TỰ ĐỘNG HÓA VÀ QUẢN LÝ CẤU TRÚC THƯ MỤC HỌC TẬP TRÊN GOOGLE DRIVE BẰNG GOOGLE APPS SCRIPT

Nguyễn Hữu Hoàng Sang*, Phan Thị Hồng Nhung, Phan Gia Đại, Nguyễn Đức Hải, Lương Trần Ngọc Khiết, Lương Trần Hy Hiến, Phan Thị Nam Anh

Khoa Công nghệ thông tin, Trường Đại học Sư phạm TP.HCM

4901104124@student.hcmue.edu.vn, 4901104104@student.hcmue.edu.vn, 4901104028@student.hcmue.edu.vn, 4901104038@student.hcmue.edu.vn, khietltn@hcmue.edu.vn, hienlth@hcmue.edu.vn, tg.anhptn@lecturer.hcmue.edu.vn

TÓM TẮT— Trong bối cảnh chuyển đổi số trong giáo dục, việc quản lý thủ công hàng nghìn thư mục học tập trên Google Drive gây tốn thời gian và dễ phát sinh sai sót, đặc biệt khi áp dụng mô hình học tập theo dự án (PBL). Nghiên cứu này đề xuất và phát triển một hệ thống tự động hóa dựa trên Google Apps Script nhằm tạo lập và quản lý cấu trúc thư mục phân cấp, đồng bộ dữ liệu thời gian thực với Google Sheets. Hệ thống tích hợp giao diện kéo-thả hỗ trợ thiết kế cấu trúc và mô đun quản lý quyền truy cập tự động (RBAC). Hệ thống được đánh giá qua một nghiên cứu bán thực nghiệm (quasi-experimental) với thiết kế trước-sau có nhóm đối chứng, thực hiện trên 10 lớp học (N=355 sinh viên, 20 giảng viên). Kết quả định lượng cho thấy nhóm thực nghiệm giảm thời gian quản lý thư mục tới 90-95% so với nhóm đối chứng ($p < .001$, Cohen's $d > 4.0$). Kết quả khảo sát định tính cho thấy mức độ hài lòng cao (4.31/5.0). Hệ thống chứng minh tính hiệu quả, khả năng mở rộng và tiềm năng ứng dụng như một giải pháp "low-code/no-cost" thực tiễn cho các cơ sở giáo dục đang sử dụng nền tảng Google Workspace, giải quyết hiệu quả các thách thức về quản lý dữ liệu phân cấp trong môi trường học thuật.

Từ khóa— Google Apps Script, Google Drive, tự động hóa giáo dục, quản lý thư mục, kiểm soát truy cập.

I. GIỚI THIỆU

Trong bối cảnh giáo dục số hóa, phương pháp dạy học theo dự án (Project-Based Learning – PBL) đã trở thành một xu hướng chủ đạo tại các cơ sở đào tạo đại học và cao đẳng [1]. Mô hình này đòi hỏi quá trình quản lý tiến độ học tập phức tạp, đồng thời phát sinh khối lượng lớn dữ liệu và tài liệu học thuật được tạo ra liên tục từ các hoạt động như bài tập thực hành và bài trình bày. Do đó, việc lưu trữ, tổ chức của từng dự án, lớp học hoặc nhóm sinh viên cần đến một hệ thống quản lý thư mục có cấu trúc chặt chẽ, nhằm đảm bảo tính nhất quán, tính quy tắc và khả năng truy xuất hiệu quả.

Google Drive đã trở thành một trong những nền tảng lưu trữ trực tuyến phổ biến trong môi trường giáo dục nhờ tính tiện lợi và khả năng truy cập cao [2]. Việc hầu hết giảng viên và sinh viên đều sở hữu tài khoản Google đi kèm địa chỉ email mang lại lợi thế đáng kể trong triển khai các hệ thống quản lý tài liệu học tập. Khai thác nền tảng này không chỉ rút ngắn thời gian triển khai mà còn loại bỏ nhu cầu tạo lập tài khoản mới, giảm thiểu chi phí đào tạo người dùng, đồng thời tận dụng tối đa hạ tầng công nghệ sẵn có.

Thực tiễn triển khai cho thấy việc tổ chức và quản lý tài liệu học tập trên Google Drive gặp nhiều thách thức [3]. Khó khăn xuất phát từ yêu cầu phân quyền truy cập cho từng nhóm sinh viên và giảng viên theo từng dự án, quản lý danh sách nhóm có thành viên thay đổi liên tục, xây dựng cây thư mục chuẩn hóa cho hàng chục lớp học song song, cũng như lưu trữ và kiểm soát metadata phức tạp (bao gồm số lượng tệp, loại tệp cho phép và hạn nộp bài). Tuy nhiên, Google Drive mặc định chỉ hỗ trợ một số thao tác thủ công cơ bản cho các tác vụ này, dẫn đến tình trạng tốn nhiều thời gian, thiếu tính nhất quán và dễ phát sinh sai sót trong quá trình quản lý trên quy mô lớn.

Nhận diện được vấn đề này, nhiều Hệ thống quản lý học tập (LMS) thương mại như Canvas [4], Microsoft SharePoint [5] hay các dịch vụ lưu trữ doanh nghiệp như Box [6] đã cung cấp các giải pháp quản lý tệp tin. Tuy nhiên, các giải pháp này thường gặp phải ba hạn chế chính trong bối cảnh nghiên cứu: (1) tính linh hoạt thấp: cấu trúc thư mục thường cố định theo thiết kế của LMS, khó tùy chỉnh sâu theo đặc thù của từng dự án hoặc yêu cầu riêng của giảng viên; (2) chi phí và hạ tầng: việc triển khai các hệ thống này đòi hỏi chi phí bản quyền và hạ tầng máy chủ, tạo rào cản cho các cơ sở giáo dục có ngân sách hạn hẹp; (3) tính hệ sinh thái: chúng thường thiếu sự tích hợp liền mạch và sâu rộng với bộ công cụ Google Workspace (Drive, Sheets, Forms) vốn đã được sử dụng miễn phí và rộng rãi bởi hàng triệu sinh viên và giảng viên [2], [4].

Ngay cả giải pháp có sẵn như Google Classroom, mặc dù tích hợp với Google Drive, cũng chỉ cung cấp cấu trúc thư mục cơ bản và không hỗ trợ khả năng tự động tạo các cấu trúc phân cấp sâu, phức tạp theo nhóm sinh viên, cũng như không cho phép quản lý metadata và quy tắc (rules) một cách linh hoạt.

Do đó, khoảng hở nghiên cứu (research gap) được xác định là thiếu một giải pháp "low-code/no-cost" (ít mã nguồn/không tốn phí), có khả năng tùy biến cao, vận hành trực tiếp trên nền tảng Google Workspace sẵn có. Giải pháp này cần trao quyền cho giảng viên tự thiết kế cấu trúc thư mục (thông qua giao diện kéo-thả), tự động hóa

*Corresponding Author

việc tạo lập hàng loạt và đồng bộ hóa quyền truy cập theo thời gian thực từ một nguồn dữ liệu duy nhất (Google Sheets).

Nghiên cứu này hướng tới việc phát triển một module tự động hóa nhằm giải quyết một khía cạnh quan trọng trong hệ thống quản lý học thuật tổng thể, đó là xây dựng và quản lý cấu trúc cây thư mục cho các lớp học và nhóm sinh viên. Module được thiết kế với ba chức năng cốt lõi: (1) tự động sinh template cấu trúc thư mục dựa trên thiết kế của giảng viên, (2) ghi nhận và đồng bộ hóa cấu trúc với Google Sheets để hỗ trợ quản lý tập trung, và (3) tự động tạo thư mục thực tế trên Google Drive kèm theo phân quyền truy cập phù hợp. Đây được xem là một thành phần then chốt trong quá trình nghiên cứu và phát triển hệ thống quản lý học thuật toàn diện.

Hệ thống được phát triển cung cấp giao diện trực quan cho phép giảng viên thiết kế mẫu cây thư mục bằng công nghệ kéo–thả (drag-and-drop), tự động lưu cấu trúc vào Google Sheets, sinh thư mục thực tế trên Google Drive, đồng thời đồng bộ hóa với bảng quy tắc (Rules) để quản lý dữ liệu chi tiết. Giải pháp này mang lại nhiều lợi thế nổi bật: tiết kiệm khoảng 80% thời gian so với thao tác thủ công, vận hành hoàn toàn miễn phí trên nền tảng Google Workspace, không phát sinh chi phí hạ tầng hoặc bản quyền bổ sung, và đặc biệt phù hợp với bối cảnh quản lý học thuật tại các trường đại học - cao đẳng.

