ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- □ 🕮 □ -----

Ảnh có chứa văn bản, ký hiệu

Mô tả được tạo tự động

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: IT4062 - Thực hành lập trình mạng**

Giảng viên : PGS.TS Trương Thị Diệu Linh

Mã lớp: 147733

Đề tài: Xây dựng ứng dụng quản lý công việc online

Nhóm 13:

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Phạm Hữu Phúc | 20215119 |
| 2 | Nguyễn Hoàng Ninh Thuận | 20215141 |

**Hà Nội, tháng 07 năm 2024**

[Lời mở đầu 2](#_Toc185703648)

[Chương 1: Mô tả bài toán 3](#_Toc185703649)

[1.1. Yêu cầu chức năng 3](#_Toc185703650)

[1.2. Yêu cầu kỹ thuật 3](#_Toc185703651)

[Chương 2: Các chức năng chính 4](#_Toc185703652)

[2.1. Các chức năng của ứng dụng: 4](#_Toc185703653)

[2.2. Sơ đồ Usecase tổng quan: 5](#_Toc185703654)

[2.3. Sơ đồ chuyển màn: 6](#_Toc185703655)

[Chương 3: Thiết kế bản tin 6](#_Toc185703656)

[3.1. Đăng ký: 7](#_Toc185703657)

[3.2. Đăng nhập: 7](#_Toc185703658)

[3.3. Đăng xuất 7](#_Toc185703659)

[3.4. Xem danh sách dự án 8](#_Toc185703660)

[3.5. Xem chi tiết một Project 8](#_Toc185703661)

[3.6. Tạo project mới 9](#_Toc185703662)

[3.7. Mời thành viên vào project 9](#_Toc185703663)

[3.8. Xem danh sách thành viên trong project 9](#_Toc185703664)

[3.9. Xem danh sách công việc trong project 10](#_Toc185703665)

[3.10. Tạo công việc mới 10](#_Toc185703666)

[3.11. Xem chi tiết một công việc 11](#_Toc185703667)

[3.12. Xem comments (nhận xét) 11](#_Toc185703668)

[3.13. Tạo comment 12](#_Toc185703669)

[3.14. Cập nhật trạng thái công việc 12](#_Toc185703670)

[3.15. Đính kèm file vào công việc 12](#_Toc185703671)

[3.16. Tải file 13](#_Toc185703672)

[3.17. Vào phòng chat 13](#_Toc185703673)

[3.18. Lấy lịch sử tin nhắn 14](#_Toc185703674)

[3.19. Gửi tin nhắn 14](#_Toc185703675)

[3.20. Rời phòng chat 15](#_Toc185703676)

[Chương 4: Công nghệ sử dụng: 16](#_Toc185703677)

[Giao diện Client 16](#_Toc185703678)

[Proxy Server (Node.js) 16](#_Toc185703679)

[Server C 16](#_Toc185703680)

[Luồng xử lý yêu cầu 16](#_Toc185703681)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 18](#_Toc185703682)

[5.1. Kết luận 18](#_Toc185703683)

[5.2. Hướng phát triển 18](#_Toc185703684)

# 

# **Lời mở đầu**

Trong bối cảnh công nghệ ngày càng phát triển và quá trình làm việc nhóm trở nên đa dạng, nhu cầu về một công cụ hỗ trợ quản lý công việc, giao tiếp và phối hợp giữa các thành viên trong dự án ngày càng trở nên cấp thiết. Xuất phát từ mong muốn ứng dụng những kiến thức về lập trình mạng đã được học, nhóm chúng em quyết định phát triển một ứng dụng quản lý công việc online sử dụng ngôn ngữ C và mô hình client-server kết nối qua socket. Mục tiêu là tạo ra một giải pháp linh hoạt, hiệu quả và dễ dàng mở rộng trong quá trình làm việc nhóm.

