**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**ỨNG DỤNG MẠNG XÃ HỘI CHAT NHÓM 15**

***Nhóm 15 - Sinh viên thực hiện***

1. Ngô Trường Giang– 19476751 ( Nhóm Trưởng )
2. Tống Gia Huy – 19470491
3. Hoàng Thị Tố Thùy – 19432541
4. Cao A Minh - 19533901
5. Nguyễn Trọng Quý - 19485331

# MỤC LỤC

[MỤC LỤC 1](#_Toc14693502)

[DANH MỤC CÁC HÌNH VẼ 1](#_Toc14693503)

[DANH MỤC CÁC BẢNG BIỂU 1](#_Toc14693504)

[CHƯƠNG 1 : GIỚI THIỆU 1](#_Toc14693505)

[1.1 Tổng quan 1](#_Toc14693506)

[1.2 Mục tiêu đề tài 1](#_Toc14693507)

[1.3 Thu thập làm rõ yêu cầu 1](#_Toc14693509)

[1.4 Phạm vi đề tài 1](#_Toc14693508)

[1.5 Mô tả yêu cầu chức năng 1](#_Toc14693509)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 1](#_Toc14693510)

[2.1 Cloud Computing với AWS 1](#_Toc14693511)

[2.2 Node.js 1](#_Toc14693512)

[2.3 Firebase 1](#_Toc14693513)

[2.4 React Native 1](#_Toc14693511)

[2.5 Visual Studio Code 1](#_Toc14693511)

[2.6 AWS Amplify 1](#_Toc14693511)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 1](#_Toc14693514)

[3.1 Phân tích yêu cầu bằng UML 1](#_Toc14693515)

[3.1.1 Usecase tổng quát 1](#_Toc14693516)

[3.1.2 Danh sách tác nhân và mô tả 1](#_Toc14693517)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 1](#_Toc14693518)

[3.1.4 Tình huống hoạt động 1](#_Toc14693519)

[3.2 Class diagram 1](#_Toc14693520)

[3.3 Deployment diagram 1](#_Toc14693521)

[CHƯƠNG 4 : HIỆN THỰC 1](#_Toc14693522)

[4.1 Cấu hình phần cứng, phần mềm 1](#_Toc14693523)

[4.2 Giao diện của hệ thống 1](#_Toc14693524)

[CHƯƠNG 5 : KẾT LUẬN 1](#_Toc14693525)

[5.1 Kết quả đạt được 1](#_Toc14693526)

[5.2 Hạn chế của đồ án 1](#_Toc14693527)

[5.3 Hướng phát triển 1](#_Toc14693528)

[TÀI LIỆU THAM KHẢO 1](#_Toc14693529)

# DANH MỤC CÁC HÌNH VẼ

[Hình 2‑1 Cloud Computing 1](#_Toc14693413)

[Hình 2‑2 Node.js development 1](#_Toc14693413)

[Hình 2‑3 CSDL Firebase 1](#_Toc14693413)

[Hình 2‑4 React Native 1](#_Toc14693413)

[Hình 2‑5 Visual Studio Code 1](#_Toc14693413)

[Hình 2‑6 AWS Amplify 1](#_Toc14693413)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1 Danh sách tác nhân và mô tả 1](#_Toc14682192)

[Bảng 3‑2 Danh sách các tình huống trong hệ thống 1](#_Toc14682193)

# : GIỚI THIỆU

## Tổng quan

Từ xưa đến nay , việc giao tiếp trong xã hội luôn là yếu tố quan trọng để phát triển bản tân và nâng cao xã hội , con người đều đã biết sự dụng các cử chỉ để miêu tả tâm tư , suy nghĩ. Phát triển thêm nữa , còn người đã sáng tạo ra ngôn ngữ để giao tiếp và trao đôi dễ dàng hơn. Cho đến ngày nay , khi cuộc cách mạng công nghiệp ra đời và cũng cho ra những công nghệ 4.0 , công nghệ đã phát triển vượt bậc , dẫn đến ngôn ngữ cũng phát triển theo , khi nhu cầu giao tiếp ngày càng cao thì những việc trao đôi qua thư từ , SMS , email là quá tối thời gian và việc gọi điện thoại qua sim là quá tốn chi phí cho người dùng. Từ những khó khắn đó , hàng loạt các giải pháp đã được đưa ra những hình thức và phù hợp nhất chính là ứng dụng Chat online.

