**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

 **ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**CÔNG NGHỆ LẬP TRÌNH ĐA NỀN TẢNG**

**CHO ỨNG DỤNG DI ĐỘNG**

**Đề tài: XÂY DỰNG ỨNG DỤNG ĐỌC TIN TỨC**

**Lớp:** IE307.K111

**Giáo viên hướng dẫn:** Ths. Võ Ngọc Tân

**Sinh viên thực hiện:**

BÙI NGUYÊN MÃO 16520724

NGUYỄN VĂN TÀI 16521792

VŨ VĂN PHÚ 16521609

***TP. Hồ Chí Minh, ngày 27 tháng 12 năm 2019***

Nhận xét

**Mục lục**

[A. GIỚI THIỆU ĐỀ TÀI 4](#_Toc28339123)

[1. Lý do chọn đề tài 4](#_Toc28339124)

[2. Mục tiêu đề tài 4](#_Toc28339125)

[3. Phương pháp giải quyết 4](#_Toc28339126)

[B. Nội dung 5](#_Toc28339127)

[1. Mô hình hệ thống 5](#_Toc28339128)

[2. Phần Client 6](#_Toc28339129)

[2.1. Đặc điểm 6](#_Toc28339130)

[2.2. Công nghệ sử dụng 7](#_Toc28339131)

[3. Phần Server: 11](#_Toc28339132)

[1.1. Đặc điểm 11](#_Toc28339133)

[1.2. Công nghệ sử dụng 11](#_Toc28339134)

[4. Phần Crawler 16](#_Toc28339135)

[1.3. Đặc điểm 16](#_Toc28339136)

[1.4. Công nghệ sử dụng 16](#_Toc28339137)

[C. TỔNG KẾT 18](#_Toc28339138)

[1. Kết quả đạt được 18](#_Toc28339139)

[2. Cải thiện và phát triển 18](#_Toc28339140)

[3. Kết luận, đánh giá 19](#_Toc28339141)

# GIỚI THIỆU ĐỀ TÀI

## Lý do chọn đề tài

Trong mọi thời đại, con người đã thay đổi rất nhiều, nhưng luôn luôn có nhu cầu nắm bắt tin tức diễn ra hàng ngày. Với sự thành công vang dội của việc phát hành báo giấy, nó đã giúp con người nắm bắt tin tức nhiều hơn. Bước qua thế kỉ 21, mọi người đã không còn nhiều thời gian cho việc đọc báo giấy, việc đọc báo trở nên khó khăn hơn. Chính vì thế, báo mạng đã ra đời, với sự phát triển mạnh mẽ của công nghệ thông tin thì việc đưa tin tức lên mạng cực kỳ dễ dành. Con người có thể đọc tin tức trên chiếc màn hình nhỏ, tiện gọn và bất kì đâu.

Chưa dừng lại ở đó, một vấn đề mới được phát sinh, đó chính là những tờ báo lá cải, dài dòng, tung hứng nhau trên mạng làm cho đọc giả khó nắm được những tin hot, tin quan trọng. Chính vì lý do này, chúng em đã tạo ra một ứng dụng giải quyết được tất cả các vấn đề trên. Sẽ không còn những tin nhảm, sẽ không còn việc cầm tờ báo vướng víu, việc đọc tin tức sẽ dễ dàng hơn, nhanh hơn, tiện hơn khi nó đã nằm trên màn hình điện thoại của bạn. Và, ngày nay với việc nhiều hệ điều hành ra đời, nhiều dòng máy được tạo ra, chúng em cũng đã giải quyết được vấn đề đấy. Với framework xamarin forms, chỉ cần viết 1 lần, ứng dụng đã có thể chạy trên nhiều nền tảng khác nhau. Giờ đây, việc đọc tin tức trên chiếc điện thoại nhỏ gọn của bạn đã được ra đời.

## Mục tiêu đề tài

Bằng cách nghiên cứu các ứng dụng đọc tin tức tương tự trên thị trường, chúng em hy vọng sẽ tạo ra được một ứng dụng đáp ứng được nhiều tiêu chí của người dùng. Ứng dụng luôn cập nhật các tin tức mới nhất, hot nhất trong ngày, tuần cũng như tháng để người dùng luôn luôn có tin để đọc. Ngoài ra, giao diện phải đơn giản, dễ sử dụng nhưng vẫn đảm bảo được đầy đủ các chức năng. Cuối cùng, phải đảm bảo được phần mềm có khả năng chịu lỗi cao, dễ cập nhật, dễ sửa chữa.

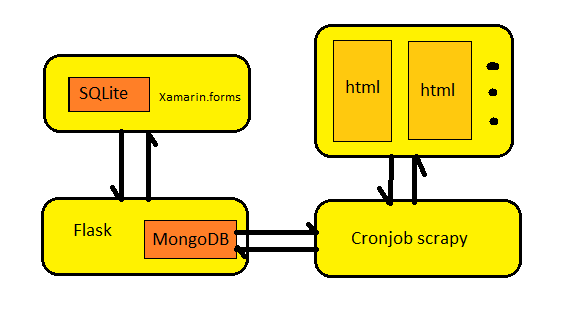
Ngoài ra, chúng em sẽ luôn tìm hiểu các ứng dụng liên quan, rút ra được các chức năng hấp dẫn, tiện ích mới để bổ sung vào phần mềm để tạo ra ứng dụng đọc tin tức tốt nhất có thể.

## Phương pháp giải quyết

Với đề tài lần này, chúng em sử dụng công nghệ nền tảng cross-platform Xamarin để biết giao diện, thư viện phần mềm SQLite để quản lý cơ sở dữ liệu bên phía client và MongoDB để quản lý dữ liệu phía server. Áp dụng mô hình textrank vào vấn đề rút gọn nội dung cho bài tin tức và scrapy để thu thập dữ liệu từ các trang báo mạng.