Ứng dụng cho phép người dùng tạo tài khoản, xác thực và bảo mật thông tin, khởi tạo dự án mới, mời thành viên tham gia, đồng thời tạo, phân công công việc và cập nhật trạng thái một cách thuận tiện. Việc thêm nhận xét, đính kèm tài liệu và quan sát biểu đồ Gantt giúp người dùng có cái nhìn tổng quan về tiến độ và hiệu quả công việc.

Không chỉ dừng lại ở việc quản lý tác vụ, ứng dụng còn hỗ trợ giao tiếp tức thời giữa các thành viên trong dự án qua chat, thậm chí thực hiện video call. Sự tương tác này thúc đẩy kết nối nội bộ, tăng cường hiệu quả phối hợp và rút ngắn thời gian trao đổi thông tin.

Với nền tảng socket vững chắc và thiết kế theo mô hình client-server linh hoạt, ứng dụng hứa hẹn trở thành công cụ hỗ trợ đắc lực cho nhiều nhóm dự án thuộc mọi lĩnh vực. Qua quá trình xây dựng và hoàn thiện, nhóm chúng em đã củng cố vững chắc hơn các kiến thức đã học, đồng thời góp phần nâng cao chất lượng, hiệu suất làm việc trong môi trường làm việc nhóm.

Nhóm chúng em xin chân thành cảm ơn cô đã tận tình hướng dẫn, cùng với sự nỗ lực và đóng góp quan trọng từ hai thành viên, giúp dự án hoàn thiện hơn và mang lại nhiều giá trị thiết thực.

**Nhóm thực hiện**

# **Chương 1: Mô tả bài toán**

## **Yêu cầu chức năng**

Quản lý người dùng: Ứng dụng cho phép tạo tài khoản, xác thực thông tin đăng nhập và quản lý dữ liệu người dùng.

Quản lý dự án: Người dùng có thể khởi tạo dự án mới và mời thêm thành viên tham gia dự án.

Quản lý công việc: Hỗ trợ tạo công việc mới, gán cho thành viên và cập nhật trạng thái (chưa bắt đầu, đang làm, hoàn thành).

Trao đổi thông tin: Cho phép thêm nhận xét, đính kèm file vào công việc, đồng thời hỗ trợ giao tiếp qua chat hoặc thực hiện video call.

Theo dõi tiến độ: Cung cấp biểu đồ Gantt để người dùng theo dõi toàn bộ tiến độ công việc trong dự án

## **Yêu cầu kỹ thuật**

Kết nối client-server qua socket: Đảm bảo giao tiếp giữa các thành viên dự án thông qua mô hình client-server, tận dụng cơ chế socket để truyền và nhận dữ liệu.

Ghi log tin nhắn ở phía server: Mọi thông điệp gửi đi và nhận được từ client sẽ được lưu lại, giúp dễ dàng kiểm tra và quản lý.

Với những yêu cầu nêu trên, ứng dụng hướng đến việc tối ưu hóa quá trình quản lý, theo dõi và phối hợp thực hiện công việc nhóm, đồng thời mở rộng khả năng giao tiếp và trao đổi thông tin giữa các thành viên trong dự án.

# **Chương 2: Các chức năng chính**

## **2.1. Các chức năng của ứng dụng:**

Dựa trên yêu cầu chức năng và kỹ thuật của bài toán, chúng em quyết định xây dựng ứng dụng quản lý công việc, với các chức năng cung cấp cho hai loại đối tượng:

* Người dùng chưa đăng ký: Tạo tài khoản mới và đăng nhập
* Người dùng đã có tài khoản:
  + Đăng nhập: Cho phép người dùng đăng nhập.
  + Khởi tạo và quản lý dự án: Người dùng có quyền tạo mới dự án, mời thành viên, tạo công việc mới và phân công công việc cho từng thành viên, cập nhật trạng thái công việc (chưa bắt đầu, đang làm, hoàn thành).
  + Tương tác trong dự án: Hỗ trợ thêm nhận xét, đính kèm tài liệu, tải tài liệu (với công việc của mình), chat hoặc thực hiện video call với các thành viên khác trong dự án.
  + Theo dõi tiến độ qua biểu đồ Gantt: Cung cấp giao diện trực quan để nắm bắt toàn bộ tiến độ và tình trạng công việc trong dự án.