Cũng có rất nhiều các ứng dụng chat online được tạo ra , điển hình là Zalo , Yahoo , Viber , Messenger ( Facebook) , Google chat , Twitter , Skype , ..v.v… Thành công trong số những ứng dụng này chính là Messenger chiếm lĩnh thị trường người dùng rất cao. Với giao diện đơn giản , thân thiện , hỗ trợ tối đa các tính năng như , chat và sự liên kết với facebook . Messenger đã trở thành kì lân đứng đầu trong lĩnh vực chat online . Cũng từ đó nhóm chúng em muốn phát triển một ứng dụng chat của riêng mình để thử sức , cũng như có thêm các cơ hội để tìm hiểu và nâng cao kiến thức cho bản thân trong lĩnh vực công nghệ thông tin nói chung và để hoàn thành đồ án nói riêng .

## Mục tiêu đề tài

Những mục tiêu quan trọng nhất đối với ứng dụng chat , chat chính là chức năng quan trọng nhất , được cải thiện về chức năng chat , tin nhắn được gửi trong thời gian thực , hỗ trợ icon giúp người dùng sử dụng thấy hứng thú và thích hơn.

## Thu thập , làm rõ yêu cầu đề tài

  Danh sách các câu hỏi khi thu thập và làm rõ yêu cầu của ứng dụng

|  |  |  |  |
| --- | --- | --- | --- |
| ***STT*** | ***Câu hỏi (Questions)*** | ***Trả lời (Answers)*** | ***Ghi chú*** |
| 1. | Ứng dụng có phân quyền không? | Ứng dụng có phân quyền |  |
| 2. | Ứng dụng có cho phép người dùng nhắn tin với nhiều người cùng lúc không? | Có cho phép bằng cách tạo nhóm |  |
| 3. | Ứng dụng có thể lưu trữ dữ liệu không? | Có, Ứng dụng sử dụng lưu trữ đám mây aws |  |
| 4. | Ứng dụng có cho phép thu hồi tin nhắn không? | Có cho phép thu hồi |  |
| 5. | Ứng dụng cho phép đăng ký bằng hình thức nào ? | . Ứng dụng cho phép đăng ký bằng : Số điện thoại | Yêu cầu phải nhập đúng số điện thoại  ( 10 số) |
| 6. | Mật khẩu khi đăng ký có quy định như thế nào ? | Mật khẩu đăng ký phải là mật khẩu dài hơn 8 chữ số , có viết hoa chữ cái đầu tiên. |  |
| 7. | Một số điện thoại có thể đăng ký được nhiều tài khoản không ? | Không . Mỗi số điện thoại chỉ cho phép đăng ký 1 tài khoản. |  |
| 8. | Có thể đổi mật khẩu được không? | Có, ứng dụng có chức năng hỗ trợ người dùng đổi mật khẩu |  |
| 9 | Ai có quyền được xem tin nhắn của người khác? | Không ai có quyền xem tin nhắn của người khác |  |
| 10 | Ứng dụng có khả năng mở rộng thêm các tính năng mới không? (ví dụ: chat video) | . Có, ứng dụng có thể mở rộng thêm nhiều tính năng khác |  |
| 11 | Ứng dụng sẽ có trên nền tảng nào | Ứng dụng sẽ có trên web và android |  |
| 12 | Ứng dụng được tạo bằng công nghệ gì? | Ứng dụng sẽ được tạo ra trên di động bằng java,trên web bằng reactjs và nodejs ,lưu trữ bằng Firebase và sử dụng AWS |  |
| 13 | Ứng dụng có cho phép người dùng nhắn tin trao đổi file không? | Ứng dụng cho phép người dùng trao đổi file |  |
| 14 | Ứng dụng có cho phép xóa tin nhắn không? | Ứng dụng cho phép người dùng xóa tin   nhắn. |  |
| 15 | Có thể tìm kiếm bạn bè thông qua số điện thoại không? | Có thể tìm kiếm bạn bè thông qua số điện thoại. |  |
| 16 | Điều kiện để tạo nhóm phải từ bao nhiêu người? | Để tạo nhóm phải có từ 3 người trở lên. |  |

## Phạm vi đề tài

- Phát triển trên nền tảng Web và Android

-Phần mềm được dùng bởi mọi người.

-Phần mềm dùng online, bảo mật.

-Phần mềm thân thiện với người dùng và có khả năng cập nhật, sửa lỗi trong tương lai.

-Phần mềm dùng ở mọi lúc mọi nơi nếu dùng có thiết bị truy cập website ứng dụng và có internet.

