TRƯỜNG ĐẠI HỌC CÔNG NGHỆ – ĐHQGHN BÁO CÁO CÁ NHÂN: KỸ NĂNG HỢP TÁC TRỰC TUYẾN TRONG DỰ ÁN NHÓM

Dự án: Hệ thống Đấu giá trực tuyến

Nhóm công cụ sử dụng: GitHub (Quản lý dự án & Lưu trữ codebase), Google Docs (Soạn thảo cộng tác), Messenger (Giao tiếp nhóm).

I. THÔNG TIN CHUNG VỀ DỰ ÁN

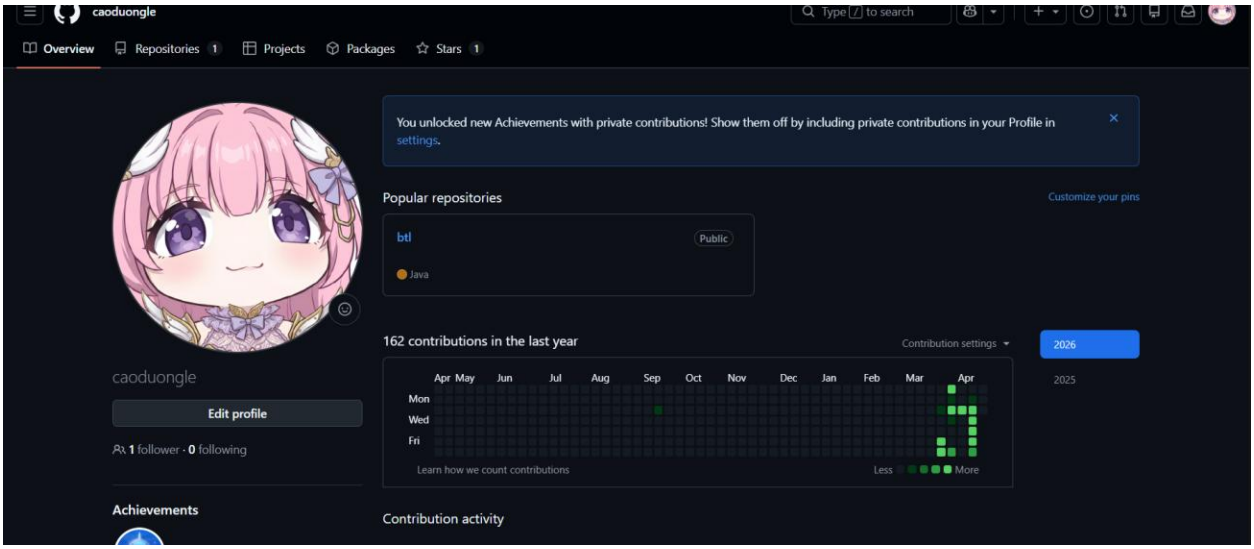
Trong khuôn khổ môn Lập trình nâng cao, tôi cùng các thành viên trong nhóm (Lộc, Kiên, Khoa, Long) đã tiến hành phát triển dự án "Hệ thống Đấu giá trực tuyến" sử dụng ngôn ngữ Java và công cụ quản lý dự án Maven.

Trong dự án này, tôi đóng vai trò là một trong những thành viên nòng cốt phụ trách phát triển logic nghiệp vụ (core logic) và quản lý phiên bản mã nguồn. Báo cáo này sẽ tập trung phân tích quá trình tôi ứng dụng các công cụ làm việc trực tuyến để điều phối công việc cá nhân, tương tác với các thành viên và quản lý tài nguyên số hiệu quả.

II. THIẾT LẬP VÀ TỐI ƯU HÓA KHÔNG GIAN LÀM VIỆC

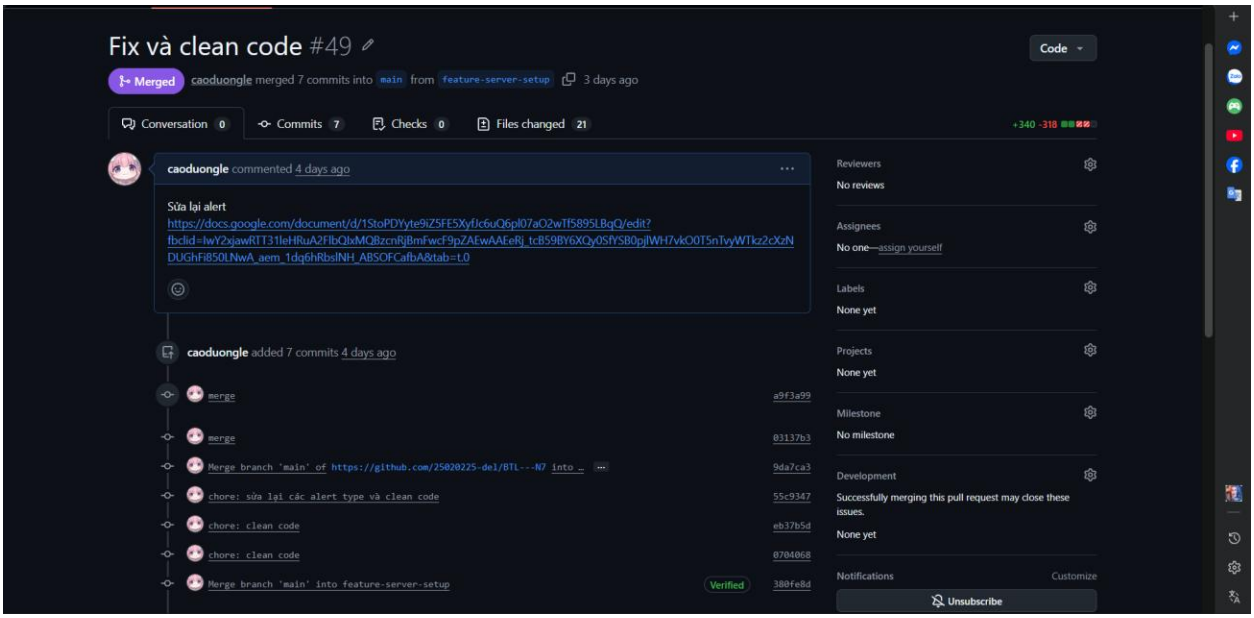
Để đảm bảo tính chuyên nghiệp và minh bạch trong làm việc nhóm, tôi đã thiết lập đồng bộ hồ sơ cá nhân trên cả 3 nền tảng: GitHub, Google Docs và Messenger.

- **Thiết lập danh tính:** Tài khoản GitHub của tôi được cập nhật đầy đủ họ tên (Cao Dương Lễ), ảnh đại diện và bio, giúp các thành viên dễ dàng nhận diện khi gắn thẻ (assign) công việc hoặc yêu cầu review code.

