**Đại học công nghiệp Tp Hồ Chí Minh Khoa công nghệ thông tin**

**Báo cáo tiểu luận**

**Môn học: Công nghệ mới trong phát triển và ứng dụng web**

**Giảng Viên: Tôn Long Phước**

**Lớp: DHKTPM15BTT Nhóm: 6**

**Đề tài:**

**Phát triển ứng dụng chat realtime**

***SVTH****:* ***Lớp****:*

*19434151- Phùng Phú Đạt DHKTPM15CTT*

*19484491- Trần Thành Nam DHKTPM15CTT*

*19487761- Nguyễn Minh Quang DHKTPM15CTT*

*19487761- Hoàng Huy Vũ DHKTPM15CTT*

*19486601- Nguyễn Minh Minh Vương DHKTPM15CTT*

*TPHCM ngày 28 tháng 10 năm 2022*

Mục lục

[**CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ DỰ ÁN** 3](#_Toc120219760)

[1.1 Tổng quan 3](#_Toc120219761)

[1.2 Mục tiêu đề tài 3](#_Toc120219762)

[1.3 Phạm vi đề tài 4](#_Toc120219763)

[1.4 Mô tả yêu cầu chức năng 4](#_Toc120219764)

[1.5 Yêu cầu phi chức năng 5](#_Toc120219765)

[1.6 Kế hoạch thực hiện 5](#_Toc120219766)

[1.7 Phân chia công việc 6](#_Toc120219767)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 7](#_Toc120219768)

[2.1 Cloud Computing với AWS 7](#_Toc120219769)

[2.1.1 EC2 7](#_Toc120219770)

[2.1.2 S3 8](#_Toc120219771)

[2.1.3 DynamoDB 9](#_Toc120219772)

[2.2 Back End 10](#_Toc120219773)

[2.2.1 Node js 10](#_Toc120219774)

[2.2.2 Socket io 12](#_Toc120219775)

[2.3 Front End 13](#_Toc120219776)

[2.3.1 Android 13](#_Toc120219777)

[2.3.2 Bootstrap 13](#_Toc120219778)

[2.4 Kiến trúc phần mềm 14](#_Toc120219779)

[**Ưu điểm mô hình MVC** 15](#_Toc120219780)

[**Nhược điểm mô hình MVC** 16](#_Toc120219781)

[2.5 Dịch vụ xác thực tin nhắn 16](#_Toc120219782)

[**Chương 3: Phân tích và thiết kế** 17](#_Toc120219783)

[3.1 Usecase tổng quát 17](#_Toc120219784)

[3.2 Danh sách tác nhân và mô tả 17](#_Toc120219785)

[3.3 Danh sách các use case hoạt động 18](#_Toc120219786)

[3.4 Tình huống hoạt động 18](#_Toc120219787)

[3.5 Class Diagram 50](#_Toc120219788)

[**Chương 4: Kiểm thử hiện thực** 51](#_Toc120219789)

[4.1 Giao diện Đăng nhập 51](#_Toc120219790)

[4.2 Giao diện Đăng ký 51](#_Toc120219791)

[4.3 Giao diện Quên mật khẩu 52](#_Toc120219792)

[4.4 Giao diện tin nhắn 53](#_Toc120219793)

**Bảng đánh giá**

|  |  |
| --- | --- |
| **Họ tên** | **Đánh giá** |
| **Phùng Phú Đạt** |  |
| **Trần Thành Nam** |  |
| **Nguyễn Minh Quang** |  |
| **Hoàng Huy Vũ** |  |
| **Nguyễn Minh Minh Vương** |  |

# **CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ DỰ ÁN**

## 1.1 Tổng quan

Ứng dụng chat LINK là ứng dụng cho phép người dùng nhắn tin và tương tác thông qua 2 nền tảng là website và di động.

## 1.2 Mục tiêu đề tài

Mục tiêu chính của ứng dụng là cung cấp cho người dùng những tính năng cơ bản như nhắn tin, gửi ảnh, gửi video, gửi file trong khung chat với thời gian thực. Hỗ trợ nhắn tin 1-1 hoặc nhắn tin theo nhóm.

Ngoài ra ứng dụng tồn tại một admin sẽ quản lý nguồn tài nguyên và dữ liệu của hệ thống.

## 1.3 Phạm vi đề tài

Hệ thống gồm 2 đối tượng chính: người dùng và admin. Người dùng được sử dụng các chức năng cơ bản của một tài khoản như: nhắn tin, gửi ảnh – video – file, các chức năng tương tác với người dùng khác (kết bạn, tạo nhóm, xem thông tin, tìm kiếm, …). Admin có quyền truy cập, xử lí, quản lý các tài nguyên trong hệ thống bao gồm dữ liệu của người dùng nhưng sẽ phải chịu sự hạn chế.

## 1.4 Mô tả yêu cầu chức năng

**Yêu cầu chức năng cho người dùng**

     Đăng ký tài khoản ( số điện thoại)

     Đăng nhập tài khoản

     Cập nhật thông tin

* + - Cập nhật thông tin cơ bản (ngày sinh, giới tính, họ tên…)
    - Cập nhật avatar

     Đổi mật khẩu

Quên mật khẩu

     Tìm kiếm

* + - Tìm kiếm user qua số điện thoại
    - Tìm kiếm user qua tên người dùng

     Xem thông tin

     Nhắn tin:

* + - Nhắn tin 1-1
    - Nhắn tin nhóm
    - Gửi ảnh
    - Gửi video
    - Gửi file
    - Thu hồi tin nhắn
    - Gửi emoji

     Tạo nhóm

     Kết bạn

* + - Gửi lời mời kết bạn
    - Phản hồi lời mời kết bạn

     Đăng xuất tài khoản

## 1.5 Yêu cầu phi chức năng

       Ứng dụng phải có giao diện thân thiện với người dùng mới, bố cục các chức năng phải hợp lý và hoạt động ổn định, chính xác. Màu sắc không được quá rối loạn.

       Đảm bảo được về các chức năng bảo mật.

       Đáp ứng được việc hoạt động server.

       Hệ thống chạy ổn định, hạn chế việc bảo trì hệ thống.

## 1.6 Kế hoạch thực hiện

|  |  |
| --- | --- |
| Thời gian | Công việc |
| Tuần 1 | -         Đăng ký nhóm  -         Chọn đề tài  -         Tiến hành lập kế hoạch, hiện thực hoá ban đầu những việc cần làm để hoàn thành đề tài.  -         Phân công, chia nhiệm vụ cho từng thành viên nhóm.  -         Chuẩn bị file báo cáo. |
| Tuần 2 | -         Phân tích yêu cầu đề tài.  -         Chỉ ra những yêu cầu, chức năng trong đề tài, đưa ra một số dẫn chứng liên quan  -         Tìm hiểu về các công cụ liên quan  -         Tóm lượt, thống nhất lại các mục tiêu |
| Tuần 3 | -         Thiết kế mô hình UML:  Use Case Diagram  Activity Diagram  Sequence Diagram  Class Diagram  Thiết kế cơ sở dữ liệu |
| Tuần 4 | * Code các chức năng   + Nhắn tin   + Gửi ảnh   + Gửi video   + Gửi file   + Gửi emoji   + Thu hồi tin nhắn |
| Tuần 5 | * Code giao diện |
| Tuần 6 | * Code các chức năng:   + Đăng nhập   + Đăng ký   + Cập nhật thông tin   + Đăng xuất |
| Tuần 7 | * Code các chức năng:   + Kết bạn   + Tìm kiếm |
| Tuần 8 | * Thực hiện việc kiểm tra hệ thống * Thực hiện việc sửa lỗi còn tồn tại trong hệ thống |
| Tuần 9 | * Hoàn tất công việc * Kiểm thử lần cuối |
| Tuần 10 | * Hoàn thành file báo cáo * Chuẩn bị cho việc báo cáo trước hội đồng. |

## 1.7 Phân chia công việc

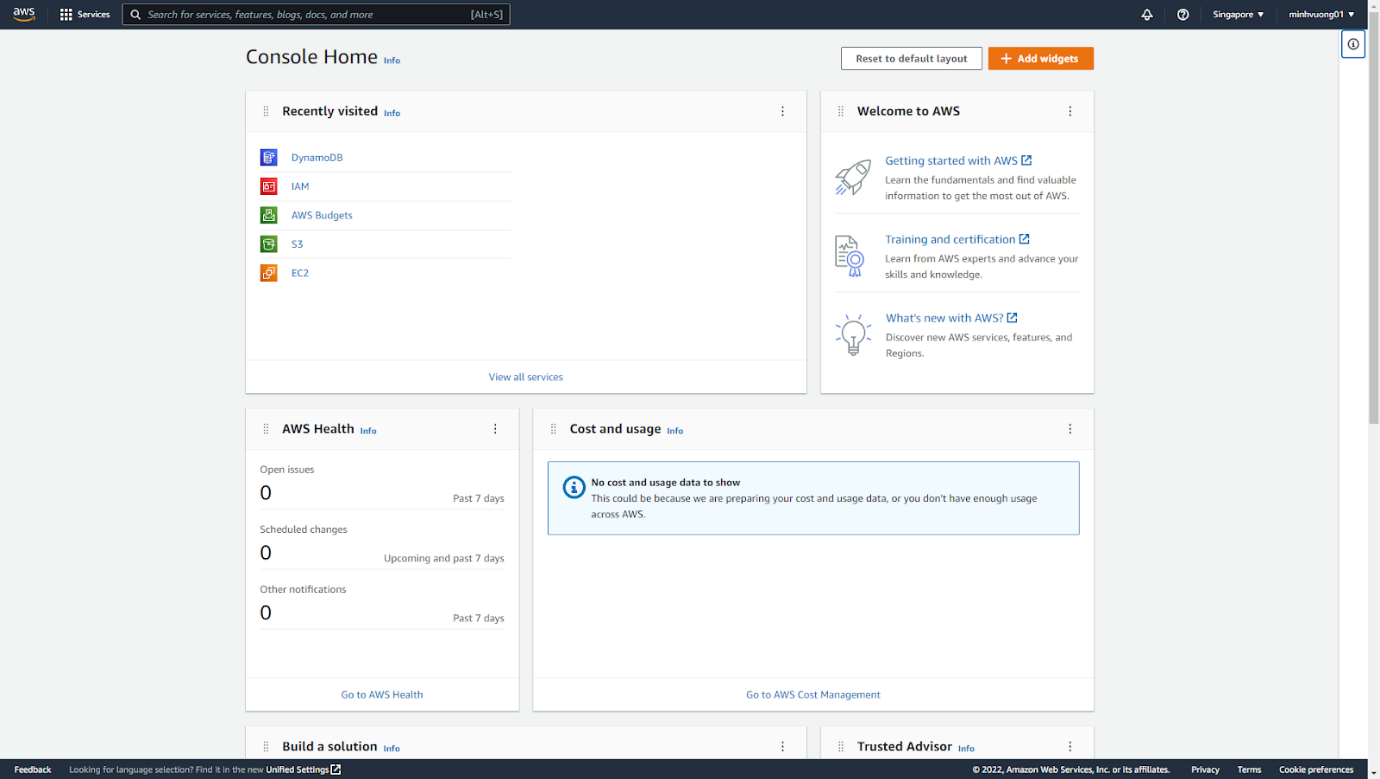
|  |  |
| --- | --- |
| Thành viên | Công việc |
| Phùng Phú Đạt | Front – end |
| Trần Thành Nam | Android |
| Nguyễn Minh Quang | Back end |
| Hoàng Huy Vũ | Android |
| Nguyễn Minh Minh Vương | Front – end |

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## 2.1 Cloud Computing với AWS

Amazon Web Service là một nền tảng điện toán đám mây được phát triển và cung cấp bởi Amazon, cung cấp cho doanh nghiệp các giải pháp về: Storage, computing power, databases, networking, analytics, developer tools, security, virtualization,…

[Amazon Web Services](https://aws.amazon.com/) là nền tảng dịch vụ điện toán đám mây an toàn, mang đến khả năng tính toán, lưu trữ cơ sở dữ liệu, phân phối nội dung và các chức năng khác nhằm giúp các doanh nghiệp mở rộng và phát triển.

