mục lục

lời cảm ơn

danh mục bảng

danh mục hình ảnh

Thuật ngữ và từ viết tắt

Cam kết kết quả

Tóm tắt

Abstract

phần 1: giới thiệu

# Đặt vấn đề

# Lịch sử giải quyết vấn đề

# Mục tiêu đề tài

# Đối tượng và phạm vi nghiên cứu

# Nội dung nghiên cứu

## Quy trình nghiên cứu

## Các công nghệ sử dụng

## Công cụ hỗ trợ và phát triển hệ thống

# Những đóng góp chính của đề tài

# Bố cục luận văn

phần 2:

chương 1: đặc tả yêu cầu

# Giới thiệu

# Mô tả tổng quan

# Các yêu cầu giao tiếp bên ngoài

# Các yêu cầu chức năng

# Các yêu cầu phi chức năng

chương 2: thiết kế và cài đặt giải pháp

phần 3: kết luận

Chương 1: cơ sở lý thuyết

# Thanh toán trực tuyến

## Thanh toán trực tuyến là gì?

Thanh toán trực tuyến là hình thức thanh toán thay vì người cần thanh toán, giao dịch bằng tiền mặt, họ có thể chuyển dòng tiền của mình thông qua các tài khoản trực tuyến.

## Cổng thanh toán trực tuyến

Cổng thanh toán trực tuyến là dịch vụ cho phép thanh toán trực tuyến, có khả năng kết nối tài khoản thanh toán của người thanh toán và người được thanh toán để thực hiện giao dịch. Nhiệm vụ chính của cổng thanh toán là xử lý và xác minh tính đúng đắn của yêu cầu thanh toán, và hơn hết nó chỉ là dịch vụ trung gian thanh toán, không phải là dạng lưu trữ giá trị.

## Cơ chế thanh toán trực tuyến

Thông thường hệ thống của đơn vị chấp nhận thanh toán gửi thông tin yêu cầu thanh toán đến cổng thanh toán. Sau khi giao dịch hoàn tất, cổng thanh toán gửi trả lại hệ thống đơn vị chấp nhận thanh toán mã kết quả giao dịch, số tiền đã thanh toán, mã giao dịch (hay còn gọi là mã tham chiếu) và chuỗi mã hóa dữ liệu (thông qua hàm băm dữ liệu) nhằm kiểm tra tính toàn vẹn của dữ liệu gửi về, đảm bảo dữ liệu không bị thay đổi trong quá trình gửi từ cổng thanh toán đến hệ thống đơn vị chấp nhận thanh toán.

# Giao hàng trong thương mại điện tử

Trong thương mại điện tử sau giai đoạn bên mua và bên bán đồng ý thỏa thuận giao dịch hàng hóa, thì hàng hóa cần được vận chuyển để đến được bên mua, việc này cần một đơn vị để thực hiện công tác này, đó là đơn vị vận chuyển (còn được gọi là đơn vị logistics). Vai trò của đơn vị vận chuyển rất quan trọng trong thương mại điện tử giúp cho sàn thương mại điện tử có khả năng thao tác đơn hàng tự động nhờ kết hợp với hệ thống đơn vị vận chuyển.