# Nội dung

## Mô hình hệ thống



Hình 1 Mô hình hệ thống

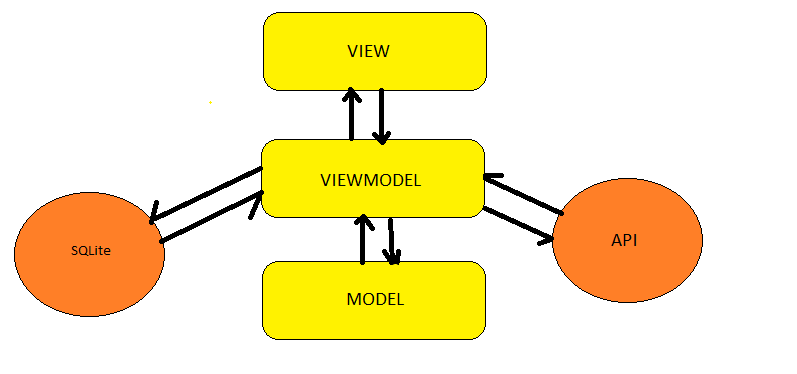
Ứng dụng gồm 3 phần:

* Sử dụng nền tảng Xamarin Forms với mô hình MVVM để xây dựng giao diện ứng dụng đọc tin tức hỗ trợ đa nền tảng (Android, IOS).
* Xây dựng một RESTful API server bằng Python và flask framework trên cơ sở dữ liệu MongoDB có khả năng xử lý các yêu cầu từ client.
* Một crawler bot để thu thập các bài báo trên các trang báo mạng bằng Python và Scrapy Framework.

Ứng dụng sẽ đáp ứng được nhu cầu thực tế với giao diện đơn giản, dễ sử dụng. Các thành phần tinh gọn, dễ bảo trì và nâng cấp.

## Phần Client

### Đặc điểm



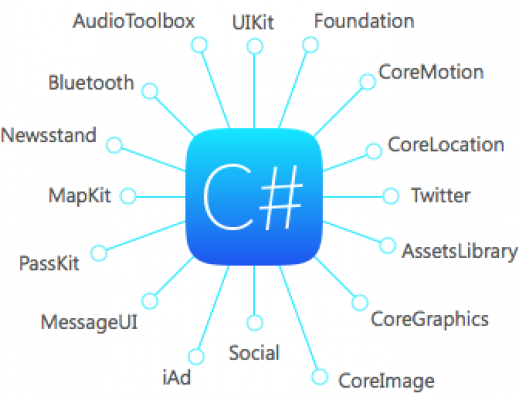
Hình 2 Mô hình phần client

Đây là bộ mặt của toàn bộ ứng dụng, tác động trực tiếp đến người dùng, chúng em cố gắng tạo ra giao diện đơn giản, thuận tiện, dễ thao tác với đầy đủ tính năng:

* Bằng mô hình MVVM, các thành phần được sắp xếp rõ ràng đúng mục đích sử dụng.
* Cơ sở dữ liệu SQLite làm bộ nhớ local, với ưu điểm gọn nhẹ được sử dụng để lưu trữ các dữ liệu tĩnh, ít bị thay đổi như danh mục menu, chủ đề tin tức.
* Xây dựng các interface dễ dàng giao tiếp với server bằng các http request, nhận và xử lý response là các nội dung bài tin tức.

### Công nghệ sử dụng

#### Xamarin.Forms

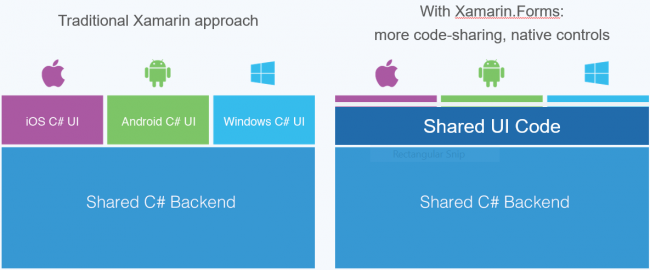


Hình 3 Xamarin được xây dựng dựa trên C#

Xamarin là một framework dùng để xây dựng các ứng dụng di động đa nền tảng được tạo ra bởi hãng phần mềm di động cùng tên. Thực chất Xamarin là một công nghệ có nền tảng cross-platorm. Công nghệ này được phát triển bởi Microsoft với mục đích chính là giúp những lập trình viên có thể viết 1 lần mã code nhưng vẫn có thể build được cả 3 ứng dụng chạy trên nền tảng IOS, Android, UWP. Với những nền tảng khác sau này cũng có thể được Microsoft hỗ trợ bằng Xamarin.

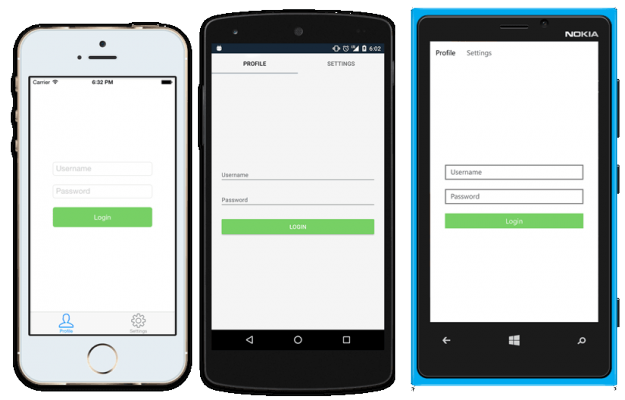
Xamarin sử dụng ngôn ngữ C# và thư viện native được “gói” trong lớp .Net để hỗ trợ phát triển ứng dụng đa nền tảng. Những ứng dụng như thế thường được so sánh với các ứng dụng native trên các nền tảng lập trình iOS và Android theo performance và trải nghiệm người dùng (user experience).

