TRƯỜNG ĐẠI HỌC QUY NHƠN

**KHOA CÔNG NGHỆ THÔNG TIN**

---------



**ĐỒ ÁN CÔNG NGHỆ PHẦN MỀM**

***Đề tài:* Xây dựng phần mềm quản lý lưu trữ và số hóa tài liệu - Đa nền tảng**

Giảng viên: **Nguyễn Thị Loan**

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Đào Nhơn Tâm | 4151190005 |

**Quy Nhơn, năm 2022**

**MỤC LỤC**

**[DANH MỤC HÌNH 3](#_Toc17401)**

**[DANH MỤC CÁC CHỮ VIẾT TẮT](#_Toc5292)** [4](#_Toc5292)

**[MỞ ĐẦU](#_Toc21275)** [5](#_Toc21275)

**[CHƯƠNG I. KHẢO SÁT THỰC TẾ, TÌM HIỂU CƠ SỞ LÝ THUYẾT](#_Toc8426)** [6](#_Toc8426)

[1.1 Tổng quan về hệ thống quản lý lưu trữ và số hóa tài liệu 6](#_Toc3099)

*[1.1.1 Tìm hiểu về hệ thống quản lý lưu trữ hiện nay](#_Toc15581)* [6](#_Toc15581)

*[1.1.2 Cơ sở khoa học và tính thực tiễn của đề tài](#_Toc17159)* [6](#_Toc17159)

[1.2 Khảo sát hệ thống và thu thập thông tin 6](#_Toc22759)

*[1.2.1 Mục đích hệ thống](#_Toc2334)* [6](#_Toc2334)

*[1.2.2 Khảo sát một số phần mềm có chức năng tương tự hiện nay](#_Toc8742)* [6](#_Toc8742)

*[1.2.3 Ưu nhược điểm và những vấn đề cần cải tiến](#_Toc7286)* [7](#_Toc7286)

[1.3 Một số công nghệ hỗ trợ cho việc hiện thực hệ thống 9](#_Toc22018)

*[1.3.1 ASP .NET Core Framework](#_Toc1495)* [9](#_Toc1495)

*[1.3.2 JSON Web Token (JWT).](#_Toc7963)* [10](#_Toc7963)

*[1.3.3 Angular Framework](#_Toc24182)* [11](#_Toc24182)

*[1.3.4 Xamarin Forms](#_Toc11933)* [12](#_Toc11933)

**[CHƯƠNG II. PHÂN TÍCH HỆ THỐNG](#_Toc17522)** [14](#_Toc17522)

[2.1 Mô tả bài toán 14](#_Toc7496)

[2.2 Mô hình hoạt động chung của hệ thống 14](#_Toc31803)

[2.3 Phân tích chức năng, nghiệp vụ 14](#_Toc14767)

*[2.3.1 Giải thích các ký hiệu](#_Toc28248)* [14](#_Toc28248)

*[2.3.2 Phân tích yêu cầu](#_Toc21885)* [14](#_Toc21885)

*[2.3.4 Mô hình Use-case tổng quan](#_Toc27679)* [14](#_Toc27679)

[2.4 Mô hình tiến trình nghiệp vụ 14](#_Toc29114)

[2.5 Phân tích dữ liệu 14](#_Toc15191)

[2.6 Kết luận chương 14](#_Toc27241)

**[CHƯƠNG III. THIẾT KẾ HỆ THỐNG](#_Toc10215)** [15](#_Toc10215)

[3.1 Thiết kế kiến trúc hệ thống 15](#_Toc1021)

[3.2 Thiết kế cơ sở dữ liệu 15](#_Toc5117)

*[3.2.1 Sơ đồ logic](#_Toc4458)* [15](#_Toc4458)

*[3.2.2 Cơ sở dữ liệu vật lý](#_Toc584)* [15](#_Toc584)

[3.3 Kết luận chương 15](#_Toc24554)

**[CHƯƠNG IV. XÂY DỰNG HỆ THỐNG KIỂM TRA, ĐÁNH GIÁ VÀ SO SÁNH](#_Toc13124)** [16](#_Toc13124)

[4.1 Cài đặt phần mềm 16](#_Toc15780)

[4.2 Đánh giá và so sánh 16](#_Toc30122)

[4.3 Giới thiệu giao diện hệ thống 16](#_Toc23258)