- 

- **Tối ưu hóa không gian làm việc (Tích hợp & Phân loại):** Thay vì sử dụng GitHub một cách đơn thuần để lưu trữ mã nguồn, tôi đã khai thác triệt để hệ thống quản lý Pull Requests (PR) và Labels trong quá trình gộp code. Cụ thể, mỗi khi tạo một PR mới để đưa code vào nhánh chính, tôi chủ động gắn các nhãn dán màu sắc như feature (tính năng mới), bug (sửa lỗi), hoặc core-logic (nghiệp vụ chính) để phân loại trực quan tính chất công việc. Hơn nữa, nhằm tối ưu hóa và kết nối xuyên suốt các không gian làm việc, trong phần mô tả của từng PR, tôi luôn đính kèm trực tiếp đường link dẫn sang tài liệu đặc tả tương ứng trên Google Docs. Việc tích hợp này giúp người chịu trách nhiệm review code có thể đối chiếu ngay lập tức logic lập trình với yêu cầu hệ thống mà không cần mất thời gian tìm kiếm thủ công qua lại giữa các nền

tảng

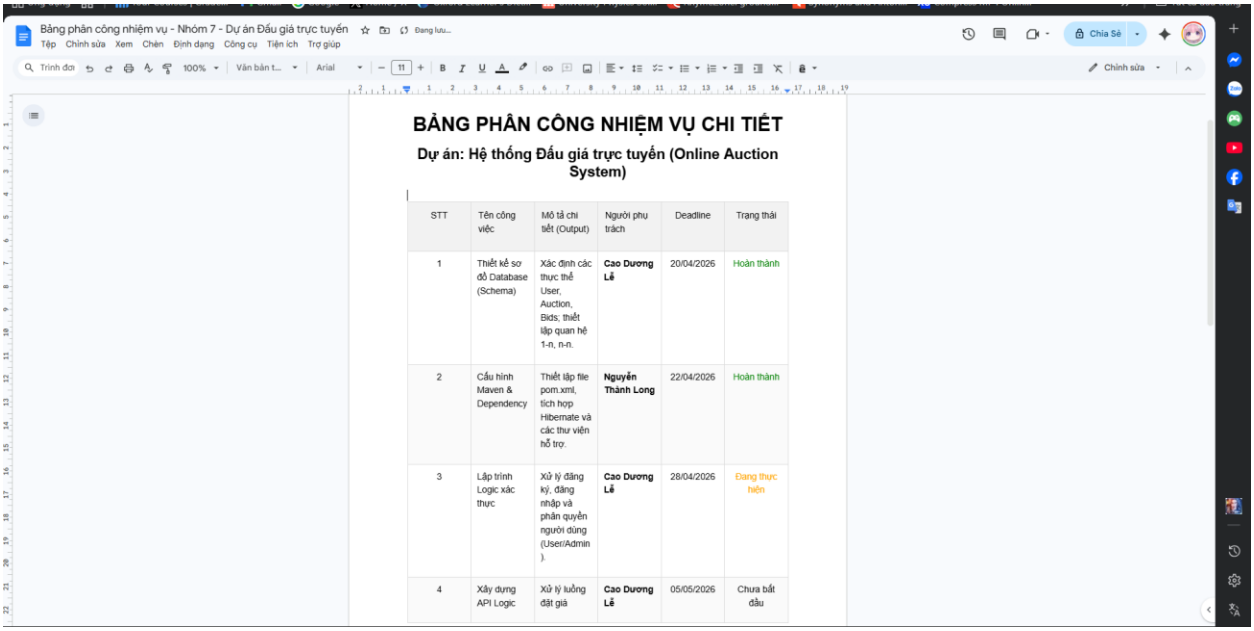


III. QUẢN LÝ TÁC VỤ VÀ TƯƠNG TÁC CÁ NHÂN

1. Tự quản lý nhiệm vụ (GitHub Projects)

Thay vì sử dụng các công cụ quản lý Kanban phức tạp, nhóm chúng tôi thống nhất quản lý tiến độ và danh sách công việc thông qua **Bảng phân công nhiệm vụ chung trên Google Docs**. 100% các đầu việc cá nhân tôi đảm nhận (ví dụ: Thiết kế schema cho bảng User/Auction, viết logic xác thực phiên đấu giá) đều được liệt kê chi tiết trong bảng này. Cụ thể:

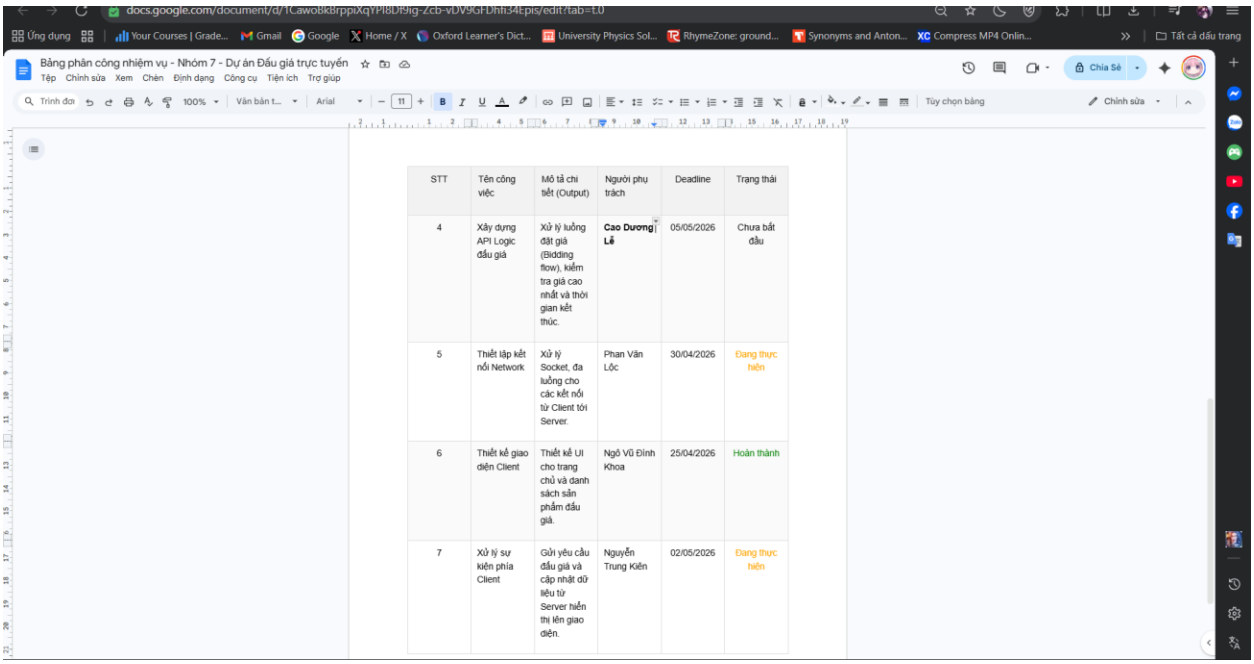
- **Người phụ trách (Assignee):** Ghi rõ tên Cao Dương Lễ ở các task liên quan đến Database và Logic.
- **Mô tả & Deadline:** Mỗi task đều đi kèm mô tả ngắn gọn về yêu cầu đầu ra vô cùng rõ ràng.
- **Cập nhật tiến độ:** Tôi thường xuyên chủ động cập nhật trạng thái công việc ngay trên bảng (chuyển trạng thái từ "Đang làm" sang "Đã hoàn thành" hoặc bôi màu nhận diện) tối thiểu 3-4 lần/tuần.

