TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng ứng dụng trò chuyện nhắn tin an toàn

Lê Đình Tài

tai.ld173352@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Lê Đức Trung _____

Bộ môn: Công nghệ phần mềm
Viện: Công nghệ thông tin – Truyền thông

HÀ NỘI, 6/2021

Lời cam kết

Họ và tên sinh viên: Lê Đình Tài

Điện thoại liên lạc: 0922100399 Email: tai.ld173352@sis.hust.edu.vn

Lớp: CNTT09 Hệ đào tạo: Cử nhân kỹ thuật

Tôi Lê Đình Tài cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của ThS.Lê Đức Trung. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 16 tháng 6 năm 2021

Tác giả ĐATN

Họ và tên sinh viên

Lê Đình Tài

Lời cảm ơn

Để hoàn thành đồ án tốt nghiệp này, em xin chân thành gửi lời cảm ơn tới các thầy giáo, cô giáo trường Đại học Bách khoa Hà Nội nói chung, viện Công nghệ thông tin và truyền thông, bộ môn Công nghệ phần mềm, đã dạy em những kiến thức, những kinh nghiệm quý báu trong suốt quá trình học tập tại trường. Tiếp đến em xin chân thành cảm ơn ThS. Lê Đức Trung-Giảng viên bộ môn Công nghệ phần mềm, viện Công nghệ thông tin và truyền thông, trường Đại học Bách khoa Hà Nội đã tận tình chỉ bảo, hướng dẫn cũng như truyền đạt những kinh nghiệm quý báu giúp em có thể hoàn thành được đồ án tốt nghiệp. Kế đến, em xin gửi lời cảm ơn chân thành tới các anh chị trong tập thể công ty TNHH Công nghệ và Truyền Thông MDC đã tạo điều kiện và dạy em những kinh nghiệm, kiến thức quý báu trong quá trình thực tập và có thể hoàn thành được đồ án tốt nghiệp. Cuối cùng là lời cảm ơn chân thành tới gia đình, người thân, bạn bè đã luôn bên cạnh động viên, góp ý cho em những ý tưởng tuyệt vời, góp phần vào việc hoàn thành đồ án của em. Nhưng do thời gian có hạn, kiến thức còn ít, và tình hình dịch bệnh Covid đang căng thẳng, tất cả điều đó nó ảnh hưởng, khiến cho đồ án này không tránh khỏi những thiếu sót, vậy nên em rất mong được sự đóng góp ý kiến của các thầy cô để em có thể hoàn thiện và rút ra được thêm kinh nghiệm. Em xin chân thành cảm ơn.

Tóm tắt

Hiện nay Internet ngày càng phát triển, đồng thời nhu cầu của con người về việc trao đổi thông tin trên Internet ngày càng tăng. Nhưng bên cạnh đó là những mỗi hiểm nguy do Internet mang lại, đó là tin tặc, chúng luôn lợi dụng những kẽ hở trong môi trường Internet để thu thập thông tin của mọi người một cách trái phép nhằm những mục đích xấu. Điều đó dẫn tới việc cần có một ứng dụng giúp con người có thể trao đổi trên Internet mà lại bảo việc an toàn khi trao đổi thông tin.

Ý tưởng về việc xây dựng một ứng dụng trò chuyện nhắn tin an toàn giúp mọi người có thể trao đổi thông tin qua Internet mà vẫn đảm bảo an toàn thông tin cá nhân. Việc xây dựng ứng dụng bảo mật này tương đối khó, trên thực tế hiện nay có rất nhiều ứng dụng đáp ứng được yêu cầu này. Các ứng dụng này hầu như đều sử dụng cơ chế mã hóa end to end, và đây cũng là hướng tiếp cận em đã chọn.

Em sẽ sử dụng cơ chế mã hóa end to end để mã hóa tin nhắn truyền đi, tuy vẫn truyền tin nhắn thông qua server nhưng nhà phát triển ứng dụng vẫn không thể biết thông tin truyền đi là gì.

Với ĐATN này em sẽ xây dựng một ứng dụng trò chuyện nhắn tin an toàn, với mục tiêu bảo vệ người dùng khỏi những kẻ xấu muốn ăn cắp thông tin người dùng.

Mục lục

Danh mục hình vẽ

Danh mục bảng

Danh mục công thức

Danh mục các từ viết tắt

API	Application Programming Interface Giao diện lập trình ứng dụng
ĐATN	Đồ án tốt nghiệp
AES	Advanced Encryption Standard Tiêu chuẩn mã hóa nâng cao
RSA	Rivest–Shamir–Adleman Tên ba nhà khoa học đã tạo ra thuật toán
MVVM	Model – View - ViewModel
XML	Extensible Markup Language
HTTP	Hypertext Markup Language
URL	Uniform Resource Locator
RAM	Random-Access Memory
ROM	Read Only Memory
JSON	JavaScript Object Notation Một dạng dữ liệu

Danh mục thuật ngữ

Client	Máy khách
Server	Máy chủ
Key	Khóa
Public key	Khóa công khai
Private key	Khóa bí mật
Secret Key	Khóa bảo mật
Database	Cơ sở dữ liệu

Chương 1 Giới thiệu đề tài

Chương 1 là phần em sẽ giới thiệu về đề tài của mình, về vấn đề trong thực tiễn dẫn tới ĐATN, tiếp đó và mục tiêu, phạm vi đề tài , về những giải pháp em sẽ đặt ra về hoàn thành các mục tiêu, và cuối cùng là sơ lược về bố cục của báo cáo

1.1 Đặt vấn đề

Ngày xưa, khi chưa có sự xuất hiện của Internet, con người liên lạc xa với nhau qua thư hay qua người trung gian truyền đi, điều này dẫn đến việc thông tin không được đảm bảo sẽ đến được nơi cần đến, hay sự bảo mật khi truyền đi.

Hiện nay với sự phát triển vượt bậc của thời đại công nghệ số, Internet ngày càng phủ rộng khắp, điện thoại thông minh hầu như mọi người ai cũng có và gần như đều có thể kết nối được Internet. Việc con người truyền các thông tin với nhau qua Internet ngày càng dễ dàng, nhưng bên cạnh đó việc bảo vệ an toàn thông tin truyền đi cũng là một vấn đề quan trọng.

Bởi lý do trên, ĐATN này em sẽ xây dựng ứng dụng trò chuyện nhắn tin an toàn.Ứng dụng sẽ đáp ứng được các nhu cầu cơ bản của mọi người như gửi văn bản, hình ảnh, giọng nói hay gọi video,.. và có thể làm quen với bạn bè mới.

1.2 Mục tiêu và phạm vi đề tài

Với những vấn đề đã đặt ra ở phần 1.1, mục tiêu đặt ra là phát triển được ứng dụng trò chuyện nhắn tin một cách an toàn. Trong đó, ứng dụng được xây dựng trên hệ điều hành android và cần đáp ứng được các yêu cầu cơ bản: (i) các chức năng cơ bản hoạt động tốt, (ii) ứng dụng phản hồi nhanh trong thời gian thực, (iii) an toàn khi trao đổi thông tin.

1.3 Định hướng giải pháp

Ứng dụng được xây dựng trên hệ điều hành android, với mục tiêu đã đề ra ở phần 1.2, ĐATN đã khảo sát nhu cầu của người dùng cũng như các sản phẩm liên quan trên thị trường. Dựa vào đó, ĐATN sẽ xây dựng những chức năng phù hợp với nhu cầu người dùng và công nghệ phù hợp để xây dựng.

ĐATN sử dụng (i) MySQL làm cơ sở dữ liệu, (ii) Node js để xây dựng server, (iii) Kotlin để xây dựng ứng dụng phía client, (iv) RESTful API và websocket để giao tiếp client-server

và (v) sử dụng thật toán RSA và AES để mã hóa các thông tin người dùng. Ngoài ra em còn sử dụng (i) Github để quản lý mã nguồn một cách an toàn và hiểu quả,.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau:

Chương 2 em sẽ trình bày về sự khảo sát của em về các ứng dụng trò chuyện nhắn tin an toàn hiện nay và nhu cầu của người dùng. Sau đó em sẽ phân tích tạo ra các chức năng cụ thể để phát triển thông qua các biểu đồ usecase.

Tiếp đến chương 3 em sẽ giới thiệu qua về các công nghệ chính em sử dụng để hoàn thành ứng dụng, và những đặc điểm đã khiến em dùng công nghệ đó để hoàn thành ứng dụng.

Và đến chương 4 em sẽ trình bày về cách triển khai và phát triển hệ thống ứng ứng dụng. Các thiết kế kiến trúc, thiết kế database và giao diện cho ứng dụng,… sau đó thể kiểm thử và triển khai cho người dùng.

Sau đó chương 5 em sẽ trình bày về các giải pháp và đóng góp nổi bật của mình. Các đóng gióp này là những vấn đề bản thân em gặp phải và em đã giải quyết được nó.

Cuối cùng, chương 6 em sẽ trình bày về thành quả đã đạt được, rút ra kinh kiệm cho bản thân và đưa ra hướng phát triển sau này.

Sau đây em sẽ đi vào chi tiết của từng phần.

Chương 2 Khảo sát và phân tích yêu cầu

Chương 2 là phần trình bày về các khảo sát về thực tết các ứng dụng tương tự, của người dùng, từ đó phân tích ra các yêu cầu cụ thể qua các usecase và đặc tả các chức năng, ngoài ra còn đưa ra các yêu cầu về phi chức năng.

2.1 Khảo sát hiện trạng

Hiện nay có rất nhiều ứng dụng nhắn tin bảo mật với các tính năng đa dạng và nổi trội. Phải kể đến như Zalo, Telegram, … Và sau đây em xin thống kê một số thông tin nổi bật của các ứng dụng này trên nền tảng android tại **Bảng 1**.

Bảng 1 Khảo sát hiện trạng Zalo và Telegram

Đặc điểm	Zalo	Telegram
Mã hóa end to end trong nhắn tin với bạn bè	Không	Có
Mã hóa end to end khi nhắn tin nhóm	Không	Không
Số ngôn ngữ hỗ trợ	3	18
Số lượt tải trên CHPlay	Hơn 100 triệu	Hơn 500 triệu
Điểm đánh giá	4.3	4.3

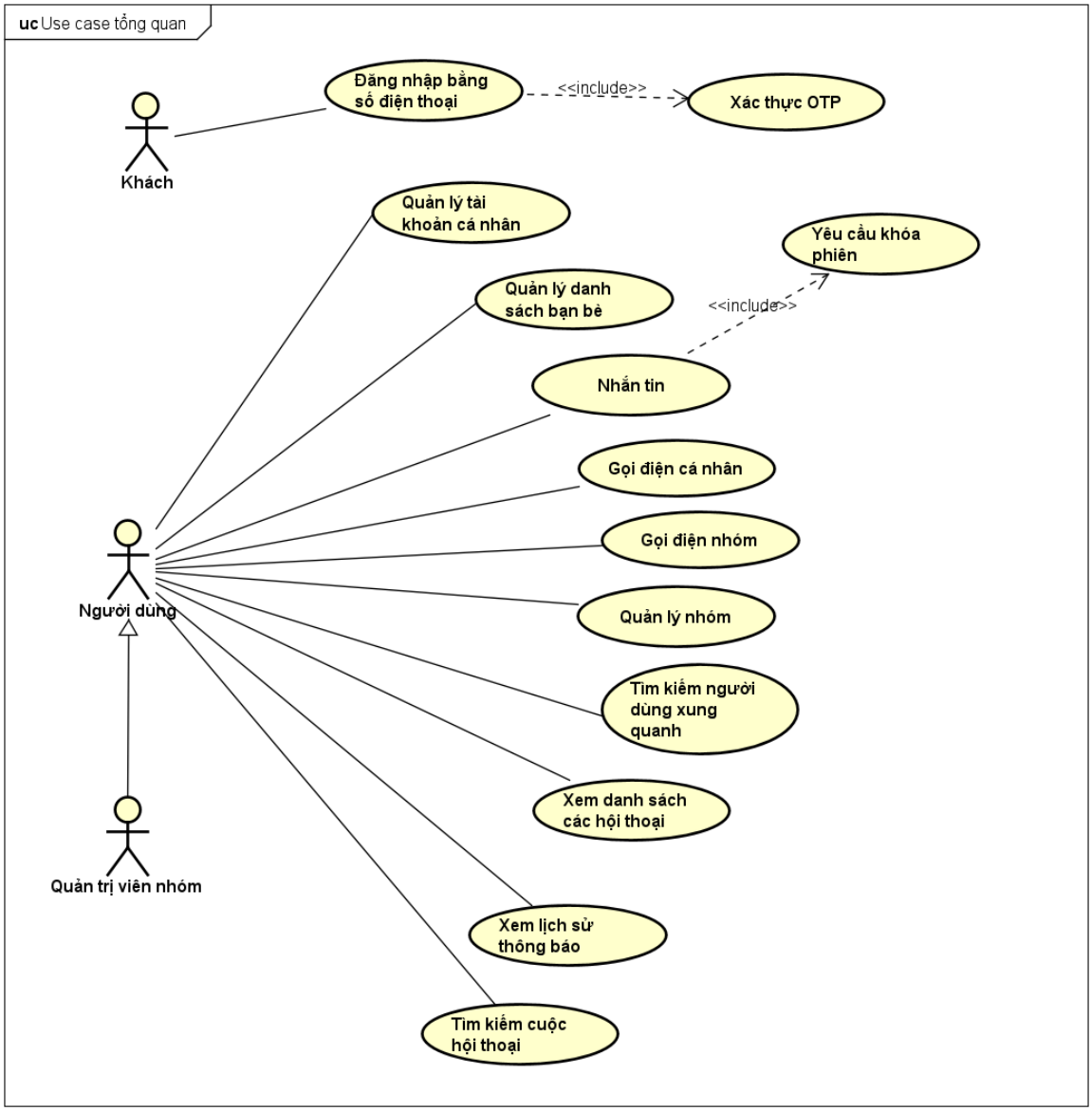
Zalo là một mạng xã hội được sử dụng tại các nước Việt Nam, Hoa Kỳ, Myanmar, Nhật Bản, Đài Loan, Hàn Quốc, Malaysia, Ả Rập Xê Út, Angola, Sri Lanka, Cộng hòa Séc, Nga. Zalo hiện đang là ứng dụng nhắn tin được ưa thích nhất hiện nay tại Việt Nam. Zalo có giao diện đẹp đẽ, thân thiện với người dùng.

Telegram là một ứng dụng nhắn tin an toàn bảo mật, được sử dụng hầu như tại tất cả các nước trên thế giới. Nó nổi trội trong việc bảo vệ an toàn thông tin khách hàng, do sử dụng cơ chế end to end nên ngay cả nhà sản xuất cũng không thể xem được thông tin người dùng đã liên lạc.

Dựa trên sự tìm hiểu và khảo sát về hai ứng dụng này, đồng thời cũng theo nhu cầu của người dùng. ĐATN của em sẽ hướng đến xây dựng ứng dụng trò chuyện nhắn tin an toàn, sử dụng cớ chế end to end và bảo mật dù có nhắn tin với bạn bè hay trong nhóm, đồng thời cũng học hỏi về hai ứng dụng trên để tạo nên một ứng dụng với giao diện thân thiện với người dùng.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quan



Hình 1 Use case tổng quan

Hình 1 mô tả use case tổng quan của hệ thống.Ứng dụng nhắn tin an toàn chỉ tập trung vào tạo lập một phiên hội thoại an toàn, đơn giản nên chỉ gồm hai tác nhân, đó là Khách và Người dùng.

