

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
KHOA CÔNG NGHỆ THÔNG TIN

---

BÁO CÁO  
PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

---

Đề tài:  
ỨNG DỤNG CHAT THỜI GIAN THỰC  
Realtime Chat Application

Sinh viên thực hiện:  
Nguyễn Hữu Mạnh  
MSSV: 20205213

Giảng viên hướng dẫn:  
Nguyễn Thị Thu Hương

Tháng 12, 2025

# Mục lục

<b>1</b>	<b>GIỚI THIỆU</b>	<b>6</b>
1.1	Đặt Vấn Đề . . . . .	6
1.2	Mục Tiêu Dự Án . . . . .	6
1.2.1	Mục tiêu tổng quát . . . . .	6
1.2.2	Mục tiêu cụ thể . . . . .	6
1.3	Phạm Vi Dự Án . . . . .	7
1.3.1	Trong phạm vi . . . . .	7
1.3.2	Ngoài phạm vi . . . . .	7
1.4	Công Nghệ Sử Dụng . . . . .	7
<b>2</b>	<b>PHÂN TÍCH YÊU CẦU</b>	<b>8</b>
2.1	Yêu Cầu Chức Năng . . . . .	8
2.1.1	Module Authentication (Xác thực) . . . . .	8
2.1.2	Module Friend Management (Quản lý bạn bè) . . . . .	8
2.1.3	Module Messaging (Nhắn tin) . . . . .	9
2.1.4	Module Conversation (Cuộc hội thoại) . . . . .	9
2.2	Yêu Cầu Phi Chức Năng . . . . .	9
<b>3</b>	<b>BIỂU ĐỒ USE CASE</b>	<b>10</b>
3.1	Tổng Quan . . . . .	10
3.2	Actors (Tác nhân) . . . . .	10
3.3	Danh Sách Use Cases . . . . .	10
3.3.1	Authentication Package . . . . .	10
3.3.2	Friend Management Package . . . . .	11
3.3.3	Messaging Package . . . . .	11
3.3.4	Conversation Package . . . . .	11
3.4	Use Case Relationships . . . . .	11
3.5	Đặc Tả Use Case Chi Tiết . . . . .	12
3.5.1	UC-01: Đăng ký . . . . .	12
3.5.2	UC-02: Đăng nhập . . . . .	12
<b>4</b>	<b>BIỂU ĐỒ SEQUENCE</b>	<b>13</b>
4.1	Tổng Quan . . . . .	13
4.2	Các Thành Phần Trong Biểu Đồ . . . . .	13
4.3	Sequence Diagram: Authentication . . . . .	13
4.3.1	Đăng Ký (Sign Up) . . . . .	13
4.3.2	Đăng Nhập (Sign In) . . . . .	14
4.3.3	Refresh Token . . . . .	14
4.3.4	Đăng Xuất (Sign Out) . . . . .	14

4.4	Sequence Diagram: Friend Management	15
4.4.1	Gửi Lời Mời Kết Bạn	15
4.4.2	Chấp Nhận Lời Mời	15
4.5	Sequence Diagram: Messaging	15
4.5.1	Gửi Tin Nhắn Trực Tiếp	15
4.6	Sequence Diagram: Conversation	16
4.6.1	Lấy Tin Nhắn (Pagination)	16
<b>5</b>	<b>BIỂU ĐỒ ACTIVITY</b>	<b>17</b>
5.1	Tổng Quan	17
5.2	Authentication Flow	17
5.2.1	Mô tả	17
5.2.2	Các bước chính	17
5.3	Send Message Flow	17
5.3.1	Mô tả	17
5.3.2	Các bước chính	18
5.4	Friend Request Flow	18
5.4.1	Mô tả	18
5.4.2	Swimlanes	18
5.4.3	Các bước chính	18
<b>6</b>	<b>BIỂU ĐỒ CLASS</b>	<b>20</b>
6.1	Tổng Quan	20
6.2	Models (Mô hình dữ liệu)	20
6.2.1	User	20
6.2.2	Session	20
6.2.3	Friend	21
6.2.4	FriendRequest	21
6.2.5	Conversation	21
6.2.6	Message	21
6.3	Controllers	22
6.4	Middleware	22
6.5	Quan Hệ Giữa Các Class	22
<b>7</b>	<b>THIẾT KẾ API</b>	<b>23</b>
7.1	Tổng Quan	23
7.2	Base URL	23
7.3	Authentication APIs	23
7.3.1	POST /auth/signup	23
7.3.2	POST /auth/signin	24
7.4	Friend APIs	24
7.5	Message APIs	24
7.6	Conversation APIs	24
<b>8</b>	<b>KẾT LUẬN</b>	<b>25</b>
8.1	Tổng Kết	25
8.2	Danh Sách Biểu Đồ	25
8.3	Hướng Phát Triển	26

