BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**



MÔN HỌC: **KHÓA LUẬN TỐT NGHIỆP**

*Đề tài****:*** **ỨNG DỤNG HỖ TRỢ HƯỚNG DẪN VIÊN DU LỊCH, GỢI Ý TOUR, CHỌN TOUR, SẢN PHẨM THEO TOUR**

Lớp học phần: DHKTPM15B - 420300314716

GVHD: ThS.Nguyễn Văn Thắng

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Mã SV | Chức vụ |
| 1 | Nguyễn Tiến Đạt | 19512891 |  |
| 3 | Võ Trung Hiếu | 19516751 | Nhóm trưởng |

Thành phố Hồ Chí Minh, 26 tháng 12 năm 2022

Mục lục

[**Chương 1: GIỚI THIỆU ĐỀ TÀI ỨNG DỤNG HỖ TRỢ HƯỚNG DẪN VIÊN DU LỊCH** 3](#_Toc122957591)

[**1.1.** **Tổng quan** 3](#_Toc122957592)

[**1.2.** **Mục tiêu đề tài** 3](#_Toc122957593)

[**1.3.** **Phạm vi đề tài** 3](#_Toc122957594)

[**1.4.** **Mô tả yêu cầu chức năng** 3](#_Toc122957595)

[**1.5.** **Yêu cầu phi chức năng** 3](#_Toc122957596)

[**1.6.** **Kế hoạch thực hiện** 3](#_Toc122957597)

[**1.7.** **Phân chia công việc** 3](#_Toc122957598)

[**Chương 2: CÔNG NGHỆ SỬ DỤNG** 3](#_Toc122957599)

[**2.1** **Cloud Computing với AWS** 3](#_Toc122957600)

[**2.1.1. Amazon EC2.** 4](#_Toc122957601)

[**2.2** **Back-end** 4](#_Toc122957602)

[**2.3** **Front-end** 4](#_Toc122957603)

[**2.4** **Database** 4](#_Toc122957605)

[**2.5** **Kiến trúc phần mềm** 4](#_Toc122957607)

[**CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ PHẦN MỀM** 4](#_Toc122957608)

[**3.1.** **Phân tích yêu cầu bằng UML** 4](#_Toc122957609)

[**3.1.1** **Usecase tổng quát** 4](#_Toc122957610)

[**3.1.2** **Danh sách usecase và mô tả** 4](#_Toc122957611)

[**3.1.3** **Đặc tả Usecase chức năng** 4](#_Toc122957612)

[**3.2.** **Class diagram** 5](#_Toc122957613)

[**CHƯƠNG 4: HIỆN THỰC** 5](#_Toc122957614)

[**4.1.** **Cấu hình phần cứng, phần mềm** 5](#_Toc122957615)

[**4.2.** **Giao diện của hệ thống (mobile)** 5](#_Toc122957616)

[**4.3.** **Kế hoạch và hiện thực kiểm thử hệ thống** 5](#_Toc122957617)

[**CHƯƠNG 5: KẾT LUẬN** 5](#_Toc122957620)

[**5.1.** **Kết quả đạt được** 5](#_Toc122957621)

[**5.1.** **Hạn chế của đồ án** 5](#_Toc122957622)

[**5.1.** **Hướng phát triển** 5](#_Toc122957623)

[**TÀI LIỆU THAM KHẢO** 5](#_Toc122957624)

# **Chương 1: GIỚI THIỆU ĐỀ TÀI ỨNG DỤNG**

## **Tổng quan**

Ngành du lịch có sự đóng góp to lớn vào tổng sản phẩm kinh tế quốc dân ở nước ta. Hiện du lịch được coi là một trong ba ngành kinh tế lớn được chú trọng đầu tư, không ngừng phát triển. Du lịch thúc đẩy hỗ trợ các ngành như giao thông vận tải, bưu chính viễn thông, bảo hiểm, dịch vụ tài chính, lưu trú và ăn uống, đồng thời giải quyết việc làm cho đa số lao động phụ ở các thành phố, thị trấn. Việt Nam nằm trong nhóm những quốc gia có tốc độ tăng trưởng khách du lịch nhanh nhất thế giới. Chính vì vậy ứng dụng hỗ trợ hướng dẫn viên du lịch đã ra đời để đáp ứng những nhu cầu du lịch của người dùng trong và ngoài nươc.

