BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA: CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**TIỂU LUẬN**

*(Chữ đậm, in hoa, cỡ chữ 30)*

**Xây dựng ứng dụng truyền và nhận tập tin (Mô phỏng FTP server - FTP client)**

**MÔN: LẬP TRÌNH ỨNG DỤNG VỚI JAVA**

Ngành: **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: **NGUYỄN THÀNH SƠN**

Sinh viên thực hiện: **NGUYỄN VĂN KHÁNH**

MSSV: **22150129**

Lớp học phần:**221402**

TP. Hồ Chí Minh, tháng 11 năm 2024

**HƯỚNG DẪN THỰC HIỆN**

1. **Hình thức báo cáo tiểu luận môn học**

* Báo cáo cuối kỳ
* Thực hiện theo cá nhân hoặc nhóm

1. **Hình thức trình bày**

* Đánh máy trên word, trang A4
* Font chữ: Times New Roman, cỡ chữ 13, Unicode
* Format/ Paragraph /Line spacing (giãn cách dòng): 1.5 lines, Before: 0pt, After: 0pt, Alignment: Justified
* Đánh số trang vị trí bên phải, phía dưới

1. **Yêu cầu số trang:**

* Số trang tối thiểu: 15 trang nội dung *(không bao gồm các trang phụ như: trang bìa, trang lót bìa, trang mục lục, …)*
* Số trang tối đa: 30 trang

1. **Nội dung**

Gợi ý cấu trúc tiểu luận như sau:

* Mục lục
* Lời ngỏ *hoặc* Lời cảm ơn
* Cơ sở lý thuyết:
  + Giới thiệu tổng quan về môn học
  + Các nội dung quan trọng cần nhớ của môn học *(chọn lọc nội dung có liên quan đến đề tài)*
* Thực hiện đề tài:
  + Khảo sát hiện trạng: Mô tả hiện trạng, nhu cầu thực tế, từ đó cho thấy sự cần thiết phải xây dựng ứng dụng
  + Đặt ra yêu cầu: Tự đặt ra các yêu cầu để sau đó giải quyết các yêu cầu này bằng phần mềm
  + Sơ đồ use case: Kích thước chữ đủ lớn để có thể đọc được sau khi in ra giấy
  + Sơ đồ lớp: Bám sát mô hình MVC. Nên có 1 sơ đồ lớp tổng thể, sau đó phân rã sơ đồ từng lớp
  + Hiện thực chương trình và kiểm thử: **Không** **đưa code vào**, chỉ chụp các cửa sổ kết quả khi chạy chương trình và giới thiệu các chức năng đã làm được. Đảm bảo các hình chụp được cắt đúng phần cần chụp, kích thước chữ đủ lớn để khi in ra và đóng bìa có thể đọc được.
* Tổng kết
  + Kết quả đạt được: Đề tài đã làm được gì, đáp ứng đến đâu?
  + Đánh giá ưu khuyết điểm: Ưu điểm của đề tài là gì? Đề tài còn những thiếu sót khuyết điểm nào?
  + Hướng phát triển, mở rộng đề tài: Có thể bổ sung, làm gì thêm để đề tài tốt hơn?
* Tài liệu tham khảo

**Khoa/Viện: KHOA CNTT**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: LẬP TRÌNH ỨNG DỤNG JAVA**

1. **Họ và tên sinh viên:** Nguyễn Văn Khánh
2. **Tên đề tài**: Xây dựng ứng dụng truyền và nhận tập tin (Mô phỏng FTP server - FTP client)
3. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

1. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên sinh viên** | **Điểm số** | **Điểm chữ** |
| 1 | Nguyễn Văn Khánh |  |  |

|  |  |
| --- | --- |
|  | *TP. HCM, ngày 21 tháng 11 năm 2024*  **Giảng viên chấm**  *(Ký và ghi rõ họ tên)*  Nguyễn Thành Sơn |

MỤC LỤC

[LỜI CAM ĐOAN 7](#_Toc181660581)

[LỜI CẢM ƠN 8](#_Toc181660582)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 9](#_Toc181660583)

[1.1. Giới thiệu tổng quan 9](#_Toc181660584)

[1.2. Các nội dung quan trọng cần nhớ của môn học 10](#_Toc181660585)

[1.3. Giới thiệu tổng quan về ngôn ngữ và công cụ 10](#_Toc181660586)

[1.3.1. Giới thiệu về Java 10](#_Toc181660587)

[1.3.2. Giới thiệu về JavaFX 11](#_Toc181660588)

[1.3.3. Giới thiệu về MySQL 11](#_Toc181660589)

[1.3.4. Giới thiệu về Eclipse 11](#_Toc181660590)

[1.3.5. Giới thiệu về Scene Builder 12](#_Toc181660591)

[CHƯƠNG 2. THỰC HIỆN ĐỀ TÀI 13](#_Toc181660592)

[2.1. Khảo sát hiện trạng 13](#_Toc181660593)

[2.1.1. Hiện trạng 13](#_Toc181660594)

[2.1.2. Nhu cầu thực tế 13](#_Toc181660595)

[2.1.3. Sự cần thiết phải xây dựng ứng dụng 14](#_Toc181660596)

[2.2. Yêu cầu của phần mềm 14](#_Toc181660597)

[2.2.1. Yêu cầu về chức năng 14](#_Toc181660598)

[2.2.2. Yêu cầu về bảo mật 15](#_Toc181660599)

[2.2.3. Yêu cầu về hiệu suất 15](#_Toc181660600)

[2.3. Sơ đồ use case của chương trình 16](#_Toc181660601)

