TRƯỜNG ĐẠI HỌC

**SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**





**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

NGÀNH CÔNG NGHỆ THÔNG TIN

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG WEBSITE ĐĂNG TIN**

**VÀ TÌM KIẾM NHÀ TRỌ**

**SVTH: CHU MINH HOÀNG 17110139**

**PHẠM HUỲNH THANH LÂM 17110168**

**Khóa: 2017 – 2021**

 **Major: CÔNG NGHỆ THÔNG TIN**

 **` Instructor: ThS. TRẦN CÔNG TÚ**

TP. Hồ Chí Minh, 23 July 2021

|  |  |
| --- | --- |
|  | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  --------  *Tp. Hồ Chí Minh, \_\_\_tháng 12, 2020* |

# **NHIỆM VỤ TIỂU LUẬN CHUYÊN NGÀNH**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Họ tên sinh viên: Chu Minh Hoàng | | | MSSV: 17110139 | |
| Họ tên sinh viên: Phạm Huỳnh Thanh Lâm | | | MSSV: 17110168 | |
| Chuyên ngành: Công nghệ thông tin \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | Lớp: 17110CLST3 - ST1 | |
| Giảng viên hướng dẫn: ThS. TRẦN CÔNG TÚ | | | Điện thoại: 0983 674 375 | |
| Ngày nhận đề tài: 27/08/2020 | | | Ngày nộp đề tài: | |
| 1. Tên đề tài: Xây dựng website đăng tin và tìm kiếm nhà trọ. | | | | |
| 2. Các số liệu, tài liệu ban đầu:   * Hiện thực thuật toán bằng HTML, CSS, JS, Bootstrap, ASP.NET CORE 3.x, SQL Server, Angular. * Thiết kế trang web đăng tin và tìm kiếm nhà trọ. * Tham khảo tài liệu của w3school. | | | | |
| 3. Nội dung thực hiện đề tài:   * Nghiên cứu các thiết kế được một website đăng tin và tìm kiếm nhà trọ với đầy đủ các tính năng như: đăng tin, quản lý tin tức, tìm kiếm nhà trọ, thanh toán sau khi đăng tin, quản lý dịch vụ, quản lý nhân viên, … * Phân tích chức năng của website và lựa chọn phương án thiết kế website phù hợp. | | | | |
| 4. Sản phẩm:  Thiết kế và Xây dựng website đăng tin và tìm kiếm nhà trọ có đầy đủ các chức năng cơ bản. | | | | |
|  | | | **GIẢNG VIÊN HƯỚNG DẪN**  *(Kí và ghi rõ họ tên)* | |
|  | | |  | |
|  | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  --------  *Tp. Hồ Chí Minh, \_\_\_ tháng 12, 2020* | |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên sinh viên: Chu Minh Hoàng Mã số sinh viên: 17110139

Họ và tên sinh viên: Phạm Huỳnh Thanh Lâm Mã số sinh viên: 17110168

Ngành: Công nghệ thông tin.

Tên đề tài: Xây dựng website đăng tin và tìm kiếm nhà trọ.

Họ và tên giáo viên hướng dẫn: ThS. TRẦN CÔNG TÚ

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

1. Ưu điểm:

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

1. Khuyết điểm:

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

1. Đề nghị cho bảo vệ hay không?

.................................................................................................................................

1. Đánh giá loại:

.................................................................................................................................

1. Điểm: …………… (bằng chữ………………………………………………....)

*Tp. Hồ Chí Minh, ngày tháng 12, 2020*

**GIẢNG VIÊN HƯỚNG DẪN**

|  |  |
| --- | --- |
|  | CỘNG HOÀ XÃ HỘI CHŨ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  --------  *Tp. Hồ Chí Minh, \_\_\_ tháng 12, 2020* |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Họ và tên sinh viên: Chu Minh Hoàng Mã số sinh viên: 17110139

Họ và tên sinh viên: Phạm Huỳnh Thanh Lâm Mã số sinh viên: 17110168

Ngành: Công nghệ thông tin.

Tên đề tài: Xây dựng website đăng tin và tìm kiếm nhà trọ.

Họ và tên giáo viên phản biện:

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

1. Ưu điểm:

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

1. Khuyết điểm:

.................................................................................................................................

.................................................................................................................................

.................................................................................................................................

1. Đề nghị cho bảo vệ hay không?

.................................................................................................................................

1. Đánh giá loại:

.................................................................................................................................

1. Điểm: ................. (bằng chữ................................................................................)

*Tp. Hồ Chí Minh, ngày tháng 12, 2020*

**GIẢNG VIÊN PHẢN BIỆN**

# **LỜI CẢM ƠN**

Lời đầu tiên, nhóm thực hiện xin được phép gửi lời cảm ơn chân thành đến khoa Đào tạo Chất Lượng Cao – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi nhất cho nhóm thực hiện có cơ hội được tự do tiếp cận, tham khảo, mở rộng thêm kiến thức trong lĩnh vực Công nghệ thông tin nói chung và môn Tiểu Luận Chuyên Ngành nói riêng.

Lời cảm ơn trân trọng nhất nhóm thực hiện xin chân thành gửi đến Thầy **Trần Công Tú** – người đã dùng mọi tâm huyết và tri thức của người Thầy, cùng đồng hành và trực tiếp hướng dẫn và tạo mọi điều kiện thuận lợi giúp đỡ cho nhóm phát huy hết khả năng cũng như nâng cao kiến thức trong suốt quá trình học tập, đặc biệt là trong quá trình chuẩn bị và thực hiện Tiểu luận chuyên ngành. Cảm ơn sự nhiệt tình của Thầy, là động lực vô cùng to lớn giúp nhóm thực hiện thực hiện, kiên trì trong suốt quá trình thực hiện đề tài và khám phá ra những kiến thức mới đầy thú vị và bổ ích liên quan đến đề tài mà cụ thể là đề tài **Xây dựng website đăng tin và tìm kiếm nhà trọ**.

Cuối cùng nhóm thực hiện kính chúc quý thầy cô dồi dào sức khỏe, niềm tin để tiếp tục sự nghiệp cao quý của mình là truyền đạt kiến thức cho thế hệ mai sau.

Nhóm thực hiện xin chân thành cám ơn.

*TPHCM*, ngày tháng 12 năm 2020

Sinh viên thực hiện

# **LỜI MỞ ĐẦU**

Thế giới ngày nay đã có nhiều tiến bộ mạnh mẽ về công nghệ thông tin từ một tiềm năng thông tin đã trở thành một tài nguyên thực sự, cùng với đó từ khi ra đời máy tính đã nhanh chóng phát triển và đóng một vai trò rất quan trọng trong nghiên cứu khoa học cũng như là đời sống. Những công việc hàng ngày đã dần được máy móc hóa để con người có thể sử dụng bớt sức lực hơn. Những nhu cầu được thực hiện điển hình như các trang thương mại điện tử để bán hàng, đặt hàng, mọi việc đang dần số hóa. Cùng với nhu cầu về tìm kiếm và đăng tin nhà trọ cũng không ngoại lệ, nhu cầu về nhà ở ngày càng cao đặc biệt là đối với những đối tượng là sinh viên hoặc nhân viên văn phòng chiếm số lượng lớn và nhu cầu thuê nhà của họ đang ở mức cao. Nhu cầu về đăng tin nhà trọ cũng chiếm một số lượng lớn, mô hình kinh doanh nhà trọ đang phát triển ở thành phố. Vậy nên nhóm chọn đề tài “Xây dựng website đăng tin và tìm kiếm nhà trọ”.

Hiện nay theo xu hướng của thị trường có rất nhiều website xây dựng để đăng tin và mua bán lớn như chotot123, mogi, homedy, .... Những website trên có rất nhiều chức năng khác nhau, nhưng nhóm thực hiện quyết định xây dựng website tập trung vào việc tìm kiếm và đăng tin nhà trọ.

