

PHPフレームワークを使った Webアプリの試作

吉 武 春 光

1. まえがき

筆者は、2006年より、SDRT 表記を用いた談話の意味処理の研究を進めており、SDRT 表記がコンピュータ処理に適したものであるかどうかの検証のために、実際に、コンピュータ上でプログラムを作成する段階にさしかかっている。ここでの問題は、プログラムが複雑になるために、一人でプログラムをゼロから書いていると時間がかかりすぎるということである。一方、最近では Web アプリケーション（略して Web アプリ）と呼ばれる、Web 上で動作するプログラムが世の中の主流を占めているために、Web アプリを簡易に開発するための「フレームワーク」と呼ばれるシステムが広範囲に使われるようになってきた。そこで、筆者も効率良い開発を目指して、フレームワークを使ってプログラムを作成するという方針を立てた。そこで、まず、フレームワークの習得を兼ねて、実用になる Web アプリを開発することにした。本稿では、プログラミング言語 PHP 用のフレームワークである CodeIgniter を使用して、メーリングリスト管理システムを構築した結果を報告するものである。

2. フレームワークとは何か？

従来より、ソフトウェア開発においては、汎用に使われる関数などをライブラリと称して、開発環境ソフト側が、一括してプログラマ側に提供してきた。一般に、アプリ開発においては、どのアプリでも必要になる機能はライブラリを使って、ほぼ似たような組み合わせで実現されてきた。インターネット利用が当たり前で、Web アプリ開発が日々、頻繁に、行われるにつれ、利用者認証部分や入出力部分などの汎用的な機能をまとめて提供しようとする動きが始まった。これはフレームワーク（framework、枠組み）と呼ばれるもので、プログラミング言語 Ruby 用に開発された Ruby on Rails（2005年12月登場）が

最初であった。Ruby on Rails のサイトには、僅か15分でプログシステムを作成することが出来るデモが置いてある。もっとも、実用になる Web アプリにするためには、見栄えなどに、更に手間を掛ける必要はあるのだが、それまでの Web アプリ開発にかかった時間や労力と比べて、格段に生産性が向上したものであった。更に、プログラミングにおける不具合の混入も減ることになった。その結果、多くの Web サイトが Ruby on Rails を使って開発された。

Ruby on Rails のもう1つの特徴が、表1のような MVC (Model View Controller) モデルと呼ばれる構造であった。MVC モデルを採用することにより、アプリ全体の見通しが良くなった。

表1 MVC モデルの構造

要素	処理の概要
Model	データの格納と管理を行う部分
View	Model のデータを取り出して表示する部分
Controller	ユーザからの入力を受け付け、Model と View に処理を割り振る。

フレームワークの3番目の特徴は、開発環境だけではなく、実行環境にもなっていることである。即ち、フレームワークを使って開発したアプリを実行させるためには、開発に使用したフレームワーク上でなければならない。従来のソフトウェア開発においては、ライブラリを組み合わせることでアプリを開発しており、アプリの組み合わせに自由度があることが利点でもあるが、逆に、アプリ全体の動作を管理するためには手数が多く必要であった。フレームワークでは、逆に、アプリ全体の動作を管理する処理過程が MVC モデルとして半固定であるので、その処理過程に適合するプログラムを利用者が作成するのである。これは、Ralph & Brian 1988 が説明した「制御の反転 (Inversion of Control, IoC)」を実現したものだと言われている。

極論すれば、フレームワークは、Web アプリの半完成品とも言えるのである。

この Ruby on Rails の成功に影響を受け、他のプログラミング言語（例えば、Perl、PHP、Java、Python、ASP.Net など）用にもフレームワークが次々に開発された。筆者は、プログラミング言語 PHP で書かれた e ラーニングシステム (Moodle) との連携を念頭に置いているので、今回は、PHP 用のフレームワークを使用することにした。

3. CodeIgniter のインストール

PHP 用のフレームワークとしては、CakePHP、CodeIgniter、Symfony、Zend Framework が有名である。今回は、この中で、もっとも小規模である CodeIgniter を使用することにした。CodeIgniter は、一言でシンプルさが特徴である。CodeIgniter には、オンラインマニュアルが組み込まれており、いつでも Web 上で参照が可能である。日本 CodeIgniter ユーザ会のサイトには、日本語に翻訳した「CodeIgniter ユーザガイド 日本語版 Version 1.7.2」が置いてある。日本 CodeIgniter ユーザ会のオフィシャルブックとして、河合・鈴木・安藤(2008)が発行されている。