[2.4. Sơ đồ lớp của chương trình 16](#_Toc181660602)

[2.5. Cơ sở dữ liệu MySQL 16](#_Toc181660603)

[2.5.1. Bảng account 16](#_Toc181660604)

[2.5.2. Bảng settings 17](#_Toc181660605)

[2.6. Hiện thực chương trình và kiểm thử 17](#_Toc181660606)

[2.6.1. FTP Server 17](#_Toc181660607)

[2.5.2. FTP Client 18](#_Toc181660608)

[CHƯƠNG 3. TỔNG KẾT 24](#_Toc181660609)

[3.1. Kết quả đạt được 24](#_Toc181660610)

[3.2. Đánh giá về đề tài 24](#_Toc181660611)

[3.2.1. Ưu điểm của đề tài 24](#_Toc181660612)

[3.2.2. Khuyết điểm của đề tài 25](#_Toc181660613)

[3.3. Hướng phát triển và mở rộng đề tài 25](#_Toc181660614)

[TÀI LIỆU THAM KHẢO 27](#_Toc181660615)

# LỜI CAM ĐOAN

Em xin cam đoan đề tài tiểu luận: “Xây dựng ứng dụng truyền và nhận tập tin (Mô phỏng FTP server - FTP client)” do Nguyễn Văn Khánh tìm hiểu và thực hiện.

Em đã kiểm tra dữ liệu theo quy định hiện hành.

Kết quả bài làm của đề tài “Xây dựng ứng dụng truyền và nhận tập tin (Mô phỏng FTP server - FTP client)” là trung thực và không sao chép từ bất kì bài tập của các cá nhân khác.

*TP. HCM, …. tháng 11 năm 2024*

**Sinh viên cam đoan**

(Ký và ghi rõ họ tên)

# LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Thành Sơn. Trong quá trình học tập và tìm hiểu môn “Xây dựng ứng dụng truyền và nhận tập tin (Mô phỏng FTP server - FTP client)”, em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tâm huyết và tận tình của thầy. Thầy đã giúp em tích lũy thêm nhiều kiến thức về môn học này để có thể hoàn thành được bài tiểu luận về đề tài “Xây dựng ứng dụng truyền và nhận tập tin (Mô phỏng FTP server - FTP client)”.

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, em kính mong nhận được những lời góp ý của thầy để bài tiểu luận của em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## ****Giới thiệu tổng quan****

Môn học “Lập trình ứng dụng với Java” là một phần quan trọng trong chương trình đào tạo công nghệ thông tin, cung cấp cho sinh viên kiến thức từ cơ bản đến nâng cao về lập trình Java và các kỹ thuật phát triển phần mềm hiện đại. Thông qua môn học này, sinh viên không chỉ làm quen với ngôn ngữ Java mà còn hiểu rõ các nguyên tắc lập trình hướng đối tượng (OOP), giúp tổ chức và quản lý mã nguồn hiệu quả, dễ mở rộng và bảo trì. Các kiến thức nền tảng này không chỉ phục vụ cho việc phát triển phần mềm mà còn tạo tiền đề cho sinh viên ứng dụng Java vào nhiều lĩnh vực khác như xử lý dữ liệu, xây dựng hệ thống và phát triển trò chơi.

Bên cạnh lập trình hướng đối tượng, môn học còn tập trung vào các kỹ thuật lập trình ứng dụng, từ việc xây dựng giao diện người dùng thân thiện, dễ sử dụng, cho đến việc quản lý cơ sở dữ liệu và xử lý các thao tác phức tạp trên hệ thống. Các thư viện như JavaFX được giới thiệu để sinh viên có thể thiết kế giao diện trực quan và sinh động, đáp ứng nhu cầu ngày càng cao của người dùng. Ngoài ra, kỹ thuật lập trình mạng cũng là một phần quan trọng trong môn học, giúp sinh viên nắm vững cách thức kết nối và truyền tải dữ liệu giữa các ứng dụng hoặc giữa client và server, chẳng hạn như sử dụng các giao thức truyền tải dữ liệu như FTP, HTTP, hoặc TCP/IP.

Môn học không chỉ dừng lại ở các kiến thức lý thuyết mà còn có nhiều bài tập và dự án thực hành, giúp sinh viên phát triển các ứng dụng thực tiễn như ứng dụng desktop, web, và mobile. Các kỹ năng về kết nối cơ sở dữ liệu với JDBC, xử lý ngoại lệ (Exception Handling), và quản lý luồng (Multithreading) được trang bị đầy đủ, giúp sinh viên có khả năng xây dựng những ứng dụng phức tạp, hiệu suất cao và đáp ứng nhu cầu của người dùng. Nhờ đó, sinh viên không chỉ nâng cao được kỹ năng lập trình mà còn phát triển tư duy giải quyết vấn đề và khả năng làm việc với các dự án thực tế.

Tóm lại, môn học “Lập trình ứng dụng với Java” trang bị cho sinh viên một nền tảng kiến thức vững chắc và các kỹ năng cần thiết để phát triển các ứng dụng Java đa dạng, từ ứng dụng đơn giản đến các hệ thống phức tạp và yêu cầu bảo mật cao. Đây là hành trang quan trọng cho sinh viên trên con đường trở thành những lập trình viên chuyên nghiệp.

## ****Các nội dung quan trọng cần nhớ của môn học****

- **Lập trình hướng đối tượng:** Các khái niệm như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism), đóng gói (encapsulation), và trừu tượng (abstraction) là nền tảng cho việc phát triển ứng dụng Java. Chúng giúp tổ chức mã nguồn hiệu quả và mở rộng dễ dàng, rất cần thiết khi xây dựng các ứng dụng phức tạp.

