**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THỒNG**

----------o0o----------



**Đồ án III: Định Hướng Công Nghệ**

**Đề tài :**

***Xây dựng website đặt tour du lịch trực tuyến***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sinh viên:** | **Vũ Thị Phương– MSSV: 20156281** | |  |
| **Giảng viên hướng dẫn:** | **ThS. Đỗ Tuấn Anh** |  | |

Hà Nội – Tháng 12/2018

# LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn tới các thầy cô trong trường Đại học Bách Khoa Hà Nội nói chung và các thầy cô trong Viện Công Nghệ Thông Tin và Truyền Thông, bộ môn Khoa Học Máy Tính nói riêng đã tận tình giảng dạy, truyền đạt vốn kiến thức quý báu cho em trong những năm học tập tại trường.

Đặc biệt, em xin cảm ơn thầy Đỗ Tuấn Anh đã trực tiếp hướng dẫn, giúp đỡ và góp ý kiến để em hoàn thành đồ án này.

Hà Nội, ngày 20 tháng 12 năm 2018

Sinh viên: Vũ Thị Phương

# 

# MỤC LỤC

[LỜI CẢM ƠN 4](#_Toc514883090)

[MỤC LỤC 5](#_Toc514883091)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 9](#_Toc514883092)

[DANH MỤC CÁC BẢNG 7](#_Toc514883093)

[DANH MỤC HÌNH ẢNH 7](#_Toc514883094)

[CHƯƠNG I. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP 10](#_Toc514883095)

[1.1 Đặt vấn đề 10](#_Toc514883096)

[1.1.1 Lý do chọn đề tài 10](#_Toc514883097)

[1.1.2 Mục đích của đề tài 10](#_Toc514883098)

[1.2 Định hướng giải pháp 10](#_Toc514883099)

[1.2.1 Nội dung đề tài 10](#_Toc514883100)

[1.2.2 Lựa chọn công nghệ 11](#_Toc514883101)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 12](#_Toc514883102)

[2.1 Giới thiệu Laravel framework 12](#_Toc514883103)

[2.1.1 Khái niệm 12](#_Toc514883104)

[2.1.2 Tại sao chọn Laravel framework 14](#_Toc514883105)

[2.2 Mô hình MVC 14](#_Toc514883106)

[2.3 jQuery, boostrap 16](#_Toc514883107)

[2.3.1 jQuery 16](#_Toc514883108)

[2.3.2 Boostrap 17](#_Toc514883109)

[2.3.3 Hệ quản trị cơ sở dữ liệu MySQL 18](#_Toc514883110)

[CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 19](#_Toc514883111)

[3.1 Xác định yêu cầu hệ thống 19](#_Toc514883112)

[3.2 Thiết kế ứng dụng web phía client 20](#_Toc514883113)

[3.2.1 Biểu đồ Use Case tổng quát 20](#_Toc514883114)

[3.2.2 Đặc tả một số chức năng chính 22](#_Toc514883115)

[3.2.3 Mô hình hóa hoạt động của hệ thống 23](#_Toc514883119)

[3.2.4 Mô hình hóa sự tương tác 24](#_Toc514883120)

[3.2.5 Thiết kế lớp (Class) 25](#_Toc514883121)

[3.3 Thiết kế ứng dụng web quản trị 26](#_Toc514883122)

[3.3.1 Biểu đồ Use Case tổng quát 26](#_Toc514883123)

[3.3.2 Đặc tả một số chức năng chính 28](#_Toc514883124)

[3.3.3 Mô hình hóa hoạt động của hệ thống 31](#_Toc514883128)

[3.3.4 Mô hình hóa sự tương tác 32](#_Toc514883129)

[3.3.5 Thiết kế lớp (class) 33](#_Toc514883130)

[3.4 Xây dựng ứng dụng và triển khai 34](#_Toc514883131)

[3.4.1 Thiết kế cơ sở dữ liệu 34](#_Toc514883132)

[3.4.2 Thiết kế giao diện 38](#_Toc514883133)

[CHƯƠNG IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 47](#_Toc514883135)

[4.1 Các vấn đề được giải quyết 47](#_Toc514883137)

[4.2 Hướng phát triển trong tương lai 47](#_Toc514883138)

[TÀI LIỆU THAM KHẢO 48](#_Toc514883139)

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

|  |  |
| --- | --- |
| **Từ viết tắt, thuật ngữ** | **Cụm từ đầy đủ** |
| CSS | Cascading Style Sheets |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer Protocol |
| MVC | Model – View - Controller |
| CSDL | Cơ sở dữ liệu |
| PHP | Hypertext Preprocessor |
| XML | Extensible Markup Language |
| AJAX | Asynchronous JavaScript and XML |
| DOM | Document Object Model |
| JSON | JavaScript Object Notation |

# CHƯƠNG I. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

## Đặt vấn đề

### **1.1.1 Lý do chọn đề tài**

Dựa trên vấn đề về nhu cầu đi du lịch của mọi người hiện nay, cùng với những ưu điểm vượt trội của Laravel framework nên em muốn xây dựng một website tour du lịch bằng Laravel. Website này giúp khách hàng có thể tìm kiếm, tham khảo, đặt các tour du lịch của công ty và giúp người quản trị dễ dàng quản lý các chương trình tour, dễ dàng cập nhật thay đổi để website của công ty luôn luôn mới mẻ, thu hút khách hàng đến với website. Với việc xây dựng website bằng Laravel giúp em hiểu rõ hơn về Larvel framework.

### **1.1.2 Mục đích của đề tài**

Tìm hiểu và xây dựng website tour du lịch cho phép người dùng có thể đặt tour, tìm kiếm các tour du lịch khác nhau dễ dàng, nhanh chóng, giúp nhân viên có thể duyệt đơn đặt tour và xem các thống kê trong công ty, giúp người quản lý có thể quản lý các dịch vụ của công ty đơn giản hơn.

## Định hướng giải pháp

Hiện nay nước ta có rất nhiều website tour du lịch như: <https://dulichviet.com.vn>, <https://www.tugo.com.vn/>, <http://newstartour.com>, <http://dulichhanoi.vn>… Đa số các trang web này có cấu trúc và tổ chức thông tin cố định, không thay đổi hoặc rất ít thay đổi chủ yếu mang tính chất giới thiệu công ty và sản phẩm dịch vụ của họ. Các trang web này hầu hết được thiết kế bằng ASP.NET là framework được phát triển bởi Microsoft. Ngoài những tính năng vượt trội công nghệ ASP.NET có 2 nhược điểm lớn là không được hỗ trợ chạy trên hệ điều hành linux và không được hỗ trợ Visual Studio trong suốt quá trình viết code.

### **Nội dung đề tài**

* Xây dựng các chức năng website phía client gồm:
  + Khách hàng có thể xem tour, đặt tour.
  + Khách hàng có thể thanh toán tour đã đặt.
  + Khách hàng có thể đăng ký làm thành viên và quản lý thông tin cá nhân của mình.
  + Khách hàng có thể gửi phản hồi về công ty.
* Xây dựng các chức năng website cho người quản trị gồm:
* Tìm kiếm các đơn đặt chỗ đã được xác nhận hay chưa xác nhận.
* Xác nhận đơn đặt chỗ hợp lệ, chỉnh sửa thông tin khách hàng và thông tin đặt chỗ trên từng đơn cụ thể.
* Thống kê tình hình đặt chỗ của từng tour, xem danh sách khách hàng tham gia các tour đó.
* Thay đổi, thêm mới hoặc xóa bỏ thông tin, hình ảnh về các điểm du lịch ở các địa phương khác nhau mà công ty muốn giới thiệu cho khách hàng.
* Theo dõi tình hình liên hệ thông qua website của khách hàng, để đáp ứng kịp thời nhu cầu của họ.

