Asynchoronous system for web application implementation support system

OHATA, Yuya / 大畑, 雄也

(発行年 / Year)
2007-03-24
(学位授与年月日 / Date of Granted)
2007-03-24
(学位名 / Degree Name)
修士(理学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)

Asynchronous System For Web Application Implementation Support System

Faculty of Computer and Information Sciences
05T0004
Yuya Ohata

Abstract

In ordinary Web application development, programs are implemented in the standalone environments. Afterwards, programs are up-loaded to the server, and behaviours are confirmed. However, it would advance the Web application developments easily by the environment which is able to confirm the behaviours directly and to edit programs on the server directly. Especially, when script languages such as PHP and Perl are used, the compilation work is not needed. Therefore, it is predictable that concise Web application development can be advanced compared with the development work that the program uploaded after it compiles like JAVA and C languages which require compilation and confirming the behaviours. In this thesis, it purposes to measure the response speed of the user request transmission to the arrival of the response from the server when Ajax is built into the Web application. Thereby, it aims to measure how to improve the response speed compared with the development techniques in the standalone. Users can proceed work without waiting for the response from the server by the asynchronous communication of data transmission with Ajax system. In old-fashioned Web application systems, the user was only able to wait until all processing was completed on the server when information data input by the user was up-loaded to the server. Moreover, it is necessary to receive all of the one page data of a browser in a past technique. By contrast, it leads to the reduction of the amount of the data transfer that sending and receiving by Ajax between the server and the use because of dealing with only necessary minimum data. Therefore, it is possible to guess that the response speed between the server and the user improves from the system where past Ajax is not implemented.

Hosei University Graduate School of Computer and Information Sciences Master's thesis

Contents

Chapter 1

Introduction

1.1 Motives

The Web application has come to be developed in a lot of developments as the telecommunication systems progression. Along with it, the communicated volume of data becomes huge, besides the data format treated becomes complex. The performance of software has hardly ever improved though the improvement of hardware is remarkable on the development site of the Web application though it is such a situation. Especially, little improvement is a current state in the improvement by the viewpoint of the client side that uses the Web application though the improvement by developer's viewpoint is repeated.

To solve such a current problem, the settlement plan is presented from the aspect of the client in this thesis. It aims to improve the transmission speed of the Web application from the aspect of the client.

1.2 Intentions

It proposes the following techniques as a method for the solution of the problem described in the foregoing paragraph.

In this thesis, it proposes the idea that communicates only a real-time, necessary data between the server and the clients, because the amount of the communication of data is reduced by using the HTTP

communication with the XMLHttpRequest object of Ajax(Asynchronous Javascript And XML). As a result, it will lead to the improvement of the response speed, because only the data of a necessary part will be communicated without exchanging the data for one page at each request from the user side. Moreover, the user comes never to be going to interrupt work by the time the response arrives because the data communication between the server and the client can be asynchronously transmitted.

This thesis consists of five chapters: In this thesis, it mentions about the technology that has been used by the time the theory named Ajax is established in Chapter 2, and it describes a useful point for the improvement of the response speed by using Ajax in Chapter 3, and the response speeds of the old-fashioned system development method are compared with Ajax systems response speed in Chapter 4. Then, the evaluation of the result in this thesis,and the problem in the future are summarized in Chapter 5.

Chapter 2

Related Works

Recently, it is remarkable that growth in hard respect of the performance of the telecommunication facilities and the server in the development of the Web application. However, it is a situation hardly seen on the improvement of the Web application excluding the growth of the linguistic level [1] . In the data communication of the Web application development, it is greatly useful for the growth of the developing method to try the reduction in volume of data in the future. Moreover, it is possible to expect the Web application implementation support system as a study support system for beginner's programmer who doesn't have the technology and knowledge that constructs the server environment.

The following technologies besides HTML including a static input form were researched by the time the theory named Ajax established it as a related research [2] .

- Nesting frameset
- Hidden iframe
- Remote scripting

2.1 Nesting frameset

In traditional HTML method including a static input form, the entire page of a browser was reloaded to sent the content of the input to the server. In this method, the user have to send and receive needless data besides it is necessary to receive the image again when the browser is reloaded.

To evade this problem, updating only a part of the page was adopted that was the technology called Frameset, which was a claptrap method of dividing the page as many fragments. In this method, how the window is divided is specified with $< Frameset >$ tag, and which file is allocated in the divided each frame is specified with $< Frame >$ tag. However, both developments and maintenances showed the tendency hardly to read code of the made program and to become difficult in this method.

Figure 2.1 is shown the behaviour when Frameset and Frame elements are reloaded.

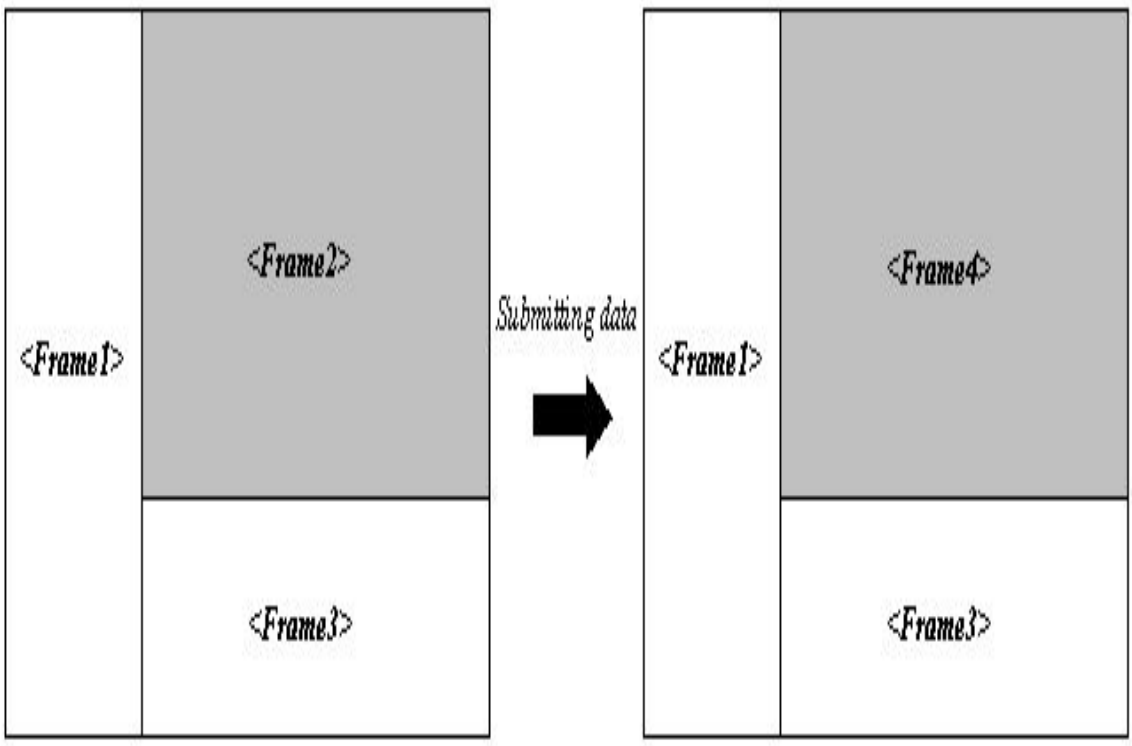


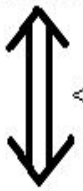
Figure 2.1. Reload behaviour of nesting frameset

2.2 Hidden iframe

The frame is made in the document when the IFRAME element is used, and another document can be buried. The iframe tag was used to attempt to change a part of the page which was the content of the display dynamically by Javascript and DHTML instead of using Frameset. However, this method could not have been accepted as the best method because of a burden of a settlement by the communication of the data between the code of the main and the iframe tags. Because a browser continues reading until communication data is completely transmitted. Therefore, it is necessary to wait for the client until the sending and receiving of data in the hidden irame part was completed.