-Phần mềm được thiết kế hoàn toàn bằng tiếng Việt giúp cho việc sử dụng đơn giản hơn.

-Phần mềm có phím tắc, có chức năng sao lưu dữ liệu, tương thích với hệ điều hành Window 10 của máy tính cửa hàng.

-Phần mềm dùng công nghệ Java Spring boot, ReactJs, dùng CSDL Firebase

## Mô tả yêu cầu chức năng

**a.Yêu cầu chức năng**

**-** Chức năng đăng ký, đăng nhập tài khoản:

Đăng ký tài khoản: để đăng ký tài khoản người dùng cần sử dụng bằng email sau khi xác thực thành công bạn chỉ cần nhập đầy đủ thông tin theo hướng dẫn.

Đăng nhập tài khoản: đúng email và mật khẩu đã đăng ký

-Chức năng nhắn tin (Chat): : text, image, video, document,

emotion (tương tự zalo) .

+ Chat 1-1: cho phép chỉ 2 người nhắn tin, giao tiếp với nhau.

+ Chat nhóm: cho phép nhiều người cùng tham gia vào nhóm và có thể nhắn tin, trao đổi với nhau (để tạo nhóm tối thiểu phải có 3 người dùng).

- Chức năng tìm kiếm: Ứng dụng có chức năng tìm kiếm tin nhắn

- Chức năng nhắn tin (Chat Video Call): gọi video call 1-1

- Chức năng cập nhật, sửa đổi ( trang thời gian): Ứng dụng có chức năng chỉnh sửa thông tin cá nhân như: ảnh đại diện, ảnh bìa, họ và tên, ngày tháng năm sinh, giới tính, đổi số điện, đổi mật khẩu.

-Chức năng đăng xuất: đăng xuất

**b.Yêu cầu phi chức năng**

- Hiệu xuất các tác vụ cơ bản nhập, xuất, thêm, xóa, sửa, sắp xếp, truy xuất thông tin phải chính xác và nhanh chóng.

- Dữ liệu có khả năng mở rộng và lưu trữ lớn.

- Các dữ liệu thống kê phải có cơ sở và tính khả dụng cao.

- Phần mềm phải có tính bảo mật dữ liệu, tính sẵn sàng dữ liệu khi truy xuất ở các máy tính cá nhân, tính toàn vẹn dữ liệu theo thời gian.

- Dữ liệu phải được sao lưu, khôi phục thường xuyên.

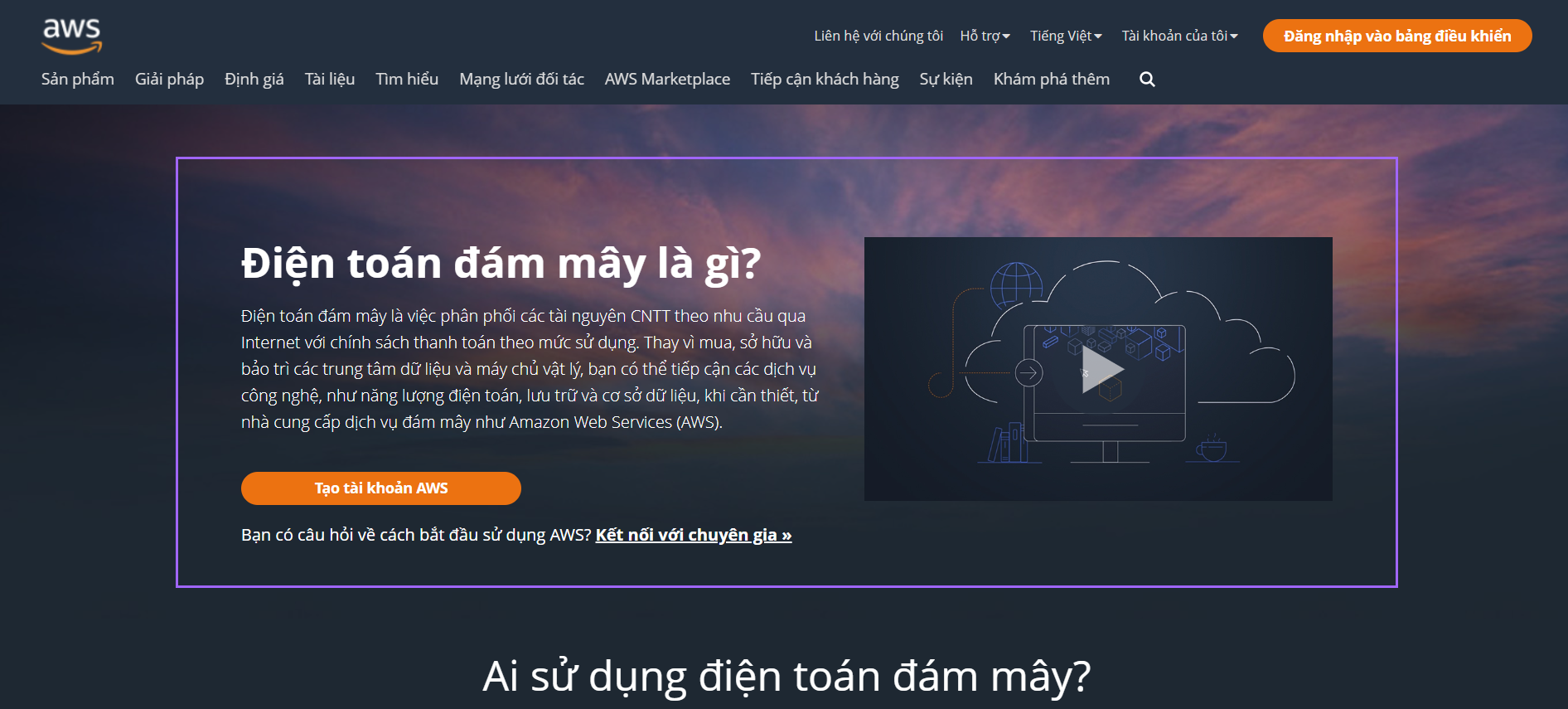
- Phần mềm phải có khả năng bảo trì, nâng cấp chức năng khi cần thiết.