3.1 CodeIgniter のインストール

日本 CodeIgniter ユーザ会のサイトより辿り、sourceforge.jp に置いてある日本語化された CodeIgniter 1.7.2-1 ja 版 (zip 形式でサイズが約 4MB) をダウンロードした。インストールは極めて簡単で、ソースを展開し、Web サーバ上に置くだけである。河合・鈴木・安藤(2008)では、GUI 開発環境である Eclipse/PDT に CodeIgniter を組み込んで使う方法を説明しているが、筆者は使い慣れた emacs テキストエディタを使ってソースコードを入力・修正したかったので、Eclipse/PDT は使わないことにした。

CodeIgniter は、PHP のバージョンや環境の違いによらず動作するように配慮されているおり、筆者は次の 2 つの環境での動作を確認した。

表2 CodeIgniter の動作を確認した環境

ホスト名	dual	w3
OS	Ubuntu 10.04 (64bit)	Solaris 10 (64bit)
Web サーバ	Apache 2.2.14	Apache 2.2.16
PHP	5.3.2-1	5.2.6
MySQL	5.1.41	5.0.45

3.2 CodeIgniter の設定

以下では、w3 上の UID=yositake の環境下での例を示す。
インストール先は、yositake/public_html/ml/ とした。

① config.php

config.php は CodeIgniter の動作内容を設定するファイルであり、次の場所に作成する。最初にひな形となる config.php が置いてあるので、それを書き直す。

```
yositake /public_html/ml/system/application/config/config.php
```

以下では、変更箇所について記載する。Linux における diff コマンドの出力結果を載せている。先頭がマイナス - で始まる行が削除され、先頭がプラス + で始まる行が追加されていることを示している。

```
$config['base_url'] に自サイトの URL を記述する。
```

```
--- config.php.original 2009-02-05 06:44:40.000000000 +0900
+++ config.php 2010-04-07 16:57:11.955161000 +0900
@@ -11,7 +11,7 @@
| http://example.com/
|
| */
-$config['base_url'] = "http://example.com/";
+$config['base_url'] = "https://w3.seinan-gu.ac.jp/~yositake/ml/";
```

```
/*
```

mod_rewrite を使って index.php を書かなくてもアクセスできるようにする。

```
@@ -23,7 +23,7 @@
| variable so that it is blank.
|
|
*/
-$config['index_page'] = "index.php";
+$config['index_page'] = "";

/*
```

なお、index.php を書かなくてもアクセスできるようにするためには、この config.php の設定に加えて .htaccess ファイルの記述 (後述) が必要となる。

言語設定を日本語にする。

```
@@ -66,7 +66,7 @@
| than english.
|
|
*/
-$config['language'] = "english";
+$config['language'] = "japanese";

/*
```

URI に含める検索キーワードに日本語を許す。

```
@@ -123,7 +123,7 @@
| DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!
|
|
*/
-$config['permitted_uri_chars'] = 'a-z 0-9~%.:_¥-';
+$config['permitted_uri_chars'] = '一-脛 あ-? ア-ケ 1-z a-z 0-9~%.:_¥-';

/*
```

デバッグ情報を多く得る。

```
@@ -173,7 +173,7 @@
| your log files will fill up very fast.
|
| */
-$config['log_threshold'] = 0;
+$config['log_threshold'] = 4;
/*
```

セッションデータの暗号化の鍵となるランダムな文字列を指定する。

```
@@ -217,7 +217,7 @@
| enabled you MUST set an encryption key. See the user guide for info.
|
| */
-$config['encryption_key'] = "";
+$config['encryption_key'] = "vcyCp_WQI9ujmucaiA-14Taax-r55U8s";
/*
```

URL にクエリ文字列を使うために。

```
@@ -233,8 +233,8 @@
| */
$config['sess_cookie_name'] = 'ci_session';
$config['sess_expiration'] = 7200;
-$config['sess_encrypt_cookie'] = FALSE;
-$config['sess_use_database'] = FALSE;
+$config['sess_encrypt_cookie'] = TRUE;
+$config['sess_use_database'] = TRUE;
$config['sess_table_name'] = 'ci_sessions';
$config['sess_match_ip'] = FALSE;
$config['sess_match_useragent'] = TRUE;
```