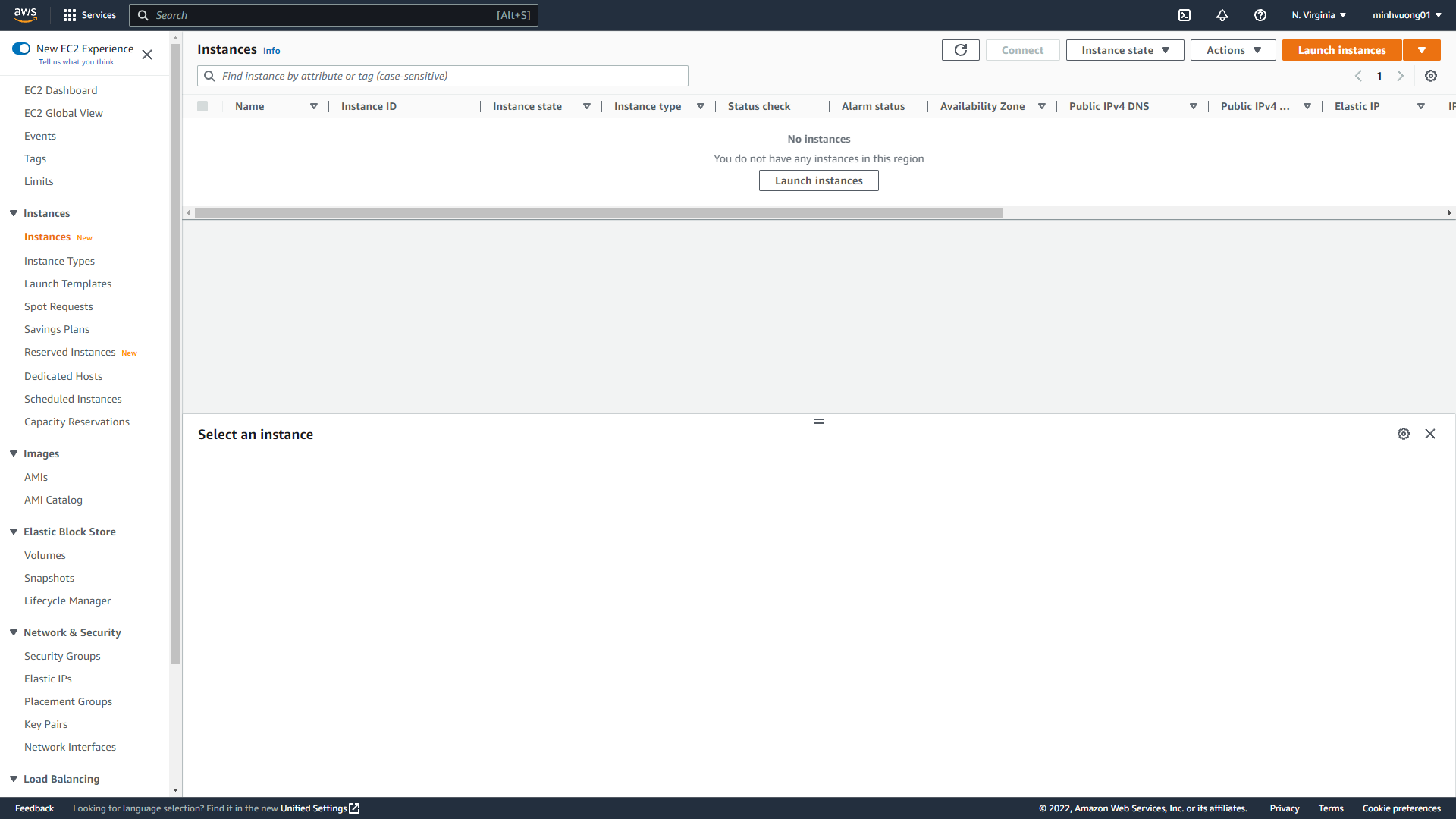


Hình 2.1 Màn hình giao diện chính của AWS

Ứng dụng chat sử dụng các dịch vụ của AWS như: EC2, S3, DynamoDB

### 2.1.1 EC2

* Amazon Elastic Compute Cloud (Amazon EC2) là một cơ sở hạ tầng điện toán đám mây được cung cấp bởi Amazon Web Services (AWS) giúp cung cấp tài nguyên máy tính ảo hoá theo yêu cầu.
* Amazon EC2 cung cấp các ứng dụng máy tính ảo hoá có thể mở rộng về khả năng xử lý cùng các thành phần phần cứng ảo như bộ nhớ máy tính (ram), vi xử lý, linh hoạt trong việc lựa chọn các phân vùng lưu trữ dữ liệu ở các nền tảng khác nhau và sự an toàn trong quản lý dịch vụ bởi kiến trúc ảo hoá đám mây mạnh mẽ của AWS.
* Amazon EC2 sẽ cung cấp một hoặc máy chủ ảo có thể kết hợp với nhau để dễ dàng triển khai ứng dụng nhanh nhất và đảm bảo tính sẵn sàng cao nhất. Thậm chí về mặt thanh toán bạn dễ dàng biết được các mức chi phí cần thanh toán dựa trên thông tin tài nguyên bạn sử dụng.



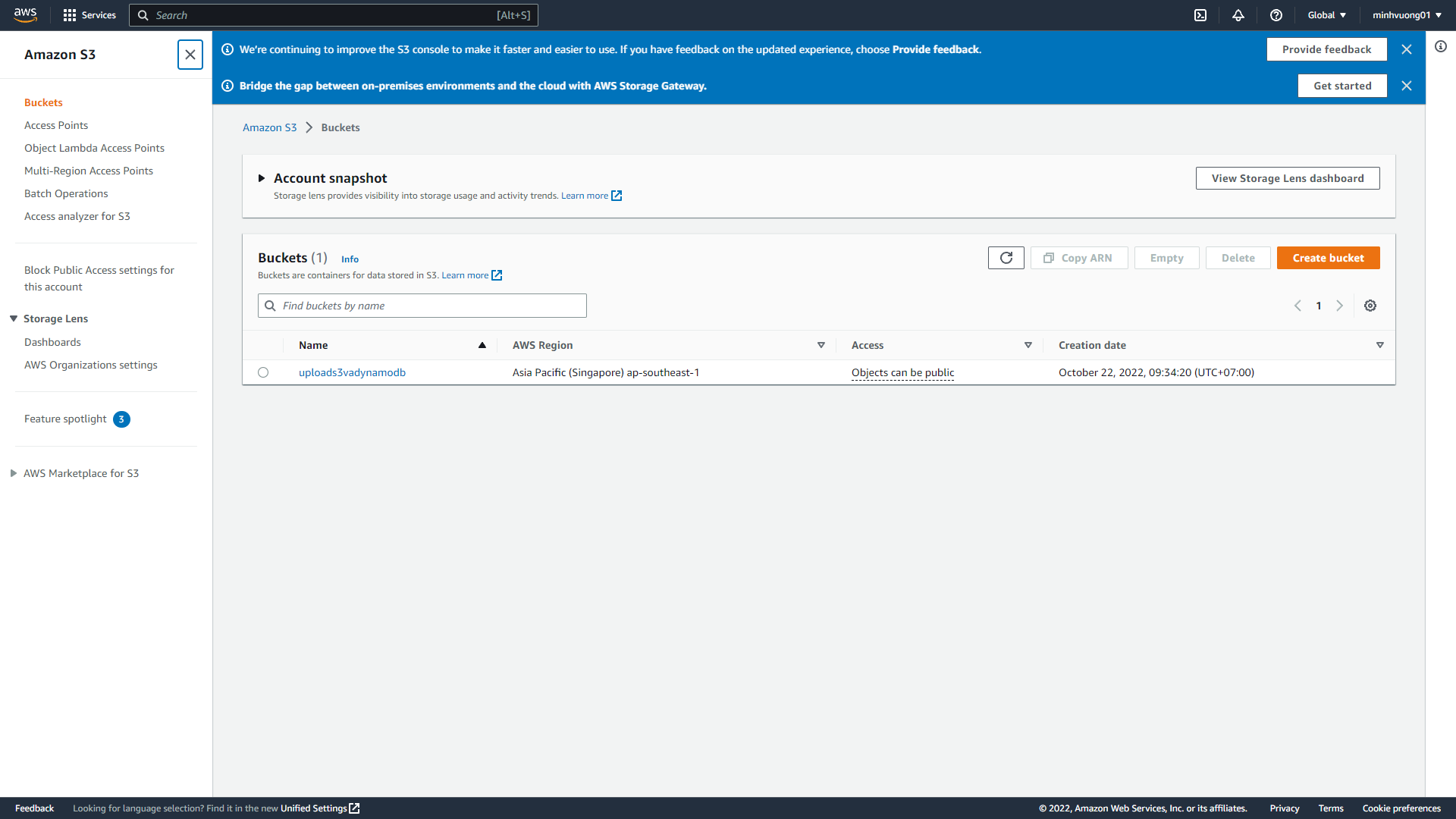
Hình 2.1.1 Màn hình chính dịch vụ EC2

### 2.1.2 S3

Amazon S3 là dịch vụ lưu trữ đối tượng được xây dựng để lưu trữ và truy xuất bất kỳ lượng dữ liệu nào từ bất cứ nơi nào. Đây là dịch vụ lưu trữ đơn giản có độ bền, độ sẵn có, hiệu suất, tính bảo mật dẫn đầu ngành và khả năng thay đổi quy mô gần như không giới hạn với chi phí cực kỳ thấp.

S3 có khả năng mở rộng cao vì nó tự động tăng dung lượng lưu trữ của bạn theo yêu cầu và bạn chỉ trả tiền cho bộ nhớ bạn sử dụng.

Sử dụng Amazon để lưu trữ và sử dụng các hình ảnh và đồng thời không cẩn lo lắng vì S3 có khả năng mã hoá dữ liệu giúp đảm bảo bảo mật thông tin.



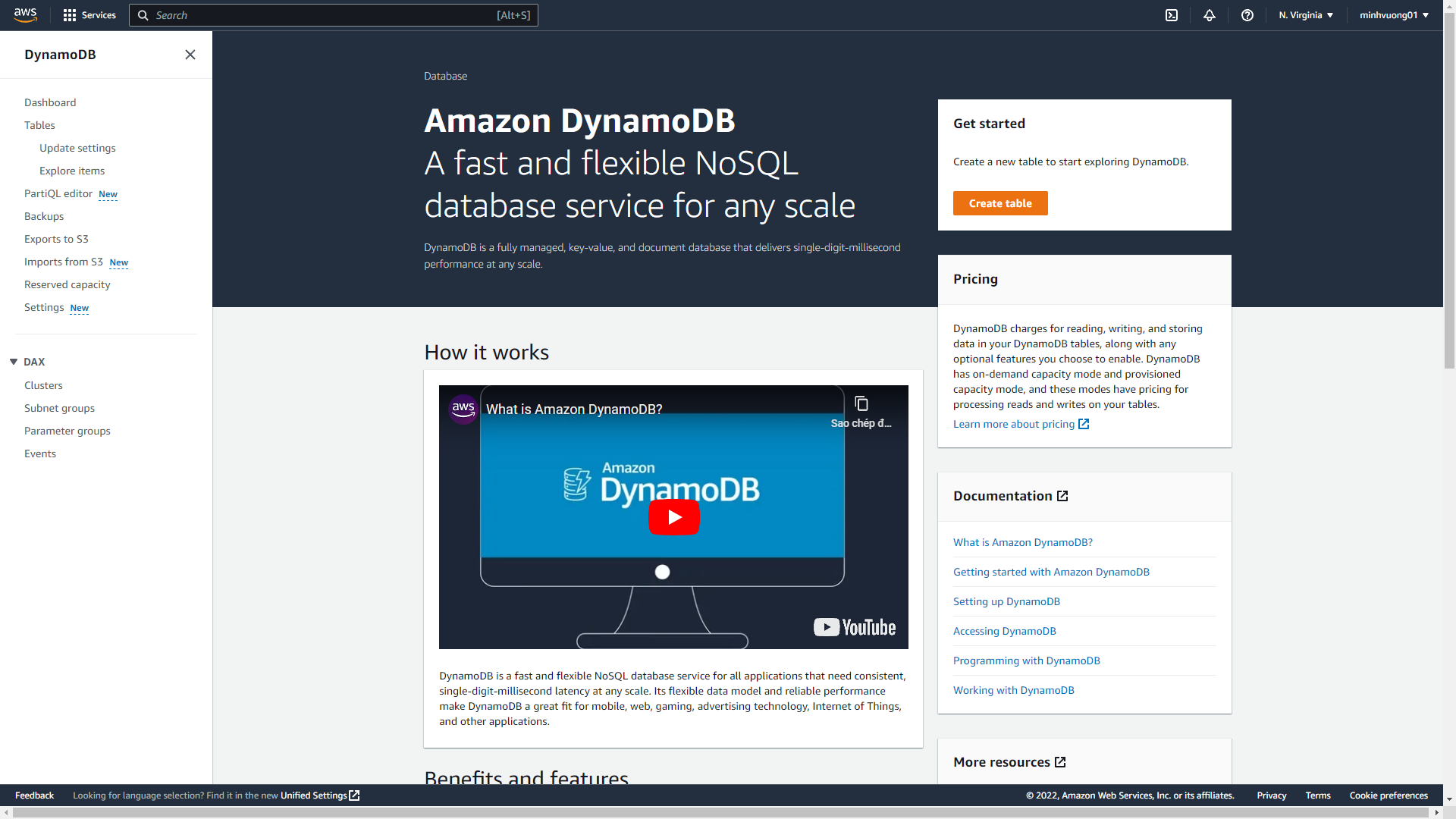
Hình 2.1.2 Dịch vụ lưu trữ S3

### 2.1.3 DynamoDB

DynamoDB là một dịch vụ quản lý NoSQL có khả năng đáp ứng hiệu suất cao và nhanh kèm theo khả năng mở rộng.

Bạn có thể sử dụng DynamoDB để tạo ra một bảng có khả năng lưu trữ và truy xuất bất kỳ số lượng dữ liệu, mà vẫn có thể phục vụ cho bất kỳ mức độ request traffic.

**Amazon DynamoDB** là một cơ sở dữ liệu bền vững, đa hoạt tính, đa khu vực, được quản lý toàn phần, có tích hợp tính năng bảo mật, sao lưu, khôi phục và lưu vào bộ nhớ đệm cho các ứng dụng trên quy mô internet.

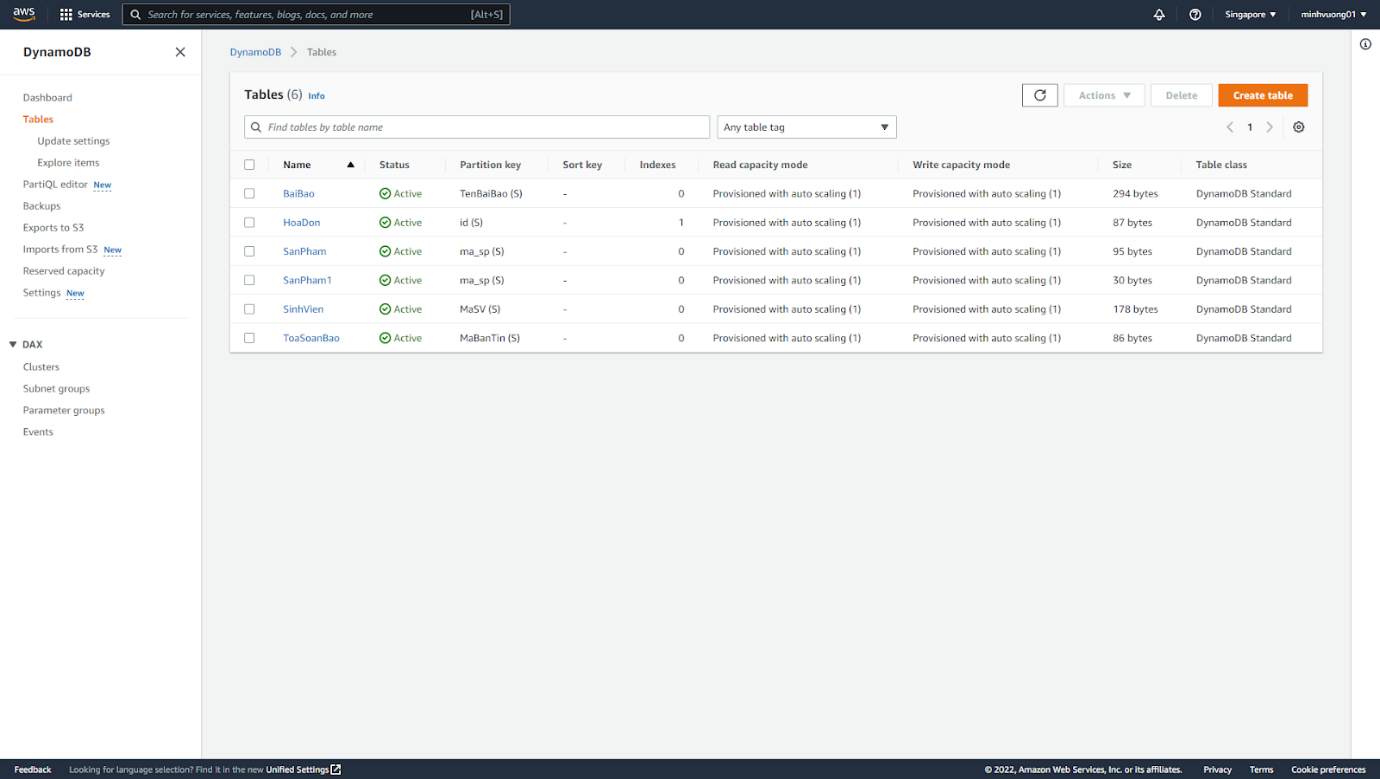


Hình 2.1.3.1 Dịch vụ Amazon DynamoDB

**Các ưu điểm và tính năng của Amazon DynamoDB:**

1. Tốc độ phản hồi nhanh
2. Hiệu năng trên quy mô lớn nhờ kiến ​​trúc phân tán
3. Không cần quản lý máy chủ (Serverless)
4. Phù hợp dùng trong doanh nghiệp
5. Các tính năng mới