- 
Bảng phân công nhiệm vụ - Nhóm 7 - Dự án Đấu giá trực tuyến

BẢNG PHÂN CÔNG NHIỆM VỤ CHI TIẾT

Dự án: Hệ thống Đấu giá trực tuyến (Online Auction System)

STT	Tên công việc	Mô tả chi tiết (Output)	Người phụ trách	Deadline	Trạng thái
1	Thiết kế sơ đồ Database (Schema)	Xác định các thực thể User, Auction, Bids; thiết lập quan hệ 1-n, n-n.	**Cao Dương Lễ**	20/04/2026	Hoàn thành
2	Cấu hình Maven & Dependency	Thiết lập file pom.xml, tích hợp Hibernate và các thư viện hỗ trợ.	**Nguyễn Thành Long**	22/04/2026	Hoàn thành
3	Lập trình Logic xác thực	Xử lý đăng ký, đăng nhập và phân quyền người dùng (User/Admin).	**Cao Dương Lễ**	28/04/2026	Đang thực hiện
4	Xây dựng API Logic	Xử lý luồng đặt giá	**Cao Dương Lễ**	05/05/2026	Chưa bắt đầu

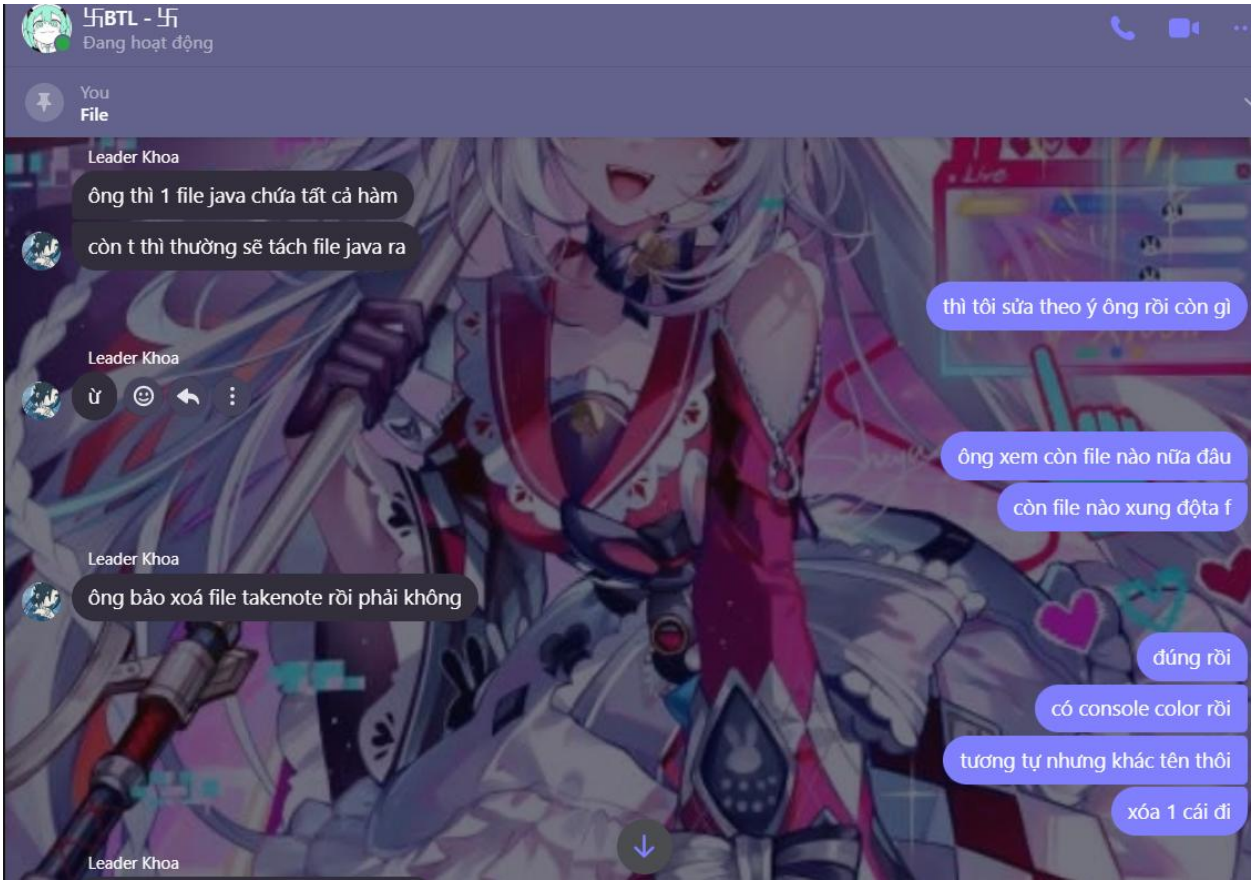
- 
Bảng phân công nhiệm vụ - Nhóm 7 - Dự án Đấu giá trực tuyến