4. mailing list 管理の必要性と、求められる機能

mailing list は電子メールの世界の同報通信の仕組みであり、メールマガジンや部署内の情報共有の手段として幅広く使われている。

4.1 UNIX システム上での mailing list システムの現状

Unix の世界で使っている電子メール配信システムは、sendmail というソフトウェアが出发点になっている。2000年頃より、sendmail に代わるソフトウェアとして Postfix や Qmail などが主流となっている。これらの電子メール配

信ソフトウェアは、`/etc/aliases` というファイルにテキストエディタを使って次のように、メールアドレスをカンマで区切って複数個記述する*と、左の mailing list 名宛に電子メールを同時に配信する機能を装備している。alias 管理とも呼ばれている。

表3 `/etc/aliases` における mailing list の記述

mailing list 名: メールアドレス1, メールアドレス2,..., メールアドレスn または mailing list 名:include:メールアドレスを記述したファイル
--

alias 管理は手軽で確実という特徴があるので、現在でも幅広く使われている。

一般に mailing list 用ソフトウェアと称しているものは、この `/etc/aliases` の管理を利用者権限にて所属する mailing list にのみ行えるようにしたものことである。Mailman や FML などが有名である。Mailman は、高機能で複雑なシステムになっており、mailing list を Web インターフェースにより一括管理することが出来る。一方、FML は、シンプルなシステムであり個人が mailing list を管理することに特化してある。しかし、設計が古いこともあり管理手段が電子メールになっており、Web インターフェースで使うことが出来ない。利用にとっては電子メールによる管理に馴染めない人もいるようだ。本学の情報処理センターでは、alias 管理と FML 管理を使い分けられているが、Web インターフェースにより利用者権限にて mailing list の管理が行えるシステムが望まれている。

4.2 alias 管理の Web インターフェース化

`/etc/aliases` というファイルの編集は管理者権限で行う必要があるが、表3の後者の include 記法を使うやり方は、メールアドレスを記述したファイルの所有者を、その mailing list の管理者とすることで、利用者権限にて mailing

* 実際には、`newaliases` というコマンドを入力することにより、`/etc/aliases` の内容を `/etc/aliases.db` というデータベースに変換する必要がある。

list の管理が行えるという特徴がある。但し、現状では、メールアドレスを記述したファイルの編集がテキストエディタを使用しなければならず、慣れていないことも手伝って編集ミスなどが混入する場合もあった。そこで、本研究では、この部分を PHP フレームワークを使うことにより、Web インターフェース化することにした。

4.3 実現させる機能

実際の運用を考慮して mailing list に所属するメンバーの追加、変更を行わせる機能を持たせることにした。そのためには、mailing list 管理者のユーザ認証を行い、mailing list 管理者が管理している mailing list のみを表示させ、希望する mailing list に所属するメンバーの追加、変更を Web インターフェースにて行うことが出来るようにした。