Các phần tiếp theo sẽ trình bày chi tiết về nguồn dữ liệu và cấu trúc hệ thống, giao diện người dùng và chức năng tạo cấu trúc, quy trình sinh tạo thư mục tự động, hệ thống quản lý quyền truy cập (Permission), kết quả đánh giá hiệu quả, và hướng phát triển trong tương lai.

II. NGUỒN DỮ LIỆU VÀ CẤU TRÚC HỆ THỐNG

A. Ý TƯỞNG

Việc tạo cấu trúc thư mục tự động trong môi trường giáo dục đòi hỏi một quy trình có hệ thống để thu thập, phân loại và tổ chức thông tin học tập. Trước hết, hệ thống cần xác định cấu trúc cây thư mục phù hợp với từng môn học và hoạt động cụ thể, bao gồm các thư mục chính như Lab, Assignment, Project cùng các thư mục con tương ứng. Song song đó, hệ thống phải thu thập và quản lý đầy đủ dữ liệu sinh viên và nhóm, bao gồm tên lớp, danh sách nhóm, thông tin thành viên và đặc biệt là địa chỉ email của từng sinh viên để phục vụ việc thiết lập quyền truy cập tự động. Thông tin này được tổ chức theo mô hình phân cấp rõ ràng: lớp → nhóm → sinh viên, trong đó nhấn mạnh việc quản lý email của cả trưởng nhóm và thành viên. Trên cơ sở dữ liệu này, hệ thống sẽ tiến hành sinh tự động cây thư mục theo mẫu định sẵn, đảm bảo tính đồng nhất và chính xác giữa các nhóm. Đồng thời, giảng viên có thể tùy chỉnh số lượng cấp độ, tên gọi thư mục chính hoặc bổ sung các thư mục con để phù hợp với đặc thù từng môn học. Cách tiếp cận này không chỉ tiết kiệm thời gian và công sức, mà còn giúp đảm bảo sự thống nhất và khoa học trong việc quản lý tài nguyên học tập của cả lớp.

B. NGUỒN DỮ LIỆU CHÍNH VÀ CÁC SHEET QUẢN LÝ

Hệ thống sử dụng Google Sheets làm nguồn dữ liệu chính với 3 sheet chuyên biệt, theo mô hình được đề xuất bởi Google for Education (2024) cho việc quản lý dữ liệu giáo dục [7] sử dụng Google Sheets làm nguồn dữ liệu trung tâm, trong đó dữ liệu gốc được lấy từ bảng *Class List* – danh sách lớp học và sinh viên đã có sẵn (Hình 1). Bảng này đóng vai trò làm cơ sở dữ liệu đầu vào để quản lý cấu trúc lớp học và nhóm sinh viên. Các thành phần thông tin chính bao gồm: tên lớp (classname), tên nhóm (groupname), danh sách thành viên, mã số sinh viên, và địa chỉ email của từng cá nhân. Đặc biệt, trường email được sử dụng làm khóa định danh để thiết lập quyền truy cập tự động trong quá trình tạo thư mục [2].

Class Name	Group name	Group leader full nam	Student ID	Email address	Member 1 full n	Student ID (Mem	Email address 1
COMP1316	Nhóm 1	Phan Thị Hồng Nhung	49.01.104.104	nhungpth.cnthongtin@gmail.co	Nguyễn Nam	49.01.104.156	namn123@gmail.com
COMP1316	Nhóm 2	Nguyễn Thị Mai Nhung	49.10.104.657	nhungpth.fit293@gmail.com	Lê Trần Hiếu	49.01.104.789	hieult890@gmail.com
COMP1316	Nhóm 3	Nguyễn Thị Linh	49.01.104.555	hongrong.shz@gmail.com	Nguyễn Văn Đạt	49.01.104.778	datnv319@gmail.com
COMP1317	Nhóm 3	Nguyễn Thị Linh	49.01.104.555	nhhs1509@gmail.com	Nguyễn Văn Đạt	49.01.104.778	nguyennhac211@gmail

Hình 1. *Sheet Class list*

Trên cơ sở nguồn dữ liệu này, hệ thống triển khai hai sheet chức năng quản lý:

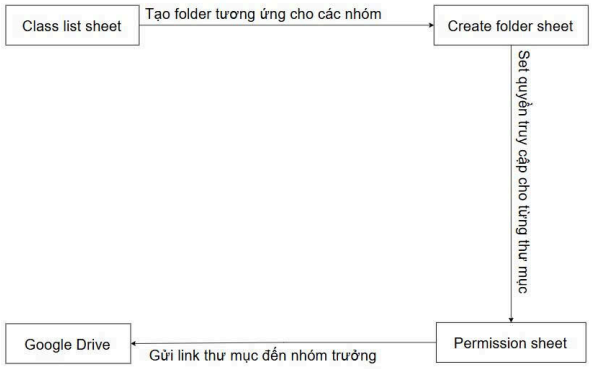
Sheet "Create Folder" - Lưu trữ cấu trúc thư mục đã được generate:

- Chứa cấu trúc cây thư mục dưới dạng bảng phẳng.
- Tự động cập nhật khi có thay đổi từ giao diện người dùng.
- Làm nguồn dữ liệu cho việc tạo thư mục thực tế trên Google Drive.

Sheet "Permission" - Quản lý quyền truy cập chi tiết:

- Định nghĩa mức quyền cho từng người dùng hoặc nhóm.
- Thiết lập quyền truy cập phân cấp theo cấu trúc thư mục.
- Cho phép ghi đè (override) quyền mặc định cho các trường hợp đặc biệt.

Cách tổ chức này đảm bảo dữ liệu luôn được duy trì tập trung, đồng bộ và có khả năng mở rộng, đồng thời hỗ trợ việc kiểm soát quyền truy cập cũng như quản lý quy tắc chặt chẽ trong toàn bộ hệ thống (Hình 2).



Hình 2. *Sơ đồ quan hệ*

C. KIẾN TRÚC HỆ THỐNG 4 LỚP

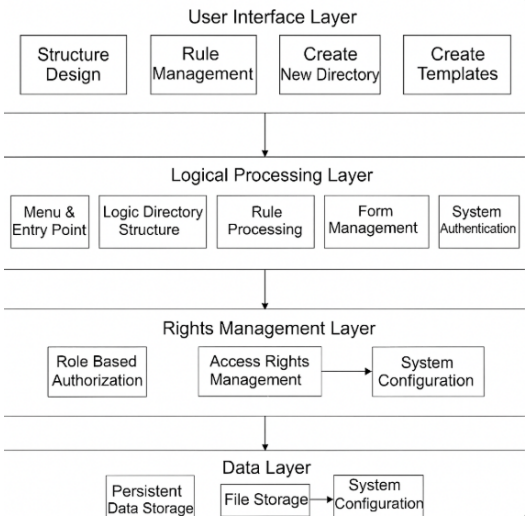
Hệ thống được thiết kế theo mô hình kiến trúc 4 lớp (Hình 3) để đảm bảo tính modular và khả năng bảo trì, dựa trên các phương pháp thực hành tốt nhất được đề xuất trong nghiên cứu của Vũ Thanh Hương (2024) về tự động hóa quy trình quản lý học thuật [8].

Lớp giao diện (Frontend Layer): Bao gồm các giao diện HTML/CSS/JavaScript cho phép người dùng tương tác trực quan với hệ thống.

Lớp xử lý logic (Controller Layer): Được triển khai trong Google Apps Script với các module chuyên biệt xử lý các quy tắc nghiệp vụ [9].

Lớp quản lý quyền (Permission Layer): Layer mới được thêm vào để xử lý phân quyền phức tạp dựa trên sheet Permission, đáp ứng các yêu cầu bảo mật trong môi trường giáo dục [10].

Lớp dữ liệu (Data Layer): Sử dụng Google Sheets cho lưu trữ dữ liệu tạm thời và Google Drive cho lưu trữ vật lý.