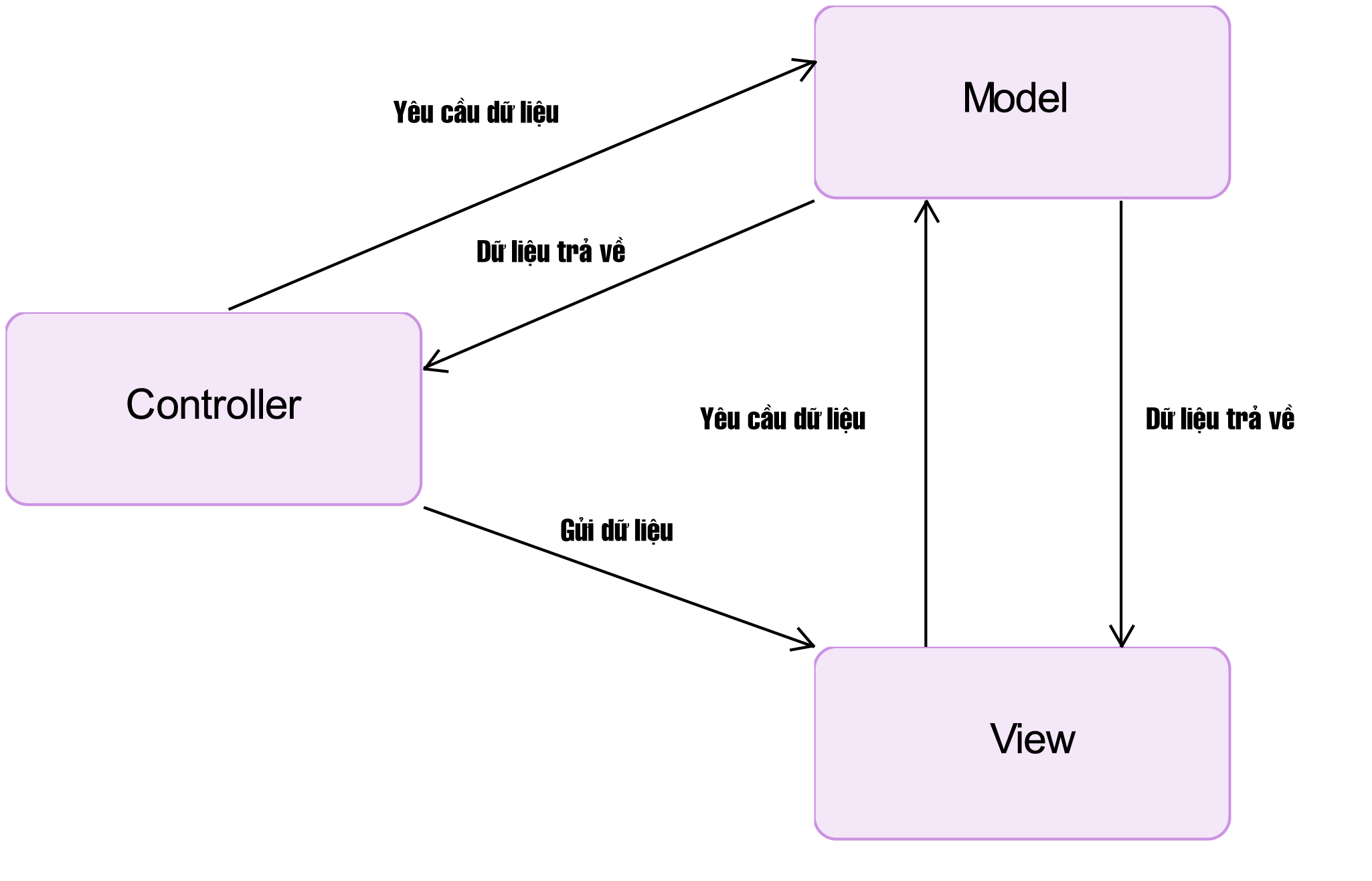
# Phát triển PHP Framework cho ứng dụng web

## Mô hình MVC

Mô hình MVC (Model-View-Controller) là một architectural design pattern được sử dụng phổ biến cho việc phát triển giao diện người dùng, chia một ứng dụng thành ba phần kết nối với nhau. Nhằm để tách biệt phần xử lý dữ liệu đầu vào và phần trình diễn dữ liệu đầu ra về phía người dùng, mô hình này còn có một lợi thế cho việc phát triển song song trong quy trình phần mềm.

MVC truyền thống được sử dụng cho phát triển ứng dụng giao diện đồ họa desktop, tuy nhiên với ưu điểm của nó nên hiện nay nó đã phổ biến trong ứng dụng web.

Có nhiều ý kiến cho rằng ở mô hình MVC thành phần View không được truy cập trực tiếp thành phần Model tuy nhiên ý kiến này sẽ làm Controller phải trao dữ liệu cho thành phần View từ thành phần Model, nếu như vậy, vì một lý do nào đó cần phải thay đổi Model sẽ dẫn đến phải thay đổi cả Controller chứ không phải chỉ thay đổi View. Nên sơ đồ mô hình MVC sẽ được trình bày như thế này cho phép thành phần View có thể truy cập dữ liệu từ thành phần Model một cách trực tiếp.