# **CONTENT**

[**NHIỆM VỤ TIỂU LUẬN CHUYÊN NGÀNH**](#_Toc59602009)

[**LỜI CẢM ƠN**](#_Toc59602010)

[**LỜI MỞ ĐẦU**](#_Toc59602011)

[**MỤC LỤC**](#_Toc59602012)

[**DANH MỤC CÁC TỪ VIẾT TẮT**](#_Toc59602013)

[**DANH MỤC CÁC BẢNG BIỂU**](#_Toc59602014)

[**DANH MỤC CÁC BIỂU ĐỒ VÀ HÌNH ẢNH**](#_Toc59602015)

[**CHƯƠNG 1 TỔNG QUAN** 1](#_Toc59602016)

[**1.1** **Lý do chọn đề tài** 1](#_Toc59602017)

[**1.2** **Mục tiêu** 1](#_Toc59602018)

[**1.3** **Công nghệ sử dụng** 1](#_Toc59602019)

[**CHƯƠNG 2 CƠ SỞ LÝ THUYẾT** 2](#_Toc59602020)

[**2.1** **Angular framework** 2](#_Toc59602021)

[**2.2** **ASP.NET API** 3](#_Toc59602022)

[**2.3** **ASP.NET Entity Framework** 3](#_Toc59602023)

[**2.4** **Microsoft SQL 2017** 3](#_Toc59602024)

[**2.5** **Firebase** 4](#_Toc59602025)

[**2.6** **Phần mềm chat trực tuyết tawk.to** 4](#_Toc59602026)

[2.6.1 Giới thiệu 4](#_Toc59602027)

[2.6.2 Mục đích 4](#_Toc59602028)

[**CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ** 5](#_Toc59602029)

[**3.1** **Khảo sát hiện trạng** 5](#_Toc59602030)

[**3.1.1** **Khảo sát hệ thống phongtro123** 5](#_Toc59602031)

[3.1.2 Trang web tham khảo, khảo sát: Chợ Tốt Nhà 13](#_Toc59602032)

[3.1.3 Trang web tham khảo, khảo sát: mogi 16](#_Toc59602033)

[3.1.4 Khảo sát hệ thống phòng trọ: Homedy 20](#_Toc59602034)

[3.1.5 Kết luận 27](#_Toc59602035)

[**3.2** **Mô hình hóa yêu cầu** 32](#_Toc59602036)

[3.2.1 Thiết kế usecase diagram 32](#_Toc59602037)

[3.2.2 Đặc tả usecase 33](#_Toc59602038)

[**3.3** **Sơ đồ trình tự (sequence diagram)** 63](#_Toc59602039)

[**3.4** **Thiết kế cơ sở dữ liệu** 68](#_Toc59602040)

[3.4.1 Mô hình liên kết thực thể 68](#_Toc59602041)

[3.4.2 Mô hình quan hệ các bảng trong cơ sở dữ liệu 69](#_Toc59602042)

[**3.5** **Thiết kế giao diện** 69](#_Toc59602043)

[3.5.1 Giao diện dành cho người dùng 69](#_Toc59602044)

[3.5.2 Giao diện admin và nhân viên 83](#_Toc59602045)

[**CHƯƠNG 4 CÀI ĐẶT SẢN PHẨM** 84](#_Toc59602046)

[**4.1** **Font end** 84](#_Toc59602047)

[4.4.1 Cấu trúc ứng dụng 84](#_Toc59602048)

[4.4.2 Các đoạn xử lý chính trong font-end 85](#_Toc59602049)

[**4.2** **Back end** 86](#_Toc59602050)

[4.2.1 Cấu trúc project API 86](#_Toc59602051)

[4.2.2 Một số Function và Procedure 88](#_Toc59602052)

[**4.3** **Sử dụng phần mềm thứ ba tawk.to** 89](#_Toc59602053)

[**CHƯƠNG 5 TỔNG KẾT** 90](#_Toc59602054)

[**5.1** **Kết quả đạt được** 90](#_Toc59602055)

[**5.2** **Ưu điểm** 91](#_Toc59602056)

[**5.3** **Nhược điểm** 91](#_Toc59602057)

[**5.4** **Khó khăn** 92](#_Toc59602058)

[**5.5** **Bài học kinh nghiệm** 92](#_Toc59602059)

[**5.6** **Hướng phát triển** 92](#_Toc59602060)

[**TÀI LIỆU THAM KHẢO** 93](#_Toc59602061)

[**PHỤ LỤC** 94](#_Toc59602062)

[Phụ lục I: Cài đặt môi trường phát triển 94](#_Toc59602063)

[Phụ lục II: Tạo một ứng dụng Angular application 94](#_Toc59602064)

# **CONTENT OF ACRONYMS**

CSDL: Cơ sở dữ liệu

ORM: Object Relational Mapping

ĐT: Đồng tốt

HCM: Hồ Chí Minh

BDS: Bất động sản

# **CONTENT TABLES**

# **CONTENT IMAGE AND DIAGRAM**

# **CHAPTER 1 OVERVIEW**

## **The reason for choosing the topic.**

To learn more about how to create a website with basic functions. Learn more about how to create a website with Angular, and ASP.Net Core. Simultaneously serving the needs of finding accommodation of everyone, especially students, or office staff in the fastest, and easiest way. In addition, also for posting information of users or businesses.

## **Target**

The goal of the product is to create a website that makes it easy for the owner to post their accommodation on the website, while also supporting a large number of students as well as those in need. Can view and contact the inn management. In addition to grasping the needs of users in viewing accommodation details, and contacting the hostel manager, they can search for the accommodation right where they live or are looking for, from there it is easy. choose an inn that suits you.

## **Technology used.**

To build a suitable application for the theoretical problem, the topic will explore, and research the following areas:

* Learn about Angular, the JavaScript programming language, and website application libraries.
* Studying on SQL databases, Firebase for data storage
* Learn about bootstrap, HTML, and CSS for interface design.
* Learn ASP.Net to write Back-end.
* Apply Technology: Leaflet, FlashPython, Flutter, VietMap Api, SignalR
* Find out how to pay via Papal (Visa card)

# **CHAPTER 2 THEORETICAL BASIS**

## **Angular framework**

Angular is considered a free open source or framework dedicated to web design. Angular has been in development since 2009 and is maintained by Google. These frameworks are considered the most powerful front-end frameworks dedicated by advanced HTML programmers. [1]

Angular is widely used to build Single Page Applications (SPA) projects. Currently, Angular's Version stable is Angular 9 (released on February 7, 2020) with TypeScript 3.6, and 3.7. [1]

* Highlights: [1]
* AngularJS is considered to be the solution for easy single-page applications.
* Front-end code is usually very friendly thanks to the ability to Binding data on HTML platforms to create all sorts of great operations.
* You can easily Unit test
* Components can be reused more easily.
* Support for programmers who can write less code with more functionality.
* You can run AngularJS in a variety of browsers, either on a PC or mobile.
* The basic features: [1]
* Used for JavaScript-based development.
* Angular can create client-side applications based on the MVC model.
* Angular possesses high compatibility and can automatically handle Javascript code so that it is suitable for most browsers.
* When free, and open-source code is available, it will be used more widely.
* Usually, the architecture of an Angular application is based on the ideas related to Components. Each Angular application usually starts with the same top levels called Root Components.

## **ASP.NET API**

API term stands for Application Programming Interface - Application Programming Interface. ASP.NET Web API is a framework, provided by Microsoft, that makes it easy to build Web API, i.e services based on HTTP protocol. The ASP.NET Web API is the ideal platform for building Restful services on top of the .NET Framework. These Web API services can be used by many different clients, such as: [2]

* Browser
* Mobile application
* Desktop application
* IOT, etc.

## **ASP.NET Entity Framework**

Entity Framework is an ORM. Entity Framework [3]

ORM is a set of technologies that allow working with Relational Database Management System (RDBMS) from object-oriented languages ​​and without needing it directly. [process SQL queries](https://tuhocict.com/thuc-thi-truy-van-sql-trong-c-lop-sqlcommand/). [3]

ORMs like Entity Framework support programmer to map (two dimensions): class with table structure; object with records (or rows) in the table; property with the column of the table; sets objects with sets of records; references to objects with relationships between tables. During this process, all SQL queries are generated, and executed automatically by the ORM. The programmer only needs to work with familiar concepts of the programming language. [3]

## **Microsoft SQL 2017**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product whose primary function is to store, and retrieve data required by other software applications. Can run on the same computer or on another computer on the network (including the Internet). [4]

Microsoft markets at least a dozen different versions of Microsoft SQL Server, aimed at different audiences, and for different workloads, from small single-machine applications to large Internet applications with many concurrent users.[4]

## **Firebase**

Firebase is a real-time database service provided by Google and operated in the cloud. It helps programmers to rapidly develop mobile applications by simplifying database operations, and storing, and synchronizing real-time user data. Applications that support this feature can store and retrieve data from the server very quickly. The data is stored in NoSQL database system and is placed on cloud server platform. Data is written and read with the lowest time in milliseconds. This platform supports the synchronization of user data even when there is no network connection. Creates a seamless experience regardless of the user's internet connection. [5]

## **Online chat software tawk.to**

### Introduce

Tawk.to is a free online chat software to help customers interact with website owners. Helping to answer customer questions as quickly as possible is to help businesses increase sales, and revenue quickly. [6]

### Purpose

Integrating tawk.to helps to support communication between customers, and website, helps to answer customers' questions, and at the same time helps employees proactively approach, and communicate with customers.

## **Flutter**

### Define

Flutter is a cross-platform application development platform for iOS and Android developed by Google that is used to create native apps for google.

Flutter consists of 2 important components:

* An SDK (Software Development Kit): A collection of tools that will help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).
* A Framework (UI Library based on widgets): A set of reusable user interface (UI) components (buttons, text inputs, sliders, etc.) only me.

### Why use flutter

Rapid app development: Hot reload makes it quick and easy to test, build user interfaces, add features, and fix bugs faster. Experience a second reload, without losing state, on emulators, simulators and devices for iOS and Android.

