

**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**VÕ NGUYỄN GIA HUY**

**MSSV: 223940**

**TRẦN NGUYỄN CHÍ KHANH**

**MSSV: 220979**

**LỚP: DH22KPM02**

**TÊN ĐỀ TÀI**

**PHÁT TRIỂN ỨNG DỤNG NHẮN TIN**

**ĐỒ ÁN CƠ SỞ**

**Ngành: Kỹ thuật phần mềm**

**Mã số ngành: 7480103**

**03/2025**

**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**



**VÕ NGUYỄN GIA HUY**

**MSSV: 223940**

**TRẦN NGUYỄN CHÍ KHANH**

**MSSV: 220979**

**LỚP: DH22KPM02**

**TÊN ĐỀ TÀI**

**PHÁT TRIỂN ỨNG DỤNG NHẮN TIN**

**ĐỒ ÁN CƠ SỞ**

**Ngành: Kỹ Thuật Phần Mềm**

**Mã số ngành: 7480103**

**GIẢNG VIÊN HƯỚNG DẪN**

**ThS. NGUYỄN MỸ PHỤNG**

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến cô **ThS. Trần Mỹ Phụng**, giảng viên Khoa Công nghệ thông tin - trường Đại học Nam Cần Thơ đã hướng dẫn để chúng em hoàn thành đồ án này.

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến quý thầy cô trường Đại học Nam Cần Thơ, đặc biệt là các thầy cô trong Khoa Công nghệ Thông tin, những người đã tận tình giảng dạy, truyền đạt kiến thức và kinh nghiệm quý báu trong suốt quá trình học tập. Nhờ sự hướng dẫn tận tâm của thầy cô, chúng em đã có được nền tảng lý thuyết vững chắc, cũng như những kỹ năng cần thiết để hoàn thành đồ án này.

Bên cạnh đó, chúng em cũng xin chân thành cảm ơn sự quan tâm, hỗ trợ và tạo điều kiện thuận lợi từ nhà trường, giúp chúng em có môi trường học tập và nghiên cứu tốt nhất.

Mặc dù đã nỗ lực hết mình để hoàn thành đồ án, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Chúng em kính mong nhận được sự đóng góp ý kiến từ quý thầy cô để có thể hoàn thiện hơn nữa.

Chúng em xin chân thành cảm ơn!

*Cần Thơ, ngày tháng năm 2025*

**Người thực hiện 2**

**Người thực hiện 1**

**Trần Nguyễn Chí Khanh**

**Võ Nguyễn Gia Huy**

[illegible]

## Giảng Viên Hướng Dẫn

2

[illegible]

## Giảng Viên Phản Biện

3

## MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU .....	1
1.1. Tổng quan đề tài .....	1
1.2. Mục tiêu đề tài .....	1
1.3. Phạm vi đề tài .....	2
1.4. Phân chia công việc trong nhóm.....	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	3
2.1. Hệ quản trị cơ sở dữ liệu SQL Server: .....	3
2.1.1. SQL Server là gì .....	3
2.1.2. Các thao tác chính trong SQL Server.....	3
2.1.3. Tính năng nâng cao của SQL Server.....	3
2.2. Ngôn ngữ lập trình C#: .....	3
2.2.1. Giới thiệu .....	3
2.2.2. Đặc điểm và tính năng chính của C#.....	4
2.3. WPF: .....	4
2.3.1. Tổng quan.....	4
2.3.2. Chức năng.....	4
2.3.3. Ưu điểm .....	5
2.4. JWT:.....	5
2.4.1. JWT là gì .....	5
2.4.2. Cấu trúc .....	5
2.4.3. Cách thức hoạt động.....	6
2.4.3. Ưu điểm.....	6
2.5. ServiceStack API: .....	7
2.5.1. Giới thiệu.....	7
2.5.2. Các chức năng nổi bật .....	7
2.6. Mẫu kiến trúc MVVM: .....	8
2.6.1. Định nghĩa .....	8
2.6.2. Thành phần của MVVM.....	8
2.6.3. Đặc điểm và ưu điểm.....	8
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	9
3.1. Mô tả hệ thống .....	9
3.1.1. Giới thiệu hệ thống.....	9

3.1.2. Chức năng chính của hệ thống .....	9
3.1.3. Kiến trúc tổng quan .....	9
3.1.4. Quy trình hoạt động.....	9
3.1.5. Công nghệ sử dụng .....	9
3.2. Thiết kế hệ thống .....	10
3.2.1. Thiết kế cơ sở dữ liệu .....	10
3.2.2. Mô hình ERD .....	17
3.2.3. Thiết kế chức năng .....	17
3.2.4. Thiết kế kiến trúc hệ thống.....	17
3.2.5. Thiết kế bảo mật .....	17
CHƯƠNG 4: GIAO DIỆN HỆ THỐNG .....	18
4.1. Giao diện đăng nhập .....	18
4.2. Giao diện đăng ký.....	18
4.3. Giao diện chính.....	18
CHƯƠNG 5: KẾT LUẬN .....	19
5.1. Kết luận.....	19
5.2. Hạn chế .....	19
5.3. Hướng phát triển .....	19
TÀI LIỆU THAM KHẢO.....	20
PHỤ LỤC.....	21

## **DANH SÁCH BẢNG**

Bảng 3. 1 Bảng hóa đơn

3



## **DANH SÁCH HÌNH**

Hình 3. 1 Sơ đồ ERD tổng quát	3
Hình 4. 1 Giao diện đăng nhập	4

## **DANH MỤC TỪ VIẾT TẮT**

1. SQL: Structured Query Language
2. WWW: World Wide Web
3. ASP: Active Server Pages
4. CGI: Computer-Generated Imagery
5. CSDL: Cơ Sở Dữ Liệu

## CHƯƠNG 1: GIỚI THIỆU

### 1.1. Tổng quan đề tài

Trong bối cảnh cuộc cách mạng công nghệ thông tin đang bùng nổ, nhu cầu giao tiếp và trao đổi thông tin qua các ứng dụng nhắn tin ngày càng trở nên thiết yếu. Các ứng dụng nhắn tin như Telegram, WhatsApp, Zalo đã và đang khẳng định vai trò quan trọng trong việc kết nối con người, hỗ trợ giao tiếp nhanh chóng và bảo mật thông tin. Đồng thời, sự phát triển của mạng Internet di động và các thiết bị thông minh đã tạo điều kiện thuận lợi cho việc phổ biến và sử dụng rộng rãi các dịch vụ truyền thông số.