|  |  |
| --- | --- |
| **Tác nhân** | **Các chức năng chính** |
| Người dùng chưa có tài khoản | Đăng ký |
| Người dùng có tài khoản, đã đăng nhập | Đăng nhập  Tạo dự án mới  Mời thành viên tham gia dự án  Tạo công việc mới và gán cho thành viên  Cập nhật trạng thái công việc của mình  Đính kèm tài liệu vào công việc của mình  Tải tệp đính kèm từ server  Thêm nhận xét vào các công việc trong dự án  Xem biểu đồ Gantt các công việc trong dự án  Chat realtime với các thành viên trong dự án  Đăng xuất |

## **2.2. Sơ đồ Usecase tổng quan:**

A diagram of a person's structure

Description automatically generated

1. Biểu đồ usecase tổng quan

## **2.3. Sơ đồ chuyển màn:**

A diagram of a flowchart

Description automatically generated

2. Sơ đồ chuyển màn hình

# 

# **Chương 3: Thiết kế bản tin**

Trong quá trình giao tiếp giữa Client và Server, các bản tin được xây dựng theo một định dạng chung để đảm bảo tính nhất quán và thuận tiện cho việc xử lý. Cụ thể, bản tin gửi đi từ Client sẽ có cấu trúc:

Mã lệnh <Trường thông tin 1> <Trường thông tin 2> …

Mã lệnh là một chuỗi gồm 3 ký tự cố định, duy nhất cho từng chức năng (ví dụ: REG cho đăng ký, LOG cho đăng nhập, ...). Các trường thông tin tiếp theo sẽ chứa dữ liệu chi tiết liên quan đến chức năng đó (chẳng hạn email, tên người dùng, mật khẩu,...).Ở phía Server, bản tin trả về được thiết kế tương tự như các mã trạng thái trong HTTP. Một số mã thông dụng bao gồm:

* 200: Thông báo thao tác thành công (OK).
* 403: Thông báo giới hạn quyền (Forbidden).
* 404: Thông báo không tìm thấy tài nguyên (Not Found).

(Ngoài ra có thể mở rộng các mã phù hợp với yêu cầu nghiệp vụ của hệ thống.)

Cách thức thiết kế này giúp Client và Server dễ dàng phân tích và xử lý các bản tin qua socket, đồng thời đảm bảo việc mở rộng hoặc bảo trì hệ thống trong tương lai được thuận tiện hơn.

## **3.1. Đăng ký:**

Client gửi đi:  
REG <email> <username> <password>

Trong đó:

* REG là mã lệnh cho chức năng đăng ký
* <email> là địa chỉ email để đăng ký (email không được trùng nhau)
* <username> là tên người dùng trong hệ thống (username không được trùng nhau)
* <password> là mật khẩu của tài khoản

Server trả lời:

* Nếu đăng ký thành công:  
  200 <Registration successful>
* Nếu trùng email/username:  
  500 <Registration failed: Email or username already exists>

## **3.2. Đăng nhập:**

Client gửi đi:  
LOG <email> <password>

Trong đó:

* LOG là mã lệnh cho chức năng đăng nhập
* <email> là địa chỉ email của tài khoản
* <password> là mật khẩu của tài khoản

Server trả lời:

* 200 <token> <username>
  + 200 thể hiện đăng nhập thành công
  + <token> là mã phiên được cấp cho người dùng
  + <username> là tên người dùng trong hệ thống
* 401 <Unauthorized: Invalid email or password>
  + 401 thể hiện đăng nhập không thành công do email hoặc mật khẩu không hợp lệ

## **3.3. Đăng xuất**

Client gửi đi:  
OUT <token>

Trong đó:

* OUT là mã lệnh cho chức năng đăng xuất
* <token> là mã phiên của người dùng, được cấp sau khi đăng nhập