[4.4 Kết luận chương 16](#_Toc2933)

**[KẾT LUẬN VÀ KIẾN NGHỊ](#_Toc337)** [17](#_Toc337)

**[TÀI LIỆU THAM KHẢO](#_Toc20523)** [18](#_Toc20523)

**DANH MỤC HÌNH**

[Hình 1. 1 Mô hình lưu trữ 7](#_Toc611)

[Hình 1. 2 Quy trình DocEye 7](#_Toc3072)

[Hình 1. 3 ASP .NET Core Framework 9](#_Toc4842)

[Hình 1. 4 Cách JWT được lấy và sử dụng để truy cập API hoặc resource 11](#_Toc14233)

[Hình 1. 5 Angular Framework 11](#_Toc28473)

[Hình 1. 6 Xamarin.Forms 12](#_Toc8579)

**DANH MỤC CÁC CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **Chữ viết tắt** | **Ý nghĩa** |
| JWT | JSON Web Token |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**MỞ ĐẦU**

Hiện nay, Công nghệ thông tin ngày nay đã có những phát triển vượt bậc trên mọi lĩnh vực của đời sống xã hội, trở thành một phần quan trọng không thể thiếu đối với con người.

Đặt giả sử chỉ cần Internet biến mất trên thế giới một ngày, chắc chắn thế giới con người như bị kéo tụt lùi lại vài trăm năm phát triển, mạng xã hội biến mất, các tập đoàn doanh nghiệp ngừng trệ hoạt động, hầu hết mọi lĩnh vực giải trí biến mất,… Điều đó cho thấy đời sống hiện đại của chúng ta hiện nay đang gắn chặt với các ứng dụng của ngành công nghệ thông tin như thế nào.

Bất cứ ngành nghề nào cũng cần sự đóng góp của ngành công nghệ thông tin để nâng cao năng suất cũng như chất lượng, giảm thiểu chi phí,… Do đó, trong công tác lưu trữ, việc công nghệ thông tin hóa các lĩnh vực quản lý lưu trữ và số hóa tài liệu mang đến rất nhiều lợi ích. Ngoài ra, chúng ta có thể thấy được mức độ phổ biến của smartphone, tablet,… Để đáp ứng được tính cơ động thì một hệ thống chạy đa nền tảng là xu thế hiện nay.

Từ những điều trên, sinh viên đã thực hiện đề tài: “Xây dựng phần mềm quản lý lưu trữ và số hóa tài liệu - đa nền tảng” nhằm xây dựng một hệ thống giúp các cơ quan lưu trữ và quản lý các tài liệu, văn bản, hồ sơ một cách hiệu quả hơn.

**CHƯƠNG I. KHẢO SÁT THỰC TẾ, TÌM HIỂU CƠ SỞ LÝ THUYẾT**

* 1. **Tổng quan về hệ thống quản lý lưu trữ và số hóa tài liệu**
     1. ***Tìm hiểu về hệ thống quản lý lưu trữ hiện nay***

Đa số các hệ thống quản lý lưu trữ hiện nay trên thị trường đều chia làm 2 phần là phần nhập liệu và phần kiểm duyệt.

Chúng ta sẽ đa nền tảng hóa với phần quản lý cùng với mục tiêu vào bộ phận nhập liệu lưu trữ, kiểm duyệt và kiểm soát dữ liệu và người quản lý giúp cho quy trình quản lý thêm đơn giản, dễ tương tác giữa các nhiệm vụ và kiểm soát hồ sơ tài liệu được nhập liệu và số hóa

* + 1. ***Cơ sở khoa học và tính thực tiễn của đề tài***

Cơ sở khoa học: Lưu trữ và số hóa tài liệu là một nhánh nhỏ trong quản lý và lư trữ kho tài liệu. Đây là công việc được kết hợp từ quản lý lưu trữ, tìm kiếm, thao tác với tài liệu và là nơi phê duyệt các loại tài liệu.

Tính thực tiễn: Hiện nay, việc quản lý tài liệu đang rất khó khăn trên thị trường thì rất ít các hệ thống quản lý lưu trữ và số hóa tài liệu. Do đó, việc xây dựng một hệ thống như thế là một điều cần thiết.

Hệ thống có giá trị trong việc lưu trữ các tài liệu mật, sao lưu các tài liệu để giúp cho việc kiểm soát tài liệu tốt hơn, lâu hơn, không bị ô nhiễm môi trường.

