

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIA ĐỊNH
KHOA: CÔNG NGHỆ THÔNG TIN**



TIỂU LUẬN

XÂY DỰNG ỨNG DỤNG MINI ZALO

MÔN: LẬP TRÌNH ỨNG DỤNG JAVA

Ngành: KỸ THUẬT PHẦN MỀM

Giảng viên hướng dẫn: NGUYỄN THÀNH SƠN

Sinh viên thực hiện: VŨ ĐỨC THỊNH

MSSV: 22140459

Lớp: 221402

TP. Hồ Chí Minh, 11 năm 2024

XÂY DỰNG ỨNG DỤNG MINI ZALO

Nhóm sinh viên thực hiện

STT	Họ và tên sinh viên	MSSV	Công sức đóng góp (tổng 100%)
1	Vũ Đức Thịnh	22140459	100%

BẢNG PHÂN CÔNG	
Họ và tên sinh viên	Công việc
Vũ Đức Thịnh	Thiết kế giao diện
Vũ Đức Thịnh	Thiết kế sơ đồ
Vũ Đức Thịnh	Viết chương trình
Vũ Đức Thịnh	Làm powerpoint
Vũ Đức Thịnh	Làm word

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 “Lập trình ứng dụng Java”, 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 mini Zalo”.

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!

MỤC LỤC

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	1
1.1. Java là gì?	1
1.2. Java trong lập trình ứng dụng.....	1
1.3. Các ưu điểm và nhược điểm của Java.	2
1.4. Tổng quan về môn học.....	3
1.5. Các nội dung của môn học.....	3
CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI.....	5
2.1. Khảo sát hiện trạng.	5
2.2. Đặt ra yêu cầu.....	5
2.3. Sơ đồ use case.	6
2.3.1. Sơ đồ use case tổng quát	6
2.3.2. Đặc tả use case	7
2.4. Sơ đồ lớp.....	9
2.5. Kiểm thử chương trình.....	10
2.5.1. Server	10
2.5.2. Client	11
CHƯƠNG 3: TỔNG KẾT	19
3.1. Kết quả đạt được.	19
3.2. Đánh giá ưu khuyết điểm.	19
3.3. Hướng phát triển và mở rộng đề tài.	19
TÀI LIỆU THAM KHẢO.....	21

MỤC LỤC HÌNH ẢNH

Hình 2.3.1.1. Sơ đồ use case tổng quát	6
Hình 2.3.2.1. Đặc tả use case đăng nhập.....	7
Hình 2.3.2.2. Đặc tả use case đăng ký.....	7
Hình 2.3.2.3. Đặc tả use case đăng xuất.....	8
Hình 2.3.2.4. Đặc tả use case gửi tin nhắn	8
Hình 2.3.2.5. Đặc tả use case nhận tin nhắn	8
Hình 2.4.1a. Sơ đồ lớp tổng quát client.....	9
Hình 2.4.1b. Sơ đồ lớp tổng quát server	10
Hình 2.5.1.1. Chạy server thành công.....	10
Hình 2.5.1.2. Chạy server khi có một server khác đang chạy.....	10
Hình 2.5.1.3. Server khi có client kết nối.....	10
Hình 2.5.1.4. Server khi có client ngắt kết nối.....	11
Hình 2.5.2.1. Form đăng nhập.....	11
Hình 2.5.2.2. Đăng nhập	11
Hình 2.5.2.3. Form chat.....	12
Hình 2.5.2.4. Thông báo đã đăng nhập ở nơi khác	12
Hình 2.5.2.5. Thông báo sai tên và mật khẩu.....	12
Hình 2.5.2.6. Thông báo tên và mật khẩu không được để trống.....	13
Hình 2.5.2.7. Form đăng ký	13
Hình 2.5.2.8. Điền đầy đủ thông tin	13
Hình 2.5.2.9. Thông báo đăng ký thành công	14
Hình 2.5.2.10. Đăng ký với tên tài khoản đã đăng ký.....	14
Hình 2.5.2.11. Thông báo không được để trống	14
Hình 2.5.2.12. Hệ thống báo có người dùng đăng nhập	15
Hình 2.5.2.13. Gửi mọi người – người gửi	15
Hình 2.5.2.14. Gửi mọi người – người nhận.....	16
Hình 2.5.2.15. Gửi một người – người gửi	16
Hình 2.5.2.16. Gửi một người – người nhận.....	17
Hình 2.5.2.17. Thông báo chưa nhập thông điệp	17
Hình 2.5.2.18. Người dùng đã đăng xuất	18

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Java là gì?

Java là một ngôn ngữ lập trình hướng đối tượng, được phát triển bởi Sun Microsystems vào giữa những năm 1990, dưới sự lãnh đạo của James Gosling. Ngôn ngữ này ra đời với mục tiêu tạo ra một môi trường lập trình đơn giản, an toàn và dễ dàng sử dụng, đồng thời có khả năng chạy trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn. Điều này được gọi là nguyên tắc "viết một lần, chạy mọi nơi" (WORA - Write Once, Run Anywhere). Để đạt được điều này, Java sử dụng Java Virtual Machine (JVM), một lớp trung gian cho phép mã Java được biên dịch thành bytecode, có thể chạy trên bất kỳ thiết bị nào có cài đặt JVM.

Java có cú pháp tương tự như C và C++, nhưng đã được tinh giản để loại bỏ một số tính năng phức tạp và dễ gây lỗi. Điều này giúp Java trở nên dễ học hơn cho những lập trình viên mới bắt đầu. Ngôn ngữ này cũng hỗ trợ nhiều tính năng hiện đại như lập trình đa luồng, quản lý bộ nhớ tự động thông qua Garbage Collection, và cung cấp một hệ thống thư viện phong phú với nhiều API cho phép lập trình viên dễ dàng phát triển ứng dụng.

Java không chỉ được sử dụng để phát triển ứng dụng máy tính để bàn mà còn được ứng dụng rộng rãi trong phát triển ứng dụng web, ứng dụng di động, hệ thống nhúng, và các dịch vụ máy chủ. Sự linh hoạt và khả năng mở rộng của Java đã giúp nó trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

1.2. Java trong lập trình ứng dụng.

Java được ứng dụng trong nhiều lĩnh vực khác nhau, nhờ vào tính linh hoạt và khả năng mở rộng của nó. Trong lĩnh vực lập trình ứng dụng web, Java đã trở thành một trong những ngôn ngữ hàng đầu nhờ vào các framework mạnh mẽ như Spring, Hibernate, và JavaServer Faces (JSF). Các framework này cung cấp các công cụ và thư viện cần thiết để phát triển các ứng dụng web động, phức tạp và có khả năng mở rộng cao. Spring, chẳng hạn, nổi tiếng với khả năng quản lý các thành phần ứng dụng và cung cấp các giải pháp cho vấn đề bảo mật, quản lý giao dịch và tích hợp.

Bên cạnh đó, Java là ngôn ngữ chính để phát triển ứng dụng di động thông qua Android SDK. Android, hệ điều hành di động phổ biến nhất thế giới, chủ yếu được xây dựng trên nền tảng Java. Điều này cho phép các lập trình viên tạo ra các ứng dụng di động đa dạng và phong phú, từ các trò chơi cho đến các ứng dụng kinh doanh, với khả năng tương tác cao và trải nghiệm người dùng tốt.

Ngoài ra, Java cũng có thể được sử dụng để phát triển các ứng dụng cho thiết bị nhúng thông qua Java ME (Micro Edition). Java ME được tối ưu hóa cho các thiết bị có tài nguyên hạn chế, như điện thoại di động cũ, máy tính bảng và các thiết bị IoT. Điều này cho phép Java mở rộng ra ngoài các ứng dụng máy tính và di động, đến với các thiết bị thông minh và hệ thống nhúng.

Java cũng rất mạnh mẽ trong việc phát triển các ứng dụng máy chủ, chẳng hạn như dịch vụ web RESTful và SOAP. Các ứng dụng này thường được sử dụng trong các hệ thống phân tán, cho phép giao tiếp giữa các dịch vụ khác nhau qua mạng. Nhờ vào khả năng mở rộng và hiệu suất cao của Java, các doanh nghiệp có thể xây dựng các hệ thống phức tạp mà vẫn đảm bảo hiệu suất và độ tin cậy.

