**Industrial Internship Report on**

**”File Manager”**

**Prepared by**

**[Divya Rai]**

This report provides details of the Industrial Internship provided by Upskill campus. This internship focused on the development of a File Manager project using Python. The project, including the report, was completed in six weeks.

The File Manager project involved creating a Python-based application to manage and organize files on a computer system efficiently. This internship offered valuable experience in addressing real-world problems and designing practical solutions. Overall, it was a highly enriching experience.

**TABLE OF CONTENTS**

1 Preface…………………………………………………………………………………………………………………………………3

2 Introduction…………………………………………………………………………………………………………………………4

2.1 About UniConverge Technologies Pvt Ltd………………………………………………………………………..4

2.2 About Upskill campus……………………………………………………………………………………………………..8

2.3 Objective…………………………………………………………………………………………………………………………9

2.4 Purpose………………………………………………………………………………………………………………………….9

2.5 Scope…………………………………………………………………………………………………………………………….10

3 Problem Statement………………………………………………………………………………………………………………11

4 Existing and proposed solution…………………………………………………………………………………………….12

5 Proposed Design………………………………………………………………………………………………………………….14

5.1 High level diagram………………………………………………………………………………………………………….14

5.2 Low level diagram…………………………………………………………………………………………………………..15

5.3 Interfaces……………………………………………………………………………………………………………………….17

6 Performance Test………………………………………………………………………………………………………………18

6.1 Test plan/Test cases……………………………………………………………………………………………………..18

6.2 Test procedure…………………………………………………………………………………………………………….20

6.3 Performance outcome………………………………….…………………………………………………………….21

7 My learnings………………………………………………………….…….……………………………………………………..22

8 Future scope……………………………………………………………………………………………………………………….23

9 Conclusion…………………………………………………………………………………………………………………………..24

# Preface

This report summarizes the work done during the six-week internship. The internship provided a significant opportunity for career development by addressing a relevant problem through the File Manager project. The internship was well-structured by Upskill campus and offered exposure to industry-standard practices and tools.

The project aimed to develop a Python-based file management system that allows users to efficiently organize, search, and manage their files. The program was well-planned and executed with the guidance and support of mentors and industry professionals.

# Introduction

## About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging variousCutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end etc.



1. UCT IoT Platform

**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

* It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
* It supports both cloud and on-premises deployments.

It has features to  
• Build Your own dashboard  
• Analytics and Reporting  
• Alert and Notification  
• Integration with third party application (Power BI, SAP, ERP)  
• Rule Engine

1.  **based Solution**

UCT is one of the early adopters of LoRAWAN technology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

1. **Predictive Maintenance**

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



## About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self -paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year



## Objectives of this Internship program

The objective of this internship was to gain practical experience in the industry, solve real-world problems, improve job prospects, and enhance understanding of our field and its applications. Personal growth, such as better communication and problem-solving skills, was also a key focus.

## Purpose

The purpose of the file management system is to provide an efficient and organized

solution for managing files within an organization. The system aims to streamline the file

management process, ensure data security, and improve productivity and collaboration among

users. The main goals of the file management system include

The purpose of the file management system is to provide an efficient and organized solution for managing files within an organization. The system aims to streamline the file management process, ensure data security, and improve productivity and collaboration among users. The main goals of the file management system include:

Centralized file storage: The system provides a centralized location for storing and organizing files, eliminating the need for scattered file storage and reducing the risk of data loss or misplacement.

Efficient file retrieval: The system allows users to easily retrieve files based on their user roles, file types, and permissions, making it quick and convenient to access relevant documents.

Secure file management: The system ensures data security through user authentication, role-based access control, and permissions management, ensuring that only authorized users can access and modify files.

User-friendly interface: The system provides a user-friendly interface for both admins and employees/clients, making it easy to upload, view, and manage files without extensive technical knowledge.

Reporting and monitoring: The system provides reporting and monitoring features for admins to track system activity, generate reports on admin and user actions, and monitor file usage and access patterns.

Collaboration and productivity: The system promotes collaboration among users by providing a central platform for sharing and managing files, allowing for version control, document tracking, and document sharing among authorized users.