### **Lựa chọn công nghệ**

Website này em xây dựng bằng Laravel framework vì:

* Tổ chức file và cấu trúc code rõ ràng
* Hệ thống thư viện và công cụ phong phú
* Form Validation
* Code ít hơn, nhưng làm được nhiều việc hơn
* Mô hình MVC rõ ràng, dễ bảo trì
* Chức năng hỗ trợ bảo mật, validate dữ liệu, SQL Injection,…
* Template phong phú, thư viện form helper giúp xây dựng form nhanh và bảo mật
* Hỗ trợ đường dẫn thân thiện (Friendly Url)
* Bảo mật session, cookie,…

# CHƯƠNG II. CƠ SỞ LÝ THUYẾT

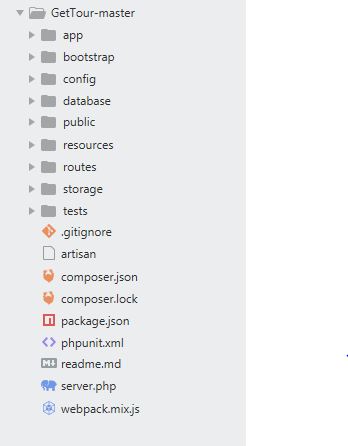
## 2.1 Giới thiệu Laravel framework

### **2.1.1 Khái niệm**

Laravel là một PHP framework mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell và nhắm vào mục tiêu hỗ trợ phát triển các ứng dụng web theo kiếm trúc model-view-controller (MVC). Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu – rõ ràng, cho phép đóng gói module và quản lý các pakage phụ thuộc, hỗ trợ nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ, nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng.

Hiện tại, Laravel là PHP Framework được sử dụng phổ biến nhất trên thế giới vì những ưu điểm vượt trội tận dụng các kỹ thuật Design Pattern, các công nghệ mới nhất của PHP và rất dễ dàng tiếp cận và sử dụng nó. Điểm yếu của Laravel là không hỗ trợ các phiên bản PHP cũ vì thế các website có nền tảng PHP version thấp hơn khá là khó khi có ý định chuyển sang Laravel và phải luôn cập nhật các thông tin mới nhất về PHP để áp dụng vào Laravel.

**Cấu trúc thư mục Laravel phiên bản 5.7**



Hình : Cấu trúc thư mục Laravel

* **app** là thư mục chứa tất cả các thư mục, các tập tin php, các lớp thư viện, thư viện, models.
  + **Console** thư mục chứa các tập tin định nghĩa các lệnh thực thi trên Artisan.
  + **Exceptions** thư mục chứa các tập tin quản lý, điều hướng lỗi.
  + **Http**
    - **Controllers** là thư mục chứa các tập tin controllers.
    - **Middleware** là thư mục chưa các tập tin lọc và ngăn chặn các requests.
    - **Kernel.php** là tập tin cấu hình, định nghĩa Middleware hoặc nhóm Middleware.
* **Providers** chứa tất cả các service providers ứng dụng. Service providers khởi động ứng dụng bằng các services trong service container, đăng ký events, hoặc thực hiện bất kỳ một công việc khác để chuẩn bị cho request đến ứng dụng.
* **bootstrap** thư mục chứa tập tin điều hướng khởi động hệ thống, thường thì chúng ta không làm gì đến nó.
* **config** chứa mọi tập tin cấu hình của Laravel.
* **database** chứa các thư mục tập tin về CSDL.
  + **Migrations** chứa các tập tin định nghĩa khởi tạo và sửa bảng.
  + **Seeds** chứa các tập tin định nghĩa dữ liệu thêm vào CSDL.
  + **Factories** chứa các tập tin định nghĩa các cột bảng dữ liệu để tạo ra các dữ liệu ảo phục vụ cho kiểm thử.
* **public** chính là webroot người dùng sẽ truy cập vào đây, đây cũng là nơi chứa các tập tin css, js, image.
* **resources** chứa các tập tin giao diện (js, css,…), views, ngôn ngữ.
* **storage** chứ các tập tin hệ thống như upload, cache, session, cookie, log…
* **routes** là thư mục chứa các tập tin định nghĩa các router, xử lý router hoặc điều hướng router bao gồm 3 loại là web, api và console.
* **tests** chứa các tập tin định nghĩa tests.
* **vendor** thư mục của composer.
* **.env và .env.example** là 2 tập tin cấu hình chính của Laravel như key app, tên app, url app, email, env mode, CSDL hay bật tắt debug.
* **composer.json, composer.lock** tập tin của composer.
* **package.js** tập tin cấu hình của nodejs chứa các package cần thiết cho projects.
* **gulpfile.js** là tập tin gulp builder.
* **phpunit.xml** là tập tin để artisan trỏ đến tạo server khi gõ lệnh php artisan server
* **Artisan** tập tin thực thi lệnh của Laravel, cũng là tập tin mà chúng ta tương tác nhiều nhất.

### **2.1.2 Tại sao chọn Laravel framework**

* Laravel có một hệ thống có kiến trúc thống nhất, khoa học và cực đơn giản giúp những người mới tiếp cận framework PHP có thể nắm bắt được ngay. Các framework khác thật sự khá là khó để bắt đầu, đặc biệt là Zend, Yii. Tuy rất đơn giản nhưng Laravel hoàn toàn có thể xây dựng những hệ thống ứng dụng web cực phức tạp, với hiệu năng cao.
* Các tài nguyên liên quan đến Laravel rất đa dạng, dễ sử dụng, kể đến có tài liệu chính thức của Laravel được chính Taylor Otwell biên soạn, tác giả của framework này muốn đưa việc lập trình lên một tầm cao mới đó là khám phá và trải nghiệm, ngoài ra cộng đồng Laravel rất lớn trên Laracast, ta có tìm được câu trả lời cho bất kỳ tình huống nào bạn gặp phải trong Laravel hoặc tiếp cận rất nhiều các video hướng dẫn cụ thể.
* Trực quan, đó là những gì mọi người nói về viết mã trong Laravel, giúp chúng ta tập trung vào các nhiệm vụ chính và các đoạn mã viết trong Laravel được rõ ràng và rất dễ dàng trong bảo trì mã nguồn.
* Bộ máy blade template thực sự rất hay, nó giúp cho làm việc giữa PHP và HTML thật đơn giản, sáng sủa và tách biệt giữa mã hiển thị và mã logic nghiệp vụ.
* Laravel sử dụng MVC pattern để xây dựng framework. Nó đảm bảo rằng các code được trình bày logic và hợp lý. Dễ dàng để có thể hiểu được nó. Vì vậy nó làm cho dự án web được trở nên dễ quản lý cũng như phát triển rất nhiều.
* Laravel hỗ trợ việc cài đặt các gói thư viện và bản thân nó sử dụng rất nhiều các gói thư viện khác để tạo ra các tính năng tuyệt vời.
* Luôn đổi mới, tích cực và sáng tạo.

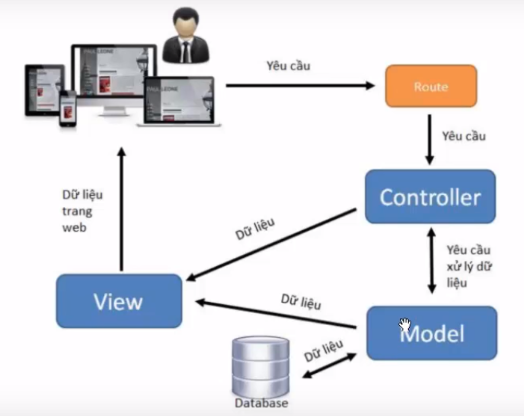
## 2.2 Mô hình MVC

Mô hình MVC (model, view, controller) là mô hình chuẩn cho ứng dụng web được sử dụng nhiều nhất ngày nay. Mô hình MVC được sử dụng lần đầu tiên trong Smalltalk, sau đó được sử dụng phổ biến trong ngôn ngữ lập trình Java. Hiện nay, đã có hơn hàng chục PHP framework dựa trên mô hình này. Mô hình MVC là viết tắt của 3 chữ Model, View, Controller. Mô hình này tách một ứng dụng web ra làm 3 thành phần đảm nhiệm chức năng tách biệt, thuận tiện cho việc xử lý và bảo trì.

**Model:** có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View.

**View:** có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, có thể hiểu nôm na đây người ta còn gọi là thành phần giao diện.

**Controller:** đóng vài trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ người dùng sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho người dùng.