5. 試作

今回、試作したプログラムの処理の流れを図1に示す。実際に作成したプログラムは、別紙として添付する。

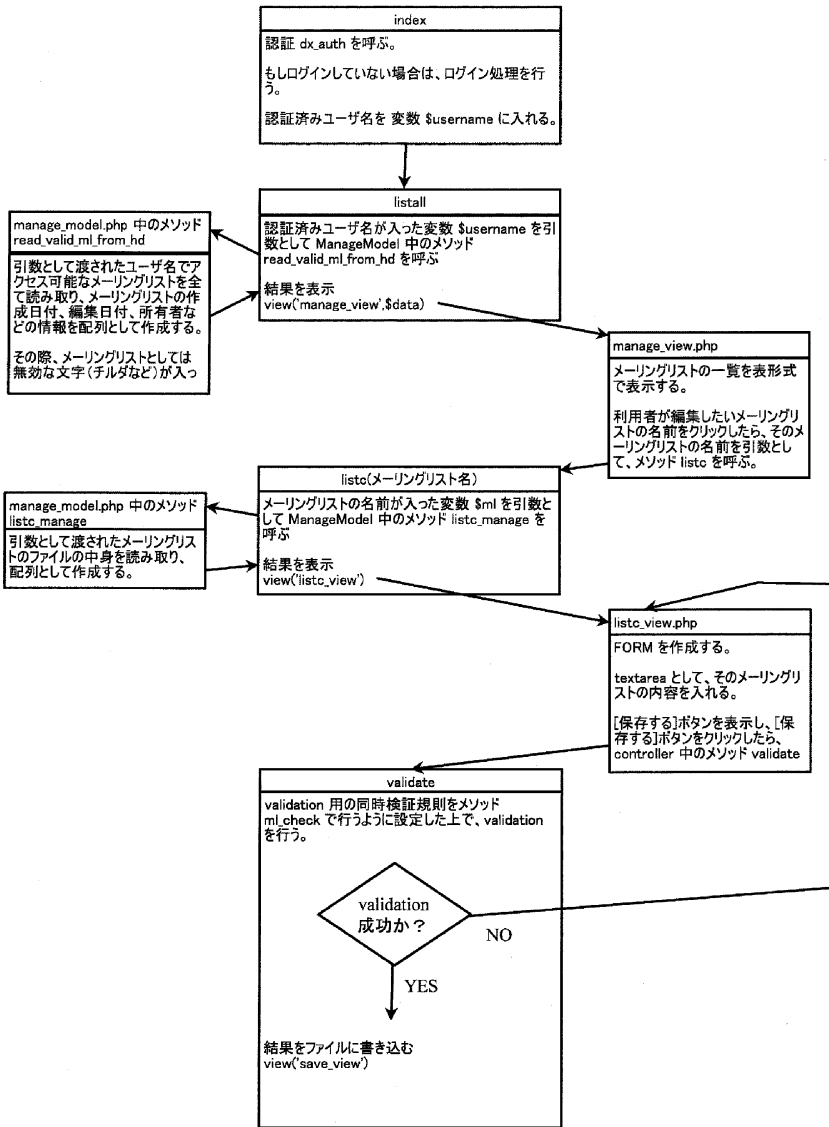


図1 試作プログラムの処理の流れ

5.1 認証機能

CodeIgniter には標準ではユーザ認証機能は含まれていない。そこで、外部の組織が開発し提供している認証ライブラリ FreakAuth, Rapyd Components, DX Auth を比較検討した結果、DX Auth を使用することにした。

5.2 バリデーション(validation) 機能

Web 画面でユーザに入力をさせる際に、入力文字の数をある数以下しか認めないとか、入力値として英数字しか認めないなどの制限をかける処理を validation と呼んでいる。CodeIgniter にも、バリデーション機能が備わっているので使用することにした。

使用する際の注意としては、view/listc_view.php と manage.php 中のメソッド validate とが再帰的に呼ばれている点である。また、バリデーションの規則はシンプルなものしか用意されていないので、複雑な規則を適用したい場合は、コールバック (callback) 機能を組み合わせる必要がある。

本プログラムでは manage.php 中のメソッド validate において

```
$this->form_validation->set_rules('mlc', 'ML コンテンツ', 'callback_ml_check');
```

と設定しておいてから、実際のバリデーション規則を manage.php 中のメソッド ml_check(\$str) に記述する。そして、変数 \$str に検査すべき値が入っているので、整形したあとで、

```
$this->form_validation->valid_email($email)
```

や

```
$this->form_validation->alpha_dash($email)
```

などのように、複数のバリデーション機能呼び、バリデーションが正しいと判断できたら TRUE を返し、正しくないと判断したら FALSE を返すようにした。

5.3 実行の様子

以下に、実際の実行画面を示す。

① トップページ

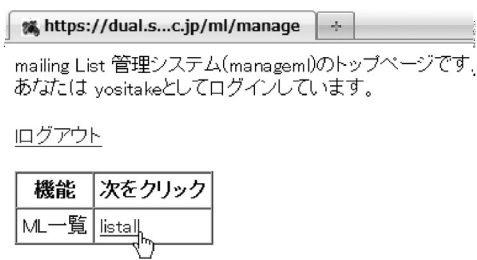


② ログイン処理

ユーザ名 yositake でログインしようとしている。



③ ログイン処理の後で、実際に listall を呼ぶ。



- ④ ユーザ名 yositake が編集可能な、つまり、ユーザ名 yositake がメーリングリストの管理者であるメーリングリストの一覧を表示する。

manageml

以下は ML の一覧です。

[トップページに戻る](#)