- Phần mềm phải tương thích với trình duyệt Chrome,...

- Phần mềm phải hỗ trợ nhập và thao tác thân thiện với người dùng.

# : CƠ SỞ LÝ THUYẾT

## Cloud Computing với AWS



Hình 2-1. Cloud computing với AWS

**Cloud computing ( Điện toán đám mây )** là dịch vụ điện toán dựa trên nền tảng internet , nó ttaajp hợp rất nhiều các máy chủ được kết nối lại với nhau thông qua internet , nó cho phép lưu trữ dữ liệu dạng tập trung và có thể truy cập trực tuyến vào các dịch vụ hoặc túi tài nguyên.

Sử dụng điện toán đám mây , các tổ chức có thể sử dụng các tài nguyên có sẵn trên Cloud thay vì tự mình xấy dựng , vận hành và cải thiện cơ sở hạ tầng.

Điện toán đám mây là một mô hình cung cấp các tính năng sau :

* Người dùng có thể cung cấp và giải phóng tài nguyên theo yêu cầu
* Tài nguyên có thể được tăng hoặc giảm tự động tùy thuộc vào lượng tải
* Các tài nguyên có thể truy cập được qua mạng với các cơ chế bảo mật thích hợp
* Người dùng có thể sử dụng tính năng **pay-as-you-go** với tính năng này khách hàng được tính phí dựa trên loại tài nguyên và lượng sử dụng tài nguyên đó.

## Node.js



Hình ‑2 Node.js development

**NodeJS** là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

**NodeJS** được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

Trong một môi trường server điển hình LAMP (Linux-Apache-MySQL-PHP), bạn có một web server là Apache hoặc NGINX nằm dưới, cùng với PHP chạy trên nó. Mỗi một kết nối tới server sẽ sinh ra một thread mới, và điều này khiến ứng dụng nhanh chóng trở nên chậm chạp hoặc quá tải - cách duy nhất để hỗ trợ nhiều người dùng hơn là bằng cách bổ sung thêm nhiều máy chủ. Đơn giản là nó không có khả năng mở rộng tốt. Nhưng với Node.js thì điều này không phải là vấn đề. Không có một máy chủ Apache lắng nghe các kết nối tới và trả về mã trạng thái HTTP - bạn sẽ phải tự quản lý kiến trúc lõi của máy chủ đó. May mắn thay, có một số module giúp thực hiện điều này được dễ dàng hơn, nhưng công việc này vẫn gây cho bạn một chút khó khăn khi mới bắt đầu. Tuy nhiên, kết quả thu được là một ứng dụng web có tốc độ thực thi cao.