- **Quản lý luồng và đa luồng**: Kiến thức về luồng và cách quản lý đa luồng giúp tối ưu hóa hiệu suất ứng dụng, cho phép xử lý đồng thời nhiều tác vụ, chẳng hạn như truyền và nhận tệp tin trong ứng dụng FTP.

- **Xử lý ngoại lệ**: Kỹ thuật này giúp ứng dụng xử lý các lỗi và sự cố không mong muốn trong quá trình thực thi, đảm bảo tính ổn định và tăng độ tin cậy.

- **Giao diện người dùng**: JavaFX là thư viện quan trọng trong Java để thiết kế giao diện người dùng. Nó hỗ trợ tạo các ứng dụng trực quan, thân thiện với người dùng, phù hợp cho các chức năng quản lý và giám sát trong ứng dụng FTP.

- **Lập trình mạng**: Các khái niệm về lập trình mạng, đặc biệt là việc sử dụng Socket và giao thức TCP/IP, đóng vai trò cốt lõi trong việc xây dựng ứng dụng truyền và nhận tệp tin, như mô phỏng FTP server và client.

- **Kết nối cơ sở dữ liệu**: JDBC là API cho phép kết nối Java với cơ sở dữ liệu, hỗ trợ lưu trữ và truy xuất dữ liệu từ bảng account và settings của ứng dụng, phục vụ cho quản lý người dùng và cấu hình server.

## ****Giới thiệu tổng quan về ngôn ngữ và công cụ****

* + 1. Giới thiệu về Java

Java là một ngôn ngữ lập trình phổ biến và đầy sức mạnh, được ra mắt vào năm 1995 bởi Sun Microsystems (nay thuộc Oracle). Với phương châm "Viết một lần, chạy mọi nơi" (Write Once, Run Anywhere), Java cho phép lập trình viên viết mã chạy trên mọi hệ điều hành có cài đặt Java Virtual Machine, từ Windows, macOS đến Linux, mang đến tính linh hoạt và đa nền tảng đáng kể.

Không chỉ đơn thuần là một ngôn ngữ lập trình, Java còn là nền tảng của nhiều lĩnh vực quan trọng: từ phát triển ứng dụng web, ứng dụng di động, đặc biệt là Android, đến các hệ thống nhúng và các ứng dụng doanh nghiệp lớn với Java EE. Java sở hữu hệ sinh thái phong phú, thư viện mã nguồn mở đa dạng và một cộng đồng lập trình viên rộng lớn luôn sẵn sàng chia sẻ kiến thức và hỗ trợ. Tất cả những điều này làm nên sức sống mạnh mẽ của Java, giúp ngôn ngữ này duy trì vị trí hàng đầu trong thế giới lập trình suốt nhiều thập kỷ.

* + 1. Giới thiệu về JavaFX

JavaFX là một bộ công cụ mạnh mẽ để xây dựng giao diện người dùng trên nền tảng Java, được Oracle giới thiệu nhằm thay thế cho công nghệ Swing truyền thống. Với khả năng tạo ra các ứng dụng đẹp mắt, trực quan và hiệu năng cao, JavaFX hỗ trợ lập trình viên thiết kế giao diện đa dạng, từ ứng dụng desktop đến các hệ thống nhúng, với đồ họa hiện đại và các hiệu ứng hoạt họa phong phú.

Điểm nổi bật của JavaFX nằm ở khả năng dễ dàng tích hợp với các công nghệ hiện đại như CSS để tùy chỉnh giao diện, FXML để xây dựng cấu trúc giao diện qua mã XML, và công cụ Scene Builder giúp tạo dựng giao diện trực quan. JavaFX cũng hỗ trợ khả năng tương tác với đồ họa 2D và 3D, cho phép lập trình viên phát triển các ứng dụng có trải nghiệm người dùng sinh động và chuyên nghiệp. Được cộng đồng đón nhận và liên tục phát triển, JavaFX trở thành lựa chọn hàng đầu cho những ai muốn xây dựng các ứng dụng Java với giao diện hiện đại, linh hoạt và dễ bảo trì.

* + 1. Giới thiệu về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến, được phát triển bởi MySQL AB và hiện nay thuộc sở hữu của Oracle. Với khả năng quản lý và xử lý dữ liệu hiệu quả, MySQL trở thành nền tảng ưa thích của nhiều ứng dụng web, đặc biệt là các trang web có lưu lượng truy cập lớn như Facebook, Twitter, và YouTube.

MySQL nổi bật nhờ tốc độ xử lý cao, khả năng mở rộng mạnh mẽ và tính bảo mật đáng tin cậy, đáp ứng tốt các yêu cầu phức tạp trong lưu trữ và truy xuất dữ liệu. MySQL hỗ trợ nhiều hệ điều hành khác nhau như Windows, macOS, và Linux, đồng thời tích hợp dễ dàng với các ngôn ngữ lập trình phổ biến như Java, PHP, và Python, giúp các lập trình viên phát triển các ứng dụng web và doanh nghiệp một cách thuận tiện.

Ngoài ra, MySQL còn hỗ trợ các tính năng mạnh mẽ như replication (sao lưu và đồng bộ dữ liệu), clustering (cụm máy chủ), và các công cụ quản lý cơ sở dữ liệu tiên tiến giúp tăng hiệu suất và độ tin cậy cho hệ thống. Chính nhờ sự linh hoạt, dễ sử dụng và chi phí thấp mà MySQL đã trở thành một trong những lựa chọn hàng đầu trong lĩnh vực quản lý dữ liệu.