<b>A</b>	<b>PHỤ LỤC</b>	<b>27</b>
A.1	Cấu Trúc Thư Mục . . . . .	27
A.2	Hướng Dẫn Xem Biểu Đồ PlantUML . . . . .	28
A.3	Hướng Dẫn Chạy Dự Án . . . . .	28
A.3.1	Backend . . . . .	28
A.3.2	Frontend . . . . .	28

# Danh sách hình vẽ

# Danh sách bảng

1.1	Công nghệ sử dụng trong dự án . . . . .	7
2.1	Yêu cầu chức năng - Authentication . . . . .	8
2.2	Yêu cầu chức năng - Friend Management . . . . .	8
2.3	Yêu cầu chức năng - Messaging . . . . .	9
2.4	Yêu cầu chức năng - Conversation . . . . .	9
2.5	Yêu cầu phi chức năng . . . . .	9
3.1	Danh sách Actors . . . . .	10
3.2	Use Cases - Authentication . . . . .	10
3.3	Use Cases - Friend Management . . . . .	11
3.4	Use Cases - Messaging . . . . .	11
3.5	Use Cases - Conversation . . . . .	11
3.6	Mối quan hệ giữa các Use Cases . . . . .	11
3.7	Đặc tả Use Case - Đăng ký . . . . .	12
3.8	Đặc tả Use Case - Đăng nhập . . . . .	12
4.1	Các thành phần trong Biểu đồ Sequence . . . . .	13
5.1	Swimlanes trong Friend Request Flow . . . . .	18
6.1	Class User - Thông tin người dùng . . . . .	20
6.2	Class Session - Phiên đăng nhập . . . . .	20
6.3	Class Friend - Quan hệ bạn bè . . . . .	21
6.4	Class FriendRequest - Lời mời kết bạn . . . . .	21
6.5	Class Conversation - Cuộc hội thoại . . . . .	21
6.6	Class Message - Tin nhắn . . . . .	21
6.7	Danh sách Controllers và Methods . . . . .	22
6.8	Middleware . . . . .	22
6.9	Quan hệ giữa các Models . . . . .	22
7.1	Authentication APIs . . . . .	23
7.2	Friend APIs . . . . .	24
7.3	Message APIs . . . . .	24
7.4	Conversation APIs . . . . .	24
8.1	Tổng hợp các biểu đồ . . . . .	25

# Chương 1

## GIỚI THIỆU

### 1.1 Đặt Vấn Đề

Trong thời đại công nghệ số hiện nay, nhu cầu giao tiếp và kết nối giữa mọi người ngày càng tăng cao. Các ứng dụng nhắn tin thời gian thực (Realtime Chat) đã trở thành công cụ không thể thiếu trong cuộc sống hàng ngày, từ việc liên lạc cá nhân đến công việc chuyên nghiệp.

Xuất phát từ nhu cầu thực tế đó, dự án **Realtime Chat Application** được phát triển nhằm xây dựng một ứng dụng chat hiện đại với các tính năng:

- Nhắn tin trực tiếp giữa hai người dùng
- Nhắn tin nhóm
- Quản lý bạn bè
- Xác thực an toàn với JWT

### 1.2 Mục Tiêu Dự Án

#### 1.2.1 Mục tiêu tổng quát

Xây dựng một ứng dụng chat thời gian thực hoàn chỉnh với đầy đủ các chức năng cơ bản, đảm bảo tính bảo mật và trải nghiệm người dùng tốt.

#### 1.2.2 Mục tiêu cụ thể

1. **Hệ thống xác thực:** Đăng ký, đăng nhập, đăng xuất với cơ chế JWT (Access Token + Refresh Token)
2. **Quản lý bạn bè:** Gửi, chấp nhận, từ chối lời mời kết bạn
3. **Nhắn tin:** Hỗ trợ chat trực tiếp (1-1) và chat nhóm
4. **Quản lý cuộc hội thoại:** Tạo, xem danh sách, xem lịch sử tin nhắn

## 1.3 Phạm Vi Dự Án

### 1.3.1 Trong phạm vi

- Xác thực người dùng (Authentication)
- Quản lý bạn bè (Friend Management)
- Nhắn tin (Messaging)
- Quản lý cuộc hội thoại (Conversation Management)

### 1.3.2 Ngoài phạm vi

- Video call / Voice call
- Chia sẻ file, hình ảnh
- Push notification
- End-to-end encryption

