**ĐẠI HỌC QUỐC GIA TP.HCM  
TRƯỜNG ĐH CÔNG NGHỆ THÔNG TIN**

**Lê Tuấn Khải  
Huỳnh Văn Khoa  
Lê Trần Nguyên Lam**

BÁO CÁO ĐỒ ÁN LẬP TRÌNH TRỰC QUAN

ĐỒ ÁN PHẦN MỀM HỖ TRỢ DU LỊCH  
GUIDE ME

**Giáo Viên Hướng Dẫn: Huỳnh Tuấn Anh**

TP. HỒ CHÍ MINH, NĂM 2017

Mục Lục

[Danh mục bảng biểu 4](#_Toc485138545)

[Chương 1. Hiện trạng và yêu cầu từ thực tế 5](#_Toc485138546)

[1.1. Hiện trạng vấn đề 5](#_Toc485138547)

[1.1.1. Vấn đề 5](#_Toc485138548)

[1.1.2. Phương hướng giải quyết 6](#_Toc485138549)

[1.2. Hiện trạng cơ sở vật chất và con người 6](#_Toc485138550)

[1.2.1. Tin học 6](#_Toc485138551)

[1.2.2. Con người 6](#_Toc485138552)

[1.3. Yêu cầu sơ bộ về phần mềm 7](#_Toc485138553)

[1.3.1. Tin học 7](#_Toc485138554)

[1.3.2. Con người 7](#_Toc485138555)

[Chương 2. Phân tích yêu cầu phần mềm và mô hình hóa 8](#_Toc485138556)

[2.1. Yêu cầu phần mềm 8](#_Toc485138558)

[2.1.1. Yêu cầu chức năng 8](#_Toc485138559)

[2.1.2. Yêu cầu phi chức năng 9](#_Toc485138560)

[2.1.3. Bảng FURPS 9](#_Toc485138561)

[2.2. Mô hình hóa 10](#_Toc485138562)

[2.2.1. Các trường hợp sử dụng thông thường 10](#_Toc485138563)

[2.2.2. Mô hình thực thể - mối quan hệ 12](#_Toc485138564)

[Chương 3. Thiết kế 13](#_Toc485138565)

[3.1. Thiết kế kiến trúc phần mềm 13](#_Toc485138567)

[3.2. Thiết kế dữ liệu 14](#_Toc485138568)

[3.2.1. Tổng quan 14](#_Toc485138569)

[3.3. Thiết kế giao diện và thành phần xử lí của giao diện 15](#_Toc485138570)

[3.3.1. Tổng thể về giao diện hệ thống 15](#_Toc485138571)

[3.3.2. Giao diện SplashScreen 16](#_Toc485138572)

[3.3.3. Giao diện Trang Chủ (Homepage) 16](#_Toc485138573)

[3.3.4. Giao diện trang thông tin địa điểm lớn 17](#_Toc485138574)

[3.3.5. Giao diện trang thông tin địa điểm cụ thể 18](#_Toc485138575)

[3.3.6. Giao diện bản đồ 20](#_Toc485138576)

[3.3.7. Giao diện trang thông tin của ghim vị trí 20](#_Toc485138577)

[3.3.8. Giao diện Journal Editor 21](#_Toc485138578)

[3.3.9. Giao diện trang cài đặt 22](#_Toc485138579)

[3.4. Thiết kế xử lý 23](#_Toc485138580)

[3.4.1. Geocoding và Reverse-Geocoding 23](#_Toc485138581)

[3.4.2. Thêm và xóa ghim bản đồ 23](#_Toc485138582)

[3.4.3. Load thông tin khu vực, thông tin từng địa điểm 25](#_Toc485138583)

[Chương 4. Cài Đặt Phần Mềm 27](#_Toc485138584)

[4.1. Tổng quan về công nghệ sử dụng 27](#_Toc485138586)

[4.1.1. Universal Windows Platform 27](#_Toc485138587)

[4.1.2. Ngôn ngữ lập trình C# 28](#_Toc485138588)

[4.1.3. Language-integrated query 28](#_Toc485138589)

[4.1.4. Một số framework được sử dụng 29](#_Toc485138590)

[4.2. Cài đặt phần mềm 30](#_Toc485138591)

[4.2.1. View 30](#_Toc485138592)

[4.2.2. View Model 31](#_Toc485138593)

[4.2.3. Model 31](#_Toc485138594)

[4.3. Vấn đề khi cài đặt 32](#_Toc485138595)

[4.3.1. Cài đặt giao diện 32](#_Toc485138596)

[4.3.2. Cài đặt lớp nghiệp vụ 32](#_Toc485138597)

[4.3.3. Cài đặt lớp truy cập cơ sở dữ liệu 32](#_Toc485138598)

[Chương 5. Kiểm Thử 34](#_Toc485138599)

[5.1. Unit Test 34](#_Toc485138601)

[5.2. Integration Test 34](#_Toc485138602)

[5.3. System Test 34](#_Toc485138603)

[5.3.1. Phương pháp kiểm tra 34](#_Toc485138604)

[Chương 6. Tổng Kết 35](#_Toc485138605)

[6.1. Tổng Kết 35](#_Toc485138607)

[6.2. Nhận Xét & Đánh Giá 35](#_Toc485138608)

[6.3. Phân Rã Công Việc 35](#_Toc485138609)

[TÀI LIỆU THAM KHẢO 38](#_Toc485138610)

# Danh mục bảng biểu

[Hình 1: Use case đọc thông tin 11](#_Toc485138611)

[Hình 2: Use case đọc tin tức 11](#_Toc485138612)

[Hình 3: Use case lên lịch du lịch 12](#_Toc485138613)

[Hình 4: Mô hình thực thể - mối quan hệ (bản lớn tại phụ lục) 12](#_Toc485138614)

[Hình 5: Hệ cơ sở dữ liệu của phần mềm 14](#_Toc485138615)

[Hình 6: Sơ đồ giao diện tổng thể 15](#_Toc485138616)

[Hình 7: Giao diện SplashScreen 16](#_Toc485138617)

[Hình 8: Giao diện trang chủ 16](#_Toc485138618)

[Hình 9: Giao diện giới thiệu tỉnh thành 17](#_Toc485138619)

[Hình 10: Giao diện giới thiệu tỉnh thành 17](#_Toc485138620)

[Hình 11: Giao diện trang giới thiệu địa điểm 18](#_Toc485138621)

[Hình 12: Giao diện trang giới thiệu địa điểm 18](#_Toc485138622)

[Hình 13: Giao diện trang giới thiệu địa điểm 19](#_Toc485138623)

[Hình 14: Giao diện bản đồ 20](#_Toc485138624)

[Hình 15: Giao diện thông tin vị trí ghim trên bản đồ 20](#_Toc485138625)

[Hình 16: Giao diện Journal Editor 21](#_Toc485138626)

[Hình 17: Giao diện trang cài đặt 22](#_Toc485138627)

[Hình 18: Activity Diagram thuật toán xóa một MapIcon khỏi danh sách 24](#_Toc485138628)

[Hình 19: Activity Dagram thuật toán thêm MapIcon 25](#_Toc485138629)

[Hình 20: Activity Diagram cho thuật toán load DataContext 26](#_Toc485138630)

[Hình 21: .NET Framework platform architecture Credit: msdn.microsoft.com 28](#_Toc485138631)

[Hình 22: Truy vấn LINQ được biên dịch thành ngôn ngữ dùng cho SQL Server 29](#_Toc485138632)

[Hình 23: Thao tác LINQ được biên dịch thành ngôn ngữ dùng cho SQL Server 29](#_Toc485138633)

[Hình 24: Các thành phần giao diện của phần mềm 31](#_Toc485138634)

[Hình 25: Các thành phần lớp trung gian View Model 31](#_Toc485138635)

[Hình 26: Các Class Model 32](#_Toc485138636)

# Hiện trạng và yêu cầu từ thực tế

## Hiện trạng vấn đề

### Vấn đề

Hiện tại, nền “công nghiệp không khói” của Việt Nam đang có nhiều dấu hiệu chững lại, trong khi tại các nước trong khu vực như: Thái Lan, Lào, Malasia, Philipines… thì lại có những bước phát triển vượt bậc, đặc biệt là Lào.

Nếu nói đến các tiềm năng du lịch, Việt Nam có lợi thế khá cao về thắng cảnh (rừng, bãi tắm, đảo...), nền văn hóa đậm đà bản sắc và nhiều công trình mang ý nghĩa lịch sử cao. Vậy nguyên nhân cho sự giậm chân tại chỗ của du lịch Việt Nam là gì?

Có lẽ nguyên nhân đầu tiên chính là sự yếu kém trong các dịch vụ ở các điểm du lịch dành cho du khách, ngoài ra việc quảng bá cho du lịch Việt cũng đang có một vấn đề khá khó để giải quyết. Bên cạnh đó, trở ngại về ngôn ngữ như một bức tường vô hình ngăn cản giữa du khách và người dân bản địa. Nhiều khi các du khách bị lạc đường hay muốn có những cuộc giao tiếp ngắn với người dân đều khá khó khăn. Tỷ lệ người dân biết và giao tiếp tốt bằng tiếng nước ngoài ở nước ta là thua xa các nước trong khu vực như: Thái, Singapore, Philipine hay Malaysia.

Bên cạnh các khó khăn liên quan đến ngôn ngữ thì việc các thông tin, trang web giới thiệu các địa điểm du lịch, vui chơi,…vẫn còn những hạn chế nhất định. Việc ứng dụng CNTT, công nghệ website nhằm phát triển du lịch không còn là một thứ quá mới mẻ đối với các cơ quan, doanh nghiệp du lịch **.**Website giúp các thông tin du lịch, các sản phẩm và dịch vụ du lịch đến với mọi khách hàng một cách nhanh chóng mà không có giới hạn về thời gian, không gian hay đối tượng. Song làm thế nào để khai thác được hết các tiện ích của website nhằm cung cấp thông tin một cách đầy đủ, kịp thời, chính xác cho người dùng tin du lịch là vấn đề đáng quan tâm. Tuy vậy, do việc quản lý các trang web này còn hạn chế, dẫn đến tình trạng bên cạnh một số ít trang có thiết kế giao diện khá ổn, nội dung đầy đủ và được cập nhật thường xuyên, thì có số một lượng lớn trang web được thiết kế quá cẩu thả, nội dung không hấp dẫn, thiếu sự nhất quán dẫn đến nhiều sự hiểu lầm không đáng. Việc thông tin trên các trang web về du lịch Việt Nam đã ít và thiếu mà độ chính xác lại không cao như vậy khiến cho việc du lịch của khách trong nước cũng gặp một số khó khăn chứ huống chi người ngoại quốc.

Bên cạnh đó, mức chi tiêu của du khách đến Việt Nam phần lớn là nằm ở mức hạn trung, lượng du khách hạng sang đến với Việt Nam là còn hơi ít so với những tiềm năng mà du lịch Việt Nam đang có.

Các điểm du lịch ở Việt Nam là rất nhiều, phân bố từ khắp Bắc – Trung – Nam, có thể kể đến các địa điểm thường được du khách nước ngoài lựa chọn để đến du lịch như: Vịnh Hạ Long ở Quảng Ninh (đây là một trong 7 kỳ quan thế giới mới; hệ thống hang động – khu nghỉ dưỡng Phong Nha – Kẻ Bàng ở Quảng Bình (nơi đây có hang Sơn Đoòng – là hang động lớn nhất thế giới từng được tìm thấy), phố cổ Hội An ở Quảng Nam, nơi đây đã nổi tiếng trên thế giới từ những thế kỷ XVI, XVII khi là khu giao lưu buôn bán giữa phong kiến Việt Nam với nhiều nước trên thế giới; đảo Phú Quốc ở tỉnh Kiên Giang – đây là hòn đảo lớn nhất Việt Nam và cũng lớn nhất nhất trong trong hệ thống đảo thuộc vịnh Thái Lan, hệ thống ruộng bậc thang Sa Pa và cao nguyên đá Đồng Văn; bãi biển Nha Trang ở Khánh Hòa – nằm trong top 50 bãi biển đẹp nhất thế giới, ngoài ra còn có các thành phố lớn như Hà Nội, Đà Nẵng, TP. Hồ Chí Minh hay du lịch sông nước Miền Tây Nam Bộ cũng thu hút rất đông du khách nước ngoài.

### Phương hướng giải quyết

Nếu có thể giải quyết tốt các vấn đề trên, sẽ rất nhanh thôi, Việt Nam sẽ vươn lên là một cường quốc du lịch. Nhưng muốn giải quyết một cách triệt để các vấn đề của nền du lịch Việt Nam là một việc rất khó, cần phải có một quá trình dài chung tay thực hiện của các cơ quan, doanh nghiệp trong ngành du lịch cũng như nhiều ngành khác trong xã hội. Nhà nước cũng phải có đường lối đúng đắn nhằm kích cầu ngành du lịch, nâng cao hiệu quả trong việc quản lý. Các doanh nghiệp cần phải thật sự nghiêm túc hơn trong việc làm du lịch, không vì những lợi ích nhỏ trước mắt mà làm mất đi những cơ hội lớn để phát triển sau này,phải chú trọng nâng cấp các dịch vụ kèm theo cho du khách với một mức phí vừa phải. Ngoài ra, cần phải tăng cường hơn nữa việc quảng bá du lịch Việt ở nước ngoài. Việc quản lý các trang web hay ứng dụng về du lịch cần hiệu quả hơn, các thông tin cần phải đầy đủ, hấp dẫn và có tính thống nhất giữa các nguồn cấp tin khác nhau.