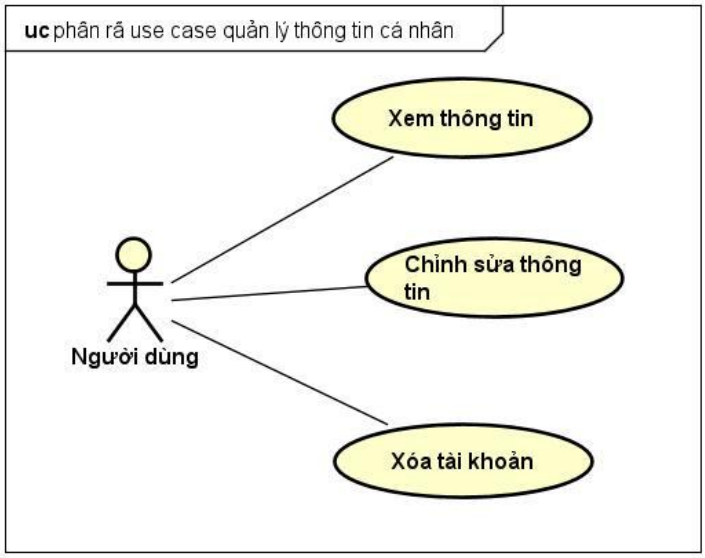
Khách sau khi đăng nhập và được xác thực sẽ trở thành một Người dùng và được sử dụng các chức năng của hệ thống.

Người dùng sẽ được sử dụng toàn bộ các chức năng chính của hệ thống.

Quản trị viên nhóm là người dùng, nhưng trong trong nhóm sẽ được sử dụng các chức năng quản trị viên.

2.2.2 Biểu đồ use case phân rã

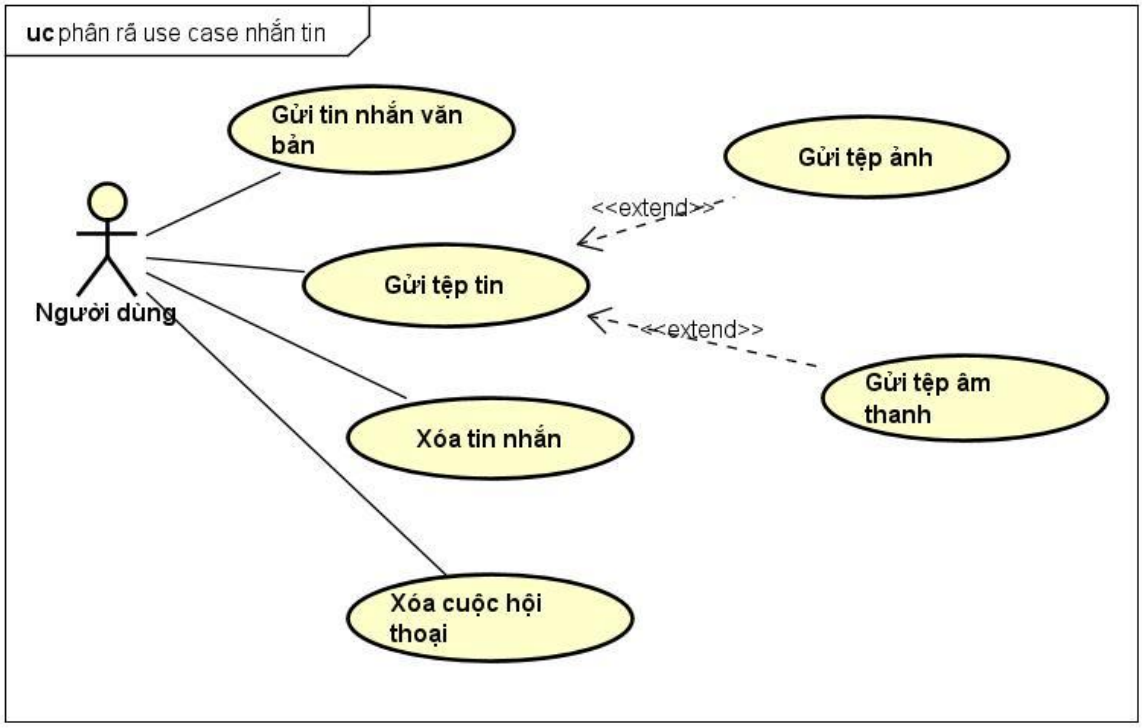
2.2.2.1 Phân rã use case quản lý thông tin cá nhân



Hình 2 Phân rã use case quản lý thông tin cá nhân

Hình 2 mô tả nhóm chức năng quản lý thông tin các nhân của người dùng, gồm có xem thông tin cá nhân, chỉnh sửa và xóa tài khoản của bản thân.

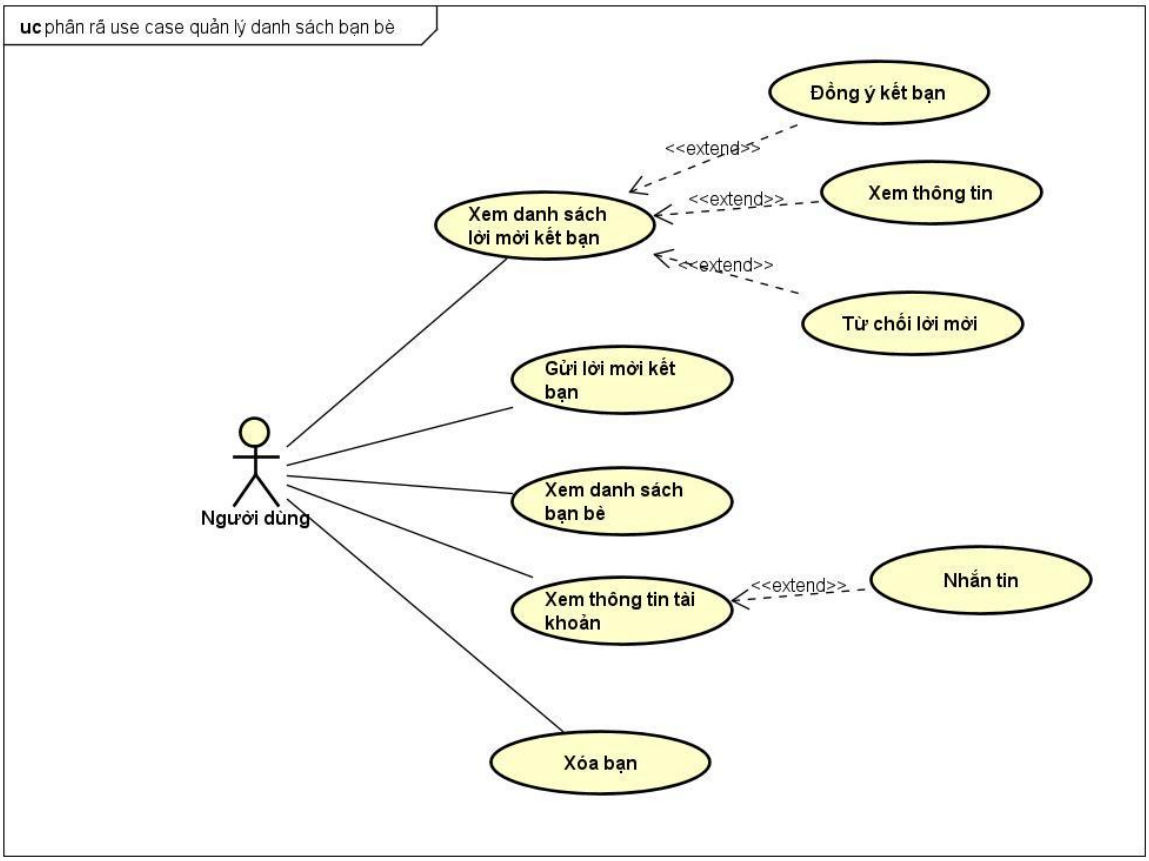
2.2.2.2 Phân rã use case nhắn tin



Hình 3 Phân rã use case nhắn tin

Hình 3 mô tả use nhóm chức năng nhắn tin, người dùng có thể gọi điện, nhắn tin bằng văn bản hay gửi các tệp tin, hoặc là xóa cuộc hội thoại đi.

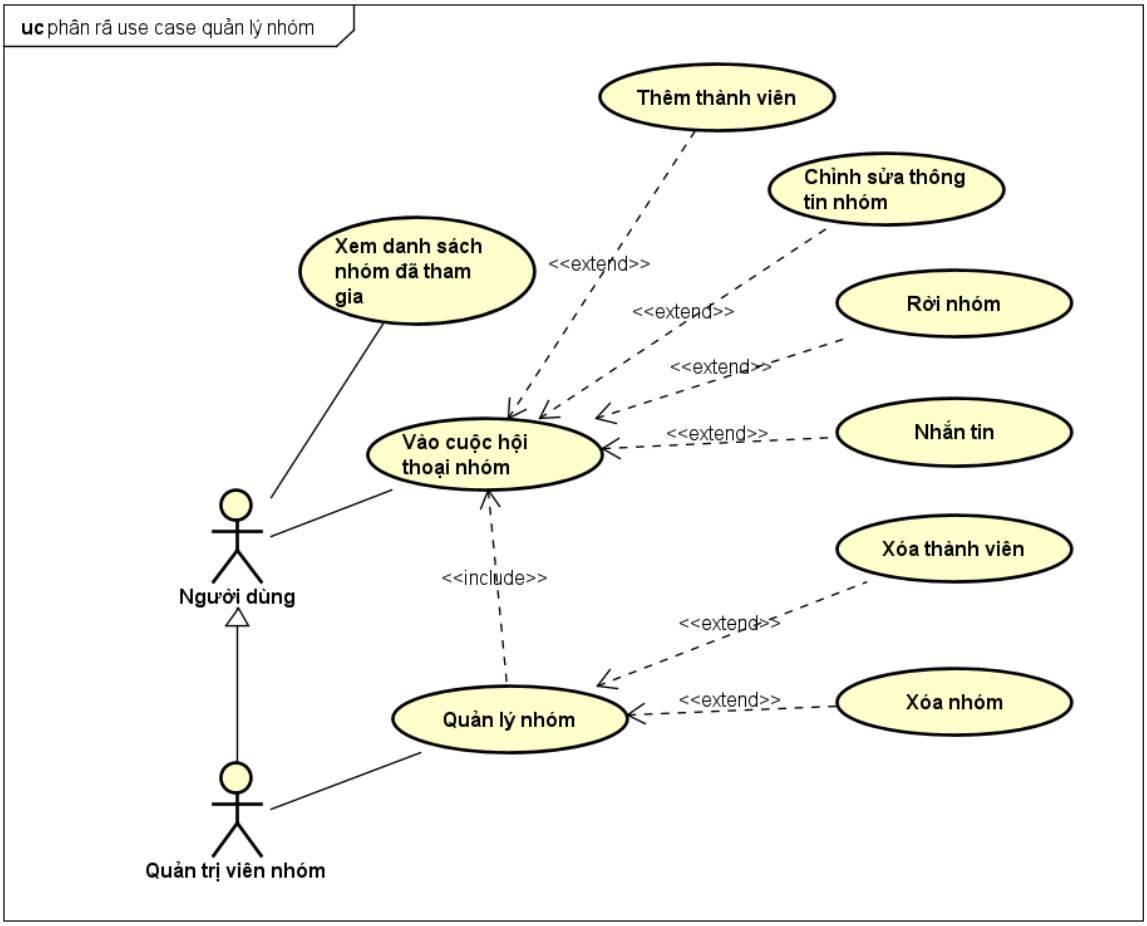
2.2.2.3 Phân rã use case quản lý danh sách bạn bè



Hình 4 Phân rã use case quản lý danh sách bạn bè

Hình 4 mô tả nhóm chức năng quản lý danh sách bạn bè, người dùng có thể xem danh sách bạn bè hiện tại,xem thông tin các nhân và đi đến chức năng nhắn tin, đồng thời cũng có thể xem danh sách các lời mời kết bạn được gửi đến để thêm bạn mới hay từ chối lời mời đó.

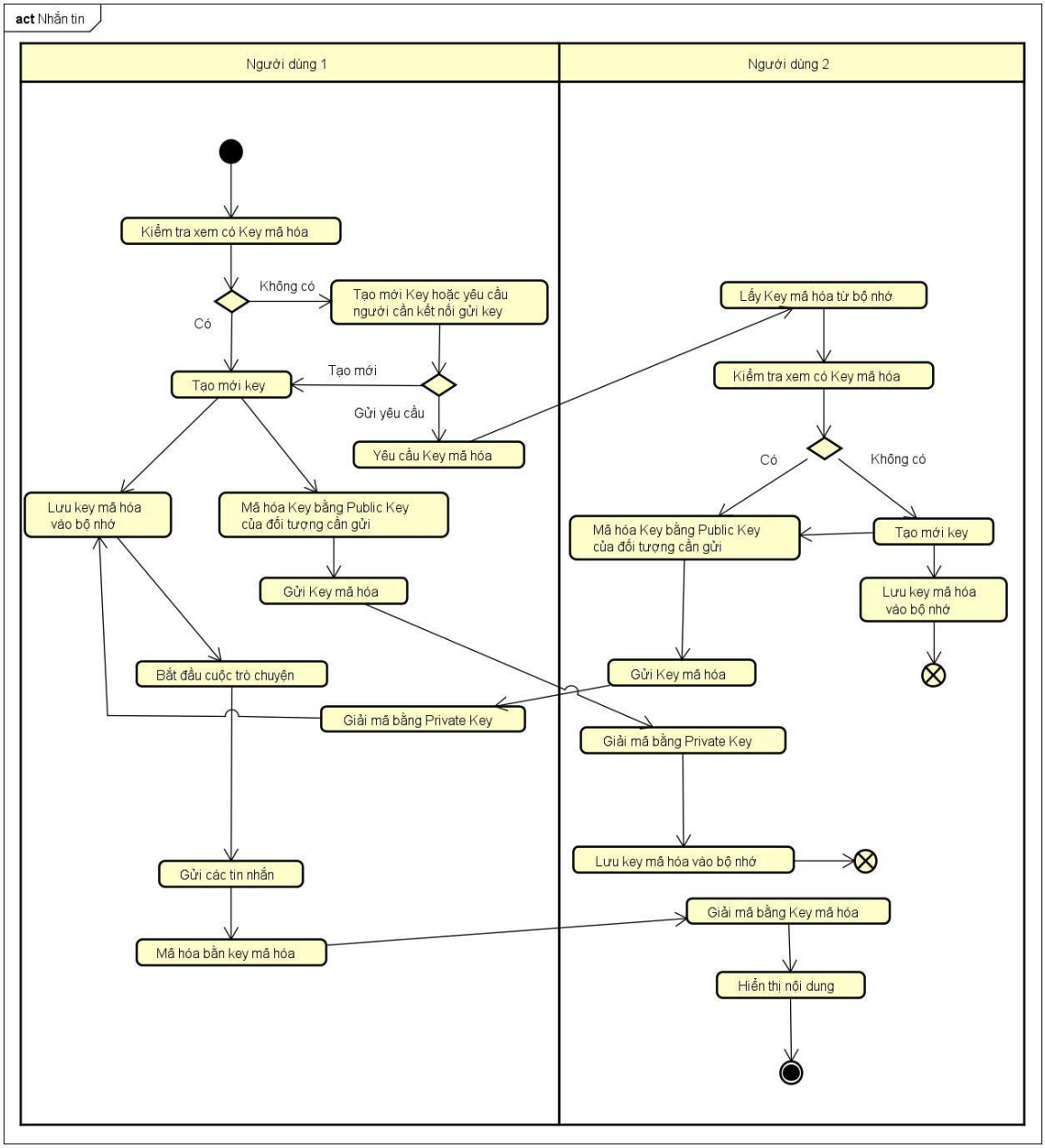
2.2.2.4 Phân rã use case quản lý nhóm



Hình 5 Phân rã use case quản lý nhóm

Hình 5 mô tả nhóm chức năng quản lý danh sách nhóm, người dùng bình thường sẽ có thể chỉnh sửa thông tin nhóm mình đã tham gia, thêm các thành viên khác hay rời nhóm. Còn quản trị viên nhóm cũng là người dùng, nên sẽ có thêm chức năng xóa thành viên trong nhóm.

