**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**THIẾT KẾ MÔ HÌNH CỦA HỆ THỐNG NHÀ THÔNG MINH**

**SMART HOME**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **TS. Nguyễn Quang Minh** |
| **Họ và tên:** | **Mai Danh Quý** |
| **Mã số sinh viên:** | **20182746** |
| **Khoá:** | **K63** |

**HÀ NỘI, 2/2024**

**ĐỀ TÀI TỐT NGHIỆP**

Biểu mẫu của Đề tài/khóa luận tốt nghiệp/thực tập theo qui định của Trường, tuy nhiên cần đảm bảo giáo viên giao đề tài ký và ghi rõ họ và tên.

Trường hợp có 2 giáo viên hướng dẫn thì sẽ cùng ký tên.

|  |
| --- |
| **Giáo viên hướng dẫn ký và ghi rõ họ tên** |
|  |

**Lời cảm ơn**

Những lời đầu tiên, em xin gửi lời cảm ơn sâu sắc nhất đến gia đình, người luôn bên cạnh, động viên và hỗ trợ tôi trong suốt thời gian học đại học. Sự ủng hộ và tình yêu thương từ gia đình là nguồn động viên lớn nhất, giúp em vượt qua những khó khan, thử thách trong học tập.

Em xin cảm ơn thầy Nguyễn Quang Minh. Trong quãng thời gian được thầy chỉ dẫn, tuy thầy khá bận với công việc giảng dạy của mình, nhưng thầy vẫn dành thời gian cho những buổi hướng dẫn em để hoàn thành đồ án tốt nghiệp. Em cảm ơn thầy.

Tiếp theo, em xin chân thành cảm ơn Trường Điện – Điện tử đã tạo lập một môi trường học tập, nghiên cứu tuyệt vời, giúp em có những hành trang về kiến thức và kĩ năng để phát triển tương lai của mình một cách tốt nhất .

Cuối cùng, em cảm ơn những người bạn, những người đồng đội đã cùng trải qua quá trình học hỏi này với em, là thứ thúc đẩy em tiến về phía trước và cũng học hỏi được nhiều điều từ nhau. Em xin chân thành cảm ơn gia đình, thầy cô, bạn bè.

**TÓM TẮT NỘI DUNG ĐỒ ÁN**

Hiện nay, con người ngày càng phát triển và sáng tạo trong lĩnh vực công nghệ, và xu hướng xây dựng và sử dụng nhà thông minh đã trở nên phổ biến. Điều này chủ yếu xuất phát từ nhu cầu tiện ích và thuận tiện trong cuộc sống hàng ngày. Công nghệ nhà thông minh mang lại sự linh hoạt cho người sử dụng, cho phép họ kiểm soát và tự động hóa nhiều hệ thống trong nhà như ánh sáng, nhiệt độ, an ninh, âm thanh và video.

Không chỉ làm cho cuộc sống trở nên dễ dàng hơn, mà còn giúp tiết kiệm năng lượng và tối ưu hóa hiệu suất của ngôi nhà. Sự tích hợp của các thiết bị di động và ứng dụng thông minh cung cấp khả năng quản lý từ xa, tăng cường sự an toàn và an ninh cho gia đình. Sự kết hợp với trí tuệ nhân tạo và máy học đưa đến khả năng học và thích ứng với thói quen cá nhân của người sử dụng, tạo ra một môi trường sống thông minh và hiện đại.

Không chỉ là một tiện ích cá nhân, mà còn là một động lực tăng giá trị cho bất động sản. Những nhà thông minh thường được đánh giá cao bởi người mua nhà vì tính hiện đại và tiện nghi của chúng. Cuối cùng, xu hướng này cũng thể hiện sự thích nghi của xã hội với thách thức của thời đại, khi con người ngày càng hướng đến sự tiến bộ và sáng tạo trong cuộc sống hàng ngày.

|  |
| --- |
| **Sinh viên thực hiện ký và ghi rõ họ tên** |
|  |

**MỤC LỤC**

**CHƯƠNG 1. GIỚI THIỆU CHUNG**

**1.1** Đặt vấn đề

**1.2** Mục tiêu

**1.3** Phạm vi đề tài

**1.4** Yêu cầu hệ thống

1.4.1 Yêu cầu chức năng

1.4.2 Yêu cầu phi chức năng

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1** Giới thiệu về **Angular**

Angular là một framework phía client mã nguồn mở, do Google phát triển và duy trì, với mục tiêu chính là hỗ trợ phát triển ứng dụng web đơn trang (Single Page Applications - SPAs). Dựa trên ngôn ngữ TypeScript, Angular mang lại sức mạnh của lập trình hướng đối tượng và kiểu dữ liệu tĩnh, giúp giảm lỗi và làm tăng khả năng bảo trì của mã nguồn.