Cuối cùng, Java cũng được sử dụng trong phát triển game, mặc dù không phổ biến như một số ngôn ngữ khác. Java cung cấp các thư viện và framework như LibGDX và jMonkeyEngine, cho phép các lập trình viên xây dựng các trò chơi đa nền tảng, từ các trò chơi 2D đơn giản đến các trò chơi 3D phức tạp.

1.3. Các ưu điểm và nhược điểm của Java.

Java mang lại nhiều ưu điểm đáng chú ý, điều này đã góp phần vào sự phổ biến của nó trong cộng đồng lập trình. Đầu tiên, tính di động của Java là một trong những điểm mạnh lớn nhất. Với khả năng chạy trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn, Java giúp lập trình viên tiết kiệm thời gian và công sức trong việc phát triển và triển khai ứng dụng. Điều này đặc biệt quan trọng trong môi trường phát triển hiện đại, nơi mà các ứng dụng cần phải hoạt động trên nhiều thiết bị và hệ điều hành khác nhau.

Java sử dụng Garbage Collection để tự động quản lý bộ nhớ, giúp giảm thiểu lỗi rò rỉ bộ nhớ và tăng tính ổn định của ứng dụng. Điều này có nghĩa là lập trình viên không cần phải lo lắng quá nhiều về việc giải phóng bộ nhớ, cho phép họ tập trung vào việc phát triển tính năng và cải thiện hiệu suất của ứng dụng. Hệ thống lập trình hướng đối tượng của Java cũng hỗ trợ các nguyên tắc như kế thừa, đa hình, và đóng gói, giúp tổ chức mã nguồn một cách rõ ràng và dễ bảo trì.

Tuy nhiên, Java cũng có một số nhược điểm. Mặc dù hiệu suất của Java là tốt, nhưng nó thường chậm hơn so với các ngôn ngữ biên dịch như C hoặc C++ do việc chạy trên JVM. Điều này có thể trở thành một vấn đề trong các ứng dụng yêu cầu hiệu suất cao hoặc thời gian thực. Thêm vào đó, các ứng dụng Java thường tiêu tốn nhiều bộ nhớ hơn, điều này có thể gây khó khăn cho các thiết bị có tài nguyên hạn chế.

Cú pháp của Java có thể gây khó khăn cho những lập trình viên mới, đặc biệt khi so với các ngôn ngữ khác như Python, vốn nổi tiếng với cú pháp đơn giản và dễ hiểu. Điều này có thể làm cho quá trình học tập trở nên khó khăn hơn đối với những người mới bắt đầu. Cuối cùng, đối với các ứng dụng nhỏ hoặc đơn giản, việc sử dụng Java có thể gây ra sự phức tạp không cần thiết so với các ngôn ngữ khác, dẫn đến việc tăng thời gian phát triển và chi phí.

1.4. Tổng quan về môn học.

Môn học lập trình ứng dụng Java là một phần quan trọng trong chương trình đào tạo của nhiều ngành học liên quan đến công nghệ thông tin, khoa học máy tính và kỹ thuật phần mềm. Môn học này nhằm cung cấp cho sinh viên những kiến thức cơ bản và nâng cao về ngôn ngữ lập trình Java, giúp họ phát triển khả năng lập trình và tư duy logic trong việc giải quyết các vấn đề thực tiễn.

Trong môn học này, sinh viên sẽ được giới thiệu về các khái niệm cơ bản của lập trình hướng đối tượng, các cấu trúc dữ liệu, thuật toán, và cách xây dựng ứng dụng từ những thành phần cơ bản nhất. Môn học không chỉ tập trung vào lý thuyết mà còn chú trọng đến thực hành, giúp sinh viên có cơ hội áp dụng kiến thức vào các dự án thực tế. Điều này sẽ giúp sinh viên nắm vững các kỹ năng cần thiết để phát triển các ứng dụng Java.

1.5. Các nội dung của môn học.

Môn học lập trình ứng dụng Java cung cấp cho sinh viên một nền tảng vững chắc về ngôn ngữ này, từ những khái niệm cơ bản đến các kỹ thuật lập trình nâng cao. Bắt đầu với việc giới thiệu về Java, sinh viên sẽ tìm hiểu về lịch sử phát triển và các ứng dụng thực tiễn của ngôn ngữ này trong thế giới công nghệ hiện đại. Họ cũng sẽ được hướng dẫn cách cài đặt JDK và sử dụng các môi trường phát triển tích hợp (IDE) như Eclipse hoặc IntelliJ IDEA để viết mã một cách hiệu quả.

Sau đó, sinh viên sẽ học cú pháp và cấu trúc chương trình Java, bao gồm các kiểu dữ liệu, biến, toán tử và cấu trúc điều khiển. Điều này sẽ giúp họ xây dựng những chương trình Java đơn giản và hiểu rõ cách thức hoạt động của ngôn ngữ. Một phần quan trọng không thể thiếu là lập trình hướng đối tượng, nơi sinh viên sẽ khám phá các khái niệm như lớp, đối tượng, kế thừa, đa hình và đóng gói. Những kiến thức này sẽ trang bị cho họ khả năng thiết kế và xây dựng các lớp và đối tượng một cách hiệu quả, giúp nâng cao tính tổ chức và khả năng bảo trì của mã nguồn.

Môn học cũng sẽ đề cập đến xử lý ngoại lệ, giúp sinh viên nắm vững cách quản lý lỗi và đảm bảo tính ổn định cho ứng dụng thông qua các cấu trúc try-catch và throw. Bên cạnh

đó, lập trình đa luồng (Thread) sẽ là một nội dung quan trọng, nơi sinh viên sẽ tìm hiểu cách tạo và quản lý các luồng để thực hiện nhiều tác vụ đồng thời, đồng thời khám phá các khái niệm về đồng bộ hóa và cách sử dụng `ExecutorService` để tối ưu hóa hiệu suất.

Sinh viên cũng sẽ được giới thiệu về cách xây dựng giao diện người dùng (GUI) bằng `JavaFX`, giúp họ tạo ra các thành phần giao diện như cửa sổ, nút bấm và bảng, từ đó nâng cao trải nghiệm người dùng. Mô hình thiết kế MVC (Model-View-Controller) sẽ được trình bày, giúp sinh viên hiểu cách tổ chức mã nguồn một cách hợp lý để tách biệt logic ứng dụng, giao diện người dùng và dữ liệu, từ đó cải thiện khả năng bảo trì và mở rộng ứng dụng.

Nội dung về xử lý Input/Output (IO) sẽ giúp sinh viên tìm hiểu các phương pháp đọc và ghi dữ liệu trong Java, bao gồm việc làm việc với tệp tin và luồng dữ liệu. Cuối cùng, lập trình mạng (Network) sẽ là một phần quan trọng, nơi sinh viên sẽ học về giao thức TCP/IP, cách tạo kết nối mạng và xây dựng các ứng dụng mạng đơn giản.

Để củng cố những kiến thức đã học, sinh viên sẽ tham gia vào các bài thực hành và dự án cuối kỳ. Đây là cơ hội để họ áp dụng lý thuyết vào thực tiễn, phát triển một ứng dụng Java hoàn chỉnh và nâng cao kỹ năng làm việc nhóm cũng như quản lý dự án. Thông qua những nội dung này, môn học sẽ trang bị cho sinh viên những kỹ năng cần thiết để trở thành những lập trình viên Java có năng lực, sẵn sàng đáp ứng nhu cầu của thị trường lao động.

CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI

2.1. Khảo sát hiện trạng.

Trong bối cảnh công nghệ thông tin ngày càng phát triển, nhu cầu giao tiếp và kết nối giữa người dùng trở nên quan trọng hơn bao giờ hết. Các ứng dụng nhắn tin trực tuyến như Zalo, Facebook Messenger, WhatsApp đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của người dùng. Tuy nhiên, bên cạnh những ứng dụng lớn, vẫn còn nhiều người dùng tìm kiếm những giải pháp đơn giản, nhẹ nhàng và dễ sử dụng hơn.

Khảo sát hiện trạng cho thấy, nhiều ứng dụng nhắn tin hiện tại vẫn còn tồn tại một số vấn đề như giao diện phức tạp, yêu cầu cấu hình cao. Hơn nữa, giao diện người dùng của nhiều ứng dụng không thân thiện, gây khó khăn cho người mới sử dụng. Một số ứng dụng cũng yêu cầu thiết bị có cấu hình cao, khiến người dùng với thiết bị cũ gặp khó khăn trong việc sử dụng.