Figure 2.2 is shown the appearance of modifying content of iframe by Javascript.

```
<HTML>
<HEAD>
<TITLE>Inline frame document</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function modify_contents() { ... }
-->
</SCRIPT>
</HEAD>
<BODY>
<IFRAME SRC="iframe_contents.html" HEIGHT="150" WIDTH="600">
unfavorable browser for iframe
</IFRAME>
</BODY>
</HTML>
```



<Modifying iframe contents>

Figure 2.2. Modifying contents of iframe by Javascript

2.3 Remote scripting

The technology named Remote scripting achieves the execution of a dynamic communication of data and the script on the user side and the server side at the same time in the code of the main. It is consisted by the combination of the standard generalized markup language such as HTML and the script languages such as VBScript and JavaScript. The content of the page was able to be rewritten dynamically by this technology which was the method of the description in Active Server Pages (asp file) executing Javascript on the user side was called. ASP can be basically performed only by combining Windows and Internet Information Services(IIS).

Figure 2.3 is shown the architecture of page dynamic generation with ASP on IIS.

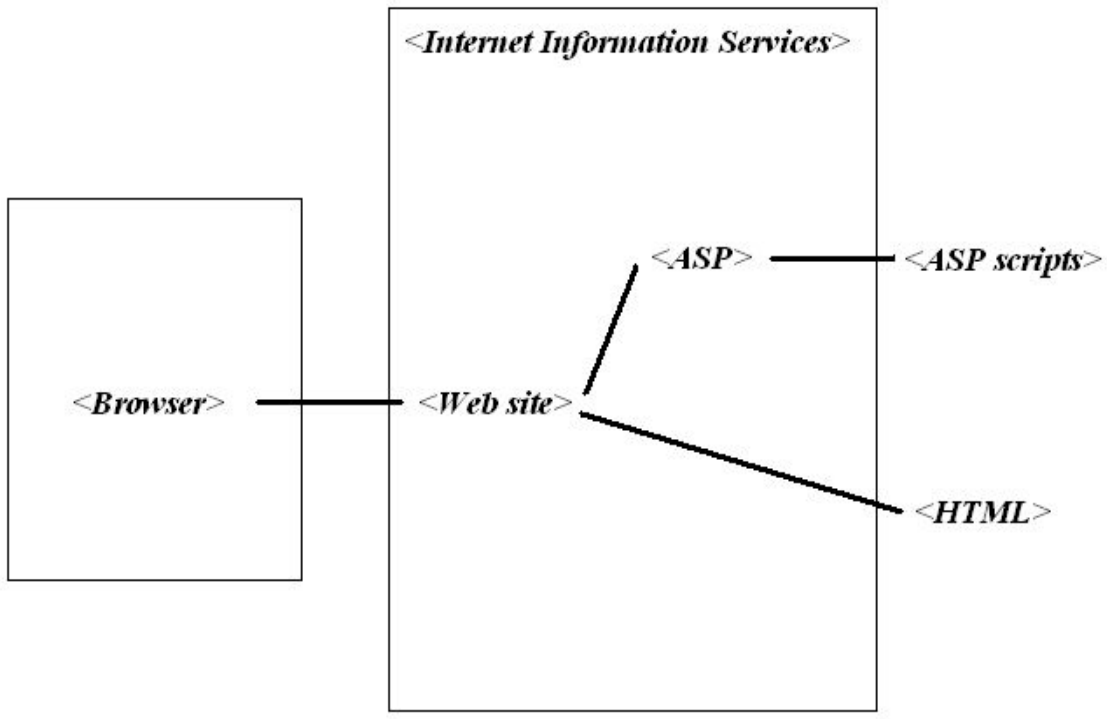


Figure 2.3. Architecture of page dynamic generation with ASP

This technology was not worth a much satisfaction to the developer of the Web application. This was due to the Active Server Pages that generated a dynamic content of the page, which was only called by IE(InternetExploer) and only requested easy GET correspondence on the user side to the server.

Chapter 3

Advantages and Features of Ajax

It is possible to construct the Web applications with the functions and the features by the built-in Ajax Web applications differed from the system constructed by a old-fashioned development method. In a old-fashioned Web application, the response was assumed to be a Web page after the request was transmitted to the server and the receipt page transition was generated. On the other hand, it becomes possible to achieve the production of a dynamic Web application not to accompany the page transition in Ajax. For instance, in real time displaying the retrieval result by applying Ajax to the Web retrieval becomes possible. It becomes possible to achieve it while the user inputs the key by the background though the retrieval was done after the user inputs the search condition so far. However, to develop the system that builds in Ajax is not to be able to develop more easily than a old-fashioned Web application is developed.

On the other hand, the fact that Ajax development embraces so many different technologies makes it a lot more useful. Here is a listing of the technologies that work together to make an Ajax Web application [3] :

- XML
- DOM
- CSS
- XMLHttpRequest
- Javascript

A lot of technologies have been invented by the time Ajax established as shown in the preceding chapter. In the technology that has been invented, there is the one named DHTML as a technology that looks like Ajax. Only a static HTML page had existed before DHTML invented. In a word, it was not possible to move it freely in the script after the characters and the images became to remain displayed on the page. This is DHTML that can freely operate data on HTML, and treats data on the page as a target. Various processing became possible with information as Document Object Model(DOM) but not treating the HTML data as the only mere text and the image. The part concerning such a display is executed by accessing CSS with JavaScript. As a result, the role of HTML, CSS, and JavaScript was clearly separated as follows.

- HTML:Showing the structure and the model
- CSS:Showing the method of the display
- Javascript:Showing the movement

The asynchronous communication of XMLHttpRequest transmission was added to such technologies is called Ajax.

The reduction method and the asynchronous communication of the sending and receiving data are treated as the especially concrete effect in this thesis. The items from which the following are enumerated is a content treated with this thesis.

- Javascript
- DOM
- XMLHttpRequest

3.1 Reduction of communication data volume

In a old-fashioned Web application, the request is transmitted from the client to the server. In that case, when the response is received from the server, it is necessary to reread the data of one browser page. On the other hand, only a part of a browser can be dynamically updated using Javascript and DOM in the Web application that builds in Ajax. It becomes possible by operating DOM of HTML by Javascript that limiting data are not reloaded and limiting the targeted data of the entire page. As a result, it becomes possible to reduce the amount of sending and receiving of data between the client and the server more than a old-fashioned development method [4] .