Beautiful and attractive user interface: Delight your users with Flutter's beautiful built-in widgets in Material Design and Cupertino (iOS-flavor), rich motion application programming interfaces (APIs), scrolling is naturally smooth and platform-aware.

Modern Framework: Easily create your user interfaces with Flutter's modern framework and rich set of platforms, layouts, and widgets. Solve your tough user interface challenges with powerful and flexible APIs for 2D, animation, gestures, effects and more.

### Flutter's Structure

Flutter has created a "new" concept that is widgets. Widgets are the foundation of Flutter, a widget that describes a part of the user interface. Every component like text, image, button or animation, theme, layout or even app is a widget. In Flutter all widgets or interfaces are coded in dart

When a widget changes state, such as due to a user click or an animation, the widget rebuilds itself to the new state. This saves developer time because UI can be described as a state function. We do not have to write extra code to only update the UI when the state changes.

### What is Flutter's feature*?*

* Flutter is now and is a react framework.
* Using Dart programming language is simple and easy to learn.
* Fast application development
* Nice and flexible user interface
* Supports a lot of different widgets.
* Show the same UI on multiple platforms.
* High performance application

## **FlashPython**

### What is Flash python?

Flask is a web framework; it belongs to the type of micro-framework built in Python programming language. Flask allows you to build simple to complex web applications. It can build small APIs, web applications such as websites, blogs, wiki pages or a time-based website or even a commercial website. Flask provides you with the tools, libraries, and technologies to help you do the above.

Flask is a micro-framework. This means that Flask is a standalone environment with little use of other external libraries. Therefore, Flask has the advantage of being lightweight, has very few errors due to less dependencies, and is easy to detect and handle security errors.

### Ideas to use.

Based on the title, it will predict related inns to suggest to users.

Steps to take:

* + - First the data processing:
      * Take the description and the title of the message to form a table.
      * Then delete unnecessary words or words that do not matter to the results.
    - Next, we deal with the correlation between the data.
      * Calculate the correlation: convert the description and headlines that form a table into a numeric matrix.
      * Calculate the correlation of the number matrix.
    - Next compare the correlation and make the appropriate suggestion.

## **Leaflet**

Leaflet is an open-source JavaScript library for building an interactive map application. This is a lightweight library, only about 38KB for the script but has all the features that most developers need.

Leaflet is designed with simplicity, performance, and usability in mind. It also works well on both desktop and mobile platforms, can be extended with dozens of plugins, plus it has a nice API document page, simple but easy to read.

## **VietMap Api**

Providing static maps, Maps API interacts with Vietmap's huge data warehouse. Developers can create great experiences for users on many different application platforms.

## **SignalR**

ASP.NET SignalR is a library for ASP.NET developers to add real-time web functionality to their applications. Real-time web functionality is the ability to have server-side code push content to the connected clients as it happens, in real-time.

SignalR takes advantage of several transports, automatically selecting the best available transport given the client's and server's capabilities. SignalR takes advantage of WebSocket, and HTML5 API that enables bi-directional communication between the browser and server. SignalR will use WebSockets under the covers when it is available, and gracefully fall back to other techniques and technologies when it isn't, while the application code remains the same.

SignalR also provides a simple, high-level API for doing server-to-client RPC (call JavaScript functions in a client's browser from server-side .NET code) in an ASP.NET application, as well as adding useful hooks for connection management, such as connect/disconnect events, grouping connections, authorization.

# **CHAPTER 3 ANALYSIS, and DESIGN**

## **Status survey**

### **Survey system phongtro123**

#### Advantages

* Dynamic interface easy to attract users.
* Support customers to post news, with the service board to show how the position will post, and wherein the website.

#### Defect

* The number of accommodation stars cannot be taken from the inn finder.
* News posted are always charged without free posting, the limit depends on membership level.

### **Reference, and survey website: Cho Tot Nha**

#### Advantages

* Notice when there is new news, activities.
* The saved message feature helps users review the news that needs to be reviewed.
* The website friends feature supports making friends and making chat with each other as a messenger platform.
* You can create your certificate page using the logged-in account.
* Create an on-page ad campaign.

#### Defect

* The number of accommodation stars cannot be taken from the inn finder.
* News posted are always charged without free posting, limit depends on membership level.

### **Reference, and survey site: Mogi**

#### Advantages

* When entering the address, the location will appear on a google map.
* Explore nearby locations.
* Housing broker
* The map function shows the hostels or realms that are zoned and marked.
* English Support.

#### Defect

* The number of accommodation stars cannot be taken from the inn finder.
* News posted are always charged without free posting, the limit depends on membership level.

### **Surveying the system of inns: Homedy**

#### Advantages

* Real estate comparison feature.
* Explore the area.
* Real estate market report.
* See feng shui.
* Loan interest calculation.
* The value of the house.
* Community: help post questions, allow communication between people on the website.

#### Defect

* Information authentication creates trustworthiness.
* There is no interaction between the tenant, and the admin page to respond if there is a problem.
* When posting a message, undergo many steps of operation many times (can perform 1 step aggregating information to be posted).
* There is no multilingual integration.

### **Conclude**

#### System description

##### *User*

* Website management
* Administrator: manage user information, create accounts, provide cost information for posting information, display system data, manage user posts.
* Management staff.
* Marketing advertising: provide the latest information.
* A user searching for news: Are students, people wishing to rent accommodation, or businesses wishing to buy, and sell. Help look up information about the inn quickly when going to an unfamiliar place.
* Post information:
* Personal: The poster can post their news to advertising, based on the cost they pay for the post, can cancel, or upgrade it. Update your information.
* Organizations: can register to post information about hostels, and rental locations they own.

##### *Function*

###### Register / Login

Users can log in with his / her permission to post news on the system.

Login / Registration can be by Facebook or google.

* When registering, will perform confirmation by phone number.

Forgot your password when sending your password via voice message.

###### Search

Search can be by:

* Search bar.
* Search filters: City, county, price, area, geographic location.
* Filter by price, nearest location, latest news, latest news.
* Private or collective news posted.

###### List of posts

* List of news items searched: Picture, the extent of news, name of accommodation, location, area, date of posting, price, and some basic information.
* Pagination (possible): based on the number of news that shows through different pages.
* Hot news, news has just been posted.

###### Private details

* Information, and data about the hostel.
* List of photos of the inn.
* Detailed location of the inn.
* Contact information (phone number, zalo, Facebook, ...)
* Google map indicates the location of the hostel news.
* There is also a list of news regarding the inn's location.

###### Post news

* Post information on the system through the steps of entry.
* Select the geographical location of the hostel on google map.
* Payment by various methods for news.
* List of levels of the news posted.
* Manage to post (cancel news, upgrade news, ...)

###### Support

* Through chatbox reply message.
* Basic guide to posting.

###### History

* Transaction posting of users.
* Fee-charged transactions.

###### Recharge

Users recharge their accounts via the form of payments.

###### Verify users by phone.

Confirm registration, confirm phone number change, or forget password via SMS.

##### *Some other information*

* Manual for the website for both posting information, logging in / registering, recharging, managing information.
* Service table, and information about the different service levels of posting.
* Management of personal information.

##### *Other function*

* Friends: The main purpose is to chat with friends connected through the messenger platform, in the chotot app we can interact directly with friends on messenger and share the good news or hot news. Only registered account holders can use the friend's feature.
* Chat: This feature allows tenants to interact directly with the person who posted, and vice versa, the publisher can interact with the tenant. The chat feature appears when users view a room rental message in addition to how to make a call, they can text vs the poster if the phone is busy. A tenant can send a text message to agree on a meeting time to check the accommodation, negotiate the price, and the rules of the inn.
* Comment on asking questions on the fance page: exchange information, questions, and news.
* Google map: regional search, zoning search, and displaying basic information on google map.
* Phone number verification

#### Major

Customer registration will enter the full name, email address, phone number, password, and confirm password. The system will check if it is valid, it will send a 6-digit code to the phone number of the customer who entered, and the customer will confirm the code. The system will again check if it is valid, it will return to the home page, or the false message will not be reported.

Searching for guests or users can choose to use the search function, search by the keyword you enter, by country, by district, or by type of accommodation. The system will check and return related inns.

Customer login service will log in with Google, Facebook, or enter a phone number. The system will check if the guest is the user who will move to the home page, the admin or the employee will move to the website management page.

Service of posting information, the user will select the posting function, and choose the category of the house, then enter the property information (house name, city name, district address, address, and phone number), and then enter the information. Information to post (rent, house area, direction, legal, news title, property description), then enter the details (a form of the house (front house, original house, alley, ...)., number of rooms, number of bedrooms), next will add pictures from the machine (can choose many pictures), next will choose the price to post by day, month or week, final payment.

Information browsing operation after the posting will be approved by the staff, and the admin, if the employee chooses, and see details, if it is valid, the user will press the browse button, the system will change the message from "false" to "true". if the invalid message will be deleted, next the admin will review which one has been approved by the employee. If the valid message is approved, the admin will press the browse button, the message will change to "showing" status.