Ứng dụng hỗ trợ hướng dẫn viên du lịch là ứng dụng cho phép gợi ý tour du lịch cho người dùng, người dùng có thể chọn tour, các sản phẩm theo tour và chọn dịch vụ tour để được hướng dẫn khi đi du lịch. Với nhiều tính năng thân thiện, giúp người dùng có thể tìm hiểu thông tin của nơi mà mình đến.

Người dùng cần có một tài khoản để dùng các chức năng như lưu trữ tours đã tham gia, nhưng nơi đã đi, giám sát lịch trình tour của mình. Trong khi đó sau khi hoàn thành tour người dùng cũng có thể đánh giá sao cho tour hoặc bình luận về tour du lịch của mình, tăng tính trải nhiểm.

## **Mục tiêu đề tài**

Mục tiêu chính của ứng dụng là cung cấp cho người dùng thông tin của các tour du lịch, người dùng có thể tham khảo lịch trình và đăng ký tham gia tour du lịch. Người dùng có thể kiểm soát được quá trình và tăng tính trải nghiệm của mình khi tham gia một tour du lịch. Người dùng có thể tìm hiểu kỹ hơn thông tin của từng địa điểm du lịch qua ứng dung và thuận tiện cho việc lựa chọn tour du lịch hơn. Với từng địa điểm người dùng sẽ biết được là họ sẽ ở khách sạn nào, ẩm thực ăn uông tại đó, khu vui chơi, … . Ứng dụng sẽ giúp cho việc thu hút và quảng bá tới khách du lịch trong và ngoài nước về các địa điểm du lịch, di sản, danh lam thắng cảnh, hay Văn hóa ẩm thực của Việt Nam.

## **Phạm vi đề tài**

Ứng dụng được xây dựng để phục vụ cho tất cả những người dùng từ 10 tuổi trở lên có nhu cầu sử dụng để tìm hiểu về các địa điểm du lịch, di sản, danh lam thắng cảnh. Những người có nhu cầu đi du lịch cần tìm hiểu lịch trình của một tour du lịch và đăng ký tham gia một tour du lịch nào đó.

Quản trị viên( Admin): là người dùng có quền quản lý thông tin của các tour du lịch, và thông tin đăng ký tour của khách hàng.

## **Mô tả yêu cầu chức năng**

* + 1. Chức năng dành cho khách hàng:
* Đăng ký tài khoản khách hàng (đăng ký bằng số điện thoại )
* Đăng nhập bằng số điện thoại
* Tìm kiếm thông tin tour:
* Tìm kiếm qua tên tour
* Tìm kiếm tour đang hot
* Tìm kiếm tour qua khu vực
* Xem thông tin và lịch trình tour
* Giám sát tour
* Đăng ký tour
* Quản lý danh sách tour yêu thích
* Đánh dấu tour
* Quản lý thông tin cá nhân
* Đổi mật khẩu tài khoản
  + 1. Chức năng dành cho quản trị viên:
* Đăng nhập tài khoản bằng số điện thoại
* Quản lý thông tin tour:
* Thêm tour
* Xóa tour
* Sửa tour
* Cập nhật lịch trình tour
* Quản lý thông tin đăng ký tour:
* Xem thông tin đăng ký tour
* Hủy thông tin đăng ký tour
* Đánh dấu hoàn thành đăng ký tour
* Quản lý tài khoản khách hàng.
* Đóng tài khoản ( khi một tài khoản vi phạm đăng ký tour du lịch quá 5 lần không thể liên hệ để xác nhận đăng ký tour hoặc có hành vi lợi dụng ứng dụng thì quản trị viên có quền đóng tài khoản).
* Mở tài khoản ( tài khoản của khách hàng nếu đang bị đóng, sau khi được gọi để hỗ xác nhận thì quản trị viên có quền mở tài khoản để tài khoản có thể hoạt động bình thường lại).
  + 1. Chức năng dành cho hướng dẫn viên du lịch:
* Quản lý thông tin tour hướng dẫn:
* Tìm tour hướng dẫn
* Hủy tour hướng dẫn