Trước thực trạng này, việc xây dựng một ứng dụng nhắn tin đơn giản và hiệu quả như "Mini Zalo" là cần thiết. Ứng dụng này không chỉ đáp ứng nhu cầu giao tiếp của người dùng mà còn mang lại trải nghiệm tốt hơn thông qua giao diện thân thiện và dễ sử dụng.

2.2. Đặt ra yêu cầu.

Dựa trên khảo sát hiện trạng, các yêu cầu cho ứng dụng "Mini Zalo" được đặt ra như sau:

Tính năng nhắn tin: Ứng dụng cần hỗ trợ người dùng gửi và nhận tin nhắn văn bản một cách nhanh chóng và dễ dàng, đồng thời cho phép gửi ảnh, video và tệp đính kèm.

Giao diện người dùng thân thiện: Giao diện của ứng dụng phải đơn giản, dễ sử dụng và trực quan, giúp người dùng dễ dàng tìm kiếm và sử dụng các tính năng.

Đăng ký và đăng nhập: Ứng dụng cần có chức năng đăng ký tài khoản mới và đăng nhập cho người dùng, đảm bảo tính bảo mật và riêng tư cho thông tin cá nhân.

Quản lý danh bạ: Người dùng có thể thêm, xóa và quản lý danh bạ của mình, giúp việc kết nối với bạn bè trở nên thuận tiện hơn.

Thông báo: Ứng dụng cần có hệ thống thông báo để người dùng nhận biết khi có tin nhắn mới hoặc hoạt động khác trong ứng dụng.

Bảo mật thông tin: Cần đảm bảo rằng thông tin cá nhân và dữ liệu của người dùng được bảo mật, không bị rò rỉ ra bên ngoài.

Tương thích với nhiều thiết bị: Ứng dụng nên hoạt động tốt trên nhiều loại thiết bị và hệ điều hành khác nhau, từ điện thoại di động đến máy tính.

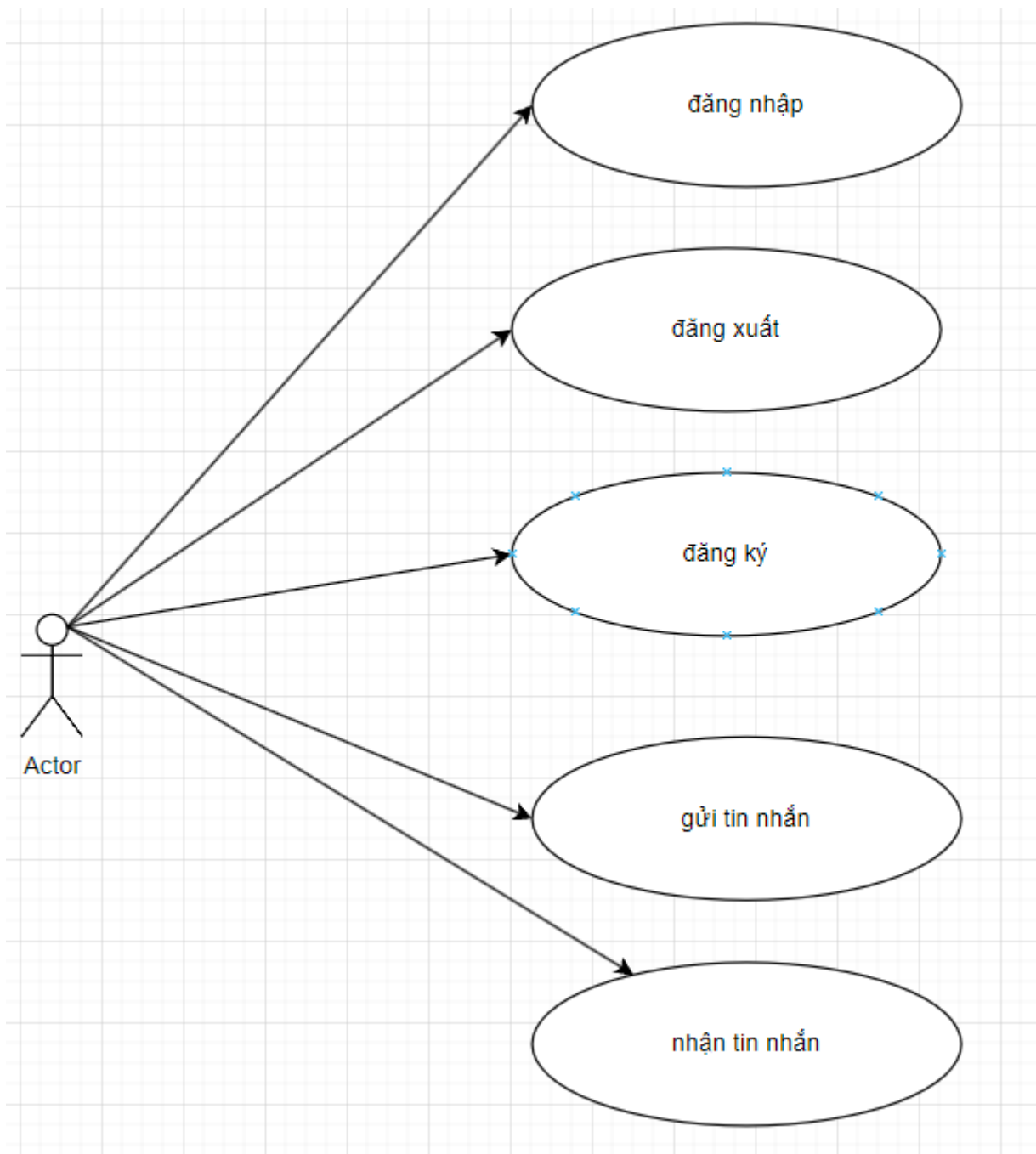
Hỗ trợ đa ngôn ngữ: Để phục vụ người dùng từ nhiều quốc gia khác nhau, ứng dụng nên hỗ trợ nhiều ngôn ngữ khác nhau.

Khả năng mở rộng: Ứng dụng cần được thiết kế với khả năng mở rộng, cho phép thêm các tính năng mới trong tương lai mà không làm ảnh hưởng đến hiệu suất hoạt động.

Thông qua việc đáp ứng các yêu cầu này, ứng dụng "Mini Zalo" sẽ không chỉ là một công cụ giao tiếp hiệu quả mà còn mang lại trải nghiệm người dùng tốt nhất, góp phần vào việc xây dựng một cộng đồng kết nối mạnh mẽ.

2.3. Sơ đồ use case.

2.3.1. Sơ đồ use case tổng quát



Hình 2.3.1.1. Sơ đồ use case tổng quát

2.3.2. Đặc tả use case

Đăng nhập		
input		tên tài khoản, mật khẩu
output		thành công / thất bại
primary course		
1		nhập vào thông tin input
2		Hệ thống kiểm tra thông tin và tình trạng đăng nhập
2.1		Nếu đúng và chưa đăng nhập -> mở form chat
3		Nếu sai hoặc đã đăng nhập -> thông báo thất bại

Hình 2.3.2.1. Đặc tả use case đăng nhập

Đăng ký		
input		tên tài khoản, mật khẩu
output		thành công / thất bại
primary course		
1		nhập vào thông tin input
2		Kiểm tra tên tài khoản đã tồn tại chưa
2.1		không tồn tại -> thông báo thành công, về form đăng nhập
3		tồn tại -> thông báo thất bại

Hình 2.3.2.2. Đặc tả use case đăng ký

Đăng xuất	
input	Đóng ứng dụng
output	đóng kết nối với server
primary course	
1	Hệ thống cập nhật trạng thái offline
2	Hệ thống thông báo cho user khác