* **Ưu điểm của Xamarin Forms:**
  + - Các công cụ của Xamarin về cơ bản sẽ cho phép bạn phát triển các ứng dụng Android và iOS bằng ngôn ngữ C# và có thể chia sẻ rất nhiều phần code giữa các ứng dụng với nhau Khi viết một ứng dụng sử dụng bộ công cụ của Xamarin thì về cơ bản bạn đanmg sử dụng một lớp trừu tượng phía trên các SDK thực sự của iOS và Android . Điều này có nghĩa là bạn sẽ thu được kết quả là một ứng dụng native hoàn toàn với giao diện dùng native trên mỗi nền tàng. Một số điểm cộng cho Xamarin:
* Có khả năng lựa chọn UI layout.
* Tích hợp Oauth.
* Tích hợp REST APIs từ xa.
* Công nghệ dẫn đường và sử lý tín hiệu thời gian thực cho ứng dụng định vị.
* Tích hợp mạng xã hội.
* Có bộ cơ sở dữ liệu SQLite nhúng.
* Thư viện XAML cho phép xây dựng một loạt ứng dụng.
* Chúng ta có thể xây dựng ứng dụng mobile với Xamarin Forms với sự chia sẻ 80 đến 100 % nội dung code của bạn cho tất cả các nền tảng.



Hình 4 Cấu trúc xamarin và xamarin.forms

* Ứng dụng khá gần với giải pháp tự nhiên.Cảm giác nhìn chung là nền tảng cho phép thực hiện nhiều chức năng để tạo ra một ứng dụng một cách rất native.
* Với 1 đoạn code chúng ta có thể build ra 3 nền tảng khác nhau:



Hình 5 Cùng một câu lệnh, giao diện hiển thị phù hợp cho từng nền tảng

* + - * **Nhược điểm của Xamarin Forms**
* Mặc dù vậy, Xamarin Forms cũng có những khuyết điểm AOT compliler. Nó không được complie các đoạn code gọn gàng như Xcode. Hạn chế của Xamarin ảnh hưởng đến cả phát triển, test tải app lên store. Hạn chế chính đối với iOS là nó không hỗ trợ sinh code tự động. Visual Studio biuld code lâu. Cái này tất cả các lập trình viên Xamarin đều đang phải sống chung với lũ, một lần build sẽ mất rất lâu từ 3 đến 10 phút cho solution với 3 project.
* Đôi lúc thông báo lỗi không rõ ràng chủ yếu trên Android vì nó là 1 công nghệ mới nên vẫn chưa được hoàn hoản mong rằng Microsoft sẽ fix trong tương lai.
* Ứng dụng Xamarin lớn hơn,nặng hơn so với ứng dụng native. So sánh với ứng dụng native nó chiếm nhiều hơn vài Mb so với Java/Objective C tương ứng. kích thước của một ứng dụng code bằng xamarin là 3Mb, trong khi code bằng Objective C chỉ chiếm 172 Kb. Càng sử dụng nhiều API, càng nhiều lưu trữ bị chiếm trên thiết bị.

#### SQLite

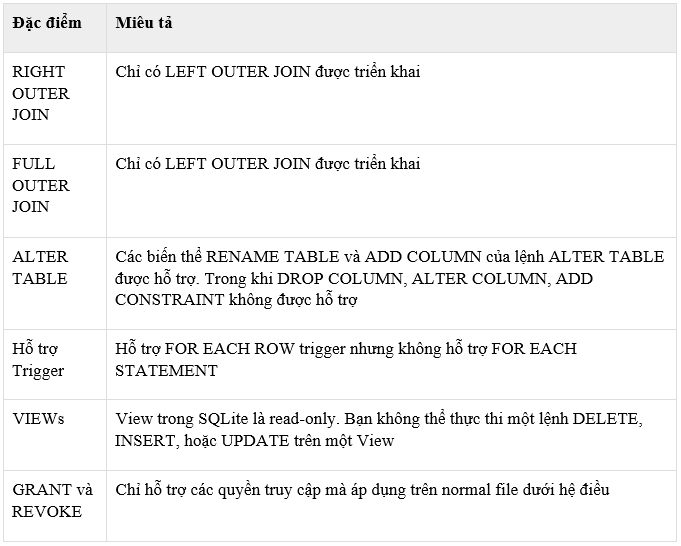


Hình 6 Logo của SQLite

SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine, không cần máy chủ, không cần cấu hình, khép kín và nhỏ gọn. Nó là một cơ sở dữ liệu, không cần cấu hình, có nghĩa là giống như các cơ sở dữ liệu khác mà bạn không cần phải cấu hình nó trong hệ thống của mình.

SQLite engine không phải là một quy trình độc lập (standalone process) như các cơ sở dữ liệu khác, bạn có thể liên kết nó một cách tĩnh hoặc động tùy theo yêu cầu của bạn với ứng dụng của bạn. SQLite truy cập trực tiếp các file lưu trữ (storage files) của nó.