Scalability and flexibility: The system is designed to be scalable and flexible, accommodating the needs of small to large organizations and allowing for future enhancements or customizations based on specific requirements.

**2.5 Scope**

The project scope for a file management system can vary depending on the specific goals and objectives of the project. However, here are some potential components that could be included in the project scope:

1. User research: Conducting user research to identify the requirements and pain points of users in existing file management systems, as well as their needs for a new system.

2. Functional requirements: Defining the features and functionalities of the file management system, such as file storage, organization, retrieval, collaboration, and security.

3. Technical requirements: Determining the technical specifications of the file management system, such as the hardware and software requirements, database design, and system architecture.

4. System design: Developing a system design that includes the user interface, system architecture, and workflow processes.

5. System development: Building and testing the file management system according to the functional and technical requirements.

6. User testing and feedback: Conducting user testing and feedback to evaluate the usability and user experience of the file management system.

7. Performance evaluation: Assessing the performance of the file management system, including its speed, reliability, and efficiency in handling various file types and sizes.

8. Deployment and training: Deploying the file management system and providing training and support to users to ensure effective adoption and usage.

# Problem Statement

With the exponential growth of digital data, users increasingly face challenges in managing and organizing their files effectively. Traditional file management systems, while functional, often lack the flexibility, customization, and efficiency required to handle large volumes of diverse file types. Users need an intuitive solution that simplifies file organization, enhances search capabilities, and streamlines overall file management processes.

The objective of this project is to develop a Python-based File Manager that addresses these challenges by providing a user-friendly interface and robust functionalities. This system will offer efficient file organization, powerful search capabilities, and customizable features to meet individual user needs. By leveraging Python's versatility and libraries, the File Manager aims to deliver a lightweight, fast, and reliable solution that improves user productivity and file management efficiency.

Key features to be implemented include:

* An intuitive graphical user interface (GUI) for easy navigation and operation.
* Efficient algorithms for file sorting, categorization, and organization.
* Advanced search functionalities to quickly locate files based on various criteria.
* Customizable options to tailor the File Manager to specific user preferences and workflows.
* Integration with common file operations such as copy, move, delete, and rename.
* Support for handling a wide range of file types and formats.

This project seeks to fill the gap left by existing file management systems by providing a comprehensive solution that enhances user experience and meets the demands of modern digital data management.

# Existing and Proposed solution

**Existing solution-**

Current file management systems, such as those built into operating systems (e.g., Windows File Explorer, macOS Finder, and Linux file managers like Nautilus), and third-party applications (e.g., Total Commander, Directory Opus, and Free Commander), provide fundamental functionalities for file browsing, organization, and management. However, they present several limitations:

1. Limited Customization: Many existing file managers offer a one-size-fits-all approach, lacking the flexibility to be tailored to specific user preferences and workflows. Users often cannot modify the interface or functionality to suit their unique needs, leading to inefficiency.
2. Inefficient Search Capabilities: While most file managers offer basic search functionalities, they can be slow and inadequate when dealing with large volumes of files or when searching by complex criteria such as file metadata, content, or custom tags.
3. Complex User Interface: Some file managers, particularly those with more advanced features, tend to have complex interfaces that can be overwhelming for users who only need basic functionalities. This can lead to a steep learning curve and reduced productivity.
4. Lack of Integration: Many file managers do not seamlessly integrate with other applications or cloud storage solutions, limiting the ability to manage files across different platforms and environments.

**Proposed solution-**

To address the limitations of existing file management systems, the proposed solution is a Python-based File Manager that offers enhanced flexibility, efficiency, and user experience. Key features and improvements include:

1. Customizable Interface and Functionality: The proposed file manager will allow users to customize the interface, enabling them to tailor the layout, themes, and functionality according to their personal preferences and workflows. This flexibility ensures that users can optimize the file manager for their specific needs.
2. Advanced Search Capabilities: The file manager will incorporate powerful search functionalities, enabling users to quickly locate files based not only on filenames but also on metadata, content, and custom tags. The search algorithms will be optimized for speed and accuracy, even when dealing with large datasets.
3. User-Friendly Interface: The proposed solution will feature an intuitive and streamlined user interface that balances simplicity with functionality. This design approach will cater to both novice users and those who require more advanced file management capabilities.
4. Integration with External Services: To enhance productivity, the file manager will support integration with popular cloud storage services (e.g., Google Drive, Dropbox), allowing users to manage local and cloud files within the same interface. Additionally, the file manager will provide APIs for integration with other applications, enabling a more seamless workflow.
5. Efficiency and Performance: The file manager will be designed to be lightweight and fast, ensuring quick startup times and responsiveness even when handling large directories or performing resource-intensive operations.
6. Support for a Wide Range of File Types: The proposed file manager will support a broad spectrum of file formats and provide features for managing them effectively, such as batch renaming, automated file sorting, and detailed file information displays.

By addressing the shortcomings of existing solutions, the proposed Python-based File Manager aims to deliver a versatile, efficient, and user-friendly tool that meets the needs of modern users who require more from their file management systems.