## 1.4 Công Nghệ Sử Dụng

Bảng 1.1: Công nghệ sử dụng trong dự án

Thành phần	Công nghệ	Phiên bản
Frontend	React	19.1.1
	TypeScript	5.9.3
	Vite	7.1.7
	TailwindCSS	4.1.14
	Zustand	5.0.8
Backend	Node.js	-
	Express.js	5.1.0
	Mongoose	8.19.0
	JWT	9.0.2
Database	MongoDB	-



## Chương 2

# PHÂN TÍCH YÊU CẦU

### 2.1 Yêu Cầu Chức Năng

#### 2.1.1 Module Authentication (Xác thực)

Bảng 2.1: Yêu cầu chức năng - Authentication

ID	Tên	Mô tả
FR-01	Đăng ký	Người dùng mới có thể tạo tài khoản với username, password, email, họ tên
FR-02	Đăng nhập	Người dùng xác thực bằng username và password, nhận về access token
FR-03	Đăng xuất	Người dùng thoát khỏi hệ thống, xóa session
FR-04	Refresh Token	Hệ thống tự động làm mới access token khi hết hạn

#### 2.1.2 Module Friend Management (Quản lý bạn bè)

Bảng 2.2: Yêu cầu chức năng - Friend Management

ID	Tên	Mô tả
FR-05	Gửi lời mời	Người dùng gửi lời mời kết bạn đến người dùng khác
FR-06	Chấp nhận lời mời	Người nhận đồng ý kết bạn
FR-07	Từ chối lời mời	Người nhận không đồng ý kết bạn
FR-08	Xem bạn bè	Hiển thị danh sách tất cả bạn bè
FR-09	Xem lời mời	Hiển thị lời mời đã gửi và nhận

### 2.1.3 Module Messaging (Nhắn tin)

Bảng 2.3: Yêu cầu chức năng - Messaging

ID	Tên	Mô tả
FR-10	Tin nhắn trực tiếp	Gửi tin nhắn 1-1 với người dùng khác
FR-11	Tin nhắn nhóm	Gửi tin nhắn trong group chat
FR-12	Xem tin nhắn	Xem lịch sử tin nhắn với phân trang

### 2.1.4 Module Conversation (Cuộc hội thoại)

Bảng 2.4: Yêu cầu chức năng - Conversation

ID	Tên	Mô tả
FR-13	Tạo hội thoại	Tạo conversation mới (direct hoặc group)
FR-14	Xem danh sách	Hiển thị tất cả conversations của người dùng
FR-15	Tạo nhóm	Tạo group chat với nhiều thành viên

## 2.2 Yêu Cầu Phi Chức Năng

Bảng 2.5: Yêu cầu phi chức năng

ID	Loại	Mô tả
NFR-01	Bảo mật	Password được mã hóa bằng bcrypt (salt = 10)
NFR-02	Bảo mật	Sử dụng JWT với access token (30 phút) và refresh token (14 ngày)
NFR-03	Hiệu năng	API response time < 500ms
NFR-04	Khả dụng	Hệ thống hoạt động 24/7
NFR-05	Tương thích	Hỗ trợ các trình duyệt hiện đại (Chrome, Firefox, Edge, Safari)

## Chương 3

# BIỂU ĐỒ USE CASE

### 3.1 Tổng Quan

Biểu đồ Use Case mô tả các chức năng mà hệ thống cung cấp từ góc nhìn của người dùng. Nó thể hiện mối quan hệ giữa các tác nhân (Actors) và các trường hợp sử dụng (Use Cases).

### 3.2 Actors (Tác nhân)

Bảng 3.1: Danh sách Actors

Actor	Mô tả
Guest	Người dùng chưa đăng nhập. Chỉ có thể thực hiện đăng ký hoặc đăng nhập.
User	Người dùng đã đăng nhập. Có thể sử dụng đầy đủ các chức năng của hệ thống.

### 3.3 Danh Sách Use Cases

#### 3.3.1 Authentication Package

Bảng 3.2: Use Cases - Authentication

ID	Use Case	Actor	Mô tả
UC-01	Đăng ký	Guest	Tạo tài khoản mới
UC-02	Đăng nhập	Guest	Xác thực vào hệ thống
UC-03	Đăng xuất	User	Thoát khỏi hệ thống
UC-04	Làm mới token	User	Lấy access token mới

### 3.3.2 Friend Management Package

Bảng 3.3: Use Cases - Friend Management

ID	Use Case	Actor	Mô tả
UC-05	Gửi lời mời kết bạn	User	Gửi yêu cầu kết bạn
UC-06	Chấp nhận lời mời	User	Đồng ý kết bạn
UC-07	Từ chối lời mời	User	Không đồng ý kết bạn
UC-08	Xem danh sách bạn bè	User	Hiển thị bạn bè
UC-09	Xem lời mời kết bạn	User	Hiển thị lời mời