ML名	PATH	owner	mtime修正日付	ctime作成日付	atimeアクセス日付
bip	/mail/aliases/bip	yositake	2009.09.15 17:34	2010.10.31 17:37	2010.10.31 13:15
cc-com	/mail/aliases/cc-com	yositake	2010.07.02 13:52	2010.10.31 17:37	2010.10.31 17:35
cc-core	/mail/aliases/cc-core	yositake	2009.07.01 12:44	2010.10.31 17:37	2010.10.31 17:35

- ⑤ 次は、例として、メーリングリスト bip を選択し、編集可能画面になっている状態である。

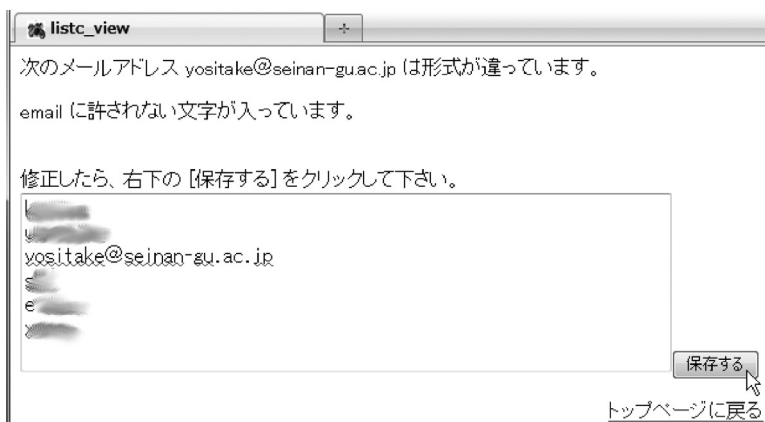
listc_view

登録されている ML メンバーの email address は次の通りです。
修正したら、右下の [保存する] をクリックして下さい。

k...
 u...
 yositake
 s...
 e...
 x...

[トップページに戻る](#)

- ⑥ 不正なメールアドレスとして許されない漢字コード@を使用した場合に、[保存ボタン]をクリックした状態で、バリデーション機能により、それが検出されている。



⑦ 編集が正常に行われたあと。



6. まとめ

本研究は、(吉武, 2007) (吉武, 2009)の延長線上行う予定の SDRT 表記のコンピュータ・システムでの実装に向けて、PHP フレームワークに慣れるというのが目的であった。そこで実際に作成したプログラムは、メーリングリストの管理を行わせるというものであった。一般に、Web 画面との入力や表示を行う大規模プログラムの場合、多くのプログラム行数が必要となるが、フレームワークを使うことにより、見通しが良く、更に、プログラム行数が少なく実現できたのは驚きであった。次は、PHP フレームワークを使って SDRT 表記のコンピュータ・システムでの実装を行い、SDRT 表記の妥当性の検証を行いたい。

参考論文

- 吉武春光(2007). "SDRTによる談話の意味記述." 西南学院大学商学論集 53巻 3・4合併号
- 吉武春光(2009). "Prologを使ったSDRT表記の実現." 西南学院大学商学論集 55巻 4号
- 佐藤匡剛(2007). "ソフトウェアのフレームワークとはなにか."
<<http://itpro.nikkeibp.co.jp/article/lecture/20070205/260697/>>
2010年4月28日アクセス
- "日本CodeIgniterユーザ会", <<http://codeigniter.jp/>>2010年4月28日アクセス
- "CodeIgniter配布サイト", <<http://codeigniter.com/>>2010年4月28日アクセス
- 河合勝彦、鈴木憲治、安藤建一(2008) "CodeIgniter徹底入門" 翔泳社
"DX Auth配布サイト", <http://dexcell.shinsengumiteam.com/dx_auth/>
2010年4月28日アクセス