## **Modeling required.**

### **Design use-case diagram**

**Figure 3.9:** Use case diagram

A request to build a website where information can be posted, and the inn's information searched.

There are 4 The actors involved in the problem: message viewer, user-user (who has an account), employee, manager:

* Private viewers who do not have an account can visit the website to view news, register an account, and comment on forums.
* User users who are registered users can view posted news, buy gold coins, log out, post news, and comment on forums.
* Staff is the user manager, and post, revenue statistics for management.
* The manager is the last reviewer to post on the site.
* When the user posts the message according to the steps and confirms with the success message, the posted message will be in line and wait for the approval staff, the staff will go to the browsing site to check the information and browse for it. manager, the manager is the last person to approve, agree, the message will be posted on the website. If the post is not valid, it will notify the user to correct it.
* When a user wants to buy Gold, he/she will go to the gold purchase page to top up, the transaction will be done via international VISA or debit card.
* All will be able to view and comment on the talk page of the website.
* Login users can log in with an account and can log in using social networks like Facebook or Google.
* Viewers, when creating a new account, can also easily register through 2 ways: using the personal phone number, after entering the phone number, the message will appear as a 6-digit code and enter 6 digits. In the confirmation section, a message will appear. When creating an account with a social network, the viewer will enter the account, and the password, the system will check, and create an account.

### **Usecase specification**

**Table 3.1:** Usecase specification login phone number

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-1 | | | |
| Use Case No. | DT-1 | Use Case Version | 1.0 |
| Use Case Name | Sign in with phone number | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest   Summary:   * Use this case allows visitors to log into the website to use the advanced services of the site.   Goal:   * The account's role will be confirmed.   Triggers:   * Click the "Login" button.   Preconditions:   * User account has been created. * User account has been authorized. * The user's device was connected to the internet when logging in   Post conditions:   * Success: The account role is changed, and login to the user page * Fail: Display the message "Your password is not correct".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The user selects the login button | The system will redirect users to the login page | | 2 | Guests enter phone number, and password | The data shows up on the information that guests have entered | | 3 | Guests will select the login button | The system notifies the user "login successfully" | | 4 | Admin click "Agree" | The system moves to "Home" |   Alternative Scenario:  Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first | Guest re-enter login information | The system returns to login moon |   **Relationships:**  Business Rules:   * Employees must log in as the user role. | | | |

**Table 3.2:** Usecase specification for social account login

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-2 | | | |
| Use Case No. | DT-2 | Use Case Version | 1.0 |
| Use Case Name | Login with Facebook social account | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest   Summary:   * Use this case allows visitors to log into the website to use the advanced services of the site.   Goal:   * The account's role will be confirmed.   Triggers:   * Click on the button "Login with Facebook”.   Preconditions:   * User account has been created. * User account has been authorized. * The user's device was connected to the internet when logging in   Post conditions:   * Success: The account role is changed, and login to the user page * Fail: Display the message "Your password is not correct".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The user selects the login button | The system will redirect users to the login page | | 2 | Guests choose the login button with Facebook | The system will redirect to the Facebook page | | 3 | The visitor will select the "Continue" button. | The system notifies the user "login successfully", and redirects the user to the home page |   Alternative Scenario:  Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first | Guest re-enter login information | The system returns to login moon |   **Relationships:**  Business Rules:   * Employees must log in as the user role. | | | |

**Table 3.3:** Usecase specification login google account.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-3 | | | |
| Use Case No. | DT-3 | Use Case Version | 1.0 |
| Use Case Name | Sign in with the Google social account | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest   Summary:   * Use this case allows visitors to log into the website to use the advanced services of the site.   Goal:   * The account's role will be confirmed.   Triggers:   * Click on the button "Sign in with Google”.   Preconditions:   * User account has been created. * User account has been authorized. * The user's device was connected to the internet when logging in   Post conditions:   * Success: The account role is changed, and login to the user page * Fail: Display the message "Your password is not correct".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The user selects the login button | The system will redirect users to login page | | 2 | Guests choose to sign in with Google | The system will redirect to the Google page | | 3 | The visitor will select the "Continue" button. | The system notifies the user "login successfully", and redirects the user to the home page |   Alternative Scenario:  Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first | Guest re-enter login information | The system returns to login moon |   **Relationships:**  Business Rules:   * Employees must log in as the user role. | | | |

**Table 3.4:** Usecase specification is registered by phone number.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-4 | | | |
| Use Case No. | DT-4 | Use Case Version | 1.0 |
| Use Case Name | Sign up for an account by phone number | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest   Summary:   * Use this case allows guests to register for an account to log into the account.   Goal:   * The account's role will be confirmed.   Triggers:   * Click the button "Register”.   Preconditions:   * The user's device was connected to the internet when logging in.   Post conditions:   * Success: The account data is saved to the database * Fail: Displays the message "Your account has not been created".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | Guests select the button "Register" | The system will redirect users to the the “Register” page. | | 2 | Guests enter their full name, phone number, password, re-enter the password | The system will display information on the web | | 3 | Guests choose the subscribe button | The system authenticates the phone number and sends a text message to the phone number with a 6-digit code | | 4 | Guests enter confirmation code | The code is displayed on the system, and the system returns the message "You have successfully created an account" | | 5 | Guests select "Continue" | The system will go to "Home". |   Alternative Scenario:  Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first | Guest re-enter login information | The system returns to login moon |   **Relationships:**  Business Rules:   * Employees must log in as the user role | | | |

**Table 3.5:** Usecase specification registered by social account.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-5 | | | |
| Use Case No. | DT-5 | Use Case Version | 1.0 |
| Use Case Name | Sign up for an account with a social account | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest   Summary:   * Use this case allows guests to register for an account to log into the account   Goal:   * The account's role will be confirmed.   Triggers:   * Click the button "Register”.   Preconditions:   * The user's device was connected to the internet when logging in   Post conditions:   * Success: The account data is saved to the database * Fail: Displays the message "Your account has not been created".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | Guests select the button "Register" | The system will redirect users to the the “Register” page. | | 2 | Guests select the button "Register with Facebook" | The system will redirect to the Facebook page | | 3 | Guests enter the username, and password of Facebook | The system receives a response from the Facebook page and displays a message | | 4 | Guests select "Continue" | The system redirects to the home page |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | Guest re-enter login information | The system returns to login moon |   **Relationships:**  Business Rules:   * Employees must log in as the user role. | | | |

**Table 3.6:** Usecase specification registered with Google social account.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-6 | | | |
| Use Case No. | DT-6 | Use Case Version | 1.0 |
| Use Case Name | Sign up for an account with the Google social account | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest   Summary:   * Use this case allows guests to register for an account to log into the account.   Goal:   * The account's role will be confirmed.   Triggers:   * Click the the “Register” button.   Preconditions:   * The user's device was connected to the internet when logging in.   Post conditions:   * Success: The account data is saved to the database. * Fail: Displays the message "Your account has not been created".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | Guests select the button "Register" | The system will redirect users to the “Register” page. | | 2 | Guests select the button "Register with Facebook" | The system will redirect to the Facebook page | | 3 | Guests enter Google account name, and password | The system receives a response from the Google page and displays a message | | 4 | Guests select "Continue" | The system redirects to the home page |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first | Guest re-enter login information | The system returns to login moon |   **Relationships:**  Business Rules:   * Employees must log in as the user role. | | | |

**Table 3.7:** The use-case specification searches for image information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-7 | | | |
| Use Case No. | DT-7 | Use Case Version | 1.0 |
| Use Case Name | Search information by province | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest, User   Summary:   * Use this case for the hostel search.   Goal:   * The actors can search for information based on the search bar in the form of accommodation.   Triggers:   * The The actor selects the type you want to search, click "Search".   Preconditions:   * The user's device was connected to the internet when logging in.   Post conditions:   * Success: The search data will be displayed on the search page. * Fail: Displays the message "No matching data found".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The The actor enters information to search for in the search box | The system displays data on the search box | | 2 | The visitor selects the "province city" button and selects the province | The system will display the province on the website | | 3 | An The actor selects the button "Search" | The system will redirect the The actor to the search page |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first |  |  |   **Relationships:**  Business Rules: | | | |

**Table 3.8:** The use-case specification searches information by accommodation type

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-8 | | | |
| Use Case No. | DT-8 | Use Case Version | 1.0 |
| Use Case Name | Search for information by accommodation type | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest, User   Summary:   * Use this case for the hostel search.   Goal:   * The actors can search for information based on the search bar in the form of accommodation.   Triggers:   * The The actor selects the type you want to search, click "Search".   Preconditions:   * The user's device was connected to the internet when logging in.   Post conditions:   * Success: The search data will be displayed on the search page. * Fail: Displays the message "No matching data found".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The The actor enters information to search for in the search box | The system displays data on the search box | | 2 | The guest selects the button "Accommodation type", and selects the province | The system will display the province on the website | | 3 | An The actor selects the button "Search" | The system will redirect the The actor to the search page |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first |  |  |   **Relationships:**  Business Rules: | | | |