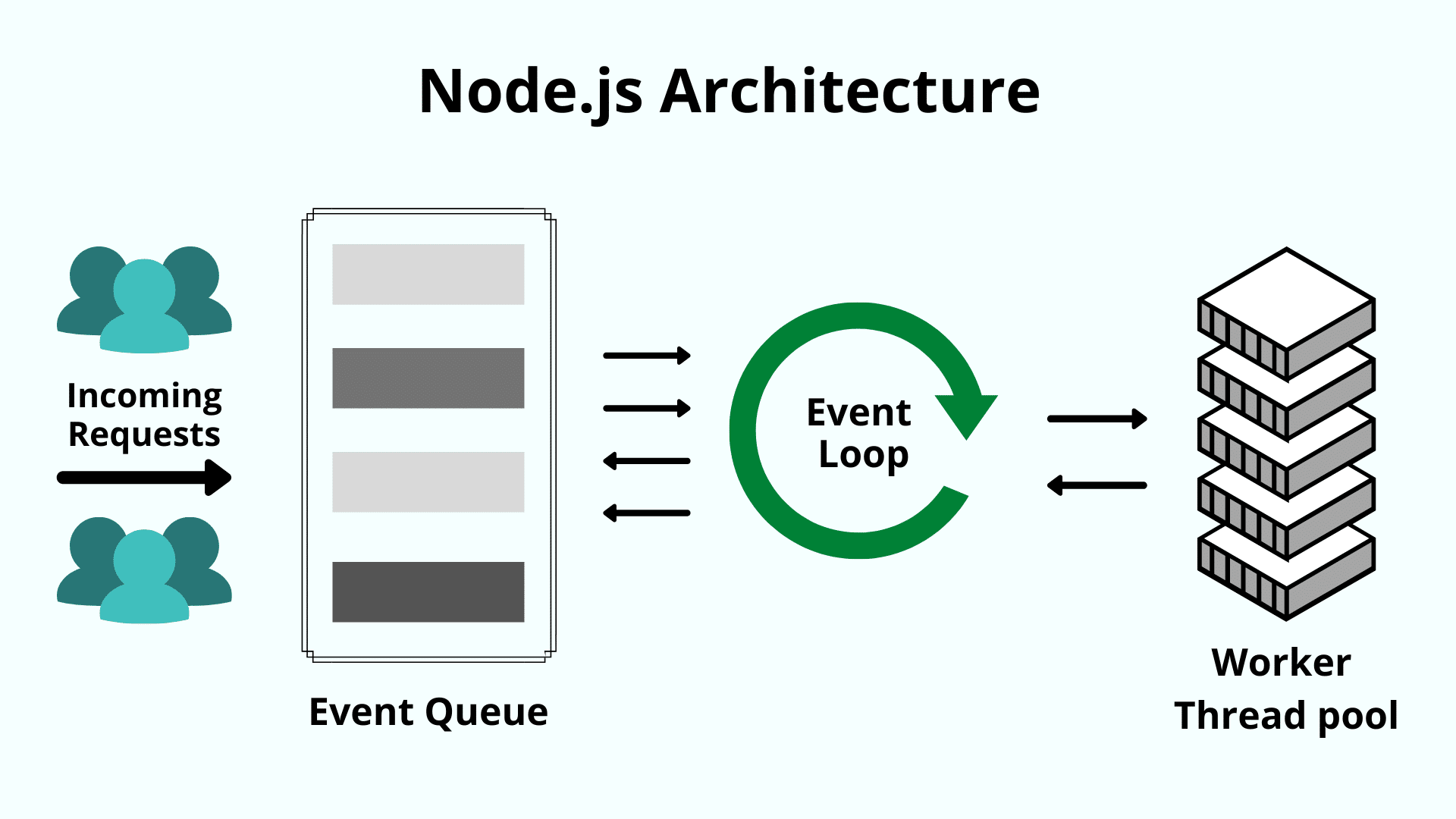
* Việc cập nhật hoặc xoá dữ liệu hay xoá một tablue là dễ dàng.
* Khôi phục các bảng DynamoDB nhanh hơn nữa.
* Công cụ tính giá AWS hiện đã hỗ trợ Amazon DynamoDB.
* DynamoDB hỗ trợ các service khác của Amazon và ứng dụng vào công việc dễ dàng ( EC2, S3,…)



Hình 2.1.3.2 Table trong DynamoDB

## 2.2 Back End

### 2.2.1 Node js



Hình 2.1.3.3 Node js

NodeJS là một môi trường runtime chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

Node.JS là một trong những nền tảng phổ biến nhất hiện nay cho mục đích phát triển ứng dụng mạng phía server.

Lý do nên sử dụng NodeJS và những đặc điểm khiến Node.JS là lựa chọn hàng đầu cho các developer hiện nay:

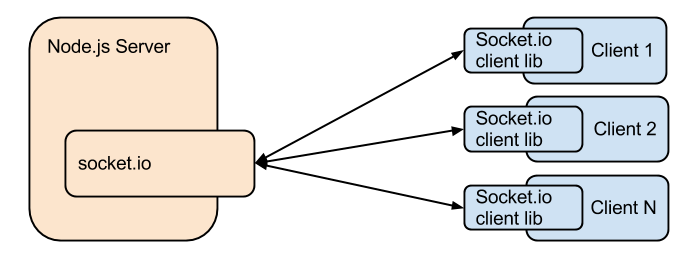
* Tốc độ cực nhanh: Được xây dựng dựa trên engine JavaScript V8 của Google Chrome, do đó các thư viện của nó có khả năng thực thi code chỉ rất nhanh.
* NPM: Với hơn 50,000 package khác nhau, các developer có thể dễ dàng lựa chọn bất kỳ tính năng nào để xây dựng cho ứng dụng của mình.
* Lập trình không đồng bộ: Mọi API của Node.JS đều có tính không đồng bộ (non-blocking), do đó một server dựa trên Node.JS không cần phải đợi API trả về dữ liệu.
* Không có buffering: Node.JS giúp tiết kiệm thời gian xử lý file khi cần upload âm thanh hoặc video vì các ứng dụng này không bao giờ buffer dữ liệu mà chỉ xuất dữ liệu theo từng phần (chunk).
* Đơn luồng: Node.JS sử dụng mô hình đơn luồng với vòng lặp sự kiện. Do đó các ứng dụng có thể xử lý số lượng request lớn hơn rất nhiều so với các server truyền thống như Apache HTTP Server.

### 2.2.2 Socket io

Socket.io là một module trong Node.js được phát triển vào năm 2010. Nó được phát triển để sử dụng các kết nối mở để tạo điều kiện giao tiếp thời gian thực, trả về giá trị thực ở tại thời điểm đó. Socket.io cho phép giao tiếp hai chiều giữa máy khách và máy chủ. Giao tiếp hai chiều được bật khi máy khách có Socket.io trong trình duyệt và máy chủ cũng đã tích hợp gói Socket.io

Socket io không phải là một ngôn ngữ, vì vậy nó phải được sử dụng kết hợp với những ngôn ngữ khác như php, asp.net, nodejs …

Một số ưu điểm: Bảo mật cao, kết nối tự động server, mã hoá nhị phân, cho phép tạo kênh và phòng…



Hình 2.2.2.1 Mô hình Socket.io

## 2.3 Front End

### 2.3.1 Android

****

Hình 2.1.3.4 Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) dành cho Android. Android Studio được xây dựng trên phần mềm IntelliJ IDEA của JetBrains và được thiết kế đặc biệt với các công cụ lập trình giúp xây dựng và phát triển các ứng dụng chất lượng cao nhất dành cho nền tảng Android của Google.

### 2.3.2 Bootstrap



Hình 2.3.2.1 Bootstrap

Bootstrap là một bộ sưu tập miễn phí của các mã nguồn mở và công cụ dùng để tạo ra một mẫu webiste hoàn chỉnh. Với các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng…, các designer có thể sáng tạo nhiều sản phẩm mới mẻ nhưng vẫn tiết kiệm thời gian khi làm việc với framework này trong quá trình thiết kế giao diện website.

Nó là một **Framework CSS** có chức năng quản lý và sắp xếp bố cục của các trang Web. HTML có nhiệm vụ quản lý cấu trúc và nội dung thì CSS sẽ xử lý bố cục của Website. Vì vậy mà hai cấu trúc này phải tồn tại cùng nhau để thực hiện được hành động cụ thể.

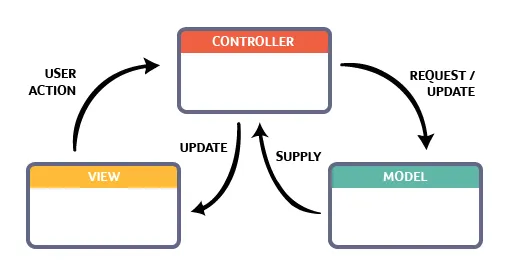
Không phải tốn thời gian để chỉnh sửa thủ công chỉ để thay đổi một thiết kế nhỏ. Thay vào đó, bạn có thể dùng CSS để tạo giao diện thống nhất trên nhiều Website mà không bị giới hạn.

Hàm CSS không chỉ giới hạn ở kiểu văn bản mà còn dùng định dạng như bảng, bố cục hình. Bạn cần mất một ít thời gian để CSS ghi nhớ hết tất cả các khai báo và bộ chọn.

## 2.4 Kiến trúc phần mềm

MVC là viết tắt của cụm từ “Model-View-Controller”. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

* Model (dữ liệu): Quản lí xử lí các dữ liệu.
* View (giao diện): Nơi hiển thị dữ liệu cho người dùng.
* Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.



Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

**Ưu điểm mô hình MVC**

* + Tải nhanh, tối ưu lượng băng thông
  + Kiểm tra dễ dàng hơn
  + Có bộ control ưu việt
  + View và size
  + Kết hợp
  + Đơn giản

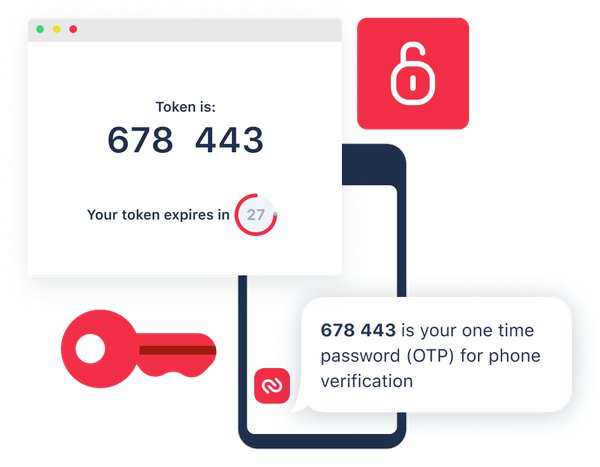
**Nhược điểm mô hình MVC**

Bên cạnh những ưu điểm MVC mang lại thì nó cũng có một số nhược điểm cần khắc phục.

MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian.

## 2.5 Dịch vụ xác thực tin nhắn

Twilio là dịch vụ cung cấp các công cụ giao tiếp có thể lập trình để gửi hoặc nhận những tin nhắn xác thực đến số di động của người dùng. Twilio được ứng dụng và có thể lập trình với Node.js.