Xem xét với khả năng hiện tại, nhóm dự định sẽ làm một phần mềm ứng dụng cung cấp những dịch vụ, trợ giúp, tìm kiếm những điểm du lịch trong nước và cung cấp thông tin chi tiết các nơi đó đến với khách du lịch một cách đơn giản, dễ hiểu và đầy đủ nhất. Giúp các chuyến du lịch của khách trong và ngoài nước dễ dàng hơn, đặc biệt là các chuyến đi “phượt” hay du lịch “bụi”.

## Hiện trạng cơ sở vật chất và con người

### Tin học

Hiện nay, việc sử dụng máy tính cá nhân hay điện thoại di động là rất phổ biến. Hầu như bất cứ lúc nào, du khách cũng mang theo bên mình 1 chiếc smartphone và laptop.

### Con người

Hiện nay, số lượng khách nước ngoài đến Việt Nam chiếm đến gần 50% là có thể sử dụng tiếng Anh trong giao tiếp, mà lượng người Việt Nam có thể giao tiếp bằng Tiếng Anh lại còn rất hạn chế. Tuy vậy, với xu hướng toàn cầu hóa hiện nay, ngành giáo dục đang đẩy nhanh việc phổ cập tiếng Anh đến với thế hệ trẻ, đây một tín hiệu rất tốt giúp du khách có thể giao tiếp thoải mái với người dân Việt Nam. Góp phần nâng cao hình ảnh người dân nước ta trong lòng bạn bè quốc tế. Và thế, du lịch cũng có thêm điều kiện để phát triển.

## Yêu cầu sơ bộ về phần mềm

### Tin học

Phần mềm chạy được trên những máy tính có cấu hình từ trung bình trở lên, và không yêu cầu quá cao về phần cứng.

Một cấu hình mẫu:

* Processor: Intel Core i7-5500U, 2.40GHz (4CPUs),
* Memory: 8192MB RAM,
* Operating System: Windows 10 Home,
* VGA: AMD Radeon R7 M360 (Total memory: 6088MB)

### Con người

* Ngôn ngữ chính: tiếng Anh,
* Giao diện: đẹp, quen thuộc và dễ sử dụng,
* Độ phức tạp: thấp, dễ học.

# Phân tích yêu cầu phần mềm và mô hình hóa



## Yêu cầu phần mềm

### Yêu cầu chức năng

* Bảng tổng hợp và định dang các yêu cầu:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Định danh*** | | ***Độ Ưu tiên*** | ***Mô tả Yêu Cầu*** |
| YC\_1 | Read\_Info | 5 | Người dùng đọc thông tin về các địa điểm du lịch. |
| YC\_2 | Live\_Map | 5 | Cung cấp bản đồ cho người dùng. |
| YC\_3 | Make\_Plans | 4 | Người dùng tạo bản kế hoạch du lịch. |
| YC\_4 | Show\_Me\_Around | 4 | Hệ thống tự động giới thiệu địa điểm du lịch cho người dùng |
| YC\_5 | Search | 4 | Tìm kiếm trong database của phần mềm, hoặc online. |
| YC\_6 | Basic\_Articles | 4 | Cung cấp thông tin, hướng dẫn chung khi đi du lịch. |
| YC\_7 | Use\_Templates | 3 | Cung cấp templates có sẵn để người dùng tạo kế hoạch. |
| YC\_8 | Export\_Info | 3 | Hệ thống xuất ra các thông tin người dùng cần ở các định dạng thông dụng khác nhau. |
| YC\_9 | Visited\_Check | 2 | Người dùng đánh dấu những địa danh họ đã đi qua. |
| YC\_10 | Interest\_Check | 2 | Nếu người dùng thích một vị trí, địa danh nào đó, họ sẽ chỉ ra và phần mềm lưu lại. |
| YC\_11 | Take\_Notes | 2 | Người dùng có thể thêm ghi chú và lưu lại. |
| YC\_12 | Update\_Events | 2 | Thông tin một số sự kiện tại các thành phố lớn. |
| YC\_13 | RSS\_Reader | 1 | Đọc tin tức từ các trang báo chí lớn. |
| YC\_14 | Review\_Location | 3 | Nhận xét, ý kiến về một địa điểm |

Bảng 1: Bảng tổng hợp yêu cầu chức năng

### Yêu cầu phi chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| ***Định danh*** | | ***Độ Ưu tiên*** | ***Mô tả Yêu Cầu*** |
| YC\_A | Chống biên dịch ngược/ reverse engineering | 1 | Sử dụng các biện pháp bảo vệ mã nguồn. |
| YC\_B | Lưu vết người dùng (logging user activity) | 1 | Sử dụng hệ thống lưu vết người dùng nhằm phát hiện lỗi, nghiên cứu về sở thích người dùng… |
| YC\_C |  |  |  |
| YC\_D |  |  |  |

Bảng 2: Bảng tổng hợp yêu cầu phi chức năng

### Bảng FURPS

|  |  |
| --- | --- |
| ***Tiêu chí chất lượng*** | ***Mô tả*** |
| Functionality | * Hệ thống hướng tới phục vụ người dùng đơn lẻ. |
| Usability | * Cung cấp kèm theo tài liệu và hướng dẫn từng trang và chức năng của hệ thống. * Giao diện được thiết kế bắt mắt, đơn giản, dễ hiểu, và dễ sử dụng. * Chức năng lên lịch du lịch hỗ trợ WYSIWIG editor và hệ thống template đẹp, bắt mắt. |
| Reliability | * Hệ thống được thiết kế kỹ với các phương thức rào bắt lỗi, giảm thiểu khả năng crash. * Hệ thống lấy thông tin từ các nguồn đáng tin cậy và sẽ cập nhật thường xuyên. |
| Performance | * Các thuật toán sẽ được tối ưu hóa, giảm thiểu thời gian phản hồi (response time.) * Công việc tra cứu thông tin sẽ nhanh hơn, hiệu quả hơn vì các thông tin sẽ được tập trung lại tại một giao diện. |
| Supportability | * Thiết kế có tính tiến hóa cao, có nhiều phương hướng để phát triển. * Database được thiết kế ở dạng chuẩn cao, dễ tương tác và nâng cấp. * Cung cấp chức năng thông báo lỗi đến nhà sản xuất cho người dùng. * Tích hợp tính năng lưu vết người dùng giúp kiểm tra và phát hiện lỗi dễ dàng hơn. Từ đó dẫn đến việc sửa lỗi sẽ nhanh hơn. |

Bảng 3: Các tiêu chí FURPS

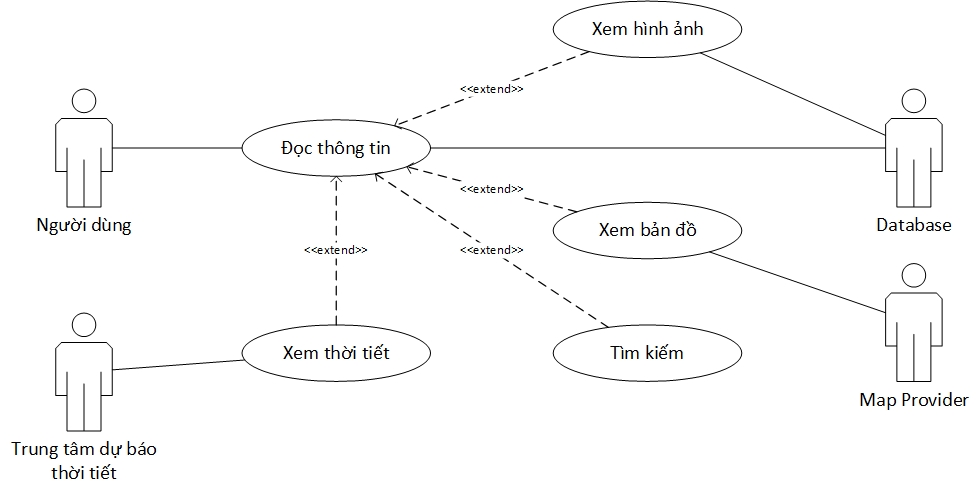
## Mô hình hóa

### Các trường hợp sử dụng thông thường

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Tên*** | ***Mô tả*** | ***Yêu cầu liên quan*** |
| UC-1 | Đọc thông tin | Đọc bài viết, blog, hình ảnh về một địa điểm. | Read\_Info, Live\_Map, Search |
| UC-2 | Đọc tin tức | Đọc tin tức mới, đọc các bài báo, đọc thông tin về địa điểm. | Read\_Info, Live\_Map, RSS\_Reader |
| UC-3 | Lên lịch du lịch | Lên lịch trước khi đi du lịch, có thời gian, địa điểm sẽ đến dự kiến… | Make\_Plans, Read\_Info, Static\_Map, Search |
| UC-4 | Xem quick tour | Nhờ phần mềm tự giới thiệu sơ qua về Việt Nam | Show\_Me\_Around, Read\_Info |
| UC-5 | Xem bản đồ | Xem bản đồ khu vực Việt Nam | Live\_Map, Static\_Map |
| UC-6 | Đánh giá | Đánh giá, nhận xét về địa điểm du lịch | Review\_Location |

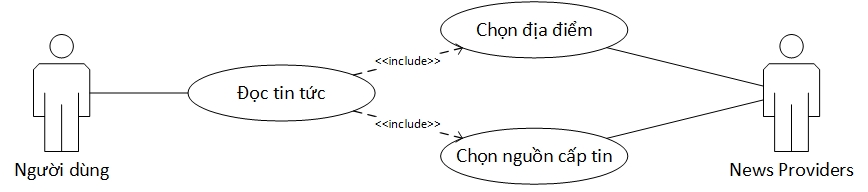
Bảng 4: Bảng tổng hợp các use-case thông dụng

#### Use Case 1 (Đọc thông tin):

****

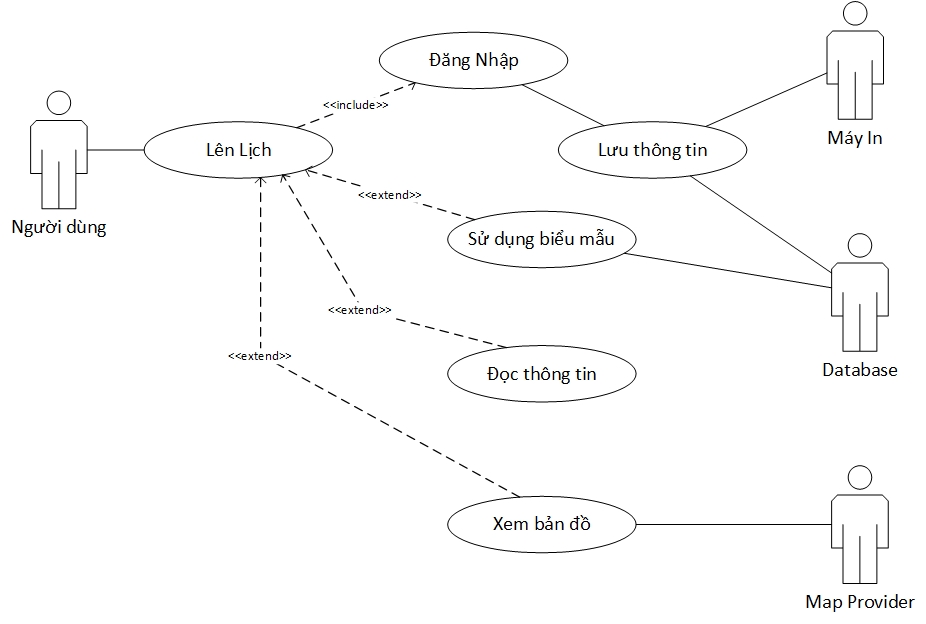
Hình 1: Use case đọc thông tin

#### Use Case 2 (Đọc tin tức)



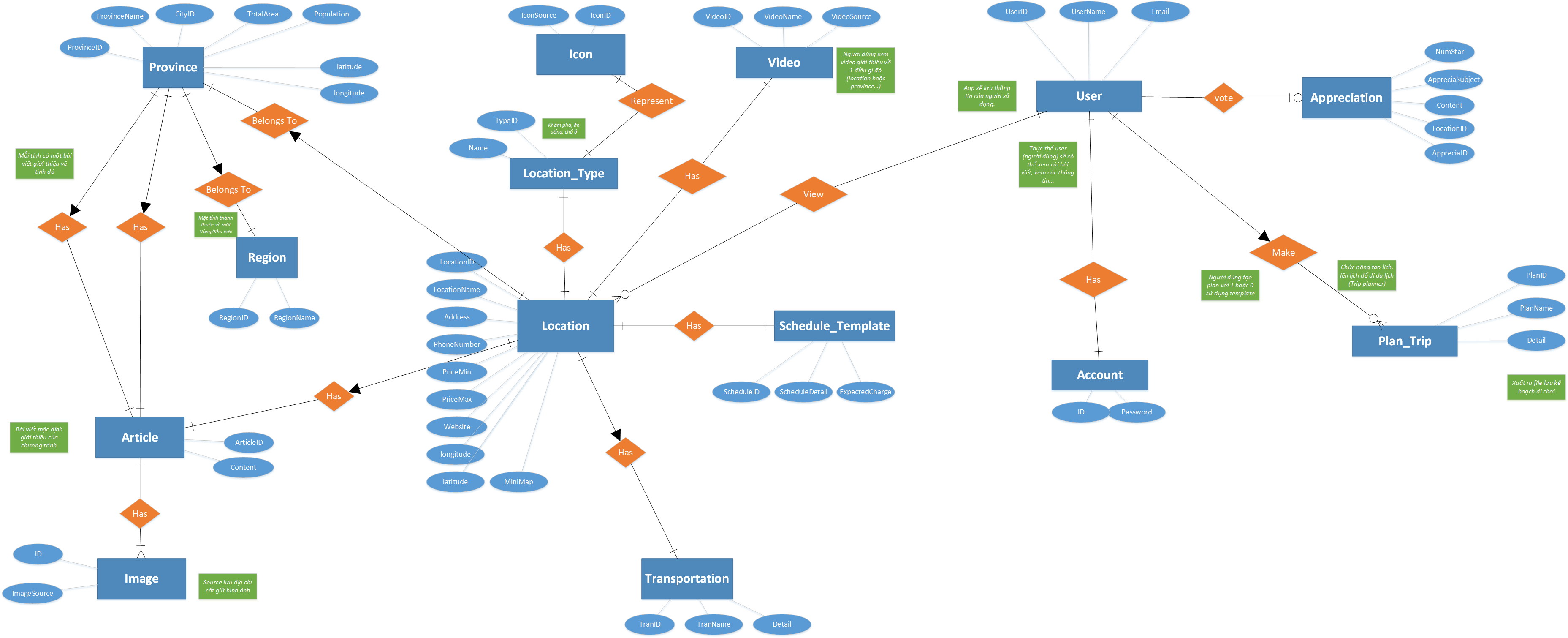
Hình 2: Use case đọc tin tức

#### Use Case 3 (Lên lịch)



Hình 3: Use case lên lịch du lịch

### Mô hình thực thể - mối quan hệ



Hình 4: Mô hình thực thể - mối quan hệ (bản lớn tại phụ lục)