**Table 3.9:** The usecase specification searches for information by region

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-9 | | | |
| Use Case No. | DT-9 | Use Case Version | 1.0 |
| Use Case Name | Search for information by region | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Guest, User   Summary:   * Use this case for the hostel search.   Goal:   * The actors can search for information based on the search bar in the form of accommodation.   Triggers:   * The actor selects the type you want to search, click "Search".   Preconditions:   * The user's device was connected to the internet when logging in.   Post conditions:   * Success: The search data will be displayed on the search page. * Fail: Displays the message "No matching data found".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The actor enters information to search for in the search box | The system displays data on the search box | | 2 | The visitor selects the button "Region" and selects the province | The system will display the province on the website | | 3 | The actor selects the button "Search" | The system will redirect The actor to the search page |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | first |  |  |   **Relationships:**  Business Rules: | | | |

**Table 3.10:** Description of usecase posting.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-10 | | | |
| Use Case No. | DT-10 | Use Case Version | 1.0 |
| Use Case Name | Post news | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User   Summary:   * Use this case for the hostel search.   Goal:   * The actors can search for information based on the search bar in the form of accommodation.   Triggers:   * The actor selects the type you want to search, click "Search".   Preconditions:   * The user's device was connected to the internet when logging in.   Post conditions:   * Success: The search data will be displayed on the search page. * Fail: Displays the message "No matching data found".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | User selects the button "Post news" | The system moves to the posting page | | 2 | User in turn entered data about the inn to post | The system will display information on the website | | 3 | User selects the button "Post news" | The system will save the data to the database |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | The system will return the posting page | The system returns to the posting page |   **Relationships:**  Business Rules: | | | |

**Table 3.11:** Usecase specification see the detailed information.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-11 | | | |
| Use Case No. | DT-11 | Use Case Version | 1.0 |
| Use Case Name | View for details | | |
| Author | Chu Minh Hoang | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User, Guest   Summary:   * Use this case for the hostel search.   Goal:   * The actor sees detailed information about the hostel.   Triggers:   * The actor selects the inn you want to view and clicks on the title or image.   Preconditions:   * The user's device was connected to the internet when logging in.   Post conditions:   * Success: The detailed search data for accommodation information is displayed on the details page. * Fail: Displays the message "No matching data found".   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The actor chooses an inn | The system moves to the details page |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | The system will return the details page but will appear the message "No matching data found". |  |   **Relationships:**  Business Rules: | | | |

**Table 3.12:** The use-case specification sees the posting list

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-12 | | | |
| Use Case No. | DT-12 | Use Case Version | 1.0 |
| Use Case Name | View list of posted news | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User   Summary:   * Use this case allows viewing the list of posted messages.   Goal:   * User can see the news he has posted before.   Triggers:   * Users select "Manage postings."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as a User.   Post conditions:   * Success: The data shows all previously posted rooms. * Fail:   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | Users select "Manage postings" | The system goes to the management page of the posting |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | |  |  |  |   **Relationships:**  Business Rules: | | | |

**Table 3.13:** For the use-case, specification see the service price list.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-13 | | | |
| Use Case No. | DT-13 | Use Case Version | 1.0 |
| Use Case Name | See service price list | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User   Summary:   * Use this case allows seeing the service price list.   Goal:   * User can see the service price of a website.   Triggers:   * User selects "Service price list."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as a User.   Post conditions:   * Success: The data shows all previously posted rooms. * Fail:   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | User selects "Service price list" | The system switches to the price list of posting services |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | |  |  |  |   **Relationships:**  Business Rules: | | | |

**Table 3.14:** Usecase specification see personal information.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-14 | | | |
| Use Case No. | DT-14 | Use Case Version | 1.0 |
| Use Case Name | View personal information | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User   Summary:   * Use this case allows users to see personal information about themselves   Goal:   * User can see personal information.   Triggers:   * User selects "Personal Information."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as a User.   Post conditions:   * Success: The data shows all previously posted rooms. * Fail: message "can't find the right information"   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | User selects "Personal Information" | The system redirects to the personal information page |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | Click on "Personal information" by the user. | A message will appear on the page |   **Relationships:**  Business Rules: | | | |

**Table 3.15:** The use-case specification changes the password.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-15 | | | |
| Use Case No. | DT-15 | Use Case Version | 1.0 |
| Use Case Name | Change the password | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User   Summary:   * Using this case allows the user to change the password.   Goal:   * Users can change passwords.   Triggers:   * User selects "Personal Information."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as a User.   Post conditions:   * Success: Successfully changed password. * Fail: message "password change failed"   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | User selects "Personal Information" | The system redirects to the personal information page | | 2 | User selects "Change password" | The system moves to the password change page | | 3 | Users enter the old password, and the new password, re-enter the new password | The system displays the information just entered | | 4 | User selects "change password" | The system will confirm, and save it to the database, redirect the user to the login page again |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | User selects "Change password" | The system says "Password does not match" |   **Relationships:**  Business Rules: | | | |

**Table 3.16:** Usecase chat specification

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-16 | | | |
| Use Case No. | DT-16 | Use Case Version | 1.0 |
| Use Case Name | Chat | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User   Summary:   * Use this case allows the user to chat with the site administrator.   Goal:   * User can communicate with staff of the site.   Triggers:   * User selects widget in the bottom right corner of the web page.   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as a User.   Post conditions:   * Success: the chat window will appear, and the user performs the chat * Fail: display the frame to send the message via mail, the user can respond by mail   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | User selects widget in the bottom right corner of the web page. | The system displays wiget chat | | 2 | A user enters the message into the message box | Chat system displays information | | 3 | User selects "send" or press Enter | The chat system will display the chat, and send it to the admin |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | User selects widget in the bottom right corner of the web page | The system displays the frame to send the message via mail, the user can respond via mail. |   **Relationships:**  Business Rules: | | | |

**Table 3.17:** The use-case specification is logged out.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-17 | | | |
| Use Case No. | DT-17 | Use Case Version | 1.0 |
| Use Case Name | Log out | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * User, admin, employee   Summary:   * Use this case to allow the actor to log out.   Goal:   * The actors are outputting outside of the website.   Triggers:   * Users select "Log out."   Preconditions:   * The user's device was connected to the internet when logging in. * Login as the actor   Post conditions:   * Success: successfully logged out * Fail: Return to the homepage with the login image still displayed   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | User selects the avatar icon. | The system displays information containing functions | | 2 | User selects the function "Log out" | The system validates and redirects the page to the homepage |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | User selects the avatar icon. | The system displays information containing functions | | 2 | User selects the function "Log out" | The system returns the home page with the login image still displayed. |   **Relationships:**  Business Rules: | | | |

**Table 3.18:** The use-case specification locks the user account.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-18 | | | |
| Use Case No. | DT-18 | Use Case Version | 1.0 |
| Use Case Name | Lock the user account | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Employee   Summary:   * Using this case allows the employee to lock the user account.   Goal:   * Employee locks the user account.   Triggers:   * Users select "Manage account."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as an employee.   Post conditions:   * Success: successfully locked the user account * Fail: account information shows the active user   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | The employee chooses the avatar icon. | The system displays information containing functions | | 2 | The employee chooses the function of "user management" | The system validates and redirects the page to user management | | 3 | The employee chooses the user to lock the account | The system updates the database, and moves to the user management page, the information is "Locked" |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | The employee chooses the avatar icon. | The system displays information containing functions | | 2 | The employee chooses the function of "user management" | The system validates and redirects the page to user management | | 3 | The employee chooses the user to lock the account | The system updates to the database and moves to the user management page, at the user information page is still "active". |   **Relationships:**  Business Rules: | | | |

**Table 3.19:** The use causes them admin specification

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-19 | | | |
| Use Case No. | DT-19 | Use Case Version | 1.0 |
| Use Case Name | Add admin | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Root   Summary:   * Using this case allows Root to add the admin account.   Goal:   * Root adds admin to the website.   Triggers:   * Users select "Manage admin."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as Root.   Post conditions:   * Success: successfully added admin * Fail: add failed   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | Root select the function to add admin | The system moved to the admin add-on page | | 2 | Roots enter the admin information, and password | The system validates and updates the database again |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | Root select the function to add admin | The system moved to the admin add-on page | | 2 | Roots enter the admin information, and password | The system verifies, and updates the database again, after returning the main page, the admin information has not been added |   **Relationships:**  Business Rules: | | | |

