*Shape, square

Description automatically generated*

**TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN**

**VIỆN CÔNG NGHỆ THÔNG TIN & KINH TẾ SỐ**

Logo

Description automatically generated**------o0o------**

**BÁO CÁO CHUYÊN ĐỀ THỰC TẬP**

**XÂY DỰNG VÍ ĐIỆN TỬ NỘI BỘ VÀ HỆ THỐNG THANH TOÁN TỰ ĐỘNG DỰA TRÊN NỀN TẢNG IOT**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **TS. Phạm Minh Hoàn** |
| **Sinh viên:** | **Ngô Khôn Tấn** |
| **Mã sinh viên:** | **11203502** |
| **Chuyên ngành:** | **Công nghệ thông tin** |
| **Lớp chuyên ngành:** | **Công nghệ thông tin 62B** |

# **MỤC LỤC**

[**MỤC LỤC** 2](#_Toc150728325)

[**DANH MỤC VIẾT TẮT** 4](#_Toc150728326)

[**DANH MỤC HÌNH ẢNH** 5](#_Toc150728327)

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI** 6](#_Toc150728328)

[1.1 Đặt vấn đề 6](#_Toc150728329)

[1.2 Mục tiêu đề tài 7](#_Toc150728330)

[1.3 Yêu cầu 8](#_Toc150728331)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 8](#_Toc150728332)

[2.1 Phân tích bài toán 8](#_Toc150728333)

[2.1.1 Phân tích yêu cầu bài toán 8](#_Toc150728334)

[2.1.2 Phân tích chức năng hệ thống 11](#_Toc150728335)

[2.1.3 Phân tích cấu trúc 23](#_Toc150728336)

[2.1.4 Phân tích hành vi 24](#_Toc150728337)

[2.2 Thiết kế 29](#_Toc150728338)

[2.2.1 Thiết kế cấu trúc 29](#_Toc150728339)

[2.2.2 Thiết kế cơ sở dữ liệu 30](#_Toc150728340)

[2.2.3 Thiết kế giao diện 33](#_Toc150728341)

[**CHƯƠNG 3: CÁC CÔNG NGHỆ, NỀN TẢNG ÁP DỤNG VÀO XÂY DỰNG HỆ THỐNG** 33](#_Toc150728342)

[Nội dung chương này liên quan gì đến đề tài? 33](#_Toc150728343)

[3.1 Kiến trúc Web Client – Server 33](#_Toc150728344)

[3.1.1 Tổng quan mô hình Client-Server 33](#_Toc150728345)

[3.1.2 Nguyên tắc hoạt động của mô hình Client-Server 34](#_Toc150728346)

[3.1.3 Ưu, nhược điểm của mô hình Client-Server 34](#_Toc150728347)

[3.2 HTML, CSS 35](#_Toc150728348)

[3.2.1 Html 35](#_Toc150728349)

[2.2.2 CSS 36](#_Toc150728350)

[3.3 Javascript 37](#_Toc150728351)

[3.3.1Client - máy khách: 38](#_Toc150728352)

[3.3.2 Server – máy chủ: 38](#_Toc150728353)

[3.4 Nodejs và Express.js 39](#_Toc150728354)

[3.5 Rest API và Json 41](#_Toc150728355)

[3.5.1 Restful API 41](#_Toc150728356)

[3.6 Firebase – Firebase cloud firestore 42](#_Toc150728357)

[Firebase Cloud Firestore 43](#_Toc150728358)

[3.7 Công nghệ RFID 44](#_Toc150728359)

[3.8 Hệ thống điều khiển IOT (Module wifi Esp32) 46](#_Toc150728360)

[3.9 Cổng thanh toán (Payment Gateway) : VNPay 48](#_Toc150728361)

[3.9.1 Cổng thanh toán 48](#_Toc150728362)

[2.8.2 VNPay 50](#_Toc150728363)

[**CHƯƠNG 4: XÂY DỰNG, CÀI ĐẶT VÀ THỬ NGHIỆM** 52](#_Toc150728364)

[4.1 Xây dựng, cài đặt hệ thống 52](#_Toc150728365)

[4.2 Thử nghiệm 53](#_Toc150728366)

[4.2.1 Người quản trị (Admin) 53](#_Toc150728367)

[4.2.2 Người dùng (User) 54](#_Toc150728368)

[4.3 Đánh giá kết quả 55](#_Toc150728369)

[4.3.1 Ưu điểm 55](#_Toc150728370)

[4.3.2 Nhược điểm 55](#_Toc150728371)

[**KẾT LUẬN** 56](#_Toc150728372)

[**TÀI LIỆU THAM KHẢO** 56](#_Toc150728373)

# **DANH MỤC VIẾT TẮT**

# **DANH MỤC HÌNH ẢNH**

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

## 1.1 Đặt vấn đề

Ngày nay con người chúng ta đang chứng kiến một làn sóng cách mạng công nghệ mạnh mẽ. Làn sóng ấy tác động mạnh mẽ lên chính con người đồng thời làm thay đổi cách chúng ta sống, làm việc và học tập. Nó mở ra một tầm nhìn mới về những thách thức và cơ hội hiện hữu, đồng thời tôn vinh vai trò không thể thiếu của công nghệ trong việc làm cho cuộc sống hàng ngày trở nên tốt hơn. Trong tầm nhìn đó, việc quản lý và tận dụng tài sản trở thành một bài toán cần phải giải quyết một cách thông minh.

Khi thế giới đang bước vào thời kỳ kinh tế số hóa với sự trỗi dậy của thương mại điện tử và các dịch vụ trực tuyến, hình thức thanh toán không còn dựa vào tiền mặt đã trở nên phổ biến. Thanh toán không dùng tiền mặt đem đến sự tiện lợi cho người dùng và chính điều đó đã thúc đẩy lượng lớn người dùng lựa chọn hình thức này thay cho việc sử dụng tiền mặt. Ví điện tử đã xuất hiện như một giải pháp hoàn hảo, đáp ứng nhu cầu ngày càng cao của mọi người. Trong khoảng thời gian 3 năm đổi lại đây, một số ví điện tử có tốc độ phát triển rất nhanh chóng. Phương thức này giúp người dân linh hoạt hơn trong giao dịch, tiết kiệm thời gian và dễ dàng quản lý tài sản của bản thân phù hợp với điệu kiện công nghệ và nhu cầu trong bối cảnh hiện nay.

Mặt khác không thể nhắc đến trong làn sóng cách mạng công nghệ đang phát triển hiện nay là IOT. Cho phép các đối tượng có khả năng kết nối internet và trao đổi dữ liệu với nhau thông qua internet mà không cần sự tác động trực tiếp của con người. Và IOT cũng được ứng dụng vào trong lĩnh vực thanh toán. Với sự kết hợp công nghệ IOT và quá trình thanh toán đã tạo một bước đột phá, tạo nên môi trường số hóa thông minh. Khiến cho việc thanh toán không dùng tiền mặt lại càng tiện lợi, nhanh chóng hơn và đặc biệt là tự động thông qua các thiết bị được kết nối.

Trường Đại học Kinh tế Quốc dân luôn được biết đến là trường đại học hàng đầu về khối ngành kinh tế, đội ngũ giảng viên chất lượng, chương trình đào tạo hiệu quả, cơ sở vật chất hang đầu. Nhà trường trang bị cho sinh viên về kiến thức đồng thời còn cung cấp nhiều dịch vụ hỗ trợ đầy đủ tiện nghi như: canteen, thư viện, y tế, … Và dịch vụ trông giữ xe của trường cũng là một dịch vụ đang được đông đảo các bạn sinh viên quan tâm.



Tuy nhiên thực trạng quan sát được là tình trạng gửi xe còn đang gặp vấn đề khó khăn. Vào những lúc cao điểm lượng sinh viên gửi xe tăng cao gây ra vấn đề ùn tắc ảnh hưởng đến trải nghiệm và quá trình học tập của sinh viên. Một trong những nguyên nhân gây ra tình trạng này là công tác thu tiền gửi xe.