### 3.3.3 Messaging Package

Bảng 3.4: Use Cases - Messaging

ID	Use Case	Actor	Mô tả
UC-10	Gửi tin nhắn trực tiếp	User	Chat 1-1
UC-11	Gửi tin nhắn nhóm	User	Chat trong group
UC-12	Xem tin nhắn	User	Đọc lịch sử chat

### 3.3.4 Conversation Package

Bảng 3.5: Use Cases - Conversation

ID	Use Case	Actor	Mô tả
UC-13	Tạo cuộc hội thoại	User	Bắt đầu chat mới
UC-14	Xem danh sách hội thoại	User	Hiển thị conversations
UC-15	Tạo nhóm chat	User	Tạo group mới

## 3.4 Use Case Relationships

Bảng 3.6: Mối quan hệ giữa các Use Cases

Relationship	Loại	Giải thích
Gửi tin nhắn trực tiếp → Tạo cuộc hội thoại	«include»	Khi gửi DM, hệ thống tự động tạo conversation nếu chưa có
Gửi tin nhắn nhóm → Xem danh sách hội thoại	«include»	Cần chọn group từ danh sách trước khi gửi
Chấp nhận lời mời → Xem lời mời kết bạn	«extend»	Mở rộng từ việc xem lời mời
Từ chối lời mời → Xem lời mời kết bạn	«extend»	Mở rộng từ việc xem lời mời

## 3.5 Đặc Tả Use Case Chi Tiết

### 3.5.1 UC-01: Đăng ký

Bảng 3.7: Đặc tả Use Case - Đăng ký

<b>Use Case ID</b>	UC-01
<b>Tên</b>	Đăng ký
<b>Actor</b>	Guest
<b>Mô tả</b>	Người dùng tạo tài khoản mới trong hệ thống
<b>Tiền điều kiện</b>	Người dùng chưa có tài khoản
<b>Hậu điều kiện</b>	Tài khoản được tạo thành công
<b>Luồng chính</b>	<ol style="list-style-type: none"> <li>1. Guest truy cập trang đăng ký</li> <li>2. Guest nhập username, password, email, firstName, lastName</li> <li>3. Hệ thống kiểm tra username chưa tồn tại</li> <li>4. Hệ thống mã hóa password</li> <li>5. Hệ thống tạo user mới</li> <li>6. Hệ thống chuyển đến trang đăng nhập</li> </ol>
<b>Luồng phụ</b>	3a. Username đã tồn tại: Hiển thị lỗi "Username đã tồn tại"

### 3.5.2 UC-02: Đăng nhập

Bảng 3.8: Đặc tả Use Case - Đăng nhập

<b>Use Case ID</b>	UC-02
<b>Tên</b>	Đăng nhập
<b>Actor</b>	Guest
<b>Mô tả</b>	Người dùng xác thực vào hệ thống
<b>Tiền điều kiện</b>	Người dùng đã có tài khoản
<b>Hậu điều kiện</b>	Người dùng được xác thực, nhận access token
<b>Luồng chính</b>	<ol style="list-style-type: none"> <li>1. Guest truy cập trang đăng nhập</li> <li>2. Guest nhập username và password</li> <li>3. Hệ thống tìm user theo username</li> <li>4. Hệ thống so sánh password</li> <li>5. Hệ thống tạo access token (JWT)</li> <li>6. Hệ thống tạo refresh token</li> <li>7. Hệ thống lưu session</li> <li>8. Hệ thống trả về access token và cookie</li> </ol>
<b>Luồng phụ</b>	<ol style="list-style-type: none"> <li>3a. User không tồn tại: Hiển thị lỗi</li> <li>4a. Password không đúng: Hiển thị lỗi</li> </ol>

## Chương 4

# BIỂU ĐỒ SEQUENCE

### 4.1 Tổng Quan

Biểu đồ Sequence (Biểu đồ tuần tự) mô tả sự tương tác giữa các đối tượng theo thứ tự thời gian. Nó thể hiện luồng xử lý của hệ thống từ khi nhận request đến khi trả về response.