* + **Ưu điểm của SQLite**
* SQLite không yêu cầu một tiến trình Server riêng rẽ để hoạt động.
* SQLite không cần cấu hình, nghĩa là không cần thiết phải cài đặt.
* Một SQLite Database đầy đủ được lưu giữ trong một disk file đơn.
* SQLite là rất nhỏ gọn, nhỏ hơn 400kB đã đươc cấu hình đầy đủ hoặc nhỏ hơn 250kB khi đã bỏ qua các tính năng tùy ý.
* SQLite là tự chứa, nghĩa là không có sự phụ thuộc vào ngoại vi.
* Các Transaction trong SQLite là tuân theo đầy đủ chuẩn ACID, đảm bảo truy cập an toàn từ nhiều tiến trình hoặc thread.
* SQLite hỗ trợ hầu hết các tính năng của một ngôn ngữ truy vấn trong chuẩn SQL92.
* SQLite được viết bằng ANSI-C và cung cấp API đơn giản và dễ dàng sử dụng.
* SQLite là có sẵn trên UNIX (Linux, Mac OS-X, Android, iOS) và Windows (Win32, WinCE, WinRT).
* **Nhược điểm của SQLite**



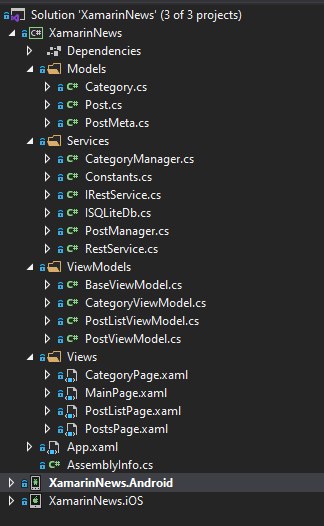
Hình 7 Các câu lệnh cơ bản SQlite

* **Tại sao phải sử dụng SQLite trong Xamarin form?**

Hầu hết các ứng dụng làm việc điều cần dữ liệu. Điều này không chỉ đúng với các ứng dụng trên máy tính để bàn và Web mà còn cho các ứng dụng di động. Trong nhiều trường hợp, các ứng dụng di động trao đổi dữ liệu qua mạng và tận dụng dịch vụ lưu trữ và dịch vụ đám mây như thông báo đẩy. Tuy nhiên, có những tình huống trong đó các ứng dụng di động chỉ cần lưu trữ dữ liệu cục bộ. Với dữ liệu đơn giản, không có cấu trúc, chẳng hạn như cài đặt và tùy chọn của người dùng, các ứng dụng có thể lưu trữ thông tin bên trong các tệp cục bộ, chẳng hạn như XML hoặc văn bản hoặc thông qua các đối tượng cụ thể được cung cấp bởi các nền tảng phát triển khác nhau. Trong trường hợp dữ liệu có cấu trúc phức tạp, các ứng dụng cần một cách khác để lưu trữ thông tin.

Tin vui là chúng ta có thể dễ dàng bao gồm các cơ sở dữ liệu cục bộ trong ứng dụng di động của mình bằng SQLite (sqlite.org). SQLite là một công cụ cơ sở dữ liệu không có máy chủ, nhẹ, mã nguồn mở, giúp đơn giản hóa việc tạo cơ sở dữ liệu cục bộ và thực hiện các thao tác trên dữ liệu. Thông tin được lưu trữ bên trong các bảng và hoạt động dữ liệu có thể được thực hiện bằng cách viết mã C # và truy vấn LINQ. SQLite hoàn toàn phù hợp với sự phát triển đa nền tảng, bởi vì nó là một công cụ cơ sở dữ liệu di động. Trên thực tế, nó cũng được cài đặt sẵn trên cả iOS và Android và cũng có thể dễ dàng triển khai lên Windows. Vì lý do này, SQLite cũng là người bạn đồng hành hoàn hảo để xây dựng các ứng dụng di động đa nền tảng, tập trung dữ liệu với Xamarin.Forms cần cơ sở dữ liệu cục bộ.

### Cấu hình

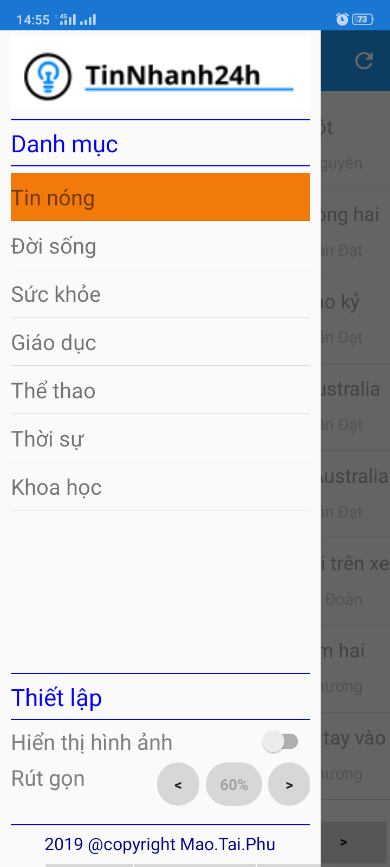


Hình 8 Các thư mục phía client

Trong hình 6, các phần có tác dụng như sau:

* Thư mục ***Models:***  chứa các khai báo các dữ liệu cho ứng dụng
* Thư mục ***Services:*** chứa các khai báo về interface giao tiếp API và giao tiếp cơ sở dữ liệu SQLite
* Thư mục ***ViewModels:*** các xử lý chính của ứng dụng
* Thư mục ***Views:*** Lưu trữ các giao diện và các xử lý render dữ liệu trên giao diện xaml

Giao diện ứng dụng cuối cùng:

## Phần Server:

### Đặc điểm

Đây là nơi nhận các yêu cầu từ phía client, xử lý và trả về dữ liệu phù hợp. hiện tại, phía server có các đặc điểm sau:

* Viết bằng ngôn ngữ python và flask framework nên cấu hình server nhỏ gọn, dễ bảo trì, phát triển.
* Quản lý cơ sở dữ liệu mongodb với 2 collection: 1 chứa danh sách các chủ đề và đường dẫn và 1 chứa danh sách các bài báo gồm nội dung, hình ảnh, tác giả, tiêu đề,…
* Khởi chạy sẵn mô hình rút gọn văn bản và thực hiện quá trình rút gọn khi được yêu cầu.
* Nhận các yêu cầu về lấy danh sách tin tức phân loại theo danh mục, dữ liệu tin tức bằng id.