Đồng thời sau khi tìm hiểu thông tin qua một số sinh viên tại trường đại học Kinh tế Quốc dân, bản thân em nhận thấy hiện nay có nhiều sinh viên có thói quen khi đóng học phí sẽ đóng dư ra vì nhiều lí do khác nhau. Số tiền đó như là tiền chết nằm nguyên trong tài chính sinh viên cho đến lần đóng học phí tiếp theo. Bên cạnh đó mỗi lần trao học bổng hay khen thưởng sinh viên nhà trường lại phải yêu cầu sinh viên kê khai, cung cấp lại số tài khoản cá nhân chính việc đó gây mất thời gian.

Sau khi tìm hiểu và nhận thức được sự bất cập hiện tại bản thân em đã quyết định thực hiện nghiên cứu giải pháp và đó là lí do em lựa chọn đề tài : “ xây dựng ví điện tử nội bộ và hệ thống thanh toán tự động dựa trên nền tảng IOT”.

## 1.2 Mục tiêu đề tài

Đề tài sẽ tập trung vào xây dựng ví điện tử nội bộ ở Trường đại học Kinh tế Quốc dân đối tượng sử dụng là sinh viên đang học và hoạt động tại trường. Giúp sinh viên có thể dễ dàng quản lý tài chính của bản thân. Đồng thời hỗ trợ nhà trường có thể dễ dàng kiểm soát và giải quyết các vấn đề về tài chính sinh viên

Bên cạnh đó bằng cách ứng dụng công nghệ IOT, một quy trình thanh toán tiện lợi và đơn giản được tạo ra. Sinh viên có thể sử dụng để thanh toán gửi xe một phần nào đó sẽ giải quyết vấn đề quá tải vào giờ cao điểm. Không chỉ thế nhờ IOT mà còn có thể tiết kiệm được chi phí nhân công như là người thu tiền. Phần chi phí đó có thể được sử dụng để mở rộng hệ thống nhằm giải quyết triệt để vấn đề. Ngoài ra giải pháp cho bãi gửi xe chỉ là 1 phần ứng dụng của hệ thống thanh toán qua IOT này. Hệ thống có thể áp dụng được cho nhiều loại hình như thanh toán các dịch vụ khác do Trường đại học Kinh tế Quốc dân cung cấp sử dụng nội bộ.

=> Rút ra mô hình của bài toán? (Vẽ sơ đồ)

A diagram of a computer system

Description automatically generated with medium confidence

Đề tài mong muốn định hình một hình ảnh mới về môi trường học tập, nơi công nghệ không chỉ là một phần của quá trình giảng dạy mà còn là một phần quan trọng trong cuộc sống hàng ngày của sinh viên.

## 1.3 Yêu cầu

Hệ thống được xây dựng lên phải đáp ứng được những yêu cầu cụ thể sau:

1. **Ví điện tử**: phải cung cấp được các chức năng cơ bản nhất của một ví điện tử : lưu trữ tiền tệ, thực hiện các giao dịch . Sau đó mở rộng thêm các tính năng bổ sung khác.
2. **Hệ thống IOT**: sử dụng các công nghệ IOT hiện có để áp dụng vào các hoạt động, dịch vụ tại trường đại học Kinh tế quốc dân nhằm cải thiện hiệu suất, nhân lực và tiết kiệm thời gian.
3. **Tích hợp hệ thống thanh toán:** sử dụng các công nghệ liên quan đến việc thanh toán để kết hợp ví điện tử với hệ thống IOT đã xây dựng nhằm hỗ trợ cho việc thanh toán và sử dụng dịch vụ tiện lợi và nhanh chóng nhất có thể.
4. **Phân quyền:** hệ thống phải đáp ứng được sự phân quyền rõ ràng giữa những đối tượng người dùng khác nhau đồng thời với mỗi đối tượng người dùng sẽ có những chức năng khác nhau được thể hiện rõ ràng.
5. **Xử lý realtime:** hệ thống xây dựng phải tiến hành dựa trên nguyên lý thời gian thực để giảm độ trễ trong quá trình xử lý và giao dịch.
6. **Giao diện người dùng:** giao diện hoàn thiện cần phải đáp ứng các tiêu chí đơn giản, dễ dùng để người dùng có thể theo dõi và sử dụng các chức năng mà hệ thống cung cấp.

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## 2.1 Phân tích bài toán

### 2.1.1 Phân tích yêu cầu bài toán

Dựa vào những yêu cầu đã đặt ra cho hệ thống ví điện tử đồng thời kết hợp với việc tìm hiểu, đã tiến hành xác định và bổ sung các chức năng cần thiết cho hệ thống. Và đây là mô hình chức năng mà hệ thống điện tử sẽ cung cấp:

A diagram of a company

Description automatically generated

Mô hình chức năng hệ thống

\* Mô tả chi tiết chức năng:

|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Mô tả chi tiết |
| 1 | Thực hiện giao dịch | Cho phép người dùng giao dịch với bên thứ 3 để nạp tiền vào ví và phải đảm bảo được tính bảo mật.  Người dùng có thể thực hiện giao dịch chuyển tiền qua lại trong nội bộ hệ thống được xây dựng. |
| 2 | Thống kê lịch sử | Người dùng có thể truy cập và theo dõi những giao dịch đã thực hiện, kiểm tra được lịch sử thanh toán. |
| 3 | Thanh toán dịch vụ | Người dùng có thể sử dụng các thiết bị được xây dựng bằng công nghệ IOT để thực hiện thanh toán một cách nhanh chóng, chính xác. |
| 4 | Nộp học phí | Nếu người dùng có nợ học phí và quyết định nộp học phí hệ thống sẽ tự động thực hiện nộp học phí cho người dùng. |
| 5 | Nhắn tin trò chuyện | Thực hiện giao tiếp giữa những người dùng thông qua tin nhắn và phải đảm bảo được thực hiện dựa trên thời gian thực (realtime). |
| 6 | Quản lý thông tin tài khoản | Hỗ trợ người dùng quản lý, cập nhập và thay đổi các thông tin cá nhân liên quan đến tài khoản cá nhân của người dùng. |
| 7 | Đăng nhập | Đăng nhập để sử dụng tài nguyên, dịch vụ được cung cấp theo quyền hạn. |

\* Đối với người quản trị hệ thống:

|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Chi tiết |
| 1 | Quản lý người dùng | Giúp quản trị viên có thể tùy chọn thêm, xóa người dùng mới.  Hỗ trợ quản trị viên trong quá trình theo dõi, quan sát và cập nhập thông tin, học phí của người dùng một cách chính xác và nhanh chóng. |
| 2 | Nhắn tin, trao đổi với người dùng | Người quản trị có thể thực hiện trò chuyện với người dùng thông qua tin nhắn được xây dựng dựa trên thời gian thực (realtime). |
| 3 | Quản lý dịch vụ tại trường | Cho phép người quản trị có thể thêm, sửa, xóa các dịch vụ tại trường.  Hỗ trợ cập nhập giá cả dịch vụ được cung cấp nội bộ. |
| 4 | Thống kê lịch sử | Theo dõi lịch sử giao dịch của toàn bộ người dùng trên hệ thống. |
| 5 | Quản lý thông tin tài khoản (admin) | Hỗ trợ người dùng quản lý, cập nhập và thay đổi các thông tin cá nhân liên quan đến tài khoản cá nhân (admin) của người dùng. |
| 5 | Nạp tiền | Hỗ trợ người dùng nạp tiền vào tài khoản khi người dùng lựa chọn phương thức tiền mặt. |
| 7 | Đăng nhập | Đăng nhập để sử dụng tài nguyên, dịch vụ được cung cấp theo quyền hạn |

### 2.1.2 Phân tích chức năng hệ thống

A diagram of a person's work flow

Description automatically generated**Biểu đồ use case tổng quát:**

Biểu đồ use case tổng quát

**Biểu đồ Use case phân rã :**

a) Sinh viên (user):

* A diagram of a person

  Description automatically generatedUse case Thực hiện giao dịch:
* User case Thống kê lịch sử :

A diagram with a green oval and black text

Description automatically generated

* Use case Quản lý thông tin tài khoản :

A diagram of a diagram

Description automatically generated

b) Người quản lý (Admin):

* A diagram of a diagram

  Description automatically generatedUse case Quản lý thông tin tài khoản:
* Use case Thống kê lịch sử:

A diagram of a person's body

Description automatically generated with medium confidenceA black text on a white background

Description automatically generated

* Use case Quản lý người dùng:

A diagram of a diagram

Description automatically generated

* Use case Quản lý dịch vụ:

A diagram of a diagram

Description automatically generated

**Đặc tả use case (activity diagram):**

* Use case Đăng nhập:

A diagram of a diagram

Description automatically generated with medium confidence

* Use case Giao dịch:

- Chuyển tiền

A diagram of a diagram

Description automatically generated with medium confidence

- Nạp tiền:

A diagram of a diagram

Description automatically generated

* Use case Thanh toán dịch vụ:

A diagram of a computer

Description automatically generated

* Use case nhắn tin:

A screenshot of a diagram

Description automatically generated

* Use case Quản lý thông tin tài khoản :
* Cập nhập thông tin

A diagram of a workflow

Description automatically generated

* Use case Nộp học phí:

A screenshot of a diagram

Description automatically generated

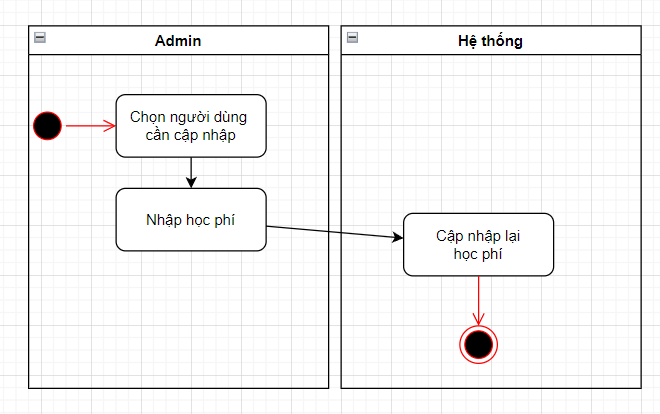
* Use case quản lý người dùng:

Tạo người dùng mới

A diagram of a diagram

Description automatically generated with medium confidence

Cập nhập học phí



Xóa người dùng

A diagram of a diagram

Description automatically generated

* Use case quản lý dịch vụ:
* Thêm dịch vụ mới

A diagram of a diagram

Description automatically generated

* Cập nhập dịch vụ

A diagram of a diagram

Description automatically generated with medium confidence

* Xóa dịch vụ

A diagram of a diagram

Description automatically generated

### 2.1.3 Phân tích cấu trúc

a) Mô hình tổng quát hệ thống

A diagram of a computer network

Description automatically generated

b) Biểu đồ lớp tổng quát

A diagram of a computer

Description automatically generated

### 2.1.4 Phân tích hành vi

a) Đăng nhập

A diagram of a diagram

Description automatically generated

b) Chuyển tiền

A diagram of a diagram

Description automatically generated

c) Nạp tiền

A diagram of a diagram

Description automatically generated

d) Cập nhập thông tin ( dịch vụ, thẻ thanh toán)

A diagram of a diagram

Description automatically generated

e) Thêm người dùng mới (dịch vụ, thẻ thanh toán)

A diagram of a diagram

Description automatically generated

f) Xóa người dùng (dịch vụ, thẻ thanh toán)

A diagram of a diagram

Description automatically generated

g) Thanh toán dịch vụ

A diagram of a diagram

Description automatically generated

h) Nhắn tin

A diagram of a diagram

Description automatically generated

## 2.2 Thiết kế

### 2.2.1 Thiết kế cấu trúc

A diagram of a computer

Description automatically generatedBiểu đồ gói:

Biểu đồ triển khai:

A diagram of a computer network

Description automatically generated

### 2.2.2 Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu được xây dựng và lưu trữ trên Cloud Firestore một dịch vụ của Firebase do Google cung cấp. Cơ sở dữ liệu sẽ là noSQL. Thay vì xây dựng các Table như SQL thì ta sẽ xây dựng các Collection. Mỗi Collection chứa các Document(bản ghi) lưu trữ thông tin.

**1. Collection ‘Users’**

Tại đây sẽ lưu trữ những thông tin cá nhân của mỗi người dùng. Với người dùng quyền hạn là quản lý role sẽ là “admin”, người dùng sử dụng hệ thống sẽ là “user”.

Password mặc định là “1” khi tạo người dùng mới

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| objectId | String | Mã định danh của bản ghi |
| username | String | Tên người dùng |
| userId | String | Mã người dùng  (tài khoản đăng nhập) |
| password | String | Mật khẩu người dùng |
| Role | String | Vai trò của người dùng |
| balance | Number | Số dư của người dùng |
| card\_uid | String | Mã số thẻ người dùng |
| tuition | Number | Học phí |
| Active | Bool | Tình trạng của tài khoản |

**2. Collection ‘Services’**

Tại đây lưu trữ thông tin của các dịch vụ mà nhà trường cung cấp để sử dụng

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| objectId | String | Mã định danh của bản ghi |
| serviceName | String | Tên dịch vụ cung cấp |
| Amount | Number | Giá của dịch vụ tương ứng |

**3. Collection ‘Transactions’**

Tại đây lưu trữ thông tin các giao dịch trên hệ thống. Đối với fields “method” sẽ là “none” khi thực hiện giao dịch chuyển tiền trong nội bộ. Và sẽ là các method tương ứng khi sử dụng các bên khác nhau để nạp tiền vào ví.

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| objectId | String | Mã định danh của bản ghi |
| createId | String | Mã người dùng tạo giao dịch |
| receiveId | String | Mã người dùng nhận được giao dịch |
| receiveName | String | Tên người nhận |
| type | String | Loại giao dịch |
| method | String | Phương thức thực hiện giao dịch |
| Amount | Number | Số tiền giao dịch |
| createTime | Timestamp | Thời gian thực hiện giao dịch |

**4. Collection ‘Activities’**

Tại đây sẽ lưu trữ các bản ghi chứa thông tin các hoạt động thanh toán của người dùng.

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| objectId | String | Mã định danh của bản ghi |
| userId | String | Mã người dùng thanh toán |
| type | String | Tên dịch vụ thanh toán |
| amount | Number | Số tiền thanh toán |
| createId | Timestamp | Thời gian thực hiện thanh toán |

**5. Collections ‘Messages’**

Tại đây sẽ lưu trữ các tin nhắn giữa 2 người dùng đã kết nối với nhau.

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| objectId | String | Mã định danh của bản ghi |
| messages | Array | Mảng chứa các thông tin tin nhắn giữa 2 người dùng |
| mesages[i] | Map | Chứa các thông tin về tin nhắn được gửi |
| sendId | ObjectId | Mã định danh của bản ghi chứa thông tin người dùng |
| message | String | Nội dung tin nhắn được gửi |
| messageId | String | Mã định danh tin nhắn |
| createTime | Timestamp | Thời gian gửi tin nhắn |

**6. Collections ‘userChats’**

Tại đây sẽ lưu trữ thông tin của người dùng và các người dùng mà họ đã thực hiện chat. Nếu chưa thì sẽ không có thông tin.

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| objectId | String | Mã định danh của bản ghi |
| userChat | Map | Chứa thông tin của những người dùng đã thực hiện nhắn tin |
| userChat.date | Timestamp | Thời gian gửi tin nhắn gần nhất |
| userChat.lastMessage | String | Nội dung tin nhắn gần nhất |
| userChat.userInfo | Map | Chứa thông tin người dùng |
| .userInfo.userId | String | Mã người dùng |
| .userInfo.username | String | Tên người dùng đã nhắn tin |

### 2.2.3 Thiết kế giao diện

UIcode

# **CHƯƠNG 3: CÁC CÔNG NGHỆ, NỀN TẢNG ÁP DỤNG VÀO XÂY DỰNG HỆ THỐNG**

## Nội dung chương này liên quan gì đến đề tài?

## 3.1 Kiến trúc Web Client – Server

### 3.1.1 Tổng quan mô hình Client-Server

- Mô hình client server **mô hình mạng máy tính gồm có 2 thành phần chính là client và server** (tức là **máy khách – máy chủ**). Client sẽ là bên yêu cầu dịch vụ cài đặt cũng như lưu trữ tài nguyên từ phía Server.