Server trả lời:

* 200 <Logout successful>
  + 200 thể hiện thao tác đăng xuất thành công
  + Logout successful là thông báo xác nhận phiên làm việc của người dùng đã được kết thúc

## **3.4. Xem danh sách dự án**

Client gửi đi:  
PRJ <token>

Trong đó:

* PRJ là mã lệnh cho chức năng lấy danh sách project
* <token> là mã phiên của người dùng, được cấp sau khi đăng nhập

Server trả lời:

* 200 <projectID1 projectName1> <projectID2 projectName2> ...
  + 200 thể hiện thao tác lấy danh sách project thành công
  + Mỗi cặp <projectID projectName> đại diện cho một project mà người dùng có quyền truy cập
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện thao tác không thành công do token không hợp lệ hoặc đã hết hạn

## **3.5. Xem chi tiết một Project**

Client gửi đi:  
PRD <projectID> <token>

Trong đó:

* PRD là mã lệnh cho chức năng xem chi tiết một project
* <projectID> là ID của project cần xem chi tiết
* <token> là mã phiên của người dùng, được cấp sau khi đăng nhập

Server trả lời:

* 200 <projectName> <createdBy> <description>
  + 200 thể hiện thao tác lấy thông tin chi tiết project thành công
  + <projectName> là tên của project
  + <createdBy> là thông tin về người tạo project
  + <description> là mô tả chi tiết về project
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện lỗi do token không hợp lệ hoặc đã hết hạn
* 403 <Not authorized or project not found>
  + 403 thể hiện người dùng không có quyền truy cập vào project này hoặc project không tồn tại

## **3.6. Tạo project mới**

Client gửi đi:  
PRO <projectName> <description> <token>

* PRO là mã lệnh cho chức năng tạo project mới
* <projectName> là tên của project
* <description> là mô tả chi tiết của project
* <token> là mã phiên của người dùng (được cấp khi đăng nhập thành công)

Server trả lời:

* 200 <projectID>
  + 200 thể hiện tạo project thành công
  + <projectID> là ID của project vừa được tạo
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ hoặc đã hết hạn, do đó không thể tạo project

## **3.7. Mời thành viên vào project**

Client gửi đi:  
INV <projectID> <email> <token>

* INV là mã lệnh cho chức năng mời một thành viên vào project
* <projectID> là ID của project mà người dùng muốn mời thành viên
* <email> là địa chỉ email của thành viên được mời
* <token> là mã phiên của người dùng (có quyền quản lý project)

Server trả lời:

* 200 <Invitation successful>
  + 200 thể hiện mời thành viên thành công
* 404 <User not found>
  + 404 thể hiện không tìm thấy người dùng có email này
* 403 <User already in project>
  + 403 thể hiện người dùng đã có trong project (không cần mời lại)

## **3.8. Xem danh sách thành viên trong project**

Client gửi đi:  
MEM <projectID> <token>

* MEM là mã lệnh để xem danh sách thành viên của project
* <projectID> là ID của project mà người dùng muốn xem thông tin thành viên
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <email\_1> <name\_1> <email\_2> <name\_2> ...
  + 200 thể hiện thao tác lấy danh sách thành viên thành công
  + <email\_i> và <name\_i> là thông tin của từng thành viên trong project
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ hoặc đã hết hạn

## **3.9. Xem danh sách công việc trong project**

Client gửi đi:  
VTL <projectID> <token>

* VTL là mã lệnh để xem danh sách task (công việc) trong một project
* <projectID> là ID của project
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <taskID\_1> <taskName\_1> <taskStatus\_1> <startDate\_1> <endDate\_1> <memberName\_1> ...
  + 200 thể hiện thao tác thành công
  + Mỗi cụm <taskID\_i> <taskName\_i> <taskStatus\_i> <startDate\_i> <endDate\_i> <memberName\_i> mô tả một công việc
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.10. Tạo công việc mới**