Vì vậy kết quả trong đề tài này có thể được sử dụng để áp dụng thực tế để giám sát, kiểm soát, lưu trữ, thiết lập hệ thống siêu dữ liệu và số hóa tài liệu cho các cơ quan nhà nước hiện nay.

* 1. **Khảo sát hệ thống và thu thập thông tin**
     1. ***Mục đích hệ thống:***

Với các phương pháp quản lý tài liệu truyền thống trước đây, việc quản lý tài liệu thường được thực hiện bằng cách ghi nhận, sao lưu lại trên tài liệu giấy, sổ sách theo dõi, … sau đó lưu lại trong thư viện hoặc các kho lưu trữ. Bằng cách đánh dấu mã số tài liệu và ghi chép thông tin vào các kẹp file hồ sơ, người ta có thể tìm kiếm lại những thông tin tài liệu khi cần.

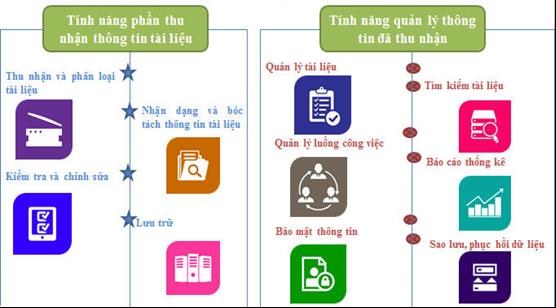
Tuy nhiên, với cách làm này khối lượng những công văn giấy tờ, hồ sơ, tài liệu của doanh nghiệp sẽ tăng lên rất nhiều theo thời gian. Điều này khiến doanh nghiệp bắt đầu lâm vào tình cảnh khó khăn khi mà việc quản lý thông tin tài liệu, tra cứu hồ sơ bắt đầu rơi vào trạng thái quá tải và khó kiểm soát. Không những thế, việc lưu trữ tài liệu trong kho cũng khiến doanh nghiệp tốn một khoản chi phí không hề nhỏ cho việc quản lý, kho bãi và bảo quản.

Thay vì phải tốn nhiều thời gian, công sức cho công việc quản lý tài liệu theo phương pháp truyền thống, việc sử dụng phần mềm quản lý tài liệu đang được khuyến khích áp dụng.

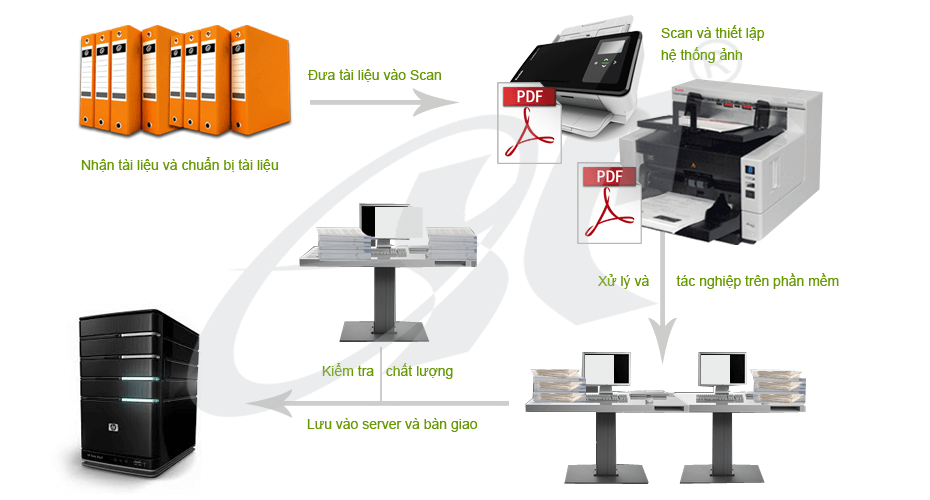
Chính vì lẽ đó, việc lựa chọn phần mềm quản lý tài liệu tốt, đáp ứng được nhu cầu về lâu về dài của doanh nghiệp được đánh giá là quan trọng.