Đề tài “Phát triển ứng dụng nhắn tin” hướng đến việc xây dựng một ứng dụng với các chức năng cơ bản như gửi nhận tin nhắn, chia sẻ hình ảnh, video và tập tin, đồng thời tích hợp các tính năng nâng cao như mã hóa tin nhắn, bảo mật thông tin và hỗ trợ giao tiếp nhóm. Mục tiêu của đề tài không chỉ là cung cấp một công cụ giao tiếp hiệu quả mà còn đảm bảo sự an toàn và bảo mật của thông tin cá nhân người dùng. Qua đó, ứng dụng hướng đến việc cải thiện trải nghiệm người dùng bằng giao diện thân thiện, tốc độ truyền tin nhanh và khả năng hoạt động ổn.

Lý do lựa chọn đề tài này xuất phát từ xu hướng toàn cầu về chuyển đổi số và nhu cầu ngày càng tăng của người dùng đối với các giải pháp giao tiếp trực tuyến an toàn, tiện lợi. Việc nghiên cứu và phát triển một ứng dụng nhắn tin hiện đại sẽ góp phần giải quyết những hạn chế của các ứng dụng hiện có, đồng thời tạo ra giá trị gia tăng cho người dùng về mặt bảo mật và trải nghiệm sử dụng.

Bên cạnh đó, đề tài còn nhấn mạnh vai trò của các công nghệ mới như mã hóa đầu cuối để nâng cao chất lượng dịch vụ. Qua đó, ứng dụng không chỉ đáp ứng nhu cầu giao tiếp cơ bản mà còn mở rộng sang các tính năng hỗ trợ giao tiếp cộng đồng và tạo ra một hệ sinh thái truyền thông linh hoạt, an toàn và thông minh.

### 1.2. Mục tiêu đề tài

Mục tiêu của đề tài là phát triển một ứng dụng nhắn tin desktop an toàn, hiệu quả và thân thiện với người dùng, đáp ứng nhu cầu giao tiếp nội bộ của doanh nghiệp hoặc tổ chức. Cụ thể, đề tài đặt ra các mục tiêu sau:

- Xây dựng ứng dụng cho phép gửi và nhận tin nhắn, cùng với khả năng chia sẻ tập tin và hình ảnh.
- Đảm bảo tính bảo mật thông tin và xác thực người dùng một cách hiệu quả.
- Thiết kế giao diện trực quan, dễ sử dụng và hỗ trợ tương tác mượt mà.

- Đạt hiệu năng cao và hoạt động ổn định trong quá trình sử dụng.

### **1.3. Phạm vi đề tài**

Nghiên cứu và phát triển ứng dụng nhắn tin cho môi trường desktop, tập trung vào các chức năng cơ bản như gửi/nhận tin nhắn và chia sẻ tập tin, hình ảnh.

Xây dựng giao diện người dùng trực quan và thân thiện, đảm bảo quá trình tương tác mượt mà.

Áp dụng các giải pháp bảo mật cơ bản nhằm đảm bảo an toàn thông tin và xác thực người dùng.

Đánh giá hiệu năng và tính ổn định của ứng dụng qua các bài kiểm thử nội bộ.

### **1.4. Phân chia công việc trong nhóm**

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Hệ quản trị cơ sở dữ liệu SQL Server:

#### 2.1.1. SQL Server là gì

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ do Microsoft phát triển, được sử dụng rộng rãi trong các hệ thống doanh nghiệp, ứng dụng web và dịch vụ dữ liệu quy mô lớn. SQL Server cung cấp một nền tảng linh hoạt để lưu trữ, truy vấn và quản lý dữ liệu với hiệu suất cao và tính bảo mật mạnh mẽ.

#### 2.1.2. Các thao tác chính trong SQL Server

SQL Server sử dụng ngôn ngữ truy vấn T-SQL (Transact-SQL), mở rộng từ SQL tiêu chuẩn, để thực hiện các thao tác với dữ liệu. Một số lệnh quan trọng bao gồm:

- SELECT: Truy vấn dữ liệu từ bảng.
- INSERT: Thêm dữ liệu mới vào bảng.
- UPDATE: Cập nhật dữ liệu hiện có.
- DELETE: Xóa dữ liệu khỏi bảng.
- CREATE: Tạo bảng, khóa, quyền và các đối tượng khác trong cơ sở dữ liệu.
- ALTER: Thay đổi cấu trúc của bảng hoặc đối tượng trong cơ sở dữ liệu.
- DROP: Xóa bảng, khóa hoặc các đối tượng khác trong cơ sở dữ liệu.

#### 2.1.3. Tính năng nâng cao của SQL Server

Ngoài các lệnh cơ bản, SQL Server còn cung cấp nhiều tính năng mạnh mẽ như:

- Stored Procedures: Lưu trữ các truy vấn và logic xử lý dữ liệu để tối ưu hóa hiệu suất.
- Triggers – Tự động thực thi khi có sự kiện xảy ra trong cơ sở dữ liệu.
- Views – Tạo bảng ảo giúp tối ưu truy vấn và bảo mật dữ liệu.
- Transactions – Đảm bảo tính toàn vẹn của dữ liệu với các thao tác COMMIT và ROLLBACK.