Hình : Mô hình MVC trong Laravel

* Người dùng sử dụng trình duyệt web (Browser) yêu cầu truy cập vào địa chỉ website.
* **Route** sẽ phân tích xem đường dẫn (link) sẽ truy cập đến Controller nào.
* **Controller** sẽ truy cập vào Model để lấy dữ liệu từ cơ sở dữ liệu (Database), sau đó trả về dữ liệu cho Controller.
* **Controller** sẽ trả dữ liệu ra View hoặc trả dữ liệu dạng JSON cho người dùng.
* **View** sẽ xuất dữ liệu ra màn hình cho người dùng xem.

**Ưu điểm:**

* Hệ thống phân ra từng phần nên dễ dáng phát triển.
* Chia thành nhiều modun nhỏ nên nhiều người có thể làm chung dự án.
* Vấn đề bảo trì cũng tương đối tốt, dễ nâng cấp.
* Dễ dàng rà soát lỗi trong quá trình xây dựng.

**Nhược điểm:**

* Hệ thống sẽ chạy chậm hơn PHP thuần, tuy nhiên nó ko phải là vấn đề.
* Xây dựng cầu kì và mất thời gian để xây dựng thư viện, cấu trúc.

## 2.3 jQuery, boostrap

### **2.3.1 jQuery**

**jQuery** là một thư viện JavaScript nhỏ gọn, chạy nhanh. jQuery giải quyết các vấn đề tương tác với HTML, bắt sự kiện, hiệu ứng động… trở nên rất đơn giản.

Các tính năng chính jQuery cung cấp gồm:

* Tương tác với HTML/DOM
* Tương tác với CSS
* Bắt và xử lý sự kiện HTML
* Các hiệu ứng và chuyển động trong HTML
* AJAX (Asynchronous JavaScript and XML)
* JSON parsing
* Các tiện ích xây dựng sẵn

**Ưu điểm của jquery**

* Dễ sử dụng: Đây là lợi thế chính khi sử dụng jquery, nó dễ dàng hơn so với nhiều thư viện javascript chuẩn khác bởi cú pháp đơn giản và ta chỉ phải viết ít dòng lệnh để tạo ra các chức năng tương tự. Chỉ với 10 dòng lệnh JQuery ta có thể thay thế cả 20 dòng lệnh DOM javaScript, tiết kiệm thời gian của người lập trình.
* Là một thư viện lớn của javascript: Thực thi được nhiều chức năng hơn so với các thư viện jascript khác
* Cộng đồng mã nguồn mở mạnh mẽ (một số plugin jquery có sẵn.
* Có nhiều tài liệu và hướng dẫn chi tiết.
* Hỗ trợ ajax: JQuery cho phép ta phát triển các template Ajax một cách dễ dàng. Ajax cho phép một giao diện kiểu dáng đẹp trên website, các chức năng có thể được thực hiện trên các trang mà không đòi hỏi toàn bộ trang được reload lại.

**Nhược điểm của jquery**

Dù ưu điểm của jquery rất nhiều nhưng cũng tồn tại một vài nhược điểm phải lưu ý.

* Làm client trở nên chậm chạp: Client không những phải chỉ mình hiển thị nữa mà còn phải xử lý nhiều chức năng được tạo thành từ jquery. Nếu lạm dụng quá nhiều jquery sẽ làm cho client trở nên chậm chạp, đặc biệt những client yếu. Chính vì vậy mà lập trình viên phải dùng thêm cache.
* Chức năng có thể không có: JQuery đã có rất nhiều chức năng, tùy thuộc vào yêu cầu trên trang web của bạn. Nhưng nhiều chức năng vẫn chưa được phát triển, do đó bạn vẫn phải sử dụng javascript thuần để xây dựng chức năng này.

### **2.3.2 Boostrap**

**Bootstrap** là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet,...

Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế reponsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.

**Tại sao chúng ta nên sử dụng Bootstrap?**

* Bootstrap là một trong những framework được sử dụng nhiều nhất trên thế giới để xây dựng nên một website. Bootstrap đã xây dựng nên 1 chuẩn riêng và rất được người dùng ưa chuộng. Chính vì thế, chúng ta hay nghe tới một cụm từ rất thông dụng "Thiết kế theo chuẩn Bootstrap".
* Từ cái "chuẩn mực" này, chúng ta có thể thấy rõ được những điểm thuận lợi khi sử dụng Bootstrap.
* Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cẩn có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.
* Responsive: Bootstrap xây dựng sẵn reponsive css trên các thiết bị Iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.
* Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, Bootstrap chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không support HTML5 và CSS3.

### **2.3.3 Hệ quản trị cơ sở dữ liệu MySQL**

**MySQL** là hệ quản trị cơ sở dữ liệu mã nguồn mở miễn phí, được tích hợp sử dụng chung với apache, PHP. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu, mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL có cơ chế phân quyền người sử dụng riêng, mỗi người dùng có thể được quản lý một hoặc nhiều cơ sở dữ liệu khác nhau, mỗi người dùng có một tên truy cập (user name) và mật khẩu tương ứng để truy xuất đến cơ sở dữ liệu. Khi ta truy vấn tới cơ sở dữ liệu MySQL, ta phải cung cấp tên truy cập và mật khẩu của tài khoản có quyền sử dụng cơ sở dữ liệu đó. Nếu không, chúng ta sẽ không làm được gì cả.

**Ưu điểm của MySQL**

* Dễ dàng sử dụng: MySQL có thể dễ dàng cài đặt. Với các công cụ bên thứ 3 làm cho nó càng dễ đơn giản hơn để có thể sử dụng.
* Giàu tính năng: MySQL hỗ trợ rất nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
* Bảo mật: Có rất nhiều tính năng bảo mật, một số ở cấp cao đều được xây dựng trong MySQL.
* Khả năng mở rộng và mạnh mẽ: MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
* Nhanh: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

**Nhược điểm của MySQL**

* Giới hạn: Theo thiết kế, MySQL không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.
* Độ tin cậy: Cách các chức năng cụ thể được xử lý với MySQL (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán,...) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.
* Sự phát triển chậm: Mặc dù MySQL vẫn là một sản phẩm công nghệ mã nguồn mở phổ biến nhưng có những phàn nàn liên quan đến quá trình phát triển kể từ khi nó được mua lại.

## 

## CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1 Xác định yêu cầu hệ thống

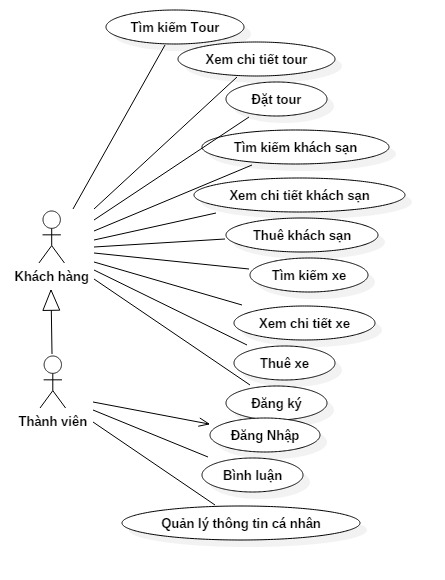
* Công ty du lịch thiết kế tour du lịch và lên lịch trình cho từng tour cụ thể. Sau đó, cập nhật thông tin các tour này lên website với đầy đủ thông tin về giá cả, loại tour, lịch trình, các địa phương đi đến và hình ảnh minh họa (nếu có).
* Khách hàng truy cập web có thể xem, tìm kiếm thông tin về các tour du lịch hiện hành hoặc tìm kiếm tour theo các yêu cầu cụ thể về giá cả, địa phương muốn đến, ngày khởi hành của tour. Sau đó, khách hàng có thể tiến hành đặt chỗ cho tour đang xem nếu muốn.
* Khách hàng có thể đặt tour trực tuyến. Website sẽ cập nhật thông tin đặt chỗ này trực tiếp lên cơ sở dữ liệu, và cho phép người quản trị được phân quyền thao tác các công việc sau:
* Tìm kiếm các đơn đặt chỗ đã được xác nhận hay chưa xác nhận.
* Xác nhận đơn đặt chỗ hợp lệ, chỉnh sửa thông tin khách hàng và thông tin đặt chỗ trên từng đơn cụ thể.
* Thống kê tình hình đặt chỗ của từng tour, xem danh sách khách hàng tham gia các tour đó.
* Thay đổi, thêm mới hoặc xóa bỏ thông tin, hình ảnh về các điểm du lịch ở các địa phương khác nhau mà công ty muốn giới thiệu cho khách hàng.
* Theo dõi tình hình liên hệ thông qua website của khách hàng, để đáp ứng kịp thời nhu cầu của họ.
* Về tổ chức lưu trữ, thực hiện các yêu cầu:
* Thêm, xóa, sửa thông tin, hình ảnh về các tour du lịch do công ty tổ chức, phục vụ cho công tác quản lý, thống kê tình hình hoạt động của công ty.
* Thêm, xóa, sửa thông tin, hình ảnh về các điểm du lịch ở từng địa phương khác nhau.