* + 1. ***Khảo sát một số phần mềm có chức năng tương tự hiện nay:***

Điển hình như hệ thống Phần Mềm Quản Lý Tài Liệu Thông Minh DocEyey - Phần mềm quản lý tài liệu thông minh DocEye là giải pháp giúp doanh nghiệp quản lý, sắp xếp, lưu trữ và bảo quản tài liệu chuyên nghiệp trên nền tảng số. Doanh nghiệp có thể tìm kiếm, chia sẻ tài liệu nhanh chóng, dễ dàng không giới hạn thời gian, địa điểm. Nhờ đó hỗ trơ quá trình tạo lập quy trình làm việc và nâng cao sự liên kết giữa các phòng ban. Ngoài ra DocEye còn bảo mật tài liệu an toàn qua cơ chế phân quyền chi tiết đến từng người dùng.



Hình 1. 1 Mô hình lưu trữ



Hình 1. 2 Quy trình DocEye

* + 1. ***Ưu nhược điểm và những vấn đề cần cải tiến***
* **Ưu điểm:**

Không chỉ giúp cải tiến công tác quản lý tài liệu, phần mềm quản lý văn bản còn có nhiều ưu điểm vượt trội, mang lại nhiều lợi ích thiết thực:

**Tiết kiệm thời gian, chi phí**

Tiết kiệm được nhiều thời gian cho việc: Tạo lập, vận chuyển, lưu trữ và xử lý văn bản. Tiết kiệm được thời gian di chuyển do sử dụng cơ sở dữ liệu tập trung. Không mất nhiều công sức để tìm kiếm, tra cứu và theo dõi hồ sơ. Cắt giảm chi phí mua văn phòng phẩm, in ấn, mực, giấy, bảo trì máy móc,… Ngoài ra còn giảm chi phí thuê nhân sự nhờ có sự hỗ trợ của hệ thống.

**Nâng cao hiệu suất công việc**

Phân bổ công việc hợp lý hơn vì các công việc được thực hiện hoàn toàn trên hệ thống. Người dùng dễ dàng tìm kiếm các thông tin cần thiết và xử lý các dữ liệu trên hệ thống, do đó tiết kiệm được thời gian xử lý công việc, tập trung cho các việc khác quan trọng hơn. Với đa nền tảng, người dùng có thể truy cập mọi nơi, mọi lúc để quản lý và theo dõi tiến độ công việc.

**Lưu trữ dữ liệu an toàn, bảo mật**

Có lẽ điểm bất lợi lớn của văn phòng truyền thống là lưu trữ văn bản, hồ sơ và tài liệu thủ công, có nhiều nguy cơ về mất, cháy, hỏng, thất lạc. Hệ thống quản lý văn bản đã khắc phục tối đa các vấn đề này

Tất cả các văn bản, hồ sơ,.. của doanh nghiệp sẽ được lưu trữ an toàn, tối đa bảo mật trên một hệ thống cơ sở dữ liệu duy nhất. Chỉ những người dùng có tài khoản và được phân quyền mới được truy cập, sử dụng và quản lý văn bản. Giảm thiểu tối đa về mất mát dữ liệu, gây thiệt hại cho các doanh nghiệp.

* **Nhược điểm:**

Tài liệu điện tử là tài liệu đọc bằng máy, chúng tồn tại trong môi trường ảo. Khác với tài liệu giấy, thông tin luôn gắn liền với vật mang tin và là một thực thể thống nhất. Loại hình tài liệu này chỉ có thể sử dụng được với sự trợ giúp của máy tính. Điều này luôn tiềm ẩn những nguy cơ. Thứ nhất – để sử dụng chúng, cần sự hiện hữu của máy tính điện tử. Thứ hai – luôn có nguy cơ hủy hoại toàn bộ tài liệu khi có sự trục trặc về máy móc hay do sự xâm nhập của virus.

Tính án toàn thông tin**:** Đối với văn bản điện tử, việc bảo đảm an toàn thông tin cao hơn so với tài liệu giấy. Tuy nhiên, sự đơn giản trong vấn đề sửa đổi và sao chép thông tin là một đe dọa đối vợi dự an toàn thông tin trong nguồn tài liệu điện tử. Chỉ với những kỹ năng sử dụng máy tính đơn giản nhất, người ta có thể sửa đổi nội dung tài liệu điện tử hoặc sao chép (một phần hay toàn bộ) tài liệu mà hoàn toàn không để lại dấu vết. Đây thực sự là một mối đe dọa lớn đối với tình an toàn thông tin của nguồn tài liệu tồn tại trong môi trường điện tử.