### 2.2. Ngôn ngữ lập trình C#:

#### 2.2.1. Giới thiệu

C# là một ngôn ngữ lập trình hướng đối tượng hiện đại, được phát triển bởi Microsoft và giới thiệu lần đầu tiên như một phần của nền tảng .NET. Mục đích chính của C# là cung cấp một môi trường lập trình an toàn, hiệu quả và dễ bảo trì, phục vụ cho việc phát triển các ứng dụng từ desktop, web đến di động. Ngôn ngữ này được thiết kế với cú pháp

rõ ràng, dễ hiểu và tích hợp nhiều tính năng tiên tiến, nhằm giúp lập trình viên xây dựng các giải pháp phần mềm có khả năng mở rộng và hiệu năng cao.

### **2.2.2. Đặc điểm và tính năng chính của C#**

Lập trình hướng đối tượng: C# tuân thủ chặt chẽ các nguyên tắc của lập trình hướng đối tượng như đóng gói, kế thừa và đa hình, tạo điều kiện cho việc tái sử dụng mã nguồn và phát triển các ứng dụng phức tạp một cách hiệu quả.

Quản lý bộ nhớ tự động: Nhờ cơ chế thu gom rác (garbage collection), C# tự động giải phóng bộ nhớ không còn được sử dụng, giúp giảm thiểu nguy cơ rò rỉ bộ nhớ và tối ưu hóa hiệu năng hệ thống.

Hệ thống kiểu tĩnh (Static Type System): Cú pháp của C# được xây dựng trên hệ thống kiểu dữ liệu tĩnh, cho phép phát hiện lỗi trong quá trình biên dịch, từ đó tăng cường độ an toàn và độ tin cậy của mã nguồn.

Hỗ trợ lập trình bất đồng bộ: Các tính năng như từ khóa `async` và `await` giúp lập trình viên triển khai các tác vụ bất đồng bộ một cách tự nhiên, tối ưu hóa khả năng xử lý các tác vụ đòi hỏi I/O cao hoặc tính toán phức tạp.

Tích hợp sâu với nền tảng .NET: C# là ngôn ngữ chính của nền tảng .NET, cung cấp một thư viện phong phú các lớp và API hỗ trợ cho việc phát triển ứng dụng, từ đó tạo nên khả năng tương tác và mở rộng hệ thống dễ dàng.

## **2.3. WPF:**

### **2.3.1. Tổng quan**

Windows Presentation Foundation (WPF) là một framework của Microsoft, được thiết kế nhằm xây dựng giao diện người dùng (UI) cho các ứng dụng desktop chạy trên nền tảng Windows. Được phát triển dựa trên nền tảng .NET, WPF sử dụng XAML (Extensible Application Markup Language) để định nghĩa giao diện, giúp tách biệt rõ ràng giữa phần trình bày và logic nghiệp vụ. Điều này không chỉ nâng cao khả năng bảo trì và mở rộng của ứng dụng mà còn tạo điều kiện thuận lợi cho việc áp dụng các mô hình kiến trúc hiện đại như MVVM (Model-View-ViewModel).

### **2.3.2. Chức năng**

WPF cung cấp một loạt các chức năng tiên tiến nhằm hỗ trợ xây dựng giao diện người dùng hiện đại:

- Định dạng giao diện bằng XAML: Cho phép mô tả giao diện một cách trực quan và có cấu trúc, dễ dàng chỉnh sửa và bảo trì.

- Data Binding: Hỗ trợ liên kết dữ liệu giữa giao diện và các đối tượng dữ liệu, giúp tự động cập nhật UI khi dữ liệu thay đổi.
- Styles và Templates: Cho phép định nghĩa giao diện thống nhất cho các thành phần UI, từ đó tạo ra một trải nghiệm người dùng nhất quán.
- Hỗ trợ đồ họa vector: Cho phép tạo ra giao diện với chất lượng đồ họa cao, dễ dàng tùy chỉnh kích thước mà không bị mất độ sắc nét.
- Animation và Media Integration: Hỗ trợ tích hợp các hiệu ứng động và đa phương tiện, giúp giao diện trở nên sinh động và hấp dẫn.
- 3D Graphics: Cung cấp khả năng hiển thị đồ họa 3D, mở rộng phạm vi ứng dụng trong các lĩnh vực cần trực quan hóa dữ liệu và hiệu ứng hình ảnh phức tạp.
- Hardware Acceleration: Tận dụng sức mạnh của phần cứng để cải thiện hiệu năng hiển thị, đặc biệt hữu ích với các ứng dụng có yêu cầu đồ họa cao.

### 2.3.3. Ưu điểm

WPF được đánh giá cao nhờ vào một số ưu điểm nổi bật:

- Tách biệt rõ ràng giữa giao diện và logic, tạo điều kiện thuận lợi cho bảo trì và kiểm thử.
- Khả năng tạo ra giao diện đẹp mắt, linh hoạt và hiện đại.
- Mở rộng và tùy chỉnh dễ dàng thông qua styles, templates và control customization.
- Hiệu năng cao với khả năng xử lý đồ họa mượt mà, kể cả với các ứng dụng có yêu cầu đồ họa phức tạp.
- Hỗ trợ tích hợp đa dạng các tính năng UI, giúp phát triển các ứng dụng desktop tiên tiến.

## 2.4. JWT:

### 2.4.1. JWT là gì

JSON Web Token (JWT) là một tiêu chuẩn mở (RFC 7519) được sử dụng phổ biến trong việc xác thực và ủy quyền trong các hệ thống phân tán. JWT cho phép truyền tải thông tin một cách an toàn giữa các bên thông qua cơ chế ký số, đảm bảo tính toàn vẹn và xác thực của dữ liệu mà không cần lưu trữ trạng thái phiên trên máy chủ.

