Data Foundations System

Data Drive

Team - Red Flags

Team Members:

1. Karmanjyot Singh
2. Divya Varshini Yakkanti
3. Soveet Kumar Nayak

Data Drive Control & Web App

Overview

The document outlines the detailed requirements and specifications for the development of a Data Drive Web Application with integration of the MinIO object storage system. This web application aims to provide users with a powerful and versatile platform for managing and accessing their data assets efficiently and securely. The application will leverage MinIO, an open-source object storage solution, to ensure robust data storage capabilities.

The primary objective of this project is to create a web-based data drive application that facilitates the seamless upload, storage, retrieval, and management of data assets.

The project is a sub-part of the **Data Drive** project, which aims to build a web application with additional drive control.
As an integral component within the broader Data Drive project, our primary focus is dedicated to the development of a specialised web application interface and component. This pivotal component is designed to empower users with robust tools for efficient file management and streamlined access, all within the encompassing data drive environment, thereby significantly enhancing the overall user experience.

This effort entails the creation of a user-friendly interface and a versatile file management component tailored to facilitate seamless file handling, encompassing

both upload and download operations within the data drive framework. The key features and functionalities of this specialised interface and component include:

1. **Intuitive Interface:** The web application will boast an intuitive and user-friendly interface, ensuring easy navigation throughout the entire file system.

2. **Comprehensive File Management:** Users will have the capability to perform essential file operations, such as copying, moving, renaming, and deleting files, thereby enhancing their control over their data assets.

3. **Multiple File Selection:** The system will support the selection of multiple files at once, simplifying bulk operations and improving efficiency.

4. **Directory and File Upload:** Users will be able to upload both entire directories and individual files, enabling the swift transfer of data assets.

5. **File Sorting:** The component will offer flexible file sorting options, including sorting by name, size, and last modified date, in ascending or descending order, allowing users to organise their data as per their preferences.

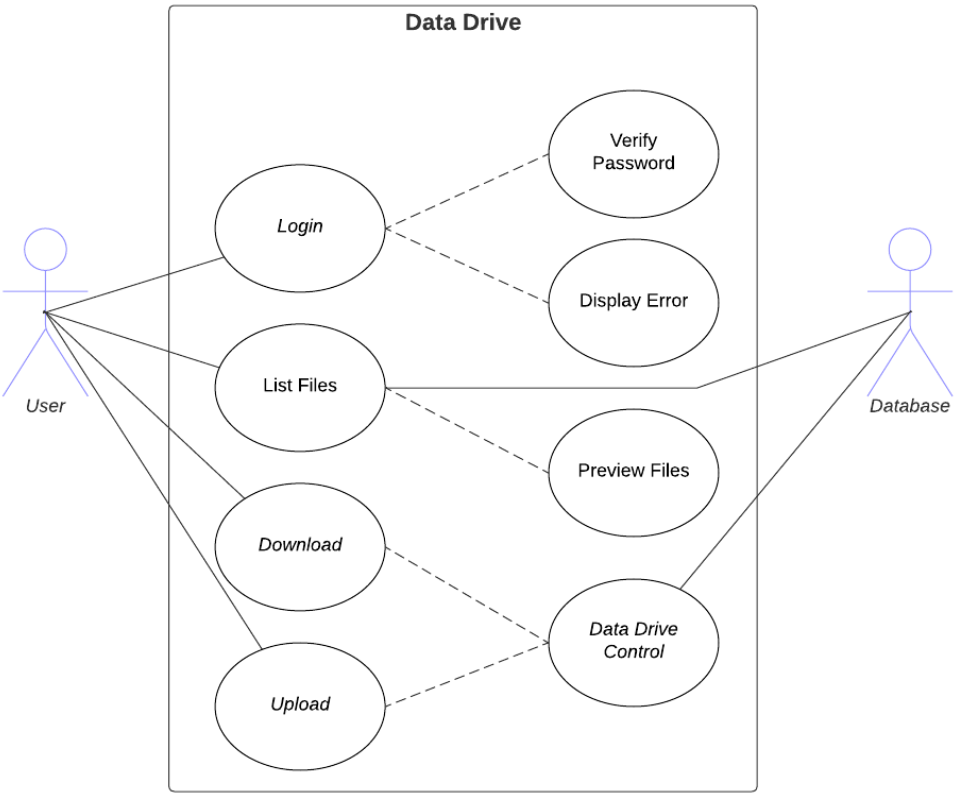
6. **Responsive Design:** The web application's layout will be highly responsive, ensuring a seamless user experience across various devices and screen sizes.