## **Yêu cầu phi chức năng**

Ứng dụng phải có giao diện thân thiện, bố cục hợp lý, màu sắc phù hợp với dịch vụ du lich, đẹp mắt và dễ sử dụng. Các chức năng phải đảm bảo hoạt động ổn đinh chính xác. Tương thích với nhiều ứng dụng di động nhât có thể đảm bảo trên mỗi loại thiết bị khác nhau thì các thành phần trên giao diện không bị thay đổi.

Đảm bảo các vấn đề cơ bản về bảo mật và xác thực.

Đáp ứng được server chạy được 24/7.

Hệ thống chạy với tốc độ ổn đinh hoặc nhanh chóng.

Đáp ứng được nhu cầu người dùng.

## **Kế hoạch thực hiện**

|  |  |
| --- | --- |
| Thời gian | Công việc |
| Tuần 1 | * Chọn đề tài. * Thảo luận về công việc thực hiện. * Chọn thời gian và địa điểm để làm việc hàng tuần. * Chuẩn bị các nguồn kênh để trao đổi * Chuẩn bị các file báo cáo. |
| Tuần 2 | * Phân tích và làm rõ các yêu cầu ( bao gồm yêu cầu chức năng và phi chức năng )của đề tài. * Lựa chọn các công nghệ và dịch vụ để thực hiện dự án * Xác định các chức năng vầ nghiệp của ứng dụng. |
| Tuần 3 | * Thiết kế các mô hình UML. * Use case Diagram. * Activity. * Sequence. * Class. * Cài đặt cơ sở dữ liệu. * Nhập liệu mẫu cho cơ sở dữ liệu. |
| Tuần 4 | * Code giao diện. * Giao diện đăng ký, đăng nhập. * Giao diện trang chủ |
| Tuần 5 | * Xử lý các chức năng. * Đăng ký, đăng nhập * Quản lý tour |
| Tuần 6 | * Các chức năng liên quan đến đăng ký tour * Đăng ký tour * Quản lý thông tin đăng ký ( Xác nhận, hủy bỏ ) * Các chức năng liên quan giám sát tour * Xem lịch trình * Đánh giá, bình luận tour |
| Tuần 7 | * Các chức năng liên quan đến cập nhật thông tin cá nhân. * Đổi mật khẩu. |
| Tuần 8 | * Trang admin. * Xử lý các lỗi hiện có. * Hoàn thiện đồ án. |
| Tuần 9 | * Deploy chương trình. * Khảo sát lại các chức năng. |
| Tuần 10 | * Viết báo cáo, tài liệu. * Chuẩn bị báo cáo đồ án. |

## **Phân chia công việc**

|  |  |
| --- | --- |
| Thành viên | Công việc thực hiện |
| Nguyễn Tiến Đạt | Phân tích thiết kế và viết báo cáo, tài liệu.  Báo cáo kết quả hàng tuần, thiết kế cơ sở dữ liệu.  Thiết kế giao diện Đăng nhập, đăng ký của app. Code các chức năng chính dành cho quản trị viên. |
| Nguyễn Trung Hiếu | Tìm kiếm và chọn lọc các thành phần cho ứng dụng. Code các chức năng chính dành cho khách hàng. |

# **Chương 2: CÔNG NGHỆ SỬ DỤNG**

## **Cloud Computing với AWS**

### **2.1.1. Amazon EC2.**



Hình 2.1.1 Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (Amazon EC2) là một cơ sở hạ tầng điện toán đám mây được cung cấp bởi Amazon Web Services (AWS) giúp cung cấp tài nguyên máy tính ảo hoá theo yêu cầu.