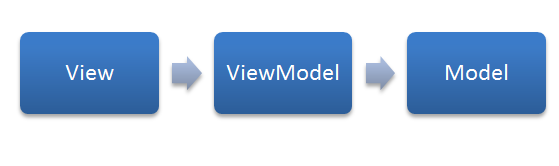
# Thiết kế



## Thiết kế kiến trúc phần mềm

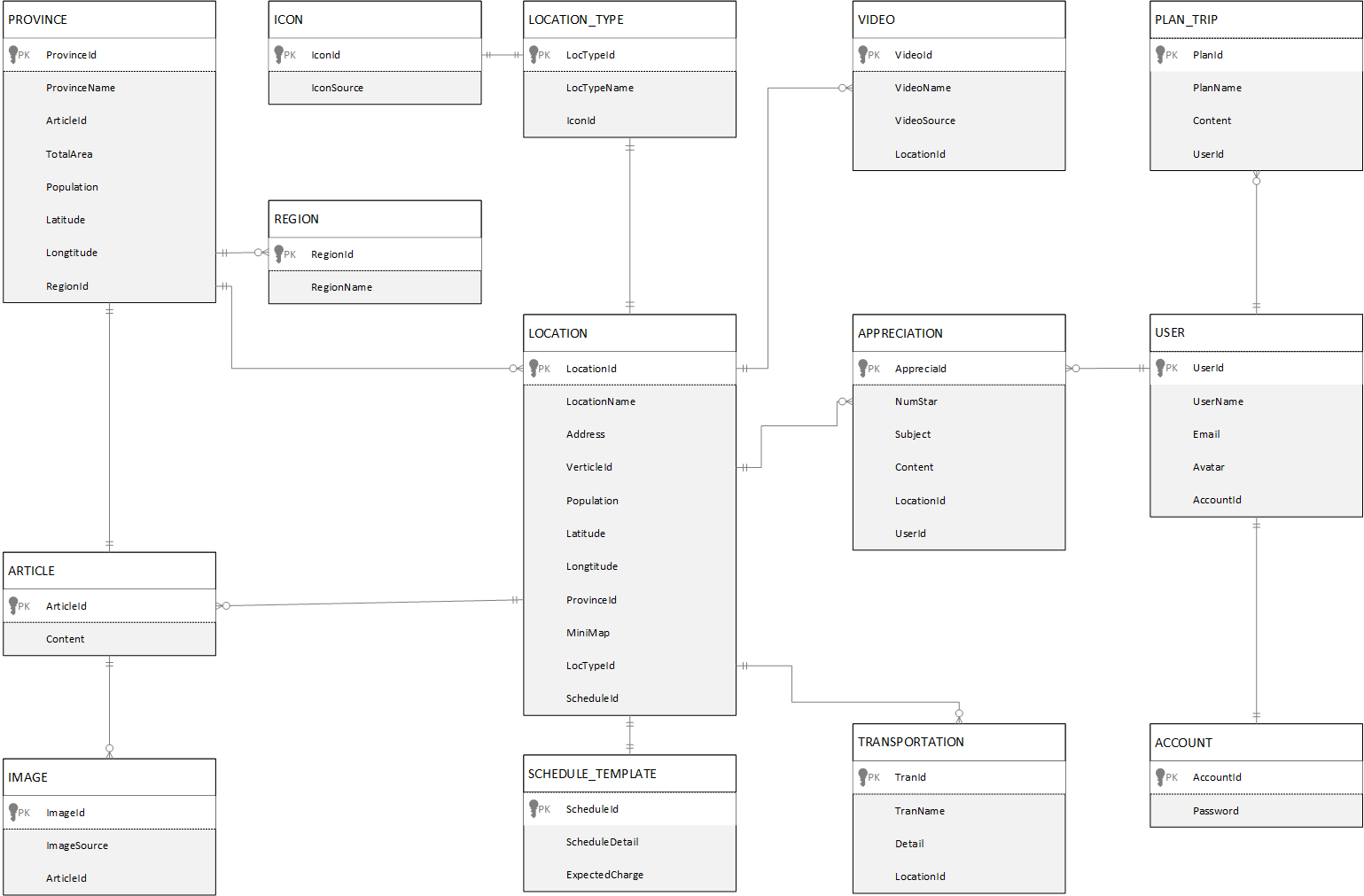
Phần mềm sẽ được thiết kế theo mô hình kiến trúc MVVM:

* View: Là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.
* View Model: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.
* Model: Là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.



## Thiết kế dữ liệu

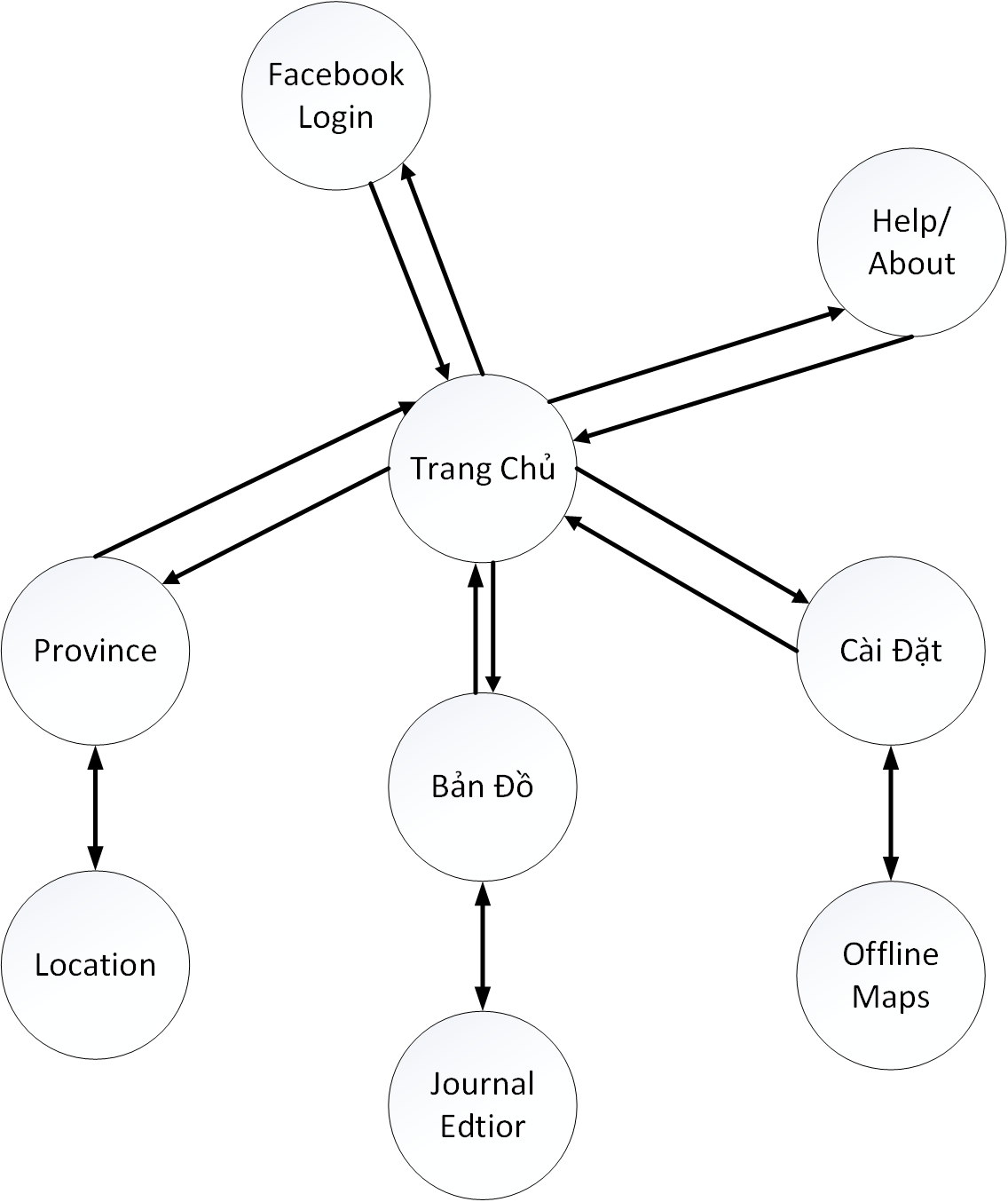
### Tổng quan



Hình 5: Hệ cơ sở dữ liệu của phần mềm

## Thiết kế giao diện và thành phần xử lí của giao diện

### Tổng thể về giao diện hệ thống



Hình 6: Sơ đồ giao diện tổng thể

Trang chủ ngoài các thành phần thông thường thì nó còn có một thành phần hoạt động tương tự như một frame để host toàn bộ các trang khác. Từ trang chủ, người dùng có thể đi tới trang đăng nhập Facebook, trang thông tin các vùng miền (Province), trang bản đồ (MapView) và trang cài đặt hệ thống.

Mỗi vùng miền khác nhau sẽ được host trên cùng một template trang Province. Mỗi vùng có nhiều địa diểm (Location) khác nhau cũng được host với cơ chế tương tự.

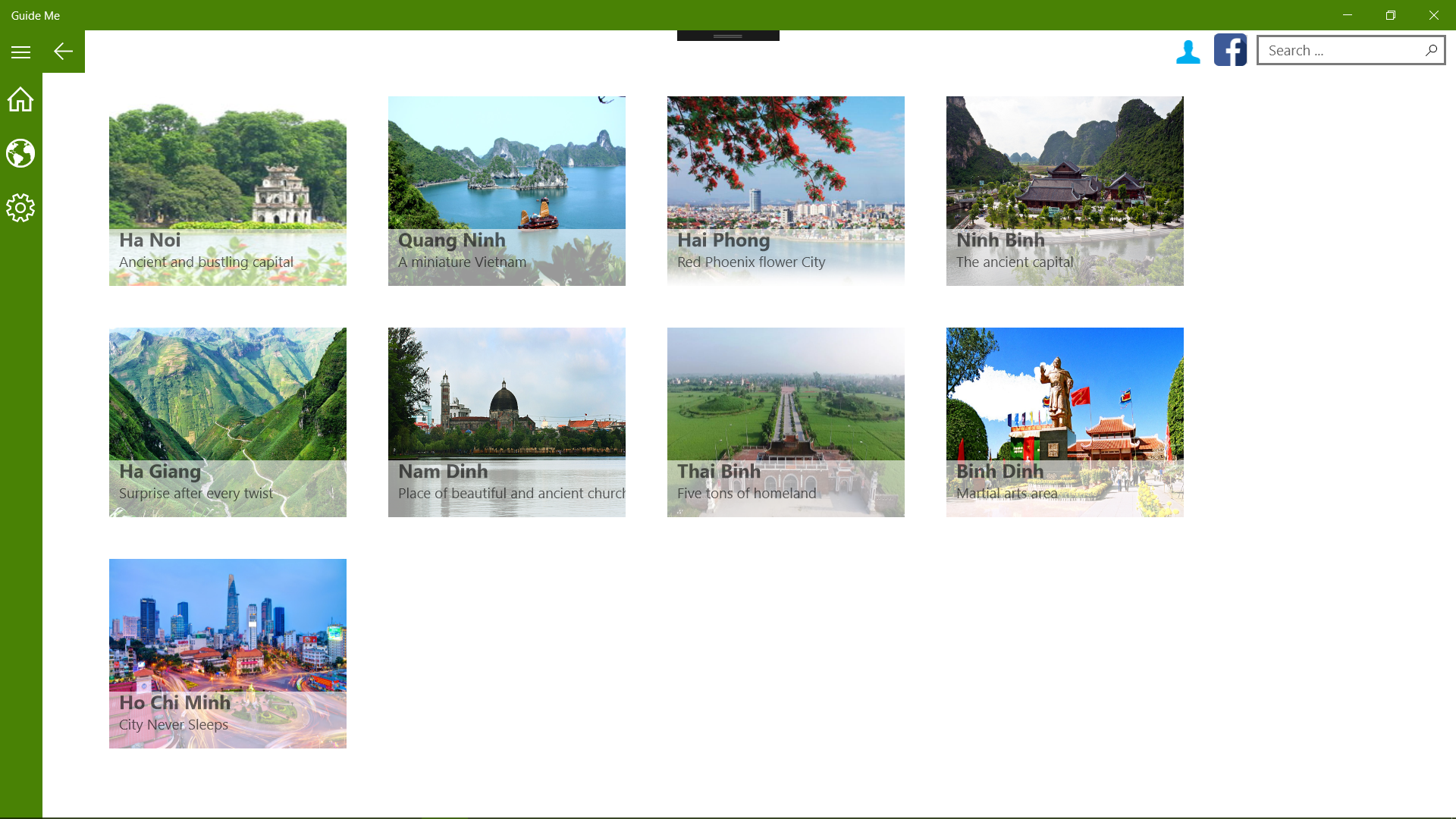
Giao diện Hỗ trợ (Help) được thiết kế riêng biệt so với phần mềm (nhằm tránh gây ảnh hưởng đến trải nghiệm người dùng) và chỉ được gọi lên khi người dùng nhấn phím F1.