Hình 2.3.2.3. Đặc tả use case đăng xuất

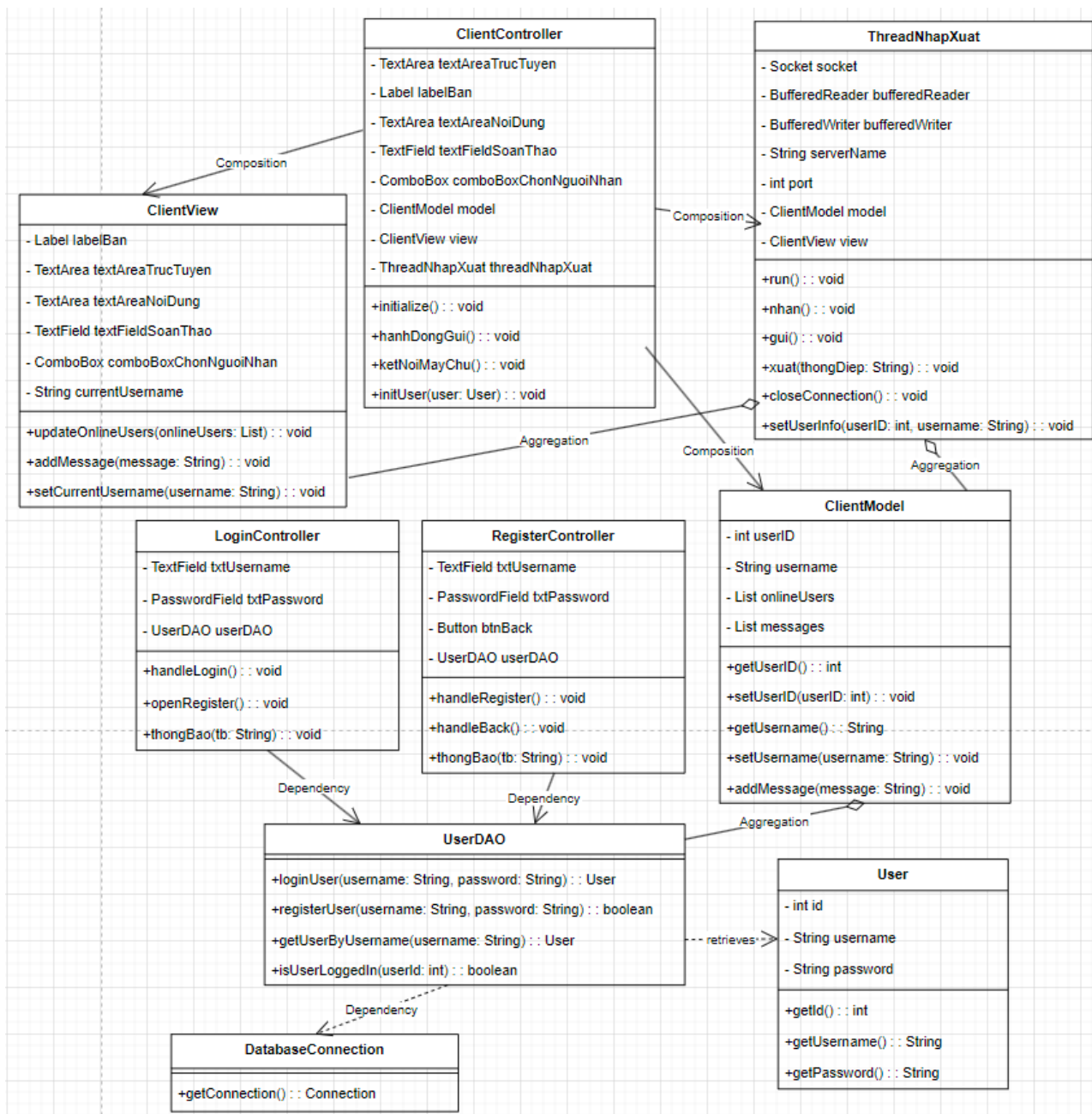
gửi tin nhắn	
input	người nhận, tin nhắn
output	hiển thị tin nhắn/thông báo thất bại
primary course	
1	chọn tên người nhận
2	nhập tin nhắn
3	Hệ thống gửi tin nhắn đến người nhận

Hình 2.3.2.4. Đặc tả use case gửi tin nhắn

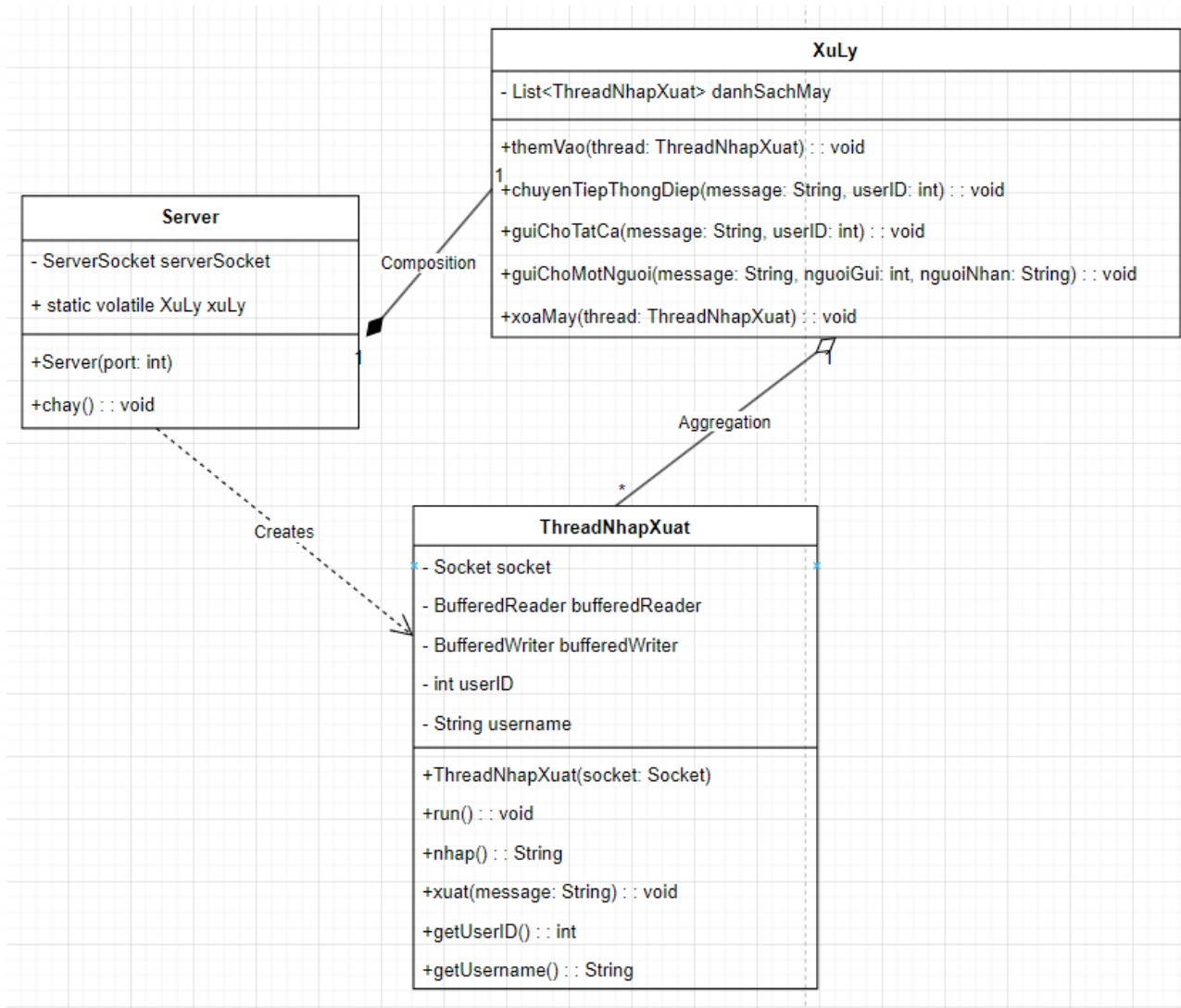
nhận tin nhắn	
input	tin nhắn từ server
output	hiển thị tin nhắn
primary course	
1	nhận tin nhắn từ người gửi
2	xuất tin nhắn cho người nhận

Hình 2.3.2.5. Đặc tả use case nhận tin nhắn

2.4. Sơ đồ lớp.



Hình 2.4.1a. Sơ đồ lớp tổng quát client.