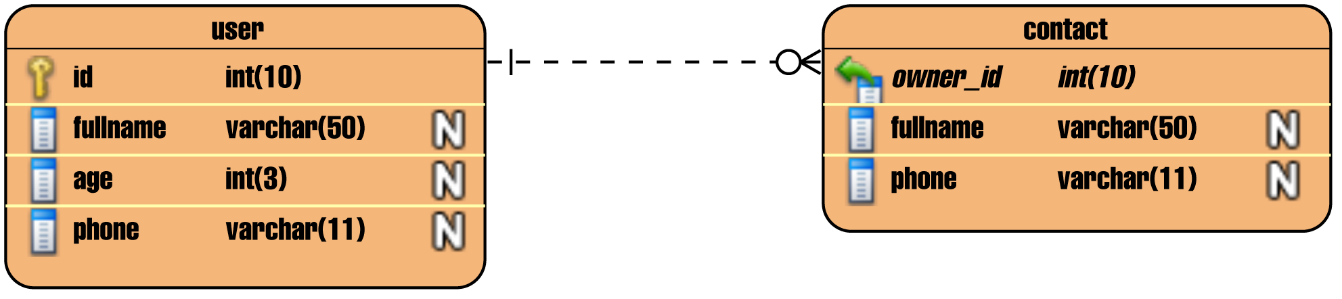
Hình 1: Mô hình MVC

## Các thành phần của MVC

### Model

Thành phần này giữ vai trò là thành phần dữ liệu, xử lý quy trình nghiệp vụ của ứng dụng

**Ví dụ**: Model có thể tương ứng với một thể hiện của một dòng ở cơ sở dữ liệu, cụ thể trong sơ đồ thực thể dưới đây, model là **user** và **contact**.



Ngoài ra Model còn có thể là các đối tượng hỗ trợ cho thành phần View để thực hiện trình diễn như: PaginationModel (dùng chứa thông tin phân trang cho một View)

### View

Thành phần chứa dữ liệu được trao bởi Controller và có nhiệm vụ xuất dữ liệu đầu ra, đối với ứng dụng web thì View xuất dữ liệu trả về cho trình duyệt của người dùng.

### Controller

Là thành phần quan trọng trong mô hình MVC, là thành phần điều khiển logic xử lý yêu cầu, là thành phần trung gian cho giao tiếp giữa View và Model. Thông thường việc cài đặt MVC Framework ở ứng dụng Web thì Controller sẽ được kích hoạt bởi một bộ điều phối (dispatcher) dựa vào bộ dữ liệu của bộ định tuyến (Router).

## Cài đặt PHP MVC Framework

### Phụ thuộc của việc cài đặt

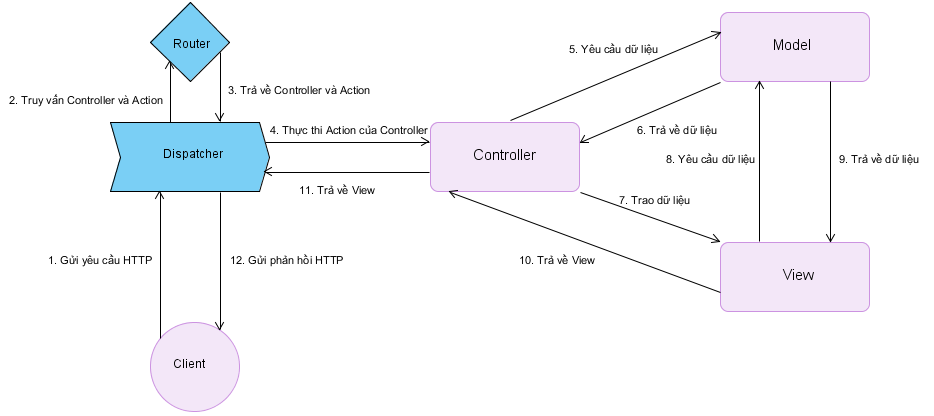
Chương trình web server là Apache có module mod\_rewrite, nhằm hỗ trợ chuyển request phù hợp về dispatcher.

Chương trình thông dịch PHP có version từ 5.3.0, nhằm hỗ trợ namespace, autoload, call\_user\_func\_array và reflection.