### 2.4.2. Cấu trúc

JWT được chia thành ba phần chính, mỗi phần được mã hóa theo định dạng Base64Url và phân cách bằng dấu chấm ('.').

- Header (Phần đầu): Chứa thông tin về loại token (thường là JWT) và thuật toán ký số được sử dụng (ví dụ: HMAC SHA256 hoặc

RSA). Phần này giúp người nhận xác định cách thức xử lý phần ký số của token.

- Payload (Phần nội dung): Bao gồm các "claim" (tuyên bố) chứa thông tin cần truyền tải, chẳng hạn như thông tin định danh người dùng, quyền truy cập, thời gian phát hành (iat) và thời gian hết hạn (exp). Payload là phần quan trọng quyết định nội dung thông tin mà token mang theo.
- Signature (Phần ký số): Được tạo ra bằng cách kết hợp header và payload với một khóa bí mật (hoặc khóa riêng đối với thuật toán bất đối xứng) thông qua thuật toán ký số đã được chỉ định. Phần ký số giúp đảm bảo rằng token không bị thay đổi trong quá trình truyền tải, từ đó tăng cường tính bảo mật.

#### 2.4.3. Cách thức hoạt động

Tạo token: Khi người dùng đăng nhập thành công, server sử dụng thông tin (như ID, vai trò) để tạo Payload, kết hợp với Header (chứa thuật toán) và mã hóa bằng khóa bí mật để tạo Signature. Ba phần này được nối lại thành JWT (Header.Payload.Signature) và gửi về client.

Lưu trữ và gửi: Client nhận JWT và lưu trữ (thường trong local storage hoặc cookie). Mỗi khi gửi yêu cầu tới server (như gọi API), client đính kèm JWT vào header (thường là Authorization: Bearer <token>).

Xác minh: Server nhận yêu cầu, lấy JWT từ header, dùng khóa bí mật để kiểm tra Signature. Nếu chữ ký hợp lệ và token chưa hết hạn, server chấp nhận yêu cầu và xử lý tiếp. Nếu không, yêu cầu bị từ chối.

#### 2.4.3. Ưu điểm

Bảo mật và tính toàn vẹn: Phần ký số của JWT đảm bảo rằng dữ liệu không bị chỉnh sửa hoặc giả mạo khi truyền qua các kênh không an toàn, qua đó giúp xác thực tính hợp lệ của thông tin người dùng.

Không trạng thái (Stateless): Vì JWT mang theo tất cả thông tin cần thiết cho quá trình xác thực, máy chủ không cần lưu trữ phiên làm việc của người dùng. Điều này giúp cải thiện khả năng mở rộng của hệ thống, đặc biệt là trong các môi trường có lưu lượng truy cập cao.

Truyền tải hiệu quả: Với cấu trúc nhỏ gọn và định dạng Base64Url, JWT dễ dàng được truyền qua các giao thức HTTP (ví dụ: trong header của yêu cầu HTTP), giúp tích hợp linh hoạt vào nhiều loại ứng dụng và dịch vụ.

Đa nền tảng và dễ tích hợp: Do JWT được xây dựng dựa trên định dạng JSON, nó tương thích với hầu hết các ngôn ngữ lập trình và nền tảng phát triển hiện đại, từ đó tạo điều kiện thuận lợi cho việc tích hợp trong các hệ thống phân tán và các ứng dụng đa dạng.



## **2.5. ServiceStack API:**

### **2.5.1. Giới thiệu**

ServiceStack là một framework mã nguồn mở, được xây dựng trên nền tảng .NET, nhằm mục đích tạo ra các dịch vụ web và API RESTful với hiệu năng cao và cấu trúc rõ ràng. Framework này hỗ trợ phát triển theo hướng “code-first”, cho phép lập trình viên định nghĩa các dịch vụ thông qua các lớp dữ liệu mạnh mẽ và rõ ràng, từ đó đơn giản hóa quá trình phát triển và bảo trì các ứng dụng phân tán.

ServiceStack cung cấp một tập hợp các công cụ toàn diện để xây dựng các API web theo chuẩn REST, hỗ trợ định dạng dữ liệu đa dạng như JSON, XML, CSV... Nhờ kiến trúc nhẹ nhàng, không phụ thuộc vào các cấu hình phức tạp, ServiceStack giúp giảm thiểu thời gian xử lý yêu cầu và tăng khả năng mở rộng của hệ thống. Framework này được thiết kế theo nguyên tắc không trạng thái (stateless), cho phép xử lý các yêu cầu từ người dùng một cách độc lập, từ đó cải thiện hiệu năng khi hoạt động trong môi trường có lưu lượng truy cập cao.

### **2.5.2. Các chức năng nổi bật**

**Xây dựng dịch vụ RESTful:** ServiceStack cho phép định nghĩa các API thông qua các lớp dữ liệu (DTOs) và các dịch vụ xử lý, đảm bảo tính nhất quán và dễ kiểm thử.

**Hỗ trợ định dạng dữ liệu đa dạng:** Các API được xây dựng với ServiceStack có thể trả về dữ liệu ở nhiều định dạng khác nhau như JSON, XML, CSV... mà không cần thay đổi cấu trúc dịch vụ.

**Tích hợp dependency injection:** Framework hỗ trợ tích hợp DI (Dependency Injection) một cách dễ dàng, giúp tăng cường khả năng mở rộng và kiểm thử của các thành phần trong ứng dụng.

**Hiệu năng cao:** Với kiến trúc nhẹ và tối ưu hóa xử lý yêu cầu, ServiceStack cho phép xây dựng các API có khả năng đáp ứng nhanh và xử lý đồng thời số lượng lớn các request.

**Bảo mật và quản lý lỗi:** ServiceStack cung cấp các cơ chế để xác thực, phân quyền truy cập và quản lý lỗi hiệu quả, đảm bảo an toàn cho dữ liệu và các dịch vụ cung cấp.