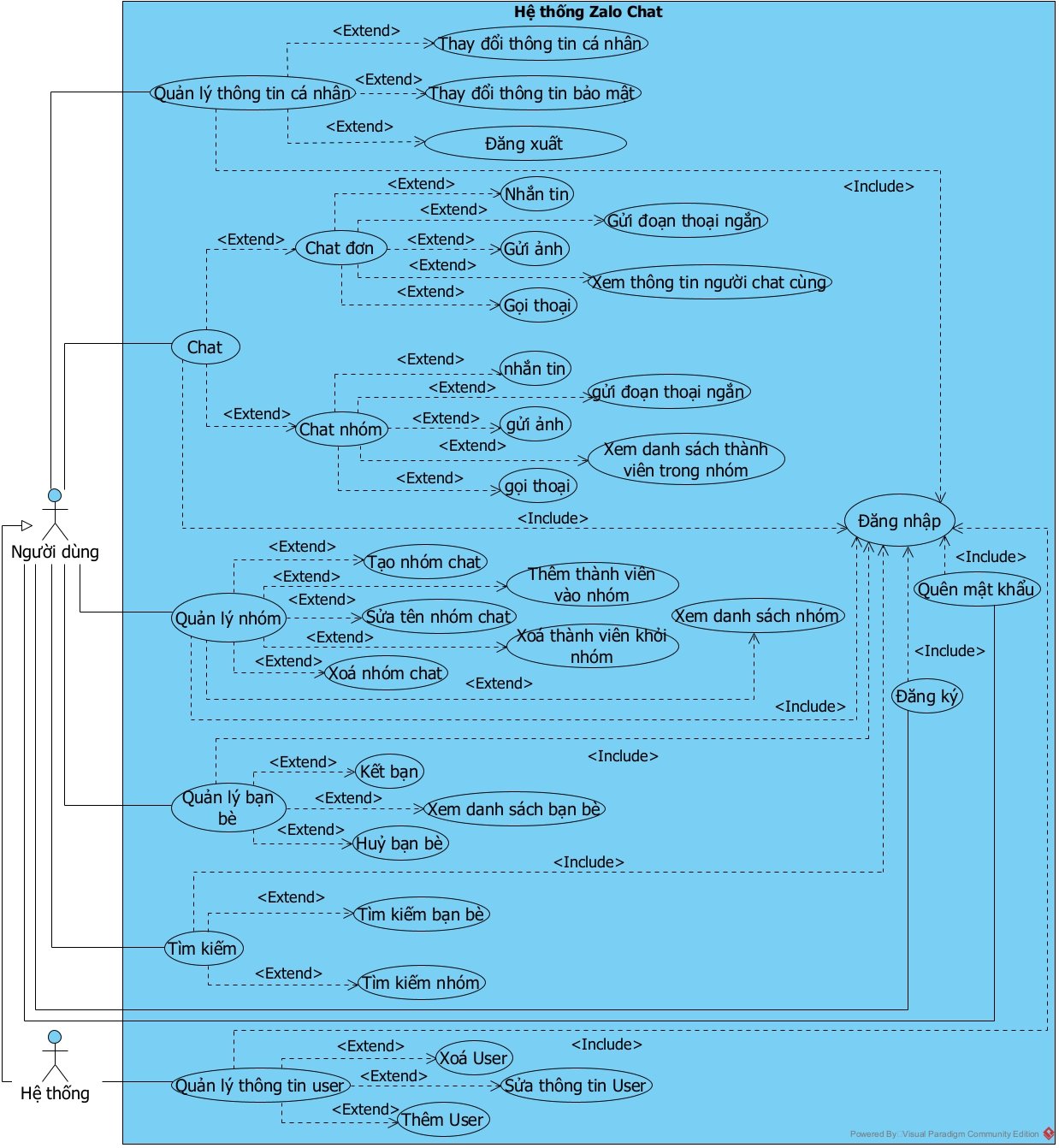


Hình 2.5.1 Dịch vụ xác thực OTP di động

# **Chương 3: Phân tích và thiết kế**

Phân tích yêu cầu bằng UML

### 3.1 Usecase tổng quát



### 3.2 Danh sách tác nhân và mô tả

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả tác nhân** |
| **User** | **Là người sử dụng chính của ứng dụng,có thể sử dụng hầu hết các chức năng mà ứng dụng có: quản lý tài khoản của mình, tìm kiếm và xem thông tin người dùng khác, quản lý tin nhắn.** |
| **Admin** | **Là người quản lý ứng dụng: Sử dụng phần mềm để quản lý ứng dụng và người dùng…** |

### 3.3 Danh sách các use case hoạt động

|  |  |
| --- | --- |
| **ID** | **Tên Use case** |
| **UC001** | **Đăng ký tài khoản** |
| **UC002** | **Đăng nhập ứng dụng** |
| **UC003** | **Cập nhật thông tin** |
| **UC004** | **Tìm kiếm** |
| **UC005** | **Xem thông tin** |
| **UC006** | **Chat 1-1** |
| **UC007** | **Tạo nhóm** |
| **UC008** | **Chat nhóm** |
| **UC009** | **Đăng xuất** |
| **UC010** | **Kết  bạn** |

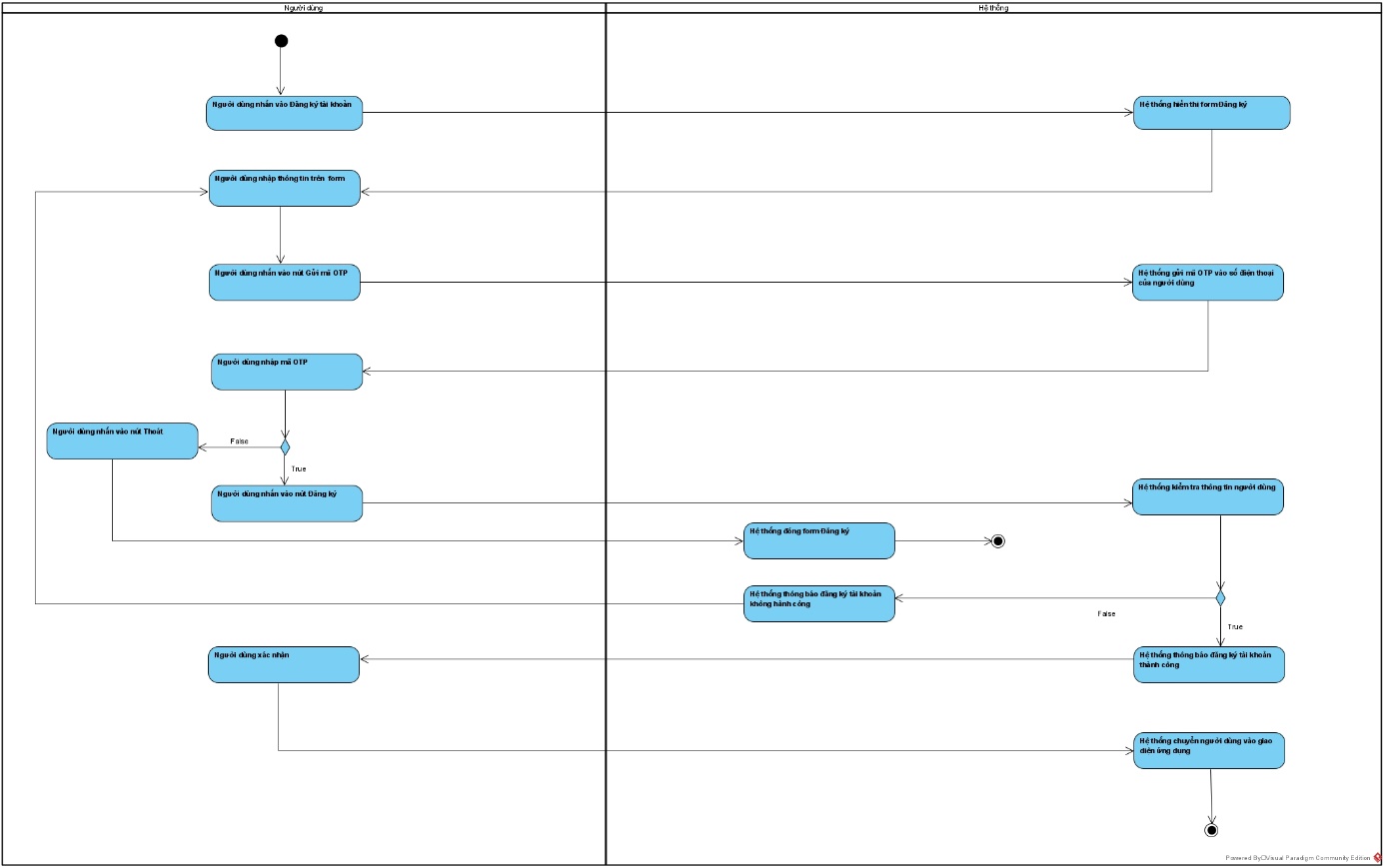
### 3.4 Tình huống hoạt động

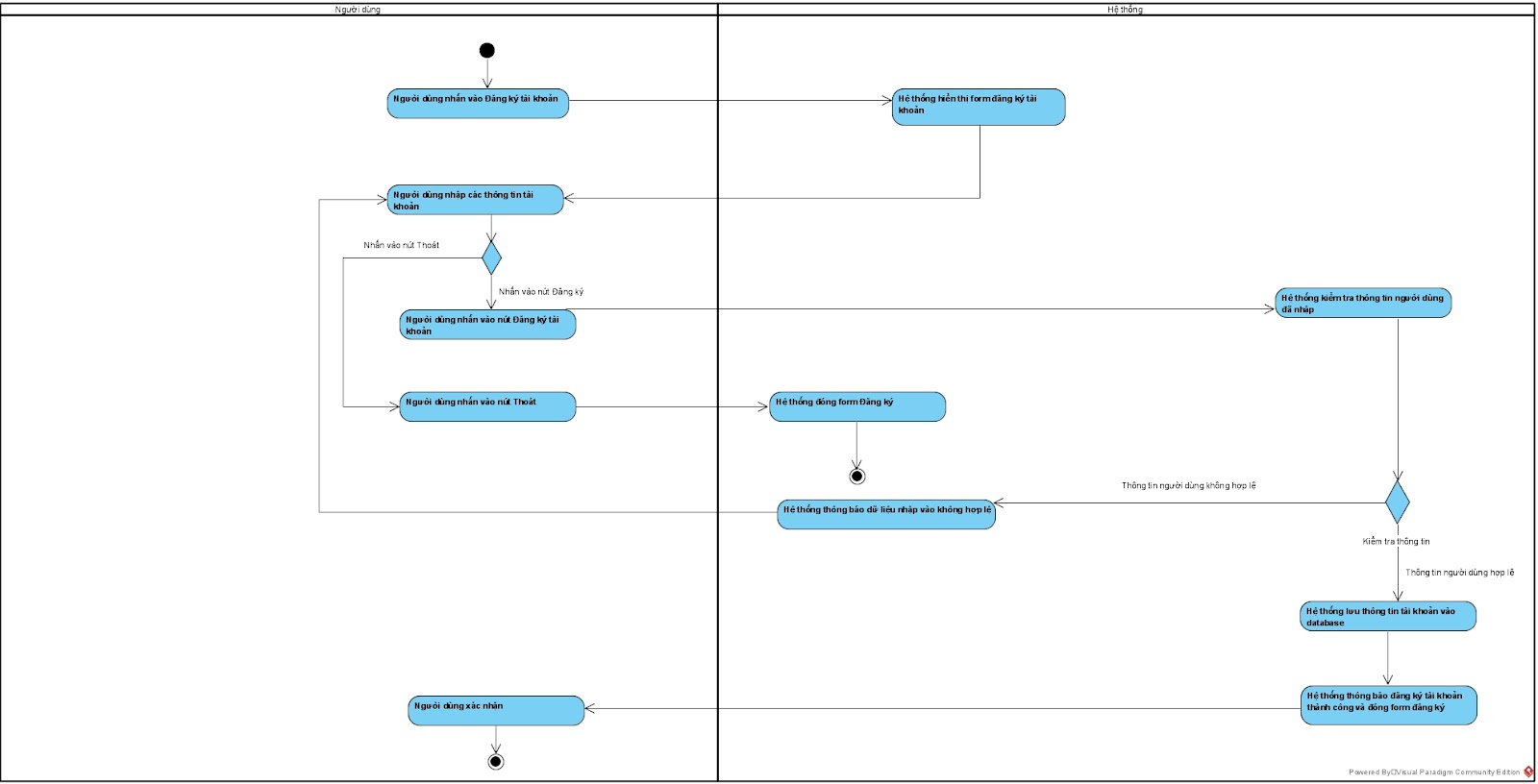
**Đặc tả use case UC01:**

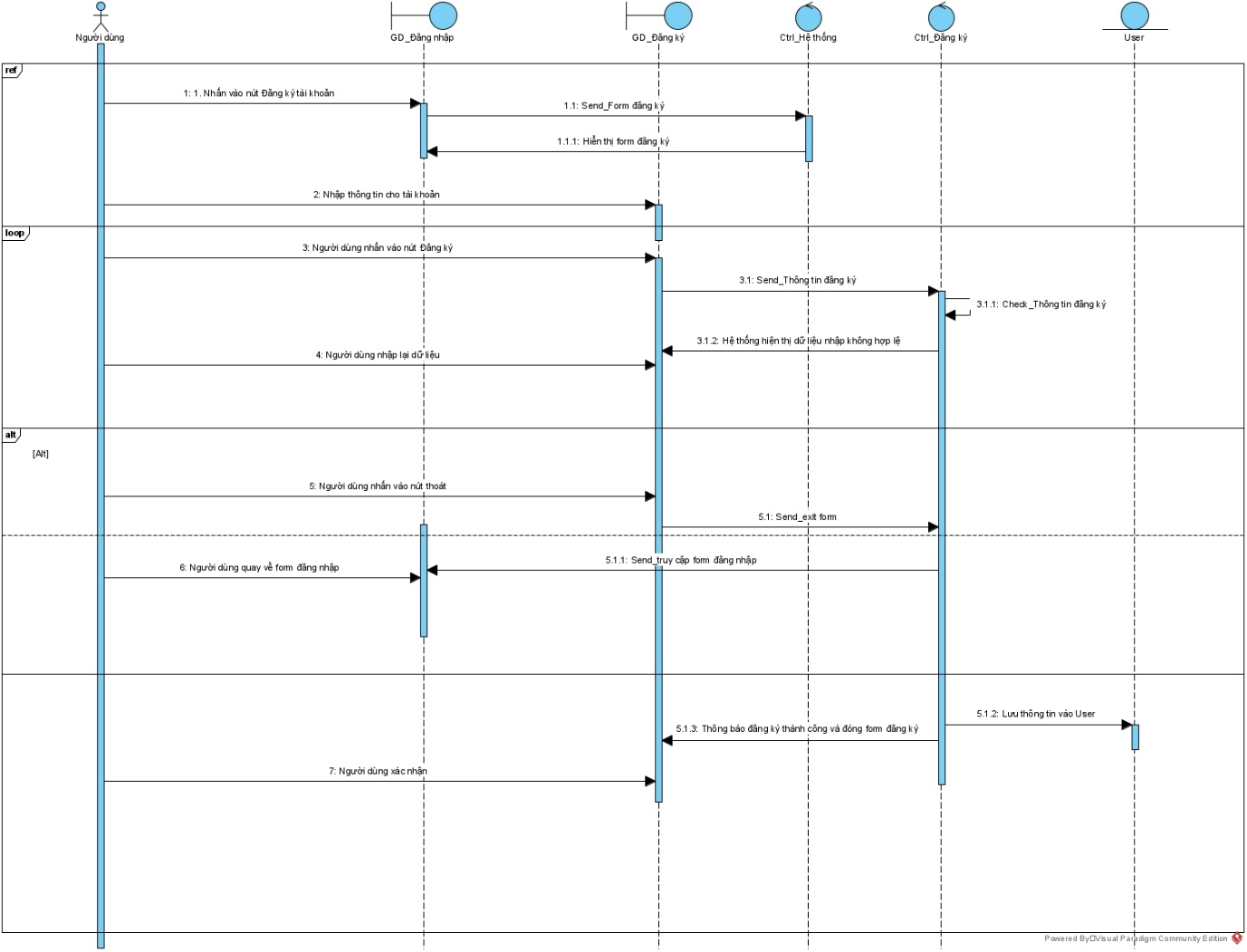
|  |  |
| --- | --- |
| **Tên use case: Đăng ký tài khoản** | |
| **Mô tả sơ lược: Tạo một tài khoản để đăng nhập vào ứng dụng** | |
| **Actor chính: Người dùng** | |
| **Actor phụ: không** | |
| **Tiền điều kiện ( Precondition): người dùng cần có số điện thoại hợp lệ** | |
| **Hậu điều kiện ( Post – condition): hệ thống thông báo đăng ký tài khoản thành công và người dùng có thể sử dụng tài khoản mới tạo đăng nhập thành công.** | |
| **Luồng sự kiện chính ( main flow):** | |
| **Actor** | **System** |
| **1.     1.Người dùng nhấn vào Đăng ký tài khoản** |  |
|  | **2.     2.Hiển thị giao diện đăng ký tài khoản** |
| **3.     3.Người dùng nhập các thông tin trên form đăng ký** |  |
| **4 4.Người dùng nhấn vào nút gửi mã OTP** |  |
|  | **5.Hệ thống gửi mã OTP vào số điện thoại của người dùng** |
| **Sd6.Người dùng nhập mã OTP** |  |
| **4.    7. Người dùng nhấn vào Đăng ký tài khoản** |  |
|  | **5.  8.   Hệ thống kiểm tra thông tin người dùng đã nhập** |
|  | **7.  9.Hệ thống thông báo đăng ký tài khoản thành công.** |
| **8.   10.Người dùng nhấn xác nhận** |  |
|  | **11.Hệ thống chuyển người dùng vào giao diện chính** |
| **Luồng sự kiện thay thế ( alternative flow):** | |
|  | **8.1 Hệ thống thông báo thông tin không chính xác** |
| **Luồng sự kiện ngoại lệ ( exception flow):** | |
| **7.1 Người dùng nhấn vào nút thoát** |  |
|  | **7.2 Hệ thống đóng form đăng ký** |
|  |  |