## Code submission (Github link)

## Report submission (Github link) :

# Proposed Design/ Model

## High Level Diagram

+-------------------+ +-------------------+

| | | |

| User Interface |<----->| Application API |

| | | |

+-------------------+ +-------------------+

| |

v v

+-------------------+ +-------------------+

| | | |

| Authentication |<----->| File Management |

| Service | | Service |

+-------------------+ +-------------------+

| |

v v

+-------------------+ +-------------------+

| | | |

| Database Server |<----->| Storage Service |

| | | |

+-------------------+ +-------------------+

Figure 1: System Architecture Diagram

## Low Level Diagram

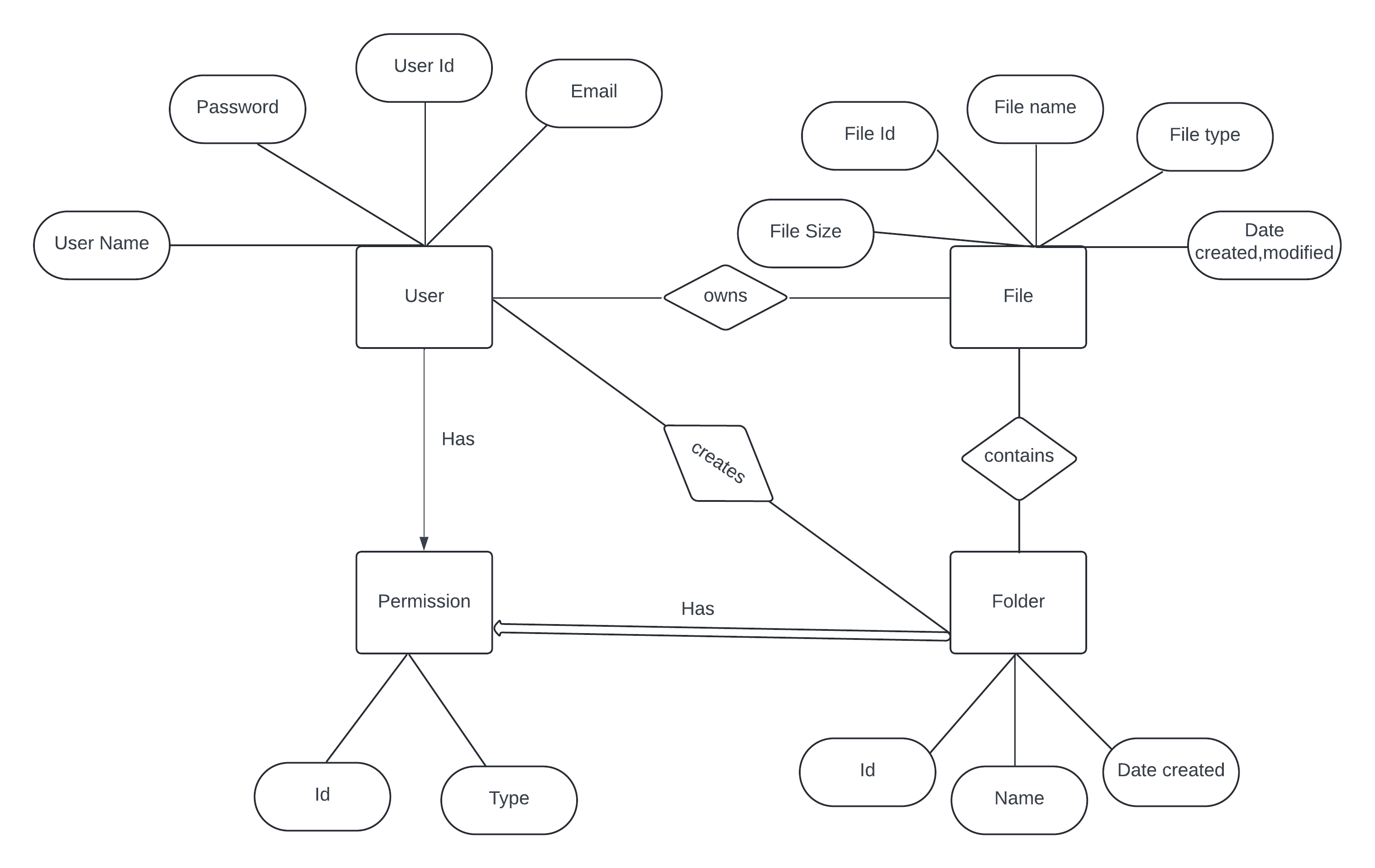


Figure 2: Entity-Relationship Diagram

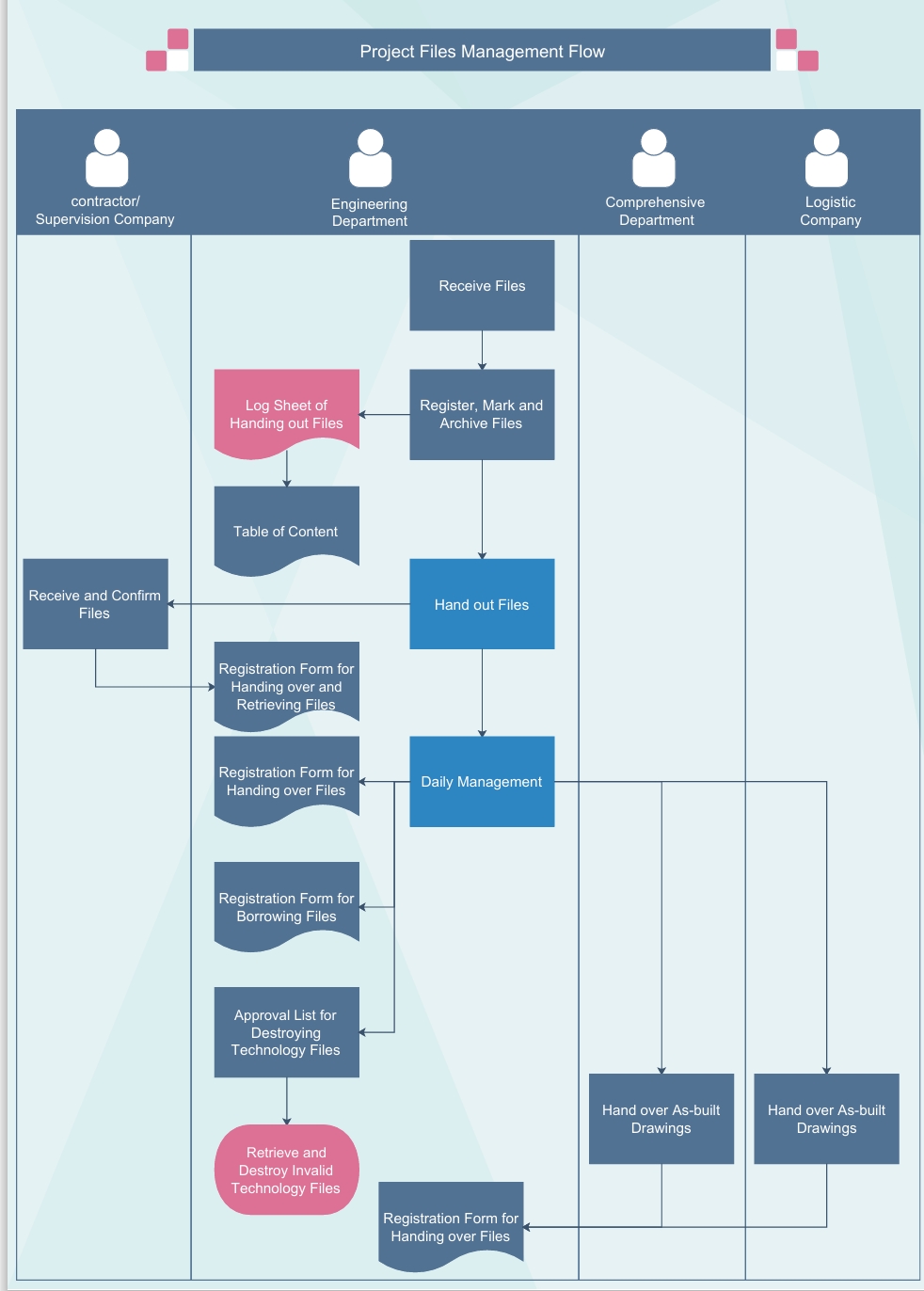


Figure 3: Flowchart

## Interfaces

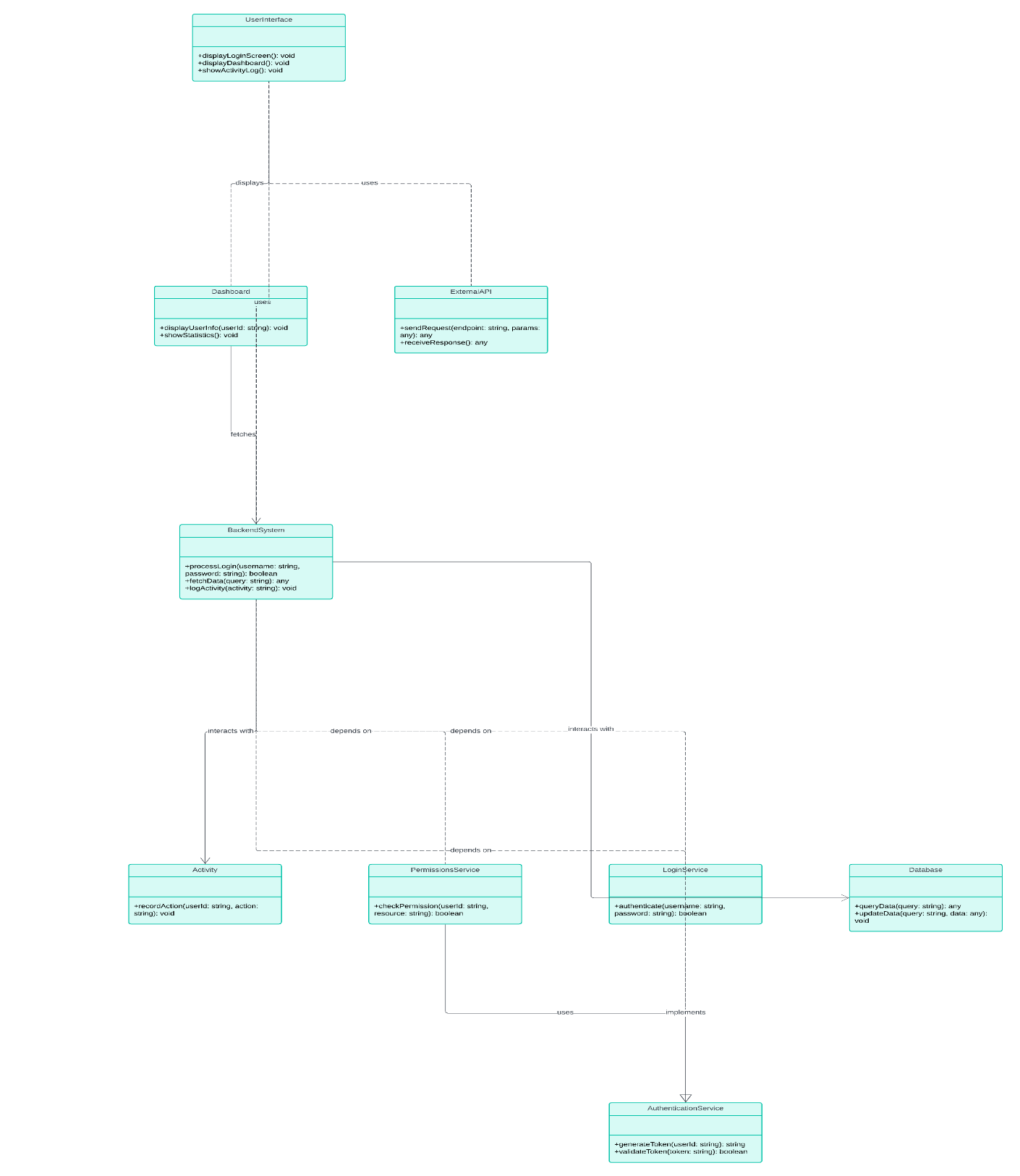


Figure 4: Interface UML Diagram

# Performance Test

## Test Plan/ Test Cases-

The test plan for the Python-based File Manager project outlines the strategy and scope of testing activities to ensure the system meets its functional and non-functional requirements. The goal is to verify that all features work as expected, are user-friendly, and perform efficiently. The testing will cover all aspects of the application, including the user interface, file operations, search functionalities, customization features, and cloud integration.

1. **User Interface (UI) Testing:**

* Objective: Verify that users can navigate through directories and access files smoothly.
* Expected Result: The navigation should be intuitive and fast, with no errors or crashes.