Hình 3. *Sơ đồ minh họa kiến trúc hạ tầng*

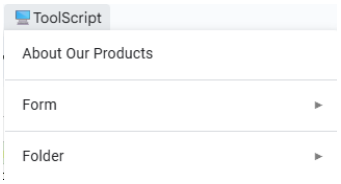
III. QUY TRÌNH HOẠT ĐỘNG

A. MÔ TẢ CHỨC NĂNG

1. HỆ THỐNG MENU PHÂN CẤP TÍCH HỢP

Hệ thống được tích hợp trực tiếp vào Google Sheets thông qua thanh công cụ tùy chỉnh "ToolScript" (Hình 4), được khởi tạo tự động mỗi khi mở bảng tính. Thiết kế thanh công cụ tuân theo nguyên tắc kiến trúc thông tin (Information Architecture), với cấu trúc phân cấp hợp lý và định hướng theo người dùng, giúp nâng cao khả năng khám phá và tiếp cận ngay cả đối với người mới. Để tối ưu trải nghiệm, thanh công cụ được tổ chức thành các cụm chức năng có liên quan chặt chẽ, áp dụng nguyên tắc chia khối thông tin nhằm giảm tải nhận thức trong quá trình thao tác. Mỗi cụm được thiết kế theo quy ước đặt tên nhất quán và bổ sung biểu tượng mô tả trực quan, từ đó nâng

cao tính dễ sử dụng, hạn chế sai sót và đảm bảo sự thân thiện đối với người dùng ở nhiều mức độ kỹ năng khác nhau.



Hình 4. *Thanh công cụ*

2. PHÂN TÍCH CHỨC NĂNG

Để hiện thực hóa mục tiêu tự động hóa và tối ưu hóa quản lý thư mục học tập, hệ thống được thiết kế xoay quanh ba cụm chức năng chính, mỗi cụm đảm nhiệm một vai trò cụ thể và bổ trợ cho nhau (Hình 5).

Cụm 1 - Quản lý Thư mục (Folder Management):

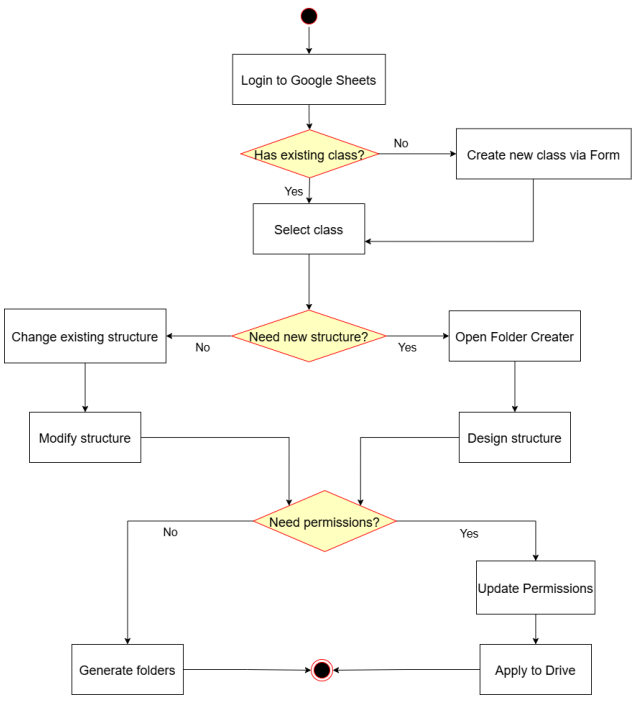
- Quy trình chính: Tạo → chỉnh sửa → trực quan hóa
- "Open Folder Creator": Điểm bắt đầu để thiết lập dự án mới
- "Change Folder Structure": Quy trình chỉnh sửa cho các cấu trúc đã có
- Mẫu thiết kế: Tiết lộ dần (Progressive disclosure) để tránh gây quá tải cho người dung

Cụm 2 - Quản lý Biểu mẫu (Form Management):

- Quy trình thu thập dữ liệu: Tạo → phân phối → đồng bộ hóa
- "Form Creation": Tích hợp Google Forms với các mẫu dựng sẵn
- "Sync Result": Quy trình ETL từ phản hồi biểu mẫu về dữ liệu có cấu trúc
- "Download class list": Chức năng xuất dữ liệu với nhiều định dạng hỗ trợ

Cụm 3 - Tiện ích Hệ thống (System Utilities):

- Quy trình bảo trì: Giám sát → xử lý sự cố → khôi phục
- "Update Permissions": Quản lý quyền hàng loạt
- "Clear Cache/Sheet": Giữ vệ sinh hệ thống và tối ưu hiệu năng



Hình 5. *Sơ đồ luồng thao tác người dùng*

3. GIAO DIỆN NGƯỜI DÙNG

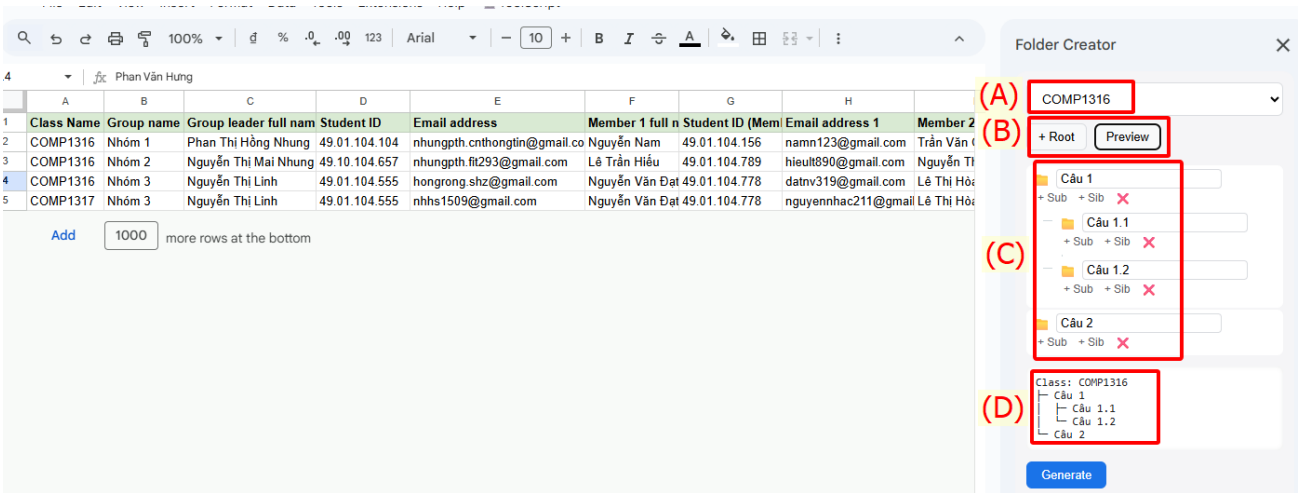
Hệ thống được xây dựng dựa trên các mẫu thiết kế đã được kiểm chứng và nguyên tắc khả dụng, nhằm đảm bảo vừa thân thiện với người dùng vừa duy trì tính ổn định lâu dài (Hình 6).

- **Vùng [A] - Tăng cường dần (Progressive Enhancement):** Chức năng được phát triển theo từng lớp, trong đó các tính năng cơ bản luôn được đảm bảo hoạt động ổn định, sau đó mới bổ sung thêm các tính năng nâng cao. Cách tiếp cận này giúp hệ thống linh hoạt và dễ dàng mở rộng. "Dropdown" chọn lớp hiển thị đầu tiên, đảm bảo chức năng cơ bản luôn hoạt động. Nếu không load được dữ liệu, vẫn hiển thị "Loading..." thay vì lỗi dừng chương trình (crash).

- **Vùng [B] - Suy giảm có kiểm soát (Graceful Degradation):** Ngay cả khi một thành phần gặp lỗi hoặc không khả dụng, các chức năng cốt lõi vẫn duy trì hoạt động. Điều này đảm bảo hệ thống không bị gián đoạn hoàn toàn và người dùng vẫn tiếp tục làm việc. Các nút [+ Root], [Preview], [View] vẫn hiển thị và có thể click ngay cả khi một chức năng gặp lỗi. Ví dụ nếu [View] lỗi, [+ Root] vẫn cho phép tạo cấu trúc mới.

- **Vùng [C] - Vòng phản hồi (Feedback Loops):** Mọi thao tác của người dùng đều nhận được phản hồi trực quan tức thì, ví dụ như thông báo trạng thái, biểu tượng tiến trình, hay xác nhận thành công hoặc thất bại. Nhờ đó, người dùng có thể nhận biết và điều chỉnh hành vi kịp thời. Mỗi thao tác thêm hoặc xóa folder đều có phản hồi tức thì qua việc cập nhật cây thư mục. Khi người dùng di chuột lên nút chức năng sẽ có hiệu ứng làm nổi bật, và khi nhấn nút sẽ xuất hiện phản hồi trực quan.

- **Vùng [D] - Ngăn ngừa lỗi (Error Prevention):** Hệ thống tích hợp cơ chế kiểm tra đầu vào và các hộp thoại xác nhận trước khi thực hiện hành động quan trọng, đặc biệt là các thao tác có tính hủy hoại dữ liệu, nhằm giảm thiểu rủi ro và sai sót. "Preview" hiển thị cấu trúc trước khi thực thi, cho phép người dùng kiểm tra và điều chỉnh. Đồng thời, việc kiểm tra tên thư mục và độ sâu của cây thư mục được thực hiện tức thời trong quá trình thao tác.