### Công nghệ sử dụng

#### Flask Framework

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các API nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.

Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

[](https://i1.wp.com/nguyenvanhieu.vn/wp-content/uploads/2018/09/flask-python-la-gi-1.png?ssl=1)

Hình 9 Flask được sử dụng để xây dựng server nhanh và hiệu quả

* **Ưu điểm**
* Tốc độ
* Hỗ trợ cho NoQuery
* Độ phức tạp tối thiểu
* Chủ nghĩa tối giản tuyệt đối
* Không có ORM, dễ dàng kết nối với tiện ích mở rộng
* Trình gỡ lỗi được nhúng trong trình duyệt
* Mã ngắn và đơn giản trong số các bộ xương Python khác
* **Nhược điểm**
* Đôi khi phải tự bản thân làm nhiều việc hơn
* Tự mình gọi thêm các tiện ích mở rộng.

#### MongoDB:



Hình 10 Logo của mongoDB

MongoDB là một trong những cơ sở dữ liệu mã nguồn mở NoSQL phổ biến nhất được biết bằng C++. Tính đến tháng 2/2015, MongoDB được xếp thứ 4 trong số các hệ thống cơ sở dữ liệu phổ biến nhất. Nó được pahts triển bởi công ty 10gen sau này được biết đến với tên MongoDB Inc..

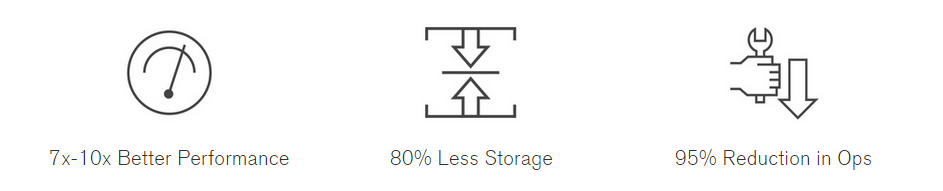
MongoDB là cơ sở dữ liệu hướng tài liệu, nó lưu trữ dữ liệu trong các document dạng JSON với schema động rất linh hoạt. Nghĩa là bạn có thể lưu các bản ghi mà không cần lo lắng về cấu trúc dữ liệu như là số trường, kiểu của trường lưu trữ. Tài liệu MongoDB tương tự như các đối tượng JSON.



Hình 11 Cấu trúc dữ liệu dạng document của mongoDB

* **Ưu điểm của MongoBD**
  + Document oriented
  + Hiệu suất cao
  + Tính sẵn sàng cao – Nhân rộng
  + Khả năng mở rộng cao – Sharding
  + Năng động – Không có lược đồ cứng nhắc.
  + Linh hoạt – thêm / xóa trường có ít hoặc không ảnh hưởng đến ứng dụng
  + Dữ liệu không đồng nhất
  + Không joins
  + Phân phối được
  + Biểu diễn dữ liệu trong JSON hoặc BSON
  + Hỗ trợ không gian địa lý (Geospatial)
  + Tích hợp dễ dàng với BigData Hadoop
  + Ngôn ngữ truy vấn dựa trên tài liệu mạnh mẽ như SQL
  + Các bản phân phối cloud như AWS, Microsoft, RedHat, dotCloud và SoftLayer, v.v … Trên thực tế, MongoDB được xây dựng cho cloud. Kiến trúc mở rộng quy mô tự nhiên của nó, được kích hoạt bởi sharding, liên kết tốt với quy mô và sự nhanh nhẹn có được nhờ điện toán đám mây.
* **Nhược điểm của MongoDB**
  + Một nhược điểm của NoSQL là hầu hết các giải pháp đều không tuân thủ ACID mạnh mẽ (Atomic, Consistency, Isolation, Durability) như các hệ thống RDBMS được thiết lập tốt hơn.
  + Giao dịch phức tạp
  + Không có chức năng hoặc thủ tục lưu trữ tồn tại nơi bạn có thể liên kết logic
* **Tại sao phải chọn MongoDB**

Như chúng ta đã biết, RDBMS lưu trữ dữ liệu trong các định dạng bảng và sử dụng ngôn ngữ truy vấn cấu trúc (SQL) để truy vấn cơ sở dữ liệu. RDBMS cũng định nghĩa sẵn các schema theo yêu cầu và tập hợp các quy tắc để định nghĩa quan hệ giữa các trường trong bảng.



Hình 12 Lựa chọn mongoDB

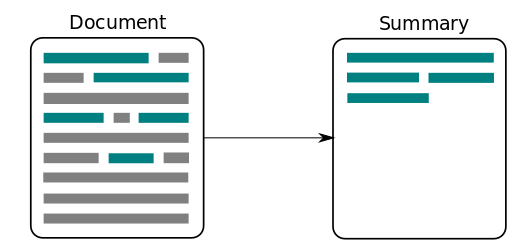
Nhưng MongoDB lưu trữ dữ liệu trong các tài liệu thay vì các bảng. Bạn có thể thay đổi cấu trúc các bản ghi (được gọi là document trong MongoDB) đơn giản chỉ cần thêm mới các trường hoặc xóa các trường có sẵn. Khả năng này của MongoDB giúp bạn trình bày các quan hệ dạng thứ bậc, để lưu trữ mảng, và các cấu trúc phức tạp khác một cách dễ dàng. MongoDB cung cấp hiệu suất cao, tính sẵn có cao và dễ dàng mở rộng.