1. **File Operations Testing:**

**File Copy/Move/Deletion-**

* + Objective: Ensure that files can be copied and moved between directories correctly, and verify the deletion functionality.
  + Steps: Select a file, copy/move it to a different directory, and verify the operation.
  + Expected Result: The file should appear in the destination directory with the correct contents, and no errors should occur during the process.

1. **Search Functionality Testing:**

**Basic Search by Filename-**

* Objective: Test the search function by searching for a file by its name.
* Steps: Enter a file name in the search bar and execute the search.
* Expected Result: The search results should accurately display the file if it exists, with correct details.

1. **Cloud Integration Testing:**

**Cloud Sync-**

* Objective: Test the synchronization of files with a cloud storage service.
* Steps: Upload a file to a cloud service, sync it with the file manager, and verify that the file is available locally.
* Expected Result: The file should appear in the file manager and be accessible both locally and in the cloud.

**Cloud File Operations-**

* Objective: Ensure that file operations (copy, move, delete) work correctly with cloud files
* Steps: Perform file operations on files stored in the cloud through the file manager.
* Expected Result: Operations should be reflected in both the local and cloud storage without errors.

1. **Performance Testing:**

**Large Directory Handling-**

* Objective: Evaluate performance when navigating through directories with a large number of files.
* Steps: Open and browse a directory containing thousands of files.
* Expected Result: The file manager should handle large directories smoothly without significant delays or crashes.

Unit Testing:

Unit testing will be performed on each module of the system to ensure that the individual

components function as intended. The unit testing will be performed by the development team

using automated testing tools to ensure that the test results are accurate and reliabl

## Test Procedure

Unit Testing: Unit testing will be performed on each module of the system to ensure that the individual components function as intended. The unit testing will be performed by the development team using automated testing tools to ensure that the test results are accurate and reliable.

Integration Testing: Integration testing will be performed to ensure that the individual modules of the system function together as intended. The integration testing will be performed by the development team using automated testing tools and manual testing to ensure that the system is functioning correctly.

System Testing: System testing will be performed to ensure that the system functions as intended in a live environment. The system testing will be performed by the quality assurance team using manual testing methods to simulate user activity and test the system's functionalities.

**Functional Testing**: This stage involves testing the overall functionality of the File Manager as a complete system. Test cases are designed to cover all user interactions, such as browsing files, performing file operations, searching for files, customizing the interface, and using cloud integration features.