## 3.2 Thiết kế ứng dụng web phía client

### **3.2.1 Biểu đồ Use Case tổng quát**

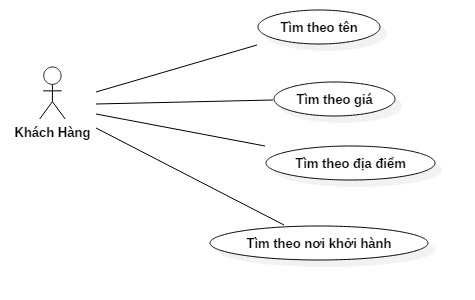
Hình 3 dưới đây mô tả tổng quan các chức năng của hệ thống website du lịch phía client, hệ thống gồm 2 nhóm người dùng: thành viên đã có tài khoản và đăng nhập vào hệ thống, khách hàng chưa có tài khoản.

* Actor khách hàng: Khách hàng vào website có thể tìm kiếm, xem tour và tiến hành đặt tour và các dịch vụ khác của công ty, đăng ký thành viên.
* Actor thành viên: Đã đăng ký tài khoản và đăng nhập vào hệ thống. Thành viên có đầy đủ chức năng của khách hàng ngoài ra có thêm chức năng bình luận, quản lý thông tin cá nhân.



Hình : Biểu đồ Use Case tổng quát web client

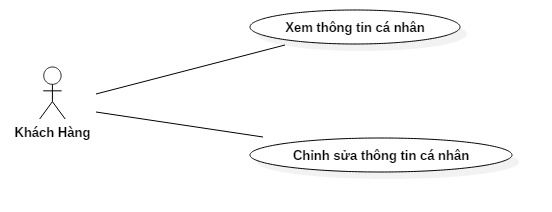
1. ***Phân rã Use Case tìm kiếm tour***

****

Hình : Phân rã Use Case tìm kiếm tour

Hình 4 mô tả chức năng tìm kiếm tour. Khách hàng có thể tìm kiếm theo tên, giá, địa điểm, nơi khởi hành của tour du lịch theo nhu cầu.

1. ***Phân rã Use Case quản lý thông tin cá nhân***

****

Hình : Phân rã UseCase quản lý thông tin cá nhân

Hình 5 mô tả chức năng quản lý thông tin cá nhân. Khách hàng có thể xem hoặc chỉnh sửa thông tin cá nhân.

### **3.2.2 Đặc tả một số chức năng chính**

***a) Use Case: Tìm kiếm tour***

|  |  |
| --- | --- |
| Tên Use Case | Tìm kiếm tour |
| Tác nhân | Khách hàng |
| Mục đích | Cho phép khách hàng tìm kiếm tour du lịch |
| Điều kiện đầu vào | Khách hàng truy cập hệ thống |
| Kịch bản chính | + Người sử dụng nhập thông tin cần thiết vào form tìm kiếm.  + Nhấn nút tìm kiếm.  + Hệ thống hiển thị kết quả.  . |

Bảng : Đặc tả Use Case Tìm kiếm tour

***b)* *Use Case: Xem chi tiết tour***

|  |  |
| --- | --- |
| Tên Use Case | Xem chi tiết tour |
| Tác nhân | Khách hàng |
| Mục đích | Cho phép khách hàng xem thông tin chi tiết về tour du lịch |
| Điều kiện đầu vào | Khách hàng đã tìm kiếm tour |
| Kịch bản chính | + Người dùng chọn tour cần xem.  + Form xem thông tin chi tiết tour hiển thị. Hệ thống hiển thị thông tin tour đã chọn  + Người dùng xem thông tin chi tiết về tour đã chọn |

Bảng : Đặc tả Use Case xem chi tiết tour



## *c) Use Case: Đặt tour*

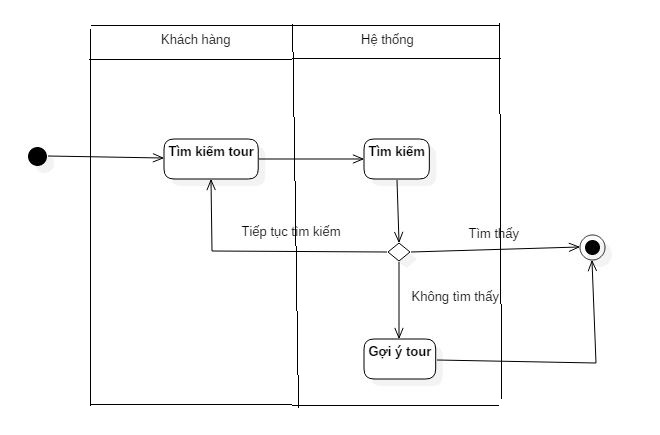
|  |  |
| --- | --- |
| Tên Use Case | Đặt tour |
| Tác nhân | Khách hàng |
| Mục đích | Cho phép khách hàng đặt tour và tiến hành thanh toán |
| Điều kiện đầu vào | Khách hàng đã chọn tour |
| Kịch bản chính | + Khách hàng nhấn thanh toán để tạo đơn đặt.  + Form thanh toàn xuất hiện, hệ thống hiển thị đơn đặt và một số thông tin mặc định của tour  + Khách hàng nhập tiếp các thông tin.  + Chọn thanh toán.  + Hệ thống kiếm tra và thông báo. |

Bảng : Đặc tả Use Case Đặt tour

### **3.2.3 Mô hình hóa hoạt động của hệ thống**

1. ***Tìm kiếm tour***

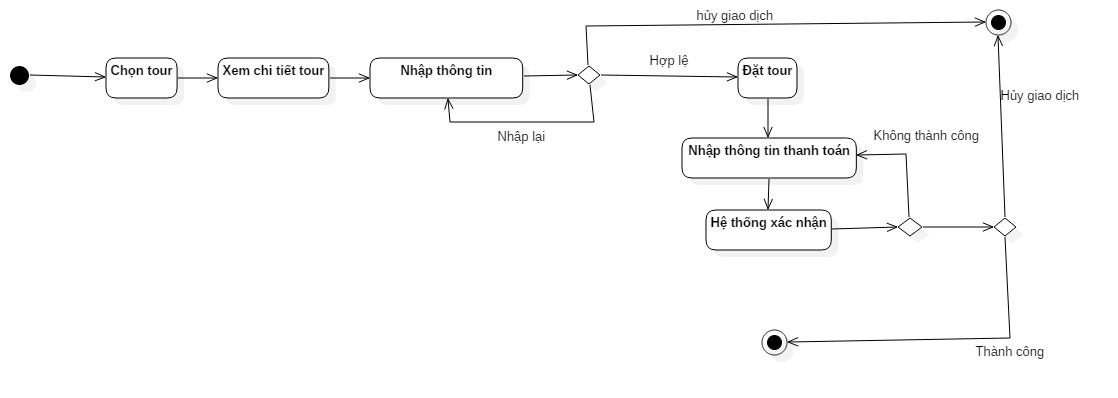
Hình 6 bên dưới là quy trình tìm kiếm tour của khách hàng. Khách hàng sử dụng chức năng tìm kiếm tour, hệ thống sẽ tìm kiếm trong cơ sở dữ liệu rồi trả về kết quả.



Hình : Biểu đồ hoạt động mô tả quá trình tìm kiếm

#### **Đặt tour**

Hình 7 là quy trình đặt tour của khách hàng. Khách hàng chọn tour mình muốn đặt, sau đó tiến hành nhập thông tin cho chức năng đặt tour và tiến hành thanh toán.