**Table 3.20:** The use-case specification fixes admin information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-20 | | | |
| Use Case No. | DT-20 | Use Case Version | 1.0 |
| Use Case Name | Edit admin information | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Root   Summary:   * Using this case allows Root to correct admin account information.   Goal:   * Root fixes admin information on the website   Triggers:   * Users select "Manage admin."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as Root.   Post conditions:   * Success: successfully edited admin information * Fail: Failure to fix admin information   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | Root select the function to edit admin information | The system moved to edit admin information | | 2 | Root fixes admin information | The system validates, and updates the database again, goes to the admin page, and updated data |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | Root select the function to edit admin information | The system moved to edit admin information | | 2 | Root fixes admin information | The system validates and updates the database again, moves to the admin management page, and the data has not been updated |   **Relationships:**  Business Rules: | | | |

**Table 3.21:** Specify the admin key use case

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - DT-21 | | | |
| Use Case No. | DT-21 | Use Case Version | 1.0 |
| Use Case Name | Admin key | | |
| Author | Pham Huynh Thanh Lam | | |
| Date | 12/2020 | Priority | High |
| The actor:   * Root   Summary:   * Using this case allows Root to lock the admin account.   Goal:   * Root lock the admin account into the website   Triggers:   * Users select "Manage admin."   Preconditions:   * The user's device was connected to the internet when logging in. * Log in as Root.   Post conditions:   * Success: successfully edited admin information * Fail: Failure to fix admin information   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | The actor Action | System Response | | 1 | Root selects the function to lock the admin account | The system switches to the admin account key | | 2 | Root the admin account key | The system validates, and updates the database again, goes to the admin page, and updated data |   Exceptions:   |  |  |  | | --- | --- | --- | | No | The actor Action | System Response | | 1 | Root selects the function to lock the admin account | The system switches to the admin account key | | 2 | Root the admin account key | The system validates and updates the database again, moves to the admin management page, and the data has not been updated |   **Relationships:**  Business Rules: | | | |