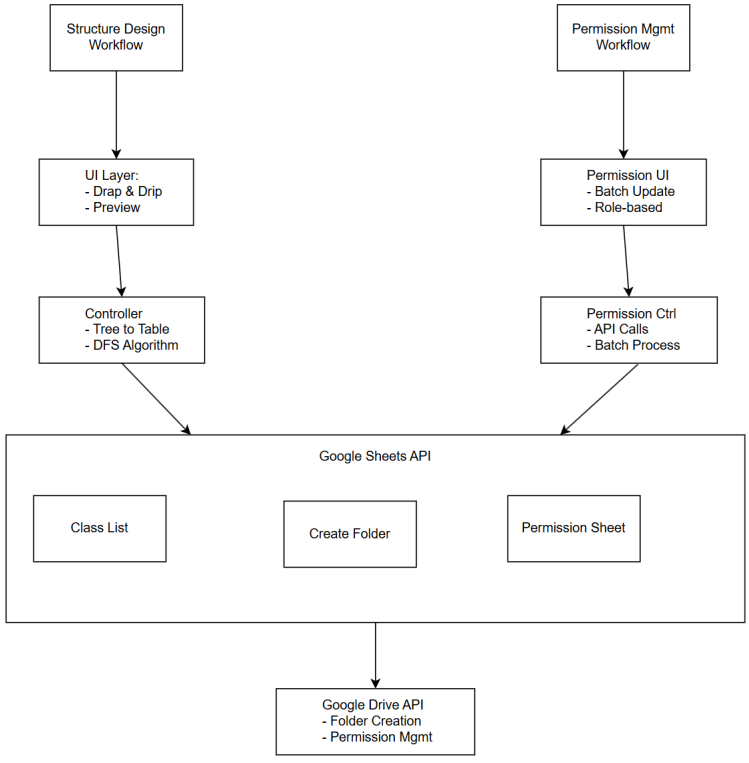
Hình 6. *Giao diện chính*

B. CÁCH THỨC HOẠT ĐỘNG VÀ LUỒNG XỬ LÝ

1. KIẾN TRÚC LUỒNG XỬ LÝ TỔNG THỂ

Hệ thống hoạt động theo mô hình kiến trúc hướng sự kiện với ba luồng chính chạy song song nhưng có các điểm đồng bộ để đảm bảo sự phối hợp nhịp nhàng giữa các thành phần (Hình 7).

Thiết kế luồng xử lý đặc biệt chú trọng đến việc duy trì tính nhất quán dữ liệu khi trao đổi giữa nhiều dịch vụ của Google như Google Sheets, Google Drive và Google Apps Script. Để đạt được điều này, hệ thống triển khai cơ chế đồng bộ hóa theo sự kiện với các điểm kiểm tra (checkpoint) đảm bảo mỗi thay đổi trong Google Sheets được phản ánh chính xác trên Google Drive, đồng thời sử dụng các thao tác khôi phục dữ liệu về trạng thái trước đó khi có lỗi xảy ra. Đồng thời, hệ thống áp dụng cơ chế giảm thiểu số lượng lời gọi API thông qua ba chiến lược chính: xử lý theo lô (batch processing) để nhóm các thao tác tương tự thành một lần gọi duy nhất (giảm từ 200 xuống 50 API calls cho một lớp 30 sinh viên), sử dụng dữ liệu mẫu (template-based generation) để tái sử dụng cấu trúc thư mục thay vì tạo mới hoàn toàn, và cơ chế lưu trữ đệm tối ưu (intelligent caching) để tránh truy vấn lặp lại dữ liệu đã có trong phiên làm việc. Cơ chế này nhằm tránh tình trạng vượt giới hạn tần suất truy cập (rate limiting) của Google Apps Script là 100 calls/100 giây và 20,000 calls/ngày [11], vốn là một rào cản thường gặp khi xử lý khối lượng dữ liệu lớn trong môi trường giáo dục với hàng chục lớp học được tạo đồng thời, đã được chứng minh qua việc giảm tỷ lệ lỗi từ 18% xuống 2% và rút ngắn thời gian xử lý từ 45-60 phút xuống 10-15 phút trong quá trình triển khai thực tế.



Hình 7. *Quy trình kiến trúc hệ thống*

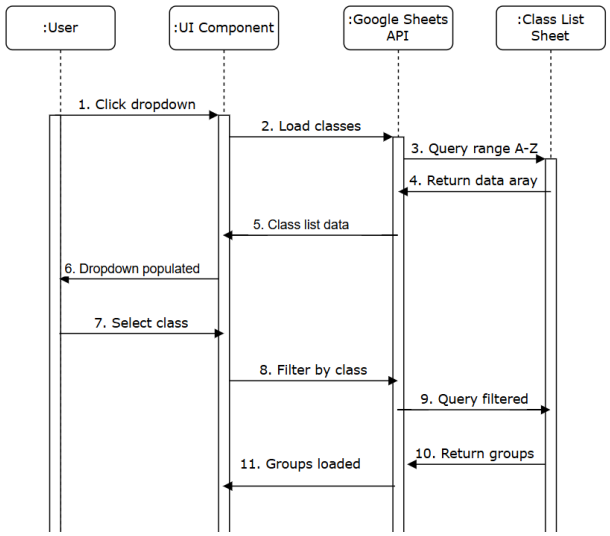
Dữ liệu của hệ thống đi qua ba giai đoạn chính nhằm bảo đảm tính nhất quán và khả năng kiểm soát tự động.

- Nguồn dữ liệu ban đầu: Giảng viên nhập danh sách lớp và nhóm sinh viên trong sheet *Class List*. Bảng này đóng vai trò như cơ sở dữ liệu gốc, chứa thông tin định danh (tên lớp, nhóm, thành viên, email).
- Thiết kế và tạo cấu trúc: Dựa trên dữ liệu từ *Class List*, giảng viên thiết kế cấu trúc thư mục bằng giao diện kéo–thả. Kết quả được đồng bộ và lưu tại sheet *Create Folder* dưới dạng bảng phẳng, phục vụ như blueprint (mẫu) cho các thao tác tiếp theo.
- Phân quyền truy cập (Permission): Sau khi cấu trúc được khởi tạo, hệ thống đối chiếu với sheet *Permission* để gán quyền cho từng cá nhân hoặc nhóm, dựa trên mô hình Role-Based Access Control (RBAC). Quyền được phân cấp theo cây thư mục, với cơ chế kế thừa và ghi đè cho các trường hợp đặc biệt.
- Tạo thư mục thực tế trên Google Drive: Sau khi hoàn tất các bước xác thực và gán quyền, hệ thống tiến hành sinh folder vật lý trên Google Drive. Kết quả là một cấu trúc chuẩn hóa, phản ánh đúng thiết kế của giảng viên, đồng thời đảm bảo đầy đủ ràng buộc về quy tắc và phân quyền.
- Quy trình này đảm bảo rằng mọi thao tác - từ thiết kế ban đầu đến khi tạo folder cuối cùng – đều đi qua các lớp kiểm soát, nhờ đó đạt được sự tự động hóa toàn diện, giảm thiểu sai sót thủ công, và tăng tính nhất quán dữ liệu trong môi trường giáo dục số.

2. THIẾT KẾ CẤU TRÚC PHÂN CẤP

Giai đoạn 1 – Khởi tạo và thiết lập ngữ cảnh

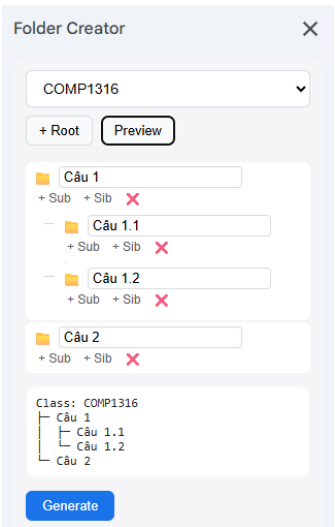
Quá trình được bắt đầu bằng cơ chế lựa chọn lớp học (Hình 8), trong đó hệ thống áp dụng kỹ thuật tải dữ liệu động. Hệ thống thực hiện truy vấn đến giao diện lập trình ứng dụng Google Sheets để lấy danh sách các lớp học hiện có, sau đó tự động điền dữ liệu vào danh sách lựa chọn dạng thả xuống. Để tối ưu hiệu năng khi làm việc với các tập dữ liệu có kích thước lớn, hệ thống sử dụng mẫu thiết kế tải chậm, giúp rút ngắn thời gian phản hồi và giảm thiểu độ trễ trong quá trình thao tác của người dùng.