Mô hình MVVM (Model-View-ViewModel) của Angular là một cách tiếp cận cơ bản giúp tổ chức mã nguồn một cách rõ ràng. Các thành phần như Components, Directives, Services, và Pipes đóng vai trò quan trọng trong việc phân chia logic và giao diện người dùng, tạo nên một kiến trúc có tổ chức.

Hệ thống module của Angular cho phép tách biệt ứng dụng thành các phần nhỏ, giúp quản lý mã nguồn một cách hiệu quả. Thư viện như RxJS được tích hợp sẵn, hỗ trợ quản lý luồng dữ liệu và thực hiện các thao tác bất đồng bộ.

Angular cung cấp cơ chế data binding mạnh mẽ, giúp đồng bộ dữ liệu giữa Model và View một cách tự động, giảm sự phức tạp của mã nguồn. Dependency Injection được sử dụng để quản lý các thành phần và dịch vụ, giúp làm giảm sự phụ thuộc và tạo ứng dụng dễ kiểm thử và mở rộng.

Angular CLI là một công cụ mạnh mẽ, hỗ trợ tự động hóa nhiều công việc như tạo component, module, service, và triển khai ứng dụng.

Với cộng đồng lớn và sự tích hợp linh hoạt với nhiều thư viện và công cụ bên ngoài, Angular trở thành một lựa chọn phổ biến cho các dự án phức tạp, mang lại ổn định và hiệu quả trong phát triển ứng dụng web.

**2.2** Giới thiệu về **Java, Java Spring Boot**

Java:

Java, một ngôn ngữ lập trình đa nền tảng, được ra đời từ tay của Sun Microsystems vào năm 1995, hiện nay đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. Điểm nổi bật của Java chính là khả năng chạy trên nhiều nền tảng khác nhau mà không yêu cầu sự sửa đổi mã nguồn, nhờ vào Java Virtual Machine (JVM).

Java theo hướng đối tượng, hỗ trợ các tính chất quan trọng như kế thừa, đa hình và đóng gói, giúp tạo ra mã nguồn linh hoạt, dễ hiểu và dễ tái sử dụng. Môi trường thực thi cảm biến lỗi (sandbox) của Java giúp tăng cường an toàn và bảo mật, ngăn chặn các vấn đề có thể xuất hiện từ mã độc hại.

Java còn được biết đến với cơ chế tự động quản lý bộ nhớ thông qua Garbage Collection, giảm áp lực quản lý bộ nhớ đối với lập trình viên. Cú pháp của Java được thiết kế để đơn giản và dễ đọc, giúp giảm độ phức tạp của mã nguồn và làm tăng tính hiệu quả trong quá trình phát triển.

Với cộng đồng lập trình viên lớn, Java sở hữu một hệ sinh thái thư viện và framework phong phú như Spring, Hibernate, Apache Struts, đồng thời được ứng dụng rộng rãi trong nhiều lĩnh vực như phát triển ứng dụng di động, web, dịch vụ máy chủ, và nhiều hệ thống lớn khác.

Java đã chứng minh sức mạnh và độ tin cậy của mình qua nhiều dự án trên toàn cầu, trở thành một công nghệ lập trình quan trọng và không thể phủ nhận trong cộng đồng phần mềm và công nghệ hiện đại.

Java spring boot :

Spring Boot, một phần quan trọng của hệ sinh thái Spring Framework, đem lại cho những nhà phát triển Java những lợi ích đáng kể trong quá trình xây dựng ứng dụng. Với sự chú trọng vào sự đơn giản và tính nhanh chóng, Spring Boot giảm bớt gánh nặng cấu hình thông thường bằng cách cung cấp cấu hình mặc định và quy ước, làm cho quá trình phát triển trở nên hiệu quả hơn.

Một điểm mạnh lớn của Spring Boot là khả năng chạy ứng dụng mà không cần máy chủ ứng dụng truyền thống. Điều này được đạt được thông qua việc tích hợp máy chủ nhúng như Tomcat hoặc Jetty, giúp đơn giản hóa quá trình triển khai và giảm độ phức tạp của hạ tầng.

Spring Boot không chỉ làm cho việc phát triển nhanh chóng mà còn giúp quản lý cấu hình thông qua tính năng tự động cấu hình. Các "starter" của nó giúp tích hợp dễ dàng với các công nghệ khác nhau như thư viện dữ liệu, web, an ninh, từ đó giảm bớt khả năng xung đột và tăng sự linh hoạt của ứng dụng.