Our project aims to develop a versatile and user-friendly tool that facilitates seamless access to a data drive, akin to popular cloud storage services like Google Drive or OneDrive. This tool will empower applications to interact with the data drive, enabling key functionalities such as uploading, downloading, imposing restrictions on file types for uploads, and enabling multiple file selections.

System Requirements

1. Functional Requirements

    a. **File Navigation Panel:** We will provide users with an effortless way to traverse through the entire file system. This ensures that users can swiftly locate and interact with their desired files and directories. This panel will display the tree-based directory structure for navigating through the file system.
    b. **File Content Panel:** This panel displays the content of the current directory.
    c. **File Preview**: Users should have the ability to preview files after logging in. When a user selects a file (e.g., an image, document, or video), a file preview feature should be available, allowing users to view the contents of the file without needing to download it.
    d. **File Upload:** Our solution facilitates the seamless upload of both directories and individual files. This versatility caters to diverse user needs, allowing them to transfer data efficiently.
    e. **File Download:** Facilitate download of the selected file.
    f. **Multiple File Selection:** To boost productivity, our component supports the selection of multiple files at once. Users can conveniently choose and manage several files simultaneously, streamlining their workflow.
    g. **Sorting Files:** Users will have the ability to sort files based on various criteria, including name, size, and last modified date. Sorting can be done in both ascending and descending order, enabling users to find files quickly and efficiently.
    h. **User Authentication**: Users must be able to log in using a valid username and password combination. If a user enters an incorrect password, the system should display a login error message. It should not log in the user or provide access to the system.
    i. **Responsive Layout:** We understand the importance of accessibility across different devices and screen sizes. Our component is designed with a responsive layout, ensuring a consistent and user-friendly experience on desktops, tablets, and mobile devices.



2. Non-Functional Requirements

a. **Usability**: Design an intuitive and user-friendly interface that simplifies data asset management.
b. **Integration**: The component should be easily integrable across various components and tools.
c. **Security**: Enforce a strong password policy that includes complexity requirements to enhance security
d. **Performance:** Specify acceptable response times, file types for file upload, download, and navigation operations.
e. **Compatibility**: Supported across different browsers
f. **Maintainability**: Develop the application in a modular fashion, separating components for easier maintenance and updates. We will use version control systems like git to manage and track changes to the codebase.

Project Deliverables

1. **Web Application:** The primary deliverable is the web application itself. This application will have all the features and functionalities that were planned for the project. These features could include user authentication, data management, reporting, communication, and any other functionalities specific to project's requirements.

2. **Cross-Platform Compatibility:** The web application will be designed to run on any machine or device with a compatible web browser. This means ensuring that the application is responsive and works well on various screen sizes and operating systems.

3. **Drive Control Tool:** This tool is a separate component or feature of the web application. It allows users to upload and download relevant files. Here are some additional details about this tool:
    a. **Upload Functionality:** Users should be able to upload files to the application. Implement file validation and constraints to ensure that only appropriate files can be uploaded.
    b. **Download Functionality:** Users should be able to download files from the application. This could involve providing a list of available files, allowing users to select files for download, and then providing a download link or mechanism.
    c. **Constraints:** These constraints could include file size limits, file type restrictions, access permissions, or any other rules that ensure data security and compliance with your project's requirements.

4. **Documentation:** In addition to the application itself, it's crucial to provide documentation for users and developers. This documentation will include user guides, installation instructions, and technical documentation for developers who may need to work on or maintain the application in the future.

5. **Testing and Quality Assurance:** We shall ensure that the application is thoroughly tested to identify and fix any bugs or issues. This involves unit testing, integration testing, and user acceptance testing to ensure that the application functions as intended.