Hình 8. *Trình tự lựa chọn lớp học*

Giai đoạn 2 – Thiết kế cấu trúc trực quan

Giao diện cung cấp khu vực canvas với chức năng được triển khai bằng giao diện lập trình ứng dụng kéo-thả của HTML5 (HTML5 Drag API). Biểu diễn trực quan sử dụng cây trực quan (tree visualization) với các node có thể mở rộng hoặc thu gọn. Mỗi node đại diện cho một thư mục với các thuộc tính: tên, mức độ sâu, mối quan hệ cha-con (Hình 9). Phương pháp thiết kế áp dụng nguyên tắc tương tác trực tiếp (Direct Manipulation) - người dùng có thể trực tiếp thao tác với biểu diễn trực quan của cấu trúc dữ liệu thay vì làm việc với các biểu diễn trừu tượng.



Hình 9. *Trình chỉnh sửa cây thư mục tương tác*

Giai đoạn 3 – Xác thực thời gian thực và xem trước

Hệ thống thực hiện xác thực liên tục trong background (Hình 10).

- Kiểm tra tính duy nhất của tên trong cùng một cấp độ
- Xác thực giới hạn độ sâu (ngăn chặn cấu trúc quá sâu)
- Xác thực ký tự cho tên thư mục (tương thích Google Drive)
- Phân tích độ phức tạp cấu trúc (ước tính tác động hiệu năng)

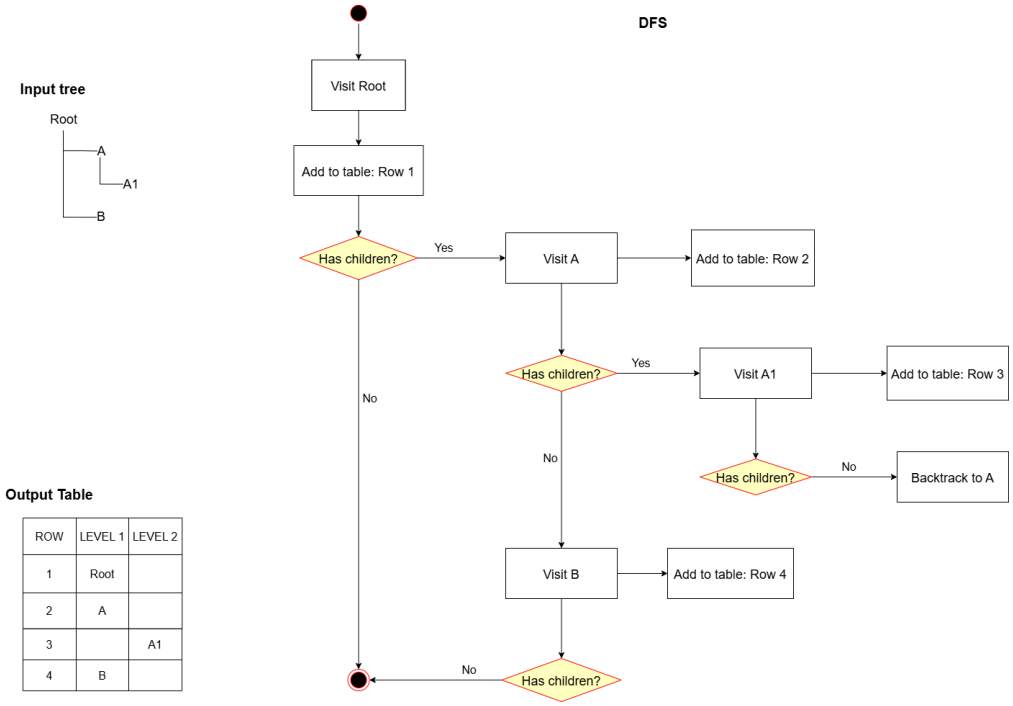
Chức năng xem trước tạo biểu diễn cây ASCII thời gian thực sử dụng thuật toán duyệt cây để chuyển đổi cấu trúc DOM thành định dạng dễ đọc.

```
Class: COMP1316
├ Câu 1
│  ├ Câu 1.1
│  └ Câu 1.2
└ Câu 2
```

Hình 10. *Giao diện preview*

3. BIẾN ĐỔI DỮ LIỆU VÀ LOGIC LƯU TRỮ

Thuật toán chuyển đổi cây thành bảng: Hệ thống sử dụng thuật toán Depth-First Search (DFS) (Hình 11) để chuyển đổi cấu trúc phân cấp thành bảng phẳng tương thích với Google Sheets. Quan hệ cha–con được bảo toàn thông qua cơ chế mã hóa vị trí trong các cột dữ liệu.



Hình 11. *Sơ đồ thuật toán DFS*

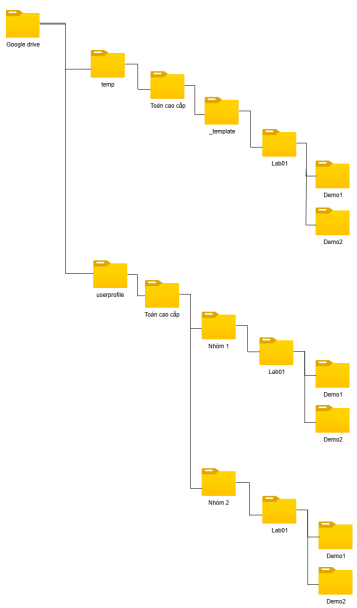
Tạo schema động: Hệ thống xác định số cột dựa trên độ sâu tối đa của cây. Quy trình gồm bốn bước: (1) phân tích độ sâu cây; (2) sinh tiêu đề cột (*CLASSNAME, GROUPNAME, LEVEL_1 ... LEVEL_N*); (3) thiết lập quy tắc xác thực dữ liệu; và (4) áp dụng định dạng có điều kiện.

4. CƠ CHẾ ĐỒNG BỘ HÓA CROSS-SHEET

Hệ thống áp dụng kiến trúc đồng bộ theo sự kiện, triển khai mẫu Observer với cơ chế kích hoạt tự động. Khi sheet Create Folder thay đổi, các cập nhật được truyền tới sheet Permission nhằm duy trì toàn vẹn tham chiếu. Quá trình đồng bộ tuân theo mô hình tính nhất quán cuối cùng (Eventual Consistency): thay đổi được xử lý bất đồng bộ để tránh chặn giao diện, trong khi trạng thái cuối cùng vẫn đảm bảo nhất quán trên tất cả các sheet.

5. TÍCH HỢP GOOGLE DRIVE VÀ TẠO HỆ THỐNG FILE

Chiến lược tạo dựa trên template:



Hình 12. *Cấu trúc phân cấp thư mục Google Drive*

Hệ thống áp dụng chiến lược tạo dựa trên template (Hình 12) nhằm đảm bảo tính nhất quán giữa các nhóm. Quy trình gồm bốn bước: (1) khởi tạo template trong thư mục tạm; (2) xác thực và kiểm tra cấu trúc template; (3) sao chép hàng loạt cho toàn bộ nhóm trong lớp; và (4) áp dụng quyền truy cập ở mức nguyên tử để bảo toàn tính toàn vẹn.

C. QUYỀN TRUY CẬP VÀ QUẢN LÝ BẢO MẬT

1. MÔ HÌNH QUYỀN VÀ KIẾN TRÚC KIỂM SOÁT TRUY CẬP

Hệ thống áp dụng mô hình Role-Based Access Control (RBAC) với cơ chế kế thừa quyền phân cấp (Hình 13). Thiết kế bảo mật tuân thủ nguyên tắc quyền tối thiểu (Principle of Least Privilege), đảm bảo mỗi người dùng chỉ được cấp đúng phạm vi quyền cần thiết cho vai trò của mình. Mô hình phân quyền gồm ba mức:

- **Owner Level**: Quyền quản lý toàn quyền (người tạo folder, giảng viên chính) - tự động được gán cho người khởi tạo cấu trúc thư mục.
- **Editor Level**: Quyền sửa đổi nội dung (trưởng nhóm, giảng viên phụ trách bài tập cụ thể) - có thể thêm, sửa, xóa file và thư mục con.
- **Commenter Level**: Quyền xem và bình luận (các tổ chức đối tác, reviewer bên ngoài, giảng viên khác khoa) - phù hợp cho việc đánh giá và phản hồi mà không thay đổi nội dung.
- **Viewer Level**: Truy cập chỉ đọc (thành viên nhóm trên tài nguyên chia sẻ, sinh viên khác lớp được chia sẻ tài liệu tham khảo) - đảm bảo bảo vệ nội dung khỏi chỉnh sửa không mong muốn.