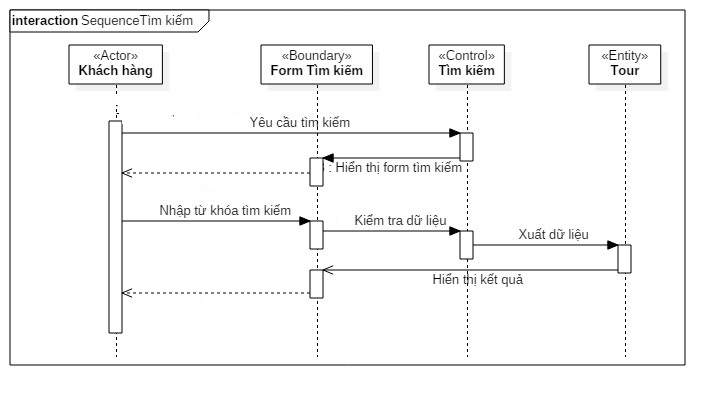


Hình : Biểu đồ hoạt động mô tả quá trình đặt tour

### **3.2.4 Mô hình hóa sự tương tác**

**a) Biểu đồ trình tự cho chức năng tìm kiếm**

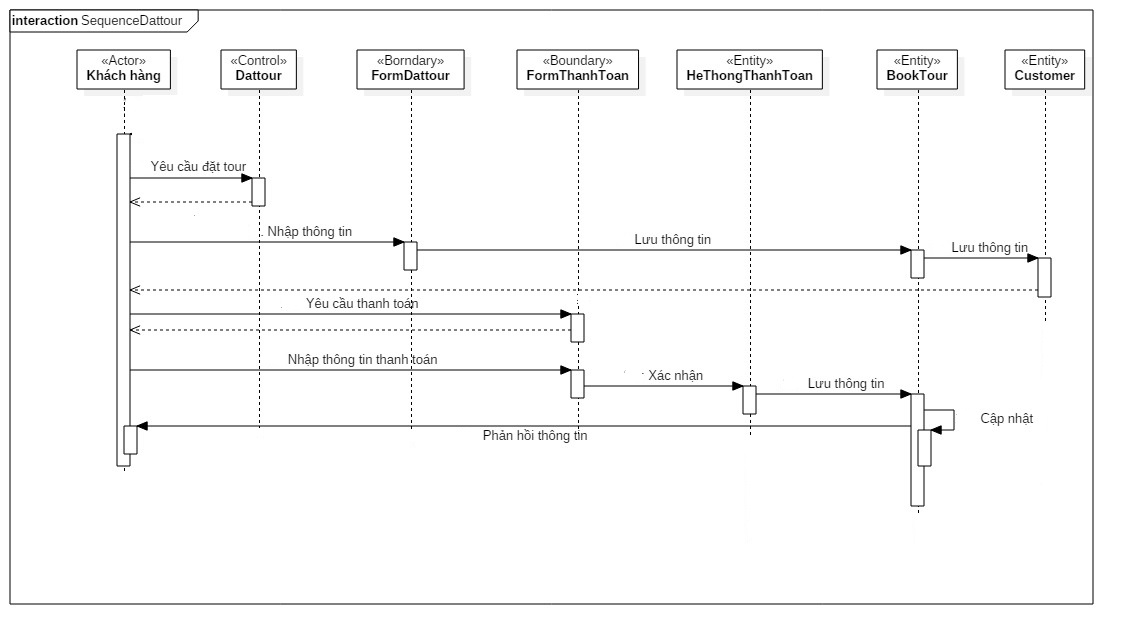
Hình 8 là biểu đồ trình tự tìm kiếm. Khách hàng yêu cầu tìm kiếm, hệ thống hiển thị form tìm kiếm, khách hàng nhập từ khóa tìm kiếm, hệ thống kiểm tra dữ liệu rồi trả về cho khách hàng trên màn hình hiển thị.

****

Hình : Biểu đồ trình tự tìm kiếm

**b) Biểu đồ trình tự cho chức năng đặt tour**

Hình 9 là biểu đồ trình tự cho chức năng đặt tour.

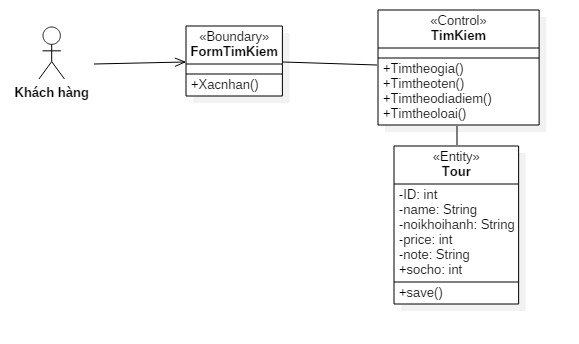


Hình : Biểu đồ trình tự chức năng đặt tour

## Thiết kế lớp (Class)

1. **Biểu đồ lớp cho chức năng tìm kiếm tour**

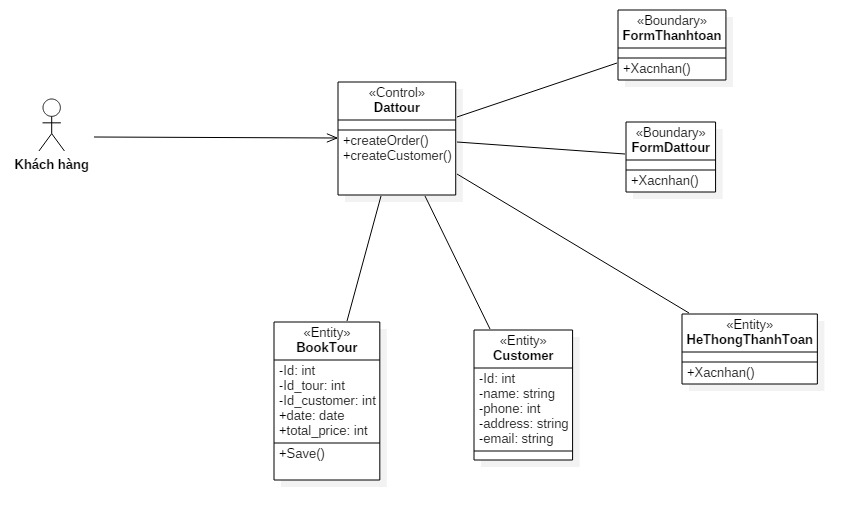
Hình 10 là biểu đồ lớp cho chức năng tìm kiếm tour, mô tả các phương thức và thuộc tính của các lớp mà khách hàng tương tác đến khi dùng chức năng tìm kiếm tour.

****

Hình : Biểu đồ lớp chức năng tìm kiếm tour

1. **Biểu đồ lớp cho chức năng đặt tour**

Hình 11 là biểu đồ lớp cho chức năng đặt tour, mô tả các phương thức và thuộc tính của các lớp mà khách hàng tương tác đến khi dùng chức năng đặt tour.

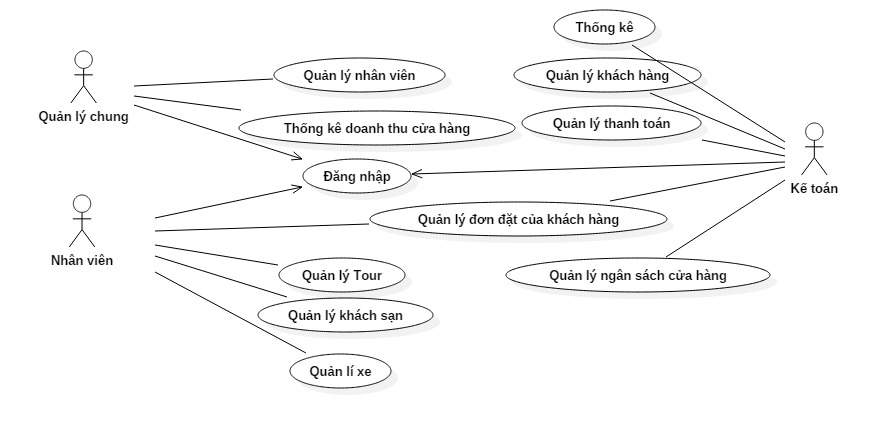
****

Hình : Biểu đồ lớp chức năng đặt tour

## 3.3 Thiết kế ứng dụng web quản trị

### **3.3.1 Biểu đồ Use Case tổng quát**

Hình 12 dưới đây mô tả tổng quan các chức năng của hệ thống website quản lý du lịch.