付録

MVC モデルであるので、モデル、ビュー、コントローラという3種類のファイルが必要であるが、本プログラムでは、ビューファイルを2つ作成した。

モデルファイル manage_model.php

```
<?php
class Manage_model extends Model {
function Manage_model()
{
    parent::Model();
    $this->load->database();

    // helper
    $this->load->helper(array('form','url','file','date'));
}
function read_valid_ml_from_hd($username)
{
    //$datestring = "%Y.%m.%d - %H:%I";
    if (! $string = get_dir_file_info('/home/solomon/work/mail/aliases/',TRUE))
    { echo 'ファイルが読み込めません<br>'; }
    else
    {
        // echo '読んだファイルの情報は... <br>';
    }
}
```

```
foreach ($string as $key => $value)
{
    $owner = posix_getpwuid(fileowner($value['server_path']));
    //echo 'owner=' . $owner['name'] . '<br>';

    if (
        strpos($key, '#') === false
        &&
        strpos($key, "%x7e") === false // to reject チルダ
        &&
        strpos($key, "%xe2") === false // to reject チルダに似た文字...??
        &&
        strpos($value['server_path'], '/old') === false
    )
        // 上の if が成立するときに、以下の if を実行する。
        if (
            $username == "admin"
            ||
            $username == $owner['name']
        )
            // 上の if が成立するときに、以下の {} を実行する。
            {
                $ml_list[$key]['ml_name'] = $key;
                $ml_list[$key]['ml_path'] = $value['server_path'];
                $ml_list[$key]['ml_owner'] = $owner['name'];
                $ml_list[$key]['ml_info_all'] = $value;

                $ml_list[$key]['ml_mtime'] = date("Y.m.d H:i",filetime($value['server_path']));
                $ml_list[$key]['ml_ctime'] = date("Y.m.d H:i",filetime($value['server_path']));
                $ml_list[$key]['ml_atime'] = date("Y.m.d H:i",filetime($value['server_path']));
            }
    };
return $ml_list;
}

function listc_manage($ml)
{
    global $valid_ml_list;

    if (! $string = read_file( '/home/solomon/work/mail/aliases/' . $ml))
        { echo 'ファイルが読み込めません<br>'; }
    else
    {
        $ml_content[$ml]['ml_content'] = $string;

        return $ml_content;
    }
}
```

```
}  
}  
}  
?>
```