**JavaScript** là một ngôn ngữ dựa trên sự kiện, vì vậy bất cứ thứ gì xảy ra trên server đều tạo ra một sự kiện non-blocking. Mỗi kết nối mới sinh ra một sự kiện; dữ liệu nhận được từ một upload form sinh ra một sự kiện data-received; việc truy vấn dữ liệu từ database cũng sinh ra một sự kiện. Trong thực tế, điều này có nghĩa là một trang web Node.js sẽ chẳng bao giờ bị khóa (lock up) và có thể hỗ trợ cho hàng chục nghìn user truy cập cùng lúc. Node.js đóng vai trò của server - Apache - và thông dịch mã ứng dụng chạy trên nó. Giống như Apache, có rất nhiều module (thư viện) có thể được cài đặt để bổ sung thêm các đặc trưng và chức năng - như lưu trữ dữ liệu, hỗ trợ file Zip, đăng nhập bằng Facebook, hoặc các cổng thanh toán. Dĩ nhiên, nó không có nhiều thư viện như PHP, nhưng Node.js vẫn đang ở trong giai đoạn ban đầu và có một cộng đồng rất mạnh mẽ ở đằng sau nó.

Một khái niệm cốt lõi của Node.js đó là các function bất đồng bộ (asynchronous functions) - vì vậy về cơ bản thì mọi thứ chạy trên nền tảng này. Với hầu hết các ngôn ngữ kịch bản máy chủ, chương trình phải đợi mỗi function thực thi xong trước khi có thể tiếp tục chạy tiếp. Với Node.js, bạn xác định các function sẽ chạy để hoàn thành một tác vụ nào đó, trong khi phần còn lại của ứng dụng vẫn chạy đồng thời. Nó là một chủ đề phức tạp mà tôi sẽ không đi vào quá sâu trong bài viết này, nhưng đó là một trong những đặc trưng tiêu biểu của Node.js, vì vậy việc nắm vững nó là điều hết sức quan trọng.

## Firebase



Hình ‑3 Cơ sở dữ liệu Firebase

**Firebase** là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Gần một thập niên trước, **Firebase** ra đời với tiền thân là Envolve. Đây là một nền tảng đơn giản chuyên cung cấp những API cần thiết để tích hợp tính năng chat vào trang web. Bên cạnh ứng dụng nhắn tin trực tuyến, Envolve còn được người dùng sử dụng để truyền và đồng bộ hóa dữ liệu cho những ứng dụng khác như các trò chơi trực tuyến,… Do đó, các nhà sáng lập đã tách biệt hệ thống nhắn tin trực tuyến và đồng bộ dữ liệu thời gian thực thành hai phần riêng biệt.

Trên cơ sở đó, năm 2012, Firebase ra đời với sản phẩm cung cấp là **dịch vụ Backend-as-a-Service**. Tiếp đến, vào năm 2014, Google mua lại Firebase và phát triển nó thành một dịch vụ đa chức năng được hàng triệu người sử dụng cho đến hiện nay.

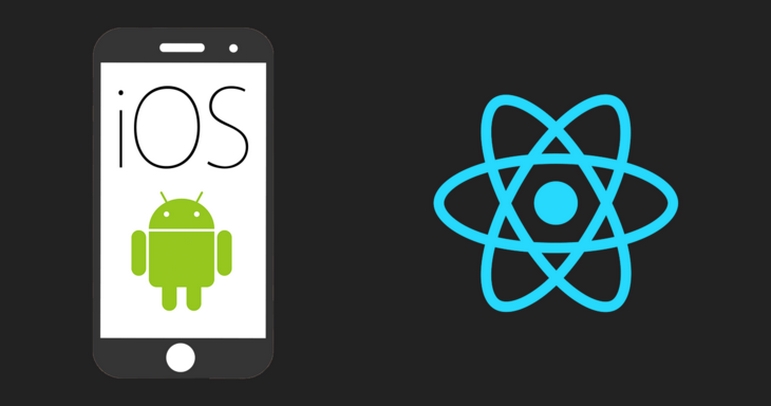
**Ưu điểm của FireBase :**

* Tạo tài khoản và sử dụng dễ dàng
* Tốc độ phát triển nhanh
* Nhiều dịch vụ trong một nền tảng
* Được cung cấp bởi Google
* Tập trung vào phát triển giao diện người dùng
* Firebase không có máy chủ
* Học máy (Machine Learning)
* Tạo lưu lượng truy cập
* Theo dõi lỗi
* Sao lưu

## React Native là gì và tại sao nó lại quan trọng? | IceTea Việt | Lập trình cuộc sốngReact Native

Hình 2-4 Framework React Native

**React Native** là 1 framework mã nguồn mở và đã được sáng tạo nên bởi Facebook, được dùng để có thể phát triển ứng dụng Web, UWP, iOS và Android bằng cách cho phép những nhà phát triển dùng React cùng môi trường ứng dụng gốc.