Client gửi đi:  
TSK <projectID> <taskName> <description> <member\_email> <startDate> <endDate> <token>

* TSK là mã lệnh cho chức năng tạo công việc mới
* <projectID> là ID của project chứa công việc mới
* <taskName> là tên công việc
* <description> là mô tả chi tiết công việc
* <member\_email> là email của thành viên được gán công việc
* <startDate> là ngày bắt đầu công việc
* <endDate> là ngày kết thúc công việc
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <taskID>
  + 200 thể hiện tạo công việc thành công
  + <taskID> là ID của công việc vừa được tạo
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.11. Xem chi tiết một công việc**

Client gửi đi:  
VOT <taskID> <token>

* VOT là mã lệnh để xem chi tiết công việc
* <taskID> là ID của công việc
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <taskName> <status> <startDate> <endDate> <memberEmail> <file1|file2|...> <description>
  + 200 thể hiện thao tác thành công
  + <taskName> tên của công việc
  + <status> trạng thái công việc (chưa bắt đầu, đang làm, đã hoàn thành)
  + <startDate> ngày bắt đầu công việc
  + <endDate> ngày kết thúc công việc
  + <memberEmail> email thành viên được gán công việc
  + <file1|file2|...> danh sách file đính kèm vào công việc
  + <description> mô tả chi tiết công việc
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ
* 404 <Task not found in project>
  + 404 thể hiện không tìm thấy công việc trong project

## **3.12. Xem comments (nhận xét)**

Client gửi đi:  
VCM <taskID> <offset> <token>

* VCM là mã lệnh để xem bình luận trong một công việc
* <taskID> là ID của công việc
* <offset> là vị trí bình luận bắt đầu lấy (phục vụ cho việc phân trang hoặc tải thêm bình luận)
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <commentID\_1> <username\_1> <content\_1> <time\_1> ... <commentID\_5> <username\_5> <content\_5> <time\_5>
  + 200 thể hiện thao tác thành công
  + Mỗi nhóm <commentID\_i> <username\_i> <content\_i> <time\_i> là một bình luận
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.13. Tạo comment**

Client gửi đi:  
CMT <taskID> <content> <token>

* CMT là mã lệnh để tạo mới một bình luận
* <taskID> là ID của công việc
* <content> là nội dung bình luận
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <Comment added successfully>
  + 200 thể hiện bình luận được thêm thành công
* 403 <You don't have permission to add comment to this task>
  + 403 thể hiện người dùng không có quyền bình luận lên công việc này
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.14. Cập nhật trạng thái công việc**

Client gửi đi:  
STT <taskID> <status> <token>

* STT là mã lệnh để cập nhật trạng thái công việc
* <taskID> là ID của công việc
* <status> là trạng thái mới của công việc (chưa bắt đầu, đang làm, hoàn thành)
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <Task status updated successfully>
  + 200 thể hiện cập nhật trạng thái thành công
* 403 <You don't have permission to update this task status>
  + 403 thể hiện người dùng không có quyền thay đổi trạng thái cho công việc này
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.15. Đính kèm file vào công việc**

Client gửi đi:  
ATH <taskID> <fileName> <token> <fileSize>

* ATH là mã lệnh để đính kèm file vào công việc
* <taskID> là ID của công việc
* <fileName> là tên file sẽ đính kèm
* <token> là mã phiên của người dùng
* <fileSize> là kích thước file (dùng để kiểm tra phân mảnh file khi gửi – nhận file)

Server trả lời:

* 200 <Ready to receive file>
  + 200 thể hiện server đã sẵn sàng nhận file từ client
* 403 <You dont have permission to attach file to this task>
  + 403 thể hiện người dùng không được quyền đính kèm file lên công việc này
* Sau khi file được gửi thành công, server trả thêm:
  + 200 <Attachment added successfully>