コントローラファイル manage.php

```
<?php  
class Manage extends Controller {  
function Manage ()  
{  
parent::Controller();  
  
global $valid_ml_list;  
  
$this->load->model('Manage_model');  
header( 'Content-Type: text/html; charset=UTF-8');  
// セッションクラスのロード  
$this->load->library('session');  
  
// table ライブラリのロード  
$this->load->library('table');  
  
// scaffolding  
$this->load->scaffolding('managemls');  
  
// helper  
$this->load->helper(array('form','url','file','date'));  
  
// DX Auth  
$this->load->library('DX_Auth');  
  
// フォーム・バリデーション  
$this->load->library('form_validation');  
}  
  
function index()  
{  
// プロファイラ  
$this->output->enable_profiler(TRUE);  
print 'mailing List 管理システム(manageml)のトップページです。 <br>';  
  
global $username;  
$username = $this->dx_auth->get_username();  
if ( ! $this->dx_auth->is_logged_in())
```



```
{
print '<a href="https://dual.seinan-gu.ac.jp/ml/auth/login">ログイン</a>';
print 'して下さい';
return;
}
else
{
// DX Auth でログインしたユーザ名を表示
print 'あなたは ';
echo $username;
print 'としてログインしています。 <br><br>';
}
//
// DX Auth からログアウト
print '<a href="https://dual.seinan-gu.ac.jp/ml/auth/logout">I ログアウト</a><br><br>';

//
$tmp1 = array ('table_open' => '<table border="1" cellpadding="4" cellspacing="0">');

$this->table->set_template($tmp1);
$data['見出し'][0] = '機能';
$data['見出し'][1] = '次をクリック';
$data['機能 1'][0] = 'ML 一覧';
$tmp1 = '<a href="https://dual.seinan-gu.ac.jp/ml/manage/listall/"';
$tmp2 = '>listall</a>';

$data['機能 1'][1] = $tmp1 . $username . $tmp2;

echo $this->table->generate($data);
}
function listall($username)
{
global $valid_ml_list;

// プロファイラー
$this->output->enable_profiler(TRUE);

//ランダムなチケットを生成し、セッションに保存します。
$this->ticket = md5(uniqid(mt_rand(), TRUE));
$this->session->set_userdata('ticket', $this->ticket);

// Manage_model をロード
$valid_ml_list = $this->Manage_model->read_valid_ml_from_hd($username);

// manage_view のビューをロードし表示します。
$data['result'] = $valid_ml_list;

$this->load->view('manage_view', $data);
```

```
}

function listc($ml)
{
    $this->load->helper(array('form', 'url'));
    $this->load->library('form_validation');

    global $ml_users;

    // プロファイラー
    $this->output->enable_profiler(TRUE);

    //ランダムなチケットを生成し、セッションに保存します。
    $this->ticket = md5(uniqid(mt_rand(), TRUE));
    $this->session->set_userdata('ticket', $this->ticket);

    // listc_manage_model をロード
    $got = $this->Manage_model->listc_manage($ml);
    $ml_users = $got;

    $data['result'] = $got;

    $this->load->view('listc_view');
}

function validate()
{
    $this->load->helper(array('form', 'url'));
    $this->load->library('form_validation');

    global $ml_users;

    // プロファイラー
    $this->output->enable_profiler(TRUE);

    $this->form_validation->set_rules('mlc', 'ML コンテンツ', 'callback_ml_check');

    if ($this->form_validation->run() == FALSE)
    {
        $this->load->view('listc_view');
    }
    else
    {
        echo "validation succeed then call save_view <br>";
        $this->load->view('save_view');
    };
}

function ml_check($str)
```

```
{
// プロファイラー
$this->output->enable_profiler(TRUE);

$emails = preg_split("/[\\s,]+/", $str);

foreach ($emails as $email)
{

if ($email == "" && $str == NULL)
{
    $this->form_validation->set_message('ml_check', 'email address を少なくとも1つ以上
        は登録して下さい。');
    return FALSE;
}

if ( $this->form_validation->valid_email($email) ||
    $this->form_validation->alpha_dash($email) ||
    $email == ""
    )
{
    // return TRUE;
}
}
else
{
    $this->form_validation->set_message('ml_check', 'email に許されない文字が入っていま
        す。');
    echo "次のメールアドレス ";
    echo $email;
    echo " は形式が違います。<br>";

    return FALSE;
}
}
return TRUE;
}
}
?>
```

ビューファイル listc_view.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ja" xml:lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>listc_view</title>
```

```
</head>

<body>

  <?php echo validation_errors(); ?>
  <br>
  <?php
  global $ml_users;

  if ($ml_users == NULL)
  {
  echo form_open('manage/validate');

  echo "修正したら、右下の [保存する] をクリックして下さい。";<br>;

  echo form_textarea('mlc', set_value('mlc'));

  echo form_submit('mysubmit', '保存する');
  echo "</form>";
  }
  else
  {
  echo form_open('manage/validate');

  echo "登録されている ML メンバーの email address は次の通りです。";<br>;
  echo "修正したら、右下の [保存する] をクリックして下さい。";<br>;

  foreach ($ml_users as $key => $value)
  {
  echo form_textarea('mlc', $ml_users[$key]['ml_content']);
  }

  //echo form_textarea('mlc', $ml_users['ml_content']);

  echo form_submit('mysubmit', '保存する');
  echo "</form>";
  }
  ?>

  <div align="right"><a href="https://dual.seinan-gu.ac.jp/ml/manage"> トップページに戻る
  </a></div>
  <br><br>
  VIEW の終わり
  </br>
  </body>
</html>
```

ビューファイル save_view.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ja" xml:lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>save_view</title>
</head>

<body>

<br>
<?php
$input_data = $this->input->post('mlc');
print ("次のデータを受け取りました<br />");

echo "<pre>";
echo "input_data in listc_view";
var_dump( $input_data );
echo "</pre>";

if ( ! write_file('/tmp/output', $input_data))
{
    echo 'ファイルに書き込めません';
}
else
{
    echo 'ファイルが書き込まれました!';
}

?>

<div align="right"><a href="https://dual.seinan-gu.ac.jp/ml/manage"> トップページに戻る
</a></div>

<br>
VIEW の終わり
</br>
</body>
</html>
```