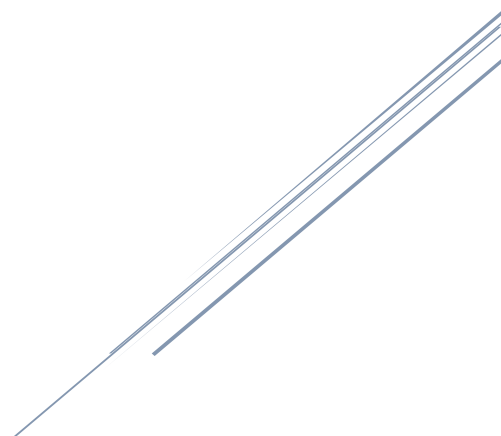




UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Grade UIT - Phần mềm thông báo điểm tự động

Đặng Minh Tiến -16521215



MỤC LỤC

MỤC LỤC	1
PHẦN 1: GIỚI THIỆU CHUNG.....	2
1.1 Tên đồ án	2
PHẦN 2: MÔ HÌNH MVC.....	2
2.1 Giới thiệu mô hình MVC trong Php	2
2.2 Ưu và nhược điểm mô hình MVC	3
2.2.1 Ưu điểm mô hình MVC	3
2.2.2 Nhược điểm mô hình MVC	3
PHẦN 3: GIỚI THIỆU PHẦN MỀM.....	3
3.1 CHỨC NĂNG CHÍNH.....	3
3.2 CÁCH HOẠT ĐỘNG CỦA PHẦN MỀM	3
3.2.1 HomeController	3
3.2.2 ProcessController.....	4
3.2.3 SeeController	6
3.2.4 UserController	6
3.2.5 Webhook Facebook	6
3.3 Demo ứng dụng Grade UIT	8
PHẦN 4: TÀI LIỆU THAM KHẢO	9

PHẦN 1: GIỚI THIỆU CHUNG

1.1 TÊN ĐỒ ÁN

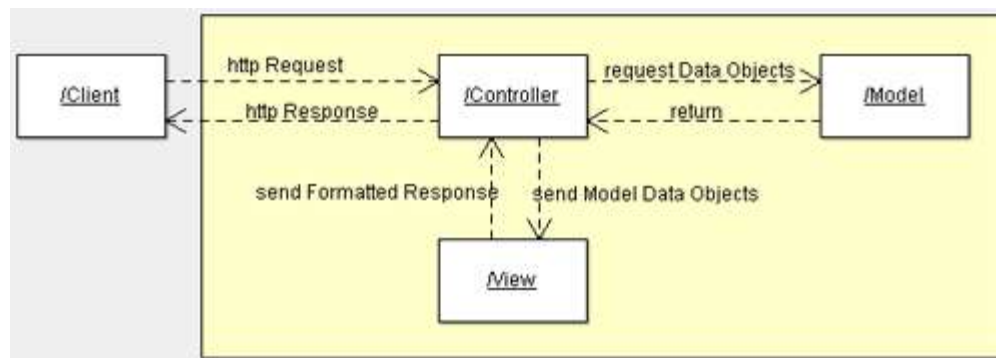
- Tên ngắn gọn: **Grade UIT**
- Tên đầy đủ: **Phần mềm báo điểm tự động**
- Ngôn ngữ code: PHP, HTML, Javascript, CSS, Mysql

PHẦN 2: MÔ HÌNH MVC

2.1 GIỚI THIỆU MÔ HÌNH MVC TRONG PHP

Mô hình MVC là viết tắt của 3 chữ Model, View, Controller. Đây là một mô hình kiến trúc phần mềm được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình này được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình web. 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. Trong PHP hiện tại có khá nhiều Framework và tất cả đều xây dựng từ mô hình MVC.

Hình ảnh minh họa các mô hình MVC hoạt động trong môi trường web:



Model: Thể hiện các cấu trúc dữ liệu. Các lớp thuộc thành phần Model thường thực hiện các tác vụ như truy vấn, thêm, xóa, cập nhật dữ liệu. Khi dữ liệu trong Model thay đổi, thành phần View sẽ được cập nhật lại. Nói đơn giản hơn, Model là lớp thao tác với database là chính.

