

# Florida International University

Fall 2023 Senior Design Project

## NextCloud App: Sync CSV Userlist

**Students:** Augusto Rodriguez

**Mentor:** Masoud Sadjadi, Professor

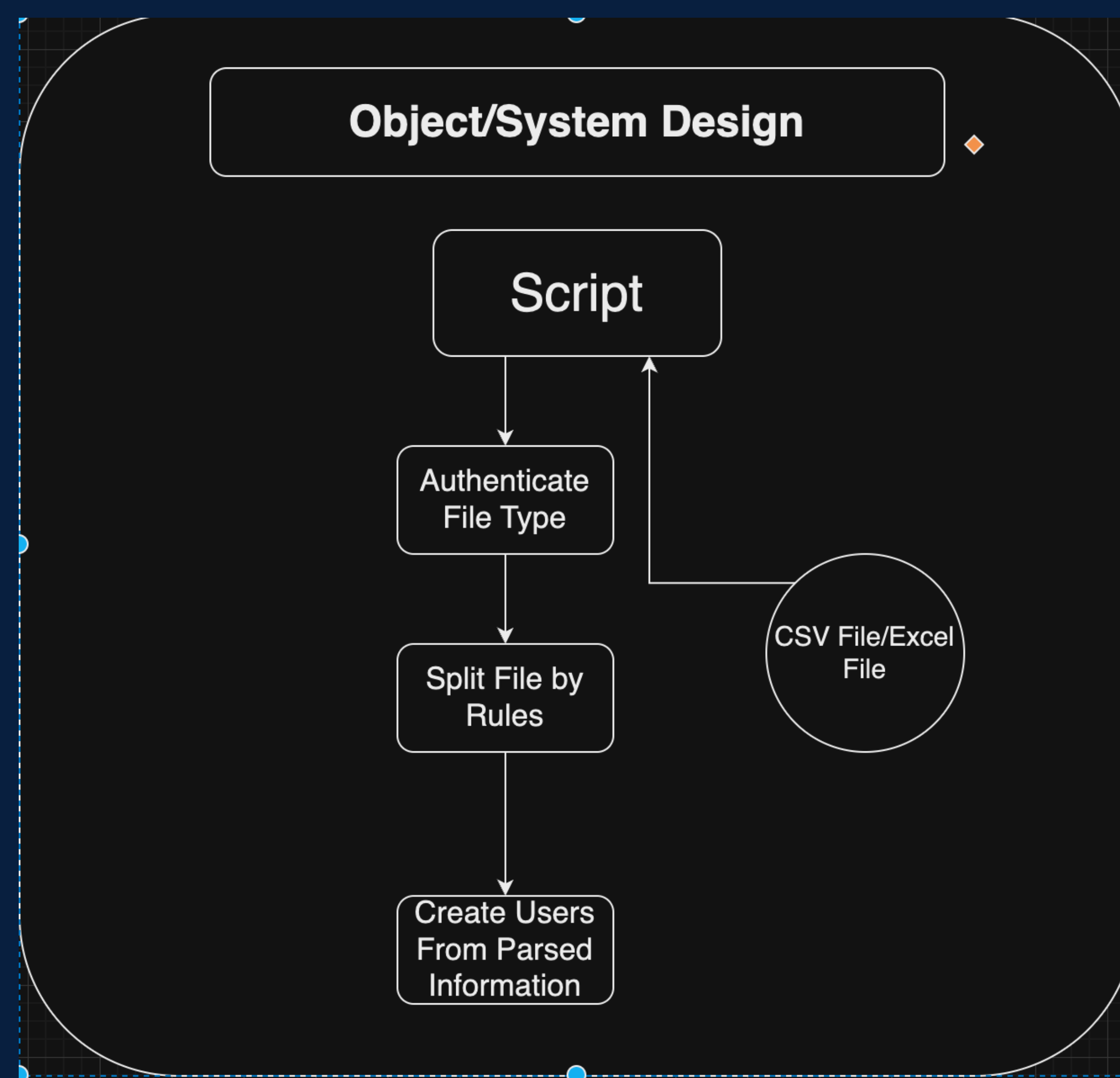
**Instructor/Faculty:** Masoud Sadjadi , Florida International University