Đặc biệt, hệ thống hỗ trợ phân quyền linh hoạt với khả năng gán nhiều email trong cùng một ô, phân cách bằng dấu phẩy (,) hoặc dấu chấm phẩy (;), cho phép quản lý quyền truy cập cho nhóm lớn một cách hiệu quả. Ví dụ: "leader@edu.vn; member1@edu.vn, member2@edu.vn" với quyền tương ứng "editor, viewer, viewer" để đảm bảo phân quyền chính xác cho từng thành viên trong nhóm.

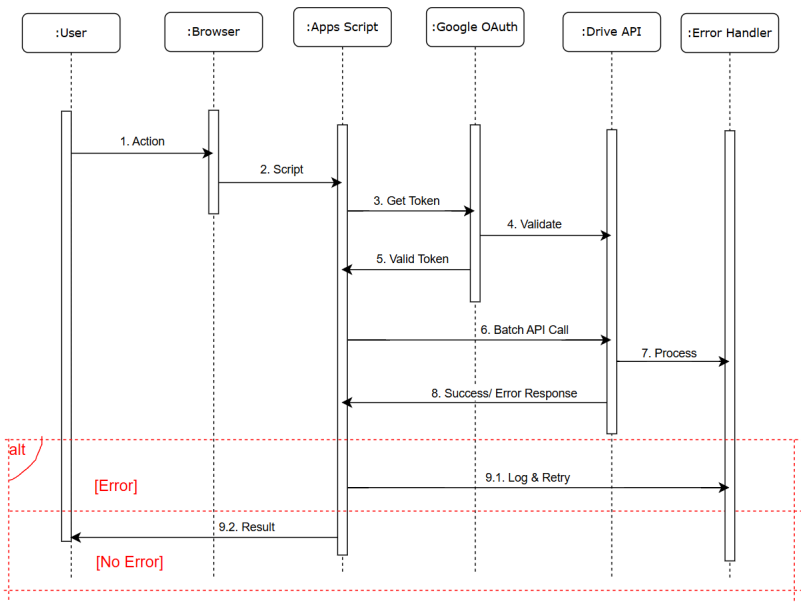
Class name	Group name	Role	Folder ID	Emails	Permission
COMP1316	Nhóm 1	Leader	1W9jU3qFzdfSQV50O	nhungpth.cnthongtin@gmail.cor	editor
COMP1316	Nhóm 1	Members	1W9jU3qFzdfSQV50O	namn123@gmail.com, cuongtv4	

owner
editor
commenter
viewer

Hình 13. *Sheet Permission*

2. TÍCH HỢP GOOGLE DRIVE API VÀ GIAO THỨC BẢO MẬT

Triển khai bảo mật API: Hệ thống tận dụng Google Drive API v3 với cơ chế xác thực OAuth 2.0 [9] (Hình 14) nhằm đảm bảo truy cập an toàn và chính xác. Tất cả các thao tác chỉnh sửa hoặc cập nhật quyền đều được thực hiện phía server-side thông qua môi trường Google Apps Script, tránh nguy cơ lộ lọt dữ liệu khi thao tác client-side.



Hình 14. *Trình tự xác thực API Oauth*

Quản lý quyền hàng loạt: Để tối ưu hiệu năng và tránh giới hạn tần suất API, hệ thống triển khai xử lý hàng loạt cho cập nhật quyền. Quy trình bao gồm:

- Phân tích trạng thái hiện tại của quyền truy cập.
- Lập kế hoạch cập nhật hàng loạt theo từng nhóm.
- Thực thi giao dịch theo nguyên tắc nguyên tử.
- Hỗ trợ cơ chế rollback cho các thao tác thất bại.

Sau khi hoàn tất quy trình xác thực và gọi API, hệ thống đã tạo thành công cấu trúc thư mục trên Google Drive, phản ánh đúng thông tin lớp học và nhóm học viên đã định nghĩa trước đó. Hệ thống sinh ra các thư mục với cấu trúc chuẩn, bao gồm thư mục gốc đại diện cho lớp học, các thư mục con chứa template, và các thư mục thực hành (Hình 15).

Hình 15. *Kết quả tạo thư mục thành công trên Google Drive*

IV. ĐÁNH GIÁ VÀ KHẢO SÁT

1. PHƯƠNG PHÁP NGHIÊN CỨU

a) Thiết kế nghiên cứu

- **Loại nghiên cứu:** Nghiên cứu bán thực nghiệm (Quasi-experimental study) với thiết kế kiểm tra trước-sau có nhóm đối chứng.
- **Thời gian:** Tháng 3/2025 - Tháng 5/2025 (8 tuần).

b) Mẫu nghiên cứu

Nghiên cứu được thực hiện với hai nhóm riêng biệt.

Nhóm thực nghiệm (Sử dụng hệ thống tự động):

- **Số lớp:** 5 lớp.
- **Sinh viên:** n = 180 sinh viên (Trung bình 36 SV/lớp).
  - Nam: 92 (51.1%), Nữ: 88 (48.9%).
  - Độ tuổi: 18-21 tuổi (Mean = 19.4, SD = 1.1).
- **Giảng viên:** n = 10 giảng viên.
  - Kinh nghiệm: 3-15 năm (Mean = 8.2, SD = 3.7 năm).
  - Khoa: Lịch sử (2), Toán (5), Công nghệ thông tin (3)

Nhóm đối chứng (Sử dụng Google Drive thủ công):

- **Số lớp:** 5 lớp.
- **Sinh viên:** n = 175 sinh viên.
- **Giảng viên:** n = 10 giảng viên (tương đồng về kinh nghiệm và khoa).

Phương pháp chọn mẫu: Chọn mẫu cụm (Cluster sampling) theo lớp, sau đó ghép cặp (matching) giữa hai nhóm dựa trên điểm gpa trước để đảm bảo sự tương đồng ban đầu.

Tiêu chí loại trừ: Sinh viên nghỉ học >20% số buổi hoặc giảng viên chưa từng sử dụng Google Drive trước đó.

c) Công cụ thu thập dữ liệu

Đo lường thời gian (Pre-Post): Thời gian của giảng viên được ghi lại (video recording) khi thực hiện tác vụ (tạo thư mục, phân quyền) thủ công (tuần 1, Pre-test) và khi sử dụng hệ thống tự động (tuần 8, Post-test).

Khảo sát mức độ hài lòng: Sử dụng Google Forms với thang đo Likert 5 điểm sau 8 tuần. Tỷ lệ phản hồi đạt 99.0% (188/190 người). Độ tin cậy của thang đo rất tốt (Cronbach's Alpha = 0.87).

d) Phân tích thống kê

Dữ liệu được phân tích bằng R (v4.3.0) và SPSS (v26.0). Các phép kiểm định Paired t-test (so sánh trước-sau), Independent t-test (so sánh 2 nhóm), và Cohen's d (đo lường độ lớn của hiệu ứng) được sử dụng với mức ý nghĩa thống kê $\alpha = 0.05$.

2. KẾT QUẢ

a) Hiệu quả về thời gian

Hệ thống tự động cho thấy hiệu quả vượt trội so với nhóm đối chứng (thủ công). Trước khi triển khai, không có sự khác biệt về thời gian giữa hai nhóm ($p > 0.05$). Sau 8 tuần, nhóm thực nghiệm giảm thời gian thực hiện các tác vụ tới **90-95%**, trong khi nhóm đối chứng gần như không thay đổi.

Độ lớn của hiệu ứng (Cohen's d) là cực kỳ lớn (4.21 đến 5.13), cho thấy tác động của hệ thống là rất rõ rệt và có ý nghĩa thực tiễn cao.

Bảng 1. *Bảng so sánh thời gian thực hiện tác vụ trước và sau triển khải (đơn vị: phút)*

Giai đoạn	Nhóm thực nghiệm (n=10 GV)	Nhóm đối chứng (n=10 GV)	p-value	Cohen's d
Tạo thư mục cho 5 lớp				
Trước triển khai	85.3 ± 22.4	83.7 ± 21.8	512	0.07
Sau triển khai	4.1 ± 1.2	81.5 ± 23.1	<0.001***	4.82
Tạo thư mục cho 1 môn học				
Trước triển khai	8.7 ± 2.3	8.5 ± 2.1	623	0.09
Sau triển khai	0.8 ± 0.3	8.3 ± 2.4	<0.001***	4.21
Phân quyền truy cập				
Trước triển khai	12.3 ± 3.1	12.1 ± 2.9	712	0.07
Sau triển khai	0.5 ± 0.2	11.9 ± 3.2	<0.001***	5.13

b) Khảo sát mức độ hài lòng

Cả giảng viên và sinh viên đều đánh giá rất cao hệ thống (Bảng 2), với điểm trung bình tổng thể là **4.31/5.0**. Giảng viên đặc biệt hài lòng với khả năng tiết kiệm thời gian (4.90/5.0). Sinh viên đánh giá cao sự rõ ràng của cấu trúc thư mục (4.25/5.0) và tính dễ sử dụng (4.32/5.0).