2.2.3 Quy trình nghiệp vụ



Hình 6 Quy trình nghiệp vụ nhắn tin

Hình 6 Quy trình nghiệp vụ nhắn tin mô tả quy trình nhắn tin bao gồm các use case trong nhóm chức năng nhắn tin, thể hiện các bước mã hóa tin nhắn một cách bảo mật, bảo đảm được độ an toàn trong khi trò chuyện.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case đăng nhập bằng số điện thoại

Bảng 2 Đặc tả use case đăng nhập bằng số điện thoại

<table>
<tr><td>Mã use case</td><td colspan="2">UC01</td><td>Tên Use case</td><td>Đăng nhập</td></tr>
<tr><td>Tác nhân</td><td colspan="4">Khách</td></tr>
<tr><td>Tiền điều kiện</td><td colspan="4">Không</td></tr>
<tr><td>Luồng sự kiện chính</td><td colspan="4">
<table>
<tr><th>STT</th><th>Thực hiện</th><th>Hành động</th></tr>
<tr><td>1.</td><td>Khách</td><td>Nhập thông tin số điện thoại và nhấn xác nhận</td></tr>
<tr><td>2.</td><td>Hệ thống</td><td>Kiểm tra đúng là số điện thoại</td></tr>
<tr><td>3.</td><td>Hệ thống</td><td>Gửi mã OTP xác thực tới số điện thoại vừa nhập</td></tr>
<tr><td>4.</td><td>Hệ thống</td><td>Hiển thị màn hình nhập mã OTP</td></tr>
<tr><td>5.</td><td>Khách</td><td>Nhập mã OTP và nhấn xác nhận</td></tr>
<tr><td>6.</td><td>Hệ thống</td><td>Kiểm tra xem thống tin OTP</td></tr>
<tr><td>7.</td><td>Hệ thống</td><td>Kiểm tra Secret Key trong máy khách đã tồn tại, lấy public key.</td></tr>
</table>
</td></tr>
</table>

	8.	Hệ thống	Tạo một token device để gửi thông báo tới thiết bị
	9	Hệ thống	Lưu số điện thoại, public key, token device lên cơ sở dữ liệu
	10	Hệ thống	Tạo token và gửi về cho khách
	11	Hệ thống	Thông báo đăng nhập thành công và hiển thị màn hình chính

Luồng sự kiện thay thế

STT	Thực hiện	Hành động
1.a	Khách	Nhập sai số điện thoại
2.a	Hệ thống	Thông báo số điện thoại không hợp lệ
5.b	Khách	Nhập sai mã OTP
6.b	Hệ thống	Thông báo sai mã OTP, yêu cầu nhập lại
6.c	Hệ thống	Mã OTP hết hạn, thông báo mã OTP đã hết hạn, quay lại màn hình nhập số điện thoại
7.d	Hệ thống	Secret Key chưa tồn tại, tạo mới Secre Key

Hậu điều kiện	

STT	Thực hiện	Hành động
1	Hệ thống	Tạo token cho người dùng
2	Hệ thống	Sẽ có Secret Key trong bộ nhớ thiết bị của người dùng

Bảng 3 Dữ liệu đầu vào use case đăng nhập bằng số điện thoại

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	phone	Số điện thoại	Có	Đúng định dạng	0910031999
2.	otp	Mã xác thực	Có	Có 6 kí tự là số	123456

2.3.2 Đặc tả use case nhắn tin

Bảng 4 Đặc tả use case nhắn tin

Mã use case	**UC02**	**Tên Use case**	Nhắn tin
Tác nhân	Người dùng		
Tiền điều kiện	Không		

Luồng sự kiện chính	

STT	Thực hiện	Hành động
1.	Người dùng	Chọn người dùng khác hoặc nhóm đã tham gia để nhắn tin
2.	Hệ thống	Mở giao diện nhắn tin
3.	Hệ thống	Kiểm tra xem đã có khóa phiên
4.	Hệ thống	Hiển thị nội dung tin nhắn đã nhắn trước đó
5.	Người dùng	Nhập văn bản vào ô nhập, sau đó nhấn nút gửi.
6.	Hệ thống	Mã hóa văn bản người dùng gửi bằng khóa phiên và gửi nội dung cho người nhận
7.	Hệ thống	Giải mã tin nhắn bằng khóa phiên và hiển thị lại cho người dùng

Luồng sự kiện thay thế

STT	Thực hiện	Hành động
3.a	Hệ thống	Người dùng chưa có khóa phiên
4.a	Hệ thống	Hiện hộp thoại thông báo chưa có khóa phiên, yêu cầu tạo mới hoặc yêu cầu người dùng khác gửi lại khóa phiên cũ
5.a	Người dùng	Chọn tạo mới khóa phiên
6.a	Hệ thống	Tạo khóa phiên cho người dùng, và gửi cho đối tượng cần gửi
5.a.1	Người dùng	Chọn yêu cầu người dùng khác gửi lại khóa phiên cũ
6.a.1	Hệ thống	Gửi yêu cầu đến người dùng cần gửi
7.a.1	Hệ thống	Thông báo đã gửi yêu cầu và thoát khỏi tin nhắn
4.a.2	Người dùng	Chọn hủy
5.a.2	Hệ thống	Thoát khỏi tin nhắn
5.b	Người dùng	Chọn tệp tin hình ảnh để gửi
6.b	Hệ thống	Kiểm tra quyền đã được chấp nhận
7.b	Hệ thống	Hiện thị hình ảnh trong bộ nhớ
8.b	Người dùng	Chọn tệp và gửi đi

	9.b	Hệ thống	Gửi nội dung cho người cần gửi và hiển thị lại nội dung đã nhắn cho người dùng
	6.b.1	Hệ thống	Quyền chưa được chấp nhận
	7.b.1	Hệ thống	Mở hộp thoại yêu cầu quyền
	8.b.1	Người dùng	Chọn cấp quyền
	9.b.1		Quay trở lại bước 6b
	8.b.2	Người dùng	Chọn không cấp quyền
	9.b.2	Hệ thống	Hiện thông báo bạn chưa cấp quyền và không thể sử dụng chức năng này
	5.c	Người dùng	Chọn gửi đoạn âm thanh
	6.c	Hệ thống	Mở hộp thoại ghi âm
	7.c	Người dùng	Ghi âm và gửi
	8.c	Hệ thống	Gửi đi và hiển thị lại nội dung đã gửi
	5.d	Người dùng	Chọn xóa tin nhắn
	6.d	Hệ thống	Xóa nội dụng tin nhắn và hiển thị thông báo đã xóa
Hậu điều kiện	Không		

Bảng 5 Dữ liệu đầu vào use case nhắn tin

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	inputText	Nội dung dạng văn bản	Không	Đúng định dạng	Chào bạn 112234
2.	inputImage	Tệp hình ảnh	Không	Có đuôi jpg,png,jpeg	ac.png
3	inputVideo	Tệp video	Không	Có đuôi mp4	sdsd.mp4
4	inputVoice	Đoạn ghi âm	Không	Nội dung ghi âm	

2.3.1 Đặc tả use case gọi điện nhóm

Bảng 6 Đặc tả use case gọi điện nhóm

<table>
<tr><td>Mã use case</td><td>UC03</td><td>Tên Use case</td><td>Gọi điện nhóm</td></tr>
<tr><td>Tác nhân</td><td colspan="3">Người dùng</td></tr>
<tr><td>Tiền điều kiện</td><td colspan="3">Người dùng đã ở trong ngóm cần gọi</td></tr>
<tr><td>Luồng sự kiện chính</td><td colspan="3">
<table>
<tr><th>STT</th><th>Thực hiện</th><th>Hành động</th></tr>
<tr><td>1.</td><td>Người dùng</td><td>Chọn chức năng gọi điện nhóm</td></tr>
<tr><td>2.</td><td>Hệ thống</td><td>Hiển thị màn hình chọn thành viên sẽ nhận được cuộc gọi</td></tr>
</table>
</td></tr>
</table>

	3.	Người dùng	Chọn thành viên và bắt đầu gọi
	4.	Hệ thống	Gửi yêu cầu và mã phòng đến các thành viên sẽ nhận được cuộc gọi
	5.	Hệ thống	Chuyển sang màn hình đợi
	6.	Hệ thống	Khi có người nghe cuộc gọi thì sẽ chuyển sang màn hình cuộc gọi
	7.	Người dùng	Chọn tắt cuộc gọi để tắt

Luồng sự kiện thay thế

STT	Thực hiện	Hành động
3.a	Người dùng	Không chọn ai và nhấn gọi điện
4.a	Hệ thống	Thông báo cần chọn thành viên
3.c	Người dùng	Chọn hủy
6.b	Hệ thống	Thoát khỏi màn hình

Hậu điều kiện	Không

Bảng 7 Dữ liệu đầu vào use case gọi điện thoại nhóm

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	memberCall	Danh sách người sẽ nhận cuộc gọi	Có	Là thành viên của nhóm	

2.3.2 Đặc tả use case quản lý danh sách bạn bè

Bảng 8 Đặc tả use case quản lý nhóm

<table>
<tr><td>Mã use case</td><td>UC03</td><td>Tên Use case</td><td>Quản lý danh sách bạn bè</td></tr>
<tr><td>Tác nhân</td><td colspan="3">Người dùng</td></tr>
<tr><td>Tiền điều kiện</td><td colspan="3">Không</td></tr>
<tr><td>Luồng sự kiện chính</td><td colspan="3">
<table>
<tr><th>STT</th><th>Thực hiện</th><th>Hành động</th></tr>
<tr><td>1.</td><td>Người dùng</td><td>Vào danh sách bạn bè</td></tr>
<tr><td>2.</td><td>Hệ thống</td><td>Hiển thị danh sách bạn bè cùng danh sách lời mời kết bạn</td></tr>
<tr><td>3.</td><td>Người dùng</td><td>Chọn vào một lời mời kết bạn</td></tr>
<tr><td>4.</td><td>Hệ thống</td><td>Hiển thị thông tin người gửi lời mời</td></tr>
</table>
</td></tr>
</table>

	5.	Người dùng	Nhấn vào nút đồng ý
	6.	Hệ thống	Cập nhật danh sách bạn bè và danh sách lời mời kết bạn

Luồng sự kiện thay thế

STT	Thực hiện	Hành động
5.a	Người dùng	Chọn từ chối lời mời
6.a	Hệ thống	Thông báo đã từ chối và cập nhật lại danh sách
3.b	Người dùng	Chọn vào một người trong danh sách bạn bè
4.b	Hệ thống	Hiển thị thông tin bạn bè
5.c	Người dùng	Chọn hủy kết bạn
6.c	Hệ thống	Thông báo và cập nhật lại danh sách kết bạn
5.c.1	Người dùng	Chọn nhắn tin
6.c.1	Hệ thống	Vào phần nhắn tin với bạn này (UC02)
3.d	Người dùng	Chọn thêm bạn
4.d	Hệ thống	Hiển thị hộp thoại thêm bạn bằng qr code hoặc nhập id
5.d	Người dùng	Chọn thêm bằng mã qr code

	6.d	Hệ thống	Hiển thị màn hình quét mã qr code
	7.d	Người dùng	Đưa hình qr code lên và quét
	8.d	Hệ thống	Kiểm tra thông tin và hiển thị thông tin người có qr code đó
	9.d	Người dùng	Chọn kết bạn
	10.d	Hệ thống	Thông báo và gửi lời mời kết bạn
	8.d.1	Hệ thống	Qr code sai, thông báo qr code không hợp lệ
	5.d.2	Người dùng	Chọn kết bạn bằng id
	6.d.2	Hệ thống	Hiển thị người dùng chứa id đó
	7.d.2		Quay trở lại 9.d
Hậu điều kiện	Không		

Bảng 9 Dữ liệu đầu vào use case quản lý nhóm

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1.	Qr code	Ảnh qr code được tạo từ ứng dụng	Có	Được tạo bởi ứng dụng	

2.	Mã id	Id của người dùng	Có	Có 32 kí tự	19108a9ccb3147f6a67923b127aba750

2.4 Yêu cầu phi chức năng

2.4.1 Yêu cầu bảo mật

Hệ thống để đảm tính bảo mật, tính xác thực chính chủ và tiện sử dụng, nên ứng dụng chỉ cho phép đăng nhập bằng số điện thoại và có xác thực mã OTP. Sau khi xác thực thành công sẽ tạo một cặp key bất đối xứng: public key và private key. Private key là khóa dùng để giải mã sẽ được lưu tại máy khách, còn public key là khóa dùng để mã hóa các thông điệp quan trọng và sẽ lưu tại máy chủ. Đồng thời sử dụng Json web token để mã hóa thông tin đăng nhập người dùng vã xác thực các request đến server trước khi xử lý.

Trong các cuộc hội thoại cá nhân sẽ luôn có khóa đối xứng hay còn gọi là khóa phiên, dùng để mã hóa các dữ liệu truyền đi, và được lưu lại ở máy khách, khi tạo tại máy khách sẽ được mã hóa bằng public key của người cần giao tiếp và gửi cho người đó, sau khi nhận được người đó sẽ giải mã bằng private key của mình, và hai người bắt đầu giao dịch.

Và để đảm bảo tính an toàn dữ liệu, các nội dung đã nhắn tin chỉ hiển thị khi có khóa phiên tin nhắn đó và hệ thống chỉ cho phép đăng nhập tại một thiết bị.

2.4.2 Yêu cầu giao diện

Bên cạnh tính bảo mật thì ứng dụng cần có giao diện thân thiện, dễ sử dụng, phù hợp với hầu hết các điện thoại hệ điều hành android. Sử dụng các icon liên quan đến các tính năng để tăng sự thân thiện, giao diện đơn giản, dễ dùng dễ tương tác.

2.4.3 Yêu cầu khác

Ngoài ra ứng dụng khi thực hiện cần đáp ứng các yêu cầu (i) tính khả thi - xây dựng ứng dụng đáp ứng các yêu cầu từ phía người dùng, (ii) tính linh động - ứng dụng dễ dàng thay đổi hay mở rộng

Chương 2 em đã trình bày về các khảo sát cùng sự phân tích yêu cầu về các chức năng cũng như về phi chức năng. Sau đây sẽ là chương 3 về các công nghệ em sẽ sử dụng để hoàn thành ĐATN

Chương 3 Công nghệ sử dụng

Từ sự phân tích các yêu cầu cụ thể ở chương 2, qua sự tìm hiểu, em đã đúc kết ra được các công nghệ chính sẽ sử dụng cho phía client và server để hoàn thành đồ án.

3.1 Server

3.1.1 NodeJs

NodeJS là nền tảng được xây dựng trên "V8 Javascript Engine" , được viết bằng Javascript và c++. Là công cụ mạnh mẽ để xây dựng phần sever một cách đơn giản và nhanh chóng.

NodeJS có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ (non-blocking), có thể xử lý nhiều kết nối cùng lúc, đồng thời cũng dễ dàng mở rộng.

NodeJS có cộng đồng lớn mạnh, bởi vậy nên khi gặp lỗi thì ta có thể dễ dàng tìm được giải pháp nhờ sự giúp đỡ của cộng đồng.

3.1.2 MySQL

MySQL là một hệ thống quản trị cơ sở dữ liệu quan hệ mã nguồn mở (RDMS), dựa trên ngỗn ngữ truy vấn có cấu trúc (SQL) và được hỗ trợ bởi tập đoàn Oracle.

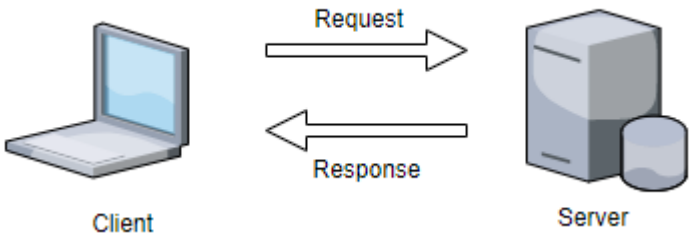
MySQL được sử dụng rỗng rãi bởi khả năng dễ sử dụng, hoạt động rất nhanh và tốt với các tập dữ liệu lớn và được sử dụng với các trang web lớn như Google, Twitter, Facebook,…

Tính năng nổi bật của MySQL:

- MySQL được phát hành theo giấy phép mã nguồn mở.
- MySQL lý tưởng cho tất cả ứng dụng lớn và nhỏ.
- MySQL có tốc độ xử lý, truy vấn nhanh, an toàn, dễ mở rộng và sử dụng.
- MySQL hoạt động trên đa nên tảng với hầu hết các ngôn ngữ (PHP, JavaScript, Java, C, Kotlin,…
- MySQL hỗ trợ cơ sở dữ liệu lớn, lên tớ 50 triệu hàng trong một bảng, giới hạn kích thược mặc định một bảng là 4GB và có thể tăng giới hạn đến 8 triệu TB.
- MySQL có tiêu chuẩn bảo mật mã hóa thông tin cao, đảm bảo an toàn dữ liệu

Các tính năng này khiến MySQL phù hợp cho đồ án ứng dụng nhắn tin của em.