Only target data can be received to the client without receiving tedious image data etc. by reducing the volume of communicating data as an advantage in cases where Ajax is built in. In this method, a browser has no occasion to receive useless data, and the necessity to reload tedious data is lost.

Figure 3.1 is shown with regard to the difference of the correspondence procedure in the old-fashioned Web application and the correspondence procedure by Ajax Web application.

It is inefficient of reloading the browser because all the HTML data is received by the browser of the client in a old-fashioned method. On the other hand, it becomes possible to update dynamically only one part of the page because the XML data is received in Ajax. Furthermore, the communication data in Ajax does not necessitate the form of XML to receive, and can be received as text data.

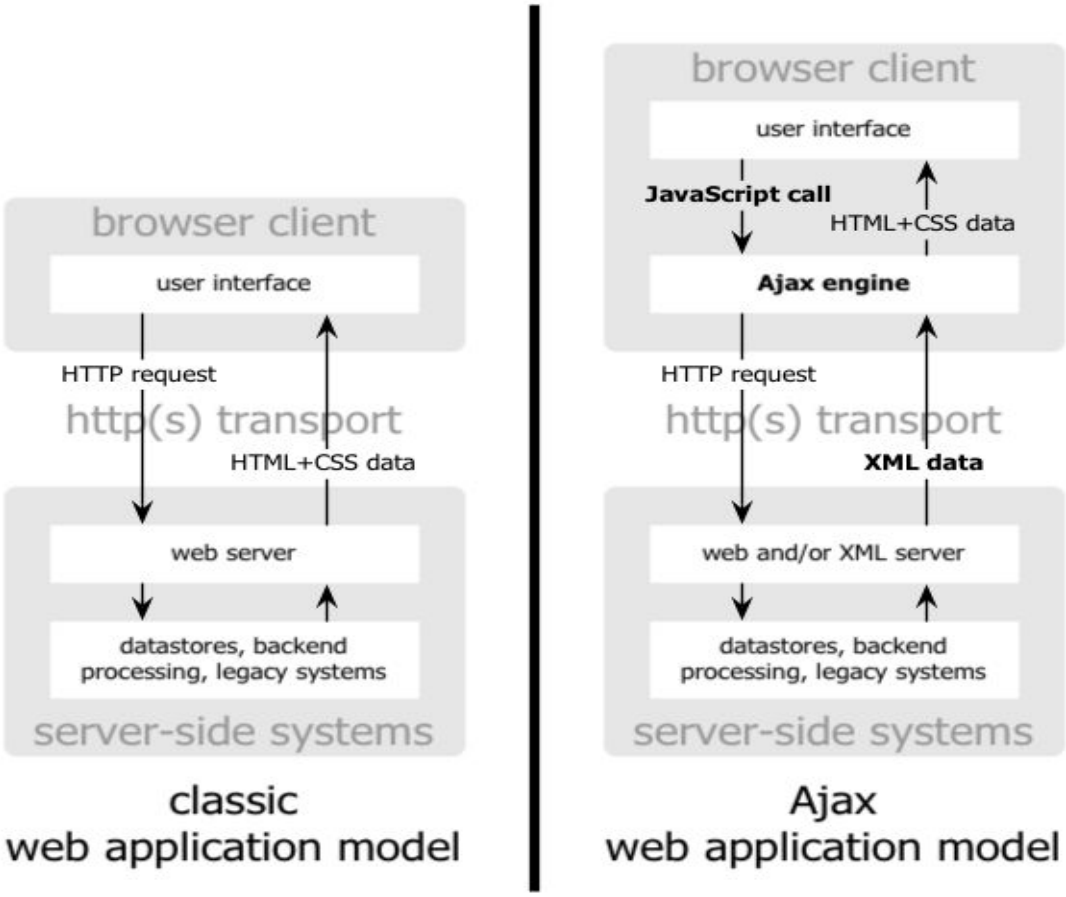


Figure 3.1. Difference between old-fashioned correspondence procedure and Ajax

3.2 Achievement of asynchronous communication

The asynchronous data communication by XMLHttpRequest is given as a different point from the old-fashioned Web application besides the reduction in the amount of the communication of the data previously described.

It was necessary to wait for a browser until the request was transmitted from a browser of the client to the server to communicate synchronously in a old-fashioned Web application, and to wait the reception of the response from the server ended completely. Contrary to this, it becomes possible for the user to work without waiting for the arrival of the response from the server to communicate asynchronously in the Web application that builds in Ajax. In a word, the page transition is not accompanied, and it becomes possible to achieve the production of a dynamic Web application. As a result, it can be said

that the response speed will improve more than old-fashioned methods.

Figure 3.2 is shown the behaviour of synchronous communications and the asynchronous communication. In synchronous communications, the request from the client reaches the server. The request data reaches the server, and the server sends processing result to the client. The page transition is done only after the browser of the client receives the data of one page at the end. On the contrary, it is understood that the client is processed waiting for neither the processing of the server nor the transmission of the response in the asynchronous communication. In consequence, the communication of data becomes possible without causing the time-out of a browser.