Khi máy client gửi yêu cầu dữ liệu đến server thông qua Internet, server sẽ chấp nhận quy trình được yêu cầu. Sau đó gửi các gói dữ liệu được yêu cầu trở lại client. Client không chia sẻ bất kỳ tài nguyên nào của họ.

A diagram of a network

Description automatically generated

**HTTP(S) :** res/req

**HTTP(S) :** res/req

**HTTP(S) :** res/req

**HTTP(S) :** res/req

*Hình 1. Mô hình web Client-Server*

### 3.1.2 Nguyên tắc hoạt động của mô hình Client-Server

a) Client

- Khi nói đến Client (khách hàng), thì nó có nghĩa là một người hay một tổ chức sử dụng một dịch vụ cụ thể nào đó. Trong thế giới kỹ thuật số cũng tương tự như vậy. Client là một máy tính (Host), tức là có khả năng nhận thông tin hoặc sử dụng một dịch vụ cụ thể từ các nhà cung cấp dịch vụ (Server).

b) Server

- Tương tự như vậy, khi nói đến **Server**thì nó có nghĩa là một máy chủ hay một phương tiện phục vụ các dịch vụ nào đó. Trong lĩnh vực công nghệ thì Server là một máy tính từ xa. Nó cung cấp các thông tin (dữ liệu) hoặc quyền truy cập vào các dịch vụ cụ thể.

### 3.1.3 Ưu, nhược điểm của mô hình Client-Server

a) Ưu điểm

1. Tính tập trung

Ưu điểm chính của mô hình mạng khách chủ là khả năng kiểm soát tập trung (Centralization) được tích hợp sẵn. Với mô hình này, tất cả thông tin cần thiết đều được đặt ở một vị trí duy nhất. Việc này rất hữu ích cho những quản trị viên mạng. Vì họ có được toàn quyền quản lý và điều hành.

Bằng tính năng này, mọi sự cố trong mạng đều có thể được giải quyết ở một nơi duy nhất. Và do đó, việc cập nhật tài nguyên, dữ liệu cũng trở nên dễ dàng hơn.

2. Bảo mật

Trong mạng Client Server, dữ liệu được bảo vệ tốt do kiến trúc tập trung của mạng. Nó có thể được áp dụng các biện pháp kiểm soát truy cập, sao cho chỉ những người dùng được ủy quyền mới có thể truy cập.

Một trong những cách để làm vậy là áp đặt thông tin đăng nhập, chẳng hạn như**Username** hay **Password**. Hơn nữa, nếu dữ liệu bị mất, các file có thể được khôi phục dễ dàng từ chỉ một bản sao lưu duy nhất.

3. Khả năng mở rộng

**Mô hình mạng Client server** có khả năng mở rộng tốt. Bất cứ khi nào người dùng cần, họ có thể tăng số lượng tài nguyên, chẳng hạn như số Client hay Server. Do đó có thể dễ dàng tăng kích thước của Server mà không bị gián đoạn nhiều.

4. Khả năng truy cập

Không có sự phân biệt giữa các vị trí hay nền tảng khác nhau, mọi client đều có thể đăng nhập vào hệ thống. Bằng cách này, tất cả nhân viên đều có thể truy cập thông tin của công ty của họ, không cần phải dùng một [terminal mode](https://en.wikipedia.org/wiki/Terminal_mode) hay một bộ xử lý nào.

5. Tách biệt

Tách biệt trách nhiệm giữa client và server, cho phép phát triển độc lập và quản lý dễ dàng.

## 3.2 HTML, CSS

### 3.2.1 Html

[**HTML**](https://topdev.vn/viec-lam-it/html-kt75) là viết tắt của cụm từ **Hypertext Markup Language**(tạm dịch là Ngôn ngữ đánh dấu [siêu văn bản](https://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n)). **HTML** được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes… và **HTML** không phải là ngôn ngữ lập trình.

Khi bạn gõ ra 1 tên miền, trình duyệt mà bạn đang sử dụng (chẳng hạn như Chrome) sẽ kết nối tới 1 máy chủ web, bằng cách dùng 1 địa chỉ IP, vốn được thấy bằng cách phân giải tên miền đó (DNS). Máy chủ web chính là 1 máy tính được kết nối tới internet và nhận các yêu cầu tới trang web từ trình duyệt của bạn. Máy chủ sau đó sẽ gửi trả thông tin về trình duyệt của bạn, là 1 tài liệu HTML, để hiển thị trang web!

Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm. Khi một tập tin HTML được hình thành, việc xử lý nó sẽ do trình duyệt web đảm nhận. Trình duyệt sẽ đóng vai trò đọc hiểu nội dung HTML từ các thẻ bên trong và sẽ chuyển sang dạng văn bản đã được đánh dấu để đọc, nghe hoặc hiểu (do các bot máy tính hiểu).

**Ưu điểm:**

* Được sử dụng rộng rãi, có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng lớn.
* Học đơn giản và dễ hiểu.
* Mã nguồn mở và hoàn toàn miễn phí.
* Markup gọn gàng và đồng nhất.
* Tiêu chuẩn thế giới được vận hành bởi World Wide Web Consortium (W3C).
* Dễ dàng tích hợp với các ngôn ngữ backend như PHP, Python…

**Khuyết điểm:**

* Được dùng chủ yếu cho web tĩnh. Đối với các tính năng động như update hay realtime thời gian thực, bạn cần sử dụng JavaScript hoặc ngôn ngữ backend bên thứ 3 như PHP.
* Một số trình duyệt chậm hỗ trợ tính năng mới.

### 2.2.2 CSS

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

Thông qua việc sử dụng CSS sẽ đem lại một số lợi ích cụ thể:

1. Giải quyết một vấn đề lớn

Trước khi có CSS, các thẻ như phông chữ, màu sắc, kiểu nền, các sắp xếp phần tử, đường viền và kích thước phải được lặp lại trên mọi trang web. Đây là một quá trình rất dài tốn thời gian và công sức. Ví dụ: Nếu bạn đang phát triển một trang web lớn nơi phông chữ và thông tin màu được thêm vào mỗi trang, nó sẽ trở thành một quá trình dài và tốn kém. CSS đã được tạo ra để giải quyết vấn đề này. Đó là một khuyến cáo của W3C.

Nhờ CSS mà source code của trang Web sẽ được tổ chức gọn gàng hơn, trật tự hơn. Nội dung trang web sẽ được tách bạch hơn trong việc định dạng hiển thị. Từ đó, quá trình cập nhập nội dung sẽ dễ dàng hơn và có thể hạn chế tối thiểu làm rối cho mã HTML.

2. Tiết kiệm rất nhiều thời gian

Định nghĩa kiểu CSS được lưu trong các tệp CSS bên ngoài vì vậy có thể thay đổi toàn bộ trang web bằng cách thay đổi chỉ một tệp. Sử dụng CSS sẽ giúp bạn không cần thực hiện lặp lại các mô tả cho từng thành phần. Từ đó, bạn có thể tiết kiệm được tối đa thời gian làm việc với nó, làm code ngắn lại giúp kiểm soát dễ dàng hơn các lỗi không đáng có.

CSS tạo ra nhiều style khác nhau nên có thể được áp dụng với nhiều trang web, từ đó giảm tránh việc lặp lại các định dạng của các trang web giống nhau.

3. Cung cấp thêm các thuộc tính

CSS cung cấp các thuộc tính chi tiết hơn HTML để định nghĩa giao diện của trang web. CSS giúp người dùng nhiều styles trên một trang web HTML nên khả năng điều chỉnh trang của bạn trở nên vô hạn.

## 3.3 Javascript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

Trước đây, các trang web có dạng tĩnh, tương tự như các trang trong một cuốn sách. Một trang tĩnh chủ yếu hiển thị thông tin theo một bố cục cố định và không làm được mọi thứ mà chúng ta mong đợi như ở một trang web hiện đại. JavaScript dần được biết đến như một công nghệ phía trình duyệt để làm cho các ứng dụng web linh hoạt hơn. Sử dụng JavaScript, các trình duyệt có thể phản hồi tương tác của người dùng và thay đổi bố cục của nội dung trên trang web.