### 4.2 Các Thành Phần Trong Biểu Đồ

Bảng 4.1: Các thành phần trong Biểu đồ Sequence

Thành phần	Ký hiệu	Mô tả
User	Actor	Người dùng tương tác với hệ thống
Frontend	Participant	Ứng dụng React chạy trên trình duyệt
Backend	Participant	Server Express.js xử lý API
MongoDB	Database	Cơ sở dữ liệu lưu trữ

### 4.3 Sequence Diagram: Authentication

#### 4.3.1 Đăng Ký (Sign Up)

**Mô tả:** Luồng xử lý khi người dùng đăng ký tài khoản mới.

**Các bước:**

1. User nhập thông tin đăng ký (username, password, email, firstName, lastName)
2. Frontend gửi request `POST /api/auth/signup`
3. Backend kiểm tra username đã tồn tại chưa trong database
4. Nếu chưa tồn tại:
  - Backend hash password bằng bcrypt (salt = 10)
  - Backend tạo user mới trong MongoDB
  - Backend trả về 204 No Content

5. Nếu đã tồn tại: Backend trả về 409 **Conflict**
6. Frontend hiển thị kết quả cho User

### 4.3.2 Đăng Nhập (Sign In)

**Mô tả:** Luồng xử lý khi người dùng đăng nhập.

**Các bước:**

1. User nhập username và password
2. Frontend gửi request `POST /api/auth/signin`
3. Backend tìm user theo username trong database
4. Backend so sánh password với hash đã lưu (`bcrypt.compare`)
5. Nếu password đúng:
  - Tạo Access Token (JWT, TTL: 30 phút)
  - Tạo Refresh Token (random bytes, TTL: 14 ngày)
  - Lưu session vào database
  - Trả về access token + Set Cookie (refresh token)
6. Nếu password sai: Trả về 401 **Unauthorized**

### 4.3.3 Refresh Token

**Mô tả:** Luồng xử lý khi access token hết hạn.

**Các bước:**

1. Frontend gửi request `POST /api/auth/refresh-token` với cookie chứa refresh token
2. Backend tìm session theo refresh token
3. Kiểm tra session còn hạn không
4. Nếu hợp lệ: Tạo access token mới và trả về
5. Nếu không hợp lệ: Trả về 403 **Forbidden**

### 4.3.4 Đăng Xuất (Sign Out)

**Mô tả:** Luồng xử lý khi người dùng đăng xuất.

**Các bước:**

1. User click nút đăng xuất
2. Frontend gửi request `POST /api/auth/signout`
3. Backend xóa session khỏi database

4. Backend xóa cookie chứa refresh token
5. Backend trả về 204 No Content
6. Frontend xóa access token và chuyển về trang đăng nhập

## 4.4 Sequence Diagram: Friend Management

### 4.4.1 Gửi Lời Mời Kết Bạn

Các bước:

1. User A gửi lời mời đến User B
2. Frontend gửi POST `/api/friends/request { to, message }`
3. Backend kiểm tra:
  - Không gửi cho chính mình
  - User B tồn tại
  - Chưa là bạn bè
  - Chưa có lời mời đang chờ
4. Nếu hợp lệ: Tạo FriendRequest và trả về 201 Created
5. Nếu không hợp lệ: Trả về lỗi tương ứng

### 4.4.2 Chấp Nhận Lời Mời

Các bước:

1. User B chấp nhận lời mời từ User A
2. Frontend gửi POST `/api/friends/accept/:requestId`
3. Backend kiểm tra quyền (chỉ người nhận mới được chấp nhận)
4. Backend tạo Friend relationship
5. Backend xóa FriendRequest
6. Backend trả về thông tin bạn mới

## 4.5 Sequence Diagram: Messaging

### 4.5.1 Gửi Tin Nhắn Trực Tiếp

Các bước:

1. User nhập nội dung tin nhắn
2. Frontend gửi POST `/api/messages/direct { recipientId, content }`



3. Backend kiểm tra conversation đã tồn tại chưa
4. Nếu chưa có: Tạo conversation mới với type = "direct"
5. Backend tạo message mới
6. Backend cập nhật lastMessage của conversation
7. Backend trả về message đã tạo

## 4.6 Sequence Diagram: Conversation

### 4.6.1 Lấy Tin Nhắn (Pagination)

#### Các bước:

1. User mở một conversation
2. Frontend gửi GET `/api/conversations/:id/messages?limit=50&cursor=...`
3. Backend query messages với cursor-based pagination
4. Backend sắp xếp theo thời gian (mới nhất trước)
5. Backend trả về messages + nextCursor cho lần query tiếp theo

# Chương 5

## BIỂU ĐỒ ACTIVITY

### 5.1 Tổng Quan

Biểu đồ Activity (Biểu đồ hoạt động) mô tả luồng công việc (workflow) của hệ thống. Nó thể hiện các hoạt động, điều kiện rẽ nhánh, và luồng xử lý song song.

### 5.2 Authentication Flow

#### 5.2.1 Mô tả

Biểu đồ này mô tả luồng xác thực khi người dùng mở ứng dụng, bao gồm các trường hợp: đã đăng nhập, token hết hạn, và cần đăng nhập mới.