View: là thành phần thể hiện dữ liệu trong Model thành các giao diện tương tác với người sử dụng. Một mô hình có thể có nhiều View phụ thuộc vào các mục đích khác nhau. Nói đơn giản hơn, View là lớp hiển thị dữ liệu ra bên ngoài cho người dùng xem.

Controller: Đóng vai trò trung gian giữa Model và View. Thông tin người dùng từ View được gửi cho Controller xử lý, sau đó Controller tương tác với Model để lấy dữ liệu được yêu cầu, sau cùng Controller trả dữ liệu này về cho View. Nói đơn giản hơn, Controller là lớp điều khiển, có chức năng điều khiển các hành vi, yêu cầu.

Nhìn vào mô hình MVC thì giữa Model và View không hề có mối liên hệ mà nó sẽ thông qua controller để giao tiếp với nhau.

Mô hình MVC thường được sử dụng trong các ứng dụng web, vì thành phần View (mã HTML/XHTML) được sinh ra từ các ngôn ngữ thiết kế website. Thành phần Controller sẽ nhận các dữ liệu GET/POST, xử lý những dữ liệu này, sau đó chuyển sang Model xử lý. Model sẽ trả dữ liệu về phía Controller, sau đó Controller sinh mã HTML/XHTML để thể hiện trên View.

2.2 ƯU VÀ NHƯỢC ĐIỂM MÔ HÌNH MVC

2.2.1 ƯU ĐIỂM MÔ HÌNH MVC

- Hệ thống phân ra từng phần riêng biệt nên dễ dàng phát triển.
- Chia thành nhiều module nhỏ nên nhiều người có thể làm chung dự án.
- Vấn đề bảo trì tương đối ổn, dễ nâng cấp.
- Dễ dàng tìm ra lỗi trong quá trình xây dựng.
- Tầng Model và View có thể thay đổi, chỉnh sửa 1 cách độc lập. Ta có thể tạo các ứng dụng giao tiếp thông minh hơn chỉ bằng cách thay đổi View hoặc cũng có thể thay thế Database bằng cách sử dụng Web service, ...

2.2.2 NHƯỢC ĐIỂM MÔ HÌNH MVC

- Viết theo mô hình MVC sẽ làm hệ thống chạy chậm hơn PHP thuần, tuy nhiên sự khác biệt này là không quá lớn.
- Việc xây dựng, tạo cấu trúc các file sẽ tốn thời gian nhưng về sau lại sử dụng rất nhanh và chuyên nghiệp.

PHẦN 3: GIỚI THIỆU PHẦN MỀM

3.1 CHỨC NĂNG CHÍNH

- Phần mềm chạy trên nền tảng web, có chức năng xem thời khóa biểu, thông báo điểm thi qua 3 phương tiện (email, messenger và trên trang web chính).

3.2 CÁCH HOẠT ĐỘNG CỦA PHẦN MỀM

Phần mềm gồm 4 controller chính dùng để nhận lệnh từ người dùng.

3.2.1 HOMECONTROLLER

Được dùng để xử lý hiển thị trang chủ, ở đây ứng dụng sẽ điều hướng về trang quản lý user.

```

class HomeController
{
    public function index($params)
    {
        header("Location: /user");
    }
}

```

3.2.2 PROCESSCONTROLLER

Những hàm xử lý chính như `sendMessage`, `sendEmail`, `CheckScore`, `LoginWebUIT`, `GetScheduleFromWebUIT`, `GetScoreFromWebUIT`

+ *sendMessage*: Được dùng để gửi tin nhắn messenger cho người dùng. Sử dụng CURL để post data json lên api của facebook.

```

public function sendMessageCurl($message, $senderid)
{
    $jsonData = "{
        \"recipient\":{
            \"id\": \"$senderid\"
        },
        \"message\":{
            \"text\": \"$message\"
        }
    }";

    $url = "https://graph.facebook.com/v2.6/me/messages?access_token=".$this->tokenfb;

    $jsonDataEncoded = $jsonData;
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonDataEncoded);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $result = curl_exec($ch);
    curl_close($ch);

    return $result;
}

```

+ *SendEmail*: Có nhiệm vụ gửi email đến người dùng. Sử dụng SMTP của google để gửi mail.