**Biểu đồ**

**Activity Diagram**



****

**Sequence Diagram **

**Đặc tả use case UC02: Đăng nhập vào ứng dụng**

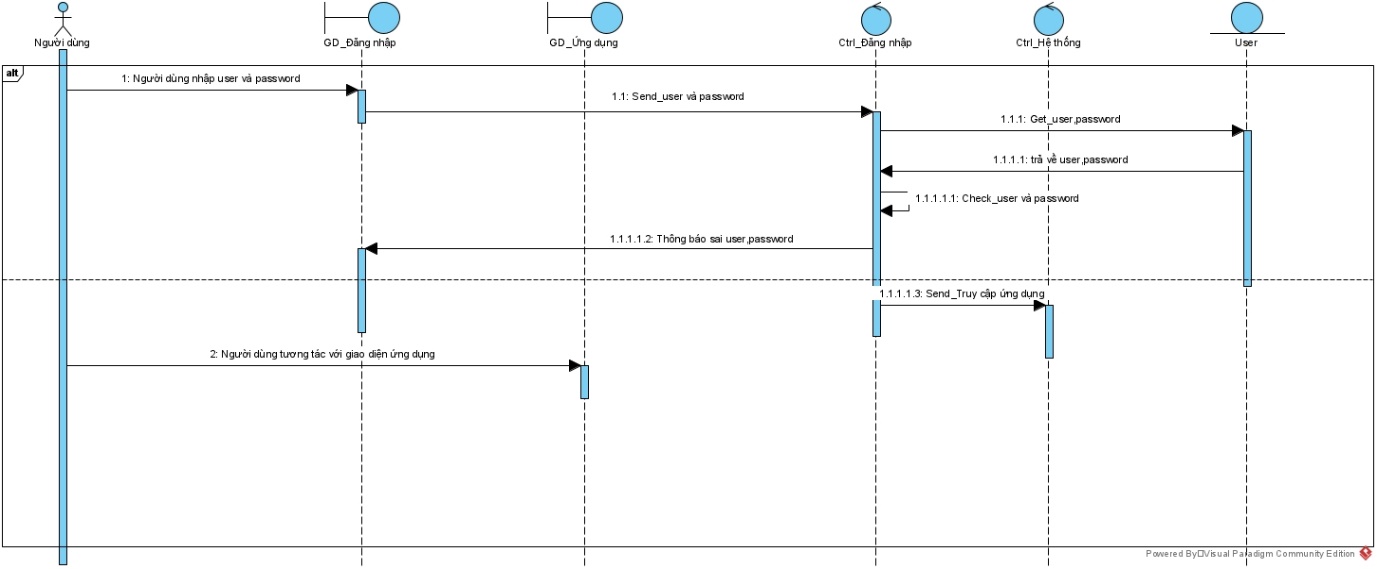
|  |  |
| --- | --- |
| **Tên use case: Đăng nhập vào ứng dụng** | |
| **Mô tả sơ lược: Người dùng đăng nhập vào ứng dụng bằng tài khoản của mình** | |
| **Actor chính: Người dùng** | |
| **Actor phụ: không** | |
| **Tiền điều kiện ( Precondition):  Người dùng đã sở hữu một tài khoản cá nhân của ứng dụng chat và truy cập thành công vào giao diện đăng nhập của ứng dụng** | |
| **Hậu điều kiện ( Post – condition): người dùng đăng nhập thành công vào ứng dụng bằng tài khoản của mình.** | |
| **Luồng sự kiện chính ( main flow):** | |
| **Actor** | **System** |
| **1.     Người dùng nhập user và password** |  |
|  | **2.     Hệ thống kiểm tra user và password** |
|  | **3.     Hệ thống cập nhật trạng thái đăng nhập của ứng dụng** |
|  | **4.     Hệ thống chuyển người dùng sang form giao diện của ứng dụng** |
| **5.     Người dùng tương tác với giao diện chính của ứng dụng** |  |
| **Luồng sự kiện thay thế ( alternative flow):** | |
|  | **2.1 Hệ thống thông báo user hoặc password không chính xác** |
|  |  |
| **Luồng sự kiện ngoại lệ ( exception flow):** | |
|  |  |
|  |  |
|  |  |

**Biểu đồ**

**Activity Diagram**

****

**Sequence Diagram**

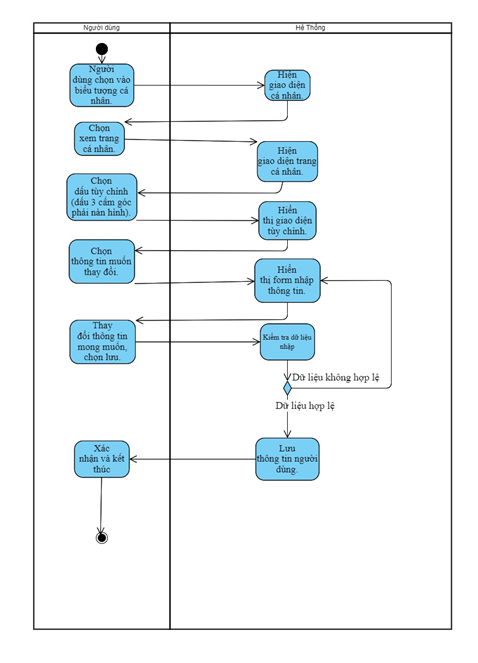
****

**Đặc tả use case UC03: Cập nhật thông tin người dùng**

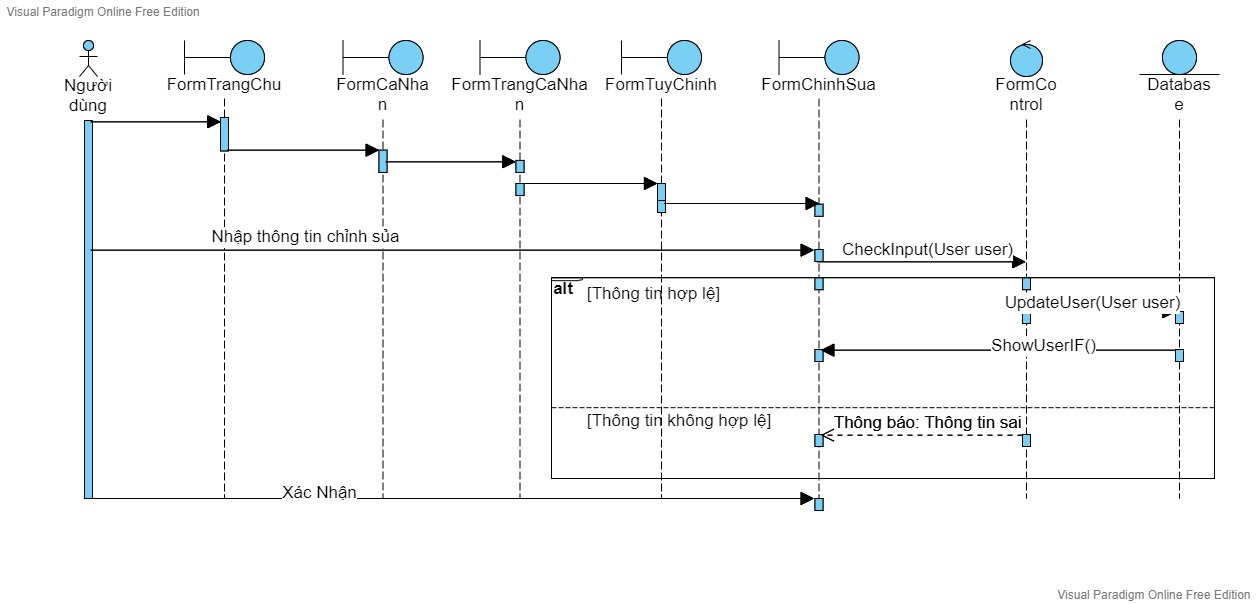
|  |  |
| --- | --- |
| **− Tên use case: Cập nhật thông tin người dùng** | |
| **− Mô tả sơ lược: Chức năng giúp người dùng cập nhật thông tin bản thân** | |
| **− Actor chính: người dùng** | |
| **− Actor phụ: không.** | |
| **− Tiền điều kiện (Pre-condition): Actor đăng nhập thành công vào hệ thống.** | |
| **− Hậu điều kiện (Post-condition): Nếu thành công thông tin người dùng xẽ thay đổi và đc lưu lại.** | |
| **− Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| **1. Người dùng chọn vào biểu tượng cá nhân.** |  |
|  | **2. Hiện giao diện cá nhân.** |
| **3. Chọn xem trang cá nhân.** |  |
|  | **4. Hiện giao diện trang cá nhân.** |
| **5. Chọn dấu tùy chỉnh (dấu 3 cấm góc phải nàn hình).** |  |
|  | **6. Hiển thị giao diện tùy chỉnh.** |
| **7. Chọn thông tin muốn thay đổi.** |  |
|  | **8. Hiển thị form nhập thông tin.** |
| **9. Thay đổi thông tin mong muốn, chọn lưu.** |  |
|  | **10. Kiểm tra dữ liệu nhập** |
|  | **11. Lưu thông tin người dùng.** |
| **12. Xác nhận và kết thúc** |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |
|  | **10.1 Hệ thống thông báo thông tin người dùng nhập ko hợp lệ yêu cầu nhập lại.** |
| **10.2 Người dùng nhập lại thông tin, nhấn lưu** |  |
|  | **10.3 Lưu thông tin người dùng vào cơ sở dữ liệu.** |

**Sơ đồ**

**Activity Diagram**

****

**Sequence Diagram**

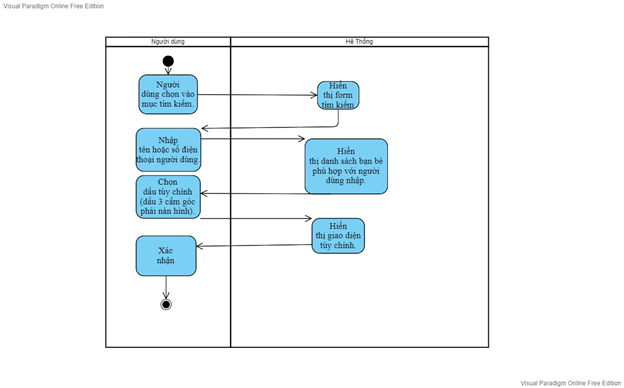
****

**Đặc tả use case UC04: Tìm kiếm bạn bè**

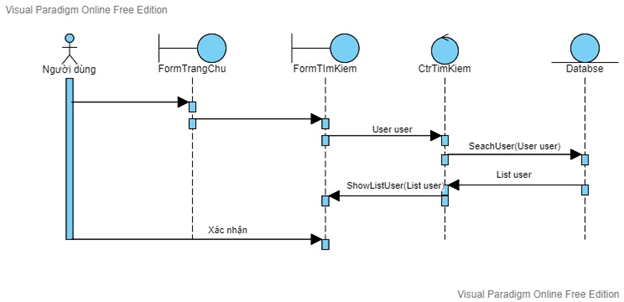
|  |  |
| --- | --- |
| **− Tên use case: Tìm kiếm bạn bè** | |
| **− Mô tả sơ lược: Chắc năng giúp người dùng tìm kiếm bạn bè** | |
| **− Actor chính: người dùng** | |
| **− Actor phụ: không.** | |
| **− Tiền điều kiện (Pre-condition): Actor đăng nhập thành công vào hệ thống.** | |
| **− Hậu điều kiện (Post-condition): Người dùng tìm được thông tin bạn bè** | |
| **− Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| **1. Người dùng chọn vào mục tìm kiếm.** |  |
|  | **2. Hiển thị form tìm kiếm** |
| **3. Nhập tên hoặc số điện thoại người dùng.** |  |
|  | **4. Hiển thị danh sách bạn bè phù hợp với người dùng nhập.** |
| **5. Xác nhận** |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |

**Sơ đồ**

**Activity Diagram**

****

**Sequence Diagram**

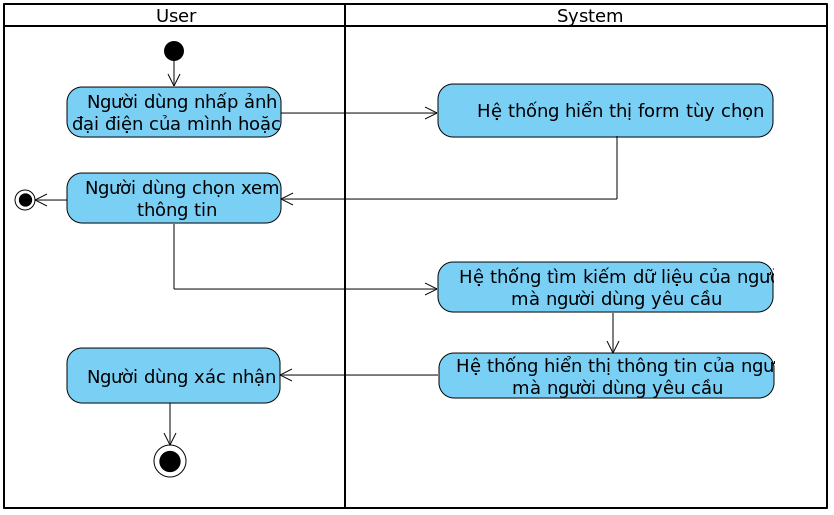
****

**Đặc tả use case UC05: Xem thông tin**

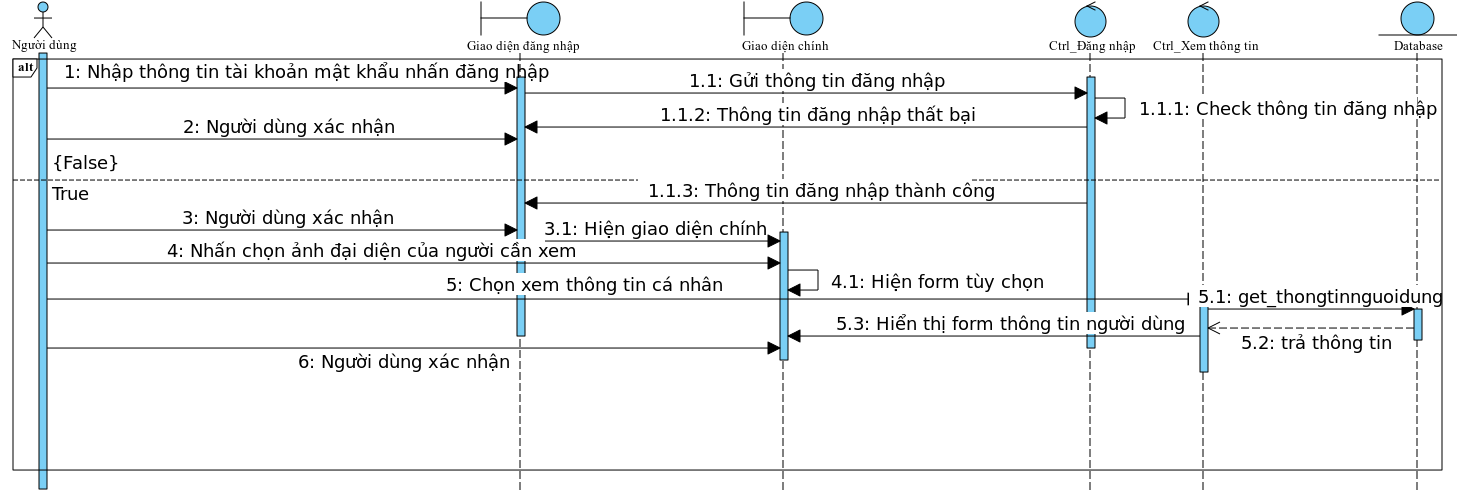
|  |  |
| --- | --- |
| **− Tên use case: Xem thông tin** | |
| **− Mô tả sơ lược:** chức năng  Xem thông tin giúp cho người dung xem được thông tin của bản thân và người khác | |
| **− Actor chính: người dùng** | |
| **− Actor phụ: không.** | |
| **− Tiền điều kiện (Pre-condition): Actor đăng nhập thành công vào hệ thống.** | |
| **− Hậu điều kiện (Post-condition):** Hiển thị thông tin của người được chọn xem thông tin | |
| **− Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1.Người dùng nhấp ảnh đại điện của mình hoặc của người khác |  |
|  | 2. Hệ thống sẽ hiển thị form tùy chọn |
| 3. Người dùng chọn xem thông tin |  |
|  | 4.  Hệ thống tìm kiếm dữ liệu của người mà người dùng yêu cầu |
|  | 5.  Hệ thống hiển thị thông tin của người mà người dùng yêu cầu |
| 6. Người dùng xác nhận |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |

**Sơ đồ**

**Activity Diagram**

****

**Sequence Diagram**

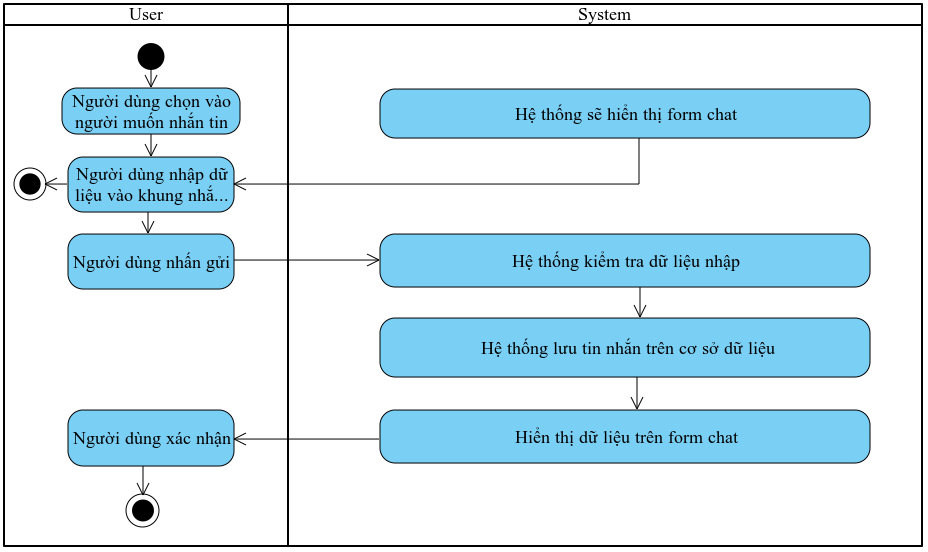
****

**Đặc tả use case UC06: Chat 1 - 1**

|  |  |
| --- | --- |
| **− Tên use case:** Chat 1-1 | |
| **− Mô tả sơ lược:** chức năng  Chat 1-1 giúp cho người dùng có thể nhắn tin với một người khác | |
| **− Actor chính:** người dùng | |
| **− Actor phụ: không.** | |
| **− Tiền điều kiện (Pre-condition): Actor đăng nhập thành công vào hệ thống.** | |
| **− Hậu điều kiện (Post-condition):** Nhắn tin thành công với người muốn nhắn tin | |
| **− Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1.Người dùng chọn vào người muốn nhắn tin |  |
|  | 2. Hệ thống sẽ hiển thị form chat |
| 3. Người dùng nhập dữ liệu vào khung nhắn tin |  |
| 4. Người dùng nhấn gửi |  |
|  | 5. Hệ thống kiểm tra dữ liệu nhập |
|  | 6. Hệ thống lưu tin nhắn trên cơ sở dữ liệu |
|  | 7. Hiển thị dữ liệu trên form chat |
| 8. Người dùng xác nhận |  |
| **− Luồng sự kiện thay thế (alternate flow):** | |

**Sơ đồ**

**Activity Diagram**

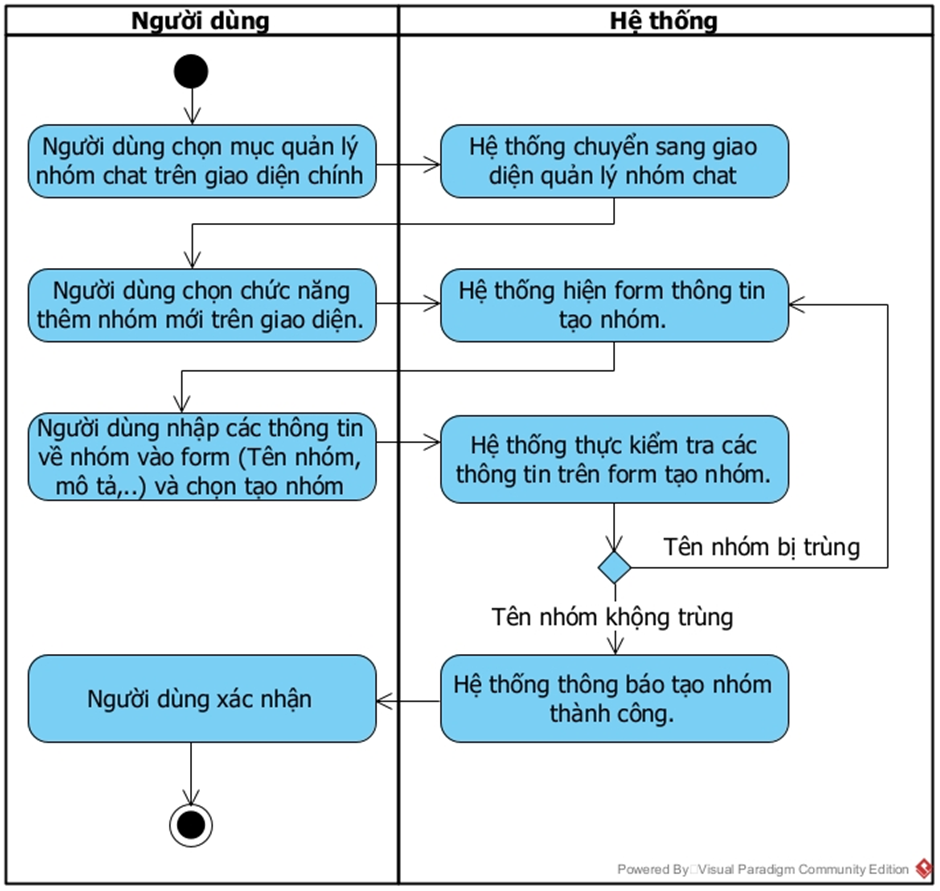
****

** Sequence Diagram**

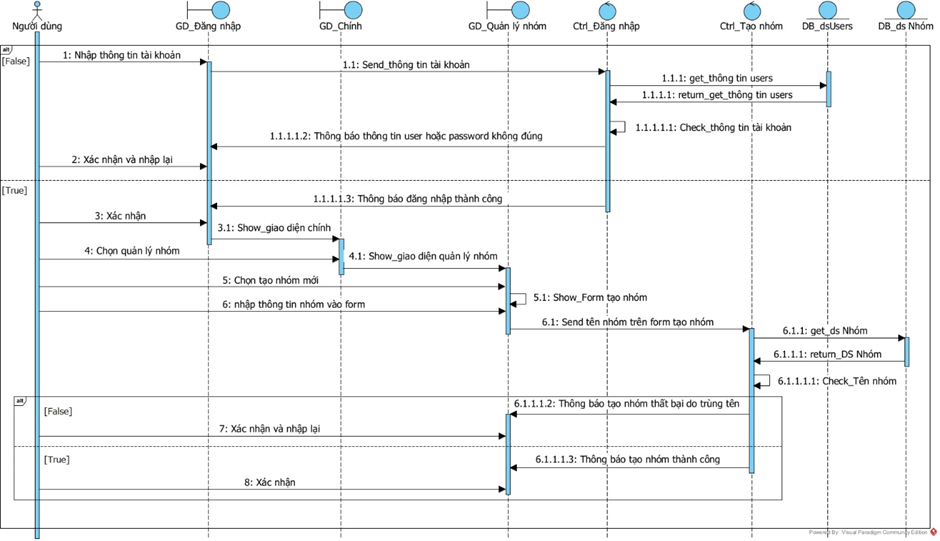
**Đặc tả use case UC07: Chức năng tạo nhóm chat**

|  |
| --- |
| **Tên use case: Tạo nhóm chat** |
| **Mô tả sơ lược: Tạo một nhóm chat để nhiều người có thể nhắn tin với nhau.** |
| **Actor chính: Người dùng** |
| **Actor phụ: không** |
| **Tiền điều kiện ( Precondition):  Actor đăng nhập thành công vào hệ thống** |
| **Hậu điều kiện ( Post – condition): Hệ thống thông báo tạo nhóm thành công và tự tạo ra một nhóm theo thiết lập của người dùng** |
| |  |  | | --- | --- | | **− Luồng sự kiện chính (main flow):** | | | **Actor** | **System** | | **1. Người dùng chọn mục quản lý nhóm chat trên giao diện chính** |  | |  | **2. Hệ thống chuyển sang giao diện quản lý nhóm chat** | | **3. Người dùng chọn chức năng thêm nhóm mới trên giao diện.** |  | |  | **4. Hệ thống hiện form thông tin tạo nhóm.** | | **5. Người dùng nhập các thông tin về nhóm vào form (Tên nhóm, mô tả,..) và chọn tạo nhóm** |  | |  | **6. Hệ thống thực kiểm tra các thông tin trên form tạo nhóm.** | |  | **7.  Hệ thống thông báo tạo nhóm thành công.** | | **8. Người dùng xác nhận và kết thúc** |  | | **− Luồng sự kiện thay thế (alternate flow):** | | |  | **2.1. Hệ thống thông báo thông tin tên nhóm bị trùng với tên nhóm đã tạo trước đó** | | **2.2. Người dùng xác nhận.** |  | |  | **2.3. Hệ thống quay lại bước 4.** | | **− Luồng sự kiện ngoại lệ (exception flow):** | | |