## **Sequence diagram**



**Figure 3.10:** Login sequence diagram



**Figure 3.11:** Sequence diagram of browsing management



**Figure 3.12:** Sequence diagram register diagram



**Figure 3.13:** Search sequence diagram



**Figure 3.14:** Posting sequence diagram.



**Figure 3.15:** Sequence diagram of phone number change



**Figure 3.16:** Sequence diagram of bill management

## **Database design**

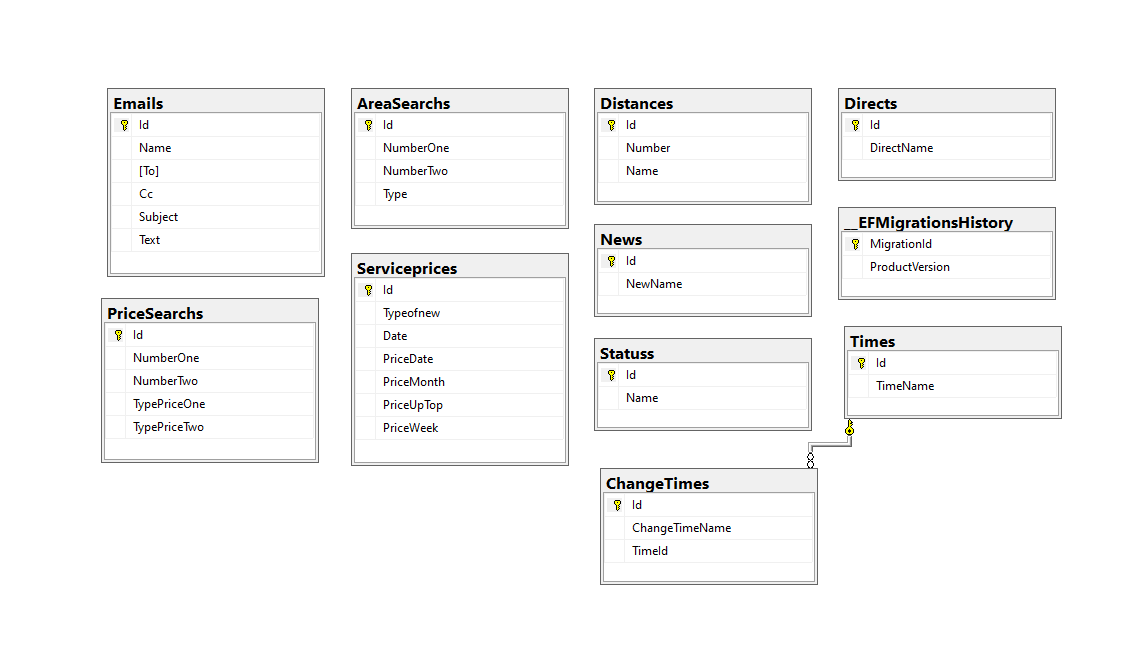
### Entity association model

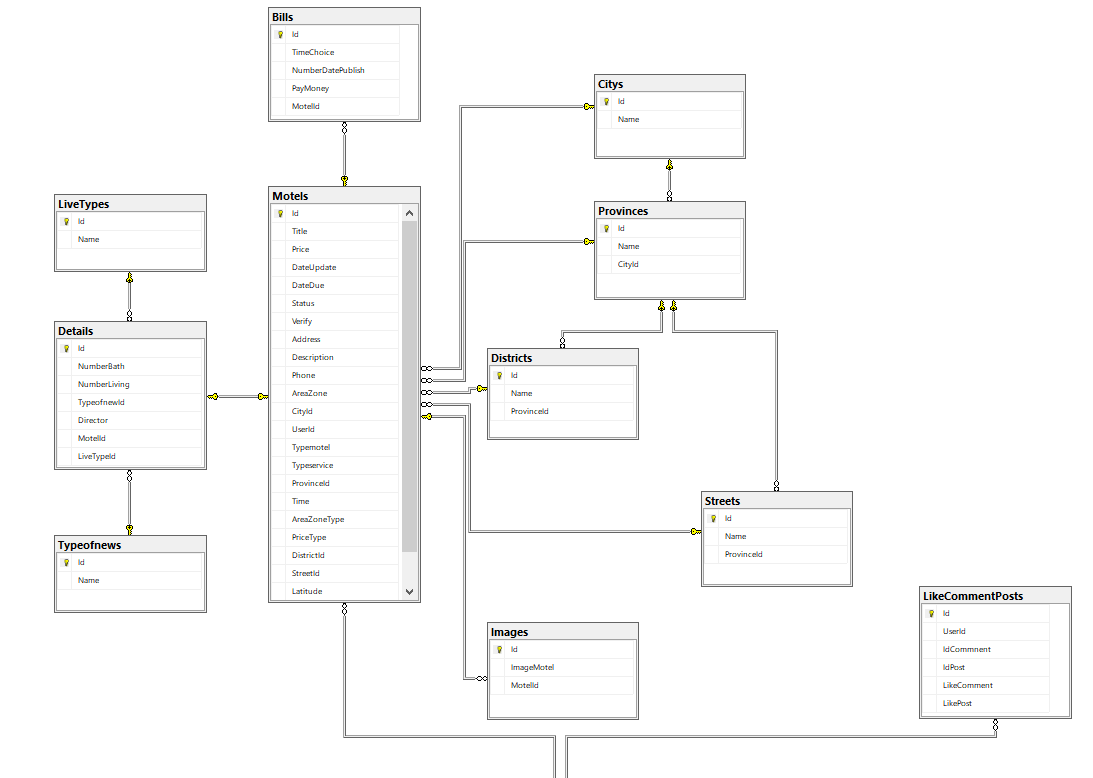
Diagram

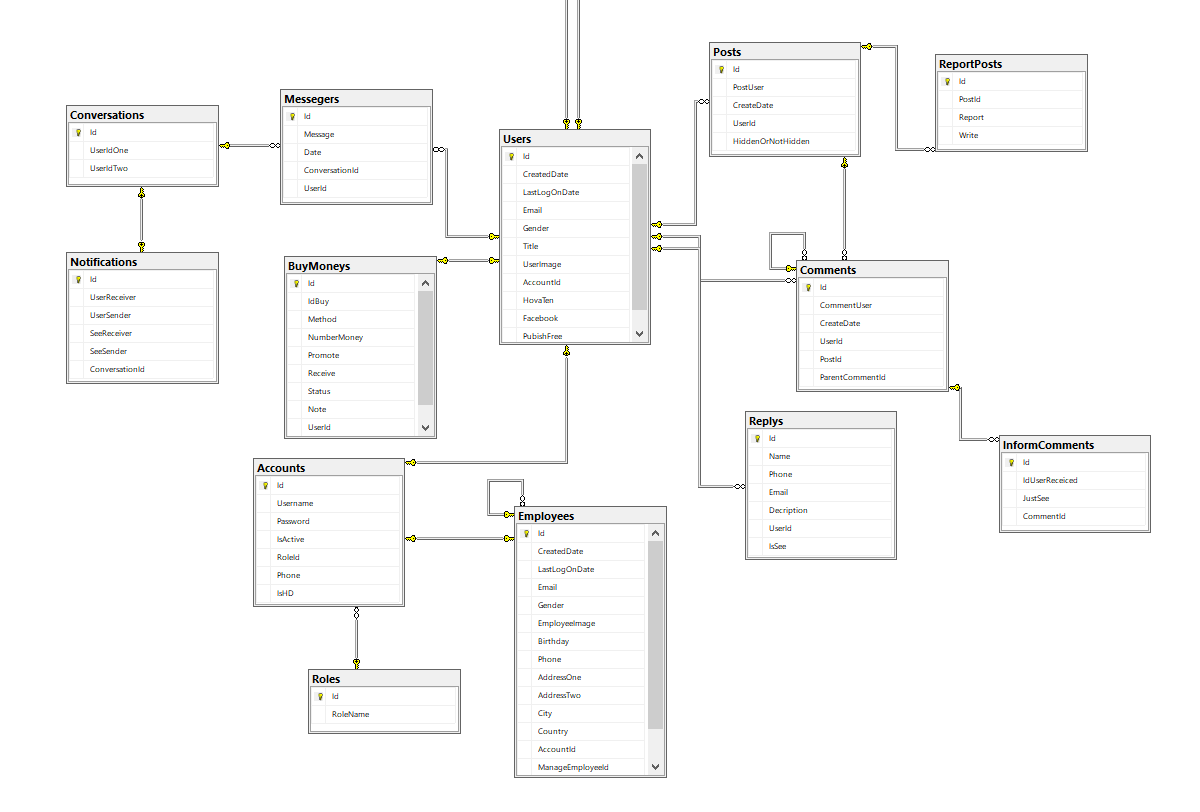
Description automatically generated

**Figure 3.17:** Entity association model

### A relational model of tables in a database







**Figure 3.18:** Relational model of tables in a database

## **Design the theme.**

### User interface

* + - * 1. Home page

**Figure 3.19:** Home page

* + - * 1. Personal information page

**Figure 3.20:** Personal information page

* + - * 1. Post management page

**Figure 3.21:** Post management page

* + - * 1. Newsletter page

**Figure 3.22:** Newsletter page

* + - * 1. Accommodation pages according to each category

**Figure 3.23:** Accommodation pages according to each category

* + - * 1. Product detail page

**Figure 3.24:** Product detail page

* + - * 1. Login page, register

**Figure 3.25:** Login page, post

### Admin, and staff interface

1. Employee management page

**Figure 3.26:** Employee management page

1. Browsing management page

**Figure 3.27:** Browsing management page.

# **CHAPTER 4 PRODUCT INSTALLATION**

## **Font end**

### Application structure web

* **e2e** (end to end): the directory used for testing.
* **node\_modules**: packages to be installed for use in the project.
* **src**: the project's code will be here. In the following section, we will learn more about this directory:
* **app**: The directory containing the application's main module.
* **assets**: directory containing resource files such as CSS, js, image
* **environment**: a folder containing the environment setup files.
* **fav.ico**: file icon, display the icon (logo) of the website on the title bar of the browser.
* **index.html**: main file of the program.
* **main.ts**: This is the first file to be run when your application is run.
* **polyfill.ts**: file used for backward compatibility.
* **style.css**: file style for the application
* **test.ts**: The file contains the test cases for the project
* **tsconfig.app.json**: file that contains the compiled configuration, "instructions" for the compiler code.
* **typings.d.ts**: file that manages definitions in Typescript.
* **angular.json**: contains project information
* **.gitignore**: file contains a list of files, folders not in the commit directory. You can learn more about gitignore here (<https://git-scm.com/docs/gitignore>)
* **karma.conf.js**- The file contains the project requirements for unit testing.
* **package.json:** The file contains information about the libraries used in the project
* **tsconfig.json:** Contains extra rules when the project is translated like where (outer), what is the base path (baseUrl), what is the target version (target),

### The main word processing in font-end

* Paypal: allows creating payments with the amount of money from the website when the user posts the news and connects to the payment page of PayPal.

Text

Description automatically generated

**Figure 4.1:** Script to embed third-party widget tawk. to

* Save the Firebase image: The image will be uploaded to the firebase storage, and then execute get download URL as the URL.

Text

Description automatically generated **Figure 4.2:** The function loads the image on firebase, and uploads the URL link

* Sending messages to phones: thanks to firebase supports sending 6-digit code codes to the phone messages that users subscribe to.

Text

Description automatically generated

**Figure 4.3:** Function to send code to the phone.

## **Back end**

### Project API structure

**Dependencies**: List of NuGet library packages to be installed and used in the project. Even the simplest empty project must use certain library packages. [7]

**Properties**: contains project configuration information. If opened with Visual Studio, you will encounter the familiar graphical interface inherent in all C # projects. In ASP.NET Core, this information is stored in a JSON file that you can directly modify. This new way of saving is useful if you are using another code editor (such as Visual Studio Code). [7]

**appsettings.json**: contains configuration information for application operations, such as connection strings, environment variables, command line parameters, etc. [7]

**Program.cs**: This file contains the Program class that is responsible for the infrastructure configuration of the application. This class also contains the application's entry point. In general, the configuration inside the Program is almost unchanged in every project. If there are no special requirements, you will not need to make any adjustments in the Program class and the Program.cs file. [7]

**Startup.cs**: This file contains classes with configuration methods for application operations, for example, which middleware you will use, what order is arranged in the pipeline middleware. You can also configure what types of services to use, like Dependency Injection, Logging. You will see some simple configurations in the next part of this lesson. Details on how to create a configuration class will see you in another lesson. [7]

**Controllers**: Department is responsible for handling user requests made through the view. From there, the Controller gives appropriate data to the user. Besides, Controller also has the function to connect with the model. [7]

**Migrations**: create a format that binds data to interact with a database

**Models**: This is the part that stores all of the application's data. The model is presented in the form of a database or sometimes simply one [XML file](https://monamedia.co/file-xml-la-gi/) normal. The model clearly shows operations with the database such as allowing viewing, accessing, processing data, ... [7]

### Several Functions, and Procedures

* The API methods:
* Get: get the data from the database, and load it up with the json form of the motel model in the database

|  |
| --- |
| // GET: api / Motels  [HttpGet]  [ActionName ("GetMotels")]  public async Task <ActionResult <IEnumerable <Motel> >> GetMotels ()  {  return await \_context.Motels.Include (e => e.Detail) .Include (e => e.User) .Include (e => e.Images) .ToListAsync ();  } |

* Post: get the json data of the model account, and then save it to the model data, then save it to the database

|  |
| --- |
| // POST: api / Accounts  [HttpPost]  [Route ("Thường")]  public async Task <ActionResult <Account>> PostAccount (Account account)  {  account.IsActive = true;  account.RoleId = 1;  context.Accounts.Add (account);  await \_context.SaveChangesAsync ();  int id = account.Id;  account.User.CreatedDate = DateTime.Now;  account.User.LastLogOnDate = DateTime.Now;  account.User.Gender = true;  \_context.Users.Add (account.User);  return CreatedAtAction ("GetAccount", new {id = account.Id}, account);} |

* Put: get the JSON data of the service prices model, and then update the data based on the id to the database

|  |
| --- |
| // PUT: api / Serviceprices / 5  [HttpPut ("{id}")]  public async Task <IActionResult> PutServiceprice (int id, Service price service price)  {  if (id! = serviceprice.Id) {  return BadRequest ();  }  \_context.Entry (serviceprice) .State = EntityState.Modified;  try{  await \_context.SaveChangesAsync ();  }  catch (DbUpdateConcurrencyException) {  if (! ServicepriceExists (id)) {  return NotFound ();  }  else{  throw;  }  }  return CreatedAtAction ("GetServiceprice", new {id = serviceprice.Id}, serviceprice);} |

* Get {name}: get data by searching by name, and the inns are in display, then return the data motel as json

|  |
| --- |
| // GET: api / Motels / GetMotelByType / name  [HttpGet]  [Route ("GetMotelByType / {name}")]  public async Task <ActionResult <IEnumerable <Motel> >> GetMotelByType (string name)  {  var models = await \_context.Motels.Include (m => m.Detail) .ThenInclude (m => m.Typeofnew) .Include (m => m.City) .Include (m => m.Province) .Include (m => m .Images)  .Where (a => a.Detail.Typeofnew.Name == name && a.Status =="News is showing") .ToListAsync ();  if (models == null) {  return NotFound ();  }  return models;  } |

4.3 Technology

4.3.1 FlashPython

Structure of python flash

Main folders

* + App.py

Home directory to initialize

* + Data.py

Get data from backend via API

* + Models.py

Declare the entity used as Motels

* + Utils.py

Functions to process data

* + remove\_stopwords: remove redundant words
  + count: calculate the similarity of the data
  + recommend: recommend accommodation
  + Requirement.txt

Announcement of libraries and their versions

* + Env

Initialize the environment

* + \_\_pycache\_\_

When you run a program in python, the interpreter compiles it to bytecode first (this is an oversimplification) and stores it in the \_\_pycache\_\_ directory

Libraries used

* import re

In python, there is a built-in package namedreusable to work with regular expressions

| **Jaw** | **Describe** |
| --- | --- |
| findall | Returns a list of matches |
| search | Returns a Match object if any of the values ​​in the string match |
| split | Returns a split list of strings at position match |
| sub | Replace positions that match the expression with another string |

The library is used to process Vietnamese strings.