**Tính mở rộng và dễ bảo trì:** Cấu trúc rõ ràng, chia tách giữa lớp dữ liệu và logic nghiệp vụ giúp cho việc mở rộng, bảo trì cũng như kiểm thử trở nên đơn giản hơn, đáp ứng tốt yêu cầu của các hệ thống doanh nghiệp hiện đại.

## **2.6. Mẫu kiến trúc MVVM:**

### **2.6.1. Định nghĩa**

MVVM là một mô hình kiến trúc phần mềm được phát triển nhằm tách biệt rõ ràng giữa giao diện người dùng (View) và logic nghiệp vụ cùng với dữ liệu (Model), thông qua một lớp trung gian gọi là ViewModel. Mô hình này đặc biệt phổ biến trong phát triển ứng dụng desktop với các công nghệ như WPF (Windows Presentation Foundation), nơi nó giúp đơn giản hóa việc xây dựng và bảo trì giao diện phức tạp.

### **2.6.2. Thành phần của MVVM**

**Model:** Đại diện cho lớp dữ liệu và logic nghiệp vụ của ứng dụng. Model thường không chứa bất kỳ thông tin nào về giao diện, giúp đảm bảo tính độc lập và dễ kiểm thử của các chức năng cốt lõi.

**View:** Là giao diện người dùng, chịu trách nhiệm hiển thị dữ liệu và nhận tương tác từ người dùng. View được thiết kế để tối ưu hóa trải nghiệm trực quan và thường được định nghĩa bằng ngôn ngữ XAML trong WPF.

**ViewModel:** Đóng vai trò là cầu nối giữa Model và View. ViewModel xử lý logic trình bày, thực hiện binding dữ liệu từ Model đến View và quản lý các lệnh (commands) từ người dùng. Lớp này giúp giảm sự phụ thuộc giữa giao diện và logic nghiệp vụ, từ đó tạo điều kiện cho việc bảo trì và kiểm thử.

### **2.6.3. Đặc điểm và ưu điểm**

**Tách biệt quan tâm (Separation of Concerns):** MVVM chia tách rõ ràng giữa giao diện người dùng, logic trình bày và dữ liệu, giúp giảm sự phức tạp của ứng dụng và dễ dàng quản lý, bảo trì hệ thống.

**Data Binding mạnh mẽ:** Khả năng binding dữ liệu tự động giữa ViewModel và View cho phép giao diện luôn được cập nhật khi dữ liệu thay đổi, giảm thiểu sự can thiệp thủ công và tăng tính tương tác.

**Hỗ trợ kiểm thử:** Việc tách riêng logic trình bày trong ViewModel giúp dễ dàng viết các bài kiểm thử đơn vị (unit test) mà không cần phụ thuộc vào giao diện người dùng, tăng cường độ tin cậy của ứng dụng.

**Tăng cường khả năng tái sử dụng:** Các thành phần trong MVVM được thiết kế để có thể tái sử dụng ở nhiều phần của ứng dụng hoặc thậm chí trong các dự án khác, nhờ vào sự phân chia rõ ràng giữa giao diện và logic nghiệp vụ.

## CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1. Mô tả hệ thống

#### 3.1.1. Giới thiệu hệ thống

Hệ thống được xây dựng nhằm mục đích tạo ra một nền tảng giao tiếp trực tuyến hiệu quả, thân thiện, hỗ trợ người dùng trong việc trao đổi thông tin một cách nhanh chóng và an toàn.

Mục tiêu chính của hệ thống là tạo ra một ứng dụng cho phép nhắn tin và gọi video, giúp người dùng tương tác, kết nối với nhau một cách dễ dàng và tiện lợi, đồng thời có thể đảm bảo được tính bảo mật và riêng tư cho người dùng.

Đối tượng sử dụng hệ thống bao gồm: người dùng cá nhân, tổ chức, quản trị viên. Người dùng cá nhân có thể sử dụng ứng dụng để giao tiếp với bạn bè và gia đình, tổ chức có thể giao tiếp công việc với nhau trong khi quản trị viên có thể quản lý và giám sát hoạt động của hệ thống.

Hệ thống giải quyết các vấn đề như giao tiếp kém hiệu quả giữa người dùng với nhau, bảo mật thông tin của các cá nhân, tổ chức, quản lý nội dung không phù hợp.

#### 3.1.2. Chức năng chính của hệ thống

Bảng chức năng gay:

#### 3.1.3. Kiến trúc tổng quan

Được xây dựng dựa trên mô hình Client-Server, trong đó:

- Client: Là ứng dụng desktop được cài đặt trên máy tính cá nhân của người dùng. Ứng dụng này cung cấp giao diện trực quan để người dùng tương tác với hệ thống dễ dàng và thuận tiện.
- Server: Đóng vai trò trung tâm, xử lý các yêu cầu từ client, lưu trữ dữ liệu và quản lý kết nối giữa các client.

Hệ thống cũng được áp dụng kiến trúc microservices để tách biệt các chức năng chính với nhau như nhắn tin, gọi video và quản lý người dùng,...

#### 3.1.4. Quy trình hoạt động

Gay DFD

#### 3.1.5. Công nghệ sử dụng

Trong đồ án này, chúng em sử dụng ngôn ngữ lập trình C# để xây dựng toàn bộ hệ thống, đảm bảo tính nhất quán và hiệu quả trong việc phát triển ứng dụng. Về phần cơ sở dữ liệu, được quản lý bằng SQL Server, cho phép lưu trữ và truy xuất dữ liệu một cách ổn định và an toàn.

Để xây dựng API, chúng em áp dụng ServiceStack, giúp tạo ra các dịch vụ RESTful API linh hoạt và dễ bảo trì, đảm bảo giao tiếp mượt mà giữa các thành phần trong hệ thống. Phần giao diện người dùng được chúng em phát triển trên nền tảng của WPF với mong muốn mang lại trải nghiệm tương tác hiện đại và trực quan hơn cho các người dùng cuối.