Hình : Biểu đồ Use Case tổng quát của người quản trị

1. **Các tác nhân chính**

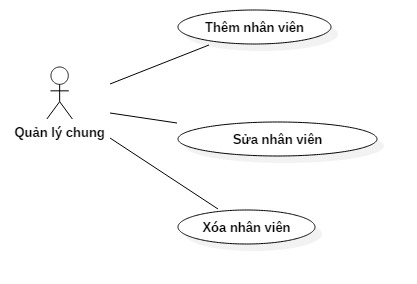
Hệ thống có 3 tác nhân chính bao gồm:

**Người quản lý chung** là người có quyền cao nhất trong toàn bộ hệ thống quản lý có thể sử dụng tất cả các chức năng mà hệ thống có. Đồng thời, người quản lý chung này sẽ trực tiếp quản lý thông tin các nhân viên, phân quyền sử dụng các chức năng của hệ thống với mỗi chức vụ mà nhân viên đảm nhận.

**Nhân viên** là người đảm nhận các công việc quản lý các đơn đặt của khách hàng, quản lý, quản lý các thông tin của chương trình tour, khách sạn, xe.

**Kế toán** là người quản lý ngân sách của cửa hàng bao gồm quản lý các việc thu, chi của cửa hàng, quản lý danh sách các tài khoản mà cửa hàng đang có (tiền mặt, tài khoản ngân hàng), quản lý thanh toán đơn đặt của khách hàng (số tiền đã thanh toán, số tiền còn nợ).

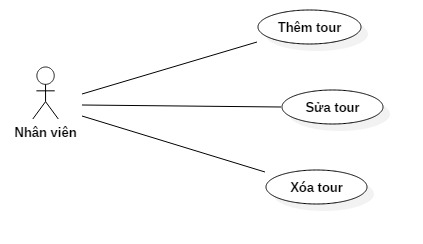
1. **Phân rã Use Case quản lý nhân viên**

****

Hình : Phân rã Use Case quản lý nhân viên

Hình 13 mô tả chức năng Quản lý nhân viên. Quản lý chung có thể tạo, xem, chỉnh sửa và xóa các thông tin về nhân viên.

1. **Phân rã Use case quản lý tour du lịch**

****

Hình : Phân rã Use Case quản lý tour

Hình 14 mô tả chức năng Quản lý tour. Nhân viên có thể thêm, xem, chỉnh sửa và xóa các thông tin về tour du lịch.

### **3.3.2 Đặc tả một số chức năng chính**

***a) Use Case: Đăng nhập***

|  |  |
| --- | --- |
| Tên Use Case | Đăng nhập |
| Tác nhân | Người quản trị (Quản lý chung, kế toán, nhân viên) |
| Mục đích | Cho phép người quản trị đăng nhập vào hệ thống để quản lý cơ sở dữ liệu |
| Điều kiện đầu vào | Người quản trị truy cập hệ thống |
| Kịch bản chính | + Người quản trị chọn chức năng đăng nhập.  + Hệ thống hiển thị form đăng nhập.  + Nhập tên, mật khẩu vào form đăng nhập  + Hệ thống kiểm tra tên, mật khẩu của người dùng. Nếu đúng thì đăng nhập vào hệ thống. Nếu sai thì yêu cầu đăng nhập lại. |

Bảng : Đặc tả Use Case đăng nhập

***b)* *Use Case: Quản lý nhân viên***

|  |  |
| --- | --- |
| Tên Use Case | Quản lý nhân viên |
| Tác nhân | Quản lý chung |
| Mục đích | Cho phép người quản lý chung thêm, sửa, xóa nhân viên |
| Điều kiện đầu vào | Người quản lý chung đăng nhập thành công, đang ở giao diện trang quản trị. |
| Kịch bản chính | + Thêm nhân viên: Người quản lý chọn mục thêm người dùng, sau đó nhập các trường thông tin, chọn chức năng thêm, hệ thống hiển thị kết quả.  + Xóa nhân viên: Người quản lý chọn nhân viên cần xóa, hệ thông hiển thị thông tin nhân viên, người quản lý chọn chức năng xóa, hệ thống thông báo kết quả.  + Cập nhật nhân viên: Người quản lý chọn nhân viên cần cập nhật, hệ thống hiển thị thông tin nhân viên, người quản lý cập nhật thông tin nhân viên, sau đó nhấn nút cập nhật, hệ thống hiển thị kết quả. |

Bảng : Đặc tả Use Case quản lý nhân viên



## *c) Use Case: Quản lý tour*

|  |  |
| --- | --- |
| Tên Use Case | Quản lý tour |
| Tác nhân | Nhân viên |
| Mục đích | Cho phép nhân viên thêm, sửa, xóa tour du lịch |
| Điều kiện đầu vào | Nhân viên đăng nhập thành công, đang ở giao diện trang quản lý. |
| Kịch bản chính | + Thêm tour: Nhân viên chọn mục thêm tour, sau đó nhập các trường thông tin, chọn chức năng thêm, hệ thống hiển thị kết quả.  + Xóa tour: Nhân viên chọn tour cần xóa, hệ thống hiển thị thông tin nhân viên, nhân viên chọn chức năng xóa, hệ thống thông báo kết quả.  + Cập nhật tour: Nhân viên chọn tour cần cập nhật, hệ thống hiển thị thông tin tour, nhân viên cập nhật thông tin nhân viên, sau đó nhấn nút cập nhật, hệ thống hiển thị kết quả. |

Bảng : Đặc tả Use Case quản lý tour

***d) Use Case: Quản lý đơn đặt của khách hàng***

|  |  |
| --- | --- |
| Tên Use Case | Quản lý đơn đặt |
| Tác nhân | Nhân viên, kế toán |
| Mục đích | Cho phép người dùng cập nhật, xóa đơn đặt của khách hàng |
| Điều kiện đầu vào | Nhân viên, kế toán đăng nhập thành công, đang ở giao diện trang quản lý. |
| Kịch bản chính | + Xóa đơn: Nhân viên chọn đơn đặt cần xóa, hệ thống hiển thị thông tin, nhân viên chọn chức năng xóa, hệ thống thông báo kết quả.  + Cập nhật đơn: Nhân viên chọn đơn đặt cần cập nhật, hệ thống hiển thị thông tin, nhân viên cập nhật thông tin, sau đó nhấn nút cập nhật, hệ thống hiển thị kết quả. |

Bảng : Đặc tả Use Case quản lý đơn đặt của khách hàng

***e) Use Case: Quản lý thanh toán***

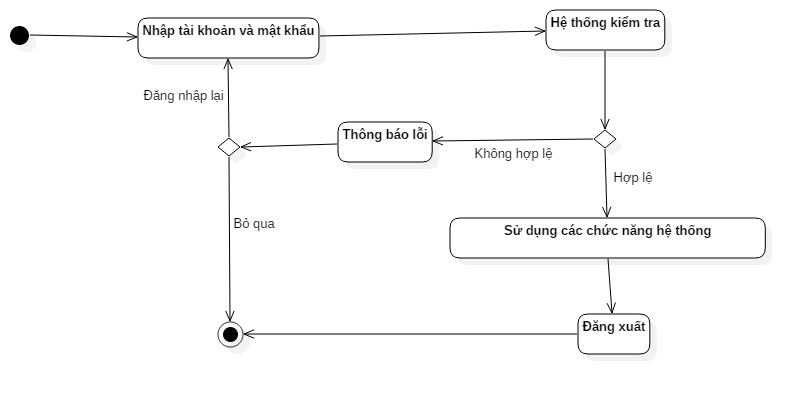
|  |  |
| --- | --- |
| Tên Use Case | Quản lý đơn đặt |
| Tác nhân | Kế toán |
| Mục đích | Cho phép kế toán thanh toán các hóa đơn của khách hàng. |
| Điều kiện đầu vào | Nhân viên, kế toán đăng nhập thành công, có quyền thanh toán đơn đặt của khách hàng. |
| Kịch bản chính | + Kế toán yêu cầu thanh toán đơn đặt của khách hàng.  + Hệ thống hiển thị giao diện thanh toán đơn đặt của khách hàng.  + Kế toán nhập số tiền thanh toán, xác nhận.  + Hệ thống hiển thị thông báo. |