MySQL hoạt động dựa trên mô hình Client-Server, được mô tả ở **Hình 7**:



Hình 7 Mô hình hoạt động của MySQL

Máy client sẽ liên lạc với server trong mạng nhất đinh, MySQL tạo ra bảng để lưu trữ, định nghĩa dữ liệu. Client sẽ gửi các truy vấn đến server và server sẽ phản hồi lại kết quả.

Cách thức hoạt động của MySQL phù hợp với hầu hết các ứng dụng, trong đó có ứng dụng nhắn tin.

3.2 Client

3.2.1 Kotlin

Kotlin là ngôn ngữ lập trình kiểu tĩnh, chạy trên máy ảo java. Đây là ngôn ngữ lập trình mã nguồn mở, được phát triển bởi JetBrains.

Kotlin được ra mắt từ 2011 và đến 2017 được Google hỗ trợ đầy đủ cho việc lập trình ứng dụng cho hệ điều hành Android. Và hiện nay Kotlin là ngôn ngữ số một trong việc lập trình hệ điều hành Android và được Google khuyến khích sử dụng để thay thế Java.

Với khả năng tương tác mạnh mẽ, Kotlin tương tác rất tốt với Java và hoàn toàn có thể code dự án Java và Kotlin song song, và phần lớn các thư viện java Kotlin đều sử dụng được. Đây là một trong những lý do em chọn Kotlin là ngôn ngữ lập trình cho phía client, đồng thời Kotlin có cách viết gọn nhẹ hơn nhiều so với Java. Và hiện nay Kotlin có hỗ trợ đa nền tảng nên về sau việc mở rộng dự án sang các nền tảng khác sẽ tốn ít chi phí hơn.

3.3 Tích hợp công nghệ

3.3.1 Websocket

Hình 8 Giao tiếp client- server thông qua websocket

Websocket là công cụ hỗ trợ giao tiếp giữa client và server để tạo một kết nối trao đổi dữ liệu. Giao thức này được thực hiện qua TCP. Websocket phù hợp với các ứng dụng real-time(thời gian thực).

Client và server sẽ trao đổi dữ liệu dưới dạng chuẩn hóa JSON. Và server sẽ truy vấn cơ sở dữ liệu qua ngôn ngữ truy vấn sql.

3.3.2 RESTful API

Để giảm tải số lượng request ở Websocket, trong đồ án của em còn sử dụng RESTful API để lấy các dữ liệu từ cơ sở dữ liệu ở một số tác vụ chỉ cần lấy dữ liệu.

RESTful API là một tiêu chuẩn trong việc thiết kế API cho các ứng dụng client server để thuận tiện cho việc quản lý tài nguyên và truyền tải qua phương thức HTTP.

Thành thần chính của RESTful API:

- API (Application Programming Interface): là tập hợp các quy tắc để tương tác giữa các thành phần ứng dụng. API trả về dạng dữ liệu JSON hoặc XML.
- REST(Representational State Transfer): là dạng chuyển đổi cấu trúc để viết API. Nó tạo tương tác giữa các máy với nhau qua HTTP và quy định việc sử dụng các phương thức HTTP và dạng URL cho các ứng dụng.

Và trong đồ án này thì dữ liệu trả về sẽ luôn là dạng JSON.

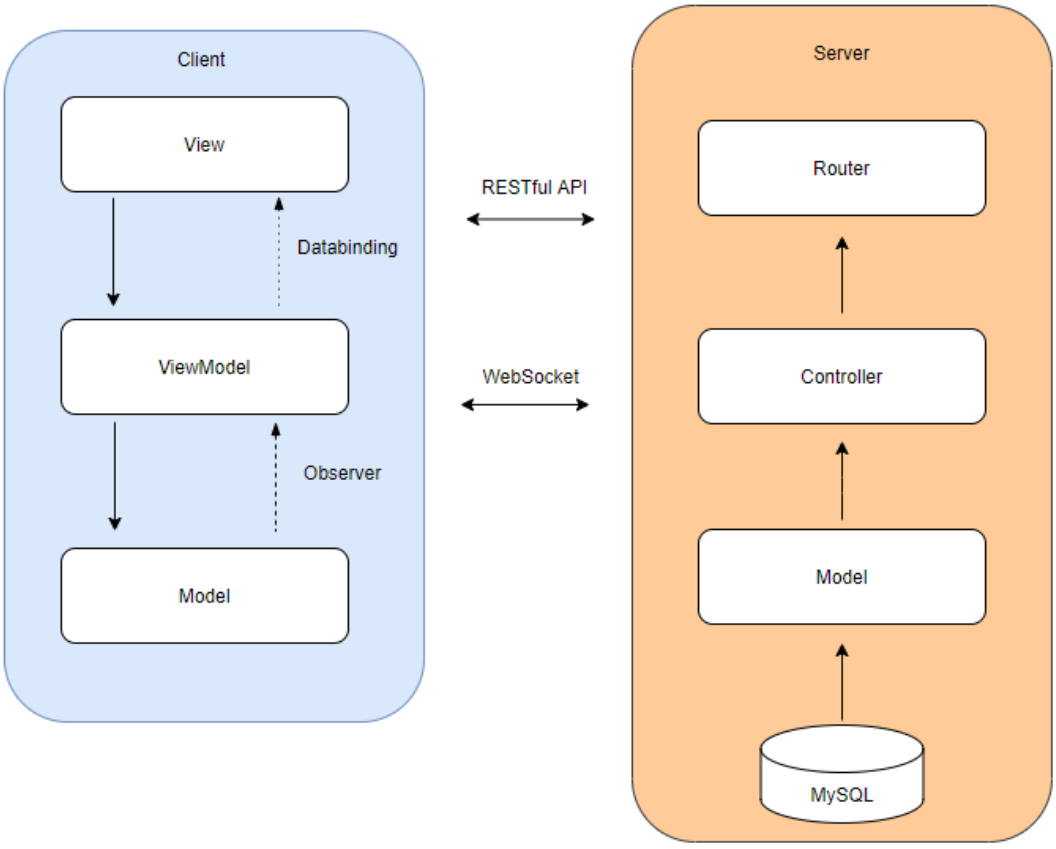
Chương 3 em đã trình bày xong về các công nghệ được em sử dụng, sau đây em sẽ trình bày chương 4 về phát triển và triển khai ứng dụng.

Chương 4 Phát triển và triển khai ứng dụng

Trong chương 4 là các thiết kế về kiến trúc để xây dựng đồ án, sau đó xây dựng ứng dụng với các thư viện và kết quả đã đạt được trong đồ án và cách triển khai hệ thống, đồng thời cũng trình bày về các kiểm thử các chức năng chính của ứng dụng.

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

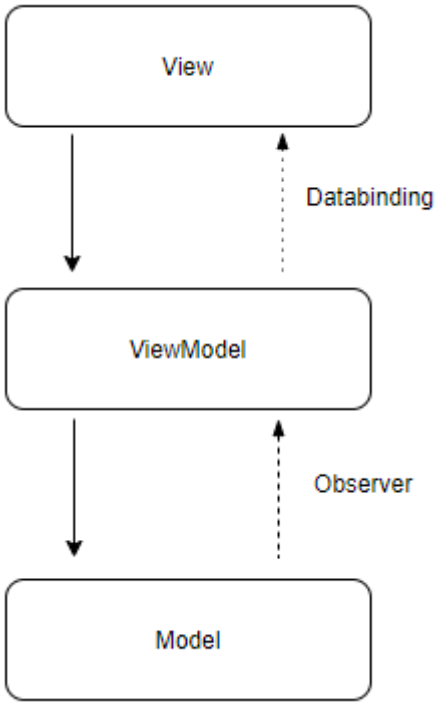


Hình 9 Tổng quan kiến trúc

Hệ thống được triển khai theo mô hình client–server, được mô tả như **Hình 9**. Server sẽ lấy dữ liệu hay thêm, sửa, xóa dữ liệu ở database thông qua model, các logic xử lý sẽ được thực hiện tại controller, và router sẽ làm nhiệm vụ định tuyến, cung cấp các phương thức cho client giao tiếp, lấy dữ liệu.Còn client đóng vai trò hiển thị dữ liệu cho người dùng thông

qua dữ liệu server cung cấp, đồng thời cung cấp giao diện tương tác với người dùng, từ đó gửi các yêu cầu lên cho server xử lý.Cả server và client được xây dựng độc lập và tương tác chung qua RESTful API hoặc là Websocket. Client được xây dựng theo mô hình MVVM.

4.1.1.1 Mô hình MVVM



Hình 10 Mô hình MVVM

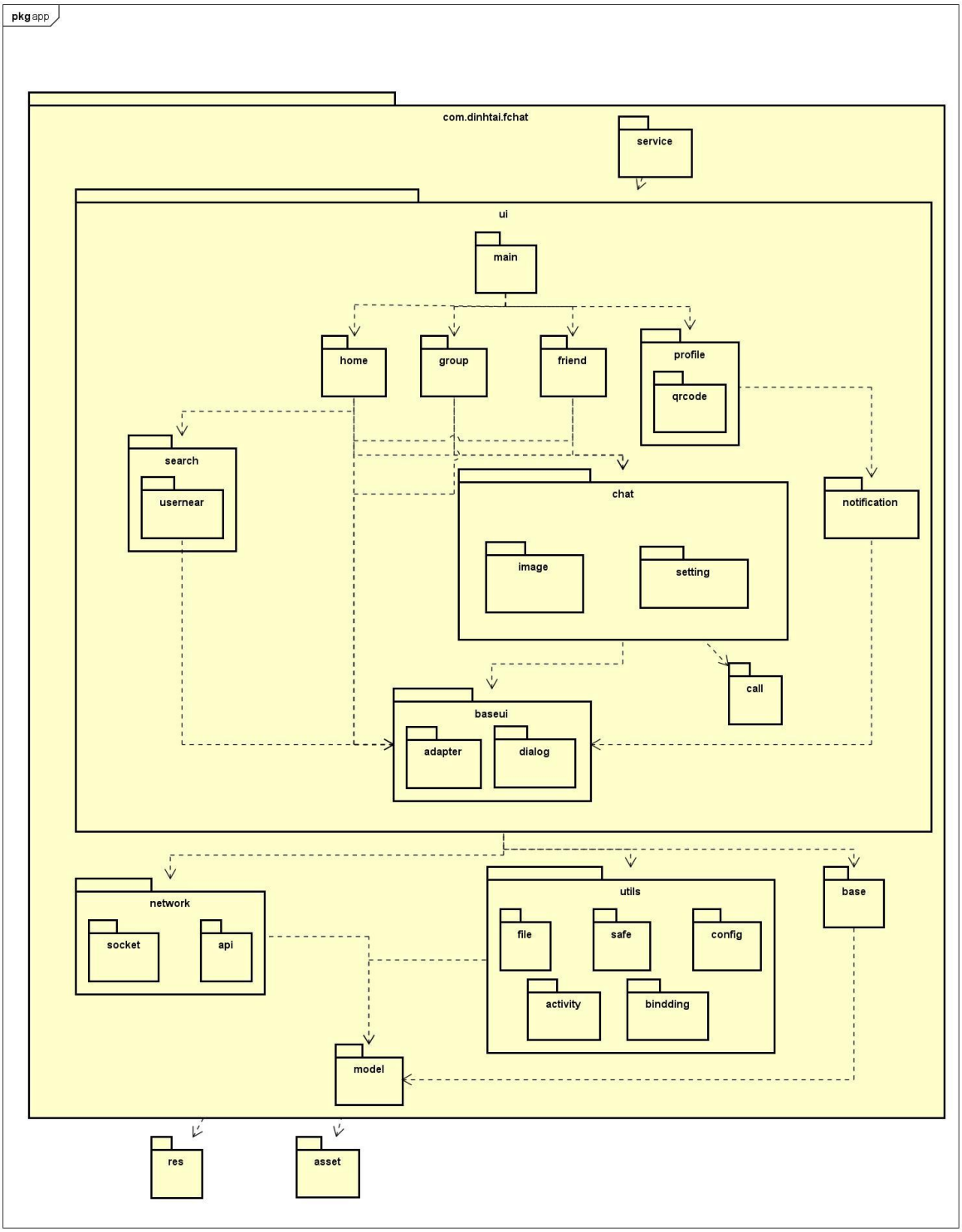
Đa số các ứng dụng trên bất kì nền tảng đều có thể chia thành hai phần: giao diện (View) và dữ liệu (Model). Vì để tách riêng hai phần này, cần có một phần trung gian để đảm bảo kết nối giữa hai phần. Và hiện nay đang có các mô hình đảm bảo điều đó MVC, MVP,MVVM,… Và đang nói ở đây là mô hình MVVM.

MVVM là viết tắt của ba từ: Model, View và ViewModel, và cách hoạt động được mô tả ở **Hình 10**. View là phần giao diện của ứng dụng để hiển thị và nhận tương tác của người dùng, và View ở mô hình này năng động hơn khi thực hiện hành vi phản hồi người dùng qua databinding. Model là phần xử lí những tác vụ liên quan đến dữ liệu, database, truy vấn, thêm, sửa, xóa dữ liệu. Còn ViewModel là lớp trung gian giữa View và Model, đảm nhận chức năng xử lý dữ liệu nhận được ở Model và xử lý logic và hiển thị lên View.

Ưu điểm của mô hình này là các phần tách riêng biệt, dễ dàng cho việc kiểm thử đơn vị. Khả năng mở rộng dễ dàng, và bảo trì nhanh chóng. Còn về nhược điểm là mô hình này thiết kế ViewModel tốn nhiều thời gian hơn, khi dự án lớn dần sẽ khiến việc quản lý khó khăn và việc debug khi cấu trúc dữ liệu phúc tạp sẽ khó khăn hơn.

