**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**TÀI LIỆU Phân tích & Thiết Kế**

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**MỤC LỤC**

[1 Bảng đánh giá thành viên 2](#_Toc150415564)

[2 Sơ đồ lớp(Class Diagram) 3](#_Toc150415565)

[3 Thiết kế kiến trúc 4](#_Toc150415566)

[4 Thiết kế dữ liệu 5](#_Toc150415567)

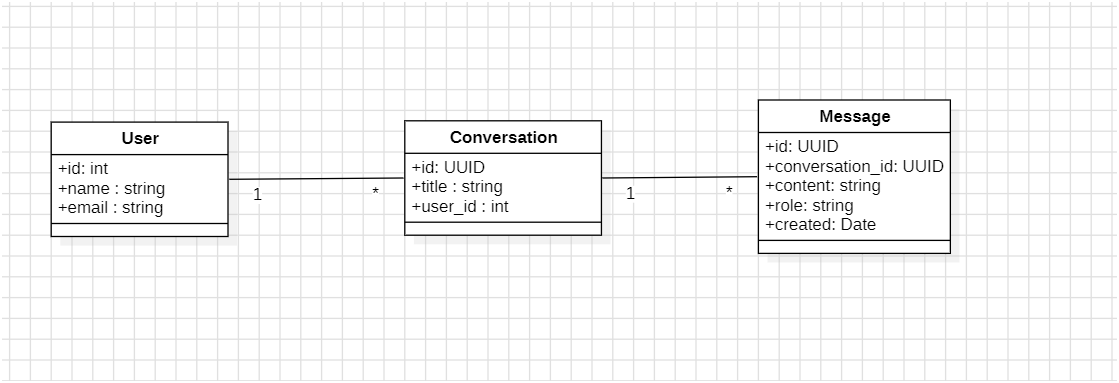
[4.1 Sơ đồ dữ liệu 5](#_Toc150415568)

[4.2 Đặc tả dữ liệu 5](#_Toc150415569)

# Bảng đánh giá thành viên

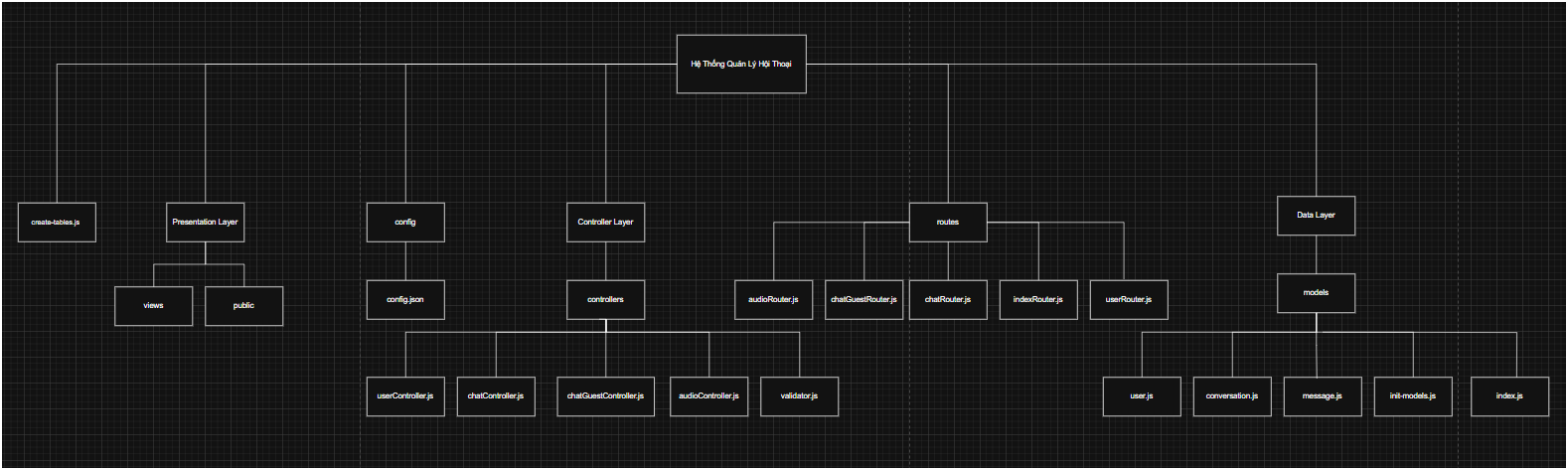
|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **% đóng góp (tối đa 100%)** | **Chữ ký** |
| 23880008 | Nguyễn Quốc Bình | 25% |  |
| 23880094 | Võ Xuân Tú | 25% |  |
| 23880047 | Võ Vĩnh Nghi | 25% |  |
| 21810226 | Phạm Hoàng Nhật Trường | 25% |  |

# Sơ đồ lớp (Class Diagram)



# Thiết kế kiến trúc

**cây phân rã hệ thống:**

**

**Điểm đặc biệt trong kiến trúc tổng thể của hệ thống:**

1. **Áp dụng mẫu thiết kế:**

* MVC (Model - View - Controller):

Hệ thống tuân theo kiến trúc MVC giúp tách biệt:

+ Model: Quản lý dữ liệu và logic nghiệp vụ (folder models/)