* + 1. Giới thiệu về Eclipse

Eclipse là một môi trường phát triển tích hợp mã nguồn mở phổ biến, được thiết kế để hỗ trợ lập trình viên phát triển phần mềm một cách nhanh chóng và hiệu quả. Ban đầu được tạo ra cho Java, Eclipse đã phát triển thành một nền tảng đa năng, hỗ trợ nhiều ngôn ngữ lập trình khác như Python, C++, PHP, và JavaScript thông qua các plugin mở rộng. Với giao diện thân thiện và khả năng tùy biến cao, Eclipse là công cụ lý tưởng cho cả lập trình viên mới bắt đầu và các chuyên gia phát triển phần mềm.

Eclipse cung cấp nhiều tính năng mạnh mẽ như tự động hoàn thành mã, gợi ý cú pháp, debug, và tích hợp với hệ thống kiểm soát phiên bản như Git, giúp quá trình phát triển phần mềm trở nên mượt mà hơn. Hơn nữa, Eclipse còn hỗ trợ quản lý dự án, kết nối với cơ sở dữ liệu, và tích hợp với các công cụ xây dựng như Maven và Gradle, đáp ứng tốt cho cả dự án cá nhân và doanh nghiệp.

Cộng đồng rộng lớn và hệ sinh thái phong phú của Eclipse cho phép lập trình viên tận dụng hàng ngàn plugin và tiện ích bổ sung, mở rộng tính năng của IDE này. Tất cả những yếu tố này đã làm cho Eclipse trở thành một trong những IDE mạnh mẽ và phổ biến nhất cho phát triển ứng dụng Java và nhiều ngôn ngữ lập trình khác.

* + 1. Giới thiệu về Scene Builder

Scene Builder là một công cụ thiết kế giao diện đồ họa cho JavaFX, giúp lập trình viên tạo và tùy chỉnh giao diện ứng dụng một cách trực quan mà không cần viết mã XML bằng tay. Được phát triển bởi Gluon và có sẵn miễn phí, Scene Builder cho phép lập trình viên kéo và thả các thành phần giao diện như nút, nhãn, hộp văn bản và bố cục vào một "canvas" để xây dựng giao diện, tạo nên sự linh hoạt và tiết kiệm thời gian trong quá trình thiết kế.

Với Scene Builder, giao diện ứng dụng được định nghĩa bằng FXML — một định dạng XML của JavaFX, lưu trữ cấu trúc và đặc điểm của từng thành phần. Điều này giúp cho việc tách biệt giao diện và logic trở nên dễ dàng hơn, cho phép các nhà thiết kế giao diện và lập trình viên làm việc đồng thời mà không gây xung đột. Scene Builder cũng hỗ trợ tích hợp chặt chẽ với các IDE phổ biến như Eclipse, IntelliJ IDEA, và NetBeans, giúp việc kết nối giữa thiết kế giao diện và mã xử lý logic Java trở nên mượt mà.

Nhờ vào khả năng dễ sử dụng và tính linh hoạt cao, Scene Builder là một công cụ mạnh mẽ và tiện lợi trong phát triển các ứng dụng JavaFX với giao diện người dùng chuyên nghiệp.

# CHƯƠNG 2. THỰC HIỆN ĐỀ TÀI

## ****2.1. Khảo sát hiện trạng****

2.1.1. Hiện trạng

Trong bối cảnh công nghệ ngày nay, việc truyền tải dữ liệu giữa các thiết bị và hệ thống trở nên ngày càng quan trọng, đặc biệt trong các lĩnh vực như quản lý thông tin, truyền thông và thương mại điện tử. Hiện tại, có nhiều phương pháp và giao thức được sử dụng để truyền và nhận tập tin, trong đó FTP (File Transfer Protocol) vẫn là một trong những giao thức phổ biến nhất. FTP cho phép người dùng truyền tải dữ liệu qua mạng một cách hiệu quả và đáng tin cậy. Tuy nhiên, việc triển khai và sử dụng FTP không phải lúc nào cũng dễ dàng, nhất là đối với người dùng không chuyên.

**Hạn chế trong việc sử dụng FTP truyền thống**: Nhiều người dùng gặp khó khăn trong việc thiết lập và cấu hình các máy chủ FTP. Họ thường phải đối mặt với các vấn đề về bảo mật, như quản lý quyền truy cập và bảo vệ dữ liệu trong quá trình truyền tải. Các ứng dụng FTP hiện tại thường phức tạp, yêu cầu kiến thức kỹ thuật nhất định để sử dụng hiệu quả. Điều này gây khó khăn cho những người dùng không có kinh nghiệm.

**Nhu cầu về truyền và nhận tập tin**: Trong môi trường làm việc ngày nay, việc chia sẻ dữ liệu nhanh chóng và hiệu quả là cần thiết. Nhu cầu gửi và nhận các tệp tin lớn, như tài liệu, hình ảnh và video, ngày càng tăng, đặc biệt trong các dự án làm việc nhóm. Nhiều tổ chức và doanh nghiệp cần một giải pháp đơn giản để quản lý việc truyền tải dữ liệu giữa các nhân viên và hệ thống mà không gặp phải các rào cản kỹ thuật.

**Thiếu hụt các ứng dụng mô phỏng FTP**: Hiện tại, trên thị trường có nhiều ứng dụng FTP, nhưng việc phát triển các ứng dụng mô phỏng FTP server và client đơn giản, dễ sử dụng và có thể tùy chỉnh theo nhu cầu của người dùng vẫn còn hạn chế. Người dùng thường mong muốn một ứng dụng dễ sử dụng, thân thiện với giao diện người dùng, có thể chạy trên nhiều nền tảng khác nhau.