Amazon EC2 cung cấp các ứng dụng máy tính ảo hoá có thể mở rộng về khả năng xử lý cùng các thành phần phần cứng ảo như bộ nhớ máy tính (ram), vi xử lý, linh hoạt trong việc lựa chọn các phân vùng lưu trữ dữ liệu ở các nền tảng khác nhau và sự an toàn trong quản lý dịch vụ bởi kiến trúc ảo hoá đám mây mạnh mẽ của AWS.

Amazon EC2 sẽ cung cấp một hoặc máy chủ ảo có thể kết hợp với nhau để dễ dàng triển khai ứng dụng nhanh nhất và đảm bảo tính sẵn sàng cao nhất. Thậm chí về mặt thanh toán bạn dễ dàng biết được các mức chi phí cần thanh toán dựa trên thông tin tài nguyên bạn sử dụng.

Lợi ích của EC2.

* Amazon EC2 cho phép bạn tăng hoặc giảm nhu cầu tài nguyên trong vòng vài phút, không cần phải là hàng giờ hoặc nhiều ngày. Bạn có thể thực hiện một, hàng trăm, thậm chí hàng ngàn trường hợp máy chủ cùng một lúc. Bạn cũng có thể sử dụng cơ chế Auto Scaling để duy trì sự sẵn có của cụm máy chủ EC2 và tự động mở rộng ứng dụng của bạn lên hoặc xuống tùy thuộc vào nhu cầu của mình để tối đa hóa hiệu suất và giảm thiểu chi phí.
* Bạn có toàn quyền kiểm soát quản lý các tính năng của mình bao gồm truy cập tài khoản AWS Root và khả năng tương tác với họ như bạn sẽ làm với bất kỳ máy nào. Bạn có thể dừng bất kỳ dịch vụ nào trong khi vẫn giữ lại dữ liệu trên phân vùng khởi động, và sau đó khởi động lại cùng một đối tượng sử dụng các API dịch vụ Web. Các hoạt động có thể được khởi động lại từ xa bằng cách sử dụng các API dịch vụ web, và bạn cũng có quyền truy cập vào đầu ra của dịch vụ đó.
* Bạn có thể lựa chọn nhiều loại instance , các hệ điều hành, và các gói phần mềm. Amazon EC2 cho phép bạn chọn cấu hình bộ nhớ, CPU, bộ nhớ instance, và kích thước phân vùng khởi động với mục tiêu tối ưu cho sự lựa chọn của hệ điều hành và ứng dụng. Ví dụ, sự lựa chọn của hệ điều hành bao gồm nhiều phân phối Linux và Microsoft Windows Server .
* Amazon EC2 cung cấp một môi trường đáng tin cậy cao. Dịch vụ này chạy trong cơ sở hạ tầng mạng đã được kiểm chứng của Amazon và các trung tâm dữ liệu đầy đủ tiêu chuẩn cao. Thỏa thuận về mức độ uptime dịch vụ của Amazon EC2 là 99,95% sẵn có cho mỗi vùng Amazon EC2.
* Bảo mật đám mây tại AWS là ưu tiên cao nhất. Là khách hàng của AWS, bạn sẽ được hưởng lợi từ trung tâm dữ liệu và kiến trúc mạng được xây dựng để đáp ứng yêu cầu của các tình huống nhạy cảm nhất.
* Amazon EC2 mang đến cho bạn những lợi ích tài chính của quy mô Amazon. Bạn phải trả một tỷ lệ rất thấp cho khả năng thanh toán bạn thực sự chi tiêu nếu bạn tính toán phù hợp.
* Dễ khởi động.

## **Back-end**

## **2.2.1. Spring boot**



*Hình 2.2.1: Hình logo hiện nay của framework spring boot*

Spring Boot là một Java framework được phát triển bởi Pivital Team dựa trên Java framework mã nguồn mở để tạo ra các microservice, nhằm mục đích xây dựng các ứng dụng Spring độc lập một cách nhanh chóng và có khả năng thực thi ngay.

Sau khi sử dụng Spring Boot để lập trình ứng dụng Spring, bạn có thể chạy ngay ứng dụng của mình mà không cần phải config (cấu hình) quá nhiều.