Cuối cùng, hệ thống bảo mật được củng cố bằng JWT (JSON Web Token), đảm bảo cơ chế xác thực và phân quyền an toàn, giúp bảo vệ dữ liệu cũng như phiên làm việc của người dùng.

## 3.2. Thiết kế hệ thống

### 3.2.1. Thiết kế cơ sở dữ liệu

- Bảng Users

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	UserID	INT	PK	Mã người dùng
2	Username	VARCHAR(32)		Tên người dùng VD: huy1, khanh2,...
3	Email	VARCHAR(32)		Email người dùng
4	PhoneNumber	VARCHAR(16)		Số điện thoại của người dùng VD: +84852182935, +12125551234,...
5	Birthday	DATE		Ngày sinh của người dùng
6	HashedPassword	VARCHAR(60)		Mật khẩu lưu dưới dạng BCrypt
7	FirstName	VARCHAR(20)		Tên người dùng
8	LastName	VARCHAR(20)		Họ người dùng
9	Bio	VARCHAR(255)		Tiểu sử (mô tả)
10	Avatar	VARCHAR(128)		Đường dẫn của Avatar
11	LastLogin	DATETIME		Lần cuối đăng nhập
12	CreatedAt	DATETIME		Thời gian tạo tài khoản

*Bảng 3.1 Bảng Users*

- Bảng Users

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	UserID	INT	PK, FK	Mã người dùng sở hữu cài đặt này
2	StatusPrivacy	TINYINT	FK	Ai được phép thấy trạng thái hoạt động

3	BioPrivacy	TINYINT	FK	Ai được phép thấy tiểu sử
4	PhoneNumberPrivacy	TINYINT	FK	Ai được phép thấy số điện thoại
5	EmailPrivacy	TINYINT	FK	Ai được phép thấy email
6	BirthdayPrivacy	TINYINT	FK	Ai được phép thấy ngày sinh
7	CallPrivacy	TINYINT	FK	Ai được phép gọi
8	InviteGroupPrivacy	TINYINT	FK	Ai được phép mời bạn vào group
9	MessagePrivacy	TINYINT	FK	Ai được phép nhắn tin

*Bảng 3.2 Bảng UserSettings*

- Bảng UserSettings

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	PrivacyID	TINYINT	PK	Mã Privacy
2	PrivacyName	VARCHAR(16)		Tên của quyền riêng tư: +NOBODY: Chỉ mình mình xem +CONTACT: Chỉ liên hệ mới có thể xem +PUBLIC: Công khai ai cũng có thể xem

*Bảng 3.3 Bảng Privacy*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ContactID	INT	PK, FK	Người lưu liên hệ
2	UserID	INT	PK, FK	Liên hệ đã lưu
3	AddedAt	DATETIME		Ngày thêm của liên hệ
4	BlockedAt	DATETIME		Ngày chặn, nếu mà là null thì là đang không chặn

*Bảng 3.4 Bảng Contacts*

ST T	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ManagerID	INT	PK	Mã Managers

2	Username	VARCHAR(32)		Tên người quản lý VD: huy1, khanh2,...
3	FirstName	VARCHAR(20)		Tên người quản lý
4	LastName	VARCHAR(20)		Họ người quản lý
5	HashedPassword	VARCHAR(60)		Mật khẩu lưu dưới dạng BCrypt

*Bảng 3.5 Bảng Managers*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	BanID	INT	PK	Mã BannedAccounts
2	CreatorID	INT	FK	Mã người tạo lệnh khoá
3	BannedID	INT	FK	Mã người bị khoá
4	Reason	VARCHAR(MAX)		Lý do khoá
5	Expired	DATETIME		Ngày hết hạn
6	CreatedAt	DATETIME		Thời gian tạo lệnh khoá

*Bảng 3.6 Bảng BannedAccount*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ReportStatusID	TINYINT	PK	Mã trạng thái báo cáo
2	ReportStatusName	VARCHAR(16)		Tên của trạng thái báo cáo: +Pending: Đang chờ xem xét +Banned: Đã khoá +Reviewed: Đã xem xét nhưng không vi phạm

*Bảng 3.7 Bảng ReportStatus*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ReportID	INT	PK	Mã Reports
2	ReporterID	INT	FK	Người báo cáo
3	ReportedID	INT	FK	Người bị báo cáo
4	MessageID	INT	FK	Tin nhắn bị báo cáo, nếu null thì chỉ có báo cáo người dùng
5	ReportReason	VARCHAR(MAX)		Lý do báo cáo
6	ReportStatusID	TINYINT	FK	Trạng thái của báo cáo
7	DeletedAt	DATETIME		Thời gian tạo báo cáo

Bảng 3.8 Bảng Reports

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ConversationID	INT	PK	Mã Conversations
2	ConversationName	VARCHAR(32)		Tên cuộc trò chuyện VD: nhom1, thongbao1,... .Cho người dùng có thể tìm kiếm và tham gia nếu visibility bật là public
3	ConversationTitle	VARCHAR(32)		Tên hiển thị cho cuộc trò chuyện VD: Nhóm 1 làm bài, Nhóm xem sóc,...
4	CreatorID	INT	FK	Người tạo cuộc trò chuyện
5	ConversationTypeID	TINYINT	FK	Dạng của cuộc trò chuyện
6	GroupTypeID	TINYINT	FK	Dạng group có thể là PUBLIC cho mọi người tham gia bằng tên cuộc trò chuyện, INVITE chỉ người được mời,...
7	CreatedAt	DATETIME		Thời gian tạo cuộc trò chuyện