Hình 2.4.1b. Sơ đồ lớp tổng quát server

2.5. Kiểm thử chương trình

2.5.1. Server

```
Server đang chờ kết nối từ client...
```

Hình 2.5.1.1. Chạy server thành công

```
Lỗi khởi tạo server: Address already in use: bind
Lỗi rồi: Address already in use: bind
```

Hình 2.5.1.2. Chạy server khi có một server khác đang chạy

```
Server đang chờ kết nối từ client...
Client đã kết nối: /192.168.1.29:62873
Số lượng client đang kết nối: 1
Server đang chờ kết nối từ client...
```

Hình 2.5.1.3. Server khi có client kết nối

```
Server đang chờ kết nối từ client...  
Client đã kết nối: /192.168.1.29:62873  
Số lượng client đang kết nối: 1  
Server đang chờ kết nối từ client...  
User ID 1 đã thoát.
```

Hình 2.5.1.4. Server khi có client ngắt kết nối

2.5.2. Client

- Đăng nhập

Khi bắt đầu chương trình → mở form đăng nhập

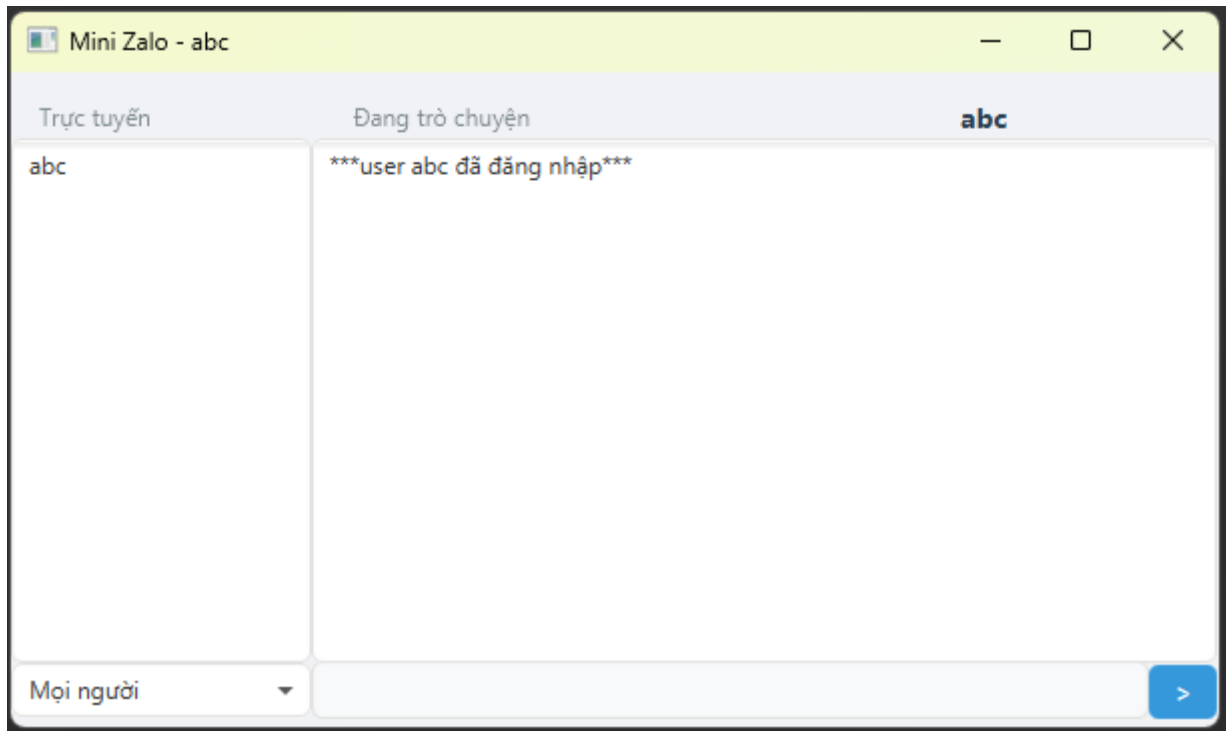
A screenshot of a web application window titled "Mini Zalo". The window displays a login form with the heading "Đăng Nhập" (Login). Below the heading, there are two input fields: "Tài khoản" (Username) and "Mật khẩu" (Password). At the bottom of the form, there are two blue buttons: "Đăng nhập" (Login) and "Đăng ký" (Register).

Hình 2.5.2.1. Form đăng nhập

Đăng nhập bằng tài khoản và mật khẩu đúng → click nút “Đăng nhập” → mở form chat và thông báo người dùng đã đăng nhập

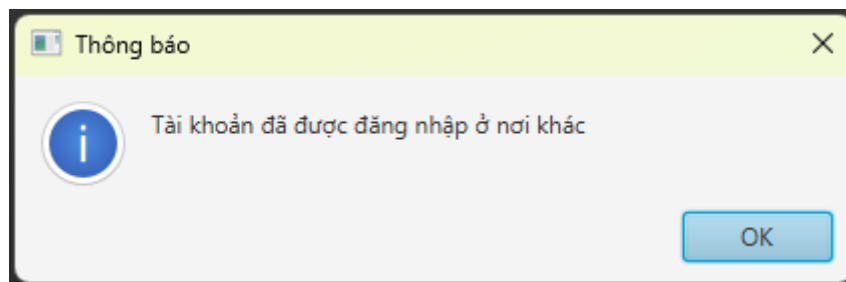
A screenshot of the same "Mini Zalo" login form. The "Tài khoản" (Username) field now contains the text "abc". The "Mật khẩu" (Password) field contains three dots, indicating a masked password. The "Đăng nhập" (Login) and "Đăng ký" (Register) buttons remain at the bottom.

Hình 2.5.2.2. Đăng nhập



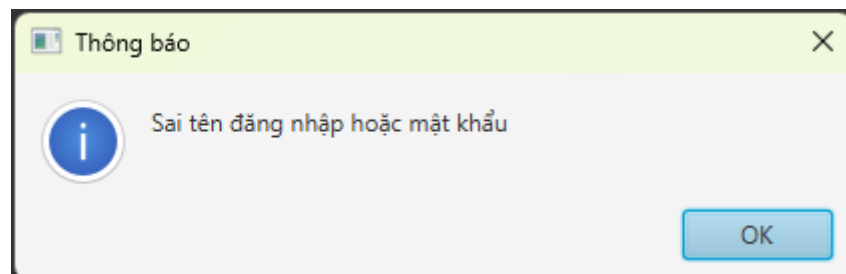
Hình 2.5.2.3. Form chat

Đăng nhập với tài khoản đang đăng nhập



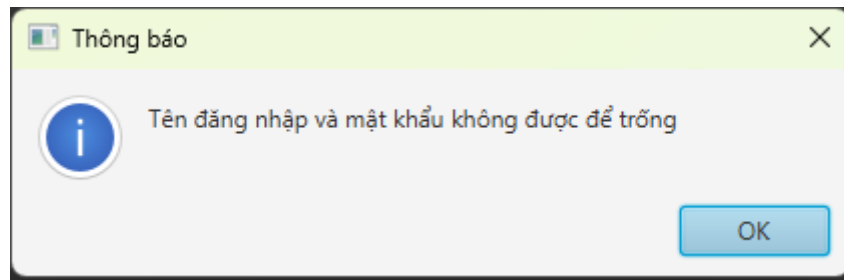
Hình 2.5.2.4. Thông báo đã đăng nhập ở nơi khác

Đăng nhập với tài khoản sai, mật khẩu sai hoặc chưa đăng ký



Hình 2.5.2.5. Thông báo sai tên và mật khẩu

Đăng nhập với tài khoản hoặc mật khẩu để trống



Hình 2.5.2.6. Thông báo tên và mật khẩu không được để trống

- Đăng ký

Click chuột vào nút đăng ký → mở form đăng ký và click nút “Quay lại” để về form đăng nhập

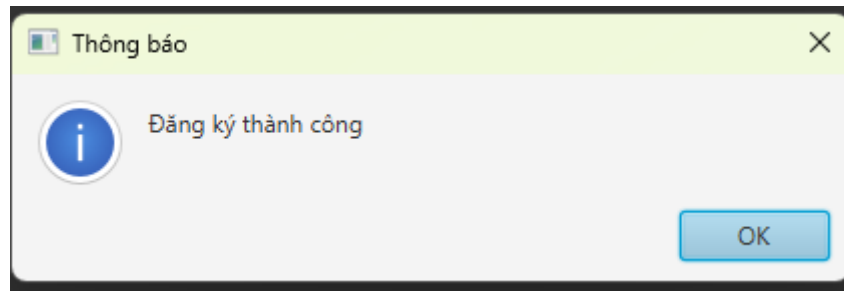
A screenshot of a web application window titled "Mini Zalo". The main heading is "Đăng ký" (Register). Below it are two input fields: "Tên tài khoản" (Username) and "Mật khẩu" (Password). At the bottom, there are two buttons: a blue "Đăng ký" (Register) button and a red "Quay lại" (Go back) button.

