.google.android.apps.nbu.files>.
- [8] C. Deleuze, "FX File Explorer," FX File Explorer, 2023. [Online]. Available: <https://play.google.com/store/apps/details?id=com.foxfi.android>.
- [9] T. Jokinen, "Solid Explorer," Solid Explorer, 2023. [Online]. Available: <https://play.google.com/store/apps/details?id=pl.solidexplorer2>.
- [10] M. Beck et al., "Agile methodology in software development," Software Engineering Journal, vol. 28, no. 5, pp. 55-62, 2021.

Appendix A: Sequence Diagram for File Organization**Appendix B: Sequence Diagram for File Searching**

Appendix C: User Activity Diagram

Appendix D: User Acceptance Testing Form

User Acceptance Testing File Management Application With AI Assistance

Name : Shazwan Shamsuddin
Date : 11 / 6 / 2025
Company : Tiang Seri Sdn. Bhd.

No.	Module / Acceptance Criteria	Test Result		Remarks (if any)
		Pass	Fail	
1.	First Launch			
	App opens without crashes on first launch	/		
	Necessary permissions are requested and granted correctly	/		
2.	Functionality Testing			
	Able to view inside internal storage	/		
	Folders and files are displayed correctly	/		
	Copy, move, delete, and rename files/folders	/		
	Multi-selection and batch operations	/		
	Search function	/		
	Filter function	/		
	Preview images	/		
	OnTap 'Open With' function	/		
	Share function (Google Drive, Whatsapp, etc.)	/		
	Image file categorizing and labelling by AI assistance (AI Gallery)	/		
	Custom labelling	/		
	Smart suggestion by AI assistance		/	
	Notifications	/		
	Personalize system settings	/		
3.	Performance & Usability			
	Smooth scrolling	/		
	App retains settings and saved data after restart	/		

I am**Shazwan Shamsuddin**..... hereby declare that the information provided is true and correct.

Agree by :

TIANG SERI
n e s i n
(+60454220-0)
57 ground floor jalan pusat
komersial saujana 3, pusat
komersial saujana seremban 2
highrise 70300 seremban
negeri sembilan

Name : Shazwan Shamsuddin
Position : Chief Operator
Date : 11 / 6 / 2025