Bảng 3.9 Bảng Conversations

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ConversationTypeID	TINYINT	PK	Mã loại cuộc trò chuyện
2	ConversationTypeName	VARCHAR(16)		Dạng cuộc trò chuyện: +CHAT: cuộc trò chuyện bình thường người dùng với người dùng +GROUP: trò chuyện nhóm

Bảng 3.10 Bảng ConversationType

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
-----	------------	--------------	------	-------

1	GroupTypeID	TINYINT	PK	Mã GroupType
2	GroupName	VARCHAR(16)		Dạng Group: +PUBLIC: mọi người có thể tham gia thông qua tìm kiếm bằng tên cuộc trò chuyện, hoặc được mời vào +INVITE: chỉ được mời +PRIVATE: chỉ người có quyền OWNER hoặc ADMIN thì mới được thêm vào

*Bảng 3.11 Bảng GroupType*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	DeletedConversationID	INT	PK	Mã cuộc trò chuyện bị xóa
2	ConversationID	INT	FK	Mã cuộc trò chuyện cần xóa
3	UserID	INT	FK	Mã người dùng đã xóa cuộc trò chuyện
4	DeletedAt	DATETIME		Thời gian xóa cuộc trò chuyện

*Bảng 3.12 Bảng DeletedConversations*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ParticipantID	INT	PK	Mã Participants
2	ConversationID	INT	FK	Mã cuộc trò chuyện mà người dùng tham gia
3	UserID	INT	FK	Mã người dùng tham gia cuộc trò chuyện
4	NickName	VARCHAR (32)		Biệt danh mà người dùng sử dụng trong cuộc trò chuyện, nếu không có thì null
5	ConversationRoleID	TINYINT	FK	Mã quyền của người dùng trong cuộc trò chuyện
6	CreatedAt	DATETIME		Thời gian tham gia cuộc trò chuyện



7	DeletedAt	DATETIME		Ngày xoá, nếu null thì đang không xoá
---	-----------	----------	--	---------------------------------------

*Bảng 3.13 Bảng Participants*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	ConversationRoleID	TINYINT	PK	Mã vai trò của cuộc trò chuyện
2	ConversationRoleName	VARCHAR(16)		Tên quyền: +OWNER: Quản trị thêm, xoá người dùng và staff +STAFF: Đồng quản trị thêm, xoá người dùng +USER: Người dùng gửi, xoá, chỉnh sửa tin nhắn.

*Bảng 3.14 Bảng ConversationRoles*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	MessageID	INT	PK	Mã Messages
2	ConversationID	INT	FK	Mã cuộc trò chuyện mà tin nhắn được gửi tới
3	SenderID	INT	FK	Mã người gửi
4	Content	VARCHAR(MAX)		Tin nhắn văn bản hoặc nội dung khác
5	MessageType	TINYINT	FK	Dạng tin nhắn
6	CreatedAt	DATETIME		Thời gian tạo tin nhắn

*Bảng 3.15 Bảng Messages*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	MessageTypeID	TINYINT	PK	Mã MessageType
2	MessageType_name	VARCHAR(16)		Tên dạng tin nhắn: +TEXT: Tin nhắn văn bản +CALL: Cuộc gọi +AUDIO: Âm thanh

*Bảng 3.16 Bảng MessageType*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	AttachmentID	INT	PK	Mã tệp đính kèm

2	MessageID	INT	FK	Mã tin nhắn có tệp đính kèm
3	AttachmentTypeID	TINYINT		Dạng của tệp đính kèm
4	ThumbnailURL	VARCHAR(128)		Hình nhỏ hiện thị mẫu của tệp, video, ảnh,...
5	FileURL	VARCHAR(128)		Địa chỉ tệp, video, ảnh,...
6	CreatedAt	DATETIME		Thời gian tạo tệp
7	DeletedAt	DATETIME		Thời gian xóa, nếu null thì không xóa

*Bảng 3.17 Bảng Attachments*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	AttachmentTypeID	TINYINT	PK	Mã loại tệp đính kèm
2	AttachmentTypeName	VARCHAR(16)		Tên dạng tệp đính kèm: +PHOTO: Tệp ảnh +VIDEO: Tệp video +Các tệp khác nếu có

*Bảng 3.18 Bảng AttachmentType*

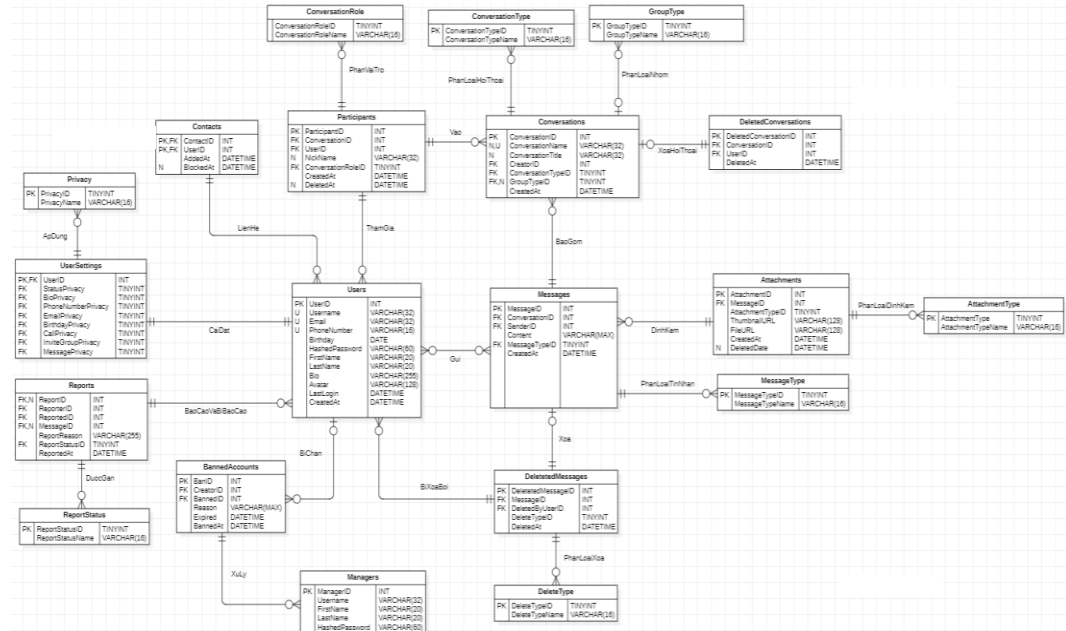
STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	DeletedMessageID	INT	PK	Mã tin nhắn bị xóa
2	MessageID	INT	FK	Mã tin nhắn cần xóa
3	DeletedByUserID	INT	FK	Xóa bởi người dùng
4	DeleteTypeID	TINYINT	FK	Dạng xóa
5	DeletedAt	DATETIME		Thời gian xóa tin nhắn