Chính vì thế mà các doanh nghiệp còn lưỡng lự trong việc có nên sử dụng hình thức quản lý tài liệu này không.

Người dùng đăng nhập vào hệ thống sẽ được tìm kiếm tài liệu theo từ khóa, từ khóa sẽ tìm trong tiêu đề và trong mô tả của tài liệu.

Kết quả tìm kiếm sẽ chỉ hiển thị danh mục những tài liệu người đó có quyền xem, không được phép xem. Ở kết quả tìm kiếm người tìm có thể xem tài liệu đó trực tuyến (nếu được quyền xem), biết được vị trí của tài liệu gốc ở kho lưu trữ để có thể hỏi mượn nếu cần.

Về thu thập tài liệu lưu trữ điện tử, Nghị định quy định, trường hợp tài liệu lưu trữ điện tử và tài liệu lưu trữ giấy có nội dung trùng nhau thì thu thập cả hai loại. Khi giao nhận tài liệu lưu trữ điện tử, Lưu trữ cơ quan, Lưu trữ lịch sử phải kiểm tra tính xác thực, tính toàn vẹn và khả năng truy cập của hồ sơ. Đồng thời, hồ sơ phải bảo đảm nội dung, cấu trúc và bối cảnh hình thành và được bảo vệ để không bị hư hỏng hoặc hủy hoại, sửa chữa hay bị mất dữ liệu.

* 1. **Một số công nghệ hỗ trợ cho việc hiện thực hệ thống**
     1. ***ASP .NET Core Framework***

ASP.NET Core là tập hợp thư viện như một Framework mới được xây dựng ứng dụng web khi có kết nối internet, nó là một sản phẩm của Microsoft khá nổi tiếng trong cộng đồng lập trình hiện nay khi có rất nhiều blogger, vblog về công nghệ thể hiện sự chú ý và quan tâm đến nó. Ngay khi vừa xuất hiện đã có hàng loạt các ASP.NET Core tutorial, bài viết so sánh, hướng dẫn, thảo luận được đưa ra mổ xẻ.

Với ASP.NET các bạn có thể áp dụng để làm nhiều mục đích làm việc khác nhau của bạn ví dụ như: Xây dựng các ứng dụng web, các dịch vụ, ứng dụng IoT và các phần backen mobile; sử dụng các công cụ phát triển ưa thích trên Windows, macOS và Linux; Triển khai trên đám mây hoặc ngay tại chỗ.



Hình 1. 3 ASP .NET Core Framework

Khi làm việc với ASP.NET Core bạn đã được những cải tiến đáng kể như:

* Tương thích hệ thống xây dựng Web UI và Web APIs.
* Tích hợp những client –side frameworks hiện đại và những luồng phát triển
* Hệ thống cấu hình dựa trên mô hình đám mây thật sự
* Dependency injection được xây dựng sẵn
* HTTP request được tối ưu nhẹ hơn
* Có thể host trên IIS hoặc self- host trong process của riêng mình
* Được xây dựng trên .NET Core, hỗ trợ thực sự app vesioning
* Chuyển thực thể, thành phần, module như những NuGet packages
* Các công cụ mới để đơn giản hóa quá trình phát triển web hiện đại.
* Xây dựng và chạy đa nền tảng
* Mã nguồn mở, tập trung vào cộng đồng
  + 1. ***JSON Web Token (JWT)***

**Token-based authentication** là phương thức xác thực bằng chuỗi má hóa. Một hệ thống sử dụng **Token-based authentication** cho phép người dùng nhập user/password để nhận về 1 chuỗi token. Chuỗi Token này được sử dụng để “xác minh” quyền truy cập vào tài nguyên mà không cần phải cung cấp lại username/password nữa.

JWT được viết tắt từ **JSON Web Token**, đây là một phương tiện đại diện cho các bên yêu cầu chuyển giao giữa server và client. JWT cũng được xem như một tiêu chuẩn mở (RFC 7519) nhằm xác minh thông tin an toàn giữa các bên Client-Server dưới dạng JSON object.