### Giao diện SplashScreen



Hình 7: Giao diện SplashScreen

### Giao diện Trang Chủ (Homepage)



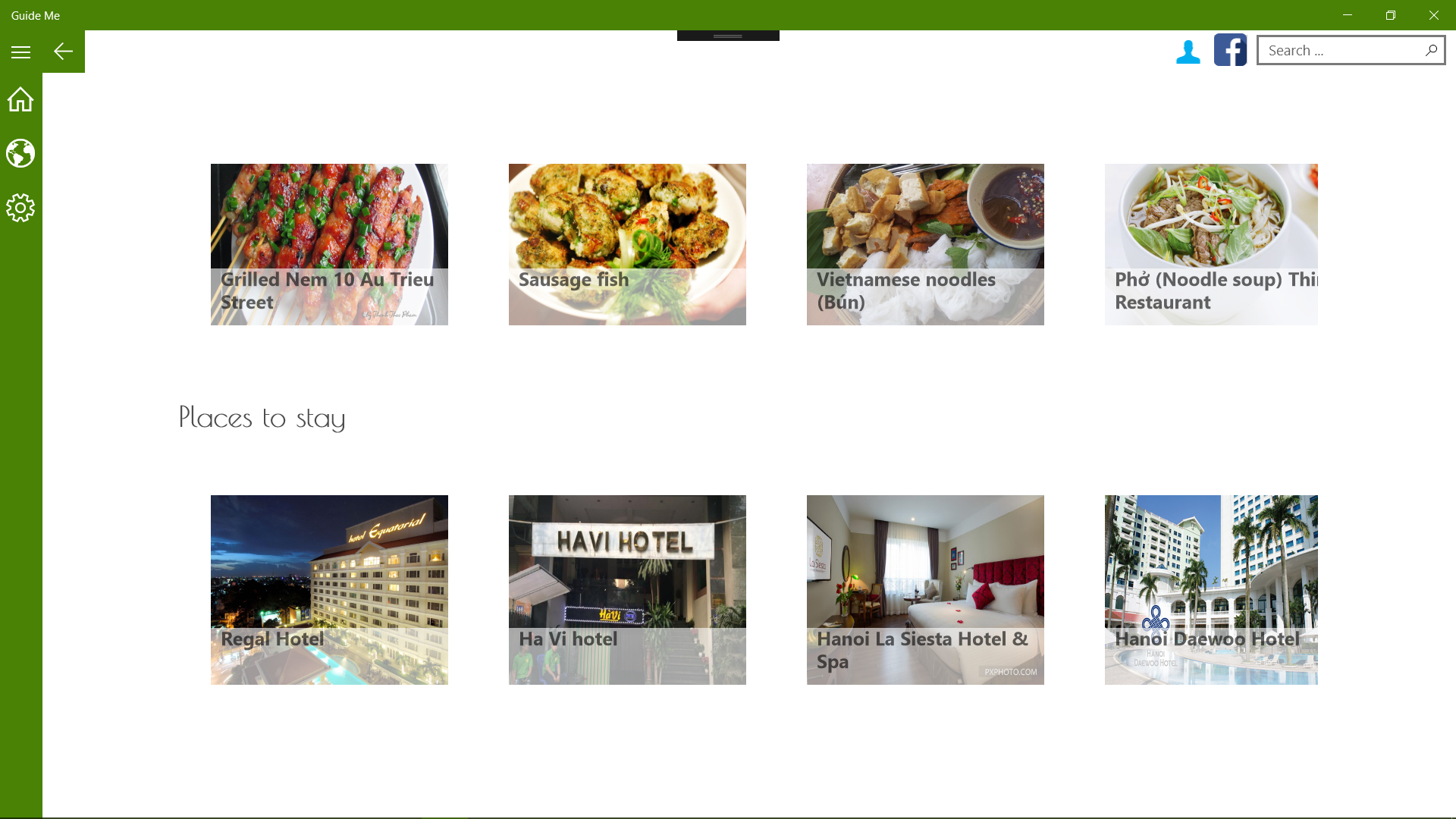
Hình 8: Giao diện trang chủ

Mô tả thành phần giao diện: Bao gồm một ListView lớn với Item con là các trang giới thiệu về từng vùng miền trong database. Mỗi Item con bao gồm tên của địa điểm, dòng giới thiệu ngắn gọn và hình ảnh của địa điểm ấy.