Khi ngôn ngữ này phát triển hoàn thiện, các nhà phát triển JavaScript đã thiết lập các thư viện, khung và cách thức lập trình cũng như bắt đầu sử dụng ngôn ngữ này bên ngoài trình duyệt web. Ngày nay, bạn có thể sử dụng JavaScript để thực hiện hoạt động phát triển cả ở phía máy khách và máy chủ.

### 3.3.1Client - máy khách:

JavaScript phía máy khách đề cập đến cách thức JavaScript hoạt động trong trình duyệt của bạn. Trong trường hợp này, công cụ JavaScript nằm trong mã trình duyệt. Tất cả các trình duyệt web phổ biến đều có các công cụ JavaScript tích hợp riêng.

Các nhà phát triển ứng dụng web viết mã JavaScript với các hàm khác nhau được liên kết với nhiều loại sự kiện, chẳng hạn như nhấp chuột hoặc khi di chuột đến. Các hàm này thực hiện các thay đổi đối với HTML và CSS.

Sau đây là tổng quan về cách thức hoạt động của JavaScript phía máy khách:

1.   Trình duyệt tải một trang web khi bạn truy cập trang đó.

2.   Trong khi tải, trình duyệt chuyển đổi trang và tất cả các phần tử của trang, chẳng hạn như các nút, nhãn và hộp thả xuống, thành một cấu trúc dữ liệu được gọi là Mô hình đối tượng tài liệu (DOM).

3.   Công cụ JavaScript của trình duyệt chuyển đổi mã JavaScript thành mã bytecode. Mã này đóng vai trò là trung gian giữa cú pháp JavaScript và máy.

4.   Các sự kiện khác nhau, chẳng hạn như nhấp vào nút, sẽ kích hoạt việc thực thi khối mã JavaScript được liên kết. Sau đó, công cụ này sẽ diễn giải bytecode và thực hiện các thay đổi đối với DOM.

5.   Trình duyệt hiển thị DOM mới.

### 3.3.2 Server – máy chủ:

JavaScript phía máy chủ đề cập đến việc sử dụng ngôn ngữ viết mã trong logic máy chủ back-end. Trong trường hợp này, công cụ JavaScript nằm trực tiếp trên máy chủ. Hàm JavaScript phía máy chủ có thể truy cập cơ sở dữ liệu, thực hiện các phép toán logic khác nhau và phản hồi nhiều loại sự kiện do hệ điều hành của máy chủ kích hoạt. Ưu điểm chính của viết tập lệnh phía máy chủ là bạn có khả năng tùy chỉnh linh hoạt phản hồi của trang web dựa trên yêu cầu và quyền truy cập của mình cũng như theo yêu cầu thông tin từ trang web.

## 3.4 Nodejs và Express.js

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

 Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.

A screenshot of a computer screen

Description automatically generated

**Expressjs** là một framework được xây dựng trên nền tảng của **Nodejs**. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. **Expressjs** hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

A diagram of a computer program

Description automatically generated

*Tổng hợp một số chức năng chính của Expressjs như sau:*

* Thiết lập các lớp trung gian để trả về các HTTP request.
* Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số.

## 3.5 Rest API và Json

### 3.5.1 Restful API

Restful API là một trong những tiêu chuẩn được sử dụng để thiết kế API cho các ứng dụng web, để tiện hơn cho việc quản lý các resource. Rest API thường chú trọng vào các tài nguyên của hệ thống bao gồm: ảnh, văn bản, âm thanh, video hoặc các dữ liệu di động,... Nó thường bao gồm các trạng thái tài nguyên đã được định dạng sẵn và được truyền tải thông qua HTTP.[giá cước vận chuyển hàng không quốc tế](https://kinkinlogistics.com/tin-tuc/gia-cuoc-van-chuyen-hang-khong-quoc-te-n335)

A diagram of a computer api

Description automatically generated

Một chức năng quan trọng nhất của REST là: quy định các cách sử dụng HTTP method chẳng hạn như: Post, Get, Delete, Put,... và cách có thể định dạng các URL cho ứng dụng web để có thể quản lý được các resource. Restful không quy định logic code ứng dụng và không được giới hạn bởi những ngôn ngữ lập trình ứng dụng. Bất kỳ framework hoặc ngôn ngữ nào cũng có thể được dùng để thiết kế được một Restful API.

## 3.6 Firebase – Firebase cloud firestore

Những năm 2011, trước khi Firebase ra đời. Tiền thân của nền tảng Firebase chính là Envolve. Envolve đến với người dùng dưới mô hình startup, họ cung cấp cho người dùng những API để dễ dàng tích hợp tính năng chat vào trang web. Thế nhưng, người dùng lại sử dụng Envolve để truyền dữ liệu ứng dụng chứ không đơn thuần là nhắn tin trò chuyện. Chính điều này đã khiến các nhà phát triển Envolve quyết định tách riêng chat system và the real-time architecture.

A screenshot of a computer

Description automatically generatedFirebase là nền tảng phát triển ứng dụng đa năng của di động và website.

Và năm 2012, Firebase ra đời dưới dạng Backend-as-a-Service với chức năng thời gian thực. Sau khi được Google mua lại vào năm 2014 thì Firebase nhanh chóng phát triển thành nền tảng phát triển ứng dụng đa năng của di động và website như ngày nay. Nền tảng này là sự kết hợp giữa cloud với hệ thống máy chủ của Google để tập trung chính cho 2 đối tượng là:

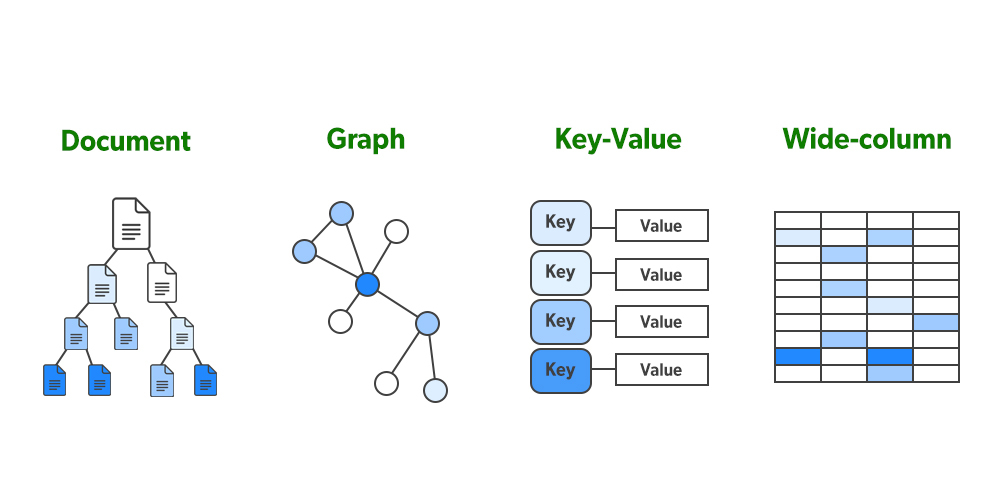
* **Develop & test your app**: phát triển và thử nghiệm các ứng dụng được thiết kế.
* **Grow & engage your audience**: phân tích dữ liệu và tối ưu hóa trải nghiệm với người dùng.

Firebase cung cấp cho chúng ta những API đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng database, bởi vậy giờ đây chúng ta chỉ cần gọi API và phần server đã có Firebase lo.

Firebase là một nền tảng đa năng, nó cung cấp rất nhiều dịch vụ khác nhau cho người dùng như là : Realtime database, authentication, firebase cloud messaging. Thế nhưng để đáp ứng đề tài này hệ thống sẽ sử dụng Firebase Cloud Firestore – có thể coi như là một bản cải tiến của realtime database.

### Firebase Cloud Firestore

Cloud Firestore là một Database linh hoạt và dễ mở rộng cho mobile, web và server được phát triển từ Firebase and Google Cloud Platform. Cloud Firestore giúp cho việc đồng bộ dữ liệu giữa các ứng dụng phía client một các nhanh chóng (Realtime) và hộ trợ lưu offline data trong ứng dụng của bạn.