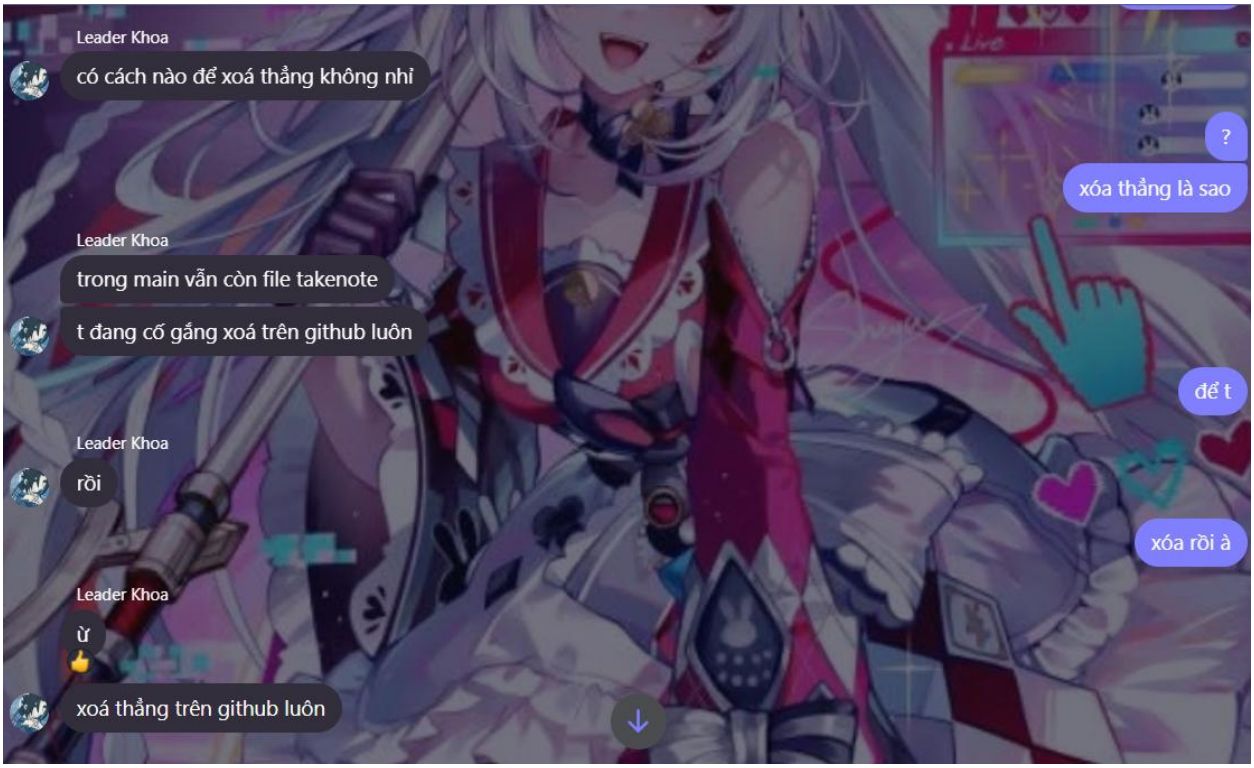
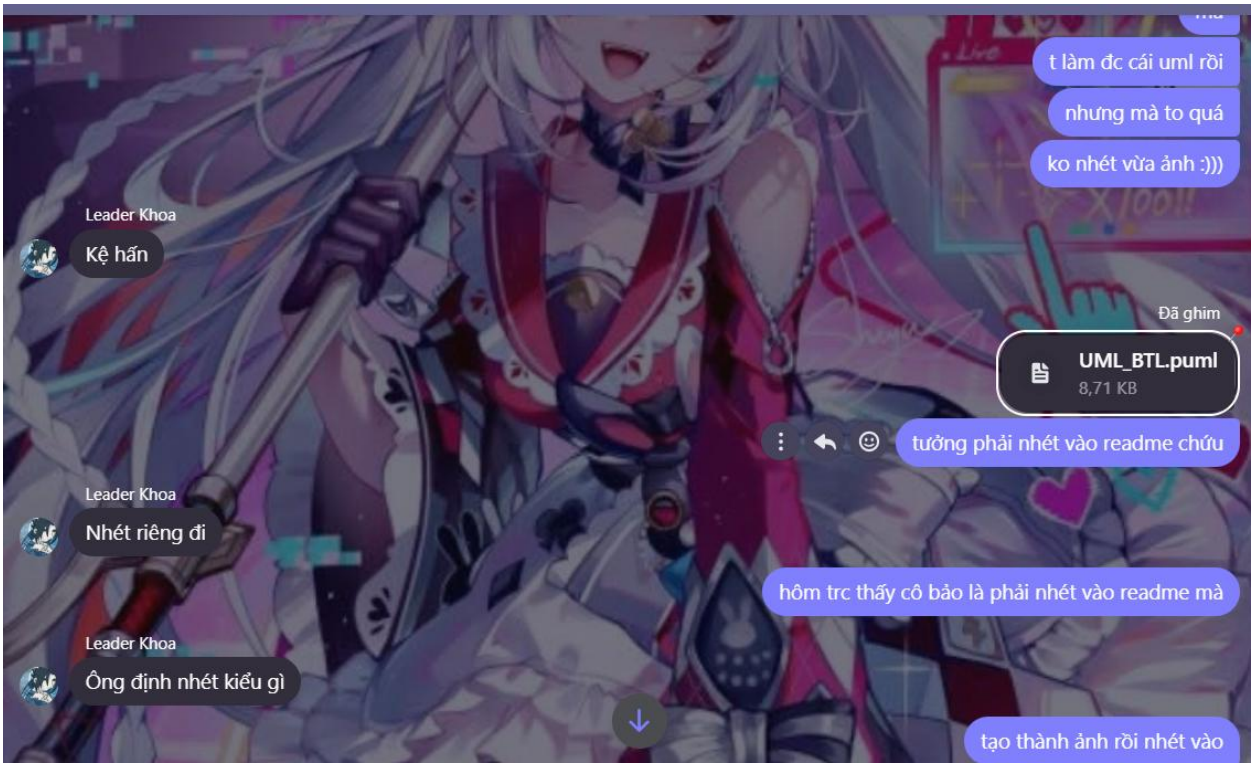
STT	Tên công việc	Mô tả chi tiết (Output)	Người phụ trách	Deadline	Trạng thái
4	Xây dựng API Logic đấu giá	Xử lý luồng đặt giá (Bidding flow), kiểm tra giá cao nhất và thời gian kết thúc.	**Cao Dương Lễ**	05/05/2026	Chưa bắt đầu
5	Thiết lập kết nối Network	Xử lý Socket, đa luồng cho các kết nối từ Client tới Server.	Phan Văn Lộc	30/04/2026	Đang thực hiện
6	Thiết kế giao diện Client	Thiết kế UI cho trang chủ và danh sách sản phẩm đấu giá.	Ngô Vũ Đình Khoa	25/04/2026	Hoàn thành
7	Xử lý sự kiện phía Client	Gửi yêu cầu đấu giá và cập nhật dữ liệu từ Server hiển thị lên giao diện.	Nguyễn Trung Kiên	02/05/2026	Đang thực hiện