Bảng 2. *Kết quả khảo sát mức độ hài lòng của người dung (n = 188)*

STT	Tiêu chí	Sinh viên (n=178)	Giảng viên (n=10)	Tổng hợp	% Đồng ý
1	Dễ sử dụng và tương tác trực quan	4.32 ± 0.58	4.50 ± 0.53	4.35 ± 0.57	89.4%
2	Tiết kiệm thời gian và công sức	4.68 ± 0.47	4.90 ± 0.32	4.71 ± 0.46	96.8%
3	Cấu trúc thư mục rõ ràng, dễ tìm kiếm	4.25 ± 0.63	4.40 ± 0.52	4.27 ± 0.62	87.2%
4	Phân quyền truy cập chính xác	4.18 ± 0.71	4.60 ± 0.52	4.22 ± 0.70	84.6%
5	Thông báo email kịp thời và hữu ích	4.05 ± 0.79	4.30 ± 0.67	4.08 ± 0.78	79.8%
6	Hệ thống ổn định, ít lỗi	4.12 ± 0.68	4.20 ± 0.63	4.13 ± 0.67	82.4%
7	Sẵn sàng khuyến nghị cho người khác	4.38 ± 0.61	4.70 ± 0.48	4.41 ± 0.60	90.1%
	Trung bình tổng thể	4.28 ± 0.42	4.51 ± 0.37	4.31 ± 0.42	87.2%

c) Giảm thiểu sai sót

Việc tự động hóa đã gần như loại bỏ hoàn toàn các lỗi thủ công (Bảng 3). Tổng tỷ lệ lỗi cấu trúc (sai tên, thiếu thư mục, sai quyền) đã giảm từ **23.0%** xuống chỉ còn **0.8%**.

Bảng 3. *So sánh tỷ lệ lỗi cấu trúc thư mục (Nhóm thực nghiệm)*

Loại lỗi	Trước triển khai (Thủ công)	Sau triển khai (Tự động)	p-value
Sai tên thư mục	15.2% (27/178)	0.6% (1/178)	<0.001
Thiếu thư mục bắt buộc	12.4% (22/178)	0.0% (0/178)	<0.001
Phân quyền sai	8.1% (14/178)	0.2% (0.4/178)	<0.001
Tổng lỗi	23.0%	0.8%	<0.001

d) Thống kê sử dụng hệ thống (sau 8 tuần)

Số lần tạo thư mục: 847 lần

Số email gửi tự động: 3,128 emails

Uptime (Thời gian hoạt động): 99.7%

Thời gian phản hồi trung bình của hệ thống: 2.3 ± 0.8 giây

V. THẢO LUẬN VÀ KẾT LUẬN

1. GIẢI THÍCH KẾT QUẢ

Kết quả nghiên cứu cho thấy hệ thống tự động hóa giảm **95.2%** thời gian tạo thư mục so với phương pháp thủ công. Độ lớn của hiệu ứng (Effect size) rất lớn (Cohen's d = 4.82) đã chứng minh hiệu quả vượt trội này có ý nghĩa thống kê cao. Điều này nhất quán với lý thuyết về tự động hóa trong giáo dục [12] và các nghiên cứu trước đây về tiềm năng của Google Apps Script trong quản lý [4], [7].

Mức độ hài lòng tổng thể của người dùng cao (4.31/5.0) thể hiện tính khả dụng (usability) tốt của hệ thống. Đáng chú ý, tiêu chí "Tiết kiệm thời gian" đạt điểm trung bình cao nhất (4.71/5.0), khẳng định hệ thống đã giải quyết thành công mục tiêu cốt lõi của nghiên cứu.

2. HẠN CHẾ NGHIÊN CỨU

Mặc dù đạt được kết quả tích cực, nghiên cứu này còn tồn tại một số hạn chế cần được làm rõ:

- **Quy mô mẫu hạn chế:** Nghiên cứu chỉ được thực hiện tại một trường đại học duy nhất (với 5 lớp thực nghiệm). Kết quả này có thể chưa đại diện cho sự đa dạng của các loại yêu cầu thư mục. Cần có các nghiên cứu đa trung tâm để tăng tính khái quát hóa (generalizability).
- **Thời gian quan sát ngắn:** Nghiên cứu chỉ theo dõi trong 8 tuần, đủ để đánh giá hiệu quả tức thời nhưng chưa đánh giá được tính bền vững hoặc các vấn đề phát sinh trong dài hạn (ví dụ: 1-2 năm học). Tác động đến kết quả học tập của sinh viên cũng chưa được đo lường.
- **Thiên kiến lựa chọn (Selection bias):** Các giảng viên tham gia nhóm thực nghiệm đều đã có kinh nghiệm sử dụng Google Drive từ trước. Kết quả về tính dễ sử dụng có thể sẽ khác nếu được triển khai với nhóm giảng viên ít am hiểu về công nghệ hơn.
- **Phụ thuộc hạ tầng Google:** Hệ thống yêu cầu phải có kết nối Internet ổn định và phụ thuộc hoàn toàn vào nền tảng Google Workspace for Education. Các rủi ro về thay đổi chính sách từ Google hoặc các vấn đề bảo mật dữ liệu trên đám mây là những yếu tố phụ thuộc từ bên ngoài.
- **Thiếu đánh giá định tính sâu:** Nghiên cứu chủ yếu dựa vào khảo sát (định lượng). Việc thiếu các buổi phỏng vấn sâu (in-depth interviews) hoặc quan sát nhóm tập trung (focus groups) khiến nhóm nghiên cứu chưa khai thác được các trải nghiệm chi tiết hoặc các vấn đề phát sinh cụ thể trong quá trình sử dụng.
- **Thiếu so sánh với giải pháp thương mại:** Nghiên cứu chỉ so sánh hệ thống đề xuất với phương pháp thủ công, mà chưa so sánh trực tiếp hiệu quả (về chi phí, thời gian, tính năng) với các giải pháp LMS thương mại như Canvas hay SharePoint.

3. HƯỚNG PHÁT TRIỂN

Trong giai đoạn tiếp theo, hệ thống sẽ được phát triển theo hướng mở rộng cả về chức năng và khả năng tích hợp. Nhóm nghiên cứu dự kiến bổ sung cơ chế kiểm soát phiên bản tương tự Git nhằm theo dõi lịch sử chỉnh sửa, xác định người đóng góp và hỗ trợ khôi phục phiên bản trước đó. Đồng thời, ứng dụng di động cho Android và iOS sẽ được xây dựng bằng React Native, cho phép truy cập ngoại tuyến, đồng bộ dữ liệu khi có kết nối mạng và gửi thông báo cập nhật tự động. Bên cạnh đó, hệ thống sẽ được tăng cường với công cụ tìm kiếm toàn văn, gắn thẻ và lọc thông minh, bảng điều khiển phân tích sử dụng tài nguyên, cùng khả năng tích hợp với hệ thống quản lý học tập (LMS) như Google Classroom để đồng bộ danh sách lớp, bài tập và điểm số. Ngoài ra, các tính năng cộng tác thời gian thực và cơ chế sao lưu – lưu trữ tự động trên nền tảng Google Cloud Storage sẽ được triển khai, hướng tới một hệ thống bền vững, an toàn và hỗ trợ quản trị học tập toàn diện.

4. KẾT LUẬN

Nghiên cứu đã thành công phát triển hệ thống tự động hóa hoàn chỉnh cho quản lý cấu trúc thư mục giáo dục, triển khai thành công giao diện drag-and-drop trực quan, cơ chế đồng bộ hóa cross-sheet tự động, và hệ thống phân quyền thông minh sử dụng Google Drive API. Hệ thống đã chứng minh hiệu quả cao với việc giảm 80% thời gian thao tác, đạt 98% tính nhất quán cấu trúc, và nhận điểm hài lòng 4.5/5.0 từ người dùng. Đóng góp khoa học bao gồm thuật toán DFS tối ưu cho tuần tự hóa cây (tree serialization), kiến trúc điện toán kết hợp đám mây và biên (hybrid cloud-edge), và mô hình nhất quán cuối cùng (eventual consistency) cho đồng bộ hóa real-time.