Cloud Firestore là một cloud-hosted, NoSQL database mà các ứng dụng phía client có thể trực tiếp truy cập thông qua native SDKs. Nó lưu dữ liệu theo mô hình dữ liệu NoSQL. Dữ liệu được lưu trữ trong các file tài liệu chứa các trường được ánh xạ vào các giá trị. Các file tài liệu này được lưu trữ trong các tập hợp chúng có thể sử dụng nó để tổ chức dữ liệu và truy vấn dữ liệu. Cloud Firestore hỗ trợ rất nhiều kiểu dữ liệu từ đơn giản như String, Integer hay những kiểu dữ liệu phức tạp như các nested object.

## 3.7 Công nghệ RFID

**RFID** là viết tắt của từ Radio Frequency Identification là công nghệ nhận dạng đối tượng bằng sóng vô tuyến. Khi đó cả hai thiết bị hoạt động thu phát sóng trong cùng tần số và tần số đó thường được sử dụng trong RFID là 125Khz hoặc 900Mhz .



Một thiết bị RFID được cấu tạo bởi 2 thành phần chính là thiết bị đọc và thiết bị phát mã có gắn chip. Trong đó thiết bị đọc được gắn antenna thu phát sóng điện từ, còn thiết bị phát mã RFID được gắn với vật cần nhận dạng, mỗi thiết bị RFID có chứa một mã số nhất định sao cho không trùng lặp với nhau.

**Thiết bị RFID** reader phát ra sóng điện từ ở một tần số cụ thể nào đó và thiết bị phát mã RFID tag trong vùng hoạt động sẽ cảm nhận được sóng được điện từ này và thu nhận năng lượng từ đó phát lại cho **thiết bị RFID** biết mã số của mình. Ngay lúc đó RFID reader biết được tag nào đang hoạt động trong vùng sóng điện từ.

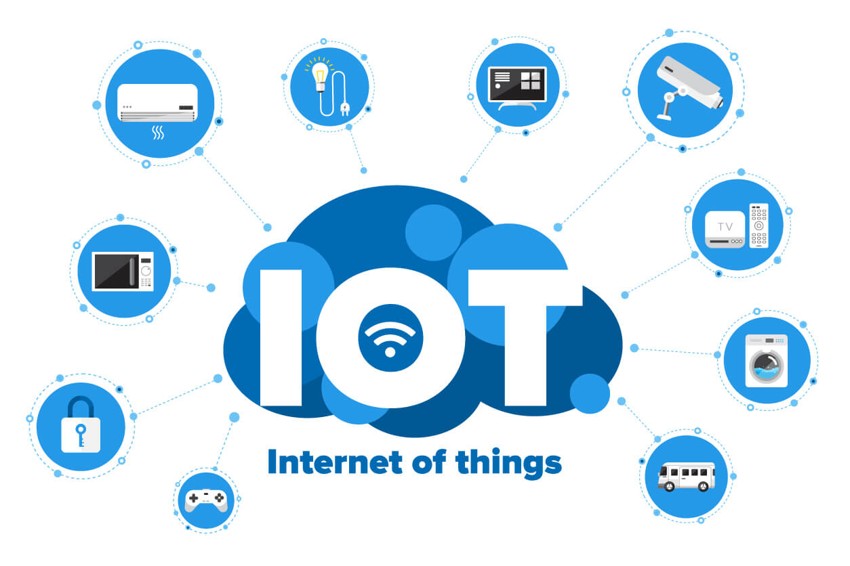
Từ đó một số cách sử dụng phổ biến cho công nghệ RFID ra đời ví dụ như:

* Theo dõi vật nuôi và vật nuôi
* Theo dõi xe
* Dịch vụ khách hàng và kiểm soát tổn thất
* Cải thiện khả năng hiển thị và phân phối trong  chuỗi cung ứng
* Kiểm soát truy cập trong các tình huống an ninh
* Thanh toán bằng thẻ tín dụng chạm và chuyển

Chính vì thế công nghệ RFID đã được chọn để xây dựng hệ thống này

## 3.8 Hệ thống điều khiển IOT (Module wifi Esp32)

Trong thời đại kỹ thuật số ngày càng phát triển, Internet of Things (IoT) đã trở thành một trong những xu hướng quan trọng của ngành công nghiệp và cuộc sống hàng ngày. IoT là khái niệm mà các thiết bị và cảm biến có khả năng kết nối và trao đổi dữ liệu thông qua internet, giúp tạo ra một mạng lưới các thiết bị thông minh và tự động hóa.

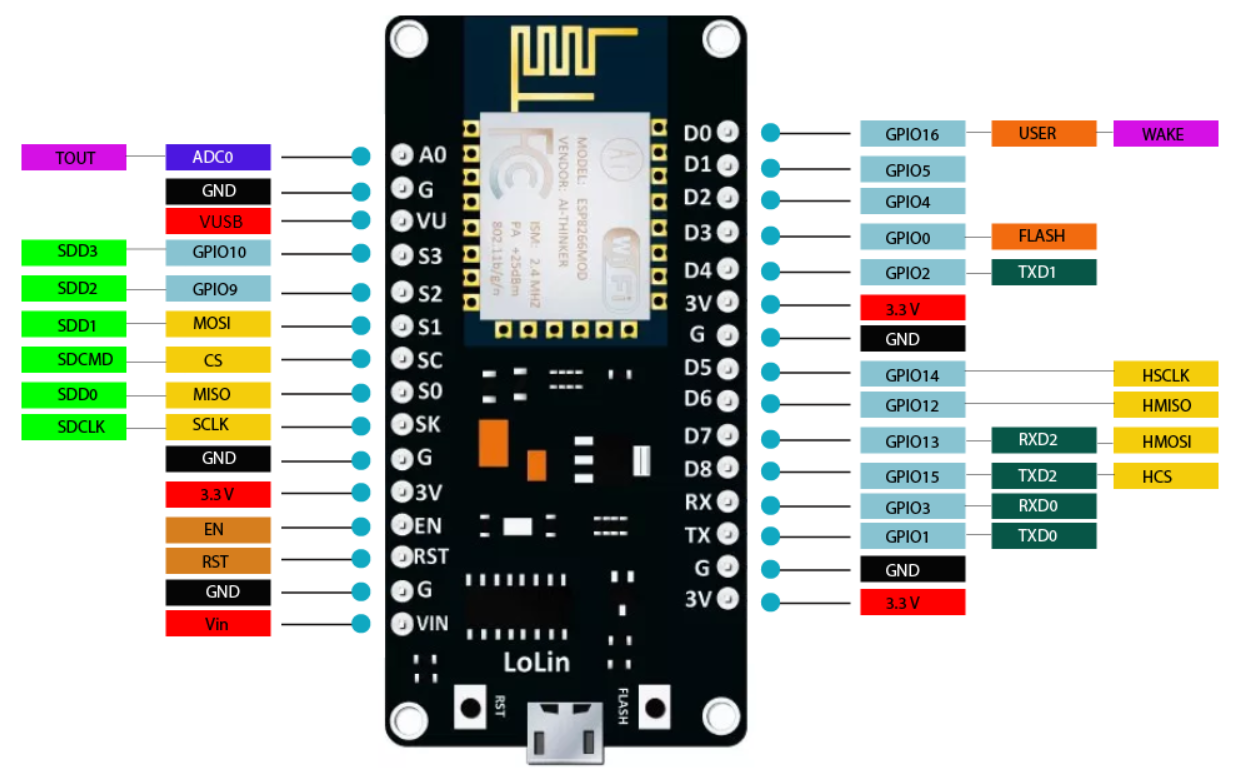


IoT và Sự Phát Triển Của Công Nghệ

Một phần quan trọng của hệ thống IoT là khả năng điều khiển và giám sát từ xa. Điều này bao gồm việc sử dụng các thiết bị và module có khả năng kết nối Wi-Fi để tạo kết nối mạng với các hệ thống điều khiển trung tâm hoặc điều khiển trực tiếp từ xa thông qua ứng dụng di động hoặc máy tính.

Một trong những module Wi-Fi phổ biến được sử dụng trong các dự án IoT là ESP32. ESP32 là một thiết bị có khả năng kết nối Wi-Fi và Bluetooth, được sản xuất bởi Tập đoàn Espressif. Module này được ưa chuộng bởi sự linh hoạt và tích hợp nhiều tính năng trên một chip duy nhất.