#### 5.2.2 Các bước chính

1. Mở ứng dụng
2. Kiểm tra đã đăng nhập?
  - **Có:** Kiểm tra access token
    - Token hợp lệ → Vào trang chính
    - Token hết hạn → Gọi refresh token
      - \* Refresh thành công → Vào trang chính
      - \* Refresh thất bại → Chuyển đến đăng nhập
  - **Không:** Hiển thị trang đăng nhập
    - Chọn đăng nhập → Nhập thông tin → Xác thực
    - Chọn đăng ký → Nhập thông tin → Tạo tài khoản

### 5.3 Send Message Flow

#### 5.3.1 Mô tả

Biểu đồ này mô tả luồng gửi tin nhắn, bao gồm validation, xử lý song song (UI và server), và xử lý lỗi.

### 5.3.2 Các bước chính

1. Mở cuộc hội thoại
2. Nhập nội dung tin nhắn
3. Kiểm tra nội dung rỗng?
  - Có → Hiển thị cảnh báo → Kết thúc
  - Không → Tiếp tục
4. Gửi tin nhắn đến server
5. **Xử lý song song (Fork):**
  - Nhánh 1: Hiển thị tin nhắn trạng thái "pending" trên UI
  - Nhánh 2: Server xử lý, lưu database, cập nhật conversation
6. **Đồng bộ (Join):** Kiểm tra kết quả
  - Thành công → Cập nhật trạng thái "sent"
  - Thất bại → Hiển thị lỗi, cho phép gửi lại

## 5.4 Friend Request Flow

### 5.4.1 Mô tả

Biểu đồ này sử dụng Swimlane để mô tả luồng gửi và xử lý lời mời kết bạn giữa User A, Server, và User B.

### 5.4.2 Swimlanes

Bảng 5.1: Swimlanes trong Friend Request Flow

Lane	Hoạt động
User A	Tìm kiếm người dùng, chọn User B, gửi lời mời, nhận thông báo kết quả
Server	Nhận yêu cầu, kiểm tra điều kiện, tạo FriendRequest, thông báo User B, xử lý chấp nhận/từ chối
User B	Nhận thông báo, chọn chấp nhận hoặc từ chối, cập nhật danh sách bạn bè

### 5.4.3 Các bước chính

1. **User A:** Tìm kiếm và chọn User B
2. **User A:** Gửi lời mời kết bạn
3. **Server:** Nhận và kiểm tra yêu cầu

- Đã là bạn bè? → Trả về lỗi
  - Đã có lời mời? → Trả về lỗi
4. **Server:** Tạo FriendRequest và thông báo User B
  5. **User B:** Nhận thông báo và chọn hành động
    - **Chấp nhận:** Server tạo Friend relationship, xóa request
    - **Từ chối:** Server xóa request
  6. **User A:** Nhận thông báo kết quả

# Chương 6

## BIỂU ĐỒ CLASS

### 6.1 Tổng Quan

Biểu đồ Class mô tả cấu trúc tĩnh của hệ thống, bao gồm các lớp (classes), thuộc tính (attributes), phương thức (methods), và mối quan hệ giữa chúng.

### 6.2 Models (Mô hình dữ liệu)

#### 6.2.1 User

Bảng 6.1: Class User - Thông tin người dùng

Thuộc tính	Kiểu	Mô tả
_id	ObjectId	ID duy nhất (MongoDB)
username	String	Tên đăng nhập (unique)
hashedPassword	String	Mật khẩu đã mã hóa
email	String	Địa chỉ email
displayName	String	Tên hiển thị
avatarUrl	String	URL ảnh đại diện
createdAt	Date	Thời gian tạo
updatedAt	Date	Thời gian cập nhật

#### 6.2.2 Session

Bảng 6.2: Class Session - Phiên đăng nhập

Thuộc tính	Kiểu	Mô tả
_id	ObjectId	ID duy nhất
userId	ObjectId	Tham chiếu đến User
refreshToken	String	Token làm mới
expiresAt	Date	Thời gian hết hạn
createdAt	Date	Thời gian tạo

### 6.2.3 Friend

Bảng 6.3: Class Friend - Quan hệ bạn bè

Thuộc tính	Kiểu	Mô tả
_id	ObjectId	ID duy nhất
userA	ObjectId	User thứ nhất
userB	ObjectId	User thứ hai
createdAt	Date	Thời gian kết bạn

### 6.2.4 FriendRequest

Bảng 6.4: Class FriendRequest - Lời mời kết bạn

Thuộc tính	Kiểu	Mô tả
_id	ObjectId	ID duy nhất
from	ObjectId	Người gửi
to	ObjectId	Người nhận
message	String	Lời nhắn kèm theo
createdAt	Date	Thời gian gửi