## **Front-end**

## **2.3.1. React-Native**

React Native là một framework mã nguồn mở được sáng tạo bởi Facebook. Nó được sử dụng để phát triển ứng dụng di động Android, iOS, Web và UWP bằng cách cho phép các nhà phát triển sử dụng React cùng với môi trường ứng dụng gốc.

Icon

Description automatically generated

Hình 2.3.2. React Native

React Native cho phép bạn tạo các ứng dụng dành cho thiết bị di động chỉ sử dụng JavaScript. Nó sử dụng thiết kế tương tự như React, cho phép bạn tạo giao diện người dùng di động phong phú từ các thành phần khai báo. Với React Native, bạn không tạo ứng dụng web di động, ứng dụng HTML5 hoặc ứng dụng kết hợp; bạn xây dựng một ứng dụng di động thực sự không thể phân biệt được với một ứng dụng được tạo bằng Objective-C hoặc Java. React Native sử dụng các khối xây dựng giao diện người dùng cơ bản giống như các ứng dụng iOS và Android thông thường. Bạn chỉ cần ghép các khối xây dựng đó lại với nhau bằng JavaScript và React.

Tính năng React Native

* React - Đây là một Framework để xây dựng ứng dụng web và thiết bị di động sử dụng JavaScript.
* Gốc - Bạn có thể sử dụng các thành phần gốc do JavaScript kiểm soát.
* Nền tảng - React Native hỗ trợ nền tảng IOS và Android.

Ưu điểm của React Native

* JavaScript - Bạn có thể sử dụng kiến ​​thức JavaScript hiện có để xây dựng các ứng dụng di động gốc.
* Chia sẻ mã - Bạn có thể chia sẻ hầu hết mã của mình trên các nền tảng khác nhau.
* Cộng đồng - Cộng đồng xung quanh React và React Native rất lớn và bạn sẽ có thể tìm thấy bất kỳ câu trả lời nào bạn cần.

Giới hạn React Native

* Thành phần gốc - Nếu bạn muốn tạo chức năng gốc chưa được tạo, bạn sẽ cần viết một số mã dành riêng cho nền tảng.

## **Database**

## **2.4.1. Cloud firebase storage**

Cơ sở dữ liệu thời gian thực Firebase là cơ sở dữ liệu NoSQL được lưu trữ trên đám mây cho phép bạn lưu trữ và đồng bộ hóa dữ liệu giữa những người dùng của mình trong thời gian thực.

Cloud Firestore cho phép bạn lưu trữ, đồng bộ hóa và truy vấn dữ liệu ứng dụng ở quy mô toàn cầu.

## **Kiến trúc phần mềm**

### **Kiến trúc MVC**

Model – view – controller là một mẫu kiến trúc phần mềm thường được sử dụng để phát triển giao diện người dùng chia logic chương trình liên quan thành ba phần tử được kết nối với nhau. Điều này được thực hiện để tách các trình bày nội bộ của thông tin khỏi cách thông tin được trình bày và chấp nhận từ người dùng.

* Model (dữ liệu): Quản lí xử lí các dữ liệu.
* View (giao diện): Nới hiển thị dữ liệu cho người dùng.
* Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.

Diagram

Description automatically generated*Hình 2.5-1 : Mô hình MVC Architecture Pattern*

**Luồng xử lý trong của mô hình MVC, bạn có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:**

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
* Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

Diagram

Description automatically generated*Hình 2.5-2: Hình ảnh mô tả luồng sử lý của mô hình MVC*

**Ưu điểm mô hình MVC**

* Đầu tiên, nhắc tới ưu điểm mô hình MVC thì đó là băng thông (Bandwidth) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.
* Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.
* Một lợi thế chính của MVC là nó tách biệt các phần Model, Controller và View với nhau.
* Sử dụng mô hình MVC chức năng Controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
* Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.
* Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
* Hỗ trợ TTD (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won test case.
* Phiên bản mới nhất của MVC hỗ trợ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ View của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.