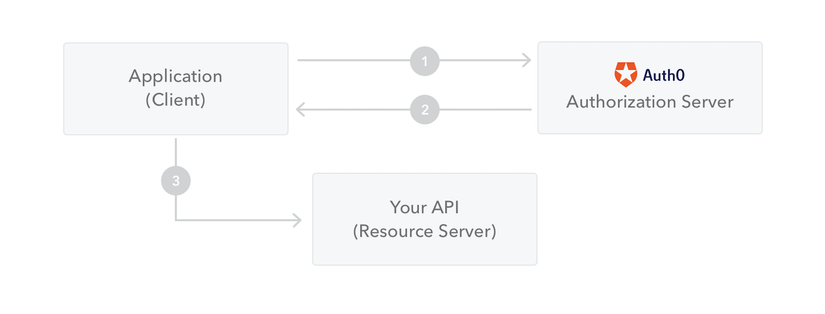
Các chuỗi thông tin dạng JSON sẽ tiến hành mã hoá để trở thành một chuỗi ký tự lộn xộn, không có trật tự nhất định và rất khó hiểu khi nhìn vào. Thông tin này có thể được xác minh và tin cậy vì nó được ký điện tử – digitally signed. JWT có thể được ký bằng cách sử dụng một secret (với thuật toán HMAC) hoặc cặp public/private key dùng chuẩn RSA hoặc ECDSA.

Trong xác thực, khi người dùng đăng nhập thành công bằng thông tin đăng nhập của họ, JSON Web Token sẽ được trả về.

Bất cứ khi nào người dùng muốn truy cập route hoặc resource được bảo vệ, tác nhân người dùng nên gửi JWT, thêm Authorization trong header với nội dung là Bearer + token. Nội dung của header sẽ trông như sau:

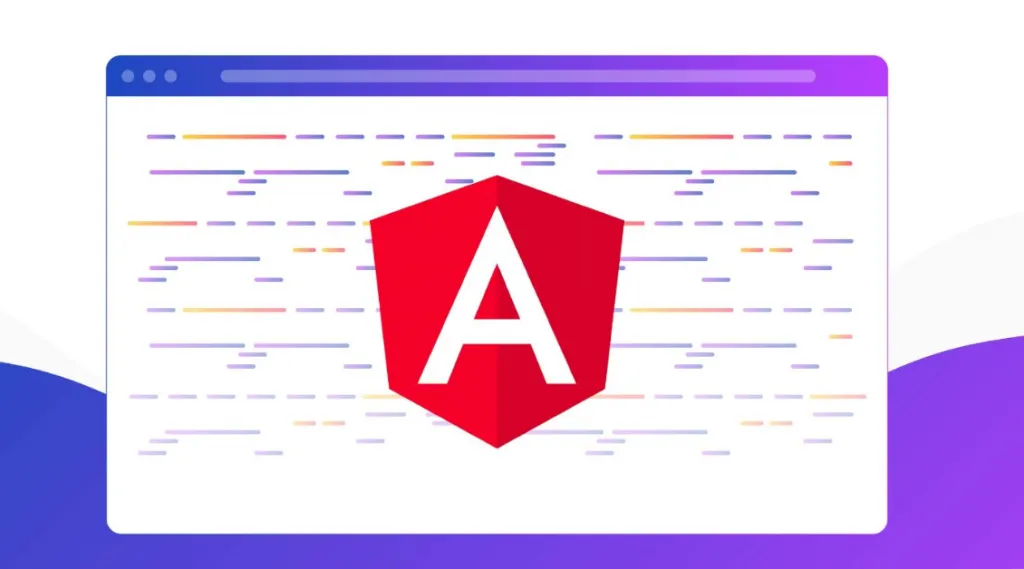
Authorization: Bearer <token>

Sơ đồ sau đây cho thấy cách JWT được lấy và sử dụng để truy cập API hoặc resource:



Hình 1. 4 Cách JWT được lấy và sử dụng để truy cập API hoặc resource

* + 1. ***Angular Framework***



Hình 1. 5 Angular Framework

Angular là một framework JavaScript và được viết bằng TypeScript. Google tạo ra framework này với mục đích viết nên giao diện web (Front-end) đúng chuẩn “ít nỗ lực”. Không chỉ đem lại những lợi ích như một framework, mà Angular còn giữ nguyên cấu trúc như ngôn ngữ lập trình tiêu chuẩn. Điều đó giúp các nhà phát triển vừa dễ mở rộng dự án cũng như việc bảo trì.

* **Ưu điểm:**

Angular cho phép người dùng xây dựng các component (thành phần) của riêng họ, có thể đóng gói chức năng cùng với logic cho các component để tạo thành các phần có thể tái sử dụng. Các component này cũng hoạt động tốt với các component khác của web.