+ *LoginWebUIT*: Đăng nhập vào website UIT

Tiến hành lấy MSSV trong DB và giải mã password, sau đó tiến hành POST dữ liệu lên web trường để đăng nhập.

```

public function LoginWebUIT($user)
{
    $username = $user->getMssv();
    $password = Encrypt::getInstance()->decode($user->getKeypw(), $user->getPass());
    $postdata = "name=".addslashes($username)."&pass=".addslashes($password).
        '&form_build_id=form-W1lt3MIKTXdLaVPkKFP9VXXaHb7XpbbZyrrLHtz375k&form_id=user_login&
        op=Đăng%20nhập';
}

```

Sau khởi tạo biến curl và cấu hình các thông số cơ bản như URL post, các giá trị thuộc tính post, và tiến hành thực thi.

```
$ch = curl_init();
$url="https://daa.uit.edu.vn/user/login/";
curl_setopt ($ch, CURLOPT_URL, $url);
curl_setopt ($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt ($ch, CURLOPT_USERAGENT, "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6");
curl_setopt ($ch, CURLOPT_TIMEOUT, 120);
curl_setopt ($ch, CURLOPT_FOLLOWLOCATION, 0);
curl_setopt ($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt ($ch, CURLOPT_COOKIEJAR, $this->cookie);
curl_setopt ($ch, CURLOPT_REFERER, $url);
curl_setopt ($ch, CURLOPT_POSTFIELDS, $postdata);
curl_setopt ($ch, CURLOPT_POST, 1);
$dataweb = curl_exec($ch);
```

+ *GetScoreFromWebUIT*:

Sử dụng CURL và hàm đăng nhập *LoginWebUIT* đã đề cập ở trên để đăng nhập trước.

Sau đó dùng hàm *preg_match* để lọc lấy đoạn html chứa đoạn hiển thị điểm mà ứng dụng cần.

```
public function GetScoreFromWebUIT($user)
{
    global $conn;
    $ch = $this->LoginWebUIT($user); // đăng nhập
    if ($ch == false)
        return false;
    curl_setopt ($ch, CURLOPT_URL, "https://daa.uit.edu.vn/print/sinhvien/kqhoclap");
    curl_setopt($ch, CURLOPT_POST, false);
    $result = curl_exec($ch);
    curl_close($ch);
    preg_match('#<center><h1>(.*?)</strong></font>#s', $result, $res);
    $data = '<div class="print-content">'.$res[0].</div>';
    return $data;
}
```

+ *CheckScore*: Kiểm tra điểm cập nhật và thông báo cho người dùng nếu có.

```
function CheckScoreById($id)
{
    global $conn, $linkSite;
    $UserController = new UserController();
    $user = $UserController->findUserById($id);

    if ($user->getActive() != 1) // nếu không đang active thì bỏ qua
        return;

    $username = $user->getMssv();
    $keypw = $user->getKeypw();
    $password = Encrypt::getInstance()->decode($keypw, $user->getPass());

    $oldcode = $user->getCode();

    $data = $this->GetScoreFromWebUIT($user);
```

Chúng ta sẽ sử dụng hàm *GetScoreFromWebUIT* để lấy thông tin điểm hiện tại. Cách kiểm tra điểm có thay đổi hay không là so sánh đoạn html get được, được lưu trước đó và vừa lấy mới. Tuy nhiên, để việc so sánh đơn giản hơn, chúng ta sẽ băm đoạn mã html đó bằng hàm băm md5.

Khi kết quả băm của đoạn html vừa lấy được khác với đoạn băm được lưu trước đó thì đồng nghĩa với việc điểm của người dùng đã thay đổi và tiến hành thông báo cho người dùng.

3.2.3 SEECONTROLLER

Dùng để xử lý hiển thị tra cứu điểm, thời khóa biểu nhanh cho người dùng.

3.2.4 USERCONTROLLER

Quản lý những thao tác đăng nhập, đăng kí tài khoản, đăng xuất tài khoản.

- Thao tác đăng nhập: dùng Session để lưu trữ trạng thái đăng nhập của người dùng. Phần mềm sẽ mã hóa mật khẩu người dùng nhập sau đó đem kiểm tra với mật khẩu trong database, nếu chính xác thì sẽ gán vào session để đánh dấu đăng nhập thành công.
- Thao tác đăng kí: sau khi người dùng nhập thông tin đăng kí và nhấn nút “đăng kí” thì dữ liệu sẽ được gửi đi bằng phương thức POST. Tiếp theo, phần mềm lấy dữ liệu đó nhập vào database bằng các câu lệnh truy vấn mysql.
- Thao tác đăng xuất: dùng hàm `session_destroy` để hủy tất cả các session và quay về màn hình đăng nhập.