2.1.2. Nhu cầu thực tế

**Dễ sử dụng và tiếp cận**: Một ứng dụng truyền và nhận tập tin cần phải thân thiện với người dùng, cho phép bất kỳ ai, kể cả những người không có kinh nghiệm kỹ thuật, có thể dễ dàng sử dụng để chia sẻ và nhận dữ liệu.

**Tính bảo mật**: Cần có các tính năng bảo mật như mã hóa dữ liệu trong quá trình truyền tải, xác thực người dùng và quản lý quyền truy cập, nhằm bảo vệ thông tin nhạy cảm.

**Khả năng mở rộng và tùy chỉnh**: Ứng dụng nên cho phép người dùng tùy chỉnh các cài đặt và mở rộng tính năng theo nhu cầu của họ, chẳng hạn như quản lý thư mục, chia sẻ tệp tin theo nhóm, và lưu trữ lịch sử truyền tải.

**Tích hợp công nghệ mới**: Việc ứng dụng công nghệ mới như điện toán đám mây hay API sẽ giúp nâng cao hiệu quả truyền tải và tạo ra các giải pháp linh hoạt hơn cho người dùng.

2.1.3. Sự cần thiết phải xây dựng ứng dụng

Trên cơ sở khảo sát hiện trạng và nhu cầu thực tế, việc phát triển một ứng dụng mô phỏng FTP server và client đã trở nên cực kỳ cần thiết trong bối cảnh hiện nay. Ứng dụng này không chỉ giúp người dùng truyền tải dữ liệu một cách nhanh chóng và hiệu quả, mà còn cung cấp một môi trường làm việc thuận lợi, hỗ trợ việc chia sẻ thông tin dễ dàng hơn giữa các thành viên trong nhóm. Bằng cách cải thiện quy trình làm việc nhóm, ứng dụng này có thể giúp tăng cường sự hợp tác và nâng cao năng suất làm việc. Hơn nữa, trong bối cảnh công nghệ đang phát triển nhanh chóng, một ứng dụng được thiết kế tốt sẽ không chỉ đáp ứng các nhu cầu hiện tại mà còn có khả năng thích ứng với những thay đổi trong tương lai, giúp người dùng có thể tiếp cận các tính năng và công nghệ mới một cách linh hoạt. Sự linh hoạt và khả năng mở rộng này sẽ giúp ứng dụng duy trì giá trị lâu dài và tiếp tục phục vụ hiệu quả cho người dùng khi nhu cầu và yêu cầu ngày càng tăng cao.

## ****2.2. Yêu cầu của phần mềm****

2.2.1. Yêu cầu về chức năng

**Chức năng FTP Server:**

- FTP server cho phép người dùng cấu hình linh hoạt với các cài đặt như cổng kết nối và thư mục gốc lưu trữ tệp tin. Ngoài ra, người quản trị có thể thiết lập các cài đặt bổ sung như băng thông tối đa và thời gian chờ phiên làm việc để phù hợp với nhu cầu cụ thể

.- Chức năng quản lý người dùng cho phép quản trị viên tạo, chỉnh sửa và xóa tài khoản người dùng một cách dễ dàng. Hệ thống cần hỗ trợ nhập thông tin như tên đăng nhập, mật khẩucho từng người dùng.

- Ứng dụng cần cung cấp giao diện quản trị trực quan để giám sát trạng thái của server như số lượng kết nối và hiệu suất hoạt động. Giao diện này hiển thị những ai đang kết nối, đang thực hiện công việc gì, tại thư mục nào.

**Chức năng FTP Client:**

- Chức năng đầu tiên của FTP client là cho phép người dùng kết nối tới FTP server bằng cách nhập địa chỉ IP, cổng, tên đăng nhập và mật khẩu. Giao diện người dùng cần được thiết kế thân thiện, giúp người dùng dễ dàng nhập thông tin và kết nối chỉ trong vài bước đơn giản. Khi kết nối thành công, ứng dụng sẽ hiển thị danh sách các tệp tin và thư mục trên server, giúp người dùng có cái nhìn rõ ràng về nội dung mà họ có thể tương tác.

- Client hỗ trợ chức năng tải lên và tải xuống các tệp tin một cách dễ dàng, cho phép người dùng di chuyển dữ liệu giữa máy tính của họ và server mà không gặp khó khăn. Chức năng chọn và chuyển tệp tin chỉ bằng một cú nhấp chuột, giúp tiết kiệm thời gian và tăng tính hiệu quả.

- Một chức năng quan trọng khác là khả năng quản lý thư mục. Người dùng có thể tạo, xóa và thay đổi tên thư mục trên server một cách dễ dàng thông qua giao diện người dùng trực quan. Ngoài ra, ứng dụng nên hỗ trợ việc di chuyển tệp tin giữa các thư mục, giúp tối ưu hóa quy trình làm việc và quản lý dữ liệu hiệu quả hơn.