Bảng : Đặc tả Use Case quản lý thanh toán

### **3.3.3 Mô hình hóa hoạt động của hệ thống**

**a) Đăng nhập**

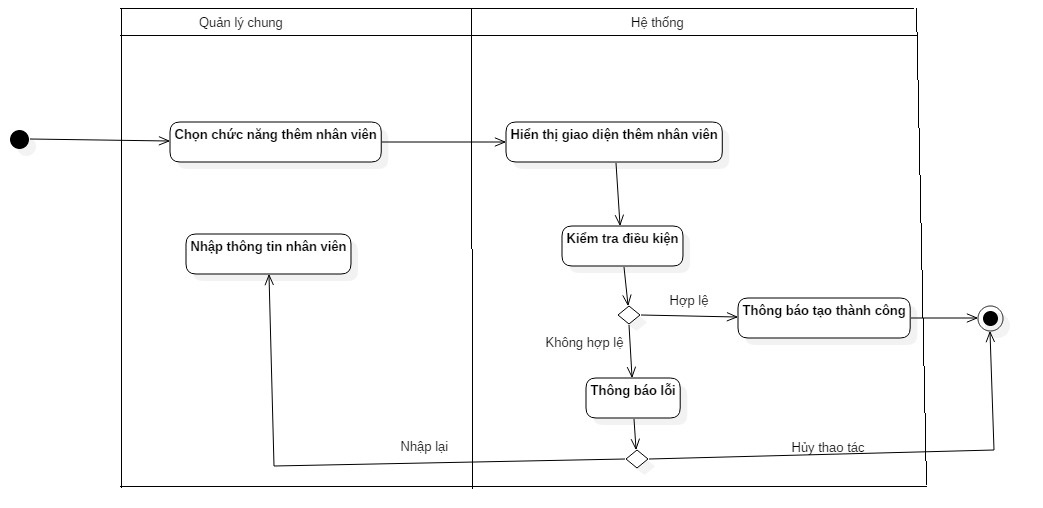
Hình 15 là biểu đồ hoạt động mô tả quy trình đăng nhập của nhân viên. Nhân viên truy cập website đăng nhập trang quản trị sau đó nhập tài khoản và mật khẩu. Hệ thống kiếm trả rồi trả kết quả lại cho người dùng.

****

Hình : Biểu đồ hoạt động mô tả quá trình đăng nhập

1. **Thêm nhân viên**

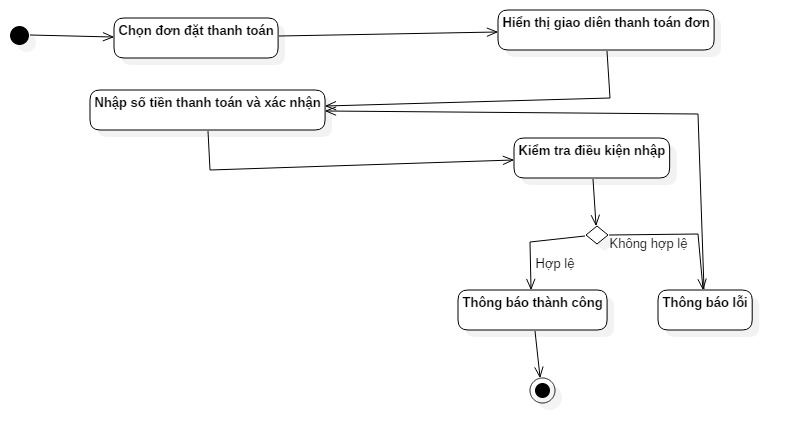
Hình 16 là biểu đồ hoạt động mô tả quy trình thêm nhân viên.



Hình : Biểu đồ hoạt động mô tả quá trình thêm nhân viên

1. **Thanh toán**

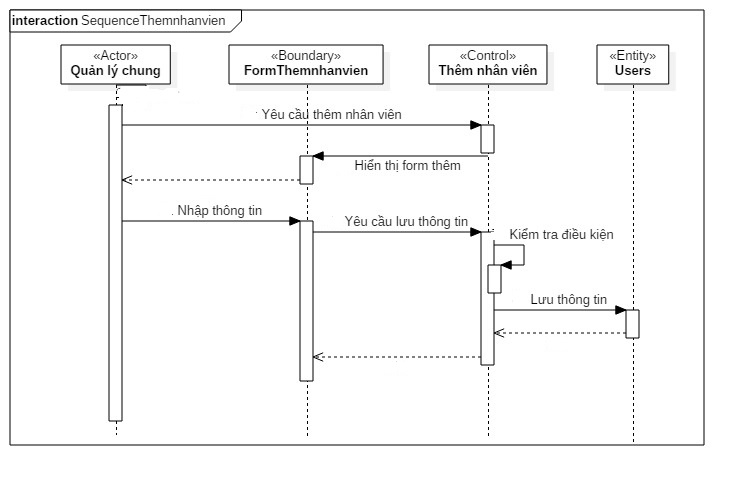
Hình 17 bên dưới là biểu đồ hoạt động mô tả quá trình nhân viên thanh toán đơn đặt tour của khách hàng.

****

Hình : Biểu đồ hoạt động Thanh toán đơn đặt khách hàng

### **3.3.4 Mô hình hóa sự tương tác**

**a) Biểu đồ trình tự cho chức năng thêm nhân viên**

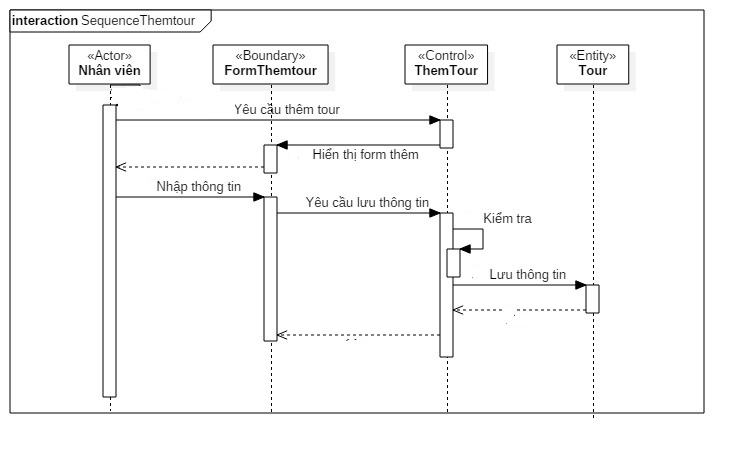
****

Hình : Biểu đồ trình tự thêm nhân viên

Hình 18 là biểu đồ trình tự cho chức năng thêm nhân viên của Actor quản lý chung.

**b) Biểu đồ trình tự cho chức năng thêm tour**

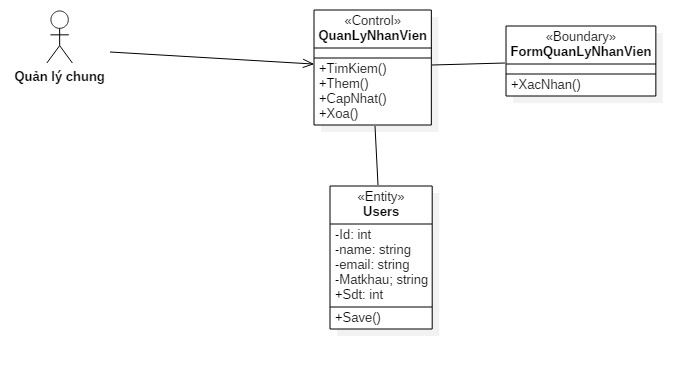
Hình 19 bên dưới là biểu đồ trình tự cho chức năng thêm tour.