Hình 2.5.2.7. Form đăng ký

Nhập đầy đủ tên tài khoản và mật khẩu → click nút “đăng ký” → thông báo thành công và trở về form đăng nhập

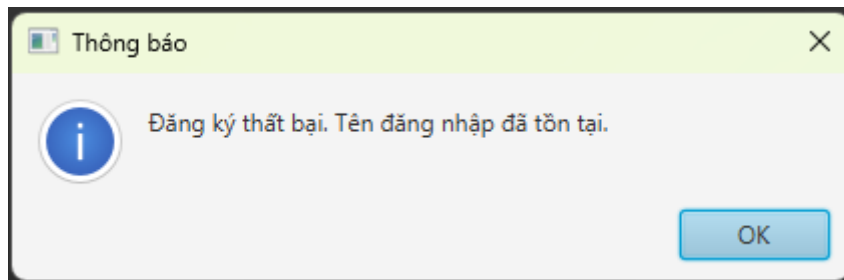
A screenshot of the "Mini Zalo" registration form. The "Tên tài khoản" (Username) field is filled with "test1". The "Mật khẩu" (Password) field is filled with three dots, indicating it is masked. The "Đăng ký" (Register) button is highlighted in blue, and the "Quay lại" (Go back) button is red.

Hình 2.5.2.8. Điền đầy đủ thông tin



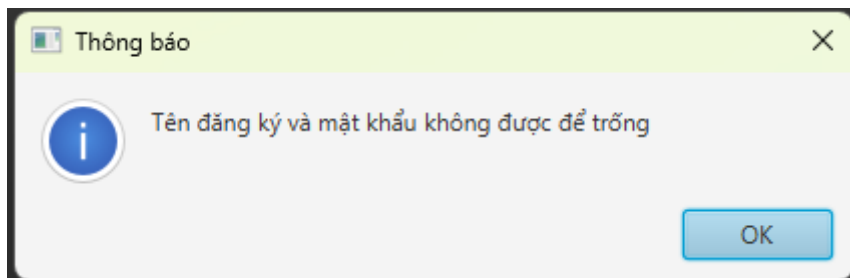
Hình 2.5.2.9. Thông báo đăng ký thành công

Đăng ký với tên tài khoản đã đăng ký



Hình 2.5.2.10. Đăng ký với tên tài khoản đã đăng ký

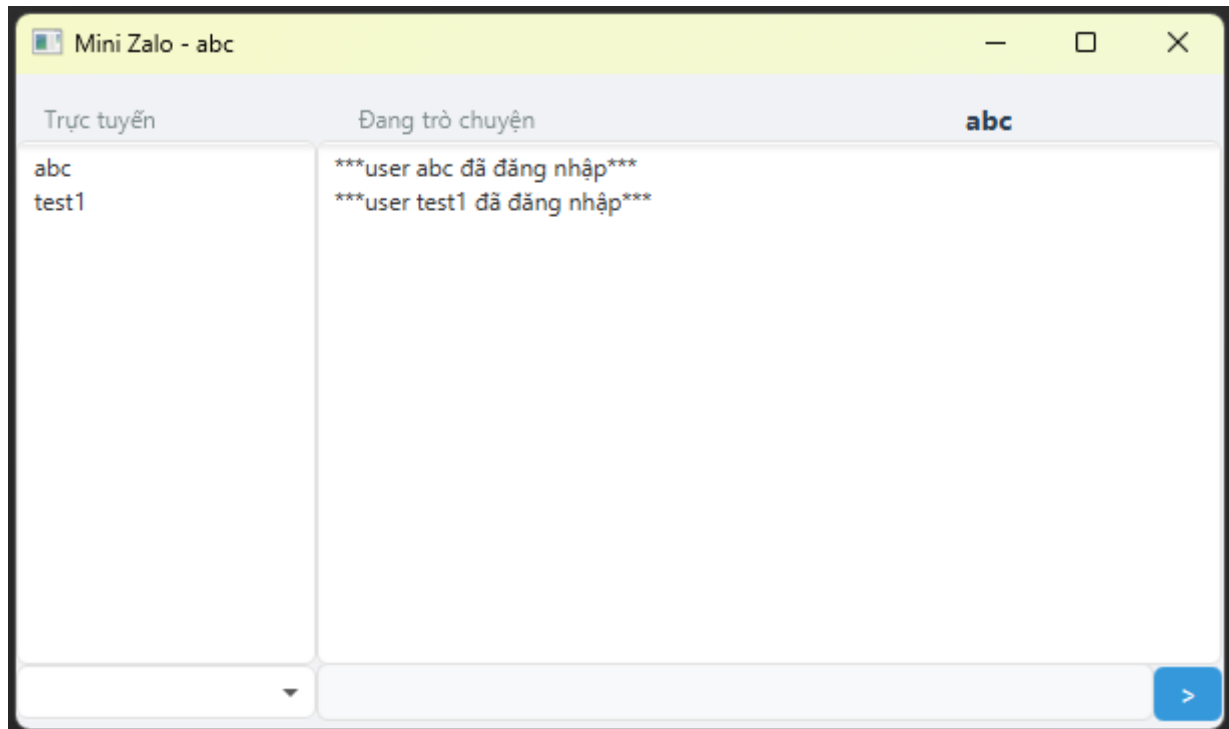
Đăng ký với tên tài khoản hoặc mật khẩu bỏ trống hoặc cả 2 đều bỏ trống