2.2.2. Yêu cầu về bảo mật

**Mã hóa dữ liệu**:

- Tất cả thông tin được truyền tải giữa server và client cần phải được mã hóa một cách an toàn để bảo vệ tính toàn vẹn và bảo mật của dữ liệu. Việc mã hóa giúp ngăn chặn sự truy cập trái phép từ những kẻ tấn công có thể cố gắng đánh cắp dữ liệu trong quá trình truyền tải. Để thực hiện điều này, ứng dụng nên sử dụng các thuật toán mã hóa mạnh mẽ

**Xác thực người dùng**:

- Hệ thống này cần đảm bảo rằng chỉ những người dùng đã được xác thực mới có thể truy cập vào các tài nguyên và chức năng của ứng dụng. Để đạt được điều này, ứng dụng cần có quy trình xác thực nghiêm ngặt, bao gồm việc yêu cầu người dùng nhập tên đăng nhập và mật khẩu. Mật khẩu nên được lưu trữ an toàn thông qua việc sử dụng các phương pháp băm, giúp bảo vệ tài khoản người dùng ngay cả khi cơ sở dữ liệu bị xâm nhập.

2.2.3. Yêu cầu về hiệu suất

**Tốc độ truyền tải**:

- Tốc độ truyền tải là một trong những yếu tố quan trọng nhất trong việc đánh giá hiệu quả của ứng dụng truyền và nhận tập tin. Ứng dụng cần được thiết kế để đảm bảo tốc độ truyền tải tệp tin nhanh chóng và ổn định, giúp người dùng có thể gửi và nhận dữ liệu một cách dễ dàng và thuận tiện.

- Ứng dụng cũng cần hỗ trợ khả năng truyền tải nhiều tệp tin đồng thời mà không làm giảm hiệu suất. Điều này có thể đạt được thông qua việc sử dụng kỹ thuật đa luồng, cho phép nhiều phiên truyền tải được thực hiện song song.

**Khả năng mở rộng**:

- Ứng dụng cần được thiết kế với kiến trúc có khả năng mở rộng, cho phép nó phục vụ nhiều người dùng cùng một lúc mà không bị giảm hiệu suất. Để đạt được điều này, ứng dụng nên áp dụng các kiến trúc phân tán, cho phép người dùng kết nối tới nhiều máy chủ khác nhau và tự động phân phối tải, từ đó giảm thiểu tình trạng quá tải trên một máy chủ duy nhất.

- Ứng dụng nên có các cơ chế để tối ưu hóa việc xử lý các yêu cầu từ người dùng, bao gồm việc quản lý kết nối và tài nguyên một cách hiệu quả, nhằm đảm bảo rằng ngay cả khi có nhiều người dùng truy cập cùng một lúc, hiệu suất vẫn được duy trì ở mức cao.

## ****2.3. Sơ đồ use case của chương trình****

Hình 2.3.1. Sơ đồ use case FTP Client

Hình 2.3.2. Sơ đồ use case FTP Client

## ****2.4. Sơ đồ lớp của chương trình****

Hình 2.4.1. Sơ đồ lớp FTP Client

Hình 2.4.2. Sơ đồ lớp FTP Server

## ****2.5. Cơ sở dữ liệu MySQL****

2.5.1. Bảng account

Bảng account lưu trữ thông tin về các tài khoản được phép truy cập vào server, đóng vai trò quan trọng trong việc quản lý và xác thực người dùng

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| username | varchar(255) | Tên người dùng, là khóa chính |
| password | varchar(255) | Mật khẩu của tên người dùng hiện tại |
| created\_at | datetime | Ngày và giờ tài khoản được tạo |

2.5.2. Bảng settings

Bảng settings lưu trữ thông tin cấu hình của server, với chỉ một dòng duy nhất để lưu các cài đặt cơ bản như đường dẫn thư mục gốc và cổng kết nối.

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| id | int(1) | Khóa chính, chỉ có giá trị duy nhất |
| path | varchar(255) | Đường dẫn thư mục gốc của server |
| port | int(5) | Cổng mà server sẽ lẳng nghe kết nối |