### Giao diện trang thông tin địa điểm lớn

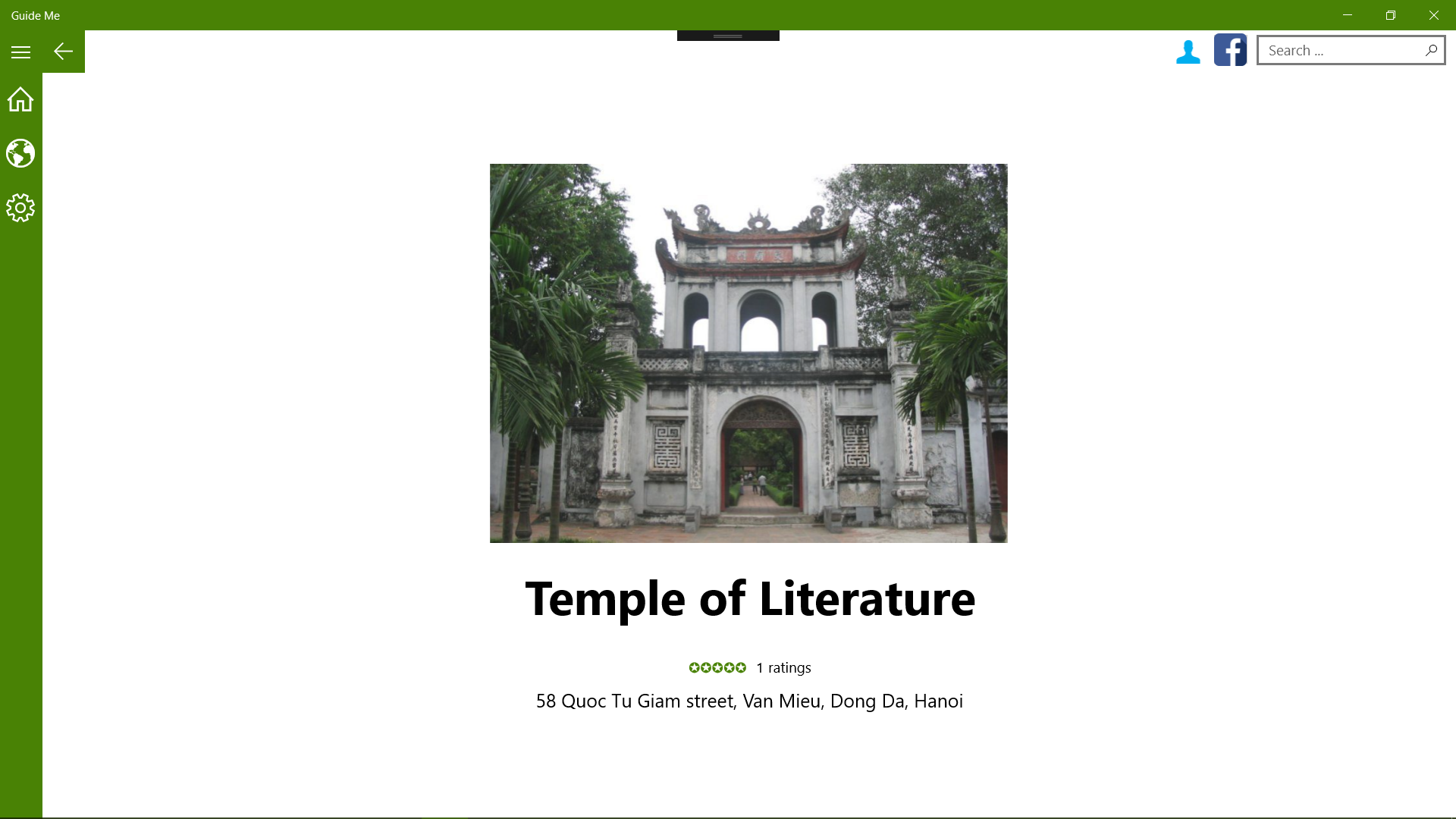


Hình 9: Giao diện giới thiệu tỉnh thành

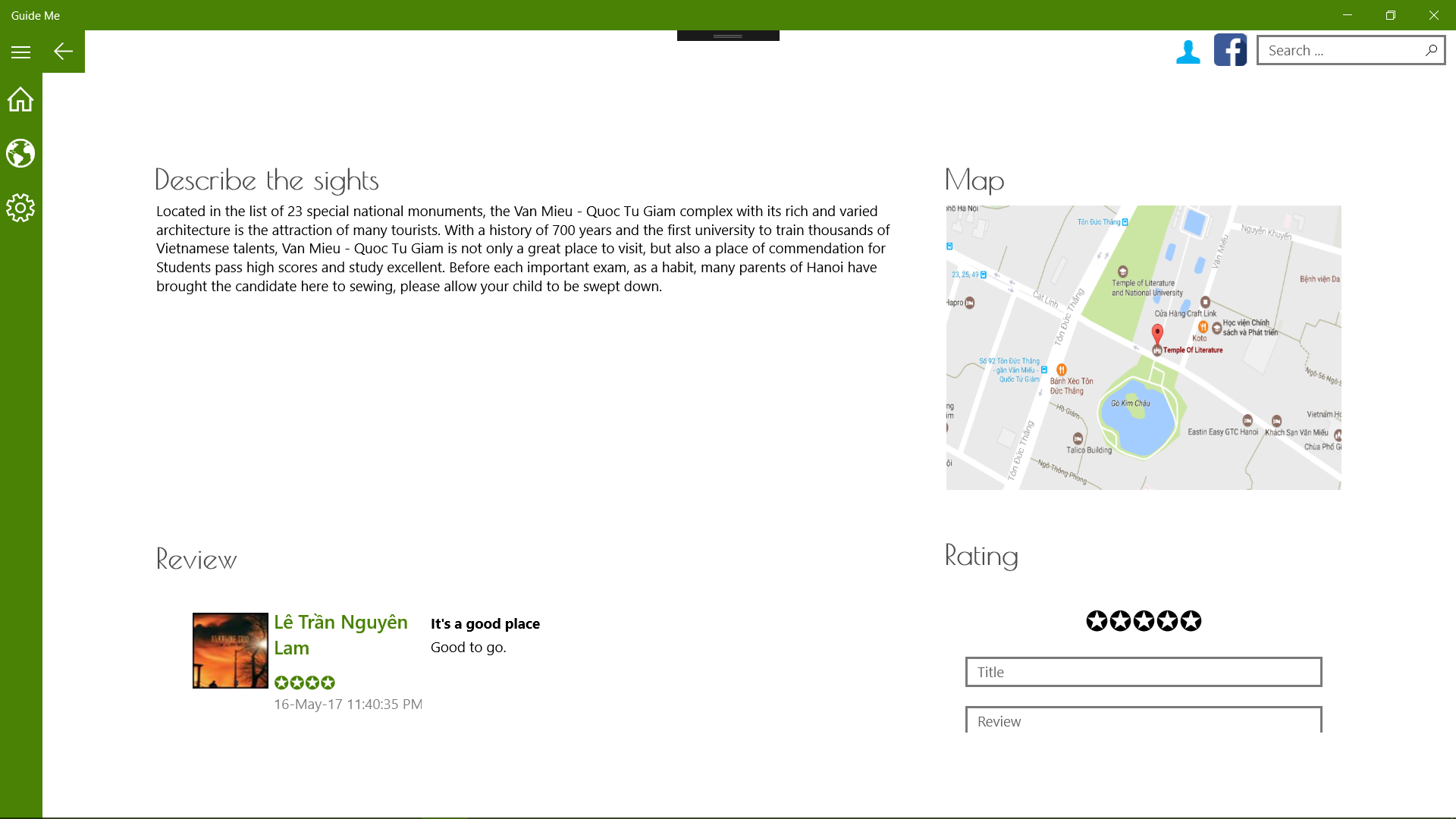


Hình 10: Giao diện giới thiệu tỉnh thành

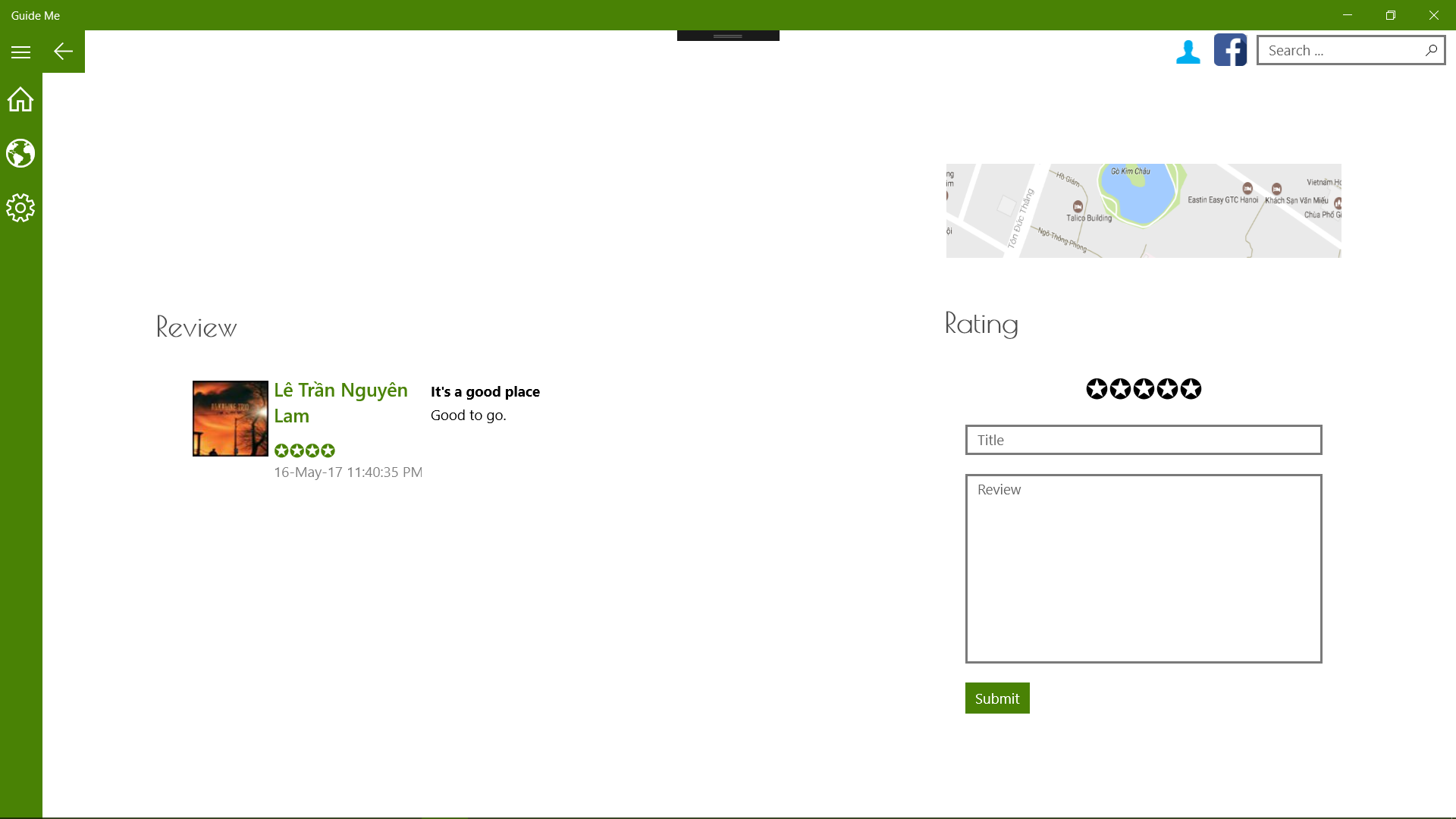
### Giao diện trang thông tin địa điểm cụ thể



Hình 11: Giao diện trang giới thiệu địa điểm



Hình 12: Giao diện trang giới thiệu địa điểm

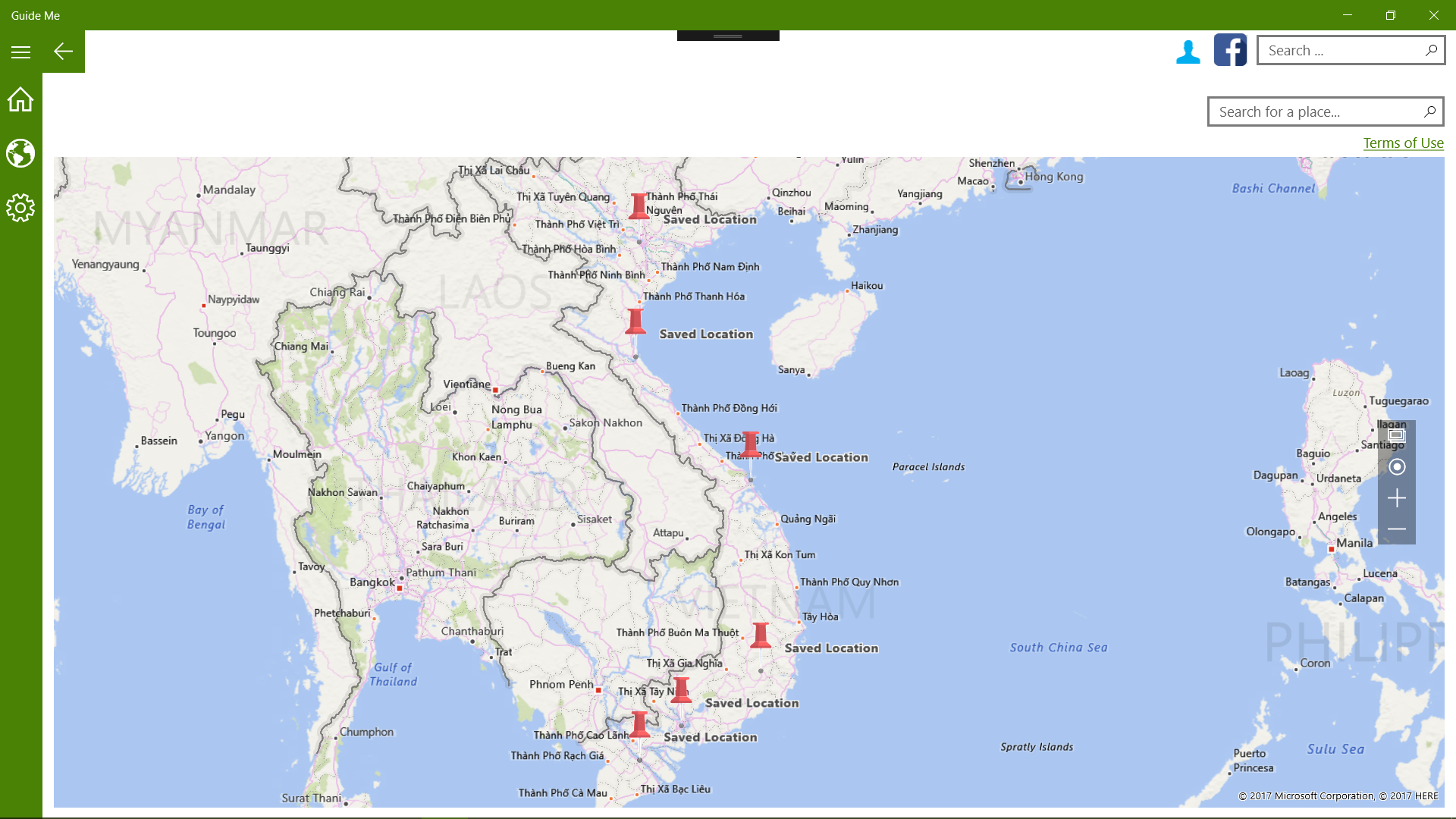


Hình 13: Giao diện trang giới thiệu địa điểm

Mô tả thành phần giao diện: Trang bao gồm các thành phần như sau:

* Phần hình ảnh và header: Một trang sẽ mở đầu bằng các hình ảnh (được đặt trong cùng một khung và sẽ tự động chuyển ảnh sau vài giây). Tiếp đến là vị trí, địa chỉ và đánh giá của người dùng.
* Phần mô tả: Gồm có phần mô tả bằng lời văn và phần bản đồ. Bản đồ có thể di chuyển, thu nhỏ và phóng to. Ngoài ra, nếu click đôi vào hình ảnh, người dùng sẽ trực tiếp chuyển đến địa điểm đó trên bản đồ lớn.
* Phần đánh giá: Một bên là các đánh giá đã có từ hệ thống. Một bên sẽ là nơi để người dùng chia sẻ đánh giá, trải nghiệm của mình về địa điểm.

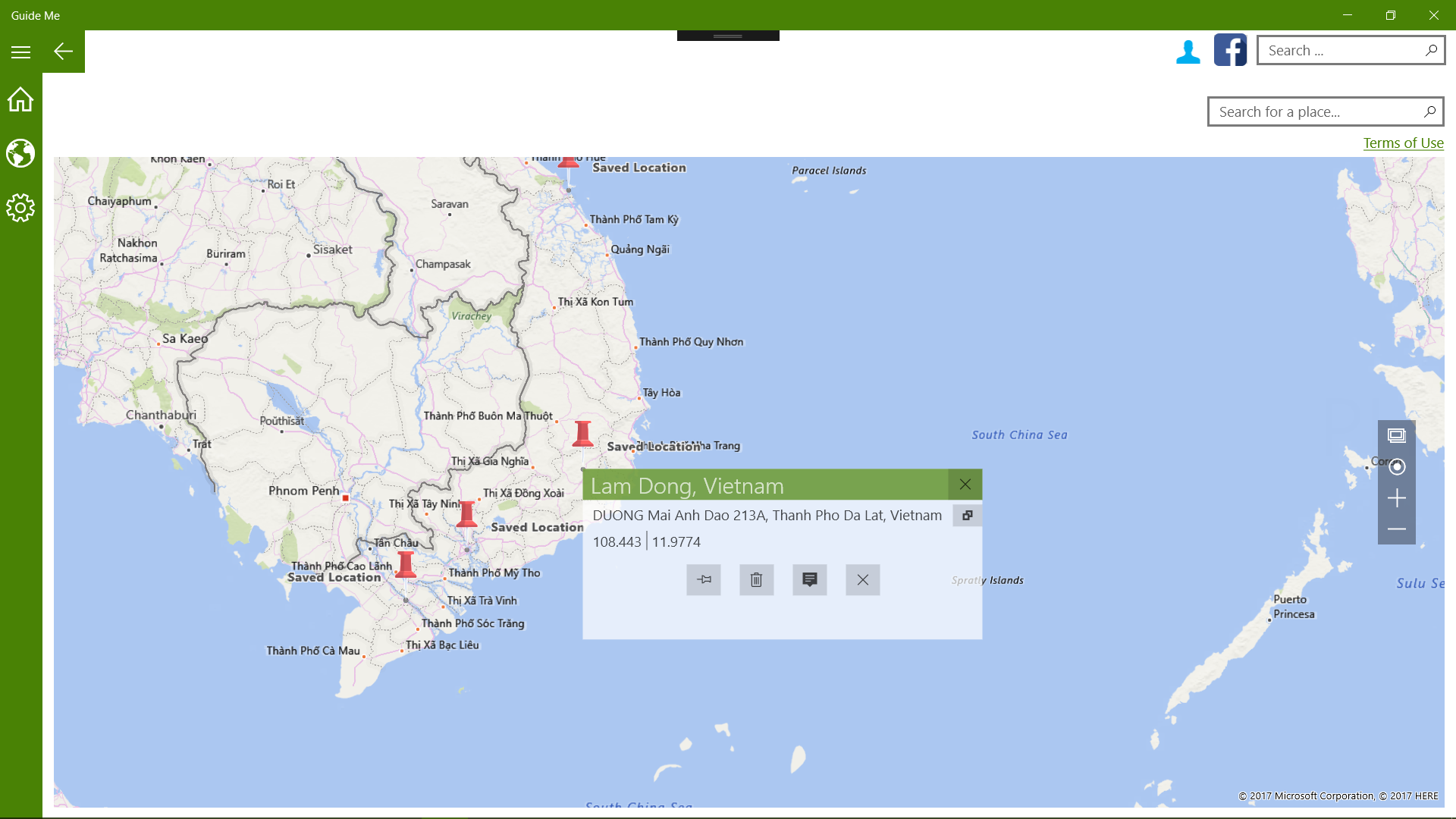
### Giao diện bản đồ



Hình 14: Giao diện bản đồ

Mô tả thành phần giao diện: Bao gồm một Map Control và một thanh công cụ (StackPanel) nằm trên bản đồ bao gồm các chức năng như thu nhỏ, phóng to, và tìm địa điểm hiện tại của người dùng. Ngoài ra, nếu click phải vào bản đồ thì phần mềm sẽ hiện ra các thao tác khác như load địa điểm đã lưu hay đặt ghim một vị trí nào đó.