**Nguyên tắc hoạt động của React Native** :

Về nguyên tắc hoạt động thì nó gần như giống với React trừ việc React Native hiện tại không thao tác cùng với DOM qua DOM ảo. Nó đang chạy 1 quá trình xử lý nền (Được phiên dịch JavaScript và viết bởi những nhà phát triển) trực tiếp ngay trên thiết bị đầu cuối cũng như giao tiếp cùng với nền tảng góp phần qua một cầu trung gian, theo đợt và bất đồng bộ.

## Visual Studio Code



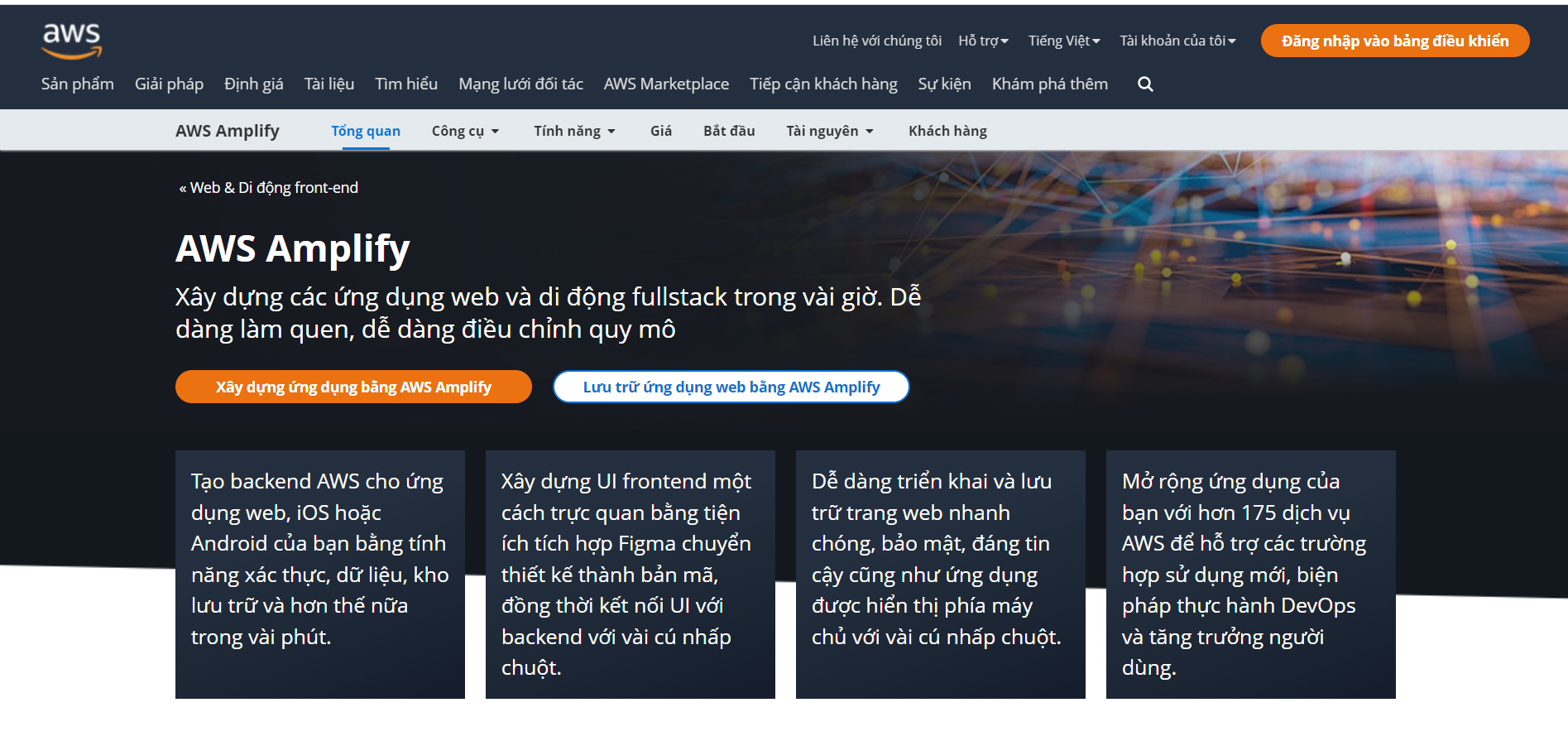
Hình 2-5 Visual Studio Code

**Visual Studio Code** (còn được gọi là VSCode) được biết là phần mềm soạn thảo, biên tập các đoạn code để hỗ trợ trong quá trình xây dựng và phát triển website một cách nhanh chóng. Phần mềm soạn thảo code Visual Studio Code hoạt động ổn định trên nhiều nền tảng máy tính phổ biến khác nhau như: Windows, MacOS, Linux. Điều giúp cho phần mềm lập trình Visual Studio Code được nhiều người dùng yêu thích đó là khả năng tương thích, VS Code hoạt động tốt trên nhiều cấu hình máy tính khác nhau từ tầm trung cho đến cao cấp.

Visual Studio Code được xây dựng và phát triển bởi tập đoàn Microsoft, chúng là sự kết hợp hoàn hảo giữa IDE và Code Editor. Phần mềm soạn code Visual Studio Code hiện đang được phát hành miễn phí trên các nền tảng Linux, Microsoft và MacOS nhằm giảm chi phí đầu tư cho các lập trình viên.