#### Mô hình tóm tắt văn bản với Fasttext và Textrank

* **Áp dụng vào dự án:**

Ở đây, chúng em tạo một mô hình tóm tắt văn bản đơn giản sử dụng textrank trên một bộ pretrained word embedding. Đầu vào textrank là một danh sách các vector số tương ứng với mỗi câu trong đoạn tin tức, sau các quá trình tính toán trọng số các vector, đầu ra sẽ là danh sách các câu có trọng số cao nhất, có khả năng làm nội dung chính của đoạn văn. Độ tóm tắt văn bản ở đây là số lượng câu bị rút gọi (ở 60% có nghĩa là nếu bài báo có 100 câu thì sẽ bị bỏ đi 60 câu và chọn giữ lại 40 câu nội dung chính) có thể được điều chỉnh.

* + - * + **Textrank**:



Hình 13 Mô hình tóm tắt văn bản

Tóm tắt văn bản là một bài toán con trong lĩnh vực Xử Lý Ngôn Ngữ Tự Nhiên (Natural Language Processing) để tạo ra một văn bản ngắn nhưng chứa các ý chính của tài liệu tham khảo. Nó dường như là một nhiệm vụ bất khả thi nhưng nhờ sự phát triển của công nghệ nói chung và ở đây là Trí Tuệ Nhân Tạo (Arfitical Intelligent) nói riêng, ngày nay chúng ta có thể tạo ra một mô hình để thực hiện việc này.

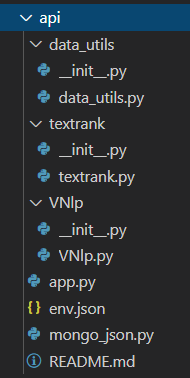
Textrank là một thuật toán cơ bản được sử dụng trong học máy để tóm tắt tài liệu. TextRank là một kỹ thuật tóm tắt văn bản theo phương pháp extractive và trong học máy thì là học không giám sát (Unsupervised Learning). TextRank không dựa trên bất kỳ dữ liệu đào tạo nào trước đó và có thể hoạt động với bất kỳ đoạn văn bản tùy ý nào.

* Ưu điểm:
* Tóm tắt giảm thiểu thời gian đọc.
* Khi đọc/nghiên cứu tài liệu, tóm tắt làm cho quá trình lựa chọn tài liệu trở nên dễ dàng hơn.
* Việc tự động tóm tắt văn bản cải thiện hiệu quả của việc lập chỉ mục.
* Thuật toán tóm tắt tự động ít sai lệch hơn so với việc tóm tắt của con người.
* Tóm tắt thông tin cá nhân rất hữu ích trong các hệ thống trả lời câu hỏi tự động vì chúng cung cấp thông tin cá nhân.
* Sử dụng các hệ thống tóm tắt tự động hoặc bán tự động cho phép các dịch vụ thương mại tăng số lượng tài liệu văn bản mà chúng có thể xử lý.
  + - * + **FastText:**

FastText là một thư viện nhẹ, miễn phí, mã nguồn mở cho phép người dùng tìm hiểu các biểu diễn văn bản và phân loại văn bản được phát triển bởi Facebook’s AI research lab. Nó hoạt động trên tiêu chuẩn, phần cứng chung. Các mô hình sau đó có thể được giảm kích thước để thậm chí phù hợp với các thiết bị di động.

FastText hiện có bộ pretrained world embedding đồ sộ hỗ trợ hơn 300 ngôn ngữ khác nhau trong đó có tiếng Việt

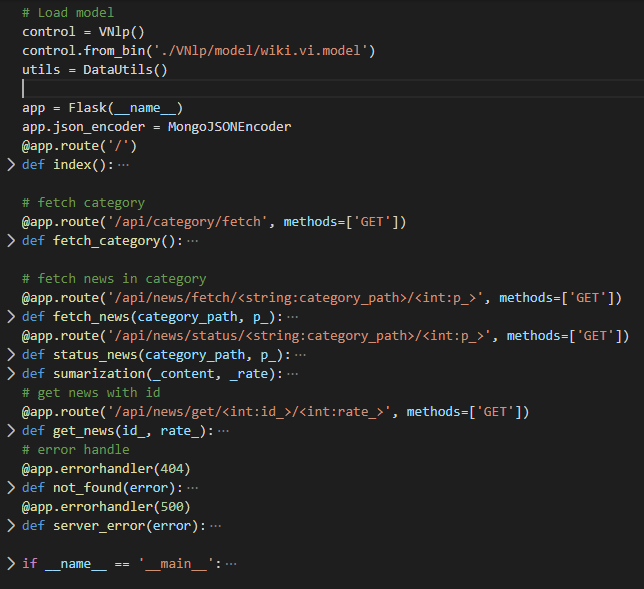
### Cấu hình



Hình 14 Cấu trúc server

Trong hình 14, có các thư mục sau:

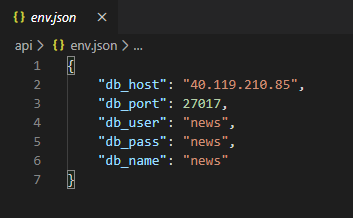
* Các thư viện gồm: **data\_utils**, **textrank, VNlp** phục vụ quá trình tóm tắt văn bản. **mongo\_json**.**py** là thư viện phục vụ xử lý, lưu trữ dạng JSON trên MongoDB.
* **app.py** cấu hình API và xử lý dữ liệu.



Hình 15 Cài đặt api trong file app.py

File app.py quy định các xử lý cho từng api path. **Ví dụ:** với path ***‘api/news/fetch/<tên chủ đề>/<trang hiển thị>’*** và phương thức là **GET**thì sẽ chạy hàm **fetch\_news()**

* **env.json** lưu cấu hình kết nối cơ sở dữ liệu.