## **3.16. Tải file**

Client gửi đi:  
DOW <taskID> <fileName> <token>

* DOW là mã lệnh để tải file đã đính kèm trong một công việc
* <taskID> là ID của công việc
* <fileName> là tên file cần tải
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <File downloaded successfully>
  + 200 thể hiện tải file thành công
* 404 <File not found>
  + 404 thể hiện không tìm thấy file trong công việc này
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.17. Vào phòng chat**

Client gửi đi:  
JCH <projectID> <token>

* JCH là mã lệnh để tham gia phòng chat của một project
* <projectID> là ID của project
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <Joined chat room successfully>
  + 200 thể hiện tham gia phòng chat thành công
* 403 <Forbidden: Access to project denied>
  + 403 thể hiện người dùng không có quyền truy cập vào project này
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.18. Lấy lịch sử tin nhắn**

Client gửi đi:  
VCH <projectID> <token> <limit> <offset>

* VCH là mã lệnh để lấy lịch sử tin nhắn trong phòng chat
* <projectID> là ID của project
* <token> là mã phiên của người dùng
* <limit> là số lượng tin nhắn muốn lấy
* <offset> là chỉ mục bắt đầu lấy tin nhắn

Server trả lời:

* 200 <messageLength\_1>msgID\_1|userName\_1|time\_1|content\_1...<messageLength\_10>msgID\_10|userName\_10|time\_10|content\_10
  + 200 thể hiện lấy lịch sử tin nhắn thành công
  + Mỗi tin nhắn bao gồm messageLength, msgID, userName, time, content được gộp trong một chuỗi
* 403 <Forbidden: Access to project denied>
  + 403 thể hiện người dùng không có quyền truy cập project
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.19. Gửi tin nhắn**

Client gửi đi:  
MSG <projectID> <token> <content>

* MSG là mã lệnh để gửi tin nhắn lên phòng chat
* <projectID> là ID của project
* <token> là mã phiên của người dùng
* <content> là nội dung tin nhắn

Server broadcast (gửi cho tất cả thành viên trong phòng chat):  
MSG <projectID> <userName> <content>

* Server sẽ gửi tin nhắn ra phòng chat dưới định dạng trên, kèm theo userName của người gửi

Server trả lời:

* 200 <Message sent successfully>
  + 200 thể hiện gửi tin nhắn thành công
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ

## **3.20. Rời phòng chat**

Client gửi đi:  
LCH <projectID> <token>

* LCH là mã lệnh để rời khỏi phòng chat của project
* <projectID> là ID của project
* <token> là mã phiên của người dùng

Server trả lời:

* 200 <Left chat room successfully>
  + 200 thể hiện rời phòng chat thành công
* 401 <Unauthorized: Invalid token>
  + 401 thể hiện token không hợp lệ
* 403 <Forbidden: Access to project denied>
  + 403 thể hiện người dùng không có quyền truy cập hoặc đã bị chặn quyền vào project

# **Chương 4: Công nghệ sử dụng:**

Ứng dụng được xây dựng trên mô hình Client – Proxy Server – Server C, database Postgresvới giao tiếp giữa các thành phần thông qua giao thức TCP socket để đảm bảo kết nối ổn định, bảo mật và hiệu quả.

## **Giao diện Client**

* **Công nghệ:** HTML, CSS, JavaScript
* **Mục đích:** Xây dựng giao diện trực quan, dễ sử dụng, hỗ trợ các chức năng như đăng nhập, quản lý dự án, công việc, chat và tải file.
* **JavaScript:** Gửi/nhận dữ liệu từ Proxy Server qua HTTP API.

## **Proxy Server (Node.js)**

* **Công nghệ:** Node.js, Express.js
* **Mục đích:** Là cầu nối giữa Client và Server C, xử lý yêu cầu từ Client và truyền qua TCP socket đến Server C.
* **Quản lý:** Phiên làm việc, tệp đính kèm (Multer), tin nhắn thời gian thực (EventEmitter).
* **Giao tiếp:** Chuyển đổi yêu cầu HTTP từ Client thành lệnh TCP và ngược lại.