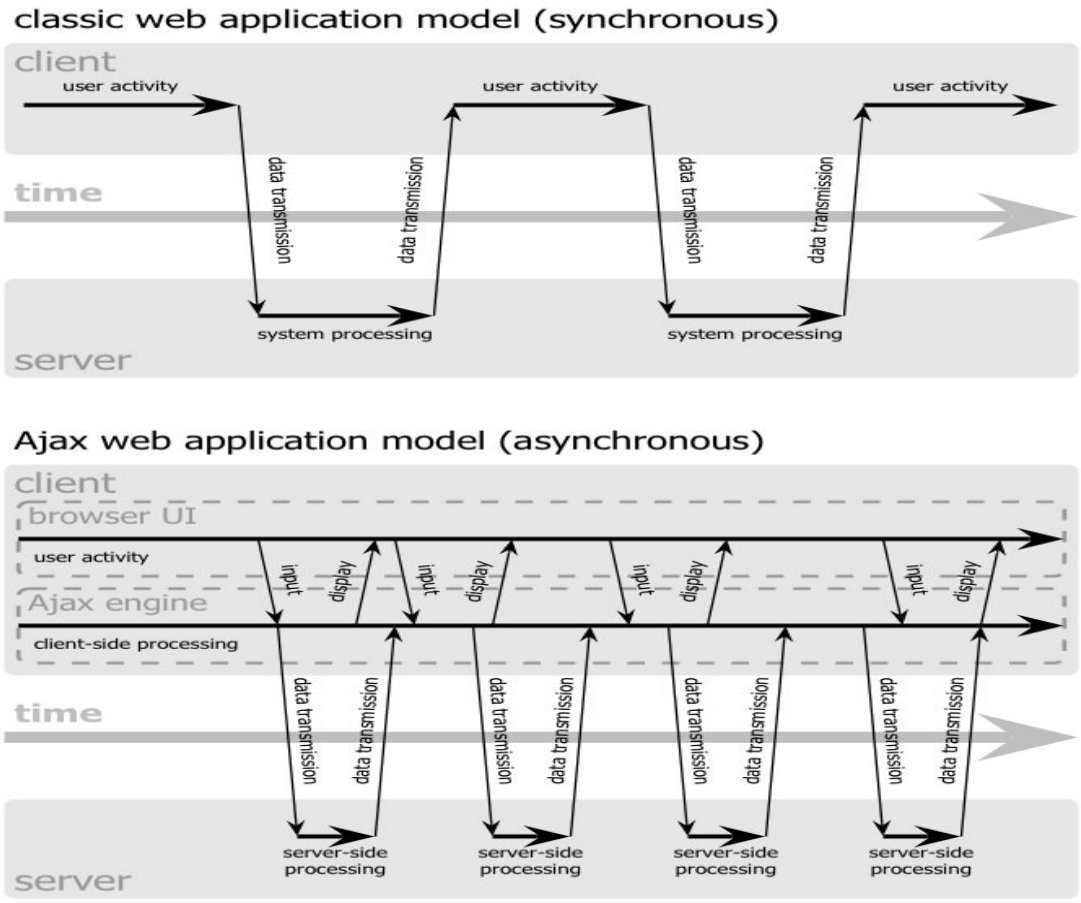


Figure 3.2. Synchronous and asynchronous communications

Chapter 4

Comparison with Old-fashioned System

The advantages to use Ajax and the problem of old-fashioned Web application has been described in previous chapters. In this chapter, the Web application that actually implements Ajax is developed, and how it is excellent is measured compared with a old-fashioned system.

4.1 Implements of Web application

First of all, the Web application that becomes an object is implemented. In this thesis, ”Web application that supports the development of the Web application” is implemented building in Ajax. Concretely, it becomes the application of the following system configurations.

To use this development support system, user ID and password are input in the login forms. Transition page is branched on this login page either the administrator user who manages this system or the developer user who uses this system. The user moves to the management page of the system, and can manage the developer user’s making, change, deletion, and the developer user’s source file when logging it in as the administrator user. Moreover, the user moves to the home page in the development workspace distributed by login ID when logging it in as the developer user. The developer user can confirm the operation of the source file and can manage the source file on the home page.

4.2 Cross-browser processing

In Ajax, there is a possibility that behaviour by each browser is different because Javascript codes are used. It becomes impossible for a lot of user to use the system where Ajax is implemented in this method though there is no problem if version and OS of a targeted browser are limited. This problem became definite in about 1998 years when Netscape Navigator and Internet Explorer were upgraded to versions 4. It was because each browser implemented DHTML that was the hot technology at that time with a different specifications. The technology of cross-browser to move it with only one code was created at that time, and the difference of the browsers was absorbed by this technology. In this thesis, to deal with this problem, the class that processed the cross-browser decided to be made.

The function named checkBrowser is implemented to judge whether Ajax can be operated on the browser of the user. This function judges whether Ajax can be operated on the browser. It returns true when the browser can operate Ajax code, and returns false when it can do. As a result, it is necessary to add processing when it is judged that Ajax can be operated. The following code is the excerpt parts of the source code file that processes cross-browser problem.

```
function checkBrowser(){
 var a,ua = navigator.userAgent;
 this.bw = {
   safari :
    ((a=ua.split('AppleWebKit/')[1]) ? a.split('.')[0] : 0) >= 124,
   konqueror :
    ((a=ua.split('Konqueror/')[1]) ? a.split(';')[0] : 0) >= 3.3,
   mozes :
    ((a=ua.split('Gecko/')[1]) ? a.split(' ')[0] : 0) >= 20011128,
   opera :
    (!!window.opera) && ((typeof XMLHttpRequest)=='function'),
   msie :
```

```
    (!!window.ActiveXObject) ? (!!createHttpRequest()) : false
 }
 return (
   this.bw.safari ||
   this.bw.konqueror ||
   this.bw.mozes ||
   this.bw.opera ||
   this.bw.msie
 )
}
```

In this function, the following browsers are judged for using this system. The message that it is not possible to correspond when a browser not recorded in following list uses this system is displayed.

- Higher version since 6.0 of Windows Internet Explorer
- Higher version since 3.3 of Konqueror
- Higher version since 1.24 of Safari, OmniWeb and Shiira
- Higher version since 2001/11/28 of Firefox, Netscape, Galeon, Epiphany, K-Meleon and Sylera
- Higher version since 8.0 of Opera

4.3 Comparison of Web applications

It is implemented that the Web application development support system with Ajax as shown in 4.2 . In this section, it is compared that the time taking to send and receive the data of the old-fashioned system to the time taking to send and receive the data of the system with Ajax.

The environment that confirms behaviours of Web application development support system with Ajax is as follows.

- Environment of client side: Windows XP, Pentium M 1.60GHz, 1.48GB RAM
- Environment of server side: CentOS 4.0, Xeon 2.4GHz, 2.0GB RAM

- Transmitted volume of data and measurement frequency: 10KB Text data, 10,000 trials

The following table is the results of measuring the time of sending and receiving the data in the both systems.

Table 4.1. Comparison of Web applications

	AVERAGE SPEED
OLD-FASHIONED SYSTEM	140msec
AJAX SYSTEM	2.2msec

It is understandable from the result that the time of the sending and receiving the data of the system implemented Ajax is about sixty times faster than the old-fashioned system's. The reason for this result is implementation of the system that is the technique for sending and receiving only the edited text data. As a result, the response speed has improved without sending and receiving useless data.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

In this thesis, the response speed of the Web application that built in Ajax is compared to the Web application constructed by a old-fashioned development method. It turns out that the response speed of the system implemented Ajax is more excellent than the Web application developed by a old-fashioned technique like as shown in the result in Table 4.1 . Moreover, it is holding promise that the system implemented Ajax becomes help that promotes programing technique of beginner programmer who does not maintain the technology that constructs environment of the server.

However, there are some problems to build Ajax into the Web application. Corresponding to a treated cross-browser is one of the problems in this thesis. It is necessary to create some means to bridge the gap of the specification between browsers. To deal with this problem, the particular class was made in this thesis. In general, limiting the utilizing environment might become a counter measure.

In addition, the cross-site scripting by client side scripts like Javascript codes is enumerated as other problems of Ajax Web applications. Furthermore, the client side script as Javascript is effective as the means to offer User Interface that the response to the user's operation is quicker than the server side program, and operativeness is higher. It is ordinary action for the Web application to process burying the input data under the HTML page. If a malicious script is included in input data, the Web application

will bury a malicious script under the HTML page. On the other hand, it is not possible to use it for the purpose of security though the client side script is profitable for user's convenience. For this reason, it is necessary to sanitize to the data sent to the server appropriately to deal with this problem. It is necessary to check the user input value on the server side program, and to acquire security.

5.2 Future Works

It can be said that the Web application implemented Ajax is very useful if these problems are solved. It is clear from the result of Table 4.1 in this thesis. It is because about sixty times the speed is measured with this thesis according to the response speed of the old-fashioned Web application.