**Activity Diagram – Tạo nhóm**

****

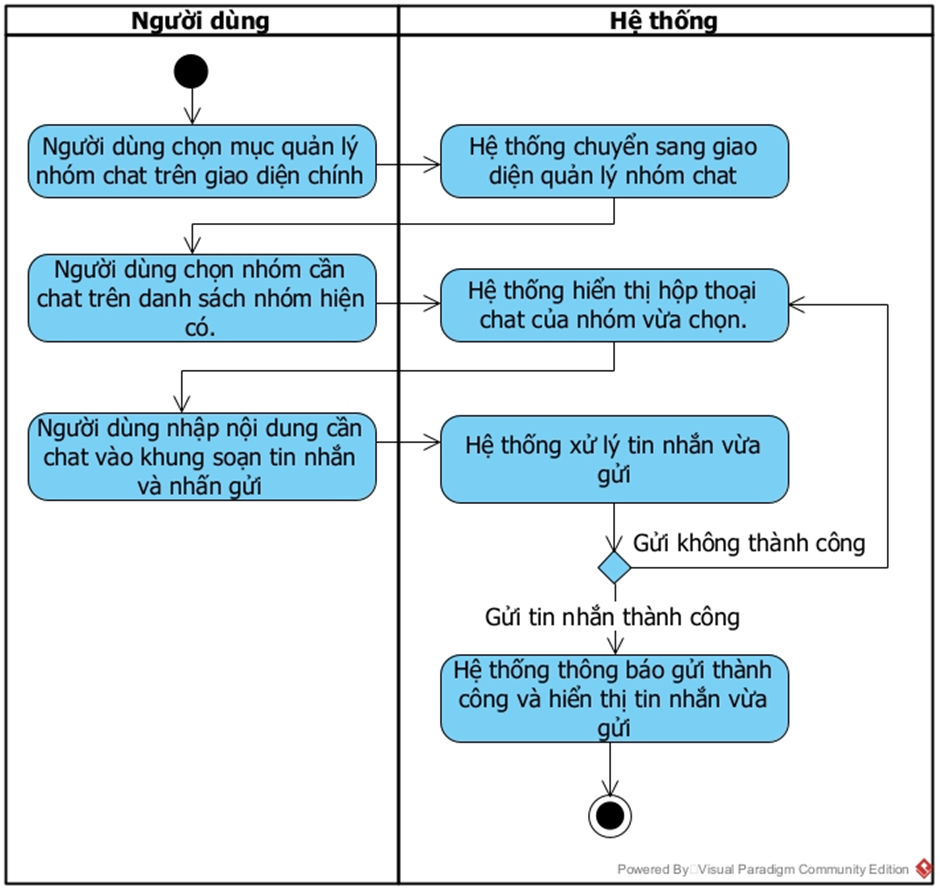
**Sequence Diagram – Tạo nhóm**

****

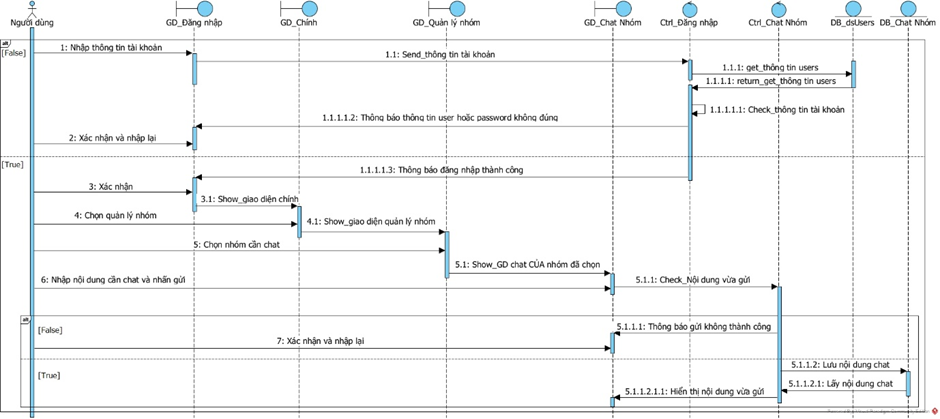
**Đặc tả use case UC08: Chức năng chat nhóm**

|  |
| --- |
| **Tên use case: Chat nhóm** |
| **Mô tả sơ lược: Tính năng chat nhóm để người dùng có thể nhắn tin cho nhiều người cùng lúc.** |
| **Actor chính: Người dùng** |
| **Actor phụ: không** |
| **Tiền điều kiện ( Precondition):  Actor đăng nhập thành công vào hệ thống** |
| **Hậu điều kiện ( Post – condition): Hệ thống lưu trữ thông tin người dùng đã nhắn lên hệ thống và hiển thị tin nhắn vừa gửi lên giao diện chat nhóm** |
| |  |  | | --- | --- | | **− Luồng sự kiện chính (main flow):** | | | **Actor** | **System** | | **1. Người dùng chọn mục quản lý nhóm chat trên giao diện chính** |  | |  | **2. Hệ thống chuyển sang giao diện quản lý nhóm chat** | | **3. Người dùng chọn nhóm cần chat trên danh sách nhóm hiện có.** |  | |  | **4. Hệ thống hiển thị hộp thoại chat của nhóm vừa chọn.** | | **5. Người dùng nhập nội dung cần chat vào khung soạn tin nhắn và nhấn gửi** |  | |  | **6. Hệ thống xử lý tin nhắn vừa gửi** | |  | **7.  Hệ thống thông báo gửi thành công và hiển thị tin nhắn vừa gửi** | |  | **8. Kết thúc** | | **− Luồng sự kiện thay thế (alternate flow):** | | |  | **6.1. Hệ thống thông báo gửi tin không thành công do quá 2000 ký tự** | | **2.2. Người dùng xác nhận.** |  | |  | **2.3. Hệ thống quay lại bước 4.** | | **− Luồng sự kiện ngoại lệ (exception flow):** | | |

**Activity Diagram – Chat nhóm**

****

**Sequence Diagram – Chat nhóm**

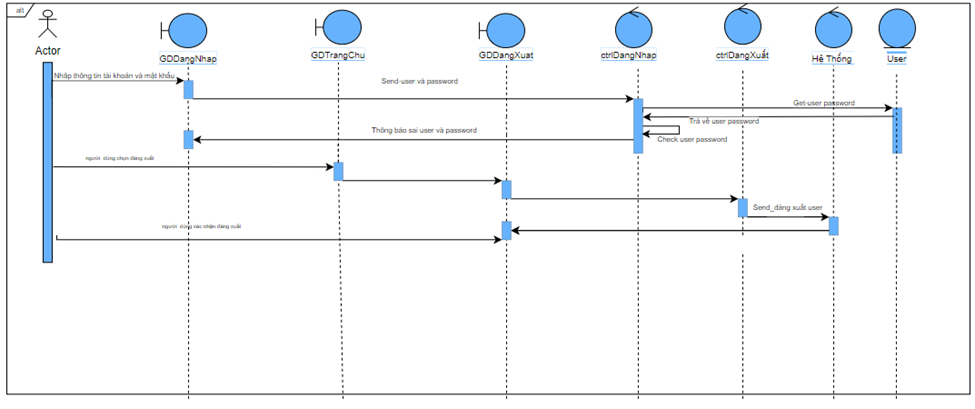
****

**Đặc tả chức năng UC09: Đăng Xuất**

|  |
| --- |
| **− Tên use case: Đăng xuất** |
| **− Mô tả sơ lược: chức năng Đăng Xuất giúp cho người dùng thoát tài khoản rakhoir ứng dụng** |
| **− Actor chính: Người dùng** |
| **− Actor phụ: không** |
| **− Tiền điều kiện (Pre-condition): Người dùng đã đăng nhập thành công vào website hoặc app mobile.** |
| **− Hậu điều kiện (Post-condition: Hệ thông thông báo Đăng Xuất thành công** |
| |  |  | | --- | --- | | **− Luồng sự kiện chính (main flow):** | | | **Actor** | **System** | | **1. Người dùng chọn đăng xuất tài khoản** |  | |  | **2. Hệ thống hiển thị trang xác nhận đăng xuất** | | **3. Người dùng nhấn vào xác nhận đăng xuất** |  | |  | **4. Hệ thống chuyển người dùng đến trang đăng nhập tài khoản** | | **5. Người dùng tương tác với giao diện đăng nhập tài khoản** |  | | **− Luồng sự kiện thay thế (alternate flow):** | | | **− Luồng sự kiện ngoại lệ (exception flow):** | | |

**Sơ đồ Activity **

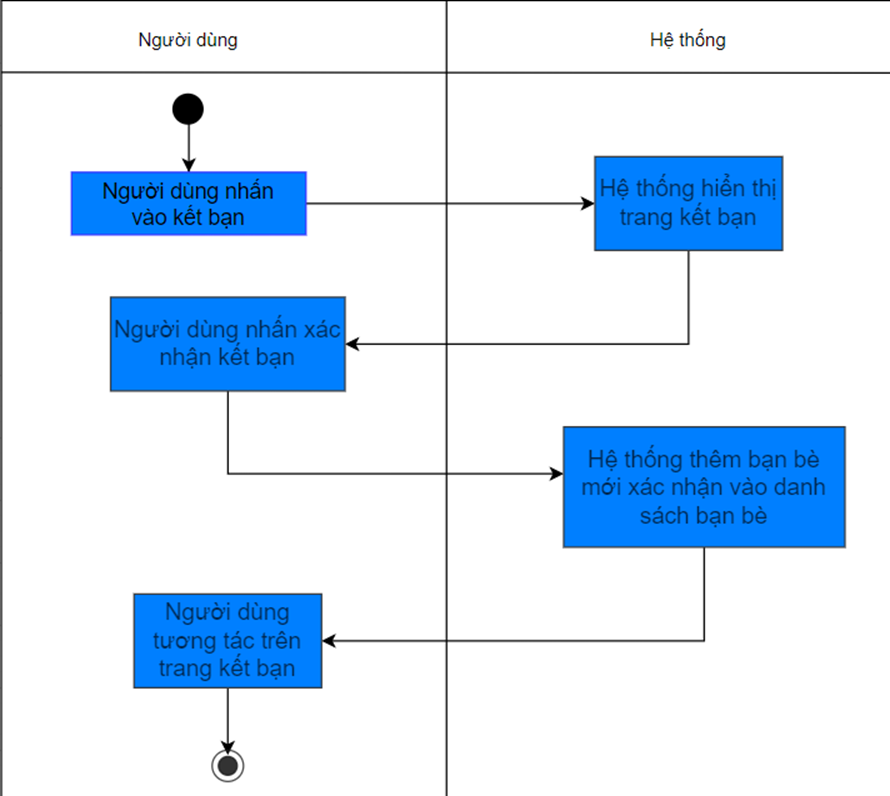
**Sequence Diagram – Đăng xuất**

****

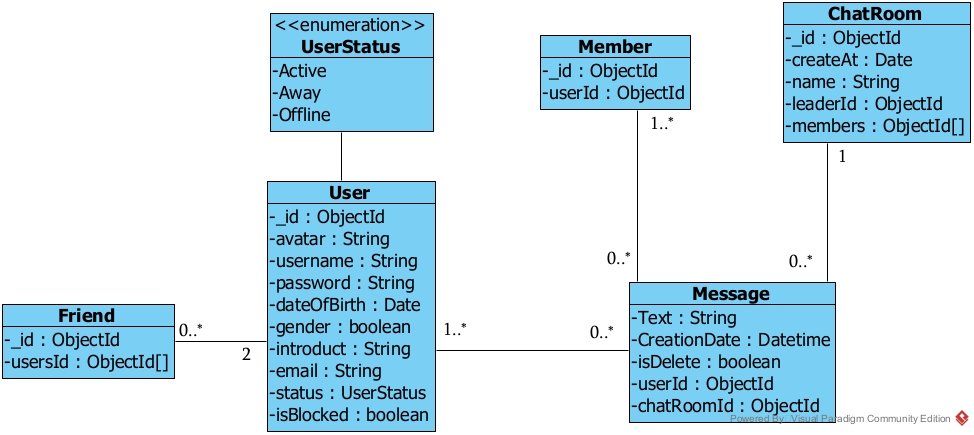
**Đặc tả chức năng use case UC10: Kết Bạn**

|  |
| --- |
| **− Tên use case: Kết Bạn** |
| **− Mô tả sơ lược: chức năng Kết Ban giúp cho người dùng them những bạn bè khác vào danh sách bạn bè** |
| **− Actor chính: Người dùng** |
| **− Actor phụ: không** |
| **− Tiền điều kiện (Pre-condition): Người dùng đã đăng nhập thành công vào website hoặc app mobile.** |
| **− Hậu điều kiện (Post-condition: Hệ thông thông báo  Kết Bạn thành công** |
| |  |  | | --- | --- | | **− Luồng sự kiện chính (main flow):** | | | **Actor** | **System** | | **1. Người dùng chọn Kết bạn** |  | |  | **2. Hệ thống hiển thị trang kết bạn** | | **3. Người dùng nhấn vào xác nhận kết bạn** |  | |  | **4. Hệ thống đưa bạn bè vừa mới xách nhận vào danh sách bạn bè** | | **5. Người dùng tiếp tục thực hiện thao tác với trang kết bạn** |  | | **− Luồng sự kiện thay thế (alternate flow):** | | | **− Luồng sự kiện ngoại lệ (exception flow):** | | |