Hình 2.5.2.11. Thông báo không được để trống

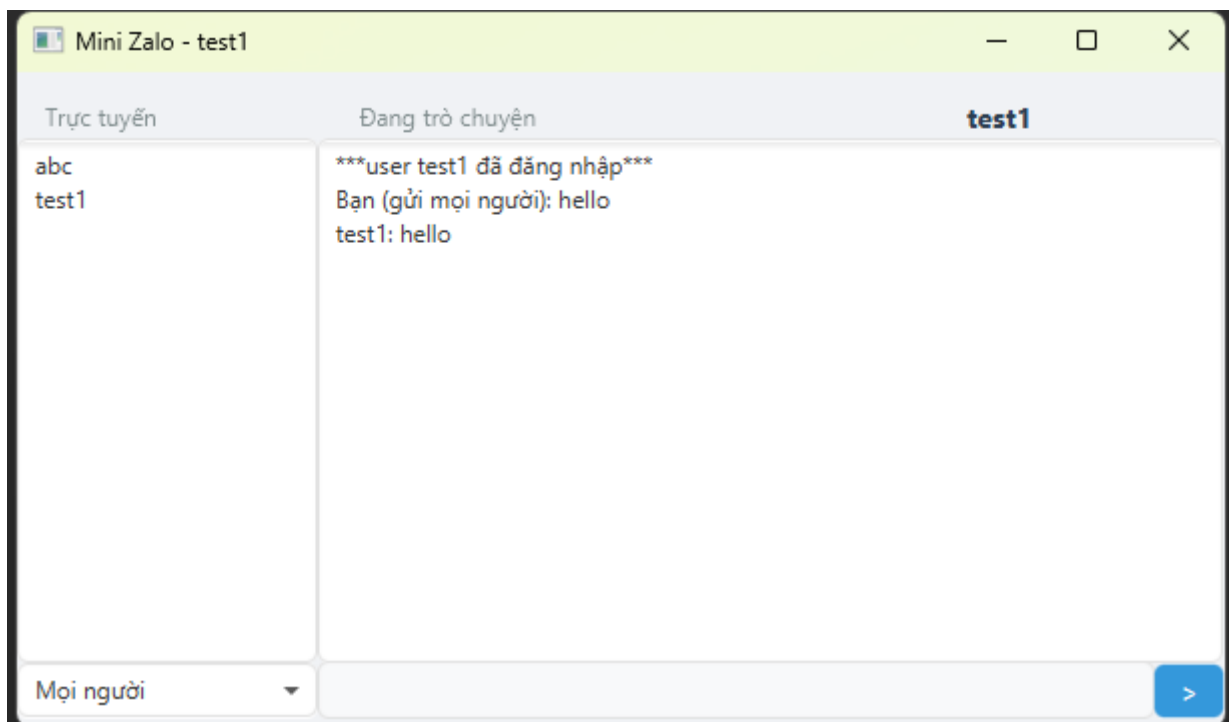
- Nhận và gửi tin nhắn

Khi có người dùng đăng nhập, hệ thống sẽ gửi thông báo

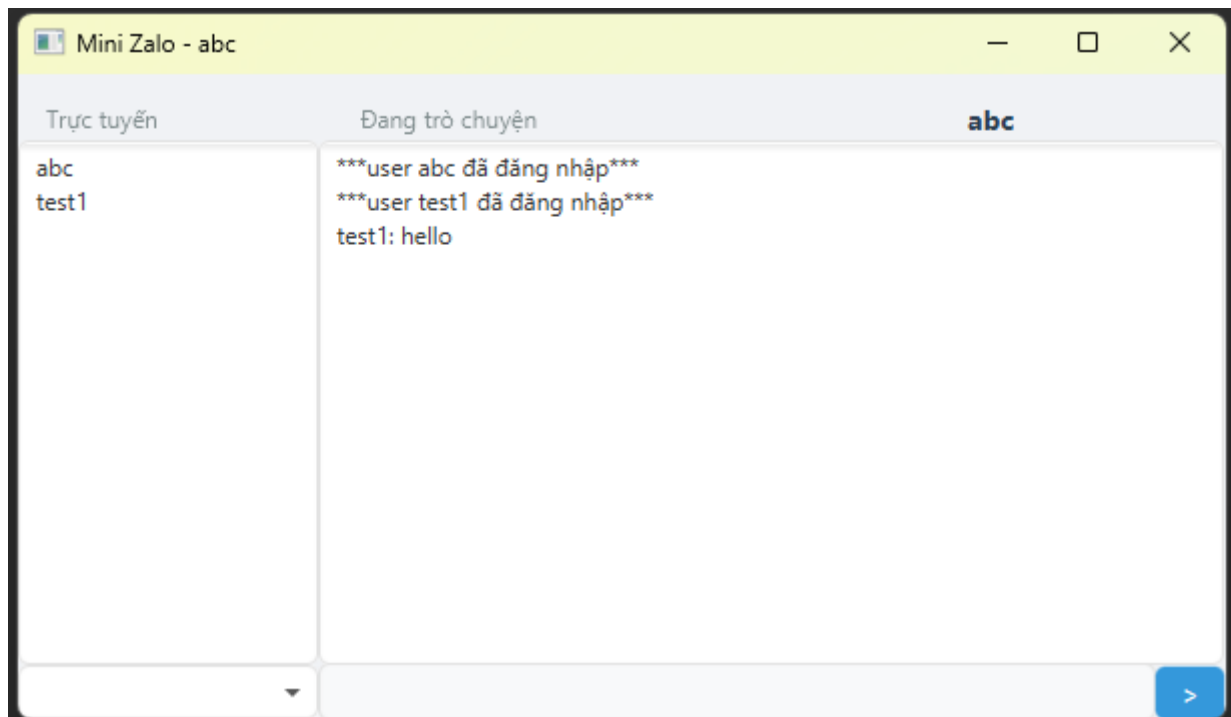


Hình 2.5.2.12. Hệ thống báo có người dùng đăng nhập

Khi người dùng chọn “Mọi người” trong ComboBox và nhập thông điệp → Click nút “>” để gửi cho mọi người

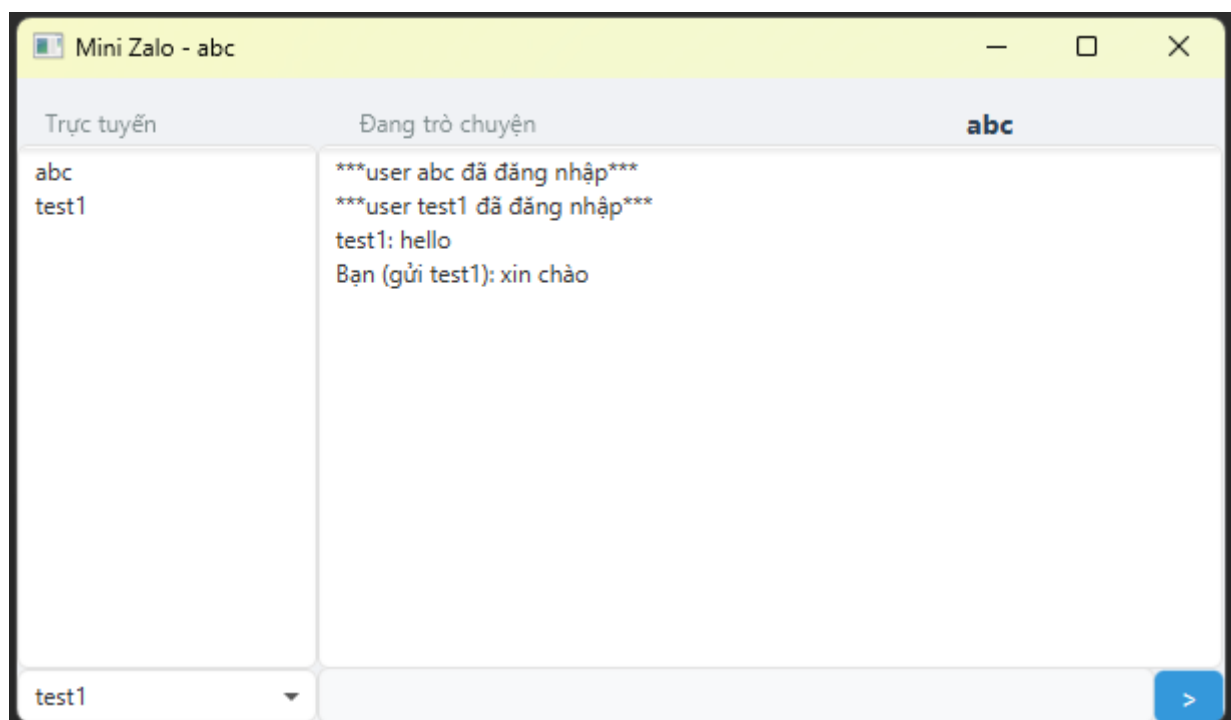


Hình 2.5.2.13. Gửi mọi người – người gửi

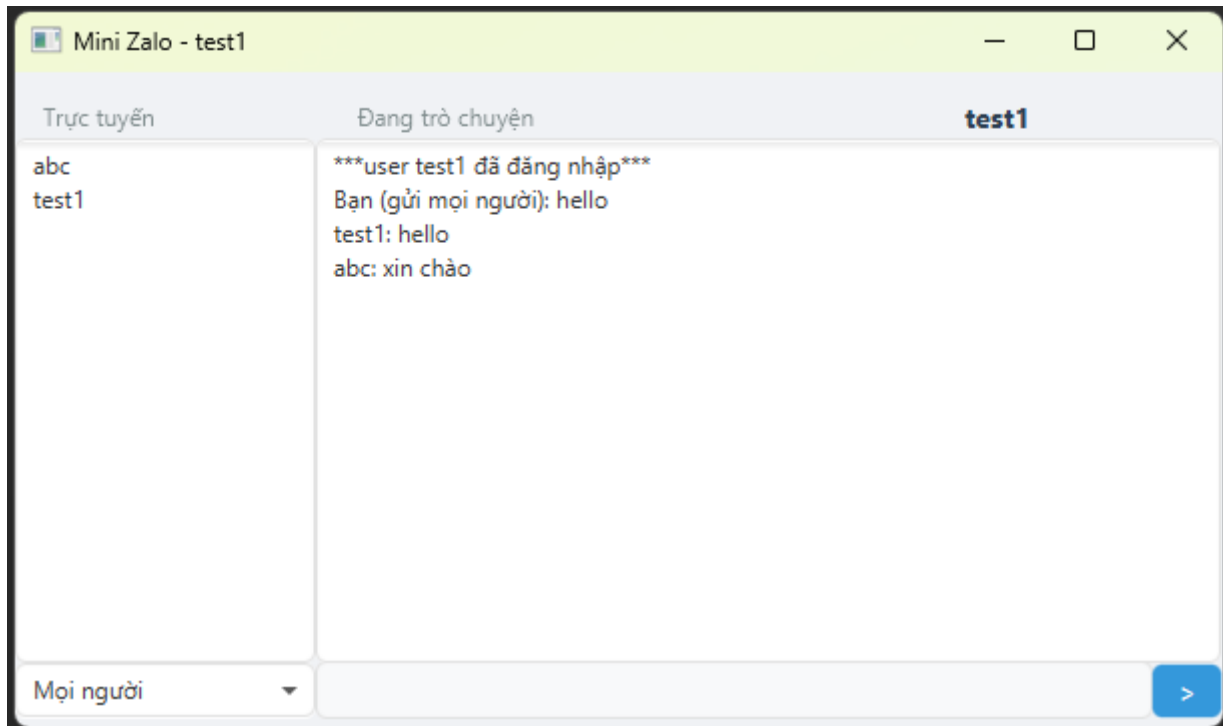


Hình 2.5.2.14. Gửi mọi người – người nhận

Khi người dùng chọn tên người nhận trong ComboBox và nhập thông điệp → click nút “>” để gửi cho người nhận