Moreover, the attempt to combine Flash being able to expect similar behaviours with Ajax is investigated. It can be expected that such a lot of researches will develop in the future.

It should be pointed out that more and more systems implemented Ajax will be developed in the Web application development in the future.

Bibliography

[1] James Edwards,”AJAX and Screenreaders: When Can it Work?”,sitepoint,http://www.sitepoint.com,2006

[2] Matthew Eernisse,”Build Your Own AJAX Web Applications”,sitepoint,http://www.sitepoint.com,2006

[3] Lincoln D. Stein,”How To Set Up and Maintain a Web Site”,Addison Wesley,1997

[4] Toshiro Takahashi,”Ajax Approach”,SOFTBANK Creative Corp.,2006

Appendix A

Source codes

A.1 Administrator source codes

admin_complete.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //確認画面で戻るボタンが押された場合
  if(isset($_POST['return'])){
    header('Location: '.HOME_ADDRESS.'admin/admin_edit.php');
    exit;
  }

  //セッション入力値の取得
  $auth_name = $_SESSION['admin_name'];
  $auth_pass = $_SESSION['admin_pass'];

  //セッション入力値の破棄
  if(isset($_SESSION['admin_name']) && isset($_SESSION['admin_pass'])){
    unset($_SESSION['admin_name']);
    unset($_SESSION['admin_pass']);
  }

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //管理者更新クエリ
  $query = 'UPDATE
              auth_tab
            SET
              auth_name = \''.$auth_name.'\' ,
```

```
            auth_pass = \''.$auth_pass.'\' ,
            edit_date = \''.date('Y-m-d H:i:s').'\'
          WHERE admin_flg = '.ADMIN_FLG;

  if($db->query($query)){
    $cs = new CustomSmarty();
    $cs->assign('massage',' 管理者情報の変更が完了しました。');
    $cs->display('admin_complete.tpl');
  }
  else{
    echo $db->get_error();
  }

?>
```

admin_confirm.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //Smarty 拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //変更画面で戻るボタンが押された場合
  if(isset($_POST['return'])){
    header('Location: '.HOME_ADDRESS.'admin/admin_info.php');
    exit;
  }

  $error_msg = '';

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['admin_name']) && isset($_SESSION['admin_pass'])){
    $auth_name = $_SESSION['admin_name'];
    $auth_pass = $_SESSION['admin_pass'];
  }
  //POST された値のエスケープ
  if(isset($_POST['confirm'])){
    $auth_name = pg_escape_string(trim($_POST['admin_id']));
    $auth_name = htmlspecialchars($auth_name);
    $auth_pass = pg_escape_string(trim($_POST['admin_pass']));
    $auth_pass = htmlspecialchars($auth_pass);
  }

  //エラーチェック
  if($auth_name == ''){
    $error_msg .= "管理者 ID が未入力です。<br>";
  }
  if(preg_match('/[^0-9a-zA-Z]/',$auth_name)){
    $error_msg .= "管理者 ID は半角英数字で入力してください。<br>";
  }
  if(strlen($auth_name) > 15){
    $error_msg .= "管理者 ID は 15 文字以内で入力してください。<br>";
  }
  if($auth_pass == ''){
    $error_msg .= "パスワードが未入力です。<br>";
  }
  if(preg_match('/[^0-9a-zA-Z]/',$auth_pass)){
    $error_msg .= "パスワードは半角英数字で入力してください。<br>";
  }
```

```
  if(strlen($auth_pass) > 15){
    $error_msg .= "パスワードは15文字以内で入力してください。<br>";
  }
  //入力内容にエラーがある場合
  if($error_msg != ''){
    $cs = new CustomSmarty();
    $cs->assign('admin_name',$auth_name);
    $cs->assign('admin_pass',$auth_pass);
    $cs->assign('error',$error_msg);
    $cs->display('admin_edit.tpl');
  }
  else{
    $_SESSION['admin_name'] = $auth_name;
    $_SESSION['admin_pass'] = $auth_pass;
    $cs = new CustomSmarty();
    $cs->assign('admin_name',$auth_name);
    $cs->assign('admin_pass',$auth_pass);
    $cs->assign('massage','この内容で更新しても宜しいですか？');
    $cs->display('admin_confirm.tpl');
  }
?>
```

admin_edit.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['admin_name']) && isset($_SESSION['admin_pass'])){
    $cs = new CustomSmarty();
    $cs->assign('admin_name',$_SESSION['admin_name']);
    $cs->assign('admin_pass',$_SESSION['admin_pass']);
    $cs->display('admin_edit.tpl');
  }
  else{
    //データベース連携オブジェクト
    $db = new DatabaseBridge();

    //管理者情報取得クエリ
    $query = 'SELECT
                *
              FROM
                auth_tab
              WHERE
                admin_flg = \''.ADMIN_FLG.'\'';

    if($db->query($query)){
      $result = $db->get_result();
      if($assoc = pg_fetch_assoc($result)){
        $cs = new CustomSmarty();
        $cs->assign('admin_name',$assoc['auth_name']);
        $cs->assign('admin_pass',$assoc['auth_pass']);
        $cs->display('admin_edit.tpl');
      }
```

```
    }
    else{
      echo $db->get_error();
    }
  }
?>
```

admin_home.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  $cs = new CustomSmarty();
  $cs->display('admin_home.tpl');
?>
```

admin_info.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['admin_name']) && isset($_SESSION['admin_pass'])){
    unset($_SESSION['admin_name']);
    unset($_SESSION['admin_pass']);
  }

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //管理者情報取得クエリ
  $query = 'SELECT
              *
            FROM
              auth_tab
            WHERE
              admin_flg = \''.ADMIN_FLG.'\'';

  if($db->query($query)){
    $result = $db->get_result();
    if($assoc = pg_fetch_assoc($result)){
      $cs = new CustomSmarty();
      $cs->assign('admin_name',$assoc['auth_name']);
      $cs->assign('admin_pass',$assoc['auth_pass']);
      $cs->display('admin_info.tpl');
    }
  }
  else{
```