* import pandas as pd

The pandas library in python is an open source library that effectively supports data manipulation. It is also a powerful data processing and analysis toolkit of the python programming language.

* import nltk

NLTK or Natural Language Toolkit - Natural Language Toolkit, is a library written in Python that supports natural language processing. By providing common language processing mechanisms and techniques, it makes natural language processing easier and faster.

* from nltk.corpus import stopwords

NLTK ships with stopword lists for most languages. Is a library of unrelated words also known as "stop words".

* from sklearn.metrics.pairwise import cosine\_similarity

Calculate the cosine similarity between the samples

* from sklearn.feature\_extraction.text import CountVectorizer

Convert text document collection to token count matrix

* import datetime

Python dates are not a data type of their own, but we can import a module named datetime to work with dates as date objects.

* import requests

The request module allows you to send HTTP requests using Python.

* from flask\_cors import CORS, cross\_origin

This package exposes a Flask extension that by default enables CORS support on all routes, for all origins and methods. It allows parameterization of all CORS headers on a per-resource level

* from flask import Flask, jsonify, request

jsonify serializes data to JavaScript Object Notation (JSON) format, wrapping it in a Response object with application/json mimetype

4.3.2 Leaflet

To use leaflet users will have to install leaflet library After installing the library, declare the scripts to use to display the map



Create a component to display the map

Main file format :

-map.component.css

Contains methods for rendering called in HTML files

-map.component.html

Show display

-map.component.spect.ts

Test code Contains

-map.component.ts

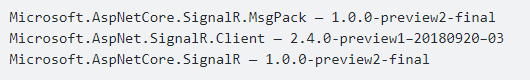
Contains methods to dump data and display the map

4.3.3 SignalR

SignalR is a free and open library that can be used to integrate real-time functionality in web applications. In the site apply SignalR to do direct messaging with each other SignalR is divided into 2 parts: font-end and back-end:

+Back-end:

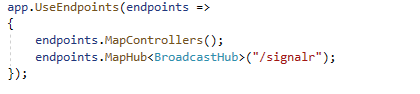
To use SignalR, users must install the SignalR package



Hub instantiation: Class Hub is client like angular will listen for different events Update Startup: Use CORS to agree to connect



To connect font-end use endpoint



+Font-end:

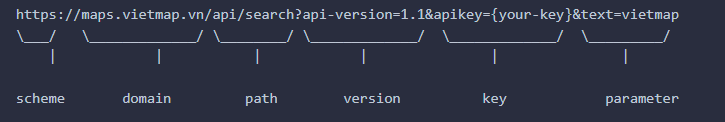
To use SignalR, users must install the SignalR package



Create model, update app.component to listen from backend

4.3.4VietMap:

The team uses the api provided by VietMap to get the location of the customer they want to search. The structure of the api uses



The method used is autocomplete  


Use get method to get parameters

| **Tham số** | **Kiểu** |  |  |  |
| --- | --- | --- | --- | --- |
| text | string |  |  |  |
| focus.point.lat | floating point number |  |  |  |
| focus.point.lon | floating point number |  |  |  |
| boundary.rect.min\_lon | floating point number |  |  |  |
| boundary.rect.max\_lon | floating point number |  |  |  |
| boundary.rect.min\_lat | floating point number |  |  |  |
| boundary.rect.max\_lat | floating point number |  |  |  |
| boundary.circle.lat | floating point number |  |  |  |
| boundary.circle.lon | floating point number |  |  |  |
| boundary.circle.radius | floating point number |  |  |  |
| layers | string |  |  |  |
| size | integer |  |  |  |
| categories | string |  |  |  |

## **Use the third software tawk.to**

Text

Description automatically generated

**Figure 4.4:** Script to embed third-party widget tawk.to.

Using scripts to embed third-party widgets tawk.to into the website, to help customers exchange to the website.

## **AppFlutter**

Includes dart files: API.dart, data.dart, detail.dart, filter.dart, Finding.dart, loading.dart, main.dart, search,dart.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

*Project flutter*

# **CHAPTER 5 CONCLUSION**

## **Result**

After working as a group, a website has been built to publish, and search for accommodation with necessary functions:

* Guest
* Sign up by phone number authentication and google.
* Sign in with Google.
* Search by name, city, county, and accommodation type, see hostel details.
* View information about hostels by category.
* Chat with admin via a third party is tawk. to.
* The user has the same functionality as the guest, and adds the functionality:
* Post information about the hostel through the steps.
* Manage posted news.
* Transaction history.
* Staff
* Browsing (authenticating for admin for final browsing).
* Chat customer feedback via third-party talk. to.
* Admin
* Browse the postings (verify that the information can be posted on the website).
* Employee manager.
* Chat customer feedback via third-party talk. to.
* Recharge management.
* Service management.

## **Advantages**

Design, and build a website for posting news, and searching inns, which have basic functions to manage professionalism for admin as well as a guest to post.

Built separate pages for easy management: pages for admin, pages for guests to see details of the hostel, build documents, and use REST APIs, can easily interact with employees through third parties to know more about the website, integrate easy payment via PayPal.

Third-party images can be saved and downloaded as URLs.

Easy-to-use interface, attracting shoppers.

## **Defect**

The interface is not complete yet, because the website has not completely responsive.

Images render slowly, by the firebase team made by the team using free.

Have not checked the characters for security (password must have at least 1 capital letter, number in the password, ...). Do not install because there is not enough time.

The untidy code in the HTML section separates each specific CSS for each case, leading to hard-to-see source code. Many common CSS files are called index.html.

Payments cannot be made through banks as well as e-wallets in Vietnam because the group does not have enough time to learn and install.

Decentralization is lacking, only the right to log in users, new management rights only for admin, and general staff, not divided into different levels for employees.

The employee's operation has not been checked, due to the faulty code, and not being executed.

## **Hard**

Limited time is not a well-developed website.

Need to register a business account to integrate bank cards in Vietnam

The image is saved on Firebase, so the image speed when rendering on the page is slow.

Not yet implemented the location of the hostel on Google map.

## **Lessons Learned**

Learn algorithms applied in the subject more deeply.

Divide the execution time accordingly.

## **Development**

Next development direction:

* Add a chatbot for the convenience of visitors to interact with the page.
* Hints accurately through your current position
* Payment via Gold dong, recharge by banks in VieNam.

# **REFERENCES**

[1] Refer to the theory of Angular.

https://blog.itnavi.com.vn/angular-la-gi/

[2] Refer to the theory of ASP.NET API

https://www.codelean.vn/2020/02/gioi-thieu-ve-aspnet-web-api.html

[3] Refer to the ASP.NET Entity Framework theory

<https://tuhocict.com/gioi-thieu-tong-quan-kien-truc-cai-dat-entity-framework>

[4] Refer to the Microsoft SQL 2017 theory

[https://en.wikipedia.org/wiki/Microsoft\_SQL\_Server](https://vi.wikipedia.org/wiki/Microsoft_SQL_Server)

[5] Refer to Firebase theory

<https://topdev.vn/blog/firebase-la-gi/>

[6] Refer to the third-party software bat awk. to

https://appnet.edu.vn/tawk-to-phan-mem-chat-truc-tuyen-hieu-qua/

[7] Refer to the project API structure.

https://tuhocict.com/cau-truc-du-an-va-cau-hinh-ung-dung-aspnet-core

# **APPENDIX**

## **Appendix I: Development environment settings**

To download, and install Visual Studio Code, follow these steps:

first. [Download the installation file of Visual Studio Code for Windows](https://quantrimang.com/url?q=aHR0cHM6Ly9kb3dubG9hZC5jb20udm4vdmlzdWFsLXN0dWRpby1jb2RlLTg2MzYw) (installation link https://code.visualstudio.com/).

2. Save the file on the device.

3. Double-click the file to launch the installation of Visual Studio Code on Windows 10.

4. Confirm the terms of the agreement.

5. Click the Next button.

6. Use the default installation location and click the Next button.

7. Use the default Start menu settings and click the Next button.

8. (Optional) Check the option Add “Open with code” action to Windows explorer file context menu.

9. (Optional) Check the option Add “Open with code” action to Windows Explorer directory context menu.

## **Appendix II: Creating an Angular application**

* Step 1: Install Angular libraries and tools: Install Angular CLI, type command at terminates # npm install -g @ angular / CLI
* Step 2: Create an Angular project

Create a directory containing your projects, for example, angular- learning, then run VS Code open that folder to work with (you can also work without openings Code right). FromVS CODEtypeCTRL + `to open the number doorterminate, from there type the following command, to create the basic initialization project # ng new <Project name>

* Step 3: Go to the project# cd <Project Name>
* Step 4: Run the newly created project# ng serve