### PROBLEM

Manually adding students to a NextCloud application can pose several challenges, primarily stemming from the potential for human error and the lack of efficiency. Inaccuracies may arise during data entry, such as misspelled names or incorrect user details, leading to complications in user identification and access. Additionally, the manual process can be time-consuming and labor-intensive, especially in educational institutions with a large student population. This approach may result in delays in granting access to essential resources and collaborative tools, hindering the seamless integration of NextCloud into the educational workflow. Furthermore, maintaining accurate and up-to-date records becomes increasingly challenging as the student body evolves, creating a risk of outdated or incomplete information within the application.

### SYSTEM DESIGN

Designing an automated system for user implementation in NextCloud involves creating a robust script or tool to parse CSV or Excel files containing user details. The system should securely authenticate with NextCloud, handle diverse user attributes, and offer customization options. Error handling and logging mechanisms are crucial, and regular synchronization ensures up-to-date user information alignment. A well-thought-out system design ensures efficiency, security, and adaptability for automating the user implementation process in NextCloud.



### CURRENT SYSTEM

The current system in the NextCloud app, which involves manually importing students into classes, presents inherent challenges and inefficiencies. This manual process requires significant time and effort, making it prone to human errors such as data entry mistakes and omissions. The lack of automation hinders the seamless integration of students into classes, leading to potential delays in accessing essential resources. Additionally, the manual importation method becomes increasingly cumbersome as the student body grows, making it difficult to maintain accurate and up-to-date records. To optimize user onboarding and ensure a more streamlined and error-resistant approach, the integration of automated features for importing students into classes would be highly beneficial, reducing administrative burdens and enhancing overall system efficiency.

### OBJECT DESIGN

The object design for automating user implementation in NextCloud involves creating modular objects like UserImporter, UserManager, and Authentication. Each object has specific responsibilities, ensuring a clean and maintainable codebase. Incorporating error-handling and logging features enhances reliability, while the design allows for scalability and adaptability to future application requirements.

### SUMMARY

The recent update to the NextCloud application brings a series of valuable enhancements to the user management process. Notable fixes include resolving a bug related to different file formats that previously caused an incorrect division of imported files. To enhance user experience, a comprehensive README template has been created, offering detailed guidance on project setup, functionality, usage, and troubleshooting. The update also introduces the ability to add users through external CSV files directly within NextCloud, streamlining the onboarding process. Default settings for user sync folders have been added for convenience, and quality-of-life changes, such as auto-updating prefixes, have been implemented. Furthermore, the update incorporates a prefix functionality for file names and introduces a mapping feature, both of which contribute to a more customizable and intuitive file management system. The mapping file functionality has been refined for improved usability, collectively making user administration and file organization more efficient and user-friendly.