**Nhược điểm mô hình MVC**

* Bên cạnh những ưu điểm MVC mang lại thì nó cũng có một số nhược điểm cần khắc phục.
* MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian.
* Không thể Preview các trang như ASP.NET.
* Khó triển khai.

**Vì sao nên sử dụng mô hình MVC?**

* Quy trình phát triển nhanh hơn
* MVC hỗ trợ phát việc phát triển nhanh chóng và song song. Nếu một mô hình MVC được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên View và một developer khác có thể làm việc với Controller để tạo logic nghiệp vụ cho ứng dụng web đó.
* Do đó, ứng dụng mô hình MVC có thể được hoàn thành nhanh hơn ba lần so với các ứng dụng mô hình khác.
* Khả năng cung cấp nhiều chế độ view
* Trong mô hình MVC, bạn có thể tạo nhiều View cho chỉ một mô hình. Ngày nay, nhu cầu có thêm nhiều cách mới để truy cập ứng dụng và đang ngày càng tăng. Do đó, việc sử dụng MVC để phát triển chắc chắn là một giải pháp tuyệt vời.
* Hơn nữa, với phương pháp này, việc nhân bản code rất hạn chế. Vì nó tách biệt dữ liệu và logic nghiệp vụ khỏi màn hình.
* Các sửa đổi không ảnh hưởng đến toàn bộ mô hình
* Đối với bất kỳ ứng dụng web nào, người dùng có xu hướng thay đổi thường xuyên. Bạn có thể quan sát thông qua những thay đổi thường xuyên về màu sắc, font chữ, bố cục màn hình. Hay là thêm hỗ trợ thiết bị mới cho điện thoại hay máy tính bảng…
* Việc thêm một kiểu view mới trong MVC rất đơn giản. Vì phần Model không phụ thuộc vào phần View. Do đó, bất kỳ thay đổi nào trong Model sẽ không ảnh hưởng đến toàn bộ kiến trúc.
* MVC Model trả về dữ liệu mà không cần định dạng
* MVC pattern có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.
* Hỗ trợ kỹ thuật Asynchronous
* Kiến trúc MVC có thể được tích hợp với cả JavaScript Framework. Có nghĩa là, các ứng dụng MVC có thể hoạt động ngay cả với các file PDF, trình duyệt riêng cho web hay các widget trên desktop.