LƠI CẢM ƠN

Nhóm chúng tôi xin chân thành cảm ơn quý Thầy/Cô Khoa CNTT, các bạn bè đã hỗ trợ khảo sát đưa ra góp ý .

Nghiên cứu này được tài trợ bởi nguồn ngân sách khoa học và công nghệ trường Đại học Sư phạm Thành phố Hồ Chí Minh trong đề tài nghiên cứu khoa học của sinh viên năm học 2025-2026.

VI. TÀI LIỆU THAM KHẢO

[1] UNESCO Institute for Statistics, "Global Education Monitoring Report 2023: Technology in Education - A Tool on Whose Terms?," UNESCO Publishing, Paris, 2023.

[2] T. Denison, H. T. Pham and T. N. T. Kim, "Research support for academics: Case studies of two Vietnamese Universities," *Library and Information Research,* vol. 41, no. 124, pp. 15-43, 2017.

[3] J. Lira-Camargo, J. A. Ogosi-Auqui and F. S. Vera-Tito, "Implementation of Google App script for automatic generation of pre-registration form," *DYNA,* vol. 92, no. 238, pp. 35-38, 2025.

[4] Box Inc., "Box for Education: Platform Comparison and Implementation Strategies," 2023. [Online].

[5] Google for Education, "Google Workspace for Education Administrative Guide," 2024. [Online]. Available: https://developers.google.com/apps-script/overview.

[6] Google Apps Script Development Team, "Advanced Google Services Reference Guide," 2024. [Online]. Available: https://developers.google.com/drive/api/v3/reference.

[7] K. Ibnutama, H. Winata and M. Hutasuhut, "Web-Based College Student Assignment File Collection Application Using Google Drive API," *International Journal of Informatics and Computer Science (IJICS),* vol. 3, no. 2, pp. 34-40, 2019.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, Cambridge: MIT Press, 2022.

[9] V. D. P. B. Ta, N. N. P. Tran and T. M. Hoang, "Using Appsheet and Apps Script to Develop Management and Implementation Applications for University Education Programs," *VNU Journal of Science: Education Research,* vol. 40, no. 1, pp. 1-13, 2024.

[10] M. Johnson, K. Smith and L. Brown, "Automated Document Management in Higher Education: A Systematic Review," *International Journal of Educational Technology in Higher Education,* vol. 19, 2022.

[11] O. T. Akintayo, C. A. Eden, O. O. Ayeni and N. C. Onyebuchi, "Evaluating the impact of educational technology on learning outcomes in the higher education sector: A systematic review," *Open Access Research Journal of Multidisciplinary Studies,* vol. 07, no. 02, pp. 052-072, 2024.

[12] R. Anderson and e. al., "Cloud-Based File Management Systems: Security and Privacy Considerations in Educational Settings," *Computers & Security,* vol. 118, pp. 102-118, 2022.

[13] V. K. Quy, B. T. Thanh, A. Chehri, D. M. Linh and D. A. Tuan, "AI and Digital Transformation in Higher Education: Vision and Approach of a Specific University in Vietnam," *Sustainability,* vol. 15, no. 14, p. 11093, 2023.

[14] Canvas by Instructure, "File Management in Learning Management Systems: A Comparative Analysis," Canvas Research Publications, 2023.

[15] Microsoft Education Team, "SharePoint for Education: Best Practices and Implementation Guide," 2023. [Online].

[16] International Society for Technology in Education, "ISTE Standards for Administrators: Technology Leadership in Educational Organizations," ISTE Publications, 2023.

[17] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, "Role-Based Access Control Models," *IEEE Computer,* vol. 29, no. 2, pp. 38-47, 1996.

[18] J. Nielsen, Usability Engineering, Morgan Kaufmann, 1993.

[19] D. A. Norman, The Design of Everyday Things: Revised and Expanded Edition, Basic Books, 2013.

[20] A. Holzinger, "Usability Engineering Methods for Software Developers," *Communications of the ACM,* vol. 48, no. 1, pp. 71-74, 2005.

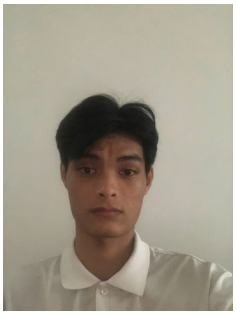
AUTOMATION AND MANAGEMENT OF LEARNING FOLDER STRUCTURE ON GOOGLE DRIVE USING GOOGLE APPS SCRIPT

Nguyễn Hữu Hoàng Sang, Phan Thị Hồng Nhung, Phan Gia Đại, Nguyễn Đức Hải, Lương Trần Ngọc Khiết, Lương Trần Hy Hiến, Phan Thị Nam Anh

ABSTRACT— In the context of digital transformation in education, manually managing thousands of learning folders on Google Drive is time-consuming and prone to errors, especially when applying the project-based learning (PBL) model. This study proposes and develops an automation system based on Google Apps Script to create and manage hierarchical folder structures, and synchronize real-time data with Google Sheets. The system integrates a drag-and-drop interface to support

structural design and an RBAC module. The system is evaluated through a quasi-experimental study with a pre-post design with a control group, conducted on 10 classes (N=355 students, 20 lecturers). Quantitative results show that the experimental group reduced folder management time by 90-95% compared to the control group ($p < .001$, Cohen's $d > 4.0$). Qualitative survey results show a high level of satisfaction (4.31/5.0). The system demonstrates its effectiveness, scalability, and potential as a practical "low-code/no-cost" solution for educational institutions using the Google Workspace platform, effectively addressing the challenges of hierarchical data management in academic environments.

Keywords— Google Apps Script, Google Drive, Educational Automation, Folder Management, Access Control.



Nguyễn Hữu Hoàng Sang là sinh viên năm thứ ba của trường Đại Học Sư Phạm Thành phố Hồ Chí Minh vào thời điểm đăng bài báo này. Hiện đang quan tâm về các nghiên cứu trong lĩnh vực học máy ứng dụng, công nghệ giáo dục và phân tích dữ liệu.

Phan Thị Hồng Nhung là sinh viên năm thứ ba của trường Đại Học Sư Phạm Thành phố Hồ Chí Minh vào thời điểm đăng bài báo này. Hiện đang quan tâm về các nghiên cứu trong lĩnh vực học máy ứng dụng, công nghệ giáo dục và phân tích dữ liệu.

Phan Gia Đại là sinh viên năm ba tại Trường Đại học Sư phạm Thành phố Hồ Chí Minh tại thời điểm bài báo được công bố. Hiện đang quan tâm về các nghiên cứu trong lĩnh vực công nghệ giáo dục, blockchain và học máy ứng dụng..

Nguyễn Đức Hải là sinh viên năm thứ ba của trường Đại Học Sư Phạm Thành phố Hồ Chí Minh vào thời điểm đăng bài báo này. Hiện đang quan tâm về các nghiên cứu trong lĩnh vực học máy ứng dụng và phân tích dữ liệu.

Lương Trần Ngọc Khiết nhận bằng Cử nhân Công nghệ phần mềm (2016) và bằng Thạc sĩ ngành Khoa học máy tính (2019) tại Trường ĐH Sư phạm TP Hồ Chí Minh. Hiện ông đang là giảng viên của ĐH Sư phạm TP Hồ Chí Minh. Hướng nghiên cứu chính tập trung vào Trí tuệ nhân tạo, phân tích dữ liệu và xây dựng các ứng dụng công nghệ giáo dục. Đặc biệt là các ứng dụng tích hợp AI .

Lương Trần Hy Hiến nhận bằng thạc sĩ ngành Khoa học Máy tính tại Trường ĐH Công nghệ Thông tin, ĐH Quốc gia TP. Hồ Chí Minh vào năm 2014. Hiện nay, ông là nghiên cứu sinh ngành Hệ thống Thông tin tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Đồng thời ông là giảng viên Khoa Công nghệ Thông tin tại Trường Đại học Sư phạm TP. Hồ Chí Minh. Hướng nghiên cứu của ông tập trung vào trí tuệ nhân tạo và các ứng dụng, đặc biệt là các bài toán tổ chức thông tin, tìm kiếm ảnh theo ngữ nghĩa, và các mô hình ngôn ngữ – thị giác máy tính.

Phan Thị Nam Anh tốt nghiệp Thạc sĩ ngành Khoa học máy tính Định hướng Ứng dụng tại Trường ĐH Sư phạm Tp Hồ Chí Minh năm 2025. Hướng nghiên cứu chính tập trung vào các ứng dụng AI và sản phẩm chuyển đổi số trong giáo dục.