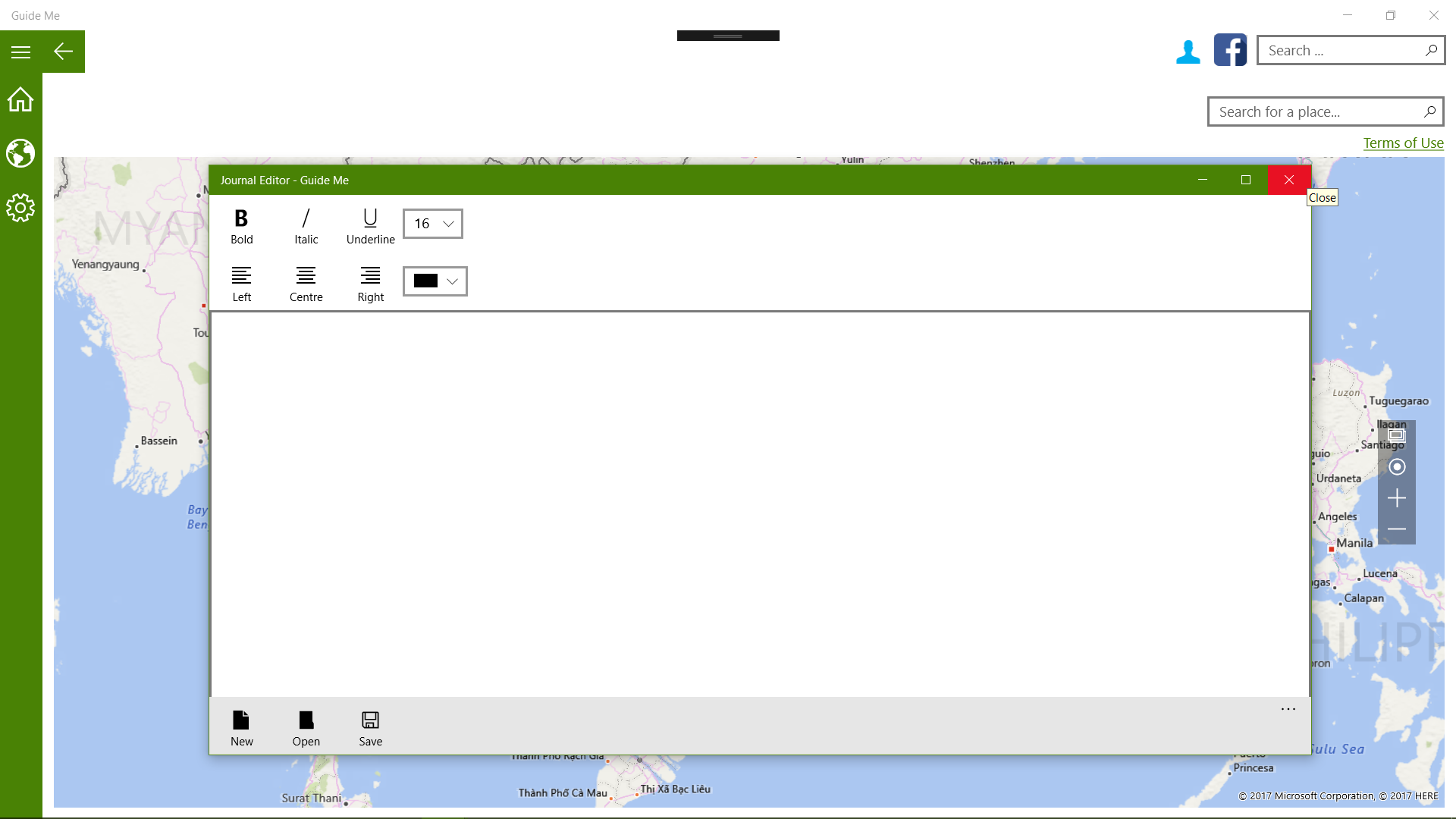
### Giao diện trang thông tin của ghim vị trí



Hình 15: Giao diện thông tin vị trí ghim trên bản đồ

Mô tả thành phần giao diện:

### Giao diện Journal Editor

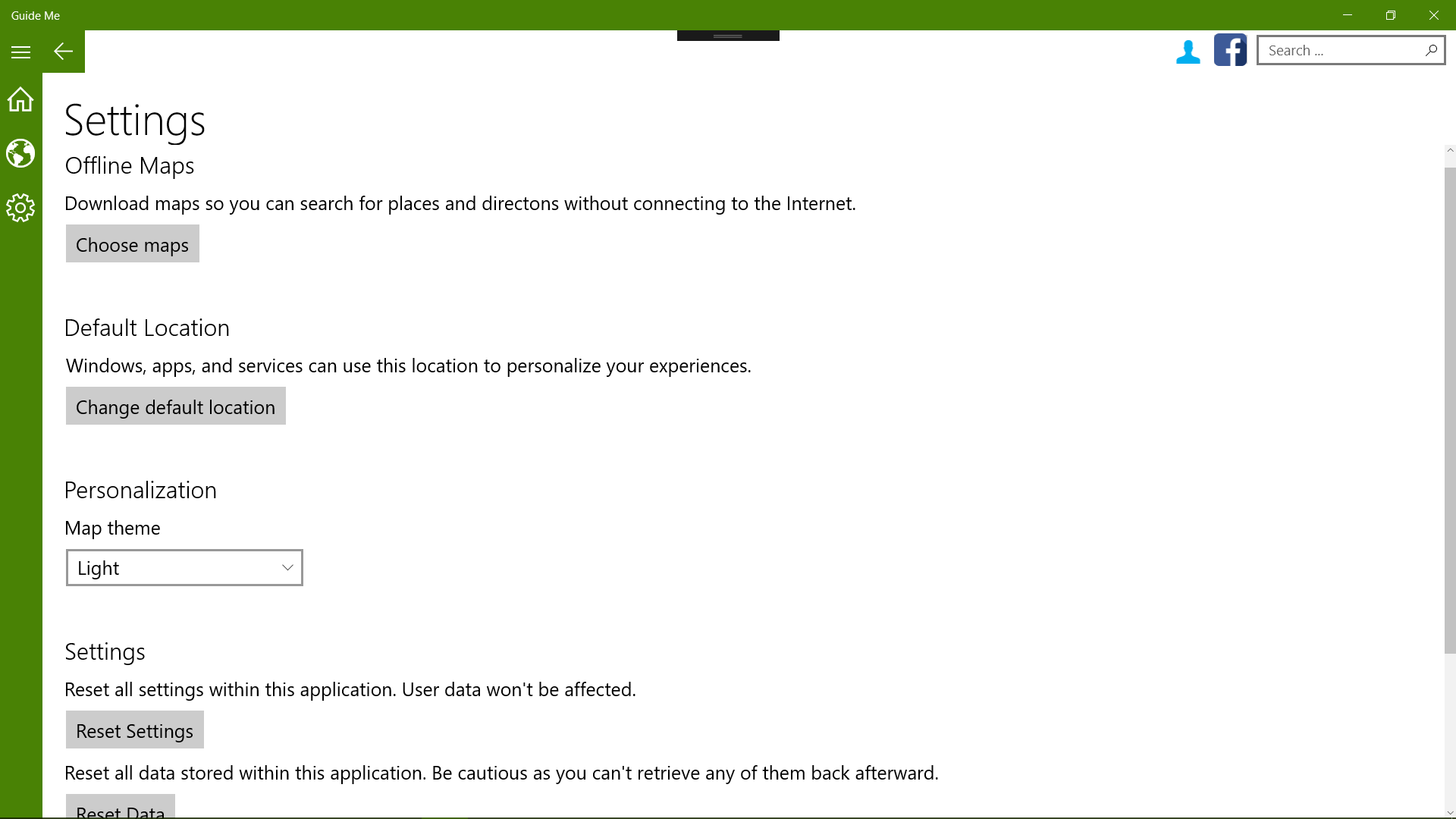


Hình 16: Giao diện Journal Editor

Mô tả thành phần giao diện: Bao gồm 1 khung RichTextBox và 3 phần định dạng nằm trong StackPanel theo thứ tự từ trên xuống:

* Định dạng kiểu chữ: Bao gồm các ToggleButton và ComboBox cho phép người dùng định dạng kiểu chữ, cỡ chữ trong khung RichTextBox.
* Định dạng văn bản và màu sắc: Bao gồm các ToggleButton và ComboBox cho việc định dạng canh lề và màu sắc chữ.
* Lưu, tạo mới hoặc mở file: Cho phép người dùng tạo một khung Journal Editor mới hoặc mở một file đã có sẵn. Sau khi người dùng nhập liệu xong, họ có thể dùng nút Save để lưu lại.

### Giao diện trang cài đặt



Hình 17: Giao diện trang cài đặt

Mô tả thành phần giao diện: Trang cài đặt chứa tất cả các tùy chọn cài đặt mà người dùng có thể thực hiện. Mỗi tùy chọn sẽ có một đề mục, dòng giới thiệu về tùy chọn đó và một control thực hiện việc tùy chỉnh.

## Thiết kế xử lý

### Geocoding và Reverse-Geocoding

Geocode là việc chuyển đổi một địa chỉ thông thường sang địa chỉ địa lý (Geopoint) gồm có kinh độ (Longitude), vĩ độ (Latitude), và độ cao (Altitude). Reverse-Geocode là thuật toán ngược lại so với Geocode. Thuật toán này chạy asynchronous để tìm kiếm ngầm bên dưới giao diện mà không ảnh hưởng nhiều đến quá trình trải nghiệm người dùng.

Mục đích của việc thiết kế và cài đặt hai thuật toán này là vì người dùng thích sử dụng những địa chỉ thực tế, thông dụng hơn là những địa chỉ địa lý khô khan, vô nghĩa đối với họ. Ngược lại, việc giao tiếp giữa các phần trong phần mềm thì yêu cầu việc sử dụng địa điểm chính xác.

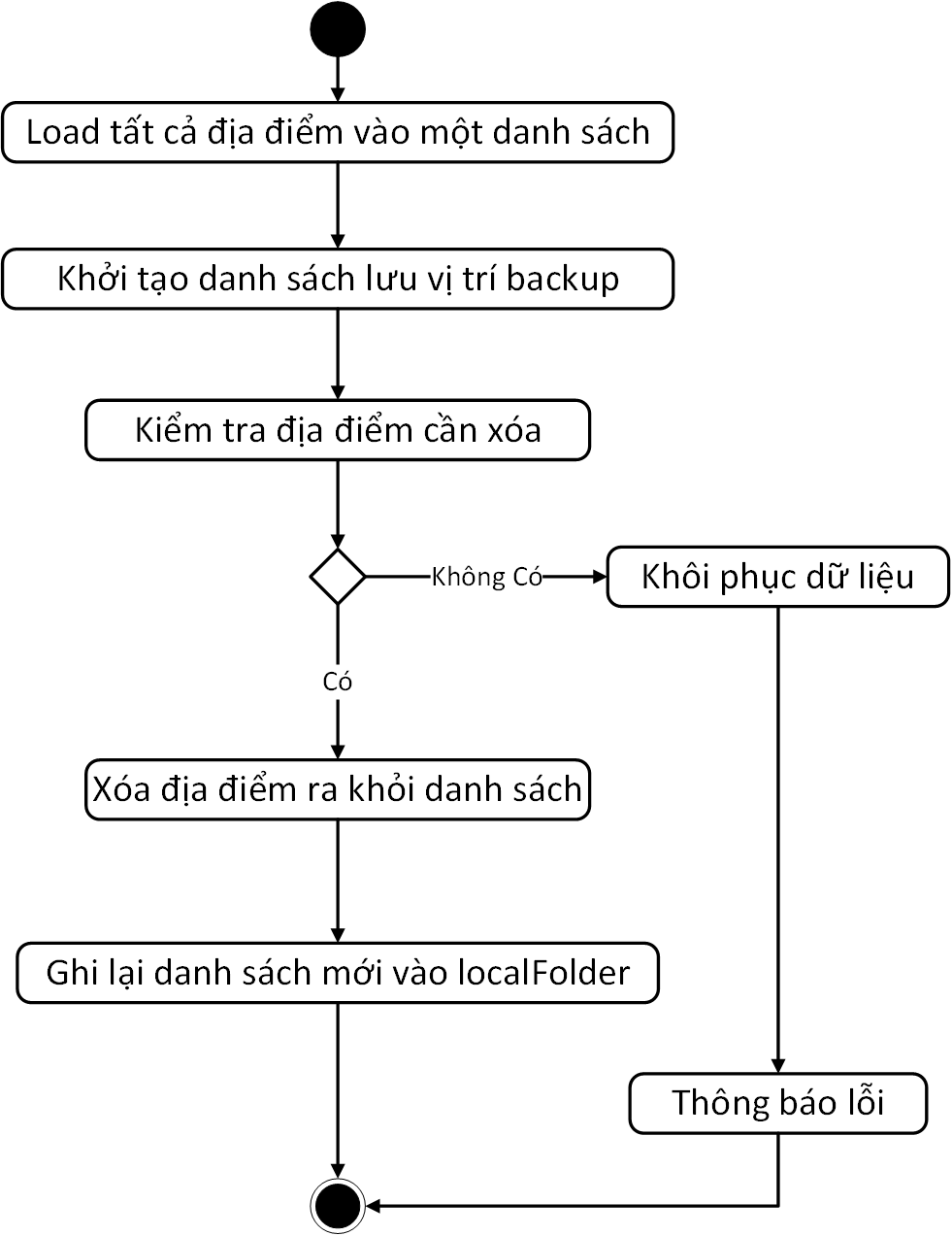
Trong đó, hệ thống bản đồ của phần mềm, hệ thống ghim sử dụng trực tiếp việc chuyển đối này rất thường xuyên.