*Bảng 3.19 Bảng DeletedMessages*

STT	THUỘC TÍNH	KIỂU DỮ LIỆU	KHOÁ	MÔ TẢ
1	DeleteTypeID	TINYINT	PK	Mã dạng xóa
2	DeleteTypeName	VARCHAR(16)		Dạng xóa: +ALL: Xóa với tất cả mọi người +ONLYME: Xóa chỉ với người dùng đã xóa

*Bảng 3.20 Bảng DeleteType*

### 3.2.2. Mô hình ERD



Hình 3.1 Sơ đồ ERD tổng quát

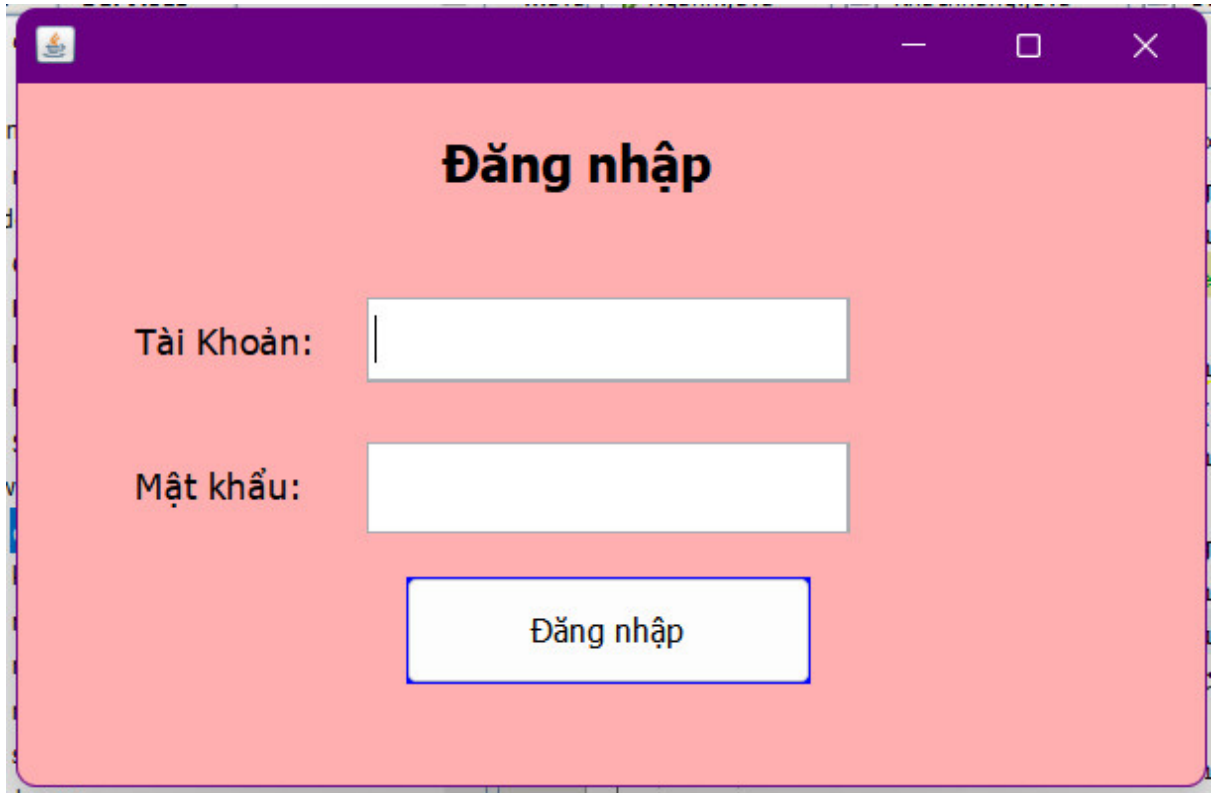
### 3.2.3. Thiết kế chức năng

### 3.2.4. Thiết kế kiến trúc hệ thống

### 3.2.5. Thiết kế bảo mật

## CHƯƠNG 4: GIAO DIỆN HỆ THỐNG

### 4.1. Giao diện đăng nhập



The image shows a login window with a purple title bar and a light pink background. The title 'Đăng nhập' is centered at the top. Below it, there are two input fields. The first is labeled 'Tài Khoản:' and the second is labeled 'Mật khẩu:'. Both fields are empty. Below the password field is a button labeled 'Đăng nhập'.

*Hình 4. 1 Giao diện đăng nhập*

### 4.2. Giao diện đăng ký

### 4.3. Giao diện chính

## **CHƯƠNG 5: KẾT LUẬN**

### **5.1. Kết luận**

### **5.2. Hạn chế**

### **5.3. Hướng phát triển**

## TÀI LIỆU THAM KHẢO

- [1]. Huỳnh Võ Hữu Trí (2023), *Lập trình Python*, nxb Cần Thơ. 100 trang.

## **PHỤ LỤC**

Trình bày:

- Hướng dẫn cài đặt triển khai CT
- HD sử dụng CT
- .....