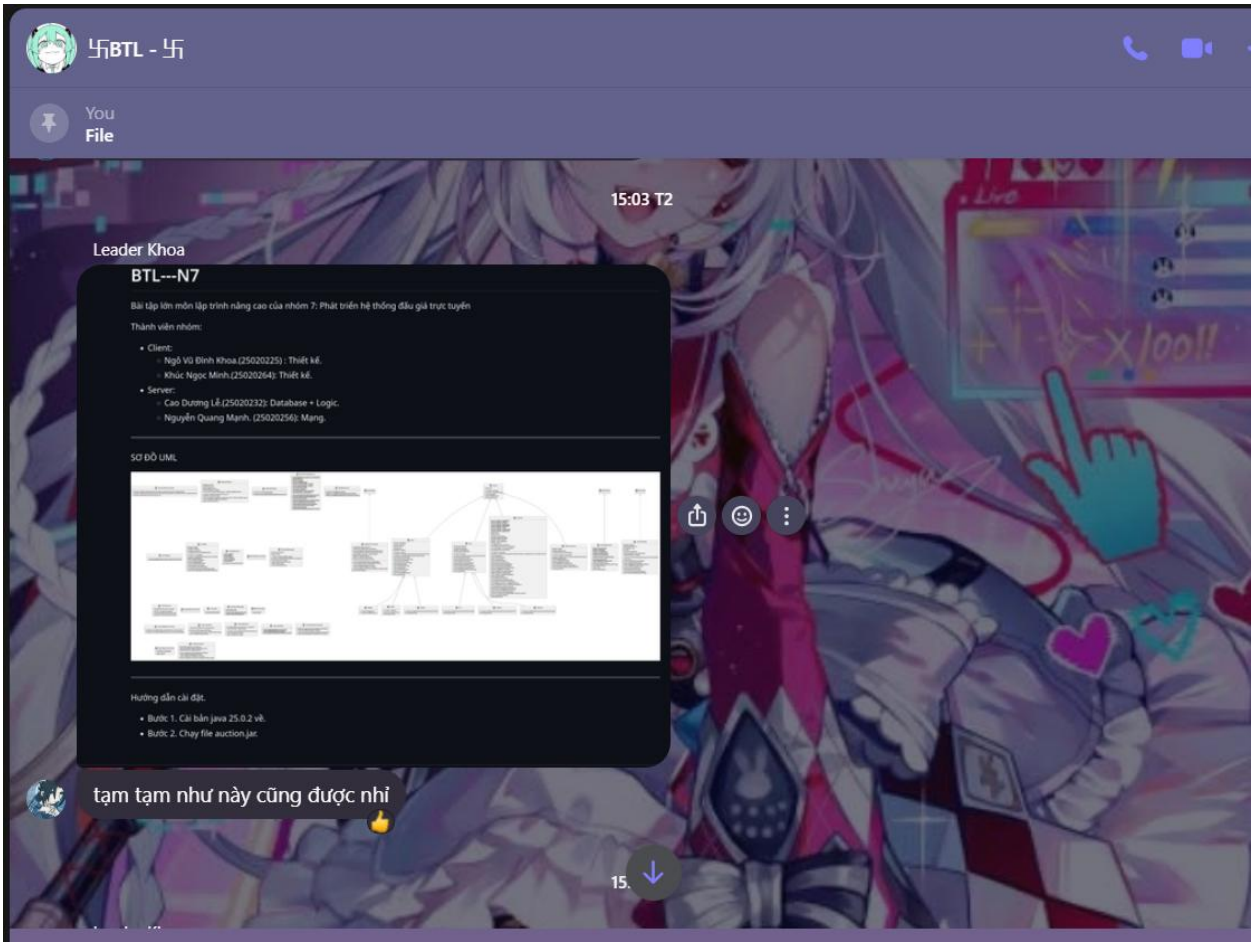
2. Tương tác và thảo luận (Messenger & GitHub)

Tôi duy trì nhịp độ giao tiếp liên tục với Lộc, Kiên, Khoa và Long trên group Messenger của nhóm. Tần suất tương tác đạt trên 10 lượt/tuần, bao gồm:

- Báo cáo tiến độ code hàng ngày (Daily stand-up ngắn).
- Chủ động hỗ trợ giải đáp khi các bạn gặp lỗi runtime hoặc lỗi cấu hình môi trường.

- Gửi tài liệu tham khảo về Design Pattern hỗ trợ cho việc thiết kế kiến trúc dự án.
- 

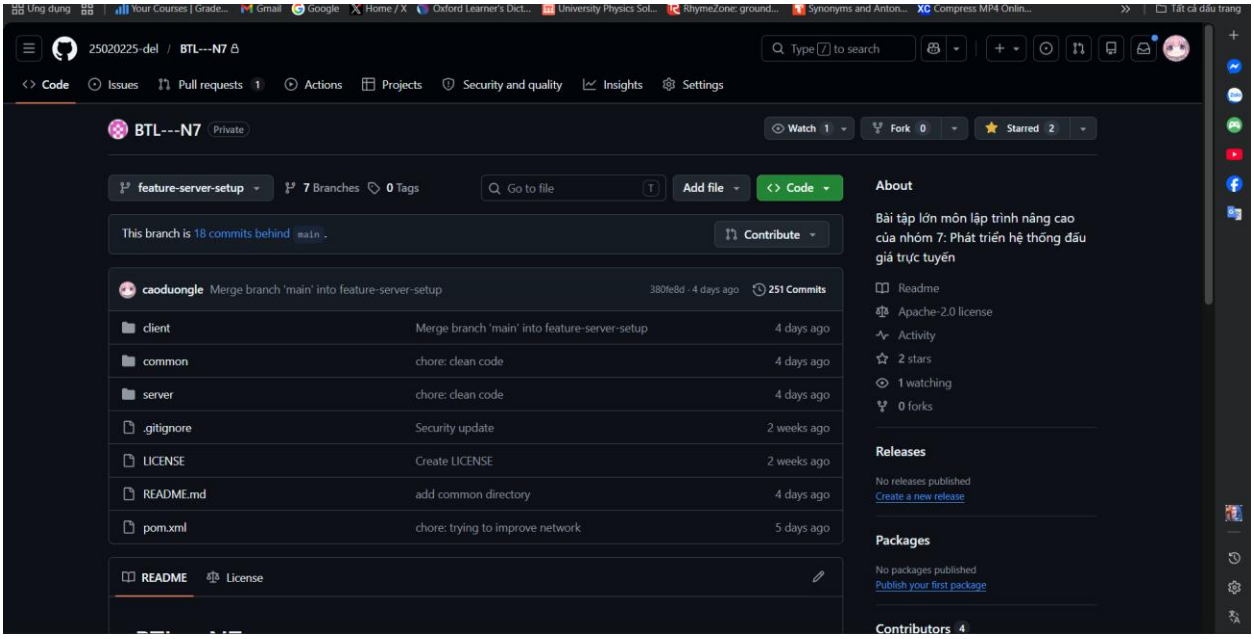
- 
Leader Khoa
có cách nào để xoá thẳng không nhỉ
?
xóa thẳng là sao
Leader Khoa
trong main vẫn còn file takenote
t đang cố gắng xoá trên github luôn
để t
Leader Khoa
rồi
xóa rồi à
Leader Khoa
ừ
xoá thẳng trên github luôn
- 
t làm đc cái uml rồi
nhưng mà to quá
ko nhét vừa ảnh :)))
Leader Khoa
Kệ hắn
Đã ghim
UML_BTL.puml
8,71 KB
tưởng phải nhét vào readme chứu
Leader Khoa
Nhét riêng đi
hôm trc thấy cô bảo là phải nhét vào readme mà
Leader Khoa
Ông định nhét kiểu gì
tạo thành ảnh rồi nhét vào