Hình : Biểu đồ trình tự chức năng thêm tour

### **3.3.5 Thiết kế lớp (class)**

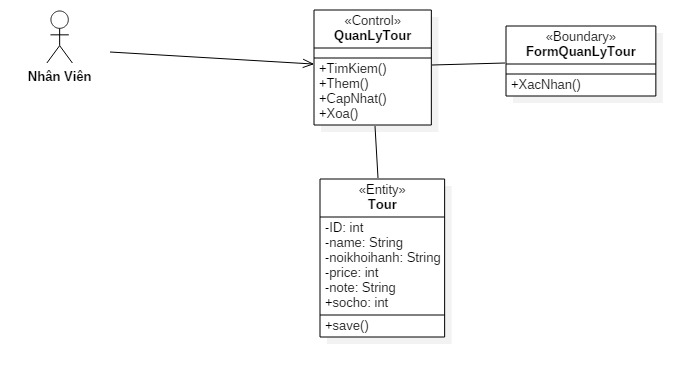
**a) Biểu đồ lớp cho chức năng quản lý nhân viên**

****

Hình : Biểu đồ lớp cho chức năng quản lý nhân viên

Hình 20 là biểu đồ lớp cho chức năng quản lý nhân viên của người quản lý chung. Biểu đồ này mô tả các phương thức và thuộc tính của các lớp mà người quản lý chung tương tác đến khi dùng chức năng quản lý nhân viên.

**b) Biểu đồ lớp cho chức năng quản lý tour**

****

Hình : Biểu đồ lớp cho chức năng quản lý tour

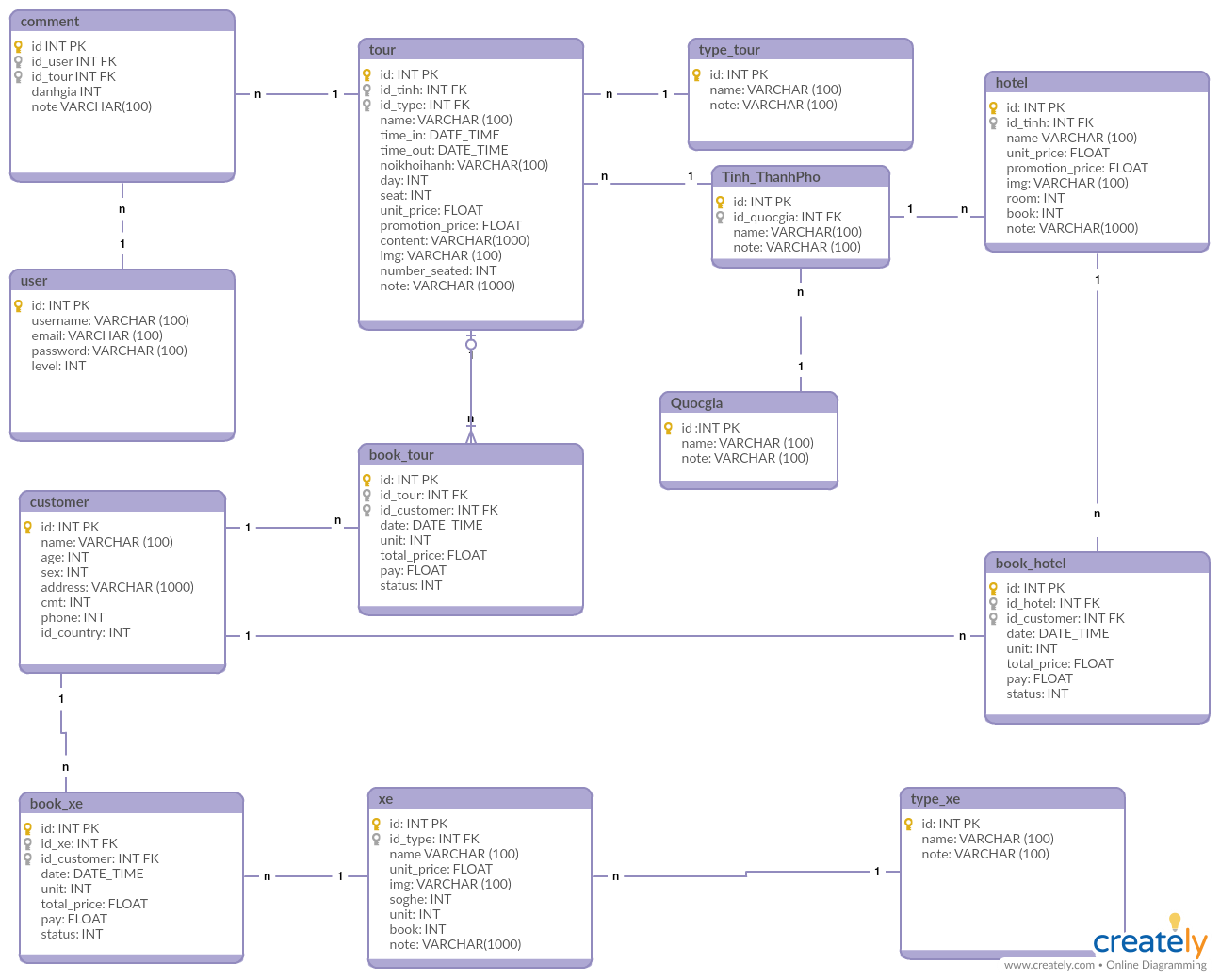
Hình 21 là biểu đồ lớp cho chức năng quản lý tour của nhân viên. Biểu đồ này mô tả các phương thức và thuộc tính của các lớp mà nhân viên tương tác đến khi dùng chức năng quản lý tour.

## 3.4 Xây dựng ứng dụng và triển khai

### **3.4.1 Thiết kế cơ sở dữ liệu**

**a) CSDL của hệ thống**

Hình 22 dưới đây là cơ sở dữ liệu của toàn hệ thống, bao gồm tên các bảng, các  
trường trong mỗi bảng và mối quan hệ giữa các bảng với nhau.

****

Hình : Cơ sở dữ liệu hệ thống

**b) Đặc tả CSDL**

***Bảng: Customer***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | id | int | Mã khách hàng |
| 2 | name | Varchar(255) | Tên khách hàng |
| 3 | email | Varchar(255) | Email khách hàng |
| 4 | phone | int | Số điện thoại khách hàng |
| 5 | address | Varchar(255) | Địa chỉ khách hàng |
| 6 | age | int | Tuổi khách hàng |
| 7 | cmt | int | Số chứng mình thư |

Bảng : Đặc tả dữ liệu bảng khách hàng

***Bảng: Tour***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | id | int | Mã tour |
| 2 | id\_tinh | int | Mã tỉnh |
| 3 | id\_type | int | Mã loại tour |
| 4 | name | Varchar(255) | Tên tour |
| 5 | time\_in | date | Ngày khởi hành |
| 6 | time\_out | date | Ngày về |
| 7 | noikhoihanh | Varchar(255) | Nơi Khởi Hành |
| 8 | day | int | Số ngày đi |
| 9 | seat | int | Số chỗ |
| 10 | Unit\_price | float | Giá tour |
| 11 | Promotion\_price | float | Giá khuyến mãi |
| 12 | image | Varchar(255) | Hình ảnh tour |
| 13 | content | Varchar(255) | Mô tả về tour |

Bảng : Đặc tả dữ liệu bảng tour du lịch

***Bảng: Book\_Tour***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | id | int | Mã đặt tour |
| 2 | id\_tour | int | Mã tour |
| 3 | id\_customer | int | Mã khách hàng |
| 4 | date | date | Ngày đặt |
| 5 | unit | int | Số chỗ đặt |
| 6 | Total\_price | float | Tổng tiền |
| 7 | pay | int | Số tiền đã thanh toán |
| 8 | Pay\_ment | Varchar(255) | Hình thức thanh toán |
| 9 | status | Varchar(255) | Trạng thái |

Bảng : Đặc tả dữ liệu bảng đặt tour du lịch

***Bảng: user***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | id | int | Mã người dùng |
| 2 | username | int | Tên tài khoản |
| 3 | email | int | email |
| 4 | password | Varchar(255) | Mật khẩu |
| 5 | level | date | Quyền |

Bảng : Đặc tả dữ liệu bảng tour du lịch

***Bảng comment***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | id | int | Mã bình luận |
| 2 | Id\_user | int | Mã người dùng |
| 3 | Id\_tour | int | Mã tour |
| 4 | note | Varchar(255) | Ghi chú |
| 5 | danhgia | int | Điểm đánh giá |

Bảng : Đặc tả dữ liệu bảng comment