Hình 16 Cấu hình kết nối cơ sở dữ liệu mongodb

## Phần Crawler

### Đặc điểm

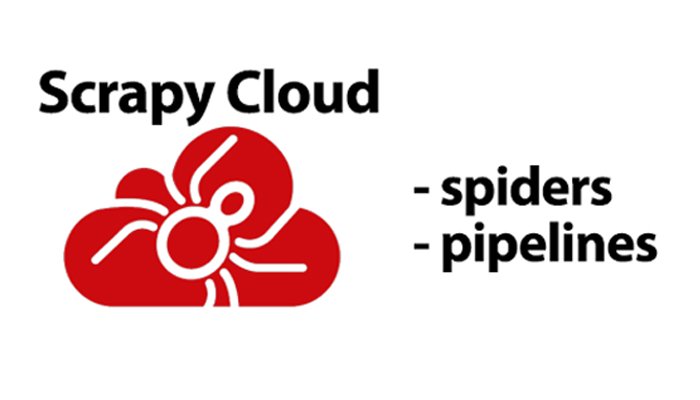
Đây là phần tìm kiếm và thêm dữ liệu bài báo vào cơ sở dữ liệu hệ thống:

* Sử dụng scrapy để viết một spider bot tự động gửi http requests liên tục đến trang vnexpress.net. Các responses nhận được sẽ được xử lý lấy nội dung bằng các xpath dựng sẵn.
* Theo lịch trình 6 tiếng một lần, spider bot được tự động khởi chạy 6 lần ở 6 danh mục tin tức.
* Dữ liệu thu được sẽ được kiểm tra trùng lặp rồi thêm vào cơ sở dữ liệu.

### Công nghệ sử dụng

#### Scrapy

Scrapy là một web framework rất mạnh mẽ trong việc trích xuất dữ liệu. Scrapy có kiến trúc sự kiện, cho phép chúng ta thực hiện các công việc như dọn dẹp, tạo, lưu trữ, làm giàu dữ diệu…v.v



Hình 17 Scrapy

Scrapy đã hiện diện từ năm 2008, một số tính năng của scrapy như sau:

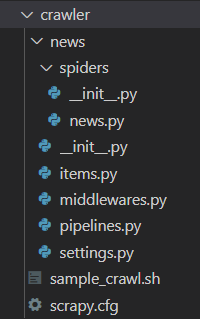
* Có thể làm việc với các đoạn code HTML “xấu”
* Cộng đồng lớn
* Mã nguồn ổn định
* Tốc độ xử lí nhanh theo dạng bất đồng bộ
* Có thể lưu dữ liệu theo nhiều định dạng như JSON, CSV, XML.
* Có thể trích xuất dữ liệu bằng cách sử dụng các biểu thức XPath hay CSS
* Ưu điểm:
* Scrapy dễ dàng mở rộng, nhanh chóng và mạnh mẽ.
* Nó là một khung ứng dụng đa nền tảng (Windows, Linux, Mac OS và BSD)
* Các http requests được lên lịch và xử lý bất đồng bộ.
* Scrapy đi kèm với dịch vụ tích hợp có tên Scrapyd cho phép tải lên các dự án và kiểm soát các spider bằng dịch vụ web JSON.
* Có thể crawl bất kỳ trang web nào, mặc dù trang web đó không có API để truy cập dữ liệu thô.
* Nhược điểm:
* Cấu trúc khá dài dòng, khó để kiểm tra và debug
* Cài đặt là khác nhau cho các hệ điều hành khác nhau.
* Tại sao nên sử dụng Scrapy:
* Nó dễ dàng hơn để xây dựng và quy mô các dự án thu thập thông tin lớn.
* Nó có một cơ chế tích hợp có tên là Selector, để trích xuất dữ liệu từ các trang web.
* Nó xử lý các yêu cầu không đồng bộ và nhanh chóng.
* Nó tự động điều chỉnh tốc độ thu thập thông tin bằng cơ chế Tự động điều chỉnh .
* Đảm bảo khả năng tiếp cận của nhà phát triển.

#### Cronjob:

Cron là một công cụ Linux dùng để lên kế hoạch cho một nhiệm vụ sẽ được thực thi trong tương lai. Thông thường, lệnh sẽ được sử dụng để lên lịch định kỳ cho một tác vụ - ví dụ, thực hiện gửi thông báo mỗi sáng. Một số tập lệnh, chẳng hạn như Drupal, WHMCS có thể sẽ yêu cầu bạn thiết lập một cron job để thực hiện một số chức năng nhất định.

* Ưu điểm:
* Nhờ Cron Jobs, bạn sẽ tiết kiệm được lượng lớn thời gian, không phải quản lý máy chủ lưu trữ và các tác vụ liên quan. Nếu là nhân viên văn phòng, bây giờ bạn hoàn toàn có thể về nhà, thư giãn sau một ngày miệt mài 8 tiếng trong văn phòng thay vì dành buổi tối của họ sao lưu các tập tin và quản lý địa chỉ liên lạc.
* Song song đó, bạn cũng không cần phải cố gắng ghi nhớ và tạo đi tạo lại những công việc định kì.
* Nhược điểm:
* Cron Jobs chỉ có thể thực hiện câu lệnh theo chu kỳ 1 phút trở lên, trong trường hợp muốn thực hiện các công việc lặp lại theo chu kỳ 1s, 5s, 10s, … thì CronTab sẽ không làm được.
* Các cấu hình crontab chỉ có thể sử dụng ở trên các máy tính khác nhau, khó có thể chia sẻ hoặc dùng chung. Cron rất khó để kiểm tra và sửa lỗi.