4.1.2 Thiết kế tổng quan

4.1.2.1 Biểu đồ gói cho client

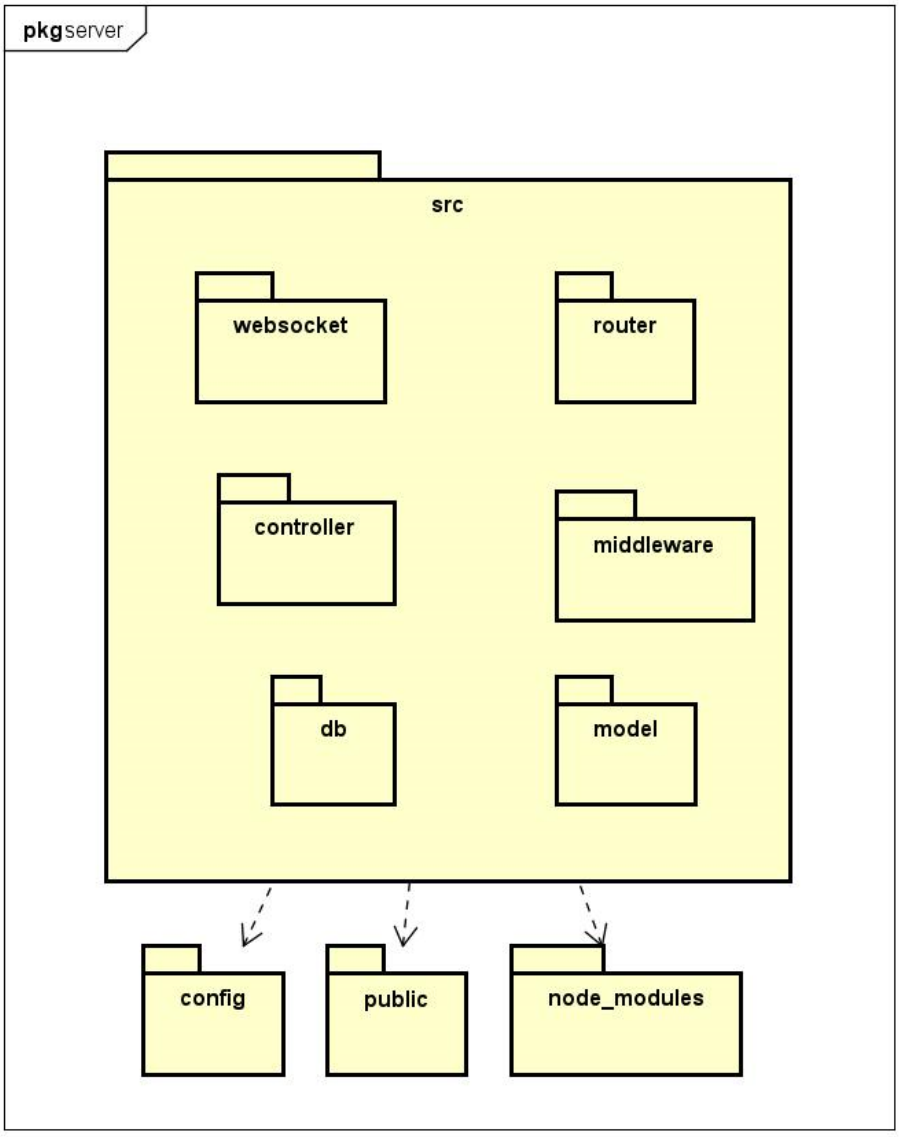


Hình 11 Biểu đồ gói client

Hình 11 là biểu đồ gói cho client, được thiết kế theo chuẩn thiết kế ứng dụng android, gồm ba thư mục chính:

- Thư mục res chứa giao diện hiển thị lên phía người dùng.
- Thư mục asset chứa tài nguyên ứng dụng.
- Thư mục com.dinhtai.fchat là thư mục chứa phần code, được chia thành 6 thư mục chính: (i) ui là chứa các code locgic liên quan đến giao điện người dùng, (ii) base chứa các class cơ sở, (iii) service chứa các class thực hiện chức năng chạy ngầm của ứng dụng, (iv) network chứa các class thực hiện phần giao tiếp với server qua websocket hay RESful api, (v) model chứa các class thực thể, (vi) utils chứa các function không thuộc bất kì class nào hay còn lại là các static function.

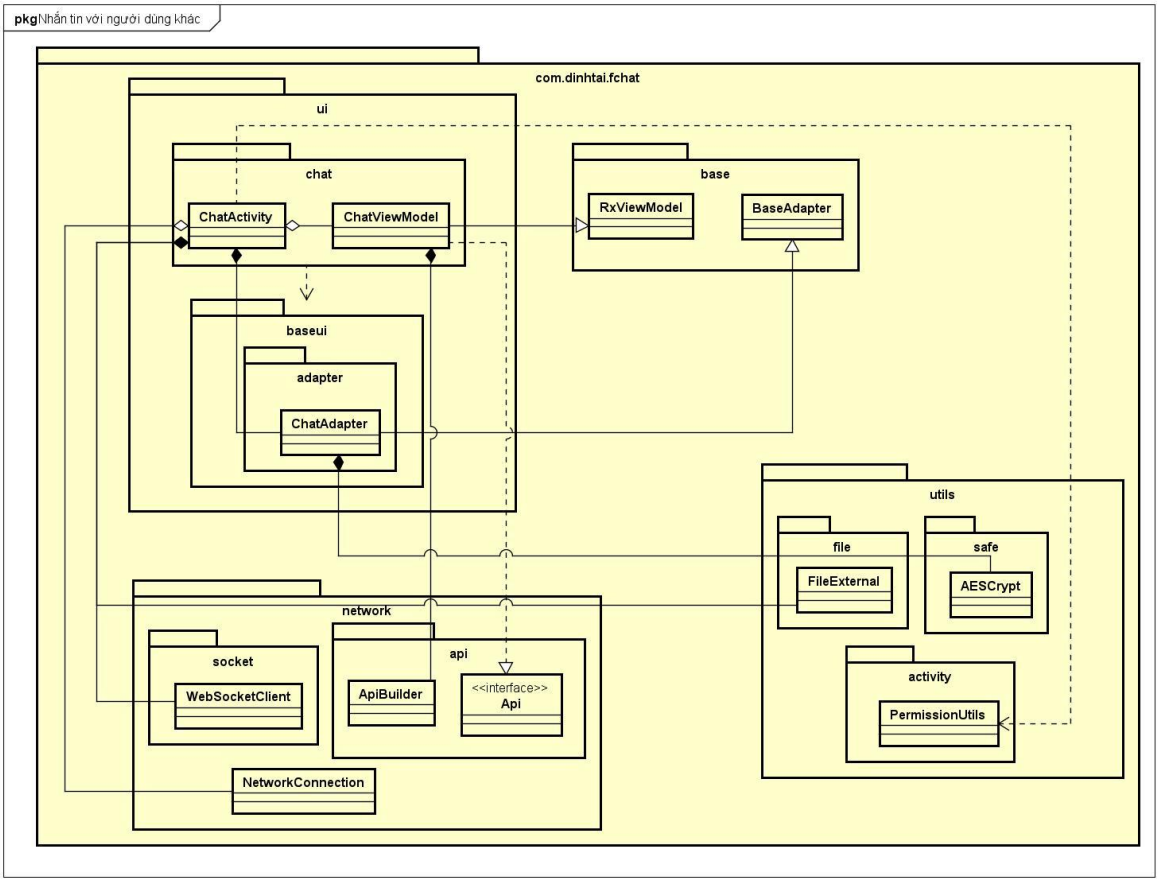
4.1.2.2 Biểu đồ gói cho server



Hình 12 Biểu đồ gói cho server

Hình 12 mổ tả biểu đồ gói cho server: (i) config chứa các config của server, (ii) public chứa tài nguyên server, (iii) websocket chứa các phương thức kết nối với client qua websocket, (iv) router cung cấp các RESful API, (v) controller chứa các function thực hiện các logic, (v) db chưa phương thức kết nối database, (vi) model chứa các thực thể.

4.1.3 Thiết kế chi tiết gói



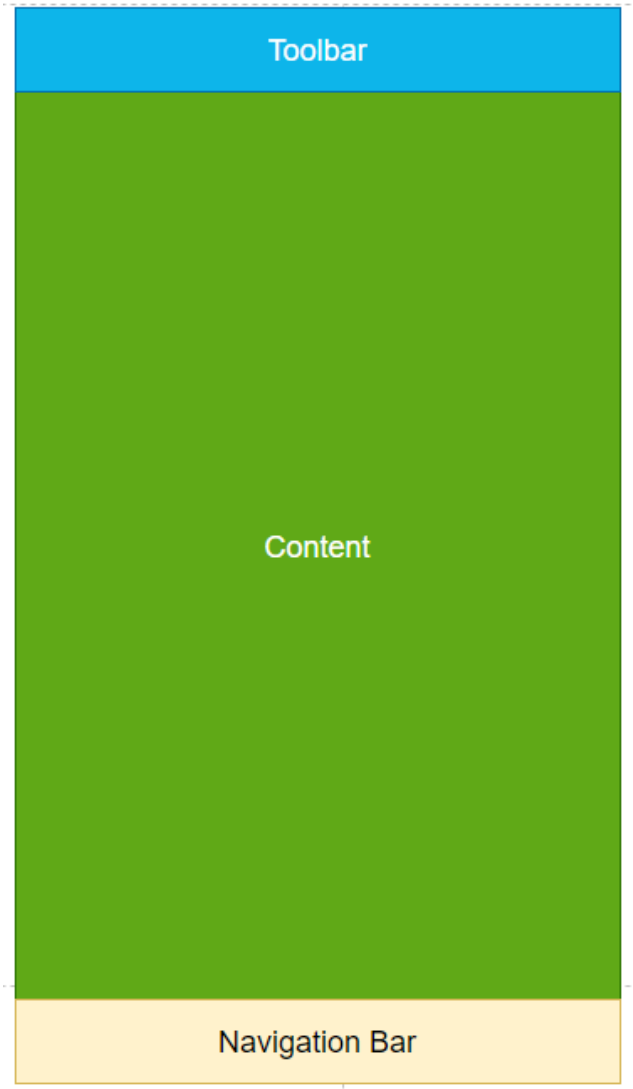
Hình 13 Thiết kế chi tiết gói nghiệp vụ nhắn tin với người dùng khác

Hình 13 mô tả chi tiết gói nghiệp vụ nhắn tin với người dùng khác. Người dùng sẽ xem các tin nhắn và tương tác qua lớp ChatActivity. Lớp ChatViewModel sẽ đảm nhận việc tương tác với server qua lớp interface Api. Gói network sẽ đảm nhận các công việc liên quan đến Internet và kết nối.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

4.2.1.1 Mockup bố cục màn hình chính



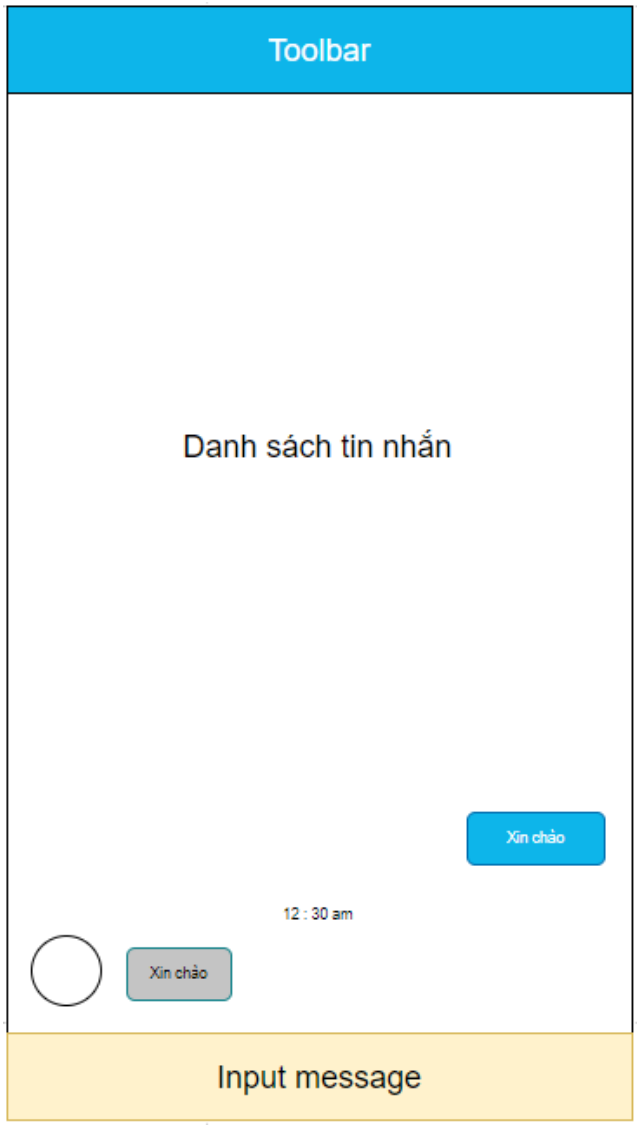
Hình 14 Bố cục màn hình chính

Ứng dụng được thiết kế trên nền tảng android, phù hợp với hầu hết màn hình điện thoại android hiện tại,nhưng tương thích nhất với độ phân giải là 1280x720 pixels. Bố cục gồm ba phần chính là Toolbar, Content, và Navigation Bar. Toolbar chứa thanh tiêu đề của màn hình, đồng thời có các giao diện tương tác nhanh tại đây. Content là nội dung chính, được thiết kế chính bởi các fragment hiển thị các thông tin chính của màn hình. Navigation Bar là thành phần điều hướng các màn hình, khi chuyển qua màn hình khác, chỉ có Content và Toolbar thay đổi phù hợp với nội dung hiển thị.

Bảng 10 Cấu hình chung cho giao diện màn hình

Thuộc tính	**Cấu hình**
Màu sắc	Màu chính: #0DB5EA Màu khác : #C4C4C4
Font chữ	Roboto
Vị trí hiển thị thông báo	Ở giữa của phía dưới màn hình

4.2.1.2 Mockup của màn hình chat



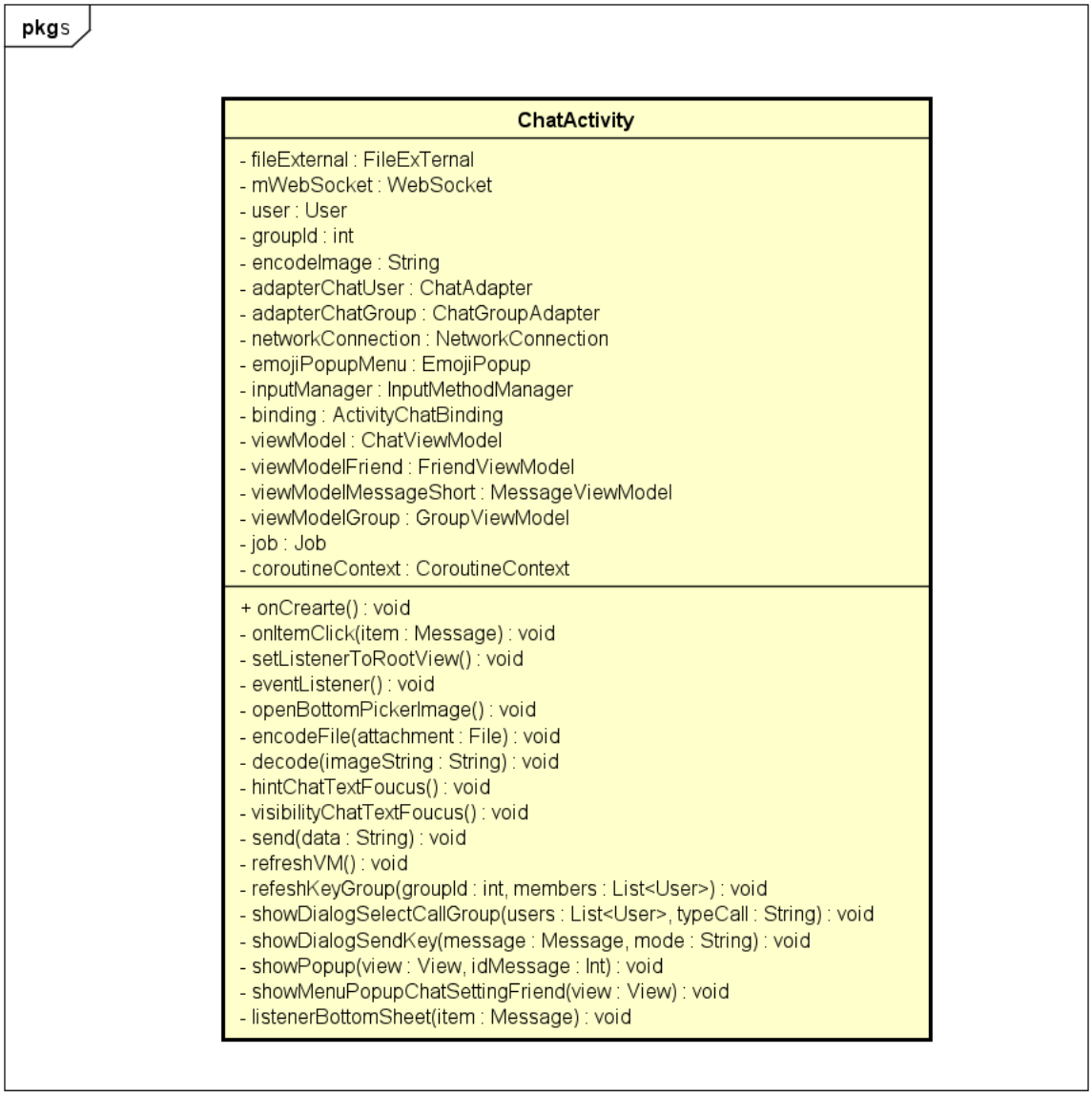
Hình 15 Màn hình chat

Tương tự như màn hình chính, màn hình chat cũng được thiết kế với độ phân giải như màn hình chính. Được mô tả như **Hình 15**, gồm có ba thành phần chính là Toolbar, danh sách tin nhắn, và phần nhập tin nhắn. Toolbar tương tự ở màn hình chính, phần này hiển thị tiêu đề và cung cấp giao diện tương tác nhanh như gọi điện, hay để vào mục tùy chỉnh. Danh sách tin nhắn là hiển thị nội dung đoạn chat như là văn bản, hình ảnh hoặc hội thoại. Còn phần nhập tin nhắn người dùng có thể nhập tin nhắn văn bản, âm thanh từ thiết bị.