### Thêm và xóa ghim bản đồ

Bản đồ hỗ trợ người dùng thêm ghim tại một địa điểm trên bản đồ, và lưu hoặc xóa địa điểm đó vào danh sách. Danh sách này sẽ được lưu tại thư mục mặc định dành riêng cho phần mềm UWP (localFolder). Các thao tác đến localFolder sẽ được thiết kế asynchronous để tối ưu hiệu suất phần mềm.

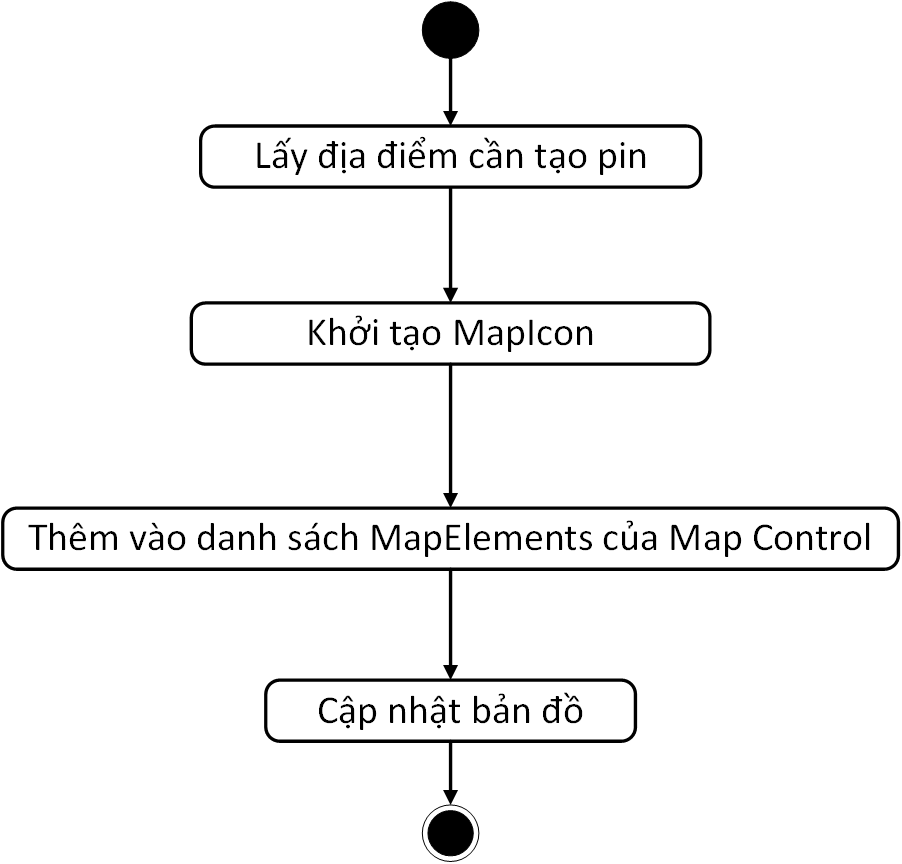
Mỗi địa điểm được lưu lại sẽ được lưu bao gồm 3 dòng: tựa đề địa điểm, vị trí vĩ độ, và vị trí kinh độ.

Để xóa một vị trí Icon ra khỏi danh sách, ta load tất cả các địa điểm hiện có vào một danh sách và xóa địa điểm cần xóa ra khỏi danh sách ấy. Sau đó, cập nhật lại file lưu địa điểm với danh sách mới. Đây là thuật toán dù hiệu suất không cao nhưng rất dễ cài đặt và phù hợp với phạm vi dữ liệu nhỏ.



Hình 18: Activity Diagram thuật toán xóa một MapIcon khỏi danh sách

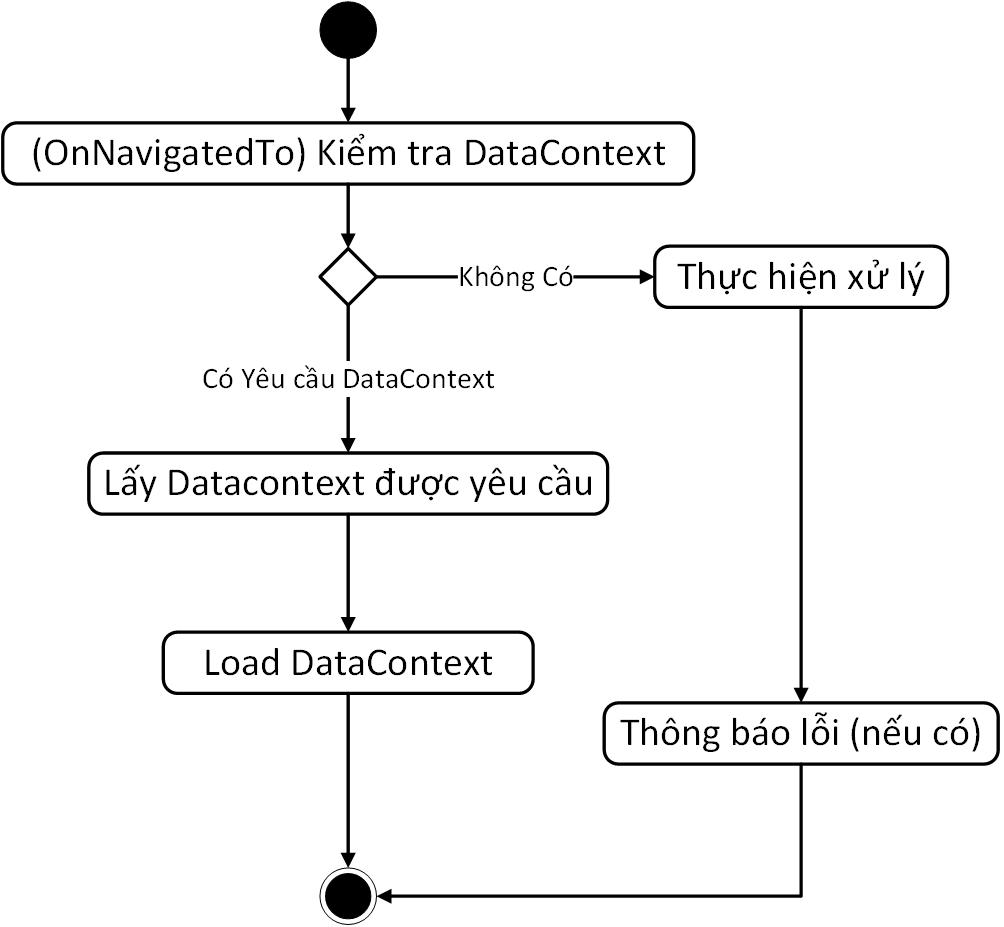
Để thêm một Icon mới vào bản đồ, ta làm theo thuật toán như hình bên dưới. Để xóa Icon trên một bản đồ, ta duyệt tất cả các MapElements của MapControl để tìm ra Icon cần xóa và xóa nó.



Hình 19: Activity Dagram thuật toán thêm MapIcon

### Load thông tin khu vực, thông tin từng địa điểm

Mỗi trang thông tin về các thành phố (Country.xaml), thông tin về các địa điểm trong thành phố/ địa phận ấy (Province.xaml), và thông tin về từng địa điểm (Location.xaml), cũng như thông tin tương ứng trong database sẽ được thiết kế theo một mẫu chung. Thông tin về địa điểm chính xác sẽ được load lên dựa trên thao tác của người dùng.



Hình 20: Activity Diagram cho thuật toán load DataContext

# Cài Đặt Phần Mềm



## Tổng quan về công nghệ sử dụng

### Universal Windows Platform



Khi bạn tạo một ứng dụng Universal Windows Platform có nghĩa là bạn đang tạo ra một ứng dụng có khả năng chạy trên bất kì thiết bị Windows nào:

* Dòng thiết bị di động: điện thoại di động…
* Dòng thiết bị máy tính cá nhân: laptop, máy tính để bàn, máy tính bảng.
* Dòng thiết bị gia đình: Surface hub
* Dòng thiết bị IoT: các thiết bị đeo, vật dụng trong nhà…
* Dòng thiết bị dành cho Xbox

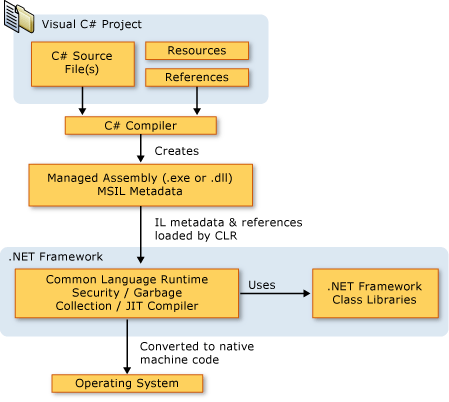
Universal Windows Platform cung cấp một số tính năng đã được xây dựng sẵn và các khối hợp nhất giúp cho việc tạo “trải nghiệm di động” (ứng dụng không chỉ hỗ trợ trên nhiều loại kích thước màn hình khác nhau mà còn phải hỗ trợ tương tác hạ tầng một cách dễ dàng như việc sử dụng chuột, bàn phím, cảm ứng….) trên Windows trở nên dễ dàng hơn.

* Chỉ có một cửa hàng ứng dụng cho mọi thiết bị Windows: sau khi đăng kí một tài khoản lập trình viên, bạn có thể đăng tải ứng dụng của bạn lên trên cửa hàng, và sẵn sàng chạy trên mọi thiết bị. Điều này rất dễ dàng trong việc đăng tải và quản lí tất cả các ứng dụng Windows của bạn ở một nơi nhất định.
* Thiết kế tương thích với giao diện người dùng: khi bạn thiết kế ứng dụng UWP, nghĩa là bạn đang thiết kế với các ‘pixel hiệu quả’ (effective pixel), chứ không phải là các pixel vật lý. Điều này giúp trung hòa các control, phông chữ và các thành phần giao diện hiển thị trên màn hình.
* Hợp nhất các đầu vào và các tương tác thông minh: các ứng dụng UWP sử dụng một hệ thống đầu vào thông minh, bạn có thể xoanh quanh một tương tác ví dụ như click mà không cần quan tâm tương tác này đến từ chuột, bút cảm ứng hay là cảm ứng. Ứng dụng UWP làm việc rất tốt trên với đa thiết bị đầu vào như bàn phím, chuột, cảm ứng, bút, và các thiết bị Xbox.
* Với duy nhất một bộ công cụ phát triển phần mềm (Software Development Kit) cung cấp cho bạn những thư viện, tập hợp toàn bộ các API để xây dựng ứng dụng Windows đa nền tảng.
* Ứng dụng UWP rất dễ dàng để đồng nhất với các dịch vụ đám mây như Azure Mobile Services (lưu trữ dữ liệu trên đám mây, gửi thông báo đến user, xác thực tài khoản), Azure Storage, Salesforce và Office 365.

### Ngôn ngữ lập trình C#

C# là một ngôn ngữ cấp cao và thuần hướng đối tượng, hỗ trợ rất mạnh trong việc xây dựng các ứng dụng trên môi trường Windows. Đối với những lập trình viên đã từng sử dụng qua C, C++, hoặc Java, việc nắm bắt và sử dụng C# một cách hiệu quả là khá dễ dàng và ít tốn thời gian.

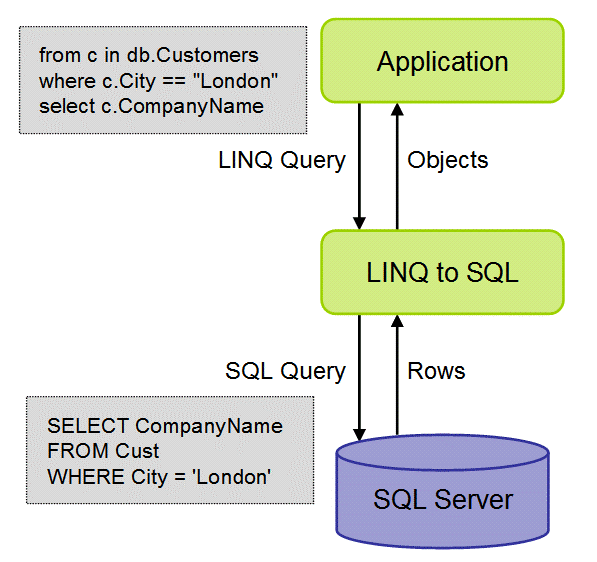
Mã nguồn được viết bằng ngôn ngữ C# sẽ được dịch sang một ngôn ngữ tầm trung (Intermediate Language) phù hợp với các đặc tả của Common Language Infrastructure (CLI), và được lưu trữ trong các tập tin thực thi (thường có đuôi là .exe hay .dll.