3.2.5 WEBHOOK FACEBOOK

Webhook facebook (File msg.php) cho phép hệ thống nhận thông báo liên quan tin nhắn của người dùng gửi đến fanpage. Ứng dụng có thể nhận các tin nhắn này để phản hồi tự động, hỗ trợ người dùng đăng ký tài khoản, tra cứu điểm, thời khóa biểu và đồng thời sẽ có thể phát triển các tiện ích liên quan trong tương lai.

```
$botman->hears('dangky', function (BotMan $bot)
{
    global $linkSite;
    $user = new UserModel();
    if ($user->findUserByIdmsg($bot->getUser()->getId())==null)
    {
        $bot->reply('Bấm vào đường dẫn sau để đăng ký tài khoản: '.$linkSite.
            'user/register/'.$bot->getUser()->getId());
    }
    else
    {
        $bot->reply('Bạn đã đăng ký trước đó với tài khoản: '.$user->getMssv().'. Để
            thay đổi vui lòng sử dụng lệnh remove để xóa tài khoản cũ.');
```

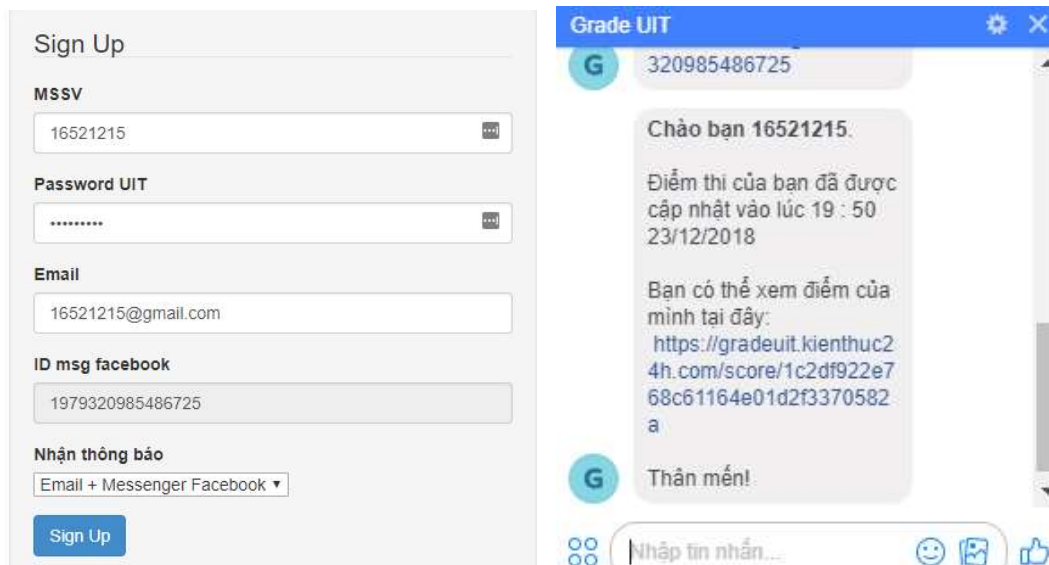
(Hình ảnh minh họa 1 và xử lý để phản hồi người dùng)

3.3 DEMO ỨNG DỤNG GRADE UIT

Đường dẫn chat bot live: <https://www.facebook.com/GradeUIT/>



Khi sử dụng lệnh **dangky**, chatbox sẽ tự phản hồi đường dẫn để người dùng có thể đăng ký tài khoản.



Sau khi đăng ký xong, ứng dụng sẽ tự động cập nhật và phản hồi lần đầu tiên. Lúc này mọi thứ đã hoàn tất.

Ngoài ra người dùng còn có thể sử dụng các lệnh như: **tkb**, **diem** để tra cứu nhanh thời khóa biểu và điểm học tập.

PHẦN 4: TÀI LIỆU THAM KHẢO

- [1] <https://github.com/themainframe/php-mvc-sample>
- [2] <https://github.com/PHPMailer/PHPMailer>
- [3] <https://github.com/botman/botman>
- [4] ...