Bảng 11 Cấu hình màn hình chat

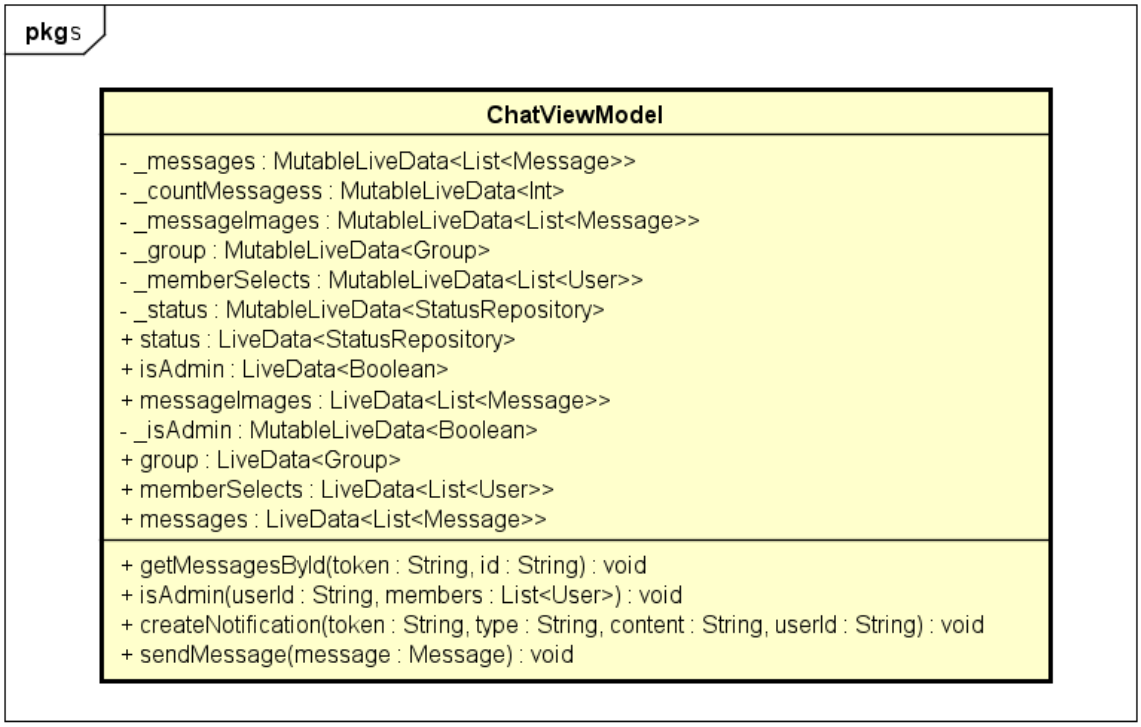
Thuộc tính	**Cấu hình**
Màu sắc	#0DB5EA #C4C4C4
Font chữ	Roboto
Vị trí hiển thị thông báo	Ở giữa của phía dưới màn hình

4.2.2 Thiết kế lớp



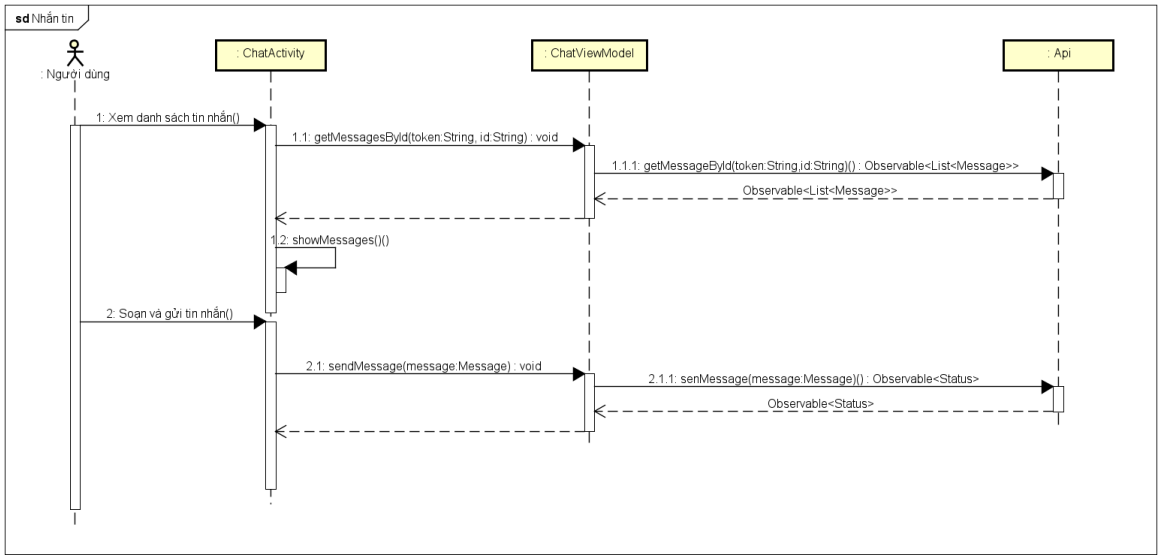
Hình 16 Biểu đồ lớp ChatActivity ở client

Hình 16 mô tả chi tiết lớp ChatActivity, lớp này có vai trò hiển thị thông tin các tin nhắn đã trao đổi của người dùng với người dùng hay trong nhóm và cung cấp giao diện tương tác cho người dùng và nó đóng vai trò View trong mô hình MVVM mà em đã sử dụng



Hình 17 Biểu đồ lớp ChatViewModel ở client

Hình 17 mô tả chi tiết lớp ChatViewModel, lớp này có vai trò là phần ViewModel trong mô hình MVVM mà em sử dụng, nó sẽ lấy dữ liệu bằng việc gọi các phương thức và trả về kết quả ở các thuộc tính MutableLiveData, sau đó hiển thị cho người dùng thông qua các thuộc tính LiveData, thuộc tính này giúp phần View có thể cập nhật giữ liệu nếu dữ liệu thay đổi mà không phải gọi lại phương thức.

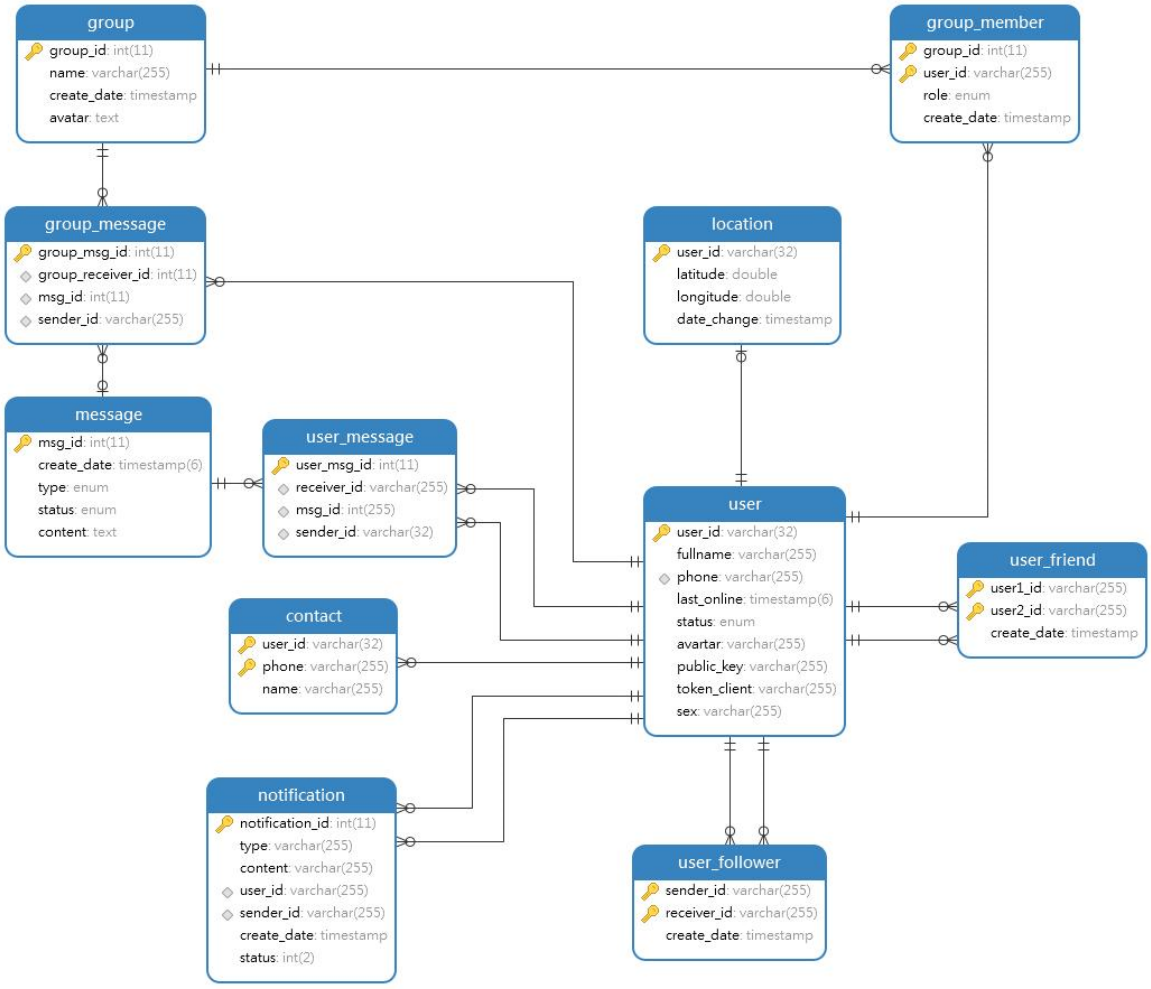


Hình 18 Biểu đồ trình tự usecase nhắn tin

Hình 18 mô tả biểu đồ trình tự của usecase nhắn tin một cách cơ bản. Lớp Api sẽ cung cấp các phương thức để tương tác với server thông qua các API

4.2.3 Thiết kế cơ sở dữ liệu

4.2.3.1 Mô hình thực thể liên kết



Hình 19 Mô hình thực thể liên kết

Sau đây, các **Bảng 12** đến **Bảng 22** sẽ đặc tả chi tiết của các bảng trong database.

Bảng 12 Đặc tả bảng user

Tên trường	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
user_id	varchar(32)	Khóa chính	Định danh của người dùng
fullname	varchar(255)		Tên của người dùng

phone	varchar(255)		Số điện thoại di động
sex	varchar(255)		Giới tính
last_online	timestamp		Thời gian online gần nhất
status	enum(onl,off)		Trạng thái đang online hay offline
avatar	varchar(255)		Địa chỉ url của avatar
public_key	varchar(255)		Khóa công khai của người dùng
token_client	varchar(255)		Token thiết bị mà người dùng đăng nhập

Bảng 13 Đặc tả bảng message

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
msg_id	int	Khóa chính	
create_date	timestamp		Thời gian tin nhắn được tạo
type	enum(text,img,video,audio,other)		Thể loại của tin nhắn
status	enum(seen,default)		Trạng thái tin nhắn đã được xem hay chưa
content	text		Nội dung tin nhắn

Bảng 14 Đặc tả bảng user_mesage

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_msg_id	int	Khóa chính	Định danh
reciver_id	varchar(255)	Khóa ngoại	Định danh của người nhận
msg_id	int	Khóa ngoại	Định danh của tin nhắn
sender_id	varchar(32)	Khóa ngoại	Định danh của người gửi

Bảng 15 Đặc tả bảng user_friend

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user1_id	varchar(32)	Khóa chính, Khóa ngoại	
user2_id	varchar(32)	Khóa chính, Khóa ngoại	
create_date	timestamp		Ngày bản ghi được tạo

Bảng 16 Đặc tả bảng user_follower

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
follower_id	Int	Khóa chính	
sender_id	varchar(255)	Khóa ngoại	Người gửi
receiver_id	varchar(255)	Khóa ngoại	Người nhận
create_date	timestamp		Thời gian tạo

Bảng 17 Đặc tả bảng group

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
group_id	int	Khóa chính	Định danh
name	varchar(255)		Tên group
create_date	timestamp		Ngày tạo
avatar	text		Đường dẫn đến ảnh đại diện

Bảng 18 Đặc tả bảng group_message

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
group_msg_id	int	Khóa chính	Định danh
group_receiver	int	Khóa ngoại	Định danh của group nhận tin nhắn
msg_id	int	Khóa ngoại	Định danh của tin nhắn
sender_id	varchar(255)	Khóa ngoại	Định danh của người gửi

Bảng 19 Đặc tả bảng group_member

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
group_id	int	Khóa chính	Định danh group
user_id	varchar(32)	Khóa chính	Định danh của thành viên
role	enum(admin,default)		Vai trò trong một nhóm
create_date	Timestamp		Ngày tham gia

Bảng 20 Đặc tả bảng notification

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
notification_id	int	Khóa chính	Định danh
type	varchar(255)		Loại thông báo
content	varchar(255)		Nội dung
create_date	timestamp		Ngày tạo
user_id	varchar(255)	Khóa ngoại	Định danh người nhận
sender_id	varchar(255)	Khóa ngoại	Định danh của người gửi

status	int		Trạng thái

Bảng 21 Đặc tả bảng location

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	varchar(32)	Khóa chính	Định danh
latitude	double		Vĩ độ
longitude	double		Kinh độ
create_date	timestamp		Ngày cập nhật

Bảng 22 Đặc tả bảng contact

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	varchar(32)	Khóa chính	Định danh
phone	varchar(255)		Số điện thoại
name	varchar(255)		Tên trong danh bạ

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Thư viện và công cụ được sử dụng cho việc hoàn thành đồ án được mô tả trong **Bảng 23**:

Bảng 23 Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Android studio	https://developer.android.com/studio
IDE lập trình	WebStorm	https://www.jetbrains.com/webstorm
Quản lí cơ sở dữ liệu	Navicate	https://www.navicat.com/

Thư viện gọi video	Jitsi	https://jitsi.org/
Database	MySQL	https://www.mysql.com/

4.3.2 Kết quả đạt được

Từ nhưng gì đã tìm hiểu và phân tích, em đã xây dựng được ứng dụng trò chuyện nhắn tin an toàn với chức năng chính là nhắn tin an toàn và gọi điện thoại với bạn bè và cả hội nhóm.