Đặc biệt, Spring Boot Actuator là một tính năng quan trọng, cung cấp các endpoint để theo dõi và quản lý ứng dụng, tạo điều kiện thuận lợi cho việc giám sát, quản lý tác vụ, và kiểm soát sức khỏe của hệ thống.

Với cộng đồng lập trình viên Java lớn và sự hỗ trợ từ Pivotal Software, Spring Boot đang trở thành một công cụ quan trọng và phổ biến trong quá trình phát triển ứng dụng Java hiện đại.Trên cùng của Biểu mẫu

**2.3** Giới thiệu về **MySQL (XAMPP)**

XAMPP là một gói phần mềm mã nguồn mở được thiết kế để tạo ra môi trường phát triển web đầy đủ tính năng trên máy tính cá nhân. Tên của XAMPP là viết tắt từ "X" (dành cho mọi hệ điều hành), "Apache" (một máy chủ web), "MySQL" (hệ quản trị cơ sở dữ liệu), "PHP" (ngôn ngữ lập trình), và "Perl" (ngôn ngữ kịch bản). Nó bao gồm cả các thành phần cần thiết để xây dựng và kiểm thử ứng dụng web local, cung cấp môi trường giả lập máy chủ web mà không cần kết nối internet.

MySQL, một phần quan trọng trong XAMPP, là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến. Với khả năng lưu trữ và quản lý dữ liệu hiệu quả, MySQL thường được sử dụng trong việc phát triển ứng dụng web và quản lý cơ sở dữ liệu cho các trang web động. Nó hỗ trợ ngôn ngữ truy vấn SQL và cung cấp một giao diện dễ sử dụng cho việc tương tác với cơ sở dữ liệu. Sự kết hợp giữa XAMPP và MySQL tạo nên một môi trường lý tưởng để phát triển và thử nghiệm ứng dụng web một cách thuận tiện và nhanh chóng trên máy tính cá nhân.

**2.4** Giới thiệu về **ESP32**

NodeMCU là một nền tảng phần cứng mã nguồn mở dựa trên vi xử lý ESP8266 hoặc ESP32 của Espressif Systems, thường được sử dụng trong các dự án IoT và các ứng dụng nhúng. Dưới đây là một đoạn văn giới thiệu về NodeMCU:

NodeMCU là một nền tảng phần cứng linh hoạt và mạnh mẽ được phát triển dựa trên vi xử lý ESP8266 hoặc ESP32, sản phẩm của Espressif Systems. Được thiết kế với mục đích chính là hỗ trợ phát triển ứng dụng IoT, NodeMCU kết hợp một vi xử lý công suất cao với khả năng kết nối không dây tích hợp.

NodeMCU hỗ trợ ngôn ngữ lập trình Lua, giúp đơn giản hóa quá trình lập trình cho những người mới bắt đầu trong lĩnh vực IoT. Nó cũng có thể được lập trình bằng Arduino IDE hoặc PlatformIO, mở rộng sự thuận tiện và linh hoạt cho cộng đồng lập trình viên.

Với khả năng kết nối Wi-Fi tích hợp, NodeMCU có thể gửi và nhận dữ liệu qua mạng, điều này làm cho nó trở thành một lựa chọn phổ biến cho các dự án như đo lường và kiểm soát từ xa, cảm biến thông minh, và các ứng dụng theo dõi môi trường. Sự linh hoạt và chi phí thấp của NodeMCU làm cho nó trở thành một công cụ quan trọng trong việc phát triển các giải pháp IoT sáng tạo.

**CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**3.1** Đặc tả Usecase

3.1.1 Biểu đồ trình tự “Đăng ký”

3.1.2 Biểu đồ trình tự “Đăng nhập”

3.1.3 Biểu đồ trình tự “Điều khiển các thiết bị”

3.1.4 Biểu đồ trình tự “Lịch và các ghi chú, sự kiện”

3.1.5 Biểu đồ trình tự “Báo cháy”

**CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ**

**4.1** Giao diện trang chủ

**4.2** Giao diện trang đăng ký, đăng nhập

**4.3** Giao diện trang quản lý các thiết bị

**4.4** Giao diện trang điều khiển các thiết bị

**4.5** Cài đặt phần cứng ESP32

**TÀI LIỆU THAM KHẢO**

**PHỤ LỤC**

1. **Mô hình nhà**
2. **Mạch nguyên lý ESP32**  
   Diagram, schematic

   Description automatically generated
3. **Code ESP32**
4. **Code ứng dụng**