ESP32 có một bộ xử lý mạnh mẽ, nhiều cổng GPIO (General Purpose Input/Output), khả năng kết nối Wi-Fi và Bluetooth, cũng như khả năng tiết kiệm năng lượng. Điều này làm cho nó trở thành một lựa chọn tốt cho các ứng dụng IoT đa dạng từ những dự án nhỏ đơn giản đến các hệ thống phức tạp.



Module Wi-Fi ESP32

*Thiết Kế Hệ Thống Điều Khiển IoT Với ESP32 :*

Thiết kế một hệ thống điều khiển IoT với module Wi-Fi ESP32 đòi hỏi một số bước quan trọng:

Lựa Chọn Các Cảm Biến và Thiết Bị: Đầu tiên, bạn cần xác định những cảm biến và thiết bị mà bạn muốn kết nối vào hệ thống IoT. Các ví dụ có thể là cảm biến nhiệt độ, độ ẩm, cảm biến chuyển động, đèn LED, máy bơm, v.v.

Kết Nối ESP32 với Cảm Biến: Bạn sẽ cần kết nối ESP32 với các cảm biến và thiết bị bằng cách sử dụng các cổng GPIO và giao thức tương ứng. Điều này có thể đòi hỏi việc lập trình để giao tiếp với cảm biến.

Thiết Kế Giao Thức Kết Nối: Bạn cần xác định giao thức kết nối cho hệ thống. MQTT (Message Queuing Telemetry Transport) là một giao thức phổ biến cho IoT, cho phép truyền tải dữ liệu giữa các thiết bị và hệ thống điều khiển.

Lập Trình Ứng Dụng ESP32: Bạn cần lập trình ứng dụng trên ESP32 để gửi và nhận dữ liệu từ các cảm biến, sau đó truyền dữ liệu đó qua giao thức đã chọn đến hệ thống điều khiển.

Hệ Thống Điều Khiển Trung Tâm: Hệ thống điều khiển trung tâm có thể là máy tính hoặc máy chủ chạy ứng dụng để theo dõi và điều khiển các thiết bị IoT. Hệ thống này sẽ xử lý dữ liệu và cung cấp tương tác người dùng.

Ứng Dụng Di Động hoặc Giao Diện Web: Để kiểm soát hệ thống từ xa, bạn cần phát triển một ứng dụng di động hoặc giao diện web. Điều này cho phép người dùng theo dõi và điều khiển các thiết bị IoT từ bất kỳ đâu.

Bảo Mật: An ninh là một phần quan trọng của hệ thống IoT. Bạn cần đảm bảo rằng kết nối giữa ESP32 và hệ thống điều khiển là an toàn và bảo mật.

Ứng Dụng Của Hệ Thống Điều Khiển IoT

Hệ thống điều khiển IoT với module Wi-Fi ESP32 có nhiều ứng dụng tiềm năng, bao gồm:

- Nhà Thông Minh: IoT có thể được sử dụng để kiểm soát ánh sáng, nhiệt độ, an ninh, và các thiết bị trong nhà thông qua điện thoại di động.

- Quản Lý Năng Lượng: Hệ thống có thể theo dõi và kiểm soát sử dụng năng lượng, giúp tiết kiệm điện và giảm đối tượng ảnh hưởng đến môi trường.

- Sản Xuất Tự Động: Các thiết bị trong quy trình sản xuất có thể được theo dõi và điều khiển từ xa, giúp tối ưu hóa quy trình sản xuất.

- Quản Lý Nước: Hệ thống IoT có thể theo dõi mức nước và kiểm soát bơm nước trong hệ thống quản lý tài nguyên nước.

Hệ thống điều khiển IoT với module Wi-Fi ESP32 đang mở ra nhiều cơ hội trong việc kiểm soát và theo dõi các thiết bị và quy trình từ xa. Sự kết hợp giữa khả năng của ESP32 và khả năng lập trình ứng dụng IoT có thể giúp tạo ra các hệ thống thông minh và tiết kiệm năng lượng, mang lại lợi ích cho nhiều lĩnh vực khác nhau và cụ thể trong hệ thống này là lĩnh vực thanh toán.

## 3.9 Cổng thanh toán (Payment Gateway) : VNPay

### 3.9.1 Cổng thanh toán

Thanh toán trực tuyến đã trở thành một phần trong cuộc sống hàng ngày của chúng ta trong thời gian ngắn. Chúng ta có thể giao dịch trực tuyến không chỉ thông qua thẻ ghi nợ (debit) hoặc thẻ tín dụng (credit card) mà còn thông qua nhiều chế độ khác như UPI, netbanking và ví điện tử.

Thanh toán trực tuyến là một tính năng cơ bản mà mọi nền tảng thương mại điện tử trên thế giới cung cấp. Và họ có thể cung cấp cơ sở này bằng cách tích hợp với một cổng thanh toán.

Cổng thanh toán trực tuyến (PG) là một đường hầm kết nối tài khoản ngân hàng của bạn với nền tảng nơi bạn cần chuyển tiền. PG là phần mềm cho phép bạn thực hiện giao dịch trực tuyến thông qua các chế độ thanh toán khác nhau như net banking, credit card, debit card, [UPI](https://en.wikipedia.org/wiki/Unified_Payments_Interface)hoặc nhiều ví trực tuyến có sẵn trong những ngày này. 

Một PG đóng vai trò của bên thứ ba chuyển tiền của bạn một cách an toàn từ tài khoản ngân hàng sang cổng thanh toán của người bán.

Cách thức hoạt động:

**Bước 1**: Một khách hàng đặt đơn hàng của mình và sau đó nhấn nút Gửi hoặc Thanh toán hoặc nút tương đương trên trang web

**Bước 2**: Một khi điều này xảy ra, trang web hoặc nền tảng thương mại điện tử sẽ đưa khách hàng đến một cổng thanh toán nơi họ nhập tất cả các thông tin liên quan về ngân hàng hoặc thẻ họ đang sử dụng để thanh toán. PG sau đó đưa người dùng trực tiếp đến trang của ngân hàng phát hành hoặc trang bảo mật 3D, yêu cầu giao dịch được ủy quyền.

**Bước 3**: Sau khi cổng thanh toán nhận được sự chấp thuận cho giao dịch, ngân hàng sẽ kiểm tra xem khách hàng có đủ số dư trong tài khoản để thực hiện giao dịch này thành công hay không

**Bước 4**: Cổng thanh toán gửi tin nhắn cho người bán phù hợp. Nếu phản hồi từ ngân hàng là "No/Không", thì người bán sau đó sẽ gửi một thông báo lỗi cho khách hàng, thông báo cho họ về vấn đề với thẻ hoặc tài khoản ngân hàng. Nếu phản hồi là "Yes/Có" từ cổng thông tin ngân hàng, thì người bán tìm kiếm giao dịch từ ngân hàng.

**Bước 5**: Ngân hàng thanh toán tiền bằng cổng thanh toán, từ đó thanh toán tiền với người bán. Khi quá trình này được hoàn thành, khách hàng sẽ nhận được thông báo xác nhận đơn hàng được đặt.



### 2.8.2 VNPay

**Cổng thanh toán VNPAY** ra mắt năm 2011 do Công ty Cổ phần Giải pháp Thanh toán Việt Nam (VNPAY) phát triển. Khi doanh nghiệp kinh doanh kết nối với cổng thanh toán VNPAY, khách hàng có thể dễ dàng thực hiện thanh toán trực tuyến qua internet banking bằng việc quét mã QR code.

A computer screen shot of a diagram

Description automatically generatedHiện nay, cổng đã kết nối tới hơn 40 ngân hàng nội địa và quốc tế khác nhau. Chính bởi sự kết nối đa dạng này, VNPAY hiện đang là đối tác của rất nhiều đơn vị: Vietnam Airlines, VinMart, tập đoàn Red Sun,..