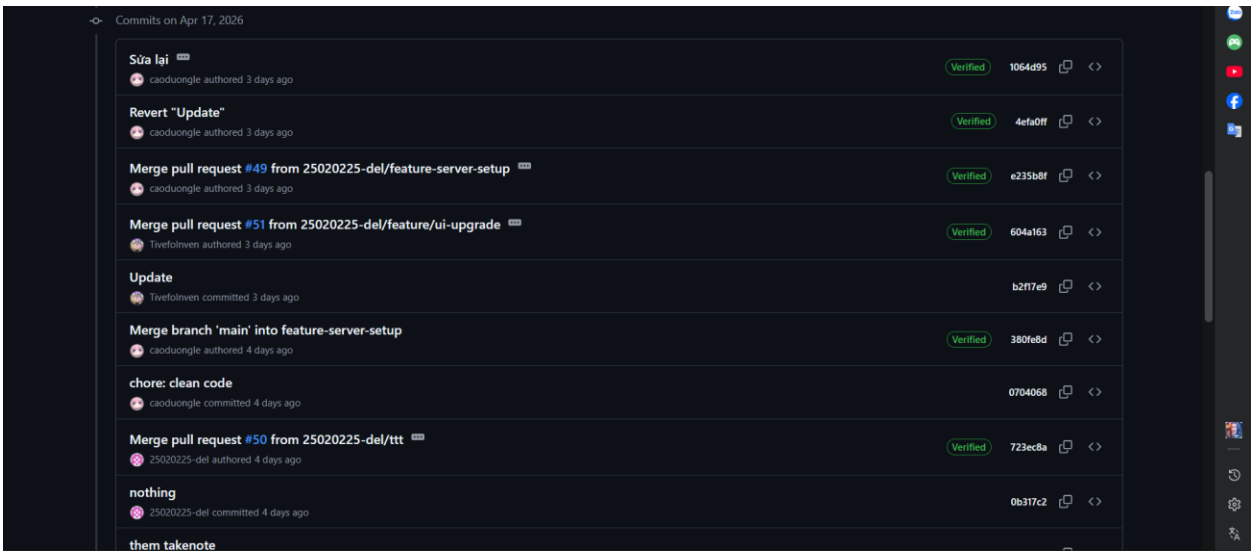
- 

IV. QUẢN LÝ TÀI NGUYÊN VÀ TỆP TIN

1. Quản lý mã nguồn và cấu trúc (GitHub)

Toàn bộ tài nguyên mã nguồn được tổ chức chặt chẽ trên GitHub Repository. Tôi tuân thủ nghiêm ngặt cấu trúc thư mục của một project Maven chuẩn (src/main/java/...). Đối với mỗi tính năng phụ trách, tôi không code trực tiếp lên nhánh main mà tạo các nhánh riêng (branch) được đặt tên theo quy tắc chuẩn: feature/tên-chức-năng (ví dụ: feature/auction-logic).



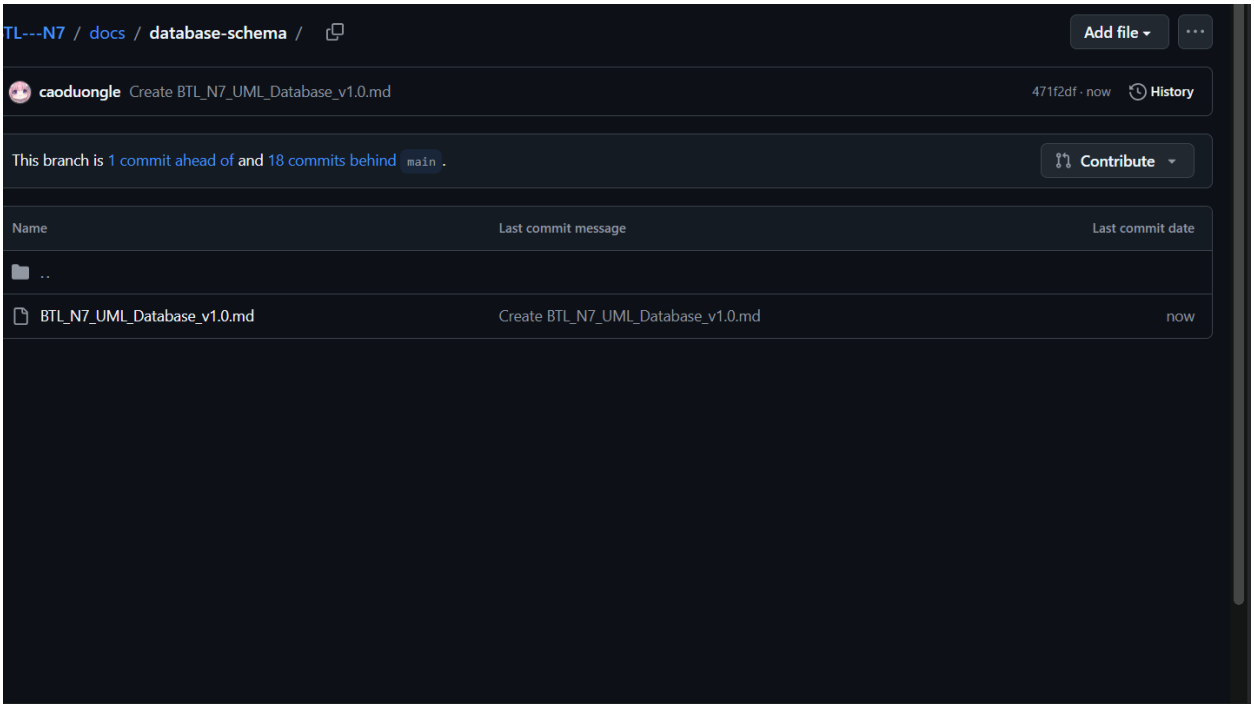


2. Tổ chức tài liệu và phân quyền (GitHub & Google Docs)

Mặc dù không sử dụng hệ thống lưu trữ đám mây truyền thống như Google Drive, tôi áp dụng tư duy quản lý tài nguyên của lập trình viên bằng cách tích hợp quản lý tài liệu trực tiếp vào Repository trên GitHub kết hợp cùng Google Docs:

- **Cấu trúc lưu trữ phân cấp (GitHub):** Các tài liệu kỹ thuật (như ảnh sơ đồ UML, file script tạo Database) được tôi lưu trữ trực tiếp vào Repository của nhóm. Tôi tạo cấu trúc thư mục 2 cấp tách biệt hoàn

toàn với mã nguồn: docs > database-schema và docs > api-specs để lưu trữ tài liệu.