# **CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ PHẦN MỀM**

## **Phân tích yêu cầu bằng UML**

### **Usecase tổng quát**

Diagram, schematic

Description automatically generated

*Hình 3.1.1: Hình ảnh sơ đồ usecase tổng quát*

### **Danh sách usecase và mô tả**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Tên usecase** | **Mô tả ngắn gọn usecase** | **Ghi chú** |
| UC001 | Đăng ký | Khách hàng: đăng ký tài khoản khách hàng bằng số điện thoại để có thể đăng nhập vào ứng dụng. | Có kiểm tra thông tin nhập vào và xác thực số điện thoại |
| UC002 | Đăng nhập | Khách hàng: Đăng nhập bằng số tài khoản để sử dụng các dịch vụ.  Quản trị viên: Đăng nhập để quản lý hệ thống.  Hướng dẫn viên: Đăng nhập để theo dõi lịch trình và thông tin tour | Có kiểm tra thông tin nhập vào và xác thực số điện thoại nếu chưa xác thực khi đăng ký |
| UC003 | Đăng ký tour | Khách hàng chọn tour và nhấn đăng ký, hệ thống sẽ hiển thị form đăng ký bao gồm các thông tin tham gia tour và ngày đi và ngày về theo mong muốn. |  |
| UC004 | Cập nhật thông tin tài khoản | Khách hàng sau khi đăng nhập có thể cập nhật các thông tin cá nhân, hình ảnh của mình. |  |
| UC005 | Xóa tài khoản | Khách hàng có thể xóa tài khoản của mình khỏi hệ thống. |  |
| UC006 | Đánh dấu tour | Khách hàng có thể đánh dấu tour yêu thích của mình, sau đó có thể xem lại qua danh sách tour yêu thích. |  |
| UC007 | Xem thông tin tour | Khách hàng có thể xem thông tin tour, lịch trình của từng tour để lựa chọn tour mà họ thấy phù hợp nhất. |  |
| UC008 | Tìm kiếm tour | Khách hàng tìm tour qua: tên tour, tên khu vực, địa điểm, độ hot của tour. |  |
| UC009 | Giám sát tour | Khách hàng có thể giám sát lịch trình của tour và mức độ hoàn thành tour khi tham gia một tour. |  |
| UC010 | Lấy lại mật khẩu | Khách hàng quên mật khẩu đăng nhập, có thể lấy lại mật khẩu qua chức năng này. Hệ thống sẽ gửi về cho khách hàng một mã OTP hoặc một mật khẩu mới qua số điện thoại. |  |
| UC012 | Xác nhận thông tin đăng ký tour | Sau khi khách hàng đăng ký tour, quản trị viên sẽ liên hệ theo thông tin đăng ký của khách hàng, nếu khách hàng xác nhận đăng ký thì quản trị viên sẽ đánh dấu chấp nhận đơn đăng ký qua chức năng này. |  |
| UC013 | Đăng ký hướng dẫn viên cho tour | Sau khi thông tin đăng ký được chấp nhận, quản chị viên sẽ đăng ký hướng dẫn viên du lịch cho tour đó. |  |
| UC014 | Hủy thông tin đăng ký tour | TH1: Nếu thông tin đăng ký không hợp lệ thì quản trị viên có thể hủy thông tin đăng ký,  TH2: không thể liên hệ khách hàng đăng ký thì sau 5 ngày quản trị viên có thể hủy đơn đăng ký này. |  |
| UC015 | Đóng tài khoản | Nếu khách hàng có hành vi vi phạm, phá hoại hoặc có mục đích lợi dụng ứng dung. Quản trị viên có thể đóng tài khoản của khách hàng và gửi thông báo qua số điện thoại. |  |
| UC016 | Mở tài khoản | Khi tài khoản của khách hàng bị đóng, khách hàng có thể liên hệ quản trị viên để được mở lại tài khoản. |  |
| UC017 | Tìm tour đã đăng ký hướng dẫn | Hướng dẫn viên có thể tìm kiếm các thông tin tour mình phải hướng dẫn. |  |
| UC018 | Hủy tour đã đăng ký hướng dẫn | Hướng dẫn viên có thể tự hủy tour đã được đăng ký hướng dẫn. | Chỉ được hủy trước một tuần khi tour diên ra. |

### **Đặc tả Usecase chức năng**

#### **3.1.3.1 Đăng ký**

Mô tả use case UC001

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: | Khách hàng đăng ký tài khoản | |
| Mô tả: | Khách hàng: đăng ký tài khoản khách hàng bằng số điện thoại để có thể đăng nhập vào ứng dụng. | |
| Tác nhân: | Khách hàng | |
| Điều kiện trước: |  | |
| Điều kiện sau: | Hiển thị thông báo đăng ký thành công hoặc thông báo lỗi | |
|  | Khách hàng | Hệ thống |
| Luồng sự kiện chính (Basic flows) | 1. Chọn đăng ký | 1. Hiển thị form thông tin khách hàng |
|  | 1. Điền các thông tin vào form |  |
|  | 1. Chọn ĐĂNG KÝ | 1. Thông báo đăng ký thành công |
| Luồng sự kiện phụ (Alternative Flows): | 1. Chọn HỦY | 1. Quay lại |
| Luồng sự kiện ngoại lệ (Exception Flows): |  | 1. Hiển thị thông báo đăng ký không thành công |