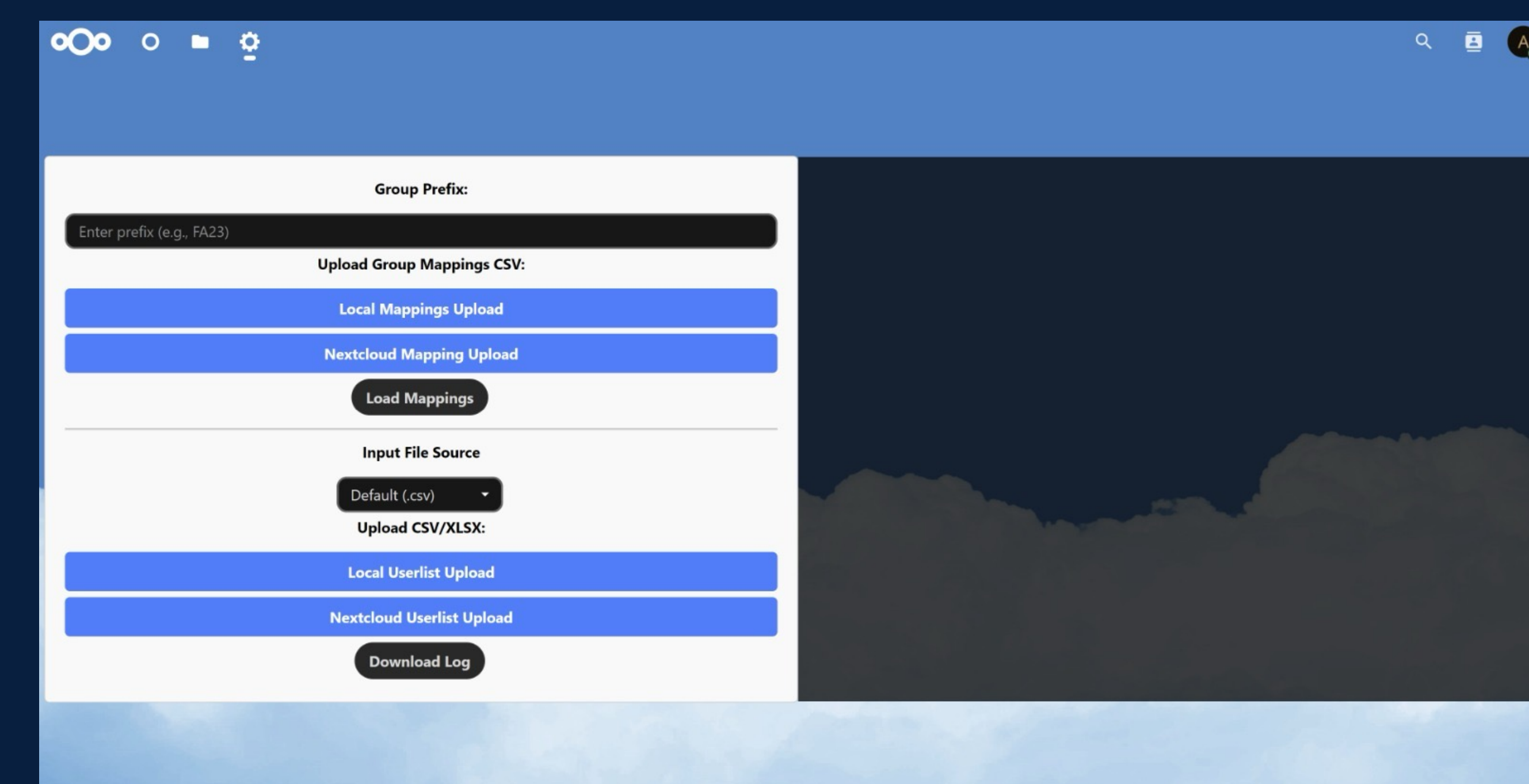
### REQUIREMENTS

Automating user addition to NextCloud from an external file requires an accurate data file in CSV or Excel format, including essential user details. A secure script or tool should be employed to seamlessly import data, with proper authentication credentials. The solution must handle diverse user attributes, allowing customization within NextCloud. Regular synchronization capabilities are crucial for maintaining up-to-date user information, ensuring an efficient and reliable process.



### IMPLEMENTATION

In implementing a NextCloud application, I introduced key features to enhance functionality. Firstly, the addition of a prefix functionality provided users with the ability to append prefixes to file names, promoting better organization and categorization of data. Secondly, I incorporated a mapping functionality, allowing users to define specific mappings for their files. This feature facilitated a more customized and intuitive organization of data within the application. In response to user feedback, I further refined the mapping file functionality, ensuring a more user-friendly and efficient process for managing file mappings. These enhancements collectively aimed at improving user experience and providing a more versatile and personalized approach to file management within the NextCloud application.



### ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by XXXXX (name of the project sponsor: federal, state or local government, or corporate partners, where applicable). We/I thank XXXX for his/her/their assistance and mentorship that I/we received throughout the senior design project.