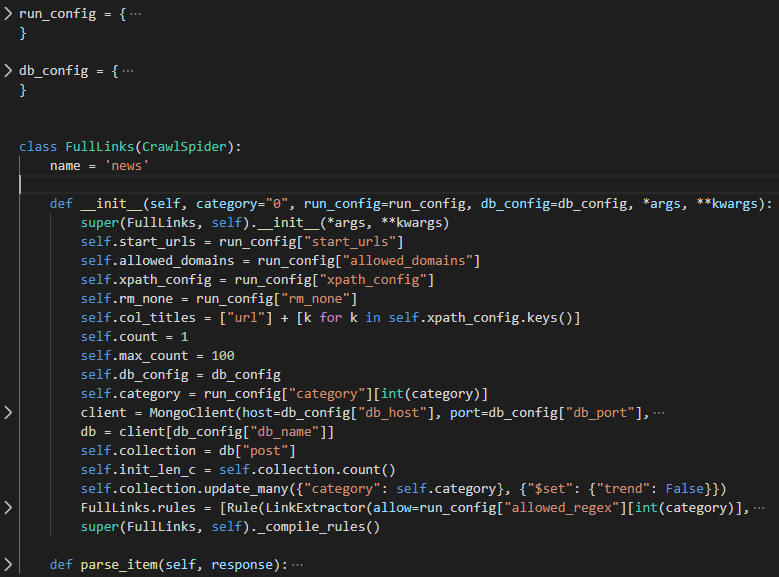
### Cấu hình



Hình 18 Cấu trúc scrawler

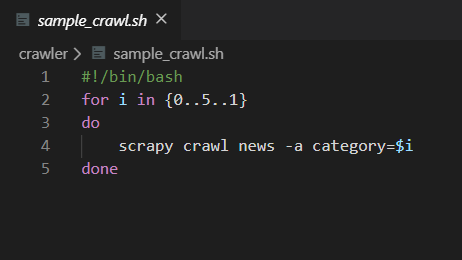
Trong hình 15, có các thư mục sau:

* **News/spiders/news.py** là nơi cấu hình Crawler bot.

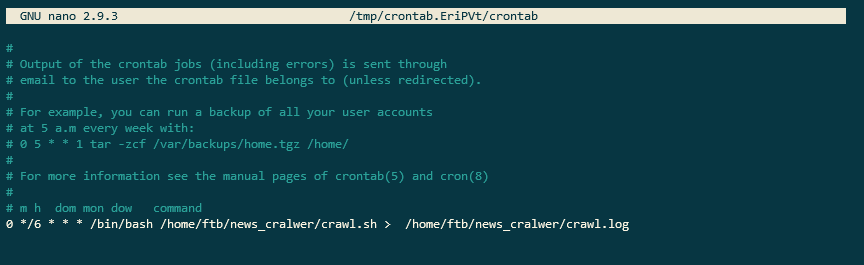


Hình 19 Cấu hình spider bot

* **sample\_crawl.sh** là file cấu hình chạy.



Hình 20 Cấu hình file khởi chạy scrapy



Hình 21 Cấu hình crontab chạy vào các giờ chia hết cho 6

# TỔNG KẾT

## Kết quả đạt được

Nhóm đã hoàn thành được một ứng dụng đọc tóm tắt nội dung tin tức với đầy đủ tính năng đề ra ban đầu:

* Cơ sở dữ liệu lưu trữ nội dung các bài báo được cập nhật liên tục theo chu kỳ 6 tiếng 1 lần ở trang vnexpress.net. mỗi lần thu thập khoảng 600 tin bài mới cũ tùy thực tế trên cả 6 đề tài (đời sống, thời sự, sức khỏe, giáo dục, thể thao, khoa học).
* Client xây dựng bằng Xamarin.Forms trên mô hình MVVM có giao diện thân thiện, dễ hiểu, dễ sử dụng đáp ứng đủ các yêu cầu cơ bản về đọc nội dung tóm tắt và bài viết gốc.
* Restful api server hoạt động và có khả năng xử lý được các yêu cầu từ phía client, mô hình máy học giải quyết bài toán rút gọn văn bản với độ chính xác khá ổn với nhu cầu cá nhân.

Ứng dụng có khả năng triển khai thực tế ở quy mô phù hợp về mặt pháp lý, giải quyết nhu cầu đọc tin nhanh của con người trong cuộc sống hiện đại.

## Cải thiện và phát triển

Ở giai đoạn hiện tại, hệ thống có thể đáp ứng những tác vụ cơ bản nhưng vẫn còn có nhiều vấn đề cần giải quyết, cải thiện:

* Xử lý các lỗi hệ thống hiện có
* Xây dựng giao diện đẹp hơn, nâng cao trải nghiệm sử dụng của người dùng
* Cập nhật thêm các nguồn dữ liệu khác ngoài vnexpress để đáp ứng đúng sự đa dạng bài viết, tốc độ cập nhật thông tin.
* Mô hình đọc hiểu cần được cải thiện để có được độ chính xác cao hơn.

Ngoài ra có những tính năng có thể phát triển thêm trong tương lai gần sẽ tạo ra được tác động tích cực:

* Tính năng lưu dữ liệu bài báo vào cơ sở dữ liệu SQLite hỗ trợ đọc khi không có kết nối internet
* Tính năng tìm kiếm và bộ lọc bài báo, lưu lại bài báo yêu thích và lịch sử đọc báo.
* Với các dữ liệu thu thập được từ các trang mạng, cần có biện pháp lọc kĩ càng hơn để có thể thu được những bài báo chất lượng có thông tin phù hợp.

## Kết luận, đánh giá

Kết thúc đồ án, nhóm em đã nắm được những kiến thức cơ bản về Xamarin.Forms, các bước để thực hiện và hoàn thành một dự án đa nền tảng trên các thiết bị di động với MVVM. Ứng dụng tạo ra có khả năng đáp ứng nhu cầu sử dụng thực tiễn.