**Sơ đồ Activity: Kết bạn**

****

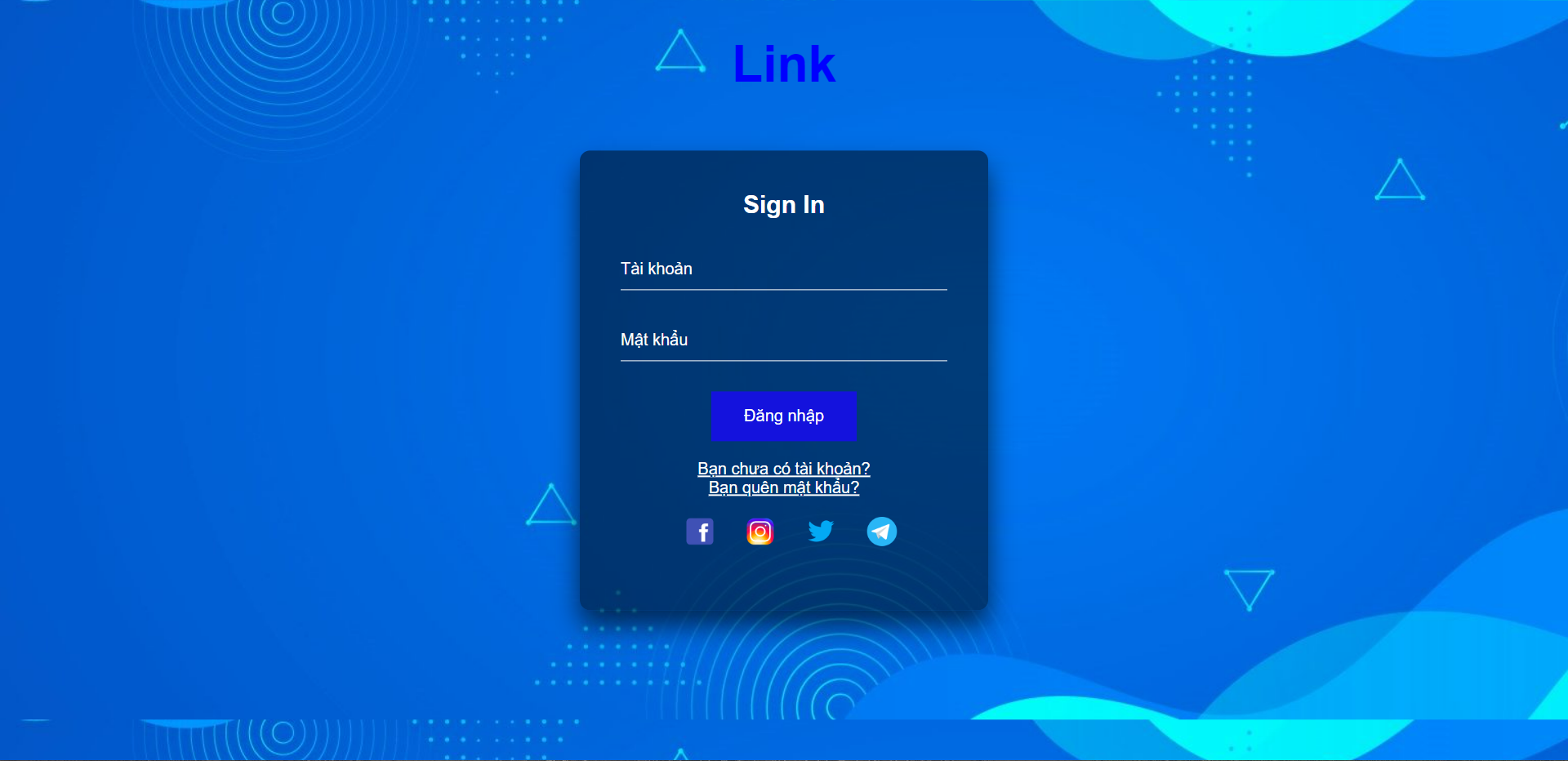
### 3.5 Class Diagram



Hình 3.5 Class Diagram

# **Chương 4: Kiểm thử hiện thực**

## 4.1 Giao diện Đăng nhập

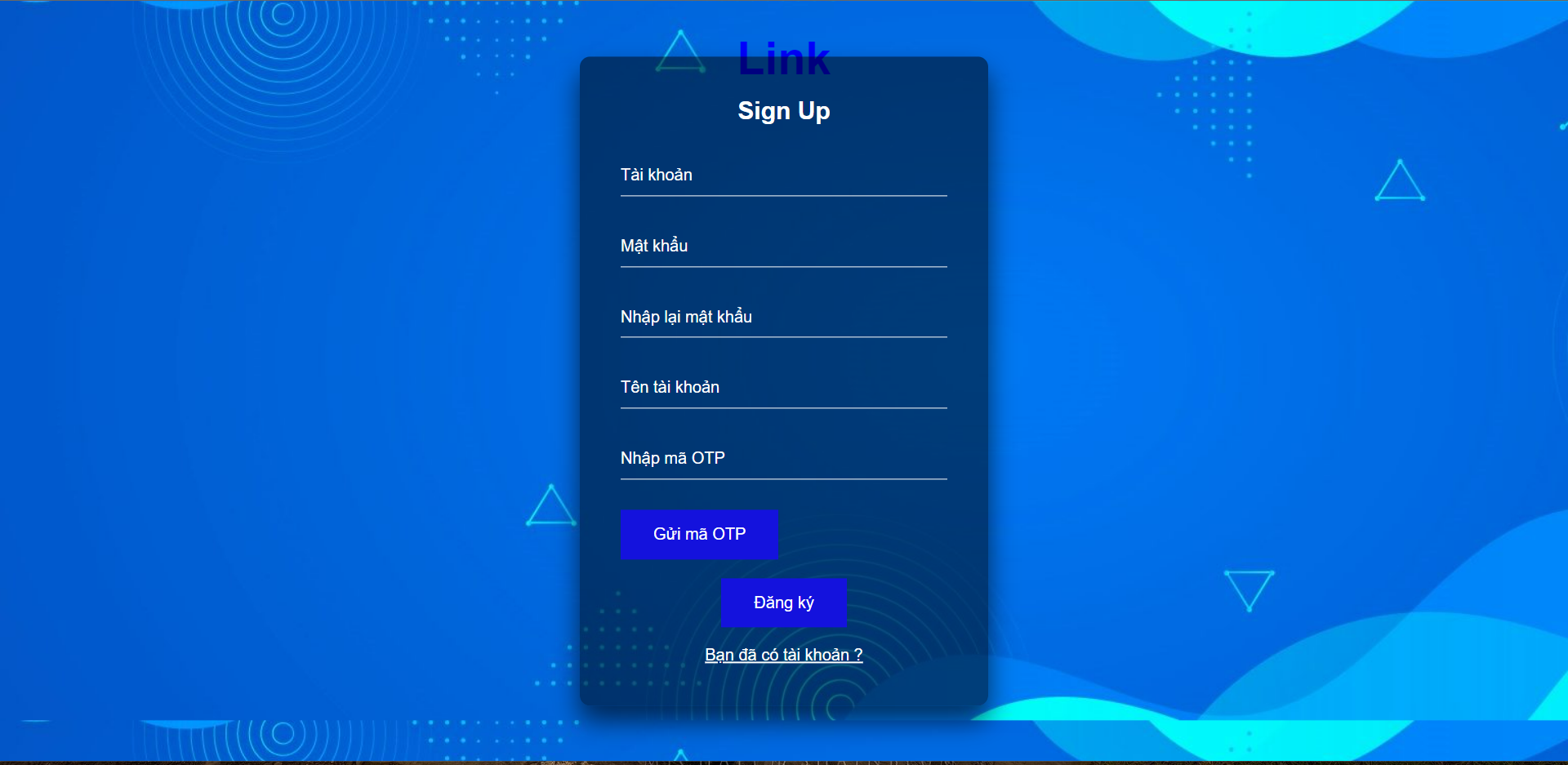


Hình 4.1 Giao diện Đăng nhập

Giao diện đăng nhập web gồm các chức năng:

1. Click vào Đăng nhập ( yêu cầu username và password) để vào giao diện người dùng
2. Click vào Bạn chưa có tài khoản? để vào trang Đăng ký tài khoản
3. Click vào Bạn quên mật khẩu để vào trang Thay đổi mật khẩu

## 4.2 Giao diện Đăng ký



Hình 4.2 Giao diện Đăng ký

Giao diện Đăng ký tài khoản yêu cầu người dùng phải nhập các thông tin như:

Số điện thoại, tên tài khoản, mật khẩu và mã OTP sẽ được gửi về số điện thoại của người dùng.

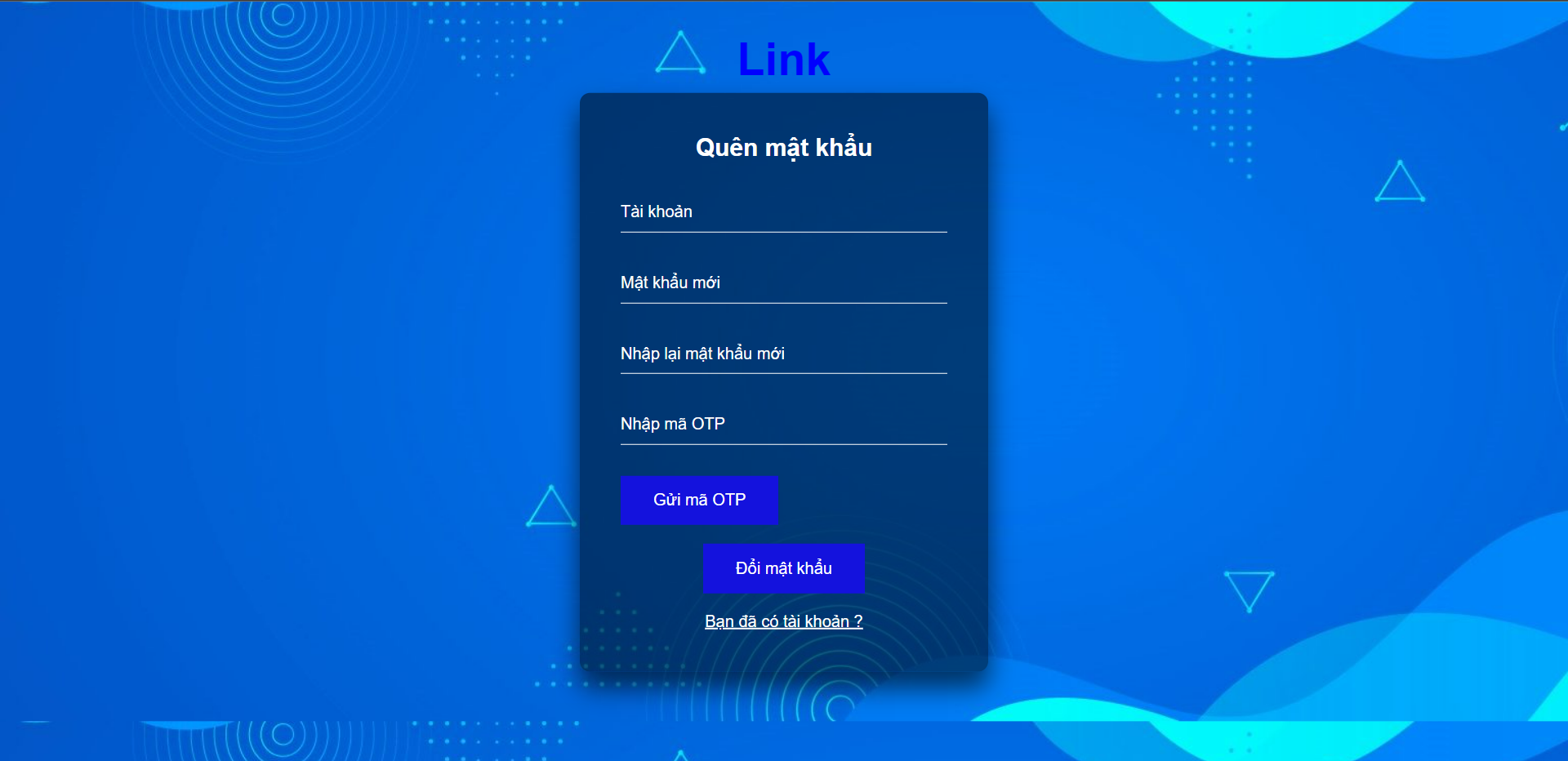
Click vào Gửi mã OTP để nhận mã xác thực trên điện thoại.

Click vào Đăng ký để tạo tài khoản ( yêu cầu nhập đầy đủ thông tin trên)

Click vào Bạn đã có tài khoản để quay lại trang Đăng nhập

## 4.3 Giao diện Quên mật khẩu

Giao diện Quên mật khẩu, người dùng cần có mã OTP để xác minh mình là chủ tài khoản để có thể thực hiện chức năng thay đổi mật khẩu.

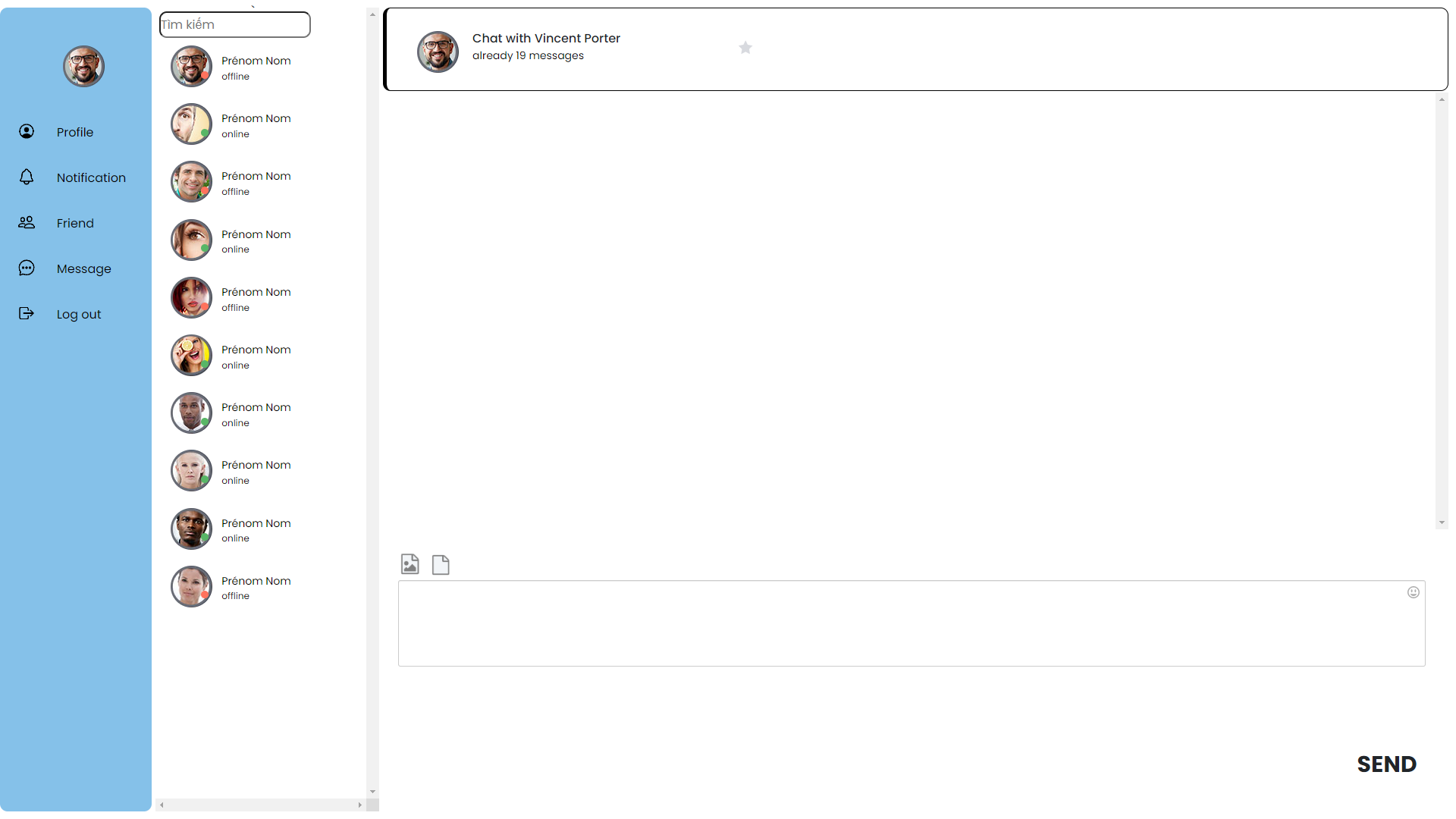
Click vào Gửi mã OTP để nhận mã xác thực trên điện thoại. 

Hình 4.3 Giao diện Quên mật khẩu

Click vào Đổi mật khẩu (yêu cầu nhập đầy đủ thông tin trên) để thay đổi mật khẩu tài khoản.

Click vào Bạn đã có tài khoản để quay lại trang Đăng nhập

## 4.4 Giao diện tin nhắn



**Hình 4.4 Giao diện người dùng**

Giao diện chính của ứng dụng:

Phần bên trái là Chat list bao gồm bạn bè, các chat room đang tồn tại.

Phần bên phải là khung chat gồm các tin nhắn giữa các người dùng với nhau.

4.5 Giao diện hồ sơ cá nhân