### Khái niệm trong Framework

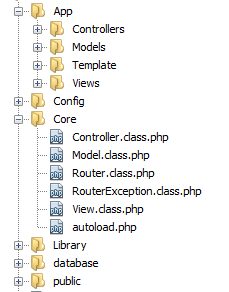
* Controller và Action: Controller là một lớp trong ứng dụng chứa các phương thức (được gọi là Action) sẽ được triệu gọi bởi dispatcher để xử lý các yêu cầu.
* View: là một lớp được Controller nạp dữ liệu vào và có các phương thức render nhằm để xuất dữ liệu đầu ra phù hợp.
* Model: là lớp đại điện cho mô hình dữ liệu có các phương thức để kiểm tra tính hợp lệ dữ liệu trên mô hình.
* Router: bộ định tuyến được sử dụng để tìm ra Action của một Controller phù hợp theo yêu cầu của dispatcher để gọi thực thi phương thức Action của lớp Controller.
* Layout: Là thành phần của View nếu được thiết lập mặc định nhờ vào tập tin khai báo layout có tên **\_ViewStart.phphtml**, khi xuất View sẽ có chung layout. Nếu không muốn có thể cấu hình lại layout khác hoặc là bỏ layout khi render

### Vận hành của Framework



Hình 2: Mô hình vận hành MVC Framework

### Cấu trúc thư mục



Hình 3: Mô tả cấu trúc thư mục của Framework

### Các chức năng chính

#### Router

Cấu hình bộ định tuyến đường dẫn của yêu cầu HTTP qua một hay nhiều lần cấu hình từ điển định tuyến, thứ tự tìm kiếm theo thứ tự khai báo các từ điển, nếu tìm thấy từ điển phù hợp thì thực hiện gọi Action của Controller tương ứng.

**Phương thức mapRoute**: thiết lập từ điển định tuyến trong Framework

**public** **function** mapRoute(string $pattern, array $defaultParams = []) : void;

Mô tả tham số:

* pattern: mẫu đường dẫn được tham số hóa dưới dạng chuỗi có thể bao gồm các nhóm tham số ở dạng **{tên tham số}** hoặc **{tên tham số:** **"biểu thức quy tắc mô tả tham số"}**, trong đó có thể có hai tham số **controller** và **action**.
* defaultParams: tham số mặc định nếu như trong pattern không mô tả.

Giá trị trả về:

Không có giá trị trả về.

Ví dụ:

#Khai báo route trước khi thực hiện mapRoute…

$route->mapRoute('{controller}/{action}', []);

Sau khi thực hiện cấu hình này đường dẫn của các yêu cầu đến, sẽ kích hoạt gọi Action của Controller tương ứng trong bảng sau:

|  |  |  |
| --- | --- | --- |
| **Đường dẫn của yêu cầu HTTP** | **Controller** | **Action** |
| Home/Index | Home | Index |
| User/Info | User | Info |
| User | Không tìm thấy Action phù hợp | |

#Khai báo route trước khi thực hiện mapRoute…

$route->mapRoute('{controller}', ['action' => 'Index']);

Sau khi thực hiện cấu hình này đường dẫn của các yêu cầu đến, sẽ kích hoạt gọi Action của Controller tương ứng trong bảng sau:

|  |  |  |
| --- | --- | --- |
| **Đường dẫn của yêu cầu HTTP** | **Controller** | **Action** |
| Admin | Admin | Index |
| User | User | Index |

#Khai báo route trước khi thực hiện mapRoute…

$route->mapRoute('{controller}/{action}/{id:"\d+"}', []);

Sau khi thực hiện cấu hình này đường dẫn của các yêu cầu đến, sẽ kích hoạt gọi Action của Controller tương ứng trong bảng sau:

|  |  |  |
| --- | --- | --- |
| **Đường dẫn của yêu cầu HTTP** | **Controller** | **Action** |
| User/UpdateInfo/123 | User | UpdateInfo  (có kèm tham số id="123" vào phương thức UpdateInfo) |
| User/UpdateInfo/abc | Không tìm thấy Action phù hợp | |
| User/UpdateInfo | Không tìm thấy Action phù hợp | |

#Khai báo route trước khi thực hiện mapRoute…

$route->mapRoute('{controller}/{action}/{id:"\d{3}"}', []);

$route->mapRoute('{controller}/{action}', []);