## ****2.6. Hiện thực chương trình và kiểm thử****

2.6.1. FTP Server

A screenshot of a computer

Description automatically generated

Hình 2.6.1.1. Khởi chạy server

A screenshot of a computer

Description automatically generated

Hình 2.6.1.2. Các thông tin khi có người kết nối tới server

2.5.2. FTP Client

A screenshot of a computer

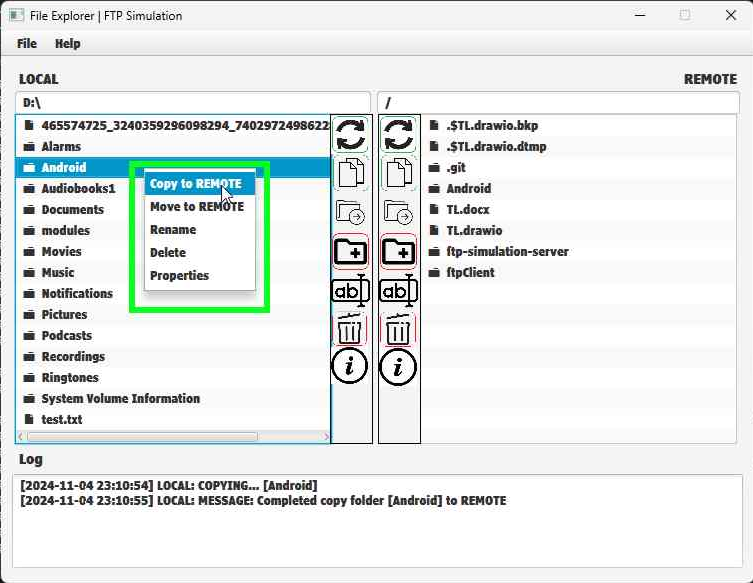
Description automatically generated

Hình 2.6.2.1. Màn hình nhập thông tin kết nối tới server

A screenshot of a computer

Description automatically generated

Hình 2.6.2.2. Màn hình chính của client

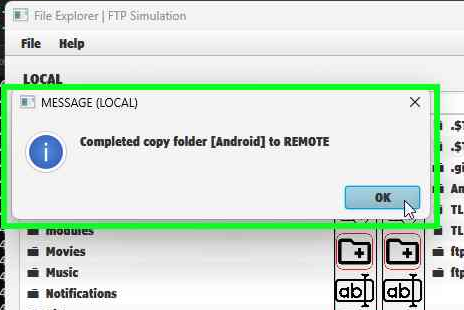


Hình 2.6.2.3. Thao tác tệp/thư mục bằng chuột phải

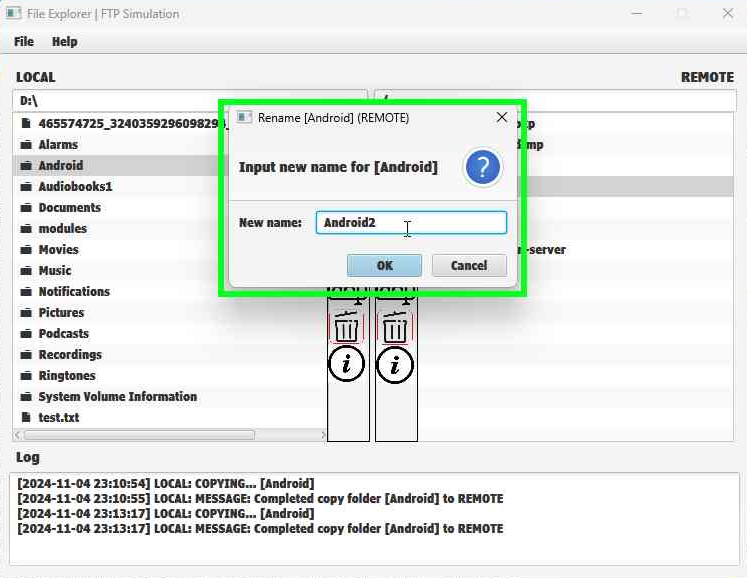
A screenshot of a computer

Description automatically generated

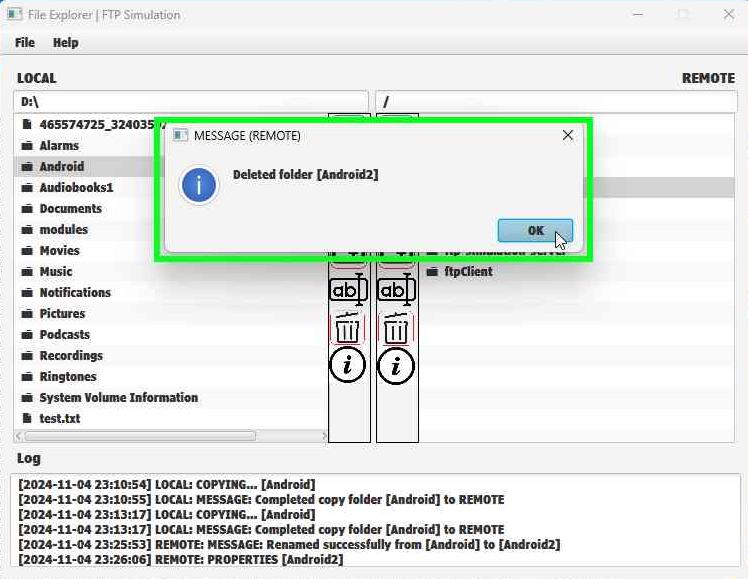
Hình 2.6.2.4. Xác nhận trước khi thực thi chức năng (Sao chép sang server)



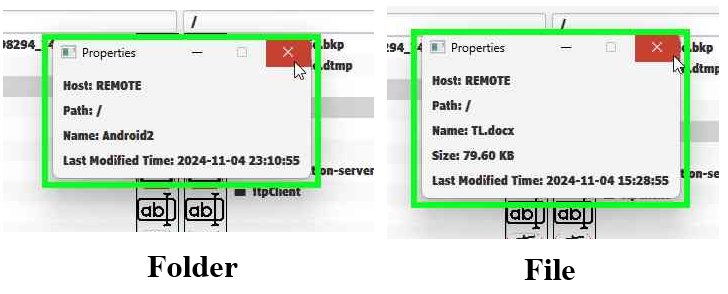
Hình 2.6.2.5. Thông báo sau khi hoàn thành chức năng



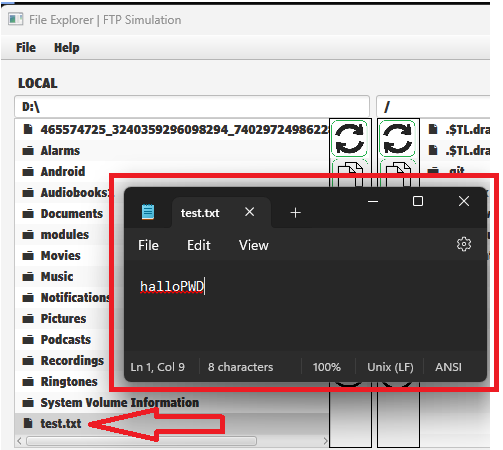
Hình 2.5.2.6. Chức năng đổi tên tệp/thư mục



*Hình 2.6.2.7. Chức năng xóa tệp/thư mục*

**

*Hình 2.6.2.8. Chức năng xem thông tin tệp/thư mục*

**

*Hình 2.6.2.9. Mở trực tiếp tệp tin tại client/server*

# CHƯƠNG 3. TỔNG KẾT

## ****3.1. Kết quả đạt được****

Sau quá trình phát triển, ứng dụng mô phỏng FTP server và client đã đạt được một số kết quả quan trọng:

**- Chức năng truyền và nhận tệp tin**: Ứng dụng cho phép người dùng kết nối, tải lên và tải xuống tệp tin giữa client và server một cách nhanh chóng và ổn định.