### 6.2.5 Conversation

Bảng 6.5: Class Conversation - Cuộc hội thoại

Thuộc tính	Kiểu	Mô tả
_id	ObjectId	ID duy nhất
type	String	Loại: "direct" hoặc "group"
participants	Array	Danh sách thành viên
group	Object	Thông tin nhóm (nếu có)
lastMessage	Object	Tin nhắn cuối cùng
lastMessageAt	Date	Thời gian tin nhắn cuối
seenBy	Array	Danh sách đã xem
unreadCounts	Map	Số tin chưa đọc theo user

### 6.2.6 Message

Bảng 6.6: Class Message - Tin nhắn

Thuộc tính	Kiểu	Mô tả
_id	ObjectId	ID duy nhất
conversationId	ObjectId	Tham chiếu đến Conversation
senderId	ObjectId	Người gửi
content	String	Nội dung tin nhắn
createdAt	Date	Thời gian gửi
updatedAt	Date	Thời gian cập nhật

## 6.3 Controllers

Bảng 6.7: Danh sách Controllers và Methods

Controller	Method	Chức năng
AuthController	signUp() signIn() signOut() refreshToken()	Đăng ký tài khoản Đăng nhập Đăng xuất Làm mới token
UserController	authMe()	Lấy thông tin user hiện tại
FriendController	sendFriendRequest() acceptFriendRequest() declineFriendRequest() getAllFriends() getFriendRequests()	Gửi lời mời Chấp nhận lời mời Từ chối lời mời Lấy danh sách bạn bè Lấy lời mời
MessageController	sendDirectMessage() sendGroupMessage()	Gửi tin nhắn 1-1 Gửi tin nhắn nhóm
ConversationController	createConversation() getConversations() getMessages()	Tạo hội thoại Lấy danh sách Lấy tin nhắn

## 6.4 Middleware

Bảng 6.8: Middleware

Middleware	Method	Chức năng
AuthMiddleware	protectedRoute()	Xác thực JWT token, gắn user vào request

## 6.5 Quan Hệ Giữa Các Class

Bảng 6.9: Quan hệ giữa các Models

Class A	Quan hệ	Class B	Mô tả
User	1:N	Session	User có nhiều sessions
User	N:M	Friend	User có nhiều bạn bè
User	1:N	FriendRequest	User gửi/nhận nhiều lời mời
User	N:M	Conversation	User tham gia nhiều hội thoại
User	1:N	Message	User gửi nhiều tin nhắn
Conversation	1:N	Message	Hội thoại chứa nhiều tin nhắn

# Chương 7

## THIẾT KẾ API

### 7.1 Tổng Quan

API được thiết kế theo chuẩn RESTful, sử dụng JSON làm định dạng dữ liệu trao đổi.

### 7.2 Base URL

```
1 http://localhost:5001/api
```

### 7.3 Authentication APIs

Bảng 7.1: Authentication APIs

Method	Endpoint	Mô tả	Auth
POST	/auth/signup	Đăng ký	Không
POST	/auth/signin	Đăng nhập	Không
POST	/auth/signout	Đăng xuất	Có
POST	/auth/refresh-token	Làm mới token	Cookie

#### 7.3.1 POST /auth/signup

Request Body:

```
1 {  
2   "username": "string",  
3   "password": "string",  
4   "email": "string",  
5   "firstName": "string",  
6   "lastName": "string"  
7 }
```

Response:

- 204 No Content - Thành công
- 400 Bad Request - Thiếu thông tin
- 409 Conflict - Username đã tồn tại



### 7.3.2 POST /auth/signin

Request Body:

```
1 {
2   "username": "string",
3   "password": "string"
4 }
```

Response:

```
1 {
2   "message": "User [displayName] da logged in!",
3   "accessToken": "jwt_token_here"
4 }
```

## 7.4 Friend APIs

Bảng 7.2: Friend APIs

Method	Endpoint	Mô tả	Auth
POST	/friends/request	Gửi lời mời	Có
POST	/friends/accept/:id	Chấp nhận	Có
POST	/friends/decline/:id	Từ chối	Có
GET	/friends	Danh sách bạn bè	Có
GET	/friends/requests	Danh sách lời mời	Có

## 7.5 Message APIs

Bảng 7.3: Message APIs

Method	Endpoint	Mô tả	Auth
POST	/messages/direct	Gửi tin nhắn 1-1	Có
POST	/messages/group	Gửi tin nhắn nhóm	Có