Data Binding là kĩ thuật dùng để tạo gắn kết giữa phần giao diện (UI) và dữ liệu thông qua phần business logic ( tạm dịch – logic xử lý việc trao đổi thông tin dữ liệu với giao diện người dùng). Nhờ Data Binding, UI có thể tự động cập nhật lại để hiển thị các thay đổi trong dữ liệu.

Angular cho phép người dùng dễ dàng di chuyển dữ liệu từ mã JavaScript sang chế độ xem và phản ứng với các sự kiện của người dùng mà không cần phải viết bất kỳ mã nào theo cách thủ công.

Angular cho phép người dùng viết các câu lệnh và đưa chúng vào bất cứ nơi nào họ cần. Điều này cải thiện khả năng kiểm tra và khả năng tái sử dụng của các câu lệnh giống nhau.

Angular là framework được Google chống lưng nên đây là framework có “ bảo hiểm” và cung cấp các giải pháp tiện ích toàn diện cho giao tiếp máy chủ.

Angular là đa nền tảng và tương thích với nhiều trình duyệt. Một ứng dụng Angular thường có thể chạy trên tất cả các trình duyệt (Ví dụ: Chrome, Firefox) và hệ điều hành, chẳng hạn như Windows, macOS và Linux.

* **Nhược điểm:**

Các thành phần cơ bản của Angular mà tất cả người dùng nên biết bao gồm chỉ thị (directives), mô-đun, trình trang trí (decorators), thành phần (components), dịch vụ (services), dependency injection, pipes, và mẫu (templates)...Các chủ đề nâng cao hơn bao gồm biên dịch AoT và Rx.js… Đối với người mới bắt đầu, Angular 4 có thể khó học vì nó là framework hoàn chỉnh.

Một vấn đề phổ biến trong cộng đồng Angular là sự rườm rà của framework này. Nó cũng khá phức tạp so với các công cụ front-end khác.

* + 1. ***Xamarin Forms***

Xamarin Forms cho phép các nhà phát triển phần mềm xây dựng các ứng dụng di động đa nền tảng sử dụng các thành phần giao diện cơ bản của hệ điều hành như: Pages, Layouts, Views, Controls. Khi chạy chương trình, mỗi thành phần giao diện của Xamarin Forms sẽ được chuyển hóa sang các thành phần giao diện tương ứng với mỗi nền tảng.

Hình 1. 6 Xamarin.Forms

Việc lựa chọn nền tảng ứng dụng phát triển mobile Xamarin vì nó làm giảm thời gian tiếp cận thị trường và chi phí kỹ thuật, bằng cách chia sẻ mã code và sử dụng single technology. Mặt khác nó cũng phù hợp để tạo ra app native giữa các nền tảng đơn giản và đây sẽ là lựa chọn hợp lý cho hệ thống quản lý lưu trữ tài liệu của chúng ta.

**CHƯƠNG II. PHÂN TÍCH HỆ THỐNG**

**2.1 Mô tả bài toán**

**2.2 Mô hình hoạt động chung của hệ thống**

**2.3 Phân tích chức năng, nghiệp vụ**

***2.3.1 Giải thích các ký hiệu***

***2.3.2 Phân tích yêu cầu***

***2.3.4 Mô hình Use-case tổng quan***

**2.4 Mô hình tiến trình nghiệp vụ**

**2.5 Phân tích dữ liệu**

**2.6 Kết luận chương**

**CHƯƠNG III. THIẾT KẾ HỆ THỐNG**

**3.1 Thiết kế kiến trúc hệ thống**

**3.2 Thiết kế cơ sở dữ liệu**

***3.2.1 Sơ đồ logic***

***3.2.2 Cơ sở dữ liệu vật lý***

**3.3 Kết luận chương**

**CHƯƠNG IV. XÂY DỰNG HỆ THỐNG KIỂM TRA, ĐÁNH GIÁ VÀ SO SÁNH**

**4.1 Cài đặt phần mềm**

**4.2 Đánh giá và so sánh**

**4.3 Giới thiệu giao diện hệ thống**

**4.4 Kết luận chương**

**KẾT LUẬN VÀ KIẾN NGHỊ**

**TÀI LIỆU THAM KHẢO**