```
    echo $db->get_error();
  }
?>
```

member_complete.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty 拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');
  //開発者関連の操作クラス
  require_once('../libs/MemberHandle.class.php');

  //セッション開始
  session_start();

  //確認画面で戻るボタンが押された場合
  if(isset($_POST['return'])){
    //開発者新規登録の場合
    if(!isset($_SESSION['auth_key'])){
      header('Location: '.HOME_ADDRESS.'admin/member_regist.php');
      exit;
    }
    else{
      if(!isset($_SESSION['del_flg'])){
        header('Location: '.HOME_ADDRESS.'admin/member_edit.php');
        exit;
      }
      else{
        header('Location: '.HOME_ADDRESS.'admin/member_list.php');
        exit;
      }
    }
  }

  //セッション入力値の操作
  if(isset($_SESSION['auth_name'])){
    $auth_name = $_SESSION['auth_name'];
    unset($_SESSION['auth_name']);
  }
  if(isset($_SESSION['auth_pass'])){
    $auth_pass = $_SESSION['auth_pass'];
    unset($_SESSION['auth_pass']);
  }
  if(isset($_SESSION['auth_key'])){
    $auth_key = $_SESSION['auth_key'];
    unset($_SESSION['auth_key']);
  }
  if(isset($_SESSION['del_flg'])){
    $del_flg = $_SESSION['del_flg'];
    unset($_SESSION['del_flg']);
  }

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //開発者新規登録の場合
  if(!isset($auth_key)){

    //開発者新規登録クエリ
    $query = 'INSERT INTO
```

```
                auth_tab(
                  auth_name ,
                  auth_pass ,
                  admin_flg
                )
              VALUES(
                \''.$auth_name.'\' ,
                \''.$auth_pass.'\' ,
                '.DEV_FLG.'
              )';

    if($db->query($query)){
      $cs = new CustomSmarty();
      $cs->assign('massage',' 開発者の新規登録が完了しました。');
      $cs->assign('title',' 管理画面：開発者新規登録完了');
      $cs->display('member_complete.tpl');
    }
    else{
      echo $db->get_error();
      exit;
    }

    //開発者番号の nextval を取得する
    $query = 'SELECT CURRVAL(\'auth_tab_auth_key_seq\') AS auth_key';
    $db->query($query);
    $result = $db->get_result();
    $assoc = pg_fetch_assoc($result);

    //開発者関連操作オブジェクト
    $mh = new MemberHandle($assoc['auth_key']);

    //開発環境のイニシャライズ
    if(!$mh->initialize()){
      echo $mh->get_error();
      exit;
    }
  }
  else{
    //開発者情報更新の場合
    if(!isset($del_flg)){
      //開発者更新登録クエリ
      $query =
              'UPDATE
                auth_tab
              SET
                auth_name = \''.$auth_name.'\' ,
                auth_pass = \''.$auth_pass.'\' ,
                edit_date = \''.date('Y-m-d H:i:s').'\'
              WHERE auth_key = \''.$auth_key.'\'';

      if($db->query($query)){
        $cs = new CustomSmarty();
        $cs->assign('massage',' 開発者情報の変更が完了しました。');
        $cs->assign('title',' 管理画面：開発者情報変更完了');
        $cs->display('member_complete.tpl');
      }
      else{
        echo $db->get_error();
      }
    }
    else{

      //開発者関連操作オブジェクト
      $mh = new MemberHandle($auth_key);

      //開発環境の削除
```

```
      if(!$mh->terminate()){
        echo $mh->get_error();
        exit;
      }

      //開発者削除の場合
      $query =
             'DELETE FROM
               auth_tab
             WHERE auth_key = \''.$auth_key.'\'';

      if($db->query($query)){
        $cs = new CustomSmarty();
        $cs->assign('massage','開発者の削除が完了しました。');
        $cs->assign('title','管理画面：開発者削除完了');
        $cs->display('member_complete.tpl');
      }
      else{
        echo $db->get_error();
      }
    }
  }
?>
```

member_confirm.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //戻るボタンが押された場合
  if(isset($_POST['return'])){
    header('Location: '.HOME_ADDRESS.'admin/member_list.php');
    exit;
  }

  $error_msg = '';

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['auth_name']) && isset($_SESSION['auth_pass'])){
    $auth_name = $_SESSION['auth_name'];
    $auth_pass = $_SESSION['auth_pass'];
  }
  //POSTされた値のエスケープ
  if(isset($_POST['confirm'])){
    $auth_name = pg_escape_string(trim($_POST['auth_name']));
    $auth_name = htmlspecialchars($auth_name);
    $auth_pass = pg_escape_string(trim($_POST['auth_pass']));
    $auth_pass = htmlspecialchars($auth_pass);
  }

  //エラーチェック
  if($auth_name == ''){
    $error_msg .= "開発者IDが未入力です。<br>";
  }
  if(preg_match('/[^0-9a-zA-Z]/',$auth_name)){
```

```
  $error_msg .= "開発者IDは半角英数字で入力してください。<br>";
}
if(preg_match('/[0-9a-zA-Z]/',$auth_name) && strlen($auth_name) > 15){
  $error_msg .= "開発者IDは15文字以内で入力してください。<br>";
}
//同一IDの登録禁止
//データベース連携オブジェクト
$db = new DatabaseBridge();
//管理者情報取得クエリ
$query = 'SELECT
            count(auth_key) AS same_count
          FROM
            auth_tab
          WHERE
            auth_name = \''.$auth_name.'\' AND
            auth_key != '.$_SESSION['auth_key'].' AND
            admin_flg = \''.DEV_FLG.'\'';
if($db->query($query)){
  $result = $db->get_result();
  $assoc = pg_fetch_assoc($result);
  if($assoc['same_count'] != 0){
    $error_msg .= "既に同じ開発者IDが登録されています。<br>";
  }
}
else{
  echo $db->get_error();
}

if($auth_pass == ''){
  $error_msg .= "パスワードが未入力です。<br>";
}
if(preg_match('/[^0-9a-zA-Z]/',$auth_pass)){
  $error_msg .= "パスワードは半角英数字で入力してください。<br>";
}
if(preg_match('/[0-9a-zA-Z]/',$auth_pass) && strlen($auth_pass) > 15){
  $error_msg .= "パスワードは15文字以内で入力してください。<br>";
}

if($error_msg != ''){
  //開発者新規登録の場合
  if(!isset($_SESSION['auth_key'])){
    $cs = new CustomSmarty();
    $cs->assign('auth_name',$auth_name);
    $cs->assign('auth_pass',$auth_pass);
    $cs->assign('error',$error_msg);
    $cs->display('member_regist.tpl');
  }
  else{
    $cs = new CustomSmarty();
    $cs->assign('auth_name',$auth_name);
    $cs->assign('auth_pass',$auth_pass);
    $cs->assign('error',$error_msg);
    $cs->display('member_edit.tpl');
  }
}
else{
  //セッションに入力値の保存
  $_SESSION['auth_name'] = $auth_name;
  $_SESSION['auth_pass'] = $auth_pass;

  $cs = new CustomSmarty();
  $cs->assign('auth_name',$auth_name);
  $cs->assign('auth_pass',$auth_pass);
  //開発者新規登録の場合
```