**Performance Testing**: The File Manager's performance is evaluated by testing it under various conditions, such as managing large volumes of files, handling complex search queries, and performing simultaneous file operations. The goal is to ensure the system remains responsive and efficient under different loads.

**User Acceptance Testing (UAT)**: The final stage involves testing the File Manager with real users to gather feedback on usability, functionality, and overall user experience. This helps identify any issues that might not have been caught in earlier testing stages.

## Performance Outcome-

The performance outcomes of the File Manager project are measured based on several key criteria:

1. Efficiency: The File Manager demonstrated efficient handling of large directories and complex file operations. The search engine was able to quickly return results even when dealing with extensive file systems.
2. Responsiveness: The user interface remained responsive during all tested scenarios, providing smooth navigation and interaction, even when performing multiple tasks simultaneously.
3. Reliability: The system successfully managed file operations without errors, ensuring data integrity and consistent behaviour across different operating environments.
4. User Satisfaction: Feedback from user acceptance testing indicated that users found the File Manager easy to use, with intuitive navigation and helpful features that improved their file management tasks.

The File Manager performed efficiently under various conditions. It successfully managed large volumes of files and performed searches quickly and accurately.

# My learnings

This internship provided significant learning opportunities in software development, project management, and problem-solving. It helped me understand industry practices and improved my technical skills in Python programming.

During this project internship, I gained valuable insights into the end-to-end development of a Python-based File Manager, from initial concept to final implementation. Key learnings include:

1. **Project Planning and Management**: I learned how to effectively plan and manage a software project, including defining requirements, setting milestones, and allocating time for development, testing, and revisions.
2. **Software Design and Architecture**: I developed a deep understanding of software architecture by designing both high-level and low-level components of the file manager. This included creating modular components such as the user interface, backend logic, and integration with external services like cloud storage.
3. **Python Programming and Libraries**: Through hands-on coding, I enhanced my skills in Python, particularly in areas such as file handling, working with operating system interfaces, and building graphical user interfaces (GUIs). I also learned how to utilize third-party libraries to extend the functionality of the project.
4. **Testing and Debugging**: I gained experience in writing test cases, performing unit and integration testing, and debugging the code to resolve issues. This taught me the importance of thorough testing to ensure software reliability and performance.
5. **Problem-Solving and Critical Thinking**: Throughout the project, I encountered various challenges that required creative problem-solving and critical thinking. This experience has improved my ability to troubleshoot issues and adapt solutions to meet user needs.

# Future work scope

The File Manager project has a solid foundation, but there is significant potential for future enhancements and additional features:

1. Advanced Search Capabilities: Incorporating AI-driven search algorithms could enable more intelligent file categorization and search functionalities based on content analysis and user behaviour.
2. Mobile Application Development: Expanding the File Manager to mobile platforms (iOS and Android) would provide users with a consistent file management experience across devices.
3. Cloud Service Expansion: Integrating additional cloud storage providers and offering seamless synchronization across multiple cloud services would further enhance the File Manager's utility.
4. Collaborative Features: Adding support for multi-user collaboration on shared files and directories, with access control and version history, would make the File Manager suitable for team-based environments.
5. Automation and Scripting: Introducing automation features, such as scheduled file backups or custom scripts for repetitive tasks, could greatly increase productivity for power users.

By exploring these future enhancements, the File Manager project can evolve into a comprehensive and versatile tool that meets the evolving needs of its users.

**9 Conclusion**

This project has been a comprehensive learning experience that allowed me to apply theoretical knowledge in a practical setting. The Python-based File Manager developed during this internship addresses key limitations of existing file management systems, offering enhanced customization, advanced search capabilities, and seamless cloud integration. The successful implementation of this project has strengthened my skills in software development, project management, and user-centred design.

The project not only met its objectives but also opened up avenues for future enhancements, such as mobile application development and the introduction of AI-driven search features. The knowledge and experience gained from this internship will undoubtedly be invaluable as I continue to pursue a career in software development, equipped with a deeper understanding of how to create robust, user-friendly applications.