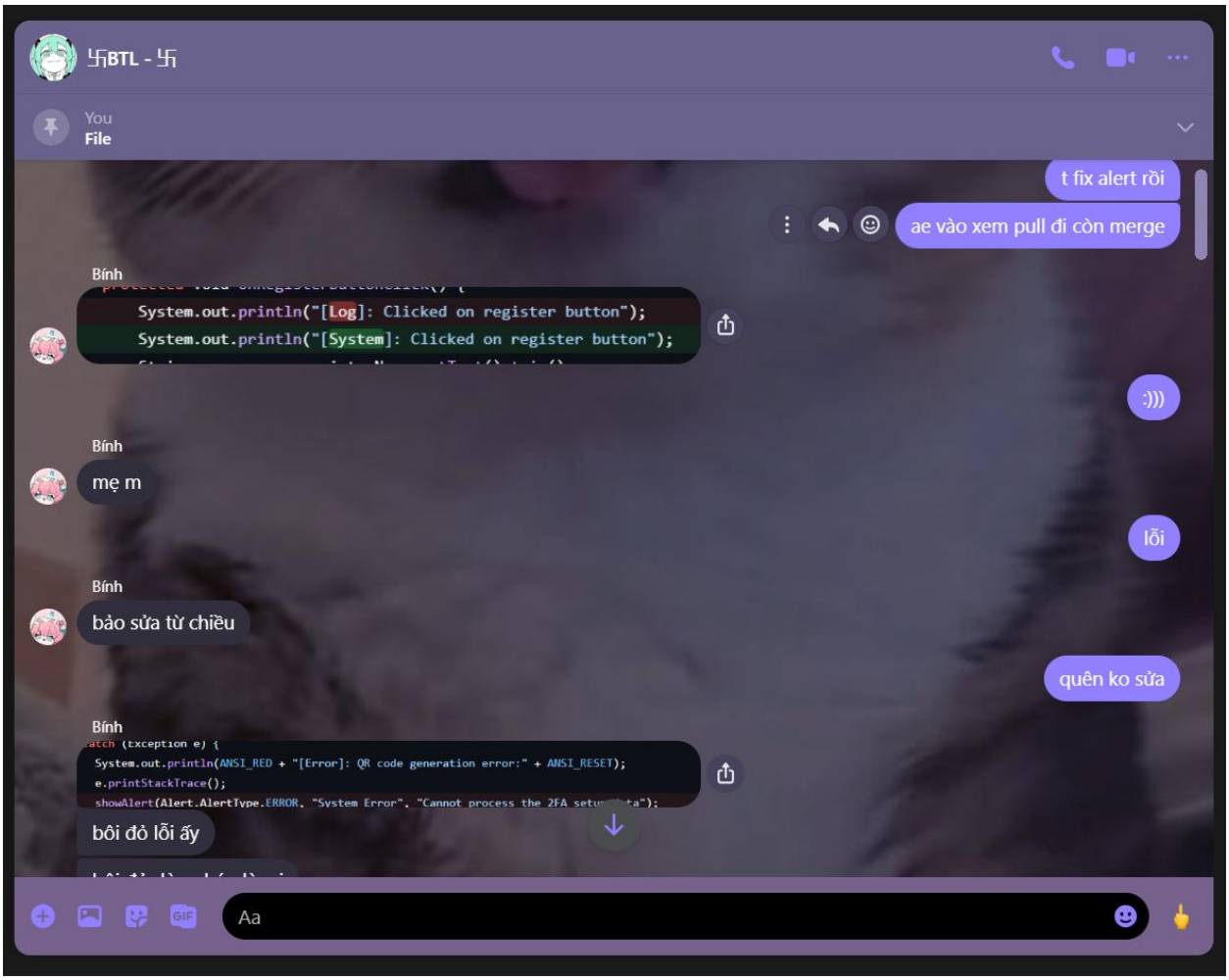
- **Quy tắc đặt tên (Naming Convention):** 100% các tệp tin tài liệu do tôi quản lý đều tuân thủ nguyên tắc: [BTL_N7]_[Tên_Tài_Liệu]_[Version].[Định_dạng]
- **Thiết lập quyền truy cập (Google Docs):** Riêng đối với các văn bản cần cả nhóm cùng soạn thảo (như báo cáo tiến độ, đặc tả hệ thống), tôi sử dụng Google Docs. Tài liệu được thiết lập chặt chẽ trong mục **Share (Chia sẻ)**: Cấp quyền "Editor" (Người chỉnh sửa) đích danh cho email của các thành viên trong nhóm, và chỉ mở quyền "Viewer" cho Giảng viên chấm bài nhằm bảo mật nội dung và tránh bị thay đổi định dạng ngoài ý muốn.
- 