```
    if(!isset($_SESSION['auth_key'])){
      $cs->assign('massage',' この内容で登録しても宜しいですか？');
      $cs->assign('title',' 管理画面：開発者新規登録確認');
      $cs->assign('submit_name','regist');
      $cs->assign('submit_value',' 登録');
    }
    else{
      $cs->assign('massage',' この内容で更新しても宜しいですか？');
      $cs->assign('title',' 管理画面：開発者情報変更確認');
      $cs->assign('submit_name','edit');
      $cs->assign('submit_value',' 更新');
    }
    $cs->display('member_confirm.tpl');
  }
?>
```

member_delete.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty 拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //セッション情報を保存
  if(isset($_POST['auth_key'])){
    $_SESSION['auth_key'] = $_POST['auth_key'];
  }
  if(isset($_POST['delete'])){
    $_SESSION['del_flg'] = TRUE;
  }

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //管理者情報取得クエリ
  $query = 'SELECT
              *
            FROM
              auth_tab
            WHERE
              auth_key = \''.$_SESSION['auth_key'].'\'
            AND admin_flg = \''.DEV_FLG.'\'';

  if($db->query($query)){
    $result = $db->get_result();
    if($assoc = pg_fetch_assoc($result)){
      $cs = new CustomSmarty();
      $cs->assign('auth_name',$assoc['auth_name']);
      $cs->assign('auth_pass',$assoc['auth_pass']);
      $cs->assign('massage',' この開発者を削除しても宜しいですか？');
      $cs->assign('title',' 管理画面：開発者削除確認');
      $cs->assign('submit_name','delete');
      $cs->assign('submit_value',' 削除');
      $cs->display('member_confirm.tpl');
    }
  }
```

```
  else{
    echo $db->get_error();
  }
?>
```

member_edit.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['auth_name']) && isset($_SESSION['auth_pass'])){
    $cs = new CustomSmarty();
    $cs->assign('auth_name',$_SESSION['auth_name']);
    $cs->assign('auth_pass',$_SESSION['auth_pass']);
    $cs->display('member_edit.tpl');
  }
  else{
    if(isset($_POST['auth_key'])){
      $_SESSION['auth_key'] = $_POST['auth_key'];
    }

    //データベース連携オブジェクト
    $db = new DatabaseBridge();

    //管理者情報取得クエリ
    $query = 'SELECT
                *
              FROM
                auth_tab
              WHERE
                auth_key = \''.$_SESSION['auth_key'].'\'
              AND admin_flg = \''.DEV_FLG.'\'';

    if($db->query($query)){
      $result = $db->get_result();
      if($assoc = pg_fetch_assoc($result)){
        $cs = new CustomSmarty();
        $cs->assign('auth_name',$assoc['auth_name']);
        $cs->assign('auth_pass',$assoc['auth_pass']);
        $cs->display('member_edit.tpl');
      }
    }
    else{
      echo $db->get_error();
    }
  }
?>
```

member_list.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
```

```
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //セッション入力値の破棄
  if(isset($_SESSION['auth_name']) && isset($_SESSION['auth_pass'])){
    unset($_SESSION['auth_name']);
    unset($_SESSION['auth_pass']);
  }
  if(isset($_SESSION['auth_key'])){
    unset($_SESSION['auth_key']);
  }
  if(isset($_SESSION['del_flg'])){
    unset($_SESSION['del_flg']);
  }

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //管理者情報取得クエリ
  $query = 'SELECT
              *
            FROM
              auth_tab
            WHERE
              admin_flg = \''.DEV_FLG.'\'
            ORDER BY auth_key';

  if($db->query($query)){
    $result = $db->get_result();
    $member = array();
    while($assoc = pg_fetch_assoc($result)){
      array_push($member,$assoc);
    }
    $cs = new CustomSmarty();
    $cs->assign('members',$member);
    $cs->display('member_list.tpl');
  }
  else{
    echo $db->get_error();
  }
?>
```

member_regist.php

```
<?php
  //ログインチェック
  require_once('../libs/admin_check.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  $cs = new CustomSmarty();

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['auth_name']) && isset($_SESSION['auth_pass'])){
    $cs->assign('auth_name',$_SESSION['auth_name']);
    $cs->assign('auth_pass',$_SESSION['auth_pass']);
```

```
  }
  $cs->display('member_regist.tpl');
?>
```

A.2 Library source codes

auth.php

```
<?php
  //データベース連携クラスファイル
  require_once('libs/DatabaseBridge.class.php');
  //Smarty 拡張クラス
  require_once('libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('libs/config.php');

  //セッション開始
  session_start();

  //POST された値のエスケープ
  $auth_name = pg_escape_string(trim($_POST['login_id']));
  $auth_pass = pg_escape_string(trim($_POST['login_password']));

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //ログイン認証クエリ
  $query = 'SELECT
              *
            FROM
              auth_tab
            WHERE
              auth_name = \''.$auth_name.'\'
            AND auth_pass = \''.$auth_pass.'\'';

  if($db->query($query)){
    $result = $db->get_result();
    if($assoc = pg_fetch_assoc($result)){
      //認証成功時
      if($assoc['admin_flg'] == ADMIN_FLG){
        //管理者権限
        $_SESSION['auth'] = ADMIN_FLG;
        //管理者画面へ遷移
        header('Location: '.HOME_ADDRESS.'admin/admin_home.php');
      }
      else{
        //開発者権限
        $_SESSION['auth'] = DEV_FLG;
        //開発者 ID
        $_SESSION['auth_key'] = $assoc['auth_key'];
        //開発者画面へ遷移
        header('Location: '.HOME_ADDRESS.'workspaces/dev_home.php');
      }
    }
    else{
      //認証失敗時
      $error_msg = "存在しない ID かパスワードが異なります。";

      //ログイン情報の削除
      foreach($_SESSION AS $key => $value){
        unset($_SESSION[$key]);
      }

      $cs = new CustomSmarty();
      $cs->assign('error',$error_msg);
      $cs->display('login.tpl');
    }
  }
```

```
  else{
    echo $db->get_error();
  }
?>
```

login.php

```
<?php
  //Smarty拡張クラス
  require_once('libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('libs/config.php');

  //セッション開始
  session_start();

  //ログイン情報の削除
  foreach($_SESSION AS $key => $value){
    unset($_SESSION[$key]);
  }

  $cs = new CustomSmarty();
  $cs->display('login.tpl');
?>
```

utils.php

```
<?php
  /**
  * name:get_files
  * detail:ディレクトリ直下のファイル属性を取得する
  * param:$dir_name(String) ディレクトリの絶対パス
  * return:$files(array) ディレクトリ直下のファイル属性の配列
  * creation date:2006/12/17
  * last modified:2006/12/17
  */
  function get_files($dir_name){
    //ディレクトリハンドルから取得するエントリ
    $entries = array();
    //ファイル名の配列
    $file_names = array();
    //ファイル属性の配列
    $files = array();
    //読み込んだファイル名
    $file_name = '';

    //開発者ホームディレクトリへのリソース取得
    if($resource = opendir($dir_name)){
      while(($file_name = readdir($resource)) !== false){
        array_push($entries,$file_name);
      }
      //リソースを開放
      closedir($resource);

      for($loop = 0 ; $loop < count($entries) ; $loop++){
        //カレントディレクトリと上位ディレクトリを除く
        if($entries[$loop] != '.' && $entries[$loop] != '..'){
          array_push($file_names,$entries[$loop]);
        }
      }
      //ファイル属性の取得
      for($loop = 0 ; $loop < count($file_names) ; $loop++){
        $files[$loop]['name'] = $file_names[$loop];
        $files[$loop]['type'] = filetype($dir_name.'/'.$file_names[$loop]);
```

```
        if((filesize($dir_name.’/’.$file_names[$loop])/1000) < 1){
          $files[$loop][’size’] =
            ceil(filesize($dir_name.’/’.$file_names[$loop])/1000);
        }
        else{
          $files[$loop][’size’] =
            round(filesize($dir_name.’/’.$file_names[$loop])/1000 , 1);
        }
      }
    }

    return $files;
  }

  /**
  * name:get_microtime
  * detail:microtime関数によって取得した"msec sec"の文字列を数値に変換する
  * param:$sec+$msec(float) ディレクトリの絶対パス
  * return:$files(array) Unix epoch (1970年1月1日 00:00:00)からの通算
秒
  * creation date:2006/12/20
  * last modified:2006/12/20
  */
  function get_microtime(){
    list($msec, $sec) = explode(" ",microtime());
    return ((float)$sec + (float)$msec);
  }
?>
```

