

Florida International University

Fall 2023 Senior Design Project

NextCloud App: Sync CSV Userlist

Students: Adam Kobke

Mentor: Masoud Sadjadi, Professor

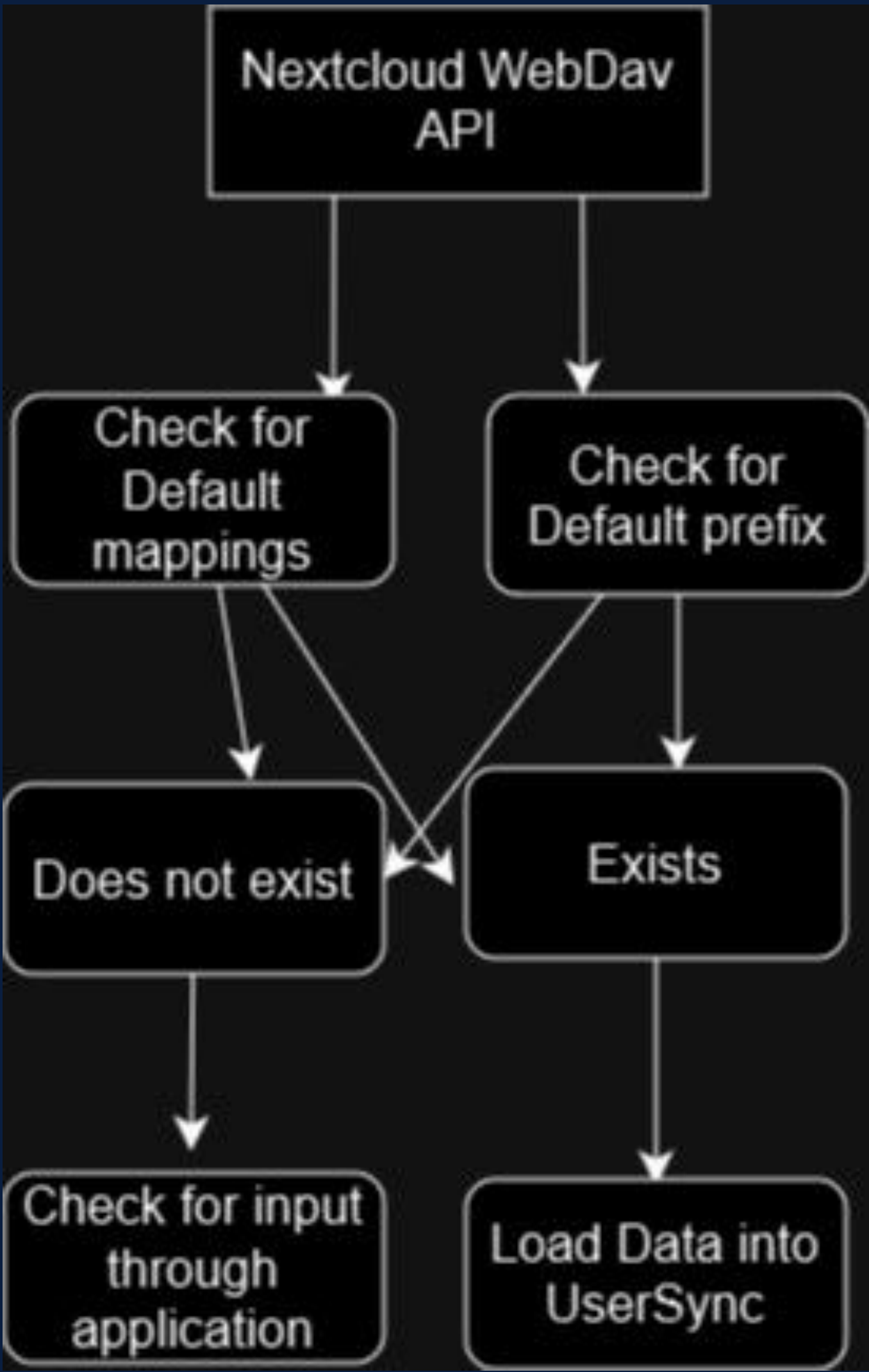
Instructor/Faculty: Masoud Sadjadi , Florida International University

PROBLEM

Adding users to Nextcloud is a simple procedure, but for organizations with a large student body like FIU, this can become a very tedious and time consuming task. This can also lead to increases in human error, professor fatigue and more. This long and tedious process has other challenged that waste both time and resources. For instance manually adding and updating users can delay students from accessing critical course information or course resources. Our application allows professors to create, update, and remove users based on readily available CSV files. This frees up course resources for other possibly more important tasks.

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 UserSync update brings several valuable enhancements to user management. Notable fixes address a bug related to different file formats that previously caused inaccurate division of imported files. To improve user experience, a comprehensive README template has been developed, providing detailed guidance on project setup, functionality, usage, and troubleshooting.

A notable addition is the capability to add users through external CSV files directly within NextCloud, streamlining the onboarding process. Default settings for user sync folders have been introduced for added convenience, along with quality-of-life changes such as the implementation of auto-updating prefixes.

Furthermore, the update incorporates a prefix functionality for file names and introduces a mapping feature, both contributing to a more customizable and intuitive user management system. The mapping file functionality has been refined for improved usability, collectively enhancing user administration and file organization for increased efficiency and user-friendliness.

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 began by crafting a comprehensive README template, offering clear instructions on project setup, functionality, usage, and troubleshooting. I enhanced user experience by incorporating NextCloud's CSV uploading feature, facilitating seamless data import. To streamline user synchronization, default folders were established, minimizing user input and ensuring a consistent experience. Additionally, I addressed quality-of-life concerns by fixing auto-updating prefixes and making other user-friendly adjustments. This approach not only prioritized functionality but also emphasized user understanding and convenience, promoting a more accessible and efficient experience for NextCloud application users.