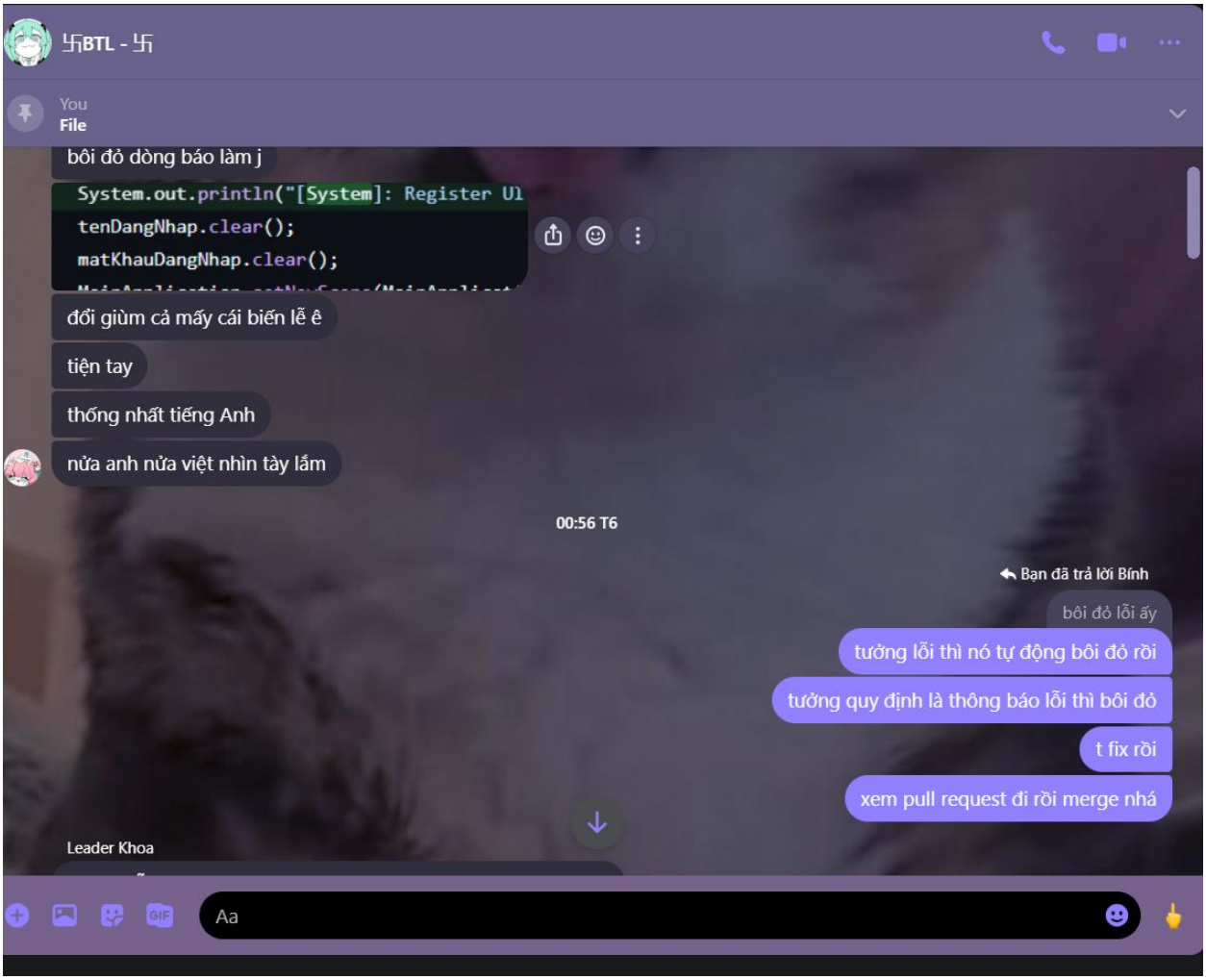
V. THÁCH THỨC VÀ GIẢI PHÁP TRONG QUÁ TRÌNH HỢP TÁC

Trong quá trình làm việc, sự kết hợp giữa các công cụ đã phát sinh một số thách thức nhất định. Dưới đây là phân tích 3 thách thức lớn nhất và cách tôi điều phối giải quyết:

Thách thức 1: Rủi ro thất thoát thông tin khi đánh giá mã nguồn (Review Code) đa nền tảng

- ***Mô tả:*** Nhóm chúng tôi thiết lập luồng làm việc (workflow) kết hợp: Tạo Pull Request (PR) trên GitHub -> Thông báo trạng thái lên Messenger -> Các thành viên đọc mã nguồn trên GitHub nhưng thảo luận và góp ý trực tiếp trên Messenger -> Người phụ trách cập nhật lại PR. Thách thức lớn nhất của quy trình phân mảnh này là tin nhắn trên Messenger trôi rất nhanh. Khi tôi quay lại IDE để sửa code, việc phải "lội ngược dòng" tin nhắn để tìm đúng các yêu cầu chỉnh sửa rải rác dễ dẫn đến việc bỏ sót lỗi hoặc hiểu sai ý đồng đội.
- ***Giải pháp:*** Để giữ vững quy trình này mà không làm mất thông tin, tôi đã đề xuất kỷ luật giao tiếp chéo: Khi một thành viên phát hiện lỗi hoặc muốn góp ý trên Messenger, bắt buộc phải gửi kèm đường link trỏ đích danh đến dòng code đó (permalink) trên GitHub PR. Ngay sau khi nhóm thảo luận và chốt xong phương án trên Messenger, tôi tiến hành sửa code, push commit mới để tự động cập nhật PR, đồng thời phải gửi một tin nhắn xác nhận đóng lại luồng thảo luận


卐BTL - 卐
You
File
t fix alert rồi
ae vào xem pull đi còn merge
Bính
System.out.println("[Log]: Clicked on register button");
System.out.println("[System]: Clicked on register button");
:)))
Bính
mẹ m
lỗi
Bính
bảo sửa từ chiều
quên ko sửa
Bính
System.out.println(ANSI_RED + "[Error]: QR code generation error:" + ANSI_RESET);
e.printStackTrace();
bôi đỏ lỗi ấy
Aa



Thách thức 2: Xung đột mã nguồn (Merge Conflict) khi gộp code

- ***Mô tả**:* Khi tôi đẩy phần logic nghiệp vụ lên GitHub trong khi thành viên khác cũng vừa cập nhật phần UI/Database, hiện tượng "đụng code" (Merge Conflict) thường xuyên xảy ra, có nguy cơ làm hỏng file chạy dự án.
- ***Giải pháp:*** Tôi thiết lập quy trình Pull Request (PR) khắt khe. Không ai được phép push code thẳng vào main. Khi tôi hoàn thành một tính năng, tôi tạo PR và tag tên một thành viên khác vào review chéo. Chỉ khi có sự đồng thuận (Approve), code mới được gộp. Nếu có Conflict, nhóm sẽ mở Google Meet để cùng nhau giải quyết trước khi chốt bản cuối.

Thách thức 3: Khó khăn trong việc chỉnh sửa chung sơ đồ UML/Tài liệu thiết kế

- ***Mô tả**:* Khi nhóm làm tài liệu phân tích hệ thống, việc nhiều người cùng vào sửa một đoạn văn bản hoặc sơ đồ trên Google Docs dẫn đến hiện tượng vô tình xóa hoặc ghi đè nội dung của nhau.
- ***Giải pháp**:* Tôi yêu cầu nhóm áp dụng tính năng "Đề xuất" (Suggesting Mode) hoặc dùng thẻ Smart Chips (gõ @) để tag tên người phụ trách. Mọi thay đổi lớn về kiến trúc Database đều phải qua dạng đề xuất, chờ người phụ trách duyệt chấp nhận thì nội dung mới chính thức được thay đổi.

VI. TỔNG KẾT VÀ BÀI HỌC KINH NGHIỆM

Việc áp dụng song song 3 nền tảng (GitHub, Google Docs, Messenger) đã giúp tôi kiểm soát tốt tiến độ cá nhân, đồng thời nâng cao hiệu suất của toàn nhóm N7 trong môn Lập trình nâng cao. Tôi nhận ra rằng, công cụ dẫu mạnh mẽ đến đâu cũng chỉ phát huy tác dụng tối đa khi có một quy trình (Workflow) rõ ràng và sự chủ động giao tiếp từ mỗi cá nhân. Trải nghiệm này không chỉ củng cố kỹ năng kỹ thuật (Git, Maven) mà còn rèn luyện cho tôi năng lực giải quyết vấn đề, điều phối giao tiếp bất đồng bộ — những kỹ năng cốt lõi của một kỹ sư phần mềm chuyên nghiệp.