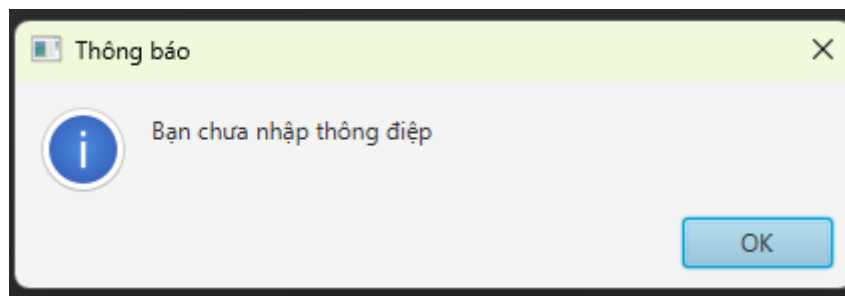


Hình 2.5.2.15. Gửi một người – người gửi



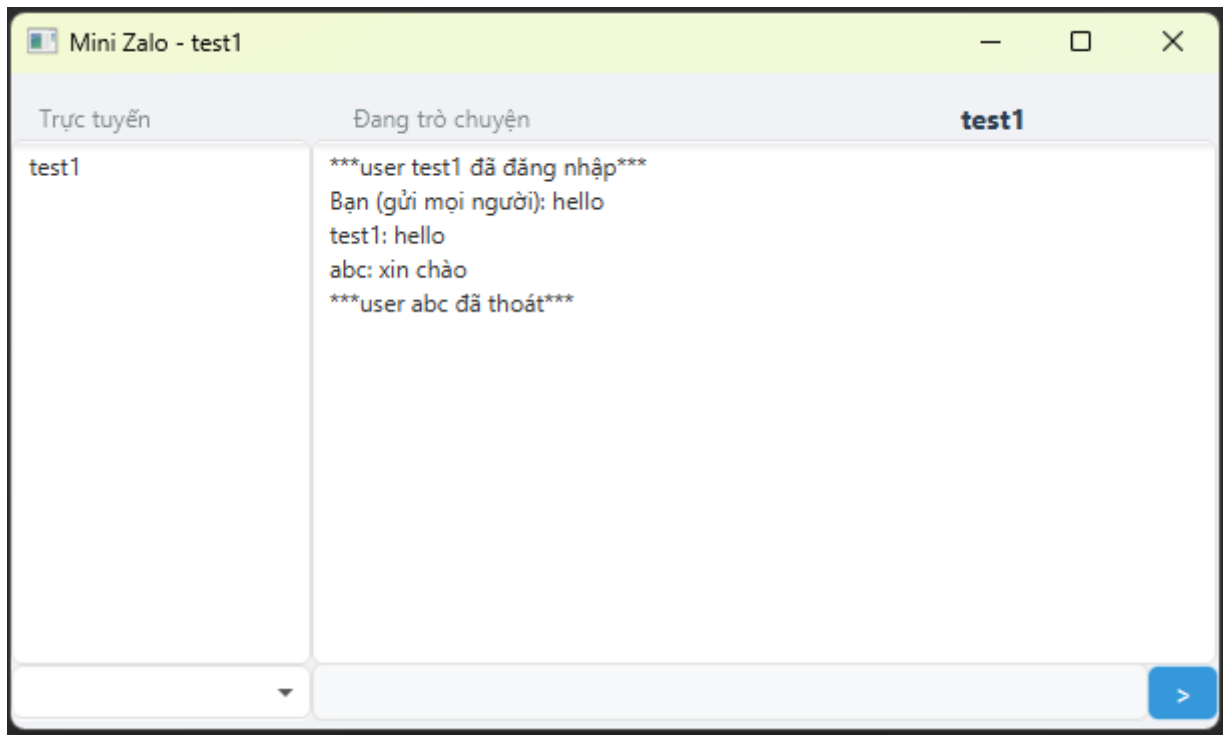
Hình 2.5.2.16. Gửi một người – người nhận

Khi người gửi chưa nhập thông điệp và click “>” → hiện thông báo



Hình 2.5.2.17. Thông báo chưa nhập thông điệp

Khi có người dùng đăng xuất → hệ thống sẽ thông báo cho những người khác



Hình 2.5.2.18. Người dùng đã đăng xuất

CHƯƠNG 3: TỔNG KẾT

3.1. Kết quả đạt được.

Sau quá trình phát triển và triển khai ứng dụng "Mini Zalo", ứng dụng đã đạt được một số kết quả đáng ghi nhận:

Tính năng nhắn tin: Ứng dụng đã hoàn thiện chức năng gửi và nhận tin nhắn văn bản, cho phép người dùng giao tiếp một cách nhanh chóng và hiệu quả. Người dùng có thể gửi tin nhắn đến tất cả người dùng trực tuyến.

Giao diện người dùng: Giao diện của ứng dụng được thiết kế đơn giản và trực quan, giúp người dùng dễ dàng làm quen và sử dụng.

Đăng ký và đăng nhập: Chức năng đăng ký tài khoản và đăng nhập hoạt động ổn định.

Hiện thị người dùng đang online: Người dùng dễ dàng nhận biết những ai đang kết nối.

3.2. Đánh giá ưu khuyết điểm.

- Ưu điểm:

Dễ sử dụng: Giao diện thân thiện và dễ sử dụng giúp người dùng nhanh chóng làm quen với ứng dụng mà không cần hướng dẫn phức tạp.

Kết nối dễ dàng: Người dùng dễ dàng kết nối với server.

Ứng dụng gọn nhẹ: Người dùng dễ dàng tải xuống và sử dụng.

- Khuyết điểm:

Giao diện còn sơ xài: Tuy giao diện dễ dùng nhưng còn sơ xài.

Khả năng mở rộng hạn chế: Một số tính năng nâng cao như video call hay gửi tin nhắn thoại vẫn chưa được triển khai, khiến ứng dụng chưa thể đáp ứng đầy đủ nhu cầu của người dùng.

3.3. Hướng phát triển và mở rộng đề tài.

Tích hợp tính năng gọi điện và video call: Việc bổ sung tính năng gọi điện và video call sẽ mang lại cho người dùng nhiều lựa chọn hơn trong việc giao tiếp. Điều này không chỉ giúp tăng cường trải nghiệm người dùng mà còn làm cho ứng dụng trở nên cạnh tranh hơn so với các ứng dụng nhắn tin khác trên thị trường.

Phát triển tính năng gửi tin nhắn thoại: Tính năng gửi tin nhắn thoại sẽ cho phép người dùng gửi thông điệp bằng giọng nói, giúp tiết kiệm thời gian và tạo sự thuận tiện trong giao tiếp.

Cải thiện bảo mật thông tin: Để đảm bảo an toàn cho dữ liệu người dùng, chúng tôi sẽ nâng cao các biện pháp bảo mật, bao gồm mã hóa dữ liệu và xác thực hai yếu tố. Việc này không chỉ giúp bảo vệ thông tin cá nhân mà còn tăng cường sự tin tưởng của người dùng đối với ứng dụng.

Nâng cao trải nghiệm người dùng: Cải thiện UX/UI để nâng cao trải nghiệm của người dùng

TÀI LIỆU THAM KHẢO

1. Horstmann, C. S., & Cornell, G. (2013). *Core Java Volume I: Fundamentals* (10th ed.). Prentice Hall.
2. Harold, E. (2004). *Java Network Programming* (3rd ed.). O'Reilly Media.
3. Sierra, K., & Bates, B. (2005). *Head First Java (2nd Edition)*. O'Reilly Media.
4. Gafter, N., & Zeldovich, N. (2014). *JavaFX 8: Introduction to JavaFX*. O'Reilly Media.