## 7.6 Conversation APIs

Bảng 7.4: Conversation APIs

Method	Endpoint	Mô tả	Auth
POST	/conversations	Tạo hội thoại	Có
GET	/conversations	Danh sách hội thoại	Có
GET	/conversations/:id/messages	Lấy tin nhắn	Có

## Chương 8

# KẾT LUẬN

### 8.1 Tổng Kết

Báo cáo đã trình bày chi tiết việc phân tích và thiết kế hệ thống cho ứng dụng **Realtime Chat Application**, bao gồm:

1. **Phân tích yêu cầu:** Xác định 15 yêu cầu chức năng và 5 yêu cầu phi chức năng
2. **Biểu đồ Use Case:** Mô tả 15 use cases với 2 actors
3. **Biểu đồ Sequence:** 4 biểu đồ chính (Authentication, Friend, Messaging, Conversation)
4. **Biểu đồ Activity:** 3 biểu đồ (Authentication Flow, Send Message Flow, Friend Request Flow)
5. **Biểu đồ Class:** 6 models, 5 controllers, 1 middleware
6. **Thiết kế API:** RESTful API với 13 endpoints

### 8.2 Danh Sách Biểu Đồ

Bảng 8.1: Tổng hợp các biểu đồ

#	Loại	Tên	File
1	Use Case	Use Case Diagram	diagrams.puml
2	Activity	Authentication Flow	diagrams.puml
3	Activity	Send Message Flow	diagrams.puml
4	Activity	Friend Request Flow	diagrams.puml
5	Class	Class Diagram	diagrams.puml
6	Sequence	Authentication	sequence.puml
7	Sequence	Friend Management	sequence.puml
8	Sequence	Messaging	sequence.puml
9	Sequence	Conversation	sequence.puml

## 8.3 Hướng Phát Triển

Trong tương lai, dự án có thể mở rộng thêm các tính năng:

- **Realtime với WebSocket/Socket.io** - Nhận tin nhắn tức thì
- **Push Notification** - Thông báo đẩy
- **Media Sharing** - Chia sẻ hình ảnh, file
- **Voice/Video Call** - Gọi điện, video call
- **End-to-End Encryption** - Mã hóa đầu cuối
- **Message Reactions** - React tin nhắn
- **Typing Indicators** - Hiển thị đang gõ
- **Read Receipts** - Trạng thái đã xem

# Phụ lục A

## PHỤ LỤC

### A.1 Cấu Trúc Thư Mục

```
1 RealtimeChatApp/  
2 |-- backend/  
3 |   |-- package.json  
4 |   |-- src/  
5 |       |-- server.js  
6 |       |-- controllers/  
7 |           |-- authController.js  
8 |           |-- conversationController.js  
9 |           |-- friendController.js  
10 |          |-- messageController.js  
11 |          |-- userController.js  
12 |       |-- libs/  
13 |           |-- db.js  
14 |       |-- middlewares/  
15 |           |-- authMiddleware.js  
16 |       |-- models/  
17 |           |-- Conversation.js  
18 |           |-- Friend.js  
19 |           |-- FriendRequest.js  
20 |           |-- Message.js  
21 |           |-- Session.js  
22 |           |-- User.js  
23 |       |-- routes/  
24 |           |-- authRoute.js  
25 |           |-- conversationRoute.js  
26 |           |-- friendRoute.js  
27 |           |-- messageRoute.js  
28 |           |-- userRoute.js  
29 |-- frontend/  
30 |   |-- package.json  
31 |   |-- src/  
32 |       |-- App.tsx  
33 |       |-- components/  
34 |       |-- pages/  
35 |       |-- services/  
36 |       |-- stores/  
37 |-- docs/  
38 |   |-- diagrams.puml  
39 |   |-- sequence.puml
```

```
40 | -- README.md
41 | -- report.tex
```

## A.2 Hướng Dẫn Xem Biểu Đồ PlantUML

1. **VS Code:** Cài extension "PlantUML" và mở file .puml
2. **Online:** Truy cập <https://www.plantuml.com/plantuml>
3. **CLI:** Chạy lệnh `plantuml diagrams.puml`

## A.3 Hướng Dẫn Chạy Dự Án

### A.3.1 Backend

```
1 cd backend
2 npm install
3 npm run dev
```

### A.3.2 Frontend

```
1 cd frontend
2 npm install
3 npm run dev
```

# Tài liệu tham khảo

- [1] React Documentation, <https://react.dev/>
- [2] Express.js Documentation, <https://expressjs.com/>
- [3] MongoDB Documentation, <https://docs.mongodb.com/>
- [4] JSON Web Token (JWT), <https://jwt.io/>
- [5] PlantUML Documentation, <https://plantuml.com/>