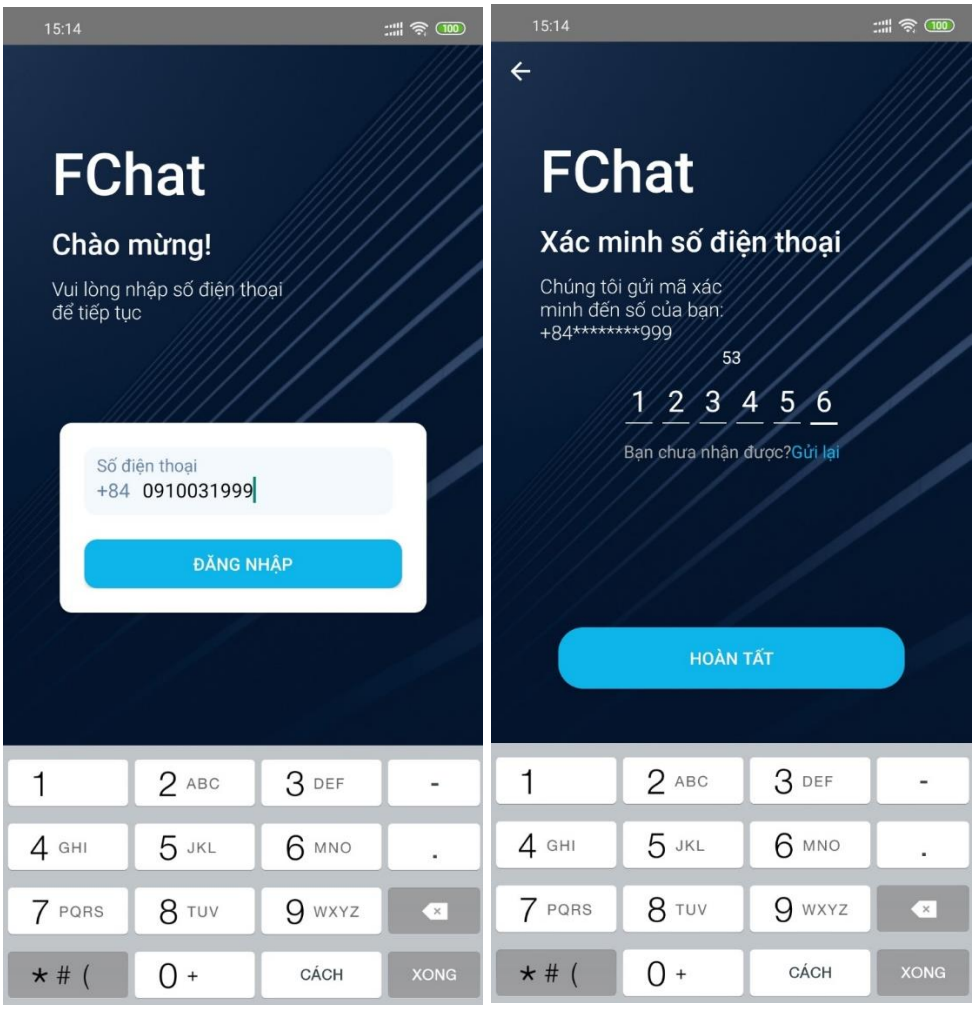
Và tất cả thống kê của em về mã nguồn được mô tả trong **Bảng 24**

Bảng 24 Thống kế ứng dụng

Thông tin	**Thống kê**
Số lớp trong mã nguồn client	116 lớp
Số file xml phía client	163 file
Tổng số file mã nguồn phía server	39 file
Dung lượng mã nguồn ứng dụng	9 MB

4.3.3 Minh hoạ các chức năng chính

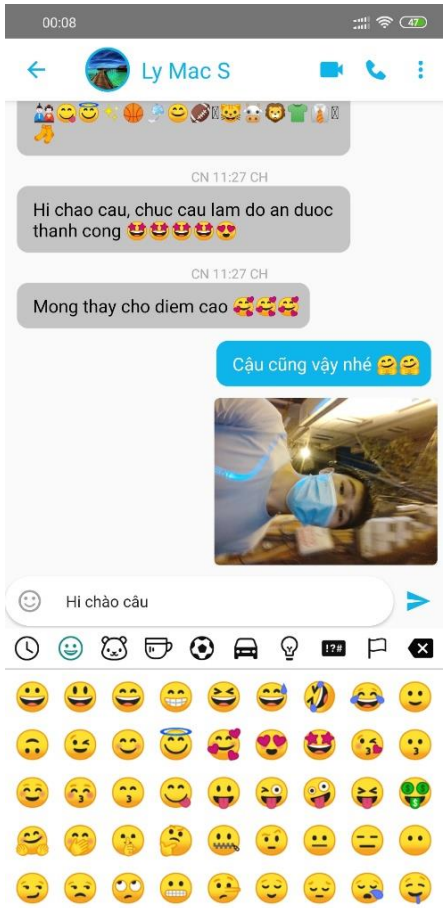
4.3.3.1 Đăng nhập



Hình 20 Giao diện đăng nhập

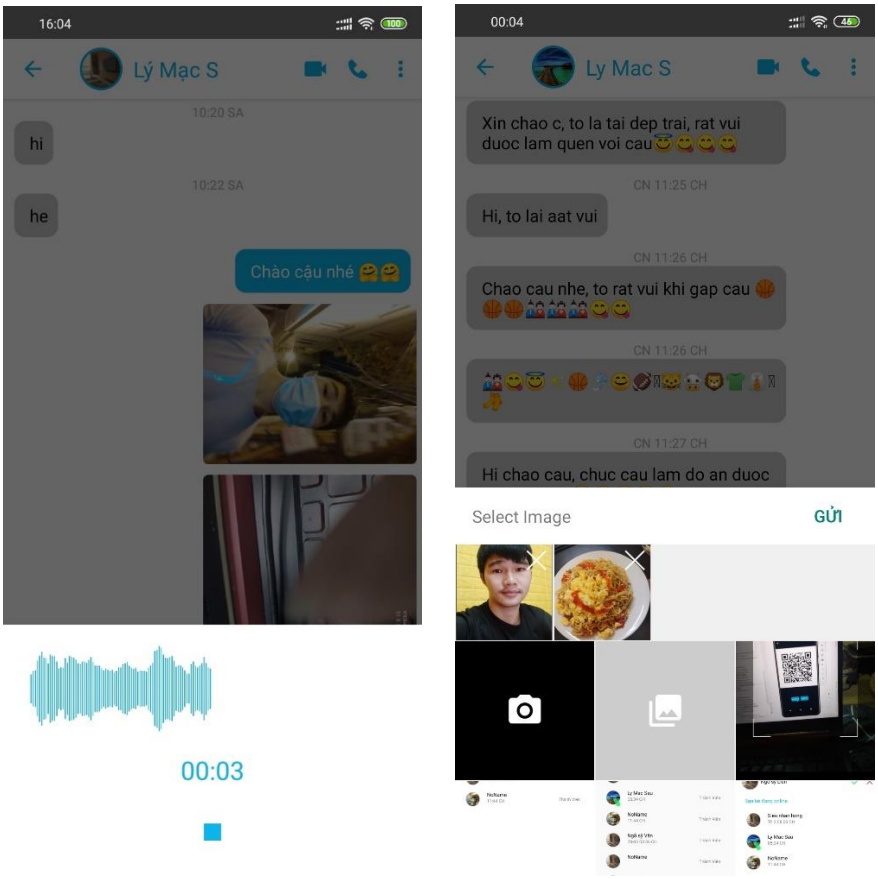
Ứng dụng sử dụng số điện thoại làm chức năng đăng nhập. **Hình 20** mô tả điều đó, với sự phối màu ảnh giao diện thân thiện, hiện đại, sẽ làm người dùng ấn tượng khi lần đầu vào ứng dụng.Sau khi nhập số điện thoại xong, sẽ được xác minh bằng mã OTP, gồm 6 chữ số, sẽ được gửi vào tin nhắn, bởi hiện nay số điện thoại gần như là chính chủ nên việc xác minh người dùng bằng mã OTP là hoàn toàn hợp lý.

4.3.3.2 Nhắn tin



Hình 21 Giao diện gửi tin nhắn văn bản

Khi người dùng sử dụng chức năng nhắn tin. Người dùng có thể soạn văn bản và sử dụng các icon ngộ nghĩnh để làm sinh động hơn nội dung tin nhắn truyền tải như **Hình 21**



Hình 22 giao diện gửi tin nhắn âm thanh

Hình 22 mô tả giao diện hai chức năng quan trong khác là gửi tin nhắn âm thanh và gửi hình ảnh. Khi gửi âm thanh thì người dùng có thể biết được âm thanh có được thu hay không qua biểu đồ âm thanh và thiều gian biểu thị đã thu trong bao lâu. Còn về chức năng gửi hình ảnh, người dùng có thể chọn nhiều hình ảnh hoặc dùng camera để chụp ảnh và gửi.

4.3.3.3 Gọi điện

Hình 23 Giao diện gọi điện thoại

Hình 23 Mô tả chức năng gọi điện thoại. Hình bên trái mô tả màn hình khi trong chế độ chờ đổ chuông, và khi người dùng nhận cuộc gọi sẽ sang màn hình được mô tả bên phải. Trong khi gọi điện, người dùng có thể bật tắt micro và bật tắt chia sẻ hình ảnh. Ngoài ra còn nhiều chức năng khác.

4.4 Kiểm thử

4.4.1 Kiểm thử tính tương thích

Bảng 25 Kiểm thử tính tương thích

Thiết bị	Thông số	Giao diện	Chức năng

Xiaomi MI 8 SE	Độ phân giải: Full HD+ (1080x2244 px) RAM: 6GB Hệ điều hành: Android 9	Đạt	Đạt
Google Pixel 3	Độ phân giải: Full HD+ (1080x2160 px) RAM: 4GB Hệ điều hành: Android 9	Đạt	Đạt
Motorola Nexus 6	Đội phân giải: 2K (1440x2560 px) RAM: 3GB Hệ điều hành: Android 5	Đạt	Đạt

4.4.1 Kiểm thử chức năng

Phần này em sẽ trình bày về kiểm thử của hai chức năng là nhắn tin (**Bảng 26**) và quản lý nhóm (**Bảng 27**).

Bảng 26 Chức năng nhắn tin

Chức năng	Test case	Kết quả
Xem tin nhắn	Không có khóa phiên	Đạt
	Có khóa phiên	Đạt
Gửi tin nhắn văn bản	Soạn tin nhắn chữ và gửi	Đạt
	Soạn tin nhắn gồm các icon và gửi	Đạt
Gửi tin nhắn dạng ảnh	Chọn một ảnh gửi	Đạt
	Chọn nhiều ảnh gửi	Đạt
	Chụp ảnh rồi gửi	Đạt
	Hủy gửi tin	Đạt
Gửi tin nhắn dạng voice	Thu âm giọng nói	Đạt
	Gửi bản đã thu	Đạt

	Hủy thu âm	Đạt
Xóa tin nhắn	Xóa tin nhắn của chín người dùng đã gửi	Đạt
Xóa cuộc hội thoại	Xóa cuộc hội thoại với bạn bè	Đạt

Bảng 27 Chức năng quản lý nhóm

Chức năng	Test case	Kết quả
Vào cuộc hội thoại nhóm	Đã có khóa phiên	Đạt
	Chưa có khóa phiên	Đạt
Thêm thành viên	Thêm bạn bè vào nhóm	Đạt
Chỉnh sửa thông tin nhóm	Thay ảnh đại diện	Đạt
	Đổi tên nhóm	Đạt
Xóa thành viên	Là quản trị viên nhóm	Đạt
	Không phải quản trị viên nhóm	Đạt
Xóa nhóm	Là quản trị viên nhóm	Đạt
	Không là quản trị viên nhóm	Đạt
Rời nhóm	Rời khỏi nhóm	Đạt

4.5 Triển khai

Ứng dụng được cài đặt trên thiết bị di động chạy hệ điều hành android. Server được triển khai trên sg.mdcsoftware.com.vn. Và **Bảng 28** dưới đây mô tả chi tiết các thiết bị đã được sử dụng để triển khai hệ thống.

Bảng 28 Thống kê các thiết bị triển khai hệ thống

Thiết bị	Vai trò	Cấu hình
	Server	Processor: Intel Core Processor (Haswell, no TSX)

		RAM: 1770 MB CPU: 2400 MHz Hệ điều hành: Cenos 8
Xiaomi MI8 SE	Client	Độ phân giải: Full HD+ (1080x2244 px) RAM: 6GB Hệ điều hành: Android 9

Chương 4 đã trình bày về cách phát triển và triển khai ứng dụng như thế nào, tiếp đến sẽ là chương 5 về các giải pháp và đóng góp nổi bật của ĐATN

Chương 5 Các giải pháp và đóng góp nổi bật

Chương 5 em sẽ trình bày về hai mục chính trong phần bảo mật của hệ thống ứng dụng, đó là về mã hóa các tin nhắn và việc truyền khóa phiên trong liên lạc.

5.1 Mã hóa các tin nhắn

5.1.1 Đặt vấn đề

Hiện nay, bảo vệ an toàn thông tin người dùng gần như là tất yếu khi phát triển các hệ thống, ứng dụng. Nguy cơ ứng dụng bị các đối tượng xấu lợi dụng và tấn công là tương đối cao. Các đối tượng xấu sẽ lợi dụng các lỗ hổng để ăn cắp thông tin người dùng, đọc lén nội dung tin nhắn,… để phục vụ cho mục đích xấu. Hay các nhà phát triển ứng dụng lại thu thập nhiều thông tin người dùng một cách trái phép để phục vụ cho lợi ích xấu, hay không được sự đồng ý của người dùng. Hiện ông chủ lớn của Facebook là Mark Zuckerberg đang vướng vào vụ kiện về việc thu thập trái phép về thông tin người dùng và bán chúng cho các tổ chức khác. Điều này đã làm giấy lên nỗi lo sợ trong niềm tin của người dùng khi sử dụng sản phẩm.

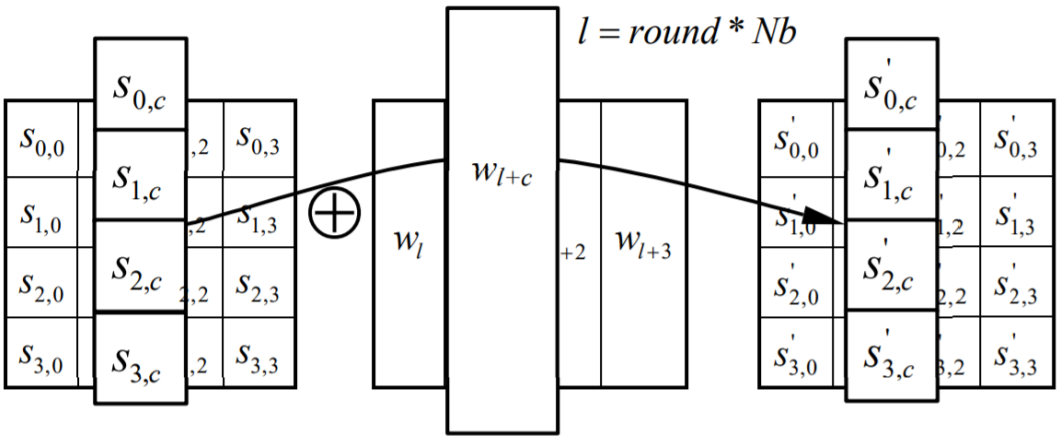
Hiện nay các ứng dụng nhắn tin an toàn như Telegram, Viber, WhatsApp hiện đang dẫn đầu trên thế giới về ứng dụng chat bảo mật. Với cơ chế là mã hóa end to end. Mã hóa end to end được hiểu là mã hóa đầu cuối, đây là phương thức đảm bảo dữ liệu truyền đi từ người gửi sẽ được bảo mật, vì vậy ngay cả bên trung gian là người cung cấp ứng dụng cũng không thể biết thông tin truyền đi là gì.

Và đồ án này em đặt tên là xây dựng ứng dụng trò chuyện nhắn tin an toàn, bởi vậy cần có một cơ chế bảo mật để bảo vệ người dùng khỏi những kẻ xấu muốn ăn cắp thông tin và cần phải có tốc mã hóa, giải mã nhanh.

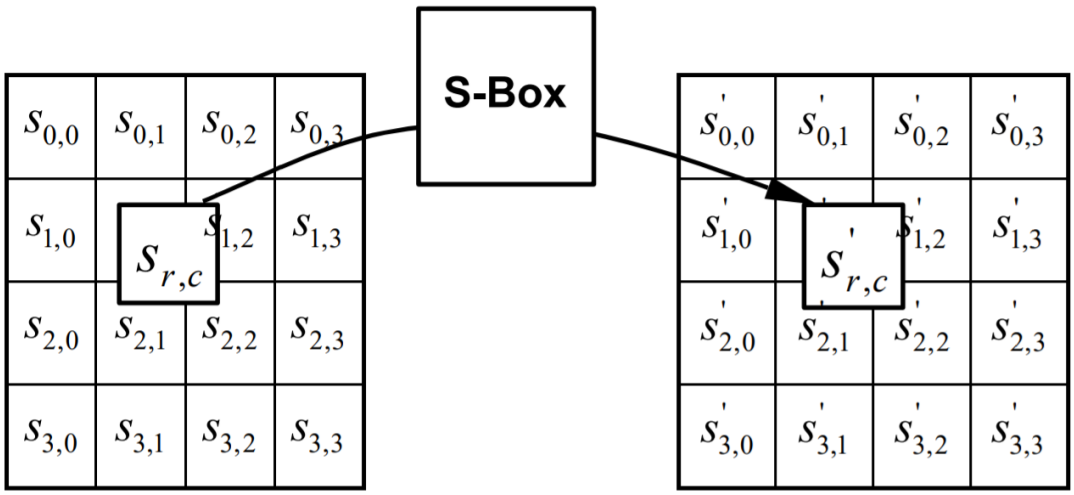
5.1.2 Giải pháp đưa ra và kết quả đạt được

Để bảo đảm an toàn thông tin khách hàng, em đã nghiên cứu các ứng dụng bảo mật như Telegram, Viber, WhatsApp,… và em quết định sử dụng mã hóa end to end. Để mã hóa cho các tin nhắn của người dùng end to end và cần một thuật toán mã hóa có tốc độ nhanh, em sử dụng thuật toán mã hóa là mã hóa đối xứng AES.