Luồng xử lý cổng thanh toán VNPAY

Dễ dàng tích hợp: Với quy trình được tối ưu giảm bỏ các bước dư thừa từ đó cải thiện thời gian tích hợp đưa vào vận hành

Giao diện sau khi tích hợp thân thiện. Khách hàng hoàn toàn có thể thực hiện các thao tác mua sắm và thanh toán với các website, sàn thương mại điện tử kết nối với VNPAY. Đặc biệt, khách hàng có thể tiện lợi sử dụng cả trên bản web và mobile

Chính sách mức phí hấp dẫn. Hiện nay, VNPAY đang triển khai chính sách miễn phí tích hợp và duy trì hàng tháng cho doanh nghiệp (tùy thời điểm).

Hỗ trợ các công cụ theo dõi vận hàng. Chủ doanh nghiệp có thể tiện lợi theo dõi tình hình mua sắm tổng kết đối soát dòng tiền theo thời gian mong muốn.

Bảo mật. Với hỗ trợ công nghệ, VNPAY tự tin là đơn vị bảo mật an toàn cho doanh nghiệp và cả khách hàng.

# **CHƯƠNG 4: XÂY DỰNG, CÀI ĐẶT VÀ THỬ NGHIỆM**

## 4.1 Xây dựng, cài đặt hệ thống

Để tránh xung đột, hệ thống sẽ hoạt động tốt nhất theo các môi trường sau

|  |  |
| --- | --- |
| Yêu cầu | Môi trường |
| Hệ điều hành | Windows |
| Nodejs | **v18.17.1** |
| Npm | **v10.2.0** |

Cài đặt và triển khai hệ thống dựa trên mã nguồn sau:

<https://github.com/tanngo14780/cdtt-e-wallet>

|  |  |
| --- | --- |
| Folder | Mô tả |
| Admin | Giao diện Admin |
| Client | Giao diện người dùng |
| Server | API hệ thống |
| IOT | Cài đặt module Esp32 |

Ngoài ra để có thể nhận diện cổng Comp kết nối đến module để upload cần cài đặt thêm driver cho máy : ‘CP210x USB đến UART VCP’. Kết nối chân đầu đọc RFID như hình:

A computer screen with text and numbers

Description automatically generated

Đối với ssid, password có thể thay bằng tên và mật khẩu wifi tại nơi kết nối. Ngoài ra khi thay đổi wifi phải lấy lại địa chỉ ip của thiết bị. Mở command line và chạy lệnh “ipconfig”.

A computer screen with numbers and letters

Description automatically generated

Thay thế địa chỉ ip của “serverUrl” bằng IPv4 Address.

## 4.2 Thử nghiệm

### 4.2.1 Người quản trị (Admin)

|  |  |  |
| --- | --- | --- |
| Test case | Kịch bản | Tình trạng |
| Đăng nhập Admin | - Đăng nhập vào hệ thống với tài khoản Admin - Kiểm tra quyền hạn | Passed |
| Quản lý người dùng | - Thực hiện thêm, sửa, xóa người dùng bất kỳ  - Kiểm tra người dùng mới đã được thêm chưa. | Passed |
| Nhắn tin với người dùng | - Gửi tin nhắn đến người dùng và đảm bảo tin nhắn được gửi đi và nhận. | Passed |
| Quản lý dịch vụ | - Thêm, sửa, xóa một dịch vụ  - Xác nhận thay đổi đã được áp dụng. | Passed |
| Thống kê lịch sử | - Thực hiện một giao dịch, thanh toán  - Kiểm tra đã hiển thị đúng và đầy đủ chưa | Passed |
| Nạp tiền | - Nạp tiền theo số tiền người dùng đã đưa  - Kiểm tra số dư người dùng | Passed |

### 4.2.2 Người dùng (User)

|  |  |  |
| --- | --- | --- |
| Test case | Kịch bản | Tình trạng |
| Đăng nhập  người dùng | - Đăng nhập vào hệ thống với tài khoản người dùng - Kiểm tra quyền hạn | Passed |
| Giao dịch: nạp tiền vào ví | - Tiến hành nạp tiền vào thông qua cổng thanh toán Vnpay  - Kiểm tra số dư, lịch sử giao dịch đã chính xác chưa | Passed |
| Giao dịch: chuyển tiền | - Tiến hành chuyển tiền nội bộ  - Kiểm tra số dư 2 tài khoản, lịch sử giao dịch | Passed |
| Quản Lý Dịch Vụ | - Thêm, sửa, xóa một dịch vụ  - Xác nhận thay đổi đã được áp dụng. | Passed |
| Thống Kê Lịch Sử | - Thực hiện một giao dịch  - Kiểm tra đã hiển thị đúng giao dịch của tài khoản này chưa. | Passed |
| Thanh toán dịch vụ | -Sử dụng thẻ thanh toán để thanh toán 1 dịch vụ (ví dụ: gửi xe).  -Đảm bảo hệ thống hoạt động và xử lý chính xác số dư, cập nhập lịch sử thanh toán | Passed |
| Nộp học phí | -Kiểm tra học phí và thanh toán học phí  -Kiểm tra xem đã cập nhập lại học phí chưa và có còn hiển thị thông báo không. | Passed |
| Quản lý thông tin tài khoản | - Thử cập nhập thông tin (password).  -Log out và thử đăng nhập lại với mật khẩu | Passed |

## 4.3 Đánh giá kết quả

Sau một khoảng thời gian thực hiện nghiên cứu, xây dựng, thử nghiệm hệ thống với các mục tiêu đã đề ra, về cơ bản hệ thống đã hoàn thiện và tồn tại những ưu điểm, nhược điểm.

### 4.3.1 Ưu điểm

- Đầu tiên hệ thống đã được xây dựng hoàn chỉnh và đã có thể hoạt động trơn tru và ổn định.

- Hệ thống đã có thể đáp ứng được các yêu cầu cơ bản của một ví điện tử.

- Kết hợp với công nghệ IOT đã giải quyết vấn đề tồn đọng.

- Thời gian phản hồi nhanh, hiệu năng hoạt động tốt.

- Giao diện gần gũi, có khả năng tương tác cao, dễ dàng sử dụng.

- Có cấu trúc rõ ràng từ đó nâng cao tiềm năng mở rộng và phát triển.

### 4.3.2 Nhược điểm

- Do vấn đề hiểu biết, hệ thống vẫn chưa thể đáp ứng được đầy đủ các chức năng như các ví điện tử hiện tại trên thị trường.

- Bảo mật cần được tăng cường để có thể đảm bảo được tính bảo mật trong quá trình giao dịch.

- Khi đưa vào hoạt động thực tế, cần đưa ra các giải pháp tối ưu hơn để cải thiện thêm hiệu suất của hệ thống.

# **KẾT LUẬN**

Trong bài báo cáo với đề tài “Xây dựng ví điện tử nội bộ và hệ thống thanh toán tự động dựa trên nền tảng IOT”. Đã trình bày được vấn đề về nhà gửi xe ở trường Đại học Kinh tế Quốc dân và xác định phương hướng giải pháp để giải quyết vấn đề. Bên cạnh đó giải pháp còn có thể áp dụng giải pháp vào nhiều lĩnh vực khác trong nội bộ trường đại học Kinh tế quốc dân như bán đồ ăn, nước uống, thanh toán dịch vụ tại trường. Nhằm hướng tới việc gia tăng trải nghiệm của sinh viên đại học Kinh tế quốc dân, góp phần xây dựng môi trường học tập hiện đại.

Kế đó thông qua việc nghiên cứu, phân tích thiết kế và xây dựng hệ thống và kết hợp với những hiểu biết, công nghệ đã được trình bày bản thân em đã hoàn thiện hệ thống ví điện tử kết hợp với công nghệ IOT và đã đáp ứng được các yêu cầu đặt ra.

Tuy nhiên do sự hạn chế về mặt khiến thức và kinh nghiệm nên hệ thống của em vẫn chỉ dừng lại ở mức thử nghiệm thông qua những thiết bị đơn giản và dễ tiếp cận. Chính vì thế hệ thống vẫn chưa thực sự hoàn thiện và không thể tránh khỏi những thiếu sót, nhưng em tin rằng với giải pháp đã đề ra thì sẽ khả thi để có thể giải quyết vấn đề tồn đọng và đem lại những lợi ích nhất định cho sinh viên và nhà trường khi sử dụng hệ thống này.

# **TÀI LIỆU THAM KHẢO**