Phần mềm Visual Studio Code hỗ trợ đa dạng các chức năng Debug, đi kèm với Git, có Syntax Highlighting… Đặc biệt là chúng có khả năng tự hoàn thành mã thông minh, Snippets và khả năng nâng cấp cải tiến mã nguồn mở. Giao diện của VS Code có khả năng tùy biến mạnh mẽ, các lập trình viên có thể thay đổi Theme, phím tắt, thay đổi bố cục làm việc sao cho phù hợp nhất với nhu cầu của bản thân

## AWS Amplify



Hình 2-6 AWS Amplify

**AWS Amplify là** bộ framework và công cụ hỗ trợ tích hợp backend(sử dụng các service cua AWS) với frontend trên các nền tảng:

   iOS

   Android

   Web(React, Angular/Ionic, Vue)

   Đa nền tảng(React native)

Hỗ trợ tự động cấu hình tài nguyên AWS cần thiết, deploy code, gen code tự động

**Cách thức hoạt động:**

AWS Amplify là một giải pháp hoàn chỉnh giúp nhà phát triển web và di động frontend xây dựng, vận chuyển và lưu trữ các ứng dụng fullstack một cách dễ dàng trên AWS với khả năng tận dụng linh hoạt các dịch vụ AWS phong phú khi các trường hợp sử dụng tiến triển.

# : PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| User |  |
| Member |  |
| Admin |  |
|  |  |
|  |  |
| … |  |

Bảng ‑ Danh sách tác nhân và mô tả

### Danh sách các tình huống hoạt động (Use cases)

|  |  |
| --- | --- |
| ID | Tên Use case |
| UC01 | Đăng nhập |
| UC02 | Đăng xuất |
| UC03 | Tìm kiếm tin nhắn |
| UC04 |  |
| UC05 |  |
| UC06 |  |
|  |  |
|  |  |
| … |  |

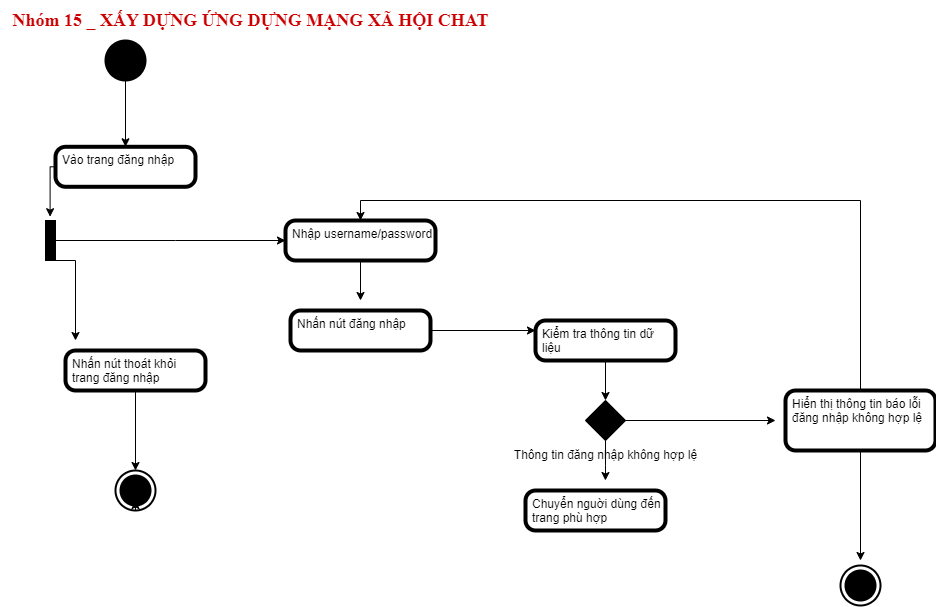
Bảng ‑ Danh sách các tình huống trong hệ thống