+ View: Giao diện người dùng (folder views/)

+ Controller: Xử lý yêu cầu và tương tác giữa View và Model (folder controllers/)

1. **Sử dụng kiến trúc đa tầng (Tiered Architecture):**

* Presentation Layer: views/ hiển thị giao diện cho người dùng
* Application Layer: controllers/ xử lý nghiệp vụ và tương tác với model
* Data Access Layer: models/ tương tác trực tiếp với cơ sở dữ liệu thông qua ORM Sequelize
* Configuration Layer: config/config.json chứa thông tin kết nối CSDL

1. **Kiến trúc Client - Server:**

* Ứng dụng được thiết kế theo kiểu client-server:

+ Client: Gửi yêu cầu HTTP (qua trình duyệt/webapp)

+ Server: Xử lý yêu cầu, truy xuất dữ liệu và trả kết quả

1. **ORM – Object Relational Mapping:**

* Sử dụng Sequelize để ánh xạ các đối tượng JavaScript (Model) với các bảng trong cơ sở dữ liệu

+ Giảm sự phụ thuộc vào SQL thuần

+ Dễ bảo trì và mở rộng

1. **Hỗ trợ cơ chế mở rộng (Plug-in/Middleware):**

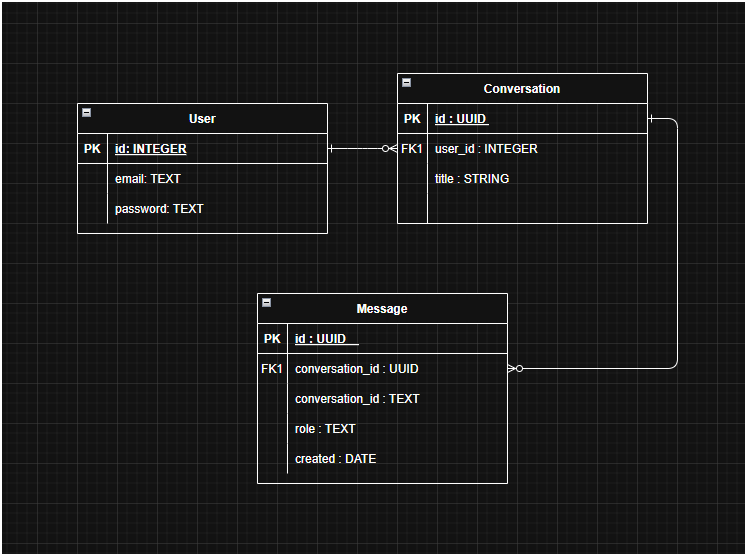
* Có thể dễ dàng mở rộng hệ thống qua các Middleware như:

+ passport.js: Xác thực người dùng

+ validator.js: Kiểm tra tính hợp lệ dữ liệu

# Thiết kế dữ liệu

## Sơ đồ dữ liệu



## Đặc tả dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| User | | | |
| Tên thuộc tính | Kiểu dữ liệu và giá trị | Ràng buộc khóa | Diễn giải |
| id | *INTEGER* | PRIMARY KEY, AUTO\_INCREMENT, NOT NULL | Mã định danh duy nhất cho người dùng |
| email | *TEXT* | NOT NULL, UNIQUE | Địa chỉ email người dùng |
| password | *TEXT* | |  | | --- | |  | | NOT NULL | | | | Mật khẩu đã được mã hóa |

|  |  |  |  |
| --- | --- | --- | --- |
| Conversation | | | |
| Tên thuộc tính | Kiểu dữ liệu và giá trị | Ràng buộc khóa | Diễn giải |
| id | UUID | PRIMARY KEY, NOT NULL, DEFAULT UUIDV4 | Mã định danh duy nhất cho cuộc trò chuyện |
| user\_id | INTEGER | NOT NULL, FOREIGN KEY → User.id | Người dùng là chủ sở hữu của cuộc trò chuyện |
| title | STRING | NOT NULL | |  | | --- | |  |  |  | | --- | | Tiêu đề cuộc trò chuyện | |

|  |  |  |  |
| --- | --- | --- | --- |
| Message | | | |
| Tên thuộc tính | Kiểu dữ liệu và giá trị | Ràng buộc khóa | Diễn giải |
| id | UUID | |  | | --- | |  |  |  | | --- | | PRIMARY KEY, NOT NULL, DEFAULT UUIDV4 | | |  | | --- | | Mã định danh duy nhất cho tin nhắn |  |  | | --- | |  | |
| conversation\_id | UUID | NOT NULL, FOREIGN KEY → Conversation.id | |  | | --- | | Thuộc về cuộc trò chuyện nào |  |  | | --- | |  | |
| content | TEXT | |  | | --- | |  |  |  | | --- | | NOT NULL | | |  | | --- | | Nội dung tin nhắn |  |  | | --- | |  | |
| role | TEXT | |  | | --- | |  |  |  | | --- | | NOT NULL | | |  | | --- | | Vai trò của người gửi (user, system) |  |  | | --- | |  | |
| created | DATE | |  | | --- | |  |  |  | | --- | | NOT NULL, DEFAULT CURRENT\_TIMESTAMP | | Thời điểm tạo tin nhắn |