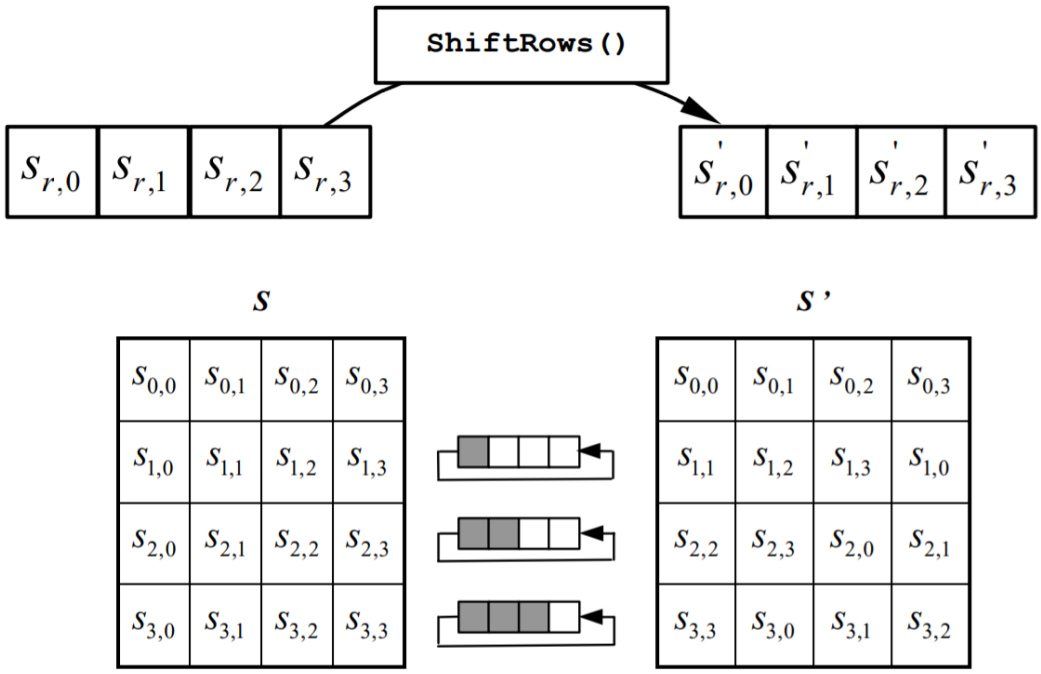
Thuật toán AES là một thuật toán mã hóa khối được chính phủ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa, gồm bốn bước cơ bản: (i) AddRoundKey — Mỗi cột của trạng thái đầu tiên lần lượt được kết hợp với một khóa con theo thứ tự từ đầu dãy khóa được mô tả ở **Hình 24**, (ii) SubBytes — đây là phép thế (phi tuyến) trong đó mỗi byte trong trạng thái sẽ được thế bằng một byte khác được mô tả ở **Hình 25**, (iii) ShiftRows — dịch chuyển, các hàng trong trạng thái được dịch vòng theo số bước khác nhau được mô tả ở **Hình 26**, (iv) MixColumns — quá trình trộn làm việc theo các cột trong khối theo một phép biến đổi tuyến tính được mô tả ở **Hình 27**.



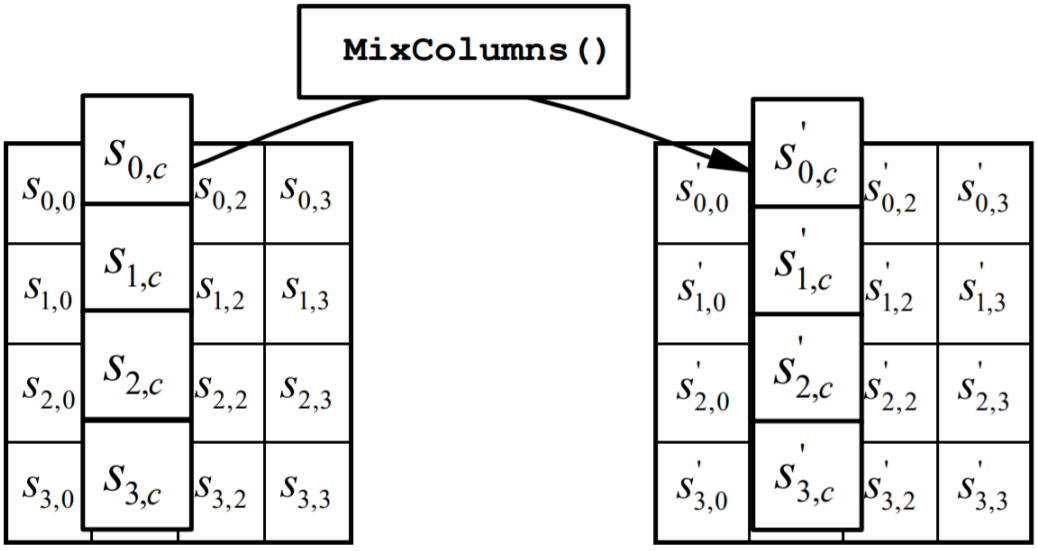
Hình 24 AddRoundKey



Hình 25 SubBytes

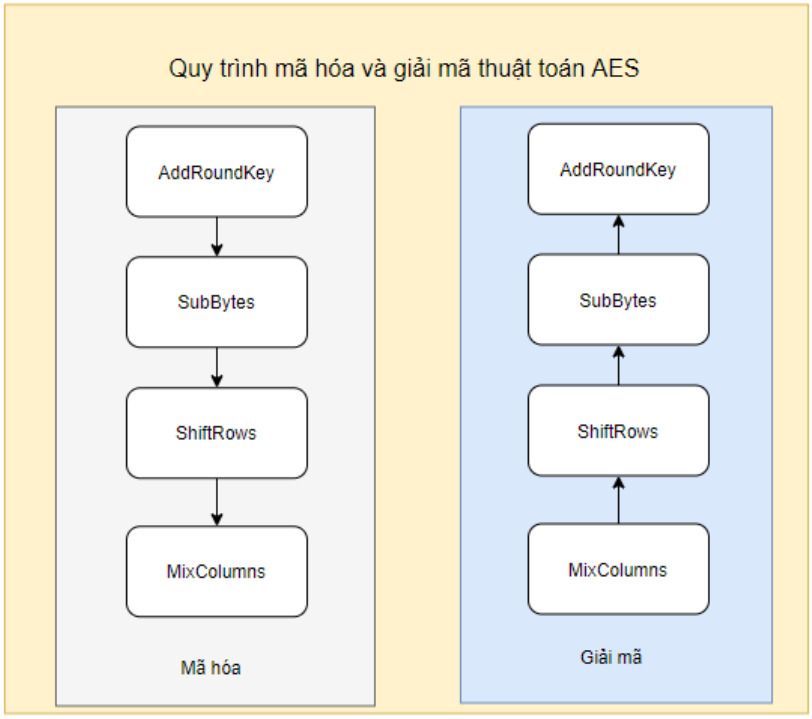


Hình 26 ShiftRows



Hình 27 MixColumns

Quy trình mã hóa và giải mã là ngược nhau, được mô tả như sau:



Hình 28 quy trình mã hóa và giải mã thuật toán AES

Mỗi khi người dùng muốn liên hệ với người dùng khác, khóa AES hay ở đồ án này của em sẽ có trách nhiệm là khóa phiên còn gọi là khóa phiên sẽ được tạo, khóa phiên sẽ được tạo ngẫu nhiên với độ dài 128 bit hoặc 192 bit hoặc 256 bit. Sau khi có khoá phiên thì các tin nhắn sẽ được mã hóa và truyền đi.

Kết quả đạt được là dữ liệu khi nhắn tin, tin nhắn vẫn được hiển thị đầu đủ cho người dùng (**Hình 29**), và server không thể biết được nội dung tin nhắn (**Hình 30**).

Hình 29 Tin nhắn hiển thị cho người dùng

	fKbFhdd4277m1IwZ0GHYUftYOHr9D3JsBcJgniteVDAb2h4GU24cwVPnVCLZX1q2GbCfKWOviCcrJdghbqYIiRNusprB/QUeYRUfKuoteJ8=
	u2NTiLC+nWX9zuiyaNR4Gar2HJXAmphI2AhEVHky6VQ=
	5j73Yqb5OBnqSudSWdFsfy94Xjtp7P/KbVzkQSegmf5ArtJ1f4htG96/aHH3REe7grEQaJpKnBC9bMfzeSri86bUJ5iIoOwGETzPaRCxzy8=
	LH3+3RrnFdJ7QhfKyNjY7YBHaiZnWPtE6Sp+Ke4p/XAUKNBmyhIrvzh0Xw/EaZSAjdrFQOpKg0iNCwlGR7iu8A==
	vs9D9nKTmjT+4WujfkkALPVcci/KSoKz4TWQBZd7kFY8H5iU2jicvxMICiQNudorEJKtau3aBdDLOCx+FFtU4sUqRnnEtFqVQYWDgi3gE8Y=
	DeBOf34BL+hQuTb7LQWOicvfUsXZKIgAGpJluxV19vfXQM7bLH9CF/k3PdcqVzEy
	NrgkJhAIINWGeijV0PUfitM6Air3sMiB8l1SWnYoDpI=

Hình 30 Nội dung tin nhắn được lưu ở database

5.2 Cơ chế truyền khóa phiên

5.2.1 Đặt vấn đề

Việc mã hóa tin nhắn bằng thuật toán AES phát sinh một vấn đề, đó là làm sao để cho hai người dùng muốn liên lạc với nhau mà có thể có khóa phiên chung. Nếu như người dùng khi thiết lập liên lạc với người dùng khác, khi tạo khóa phiên mà gửi trực tiếp qua server để gửi cho người dùng kia thì sẽ không đảm bảo được cơ chế mã hóa end to end như em đã trình bày ở phần 5.1, bởi vậy khi truyền khóa phiên ta cần được mã hóa.

5.2.2 Giải pháp đưa ra và kết quả đạt được

Sau khi tìm hiểu và nghiên cứu, em sử dụng thuật toán mã hóa RSA. Thuật toán RSA hay còn gọi là hệ mã hóa bất đối xứng, sử dụng một cặp key để mã hóa: (i) public key sẽ đưa cho người dùng khác để mã hóa thông điệp đến và (ii) private key người dùng sẽ giữ lại để giải mã thông điệp. Thuật toán RSA có bốn hoạt động chính: sinh khóa, chia sẻ public key, mã hóa và giải mã.

Khi sinh khóa ở thuật toán RSA, mấu chốt là ba số tự nhiên m, e, d sao cho:

$$m^{ed} \equiv m \; mod \; n$$

Công thức 1 Sinh khóa cho thuật toán RSA

Và sau cùng ta thu được public key là bộ số (n, e) và private key là bộ số (n, d).

Sau khi sinh khóa, public key sẽ được server lưu lại, còn private key sẽ được lưu lại tại client. Và khi người dùng yêu cầu public key sẽ lấy tại server.

Khi người dùng muốn trao đổi thông điệp sẽ yêu cầu ở server để lấy public key của người cần trao đổi và dùng public key đó để mã hóa thông điệp theo công thức:

$$c \equiv m^{e} \; mod \; n$$

Công thức 2 Mã hóa bằng thuật toán RSA

Khi ta có thông điệp M, ta cần chuyển nó thành số tự nhiên m trong khoảng (0, n) sao cho m,n là số nguyên tố cùng nhau, và sau khi áp dụng công thức trên ta sẽ được bản đã được mã hóa c và người dùng sẽ gửi c trên đường truyền.

Sau khi người dùng nhận được thông điệp đã được mã hóa c, ta sẽ giải mã bằng bộ số private key (n, d) theo công thức:

$$m \equiv c^d mod\ n$$

Công thức 3 Giải mã bằng thuật toán RSA

Mỗi khi một tài khoản được tạo hay được đăng nhập tại một thiết bị khác, người dùng sẽ phải sinh khóa RSA.Và cơ chế truyền khóa phiên như sau: khi người dùng A muốn tạo liên hệ với người dùng B, (i) A sẽ tạo khóa phiên, (ii) A yêu cầu server gửi public key của B, (iii) server sẽ xác nhận danh tính của A và gửi public key của B cho A, (iv) A sau khi có public key của B sẽ mã hóa khóa phiên bằng public key đó và gửi cho B, (v) B sau khi nhận khóa phiên đã được mã hóa thì sẽ giải mã bằng private key của mình và hai người đã có khóa phiên mà trong quá trình tuyền đi vẫn đảm bảo cơ chế end to end.

Chương 5 là các giải pháp và đóng góp nổi bật của hệ thống, sau đây sẽ là chương 6 về nhưng kết luận và hướng phát triển sau này.

Chương 6 Kết luận và hướng phát triển

Chương 6 cũng là chương cuối cùng, chương này em sẽ trình bày về nhưng kết luận rút ra được khi em thực hiện ĐATN này, và tương lai em sẽ phát triển ứng dụng này như thế nào cho hiệu quả hơn.

6.1 Kết luận

Sau quá trình tìm hiểu một cách nỗ lực và cùng với sự hướng dẫn tận tình của thầy ThS Lê Đức Trung, em đã xây dựng thành công ứng dụng trò chuyện nhắn tin an toàn. Ứng dụng đã giải quyết được các vấn đề lớn như: (i) cung cấp đầy đủ các chức năng cơ bản của một ứng dụng trò chuyện nhắn tin, (ii) bảo đảm an toàn nội dung tin nhắn của người dùng theo cơ chế end to end.

Trong quá trình thực hiện, ĐATN đã phân tích các yêu cầu từ phía người dùng đồng thời cũng tìm hiểu, đánh giá các ưu điểm và nhược điểm các ứng dụng tương tự đã có, và từ đó xây dựng được các chức năng cần thiết. Tuy nhiên, việc xây dựng ứng dụng trong thời gian ngắn, và một phần hạn chế về kỹ năng và kiến thức, nên ứng dụng vẫn còn nhiều thiếu sót và hạn chế. Nhưng sau này em vẫn sẽ tiếp tục phát triển đồ án này và thu thập thêm các phản hồi từ phía người dùng để hoàn thiện hơn.

6.2 Hướng phát triển

Để hoàn thiện thêm về các chức năng, em cần (i) tiếp tục thu thập các phản hồi phía người dùng, (ii) học hỏi thêm từ các chức năng hay từ các ứng dụng tương tự hiện nay.

Ngoài ra, ứng dụng nhắn tin an toàn cần phải phát triển để hỗ trợ các nền tảng khác nhau, tạo sự tiện lợi cho người dùng cài đặt và sử dụng.

Tài liệu tham khảo

[1] Morris J. Dworkin, Elaine B. Barker, James R. Nechvatal James Foti, Lawrence E. Bassham, E. Roback, James F. Dray Jr, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, United States National Institute of Standards and Technology (NIST) November 26, 2001.

[2] R.L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, CiteSeerX 10.1.1.607.2677.

[3] McLaughlin and Jenna, Democratic Debate Spawns Fantasy Talk on Encryption, The Intercept, Archived from the original on 23 December 2015

[4] Nguyễn Khanh Văn, Giáo trình cơ sở an toàn thông tin, Nhà xuất bản bách khoa Hà Nội, tái bản lần 3, 2019.