**- Giao diện người dùng thân thiện**: Client đều được thiết kế với giao diện trực quan, dễ sử dụng. Các chức năng như đăng nhập, quản lý tệp tin, và cấu hình server đều dễ tiếp cận và thao tác.

**- Cấu hình server linh hoạt**: Ứng dụng cho phép quản trị viên tùy chỉnh các cài đặt server như cổng kết nối và thư mục gốc. Điều này giúp đáp ứng nhu cầu linh hoạt của người dùng và dễ dàng thích ứng với các môi trường sử dụng khác nhau.

- **Quản lý tệp tin và thư mục:** Ứng dụng cho phép người dùng sao chép, di chuyển, đổi tên, xóa, xem thông tin tệp và thư mục một cách nhanh chóng và dễ dàng chỉ với vài click chuột.

**- Khả năng quản lý và giám sát server**: Giao diện quản trị cung cấp thông tin về trạng thái server, số lượng kết nối và hiệu suất hoạt động, giúp quản trị viên giám sát và tối ưu hóa hiệu suất hệ thống.

**- Cơ sở dữ liệu MySQL**: Hệ thống cơ sở dữ liệu lưu trữ thông tin người dùng và cấu hình server, đảm bảo quản lý dữ liệu nhất quán, bảo mật và dễ dàng truy cập.

## ****3.2. Đánh giá về đề tài****

3.2.1. Ưu điểm của đề tài

**- Ứng dụng có tính ứng dụng cao**: Ứng dụng mô phỏng FTP server và client đáp ứng nhu cầu truyền tải và chia sẻ dữ liệu, có thể sử dụng trong nhiều lĩnh vực và môi trường khác nhau.

- **Dễ sử dụng**: Ứng dụng mô phỏng FTP server và client được thiết kế với giao diện thân thiện, giúp người dùng dễ dàng kết nối, quản lý tệp tin và thư mục mà không cần kiến thức kỹ thuật cao.

- **Hiệu suất ổn định**: Ứng dụng đảm bảo tốc độ truyền tải ổn định, ngay cả khi có nhiều người dùng truy cập đồng thời, giúp nâng cao hiệu quả làm việc.

3.2.2. Khuyết điểm của đề tài

**- Chưa hỗ trợ đa nền tảng**: Ứng dụng mới được phát triển cho một nền tảng cụ thể, chưa có tính tương thích cao với nhiều hệ điều hành khác nhau như macOS, Linux, iOS hoặc Android.

**- Còn phụ thuộc vào kiến trúc FTP**: Ứng dụng sử dụng giao thức FTP truyền thống, có thể gặp hạn chế về tốc độ và bảo mật so với các phương thức truyền tải dữ liệu hiện đại.

- **Thiếu khả năng phục hồi kết nối**: Ứng dụng chưa có chức năng khôi phục khi người dùng có sự cố về mạng.

## ****3.3. Hướng phát triển và mở rộng đề tài****

Để nâng cao hiệu quả và tính năng của ứng dụng mô phỏng FTP server và client, sau đây là các hướng có thể cải thiện trong tương lai:

- **Hổ trợ đa ngôn ngữ:** Hỗ trợ đa ngôn ngữ để ứng dụng có thể phục vụ người dùng ở nhiều quốc gia và khu vực khác nhau.

- **Quản lý phiên đăng nhập:** Tích hợp quản lý phiên làm việc để giám sát và kiểm tra các hoạt động đăng nhập bất thường.

- **Chuyển sang giao thức bảo mật SFTP hoặc FTPS:** Thay thế FTP thông thường bằng các giao thức bảo mật như SFTP hoặc FTPS, giúp nâng cao khả năng bảo vệ dữ liệu và tính bảo mật toàn diện.

- **Quản lý quyền truy cập theo vai trò:** Cung cấp khả năng phân quyền theo vai trò (người dùng, quản trị viên, khách), cho phép quản lý quyền truy cập từng tệp hoặc thư mục chi tiết hơn.

- **Lưu trữ và quản lý lịch sử truyền tải**: Bổ sung chức năng lưu lại lịch sử truyền tải và các thao tác của người dùng để theo dõi và kiểm tra khi cần thiết.

- **Khả năng phục hồi kết nối:** Cài đặt chức năng khôi phục các phiên truyền tải bị gián đoạn, cho phép người dùng tiếp tục tải lên hoặc tải xuống tệp tin khi có sự cố mạng.

- **Xây dựng ứng dụng đa nền tảng**: Phát triển ứng dụng cho nhiều hệ điều hành như macOS, Linux, iOS và Android để đáp ứng nhu cầu đa dạng của người dùng trên các thiết bị khác nhau.

- **Cải tiến giao diện người dùng**: Tăng cường trải nghiệm người dùng bằng cách cải tiến giao diện, bổ sung chế độ tối (dark mode), và hỗ trợ kéo-thả trực quan để cải thiện khả năng sử dụng của ứng dụng.

# ****TÀI LIỆU THAM KHẢO****

1. Trương Đình Triết, Đỗ Minh Hằng. (2010). *Quản lý Yêu cầu Phần mềm*. Nhà Xuất Bản Lao Động - Xã Hội.