Sau khi thực hiện cấu hình này đường dẫn của các yêu cầu đến, sẽ kích hoạt gọi Action của Controller tương ứng trong bảng sau:

|  |  |  |
| --- | --- | --- |
| **Đường dẫn của yêu cầu HTTP** | **Controller** | **Action** |
| User/UpdateInfo/123 | User | UpdateInfo  (có kèm tham số id="123" vào phương thức UpdateInfo) |
| User/UpdateInfo | User | UpdateInfo |

#### Tham số yêu cầu HTTP nhận được vào Action của Controller được gọi

Khai báo phương thức Action của Controller để được bộ điều phối gọi thực thi khi có yêu cầu HTTP đến, sẽ có thể thêm tham số vào phương thức Action này, bao gồm kiểu dữ liệu nguyên thủy và kiểu dữ liệu lớp (thường là các Model).

**Ví dụ:** khai báo Action Info của Controller User nhận vào tham số id

**public** **class** UserController **extends** \Core\Controller{

**public** **function** Info($id){

}

}

Khi Action Info của Controller User được bộ điều phối gọi thực thi, Action Info này sẽ có tham số **id** được bộ điều phối truyền vào, tham số **id** này xuất phát từ yêu cầu HTTP.

**Ví dụ:** khai báo Action Add của Controller User nhận vào tham số là UserModel (là một Model của ứng dụng)

Tập tin UserModel.class.php

#UserModel ở namespace \App\Models

**class** UserModel **extends** \Core\Model{

**public** $id, $fullname, $age, $phone;

}

Tập tin UserController.class.php

#UserController ở namespace \App\Controllers

**public** **class** UserController **extends** \Core\Controller{

**public** **function** Add(UserModel $user){

}

}

Khi Action Add của Controller User được bộ biều phối gọi thực thi, Action Add này sẽ có tham số **user** là thể hiện của lớp UserModel, lúc này Action Add có thể truy cập được các tham số của yêu cầu HTTP (cặp khóa giá trị) thông qua tham số **user,** giả sử yêu cầu HTTP được mô tả tóm tắt trong bảng sau:

|  |  |
| --- | --- |
| **Loại yêu cầu** | POST |
| **URL yêu cầu** | /User/Add |
| **Phần nội dung** | id=123&fullname=nguyenvana&age=23&phone=0123456789 |

Lúc này tham số **user** sẽ có các thuộc tính và giá trị như trong bảng sau:

|  |  |
| --- | --- |
| **Thuộc tính** | **Giá trị** |
| id | 123 |
| fullname | nguyenvana |
| age | 23 |
| phone | 0123456789 |

## Chương 2: mô tả hệ thống

### Mô tả bài toán

### Phân tích bài toán

#### Các yêu cầu chức năng

#### Các yêu cầu phi chức năng

#### Các yêu cầu giao tiếp bên ngoài

### Lựa chọn và đánh giá và giải pháp

## chương 3: thiết kế và cài đặt giải pháp

### Tổng quan về hệ thống

### Kiến trúc hệ thống

#### thiết kế kiến trúc

#### mô tả sự phân rã

#### cơ sở thiết kế

### Thiết kế dữ liệu

#### mô tả dữ liệu

#### từ điển dữ liệu

### Thiết kế theo chức năng

#### chức năng đăng nhập

#### …

## CHƯƠNG 4: Kiểm thử và đánh giá

### Giới thiệu

#### Mục tiêu

#### Phạm vi kiểm thử

### Chi tiết kế hoạch kiểm thử

#### Các chức năng sẽ được kiểm thử

#### Cách tiếp cận

#### Tiêu chí kiểm thử thành công / thất bại

#### Tiêu chí đình chỉ và yêu cầu bắt đầu lại

### Quản lý kiểm thử

#### Các hoạt động / công việc được lập kế hoạch, sự tiến hành kiểm thử

#### Môi trường kiểm thử

#### Tài nguyên và sự cấp phát chúng

#### Huấn luyện

### Các trường hợp kiểm thử

#### Chức năng đăng nhập

#### Chức năng tìm kiếm

### Kết quả kiểm thử

#### Kiểm thử lần 1

#### Kiểm thử lần 2

## Phần kết luận

### Kết quả đạt được

#### Về lý thuyết

#### Về chương trình

#### Hạn chế

### Hướng phát triển

## Tài liệu tham khảo

## Phụ lục 1: hướng dẫn cài đặt

## Phụ lục 2: bảng công việc thực hiện