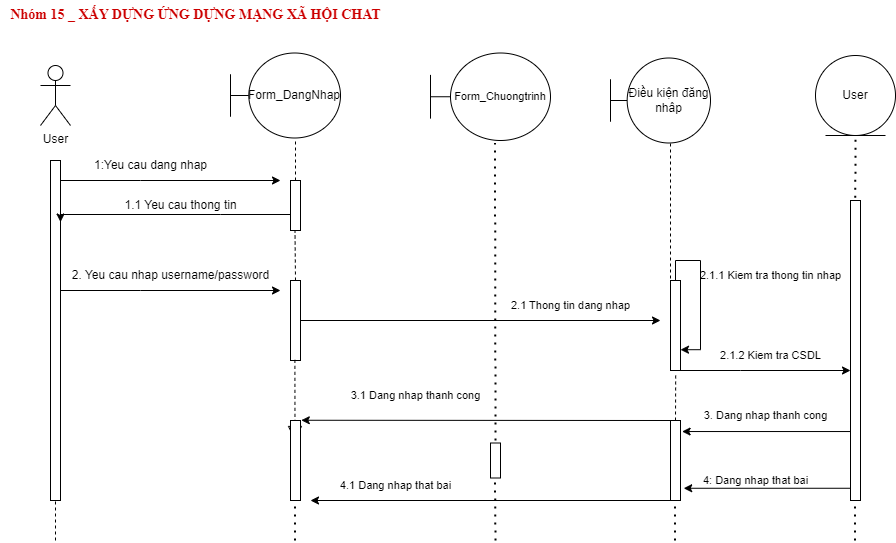
### Tình huống hoạt động

#### Mô tả use case UC01

|  |  |
| --- | --- |
| **Use case:** UC001\_ Đăng nhập | |
| Mục đích: | Usecase này giúp người dùng đăng nhập vào hệ thống |
| Mô tả: | Usecase này mô tả cách người dùng đăng nhập vào hệ thống |
| Tác nhân: | Quản trị + cá nhân người dùng |
| Điều kiện trước: | Hệ thống yêu cầu người dùng nhập tên và mật khẩu |
| Điều kiện sau: | Người dùng nhập tên và mật khẩu |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng vào trang đăng nhập 2. Người dùng nhập usename , password 3. Nhấn nút đăng nhập 4. Hệ thống kiểm tra thông tin đăng nhập 5. Nếu thông tin đăng nhahap hợp lệ , hệ thống chuyển người dùng đến trang chủ phù hợp |
| Luồng sự kiện phụ (Alternative Flows): | Nếu trong dòng sự kiện chính người dùng nhập tên và mật khẩu sai thì hệ thống sẽ thông báo lỗi. Người dùng có thể quay trở về hoặc huỷ bỏ việc đăng nhập, lúc này usecase kết thúc. |

* **Activity diagra:**



* **Sequence diagram**

## Class diagram

Mô hình lớp (Mô hình lớp đầy đủ + Mô hình lớp bao gồm các lớp Entity)

## Database diagram

Mô hình CSDL

## Deployment diagram

Mô hình triển khai hệ thống.

# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

Ghi rõ thông tin sử dụng các dịch vụ nào của AWS để thực hiện đồ án.

Cấu hình phần cứng, phần mềm của Server (nếu có).

Cấu hình phần cứng, phần mềm của Client trong hệ thống (nếu có).

## Giao diện của hệ thống

Màn hình chức năng đã thực hiện.

## Kế hoạch và hiện thực kiểm thử hệ thống

### Kế hoạch kiểm thử

### Kiểm thử hệ thống

# : KẾT LUẬN

## Kết quả đạt được

## Hạn chế của đồ án

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Anh

1. Kevin Faaborg & Sandro Pasquali - Mastering Node.js Second Edition – December 2017
2. David Herron, Diogo Resende & Valentin Bojinov - Node.js Complete Reference Guide – December 2018

Các tài liệu từ Internet

1. <https://nodejs.org/en/docs/>
2. <https://expressjs.com/en/guide/routing.html>
3. https://itviec.com