**Ví dụ:**

* Client → Proxy: POST /login → LOG<thuan@gmail.com><20042003>
* Proxy → Client: { "token": "abcdefghijklmnopqrs123456789", "username": "thuan204" }

## **Server C**

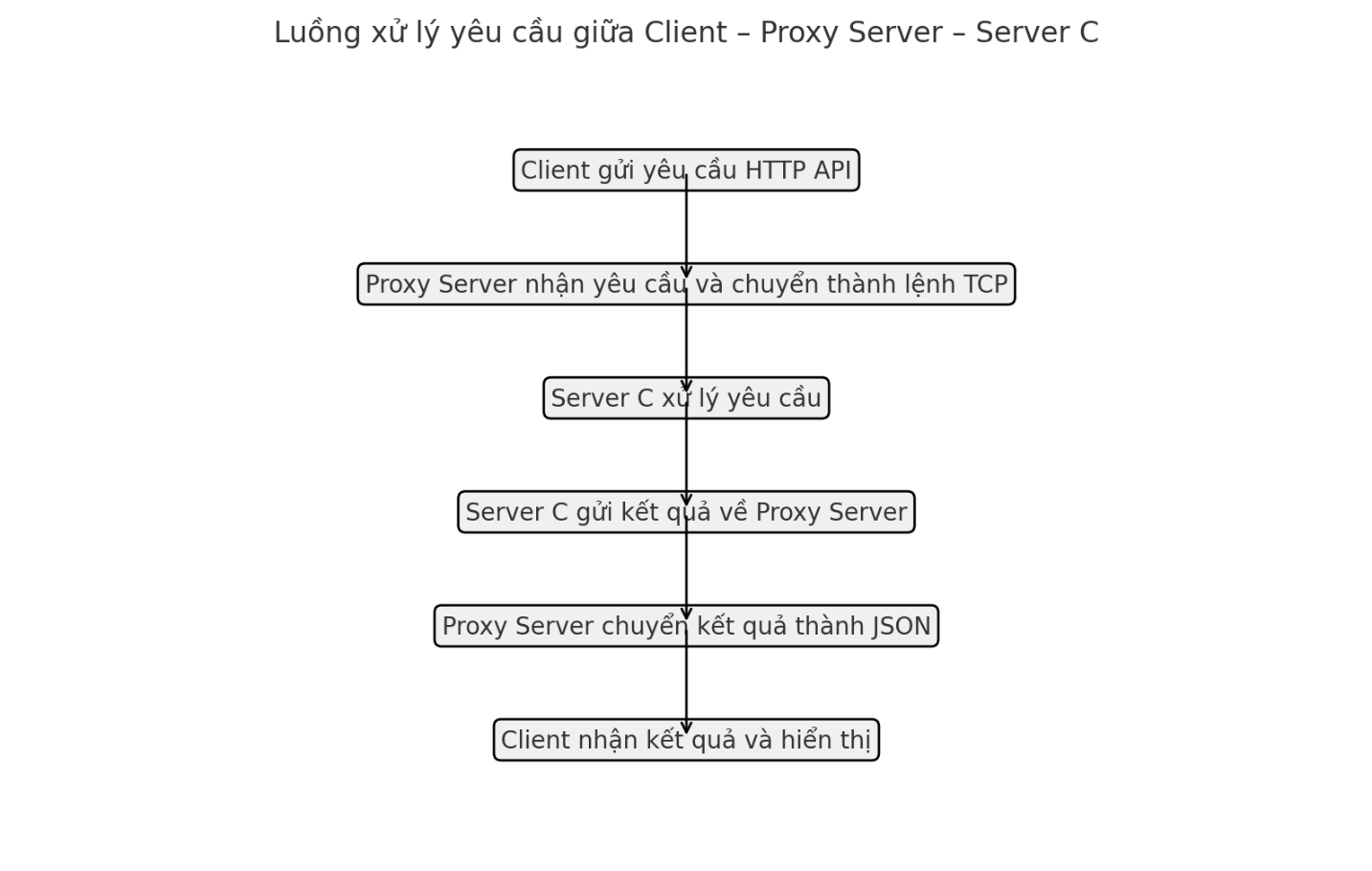
* **Công nghệ:** Ngôn ngữ C
* **Mục đích:** Xử lý logic nghiệp vụ, quản lý dự án, công việc, xác thực, chat và xử lý tệp tin.
* **Giao tiếp:** Duy trì kết nối TCP với Proxy Server để xử lý yêu cầu và trả về kết quả.
* **Xử lý file:** Hỗ trợ tải lên/tải xuống tệp tin thông qua lệnh TCP.