Bảng 3.1.3.1 Đặc tả của use case UC003 – Đăng ký

**Biểu đồ**

* 1. **Sơ đồ activity:**

*Hình 3.1.3.1a: Sơ đồ activity*

* 1. **Sơ đồ sequence :**

*Hình 3.1.3.1b: Sơ đồ activity*

#### **3.1.3.2 Đăng nhập**

Mô tả use case UC002

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: | Người dùng đăng nhập | |
| Mô tả: | Khách hàng: Đăng nhập bằng số tài khoản để sử dụng các dịch vụ.  Quản trị viên: Đăng nhập để quản lý hệ thống.  Hướng dẫn viên: Đăng nhập để theo dõi lịch trình và thông tin tour | |
| Tác nhân: | Khách hàng, quản trị viên, hướng dẫn viên | |
| Điều kiện trước: |  | |
| Điều kiện sau: | Hiển thị thông báo đăng nhập thành công hoặc thông báo lỗi | |
|  | Khách hàng | Hệ thống |
| Luồng sự kiện chính (Basic flows) | 1. Chọn đăng nhập | 1. Hệ thống hiển thị form thông tin |
|  | 1. Điền thông tin vào form |  |
|  | 1. Chọn đăng ký | 1. Hiển thị thông báo đăng ký thành công |
| Luồng sự kiện phụ (Alternative Flows): | 1. Chọn HỦY | 1. Quay lại |
| Luồng sự kiện ngoại lệ (Exception Flows): |  | 1. Hiển thị thông báo đăng ký không thành công |

Bảng 3.1.3.2 Đặc tả của use case UC002 – Đăng nhập

**Biểu đồ**

* 1. **Sơ đồ activity:**

*Hình 3.1.3.2a: Sơ đồ activity*

* 1. **Sơ đồ sequence :**

*Hình 3.1.3.2b: Sơ đồ activity*

#### **3.1.3.3 Đăng ký tour**

Mô tả use case UC003

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: | Khách hàng đăng ký tour | |
| Mô tả: | Khách hàng chọn tour và nhấn đăng ký, hệ thống sẽ hiển thị form đăng ký bao gồm các thông tin tham gia tour và ngày đi và ngày về theo mong muốn. | |
| Tác nhân: | Khách hàng | |
| Điều kiện trước: | Khách hàng phải đăng nhập thành công | |
| Điều kiện sau: | Hiển thị thông báo đăng ký thành công hoặc thông báo lỗi | |
|  | Khách hàng | Hệ thống |
| Luồng sự kiện chính (Basic flows) | 1. Chọn tour cần đăng ký | 1. Hiển thị thông tin tour được chọn |
|  | 1. Chọn chức năng đăng ký | 1. Hệ thống hiển thị form đăng ký thông tin khách hàng |
|  | 1. Điền các thông tin vào form đăng ký thông tin khách hàng |  |
|  | 1. Chọn ĐĂNG KÝ | 1. Hiển thị thông báo thành công |
| Luồng sự kiện phụ (Alternative Flows): | 1. Chọn HỦY | 1. Quay lại bước 2 |
| Luồng sự kiện ngoại lệ (Exception Flows): |  | 1. Hiển thị thông báo đăng ký không thành công |

Bảng 3.1.3.3 Đặc tả của use case UC003 – Đăng ký tour

**Biểu đồ**

* 1. **Sơ đồ activity:**

*Hình 3.1.3.3a: Sơ đồ activity*

* 1. **Sơ đồ sequence :**

*Hình 3.1.3.3b: Sơ đồ activity*

## **Class diagram**

*Hình 3.2: Sơ đồ Class diagram*

# **CHƯƠNG 4: HIỆN THỰC**

## **Cấu hình phần cứng, phần mềm**

**Các dịch vụ của AWS**

**Cấu hình phần cứng phần mềm của server:**

**Cấu hình phần cứng phần mêm của Client trong hệ thống:**

## **Giao diện của hệ thống (mobile)**

## **Kế hoạch và hiện thực kiểm thử hệ thống**

### **4.4.1 Kế hoạch kiểm thử :**

### **4.4.2 Hiện thực kiểm thử**

# **CHƯƠNG 5: KẾT LUẬN**

## **Kết quả đạt được**

## **Hạn chế của đồ án**

## **Hướng phát triển**

# **TÀI LIỆU THAM KHẢO**

[1] https://rnfirebase.io/firestore/usage

[2] https://reactnative.dev/docs/getting-started