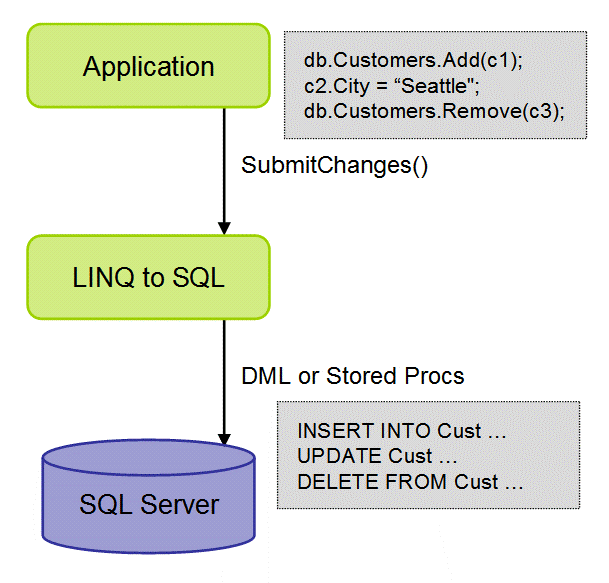
Hình 21: .NET Framework platform architecture  
Credit: msdn.microsoft.com

### Language-integrated query

Language-Integrated Query (LINQ) là một ngôn ngữ truy vấn trong .NET Framework, được tích hợp vào trong những ngôn ngữ lập trình chính (chẳng hạn, C# hay Visual Basic). Thay vì phải sử dụng một ngôn ngữ thứ ba để truy xuất đến các cơ sở dữ liệu, phần mềm sẽ trực tiếp sử dụng C# trong lớp Data Access của mình. Việc cài đặt bằng LINQ này giúp giảm một lượng chi phí không nhỏ trong việc cài đặt phần mềm.



Hình 22: Truy vấn LINQ được biên dịch thành ngôn ngữ dùng cho SQL Server

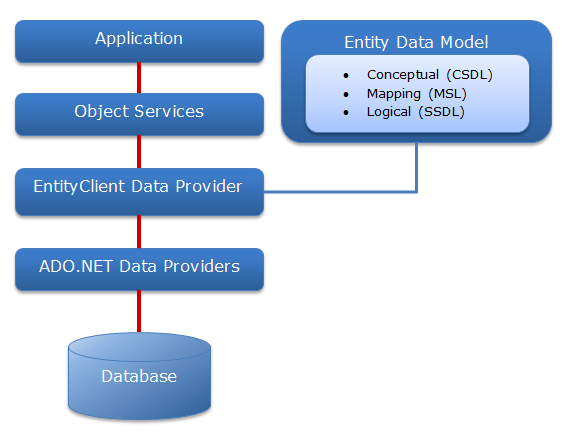


Hình 23: Thao tác LINQ được biên dịch thành ngôn ngữ dùng cho SQL Server

### Một số framework được sử dụng

* **Entity Framework Core**

Nền tảng được sử dụng để làm việc với database thông qua cơ chế ánh xạ Object/Relational Mapping (ORM). Nhờ đó, bạn có thể truy vấn, thao tác với database gián tiếp thông qua các đối tượng lập trình.



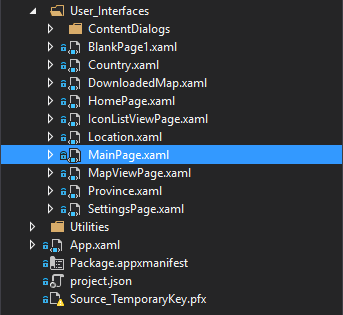
* **Windows SDK for Facebook**

Sử dụng để thực hiện việc đồng bộ đăng nhập thông qua facebook nhằm cho User có khả năng thực hiện chức năng Review\_Location

## Cài đặt phần mềm

### View

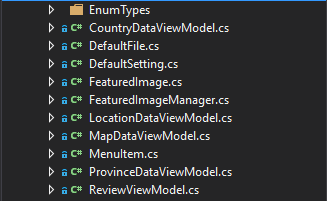
Sử dụng một Window chính là MainWindow.xaml để chứa thanh định hướng (Navigation Bar) hoạt động như một menu và khung chứa (Frame) để host những trang (Page) giao diện khác.



Hình 24: Các thành phần giao diện của phần mềm

Mỗi cửa sổ và trang giao diện đều được cài đặt bằng ngôn ngữ XAML và cài đặt thuật toán nền bên dưới bằng C#. Các phương thức hoạt động trong lớp giao diện sẽ lần lượt gọi đến các đối tượng bên dưới (ở lớp nghiệp vụ) để thực hiện nhiệm vụ của mình.

### View Model

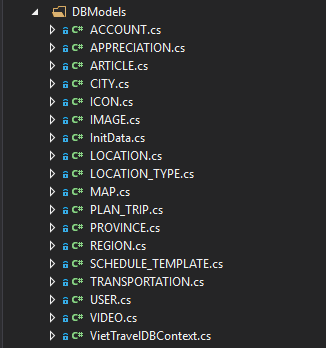


Hình 25: Các thành phần lớp trung gian View Model

Nhiệm vụ quan trọng của nó là chứa các mã lệnh cần thiết để thực hiện data binding, command.

### Model

Trực tiếp thao tác trên dữ liệu thực sự



Hình 26: Các Class Model

Nhóm sử dụng Code First Migration để hiện thực Database vì:

* Đơn giản, Code First Migration đề cập đến vấn đề bạn gặp phải khi cập nhật CSDL như thêm cột, xóa cột, tạo bảng mới và cập nhật vào CSDL đang có khi bạn sử dụng Code First trong Entity Framework. Việc cập nhật sẽ được thực hiện tự động mỗi khi có thay đổi trong các model mà bạn đã khai báo.
* Nhờ nó bạn có thể thực hiện các thao tác nâng cấp (upgrade) hoặc hạ cấp (downgrade) CSDL bằng chính mã lệnh bên trong dự án của bạn.

## Vấn đề khi cài đặt

### Cài đặt giao diện

* Cơ chế binding, command của XAML với C# khá phức tạp.
* Các control dạng bảng có sẵn của .NET chỉ thỏa mãn được những nhu cầu rất nền tảng. Những yêu cầu nâng cao thường thấy rất khó cài đặt.
* Những giao diện cần kiểm tra các trường nhập liệu thường bị phình to và trở nên khá rối rắm.

### Cài đặt lớp nghiệp vụ

Chỉ có một vài vấn đề nhỏ, không tồn tại các vướng mắc lớn: cách try-catch cho phù hợp, cách cài đặt hàm và lớp cho hợp lí, đồng nhất…

### Cài đặt lớp truy cập cơ sở dữ liệu

* LINQ to SQL có một số lỗi (vẫn chưa được sửa trong những bản cập nhật mới nhất) khi việc làm việc và truy vấn dữ liệu ở các trường có thuộc tính như nchar, nvarchar, text… Đó là: không thể sử dụng một số các hàm và các operator mặc định trong LINQ. Có một số workaround từ cơ bản đến nâng cao nhưng nhìn chung chúng khiến mã nguồn bị phình to một lượng đáng kể.
* Các sai sót về khóa ngoại và khóa chính trong quá trình cài đặt cơ sở dữ liệu nằm tiềm ẩn cho đến khi thực sự sử dụng đến chúng, và chúng khá khó tìm ra. Lỗi đã gặp: khóa ngoại được liên kết sai dẫn đến việc thêm record vào bảng bị lỗi. Điểm khó ở đây là lỗi chỉ xuất hiện với một thuật toán nhất định. Khi sử dụng thuật toán cũ, mọi thứ đều hoạt động như ý muốn. Điều này dẫn đến việc tập trung nhân lực tìm kiếm lỗi ở thuật toán, gây mất thời gian và sức lực.
* Cơ sở dữ liệu sau khi cài đặt thường là yếu tố cố định trong việc tìm kiếm các thuật toán giải quyết các bài toán. Nguyên do là những thay đổi trong cơ sở dữ liệu khá phức tạp và đòi hỏi nhiều thay đổi liên đới.

# Kiểm Thử



## Unit Test

Để đảm bảo chất lượng của chương trình, từng lớp sẽ được kiểm tra một cách riêng rẽ bằng phương pháp hộp trắng. Người tiến hành sẽ là những người trực tiếp cài đặt các lớp và phương thức ấy.

## Integration Test

Kiểm tra sự liên kết và giao diện giữa các lớp trong phần mềm.

## System Test

### Phương pháp kiểm tra

Hệ thống sẽ được kiểm tra theo phương pháp hộp đen. Tức là dựa vào những đặc tả và yêu cầu của hệ thống đã thu thập được từ đầu để kiểm tra output chung của hệ thống mà không cần quan tâm đến việc thực hiện bên trong của chúng.

# Tổng Kết



## Tổng Kết

Phần mềm bước đầu đã được phát triển đi đúng theo dự tính và kế hoạch. Tuy gặp khá nhiều khó khăn trong vấn đề tiếp cận với công nghệ lập trình mới, nhóm vẫn đảm bảo được các tiến độ làm việc do đã đặt ra.

## Nhận Xét & Đánh Giá

Trong thời gian có hạn, việc hoàn thành dự án đã vượt qua được nhiều thử thách nhất định nhưng phần mềm vẫn còn nhiều điểm có thể cải tiến nhằm nâng cao chất lượng phục vụ và trải nghiệm của người dùng:

* Cải thiện giao diện người dùng,
* Cài đặt và hoàn thiện nhiều tính năng nâng cao,
* Mở rộng dữ liệu trong database hoặc hướng đến sử dụng dịch vụ cloud database.

## Phân Rã Công Việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Công việc** | **Duration (days)** | **Assign To** | **% Finished** |
| 1 | Tìm hiểu sơ bộ & đăng ký đồ án |  | Lam, Khải, Khoa | 100% |
| 2 | Tìm hiểu công nghệ liên quan |  | Lam, Khải | 100% |
| **I. VIẾT BÁO CÁO** | | | | |
| 3 | **Chương 1 – Hiện trạng** |  |  |  |
| 4 | 1.1. Hiện trạng vấn đề |  | Lam, Khoa | 100% |
| 5 | 1.2. Hiện cơ sở vật chất và con người |  | Lam, Khoa | 100% |
| 6 | 1.3. Yêu cầu về phần mềm |  | Lam | 100% |
| 7 | **Chương 2: Phân tích** |  |  |  |
| 8 | 2.1. Yêu cầu phần mềm |  | Lam | 100% |
| 9 | 2.2. Mô hình hóa |  | Lam | 100% |
| 10 | **Chương 3: Thiết kế** |  |  |  |
| 11 | 3.1. Thiết kế kiến trúc phần mềm |  | Khải | 100% |
| 12 | 3.2. Thiết kế dữ liệu |  | Khải | 100% |
| 13 | 3.2.1. Thiết kế dữ liệu bộ nhớ chính |  | Khải | 100% |
| 14 | 3.3. Thiết kế giao diện và thành phần xử lí của giao diện |  | Lam | 100% |
| 15 | **Chương 4: Cài đặt** |  |  |  |
| 16 | 4.1. Tổng quan về công nghệ sử dụng |  | Lam, Khải | 100% |
| 17 | 4.2. Cài đặt phần mềm |  | Lam, Khải | 100% |
| 18 | 4.3. Vấn đề khi cài đặt |  | Lam | 100% |
| 19 | **Chương 5: Kiểm thử** |  | Lam | 80% |
| 20 | 5.1. Unit Test |  | Lam |  |
| 21 | 5.2. Integration Test |  | Lam |  |
| 22 | 5.3. System Test |  | Lam |  |
| 23 | **Chương 6: Kết luận** |  | Lam | 100% |
| 24 | **Tài liệu tham khảo** |  | Lam | 100% |
| **II. LẬP TRÌNH** | | | | |
|  | 1. Database | 15 | Khải, Khoa | 90% |
|  | 2. Giao diện | 20 | Lam, Khải | 90% |
|  | 3. Map, Text Editor | 15 | Lam | 90% |
|  | 4. Trang giới thiệu thông tin địa điểm. | 15 | Khải. | 90% |
|  | 5. Kết nối Facebook | 7 | Khải | 60% |
| **III. KIỂM THỬ** | | | | |
|  | 1. Unit Test |  |  |  |
|  | 2. Integration Test |  |  |  |
|  | 3. System Test |  |  |  |
|  | 4. Acceptance Test |  |  |  |
| **IV. NỘP BÁO CÁO TIẾN ĐỘ & SẢN PHẨM** | | | **Tiến độ sản phẩm** | |
|  | **Ngày 12 tháng 04 năm 2017** | | Cơ bản hoàn thành định hướng đồ án. Thiết kế sơ bộ giao diện. | |
|  | **Ngày 09 tháng 05 năm 2017** | | Hoàn thành được các chức năng lớn của bản đồ.  Hoàn thành được prototype cho việc binding các trang thông tin. | |
|  | **Ngày 20 tháng 05 năm 2017** | | Hoàn thành sơ bộ kết nối với Facebook.  Hoàn thành mở rộng cơ sở dữ liệu.  Hoàn thành Journal Editor. | |
|  | **Ngày 10 tháng 06 năm 207** | | Hoàn thành các yêu cầu về release (Windows Store). | |

Bảng 5: Bảng phân công công việc và báo cáo tiến độ sản phẩm

# TÀI LIỆU THAM KHẢO

(IEEE Citation Format)

[1]" Universal Windows Platform (UWP) app samples", *Microsoft*. [Online]. Available: https://github.com/Microsoft/Windows-universal-samples/tree/master/.

[2]"LINQ: .NET Language Integrated Query", *Msdn.microsoft.com*, 2007. [Online]. Available: https://msdn.microsoft.com/en-us/library/bb308959.aspx.

[3]I. Marsic, "Software Engineering Project Report - Requirements", *Eceweb1.rutgers.edu*, 2012. [Online]. Available: http://eceweb1.rutgers.edu/~marsic/Teaching/SE/report1.html.

[4]"Software Testing Fundamentals", *Softwaretestingfundamentals.com*. [Online]. Available: http://softwaretestingfundamentals.com/.

[5]A. Troelsen and P. Japiske, C# 6.0 and the .NET 4.6 framework, 7th ed. Apress, 2015.