**Ví dụ:**

* Proxy → Server C: DOW<55><requirements.docx>< abcdefghijklmnopqrs123456789>
* Server C → Proxy: 200 <File downloaded successfully>

## **Luồng xử lý yêu cầu**

1. **Client:** Gửi yêu cầu HTTP API đến Proxy Server.
2. **Proxy Server:** Chuyển yêu cầu thành lệnh TCP gửi Server C.
3. **Server C:** Xử lý yêu cầu và trả phản hồi về Proxy.
4. **Proxy Server:** Chuyển kết quả từ Server C thành JSON và gửi về Client.
5. **Client:** Hiển thị kết quả trên giao diện.



3. Luồng xử lý dữ liệu

# **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **5.1. Kết luận**

Ứng dụng quản lý công việc đã được xây dựng thành công trên nền tảng kiến trúc **Client – Proxy Server – Server C** với giao tiếp thông qua **TCP socket**. Hệ thống cung cấp đầy đủ các tính năng cốt lõi như **đăng ký, đăng nhập, tạo và quản lý dự án, phân công công việc, chat nhóm và theo dõi tiến độ công việc**. Quá trình phát triển đã đảm bảo tính ổn định, hiệu suất cao và khả năng mở rộng trong tương lai.

Sự kết hợp giữa **HTML, CSS, JavaScript** ở phía **Client**, **Node.js** ở phía **Proxy Server**, và **C** ở phía **Server C** đã tạo nên một hệ thống linh hoạt, dễ mở rộng và đáp ứng tốt nhu cầu của người dùng.

## **5.2. Hướng phát triển**

Trong tương lai, nhóm phát triển dự kiến sẽ cải tiến và mở rộng ứng dụng với các tính năng sau:

1. **Cải thiện giao diện người dùng:**
   * Nâng cấp giao diện trực quan, hiện đại và thân thiện hơn.
   * Tối ưu hóa trải nghiệm trên các thiết bị di động và máy tính bảng.
2. **Hỗ trợ nhắn tin đa phương tiện:**
   * Cho phép người dùng gửi file, hình ảnh và video qua hệ thống nhắn tin.
   * Tăng cường khả năng chia sẻ tài liệu trực tiếp giữa các thành viên.
3. **Thông báo thời gian thực:**
   * Triển khai hệ thống thông báo (notification) thời gian thực khi có cập nhật mới, tin nhắn hoặc thay đổi trạng thái công việc.
4. **Quản trị viên (Admin):**
   * Phân quyền người dùng với vai trò Admin.
   * Quản lý tài khoản người dùng, dự án, và công việc.
   * Khả năng khóa/mở tài khoản người dùng vi phạm.
5. **Tích hợp Video Call:**
   * Hỗ trợ gọi video trực tiếp giữa các thành viên dự án.
   * Cho phép họp nhóm trực tuyến ngay trong ứng dụng.
6. **Nhắn tin riêng (Private Chat):**
   * Cho phép người dùng nhắn tin riêng với nhau thay vì chỉ chat nhóm.
   * Tăng tính bảo mật và linh hoạt trong trao đổi thông tin.
7. **Chức năng tìm kiếm:**
   * Cho phép người dùng tìm kiếm dự án, công việc, thành viên và tin nhắn một cách nhanh chóng và hiệu quả.
   * Bộ lọc nâng cao giúp cải thiện trải nghiệm tìm kiếm.