admin_check.php

```
<?php
  //コンフィグファイル
  require_once(’config.php’);

  //セッション開始
  session_start();

  if($_SESSION[’auth’] != ADMIN_FLG){
    header(’Location: ’.HOME_ADDRESS.’login.php’);
  }
?>
```

ajax.js

```
 //動作可能なブラウザ判定
 function checkBrowser(){
  var a,ua = navigator.userAgent;
  this.bw = {
    safari    : ((a=ua.split(’AppleWebKit/’)[1])?a.split(’.’)[0]:0)>=124 ,
    konqueror : ((a=ua.split(’Konqueror/’)[1])?a.split(’;’)[0]:0)>=3.3 ,
    mozes     : ((a=ua.split(’Gecko/’)[1])?a.split(’ ’)[0]:0) >= 20011128 ,
    opera     : (!!window.opera) && ((typeof XMLHttpRequest)==’function’) ,
    msie      : (!!window.ActiveXObject)?(!!createHttpRequest()) : false
  }
  return(
    this.bw.safari ||
    this.bw.konqueror ||
    this.bw.mozes ||
    this.bw.opera ||
    this.bw.msie)
 }

 //XMLHttpRequestオブジェクト生成
 function createHttpRequest(){
```

```
 if(window.XMLHttpRequest){
   //Win Mac Linux m1,f1,o8 Mac s1 Linux k3 & Win e7用
  return new XMLHttpRequest() ;
 }
 else if(window.ActiveXObject){
   //Win e4,e5,e6用
  try{
   return new ActiveXObject('Msxml2.XMLHTTP') ;
  }
  catch(e){
   try{
    return new ActiveXObject('Microsoft.XMLHTTP') ;
   }
   catch(e2){
    return null ;
   }
  }
 }
 else{
  return null ;
 }
}

//送受信関数
function sendRequest(callback,method,url,async,sload,user,password){
 //XMLHttpRequestオブジェクト生成
 var oj = createHttpRequest();
 if(oj == null){
  return null;
 }

 //強制ロードの設定
 var sload = (!!sendRequest.arguments[5]) ? sload : false;
 if(sload || method.toUpperCase() == 'GET'){
  url += '?';
 }
 if(sload){
  url=url+'t='+(new Date()).getTime();
 }

 //ブラウザ判定
 var bwoj = new checkBrowser();
 var opera   = bwoj.bw.opera;
 var safari   = bwoj.bw.safari;
 var konqueror = bwoj.bw.konqueror;
 var mozes   = bwoj.bw.mozes;

 //callbackを分解
 //{onload:xxxx,onbeforsetheader:xxx}
 if(typeof callback=='object'){
  var callback_onload = callback.onload;
  var callback_onbeforsetheader = callback.onbeforsetheader;
 }
 else{
  var callback_onload = callback;
  var callback_onbeforsetheader = null;
 }

 //受信処理
 if(opera || safari || mozes){
  oj.onload = function(){
               callback_onload(oj);
              }
 }
 else{
  oj.onreadystatechange = function(){
   if ( oj.readyState == 4 ){
```

```
   callback_onload(oj);
  }
 }
}

//URLエンコード
var data = '&file_number='+document.getElementById('file_number').value+
           '&source='+encodeURL(document.getElementById('source').value);
if(method.toUpperCase() == 'GET'){
 url += data;
}

//open メソッド
oj.open(method,url,async,user,password);


//コールバック
if(!!callback_onbeforsetheader){
 callback_onbeforsetheader(oj);
}

//デフォルトヘッダ
setEncHeader(oj);

//send メソッド
oj.send(data);

//URIエンコードヘッダセット
function setEncHeader(oj){
 //ヘッダapplication/x-www-form-urlencodedセット
 var contentTypeUrlenc = 'application/x-www-form-urlencoded; charset=UTF-8';
 if(!window.opera){
  oj.setRequestHeader('Content-Type',contentTypeUrlenc);
 }
 else{
  if((typeof oj.setRequestHeader) == 'function'){
   oj.setRequestHeader('Content-Type',contentTypeUrlenc);
  }
 }
 return oj
}

//URLエンコード (UTF-8)
function encodeURL(str) {
  var character = '';
  var unicode   = '';
  var string    = '';
  var i         = 0;

  for (i = 0; i < str.length; i++) {
    character = str.charAt(i);
    unicode = str.charCodeAt(i);

    if(character == ' '){
      string += '+';
    }
    else{
      if(unicode == 0x2a ||
         unicode == 0x2d ||
         unicode == 0x2e ||
         unicode == 0x5f ||
         ((unicode >= 0x30) && (unicode <= 0x39)) ||
         ((unicode >= 0x41) && (unicode <= 0x5a)) ||
         ((unicode >= 0x61) && (unicode <= 0x7a))
        ){
        string = string + character;
```

```
        }
        else{
          if((unicode >= 0x0) && (unicode <= 0x7f)){
            character = '0' + unicode.toString(16);
            string += '%' + character.substr(character.length - 2);
          }
          else if(unicode > 0x1fffff){
            string += '%' + (oxf0 + ((unicode & 0x1c0000) >> 18)).toString(16)
            string += '%' + (0x80 + ((unicode & 0x3f000) >> 12)).toString(16);
            string += '%' + (0x80 + ((unicode & 0xfc0) >> 6)).toString(16);
            string += '%' + (0x80 + (unicode & 0x3f)).toString(16);
          }
          else if(unicode > 0x7ff){
            string += '%' + (0xe0 + ((unicode & 0xf000) >> 12)).toString(16);
            string += '%' + (0x80 + ((unicode & 0xfc0) >> 6)).toString(16);
            string += '%' + (0x80 + (unicode & 0x3f)).toString(16);
          }
          else{
            string += '%' + (0xc0 + ((unicode & 0x7c0) >> 6)).toString(16);
            string += '%' + (0x80 + (unicode & 0x3f)).toString(16);
          }
        }
      }
    }
    return string;
  }

  return oj
 }
```

config.php

```
<?php
  //システムの HOME ディレクトリ
  define('DEV_HOME','/usr/local/develop/');
  //Apache の DocumentRoot
  define('DOC_ROOT','/usr/local/');
  //システムワークスペース
  define('WORK_SPACES','/usr/local/develop/workspaces/');
  //システムのホームアドレス
  define('HOME_ADDRESS','http://localhost/develop/');
  //管理者権限
  define('ADMIN_FLG',1);
  //開発者権限
  define('DEV_FLG',2);
  //DB 接続情報
  define('DB_CONFIG','host=localhost port=5432 dbname=dev user=admin password=a
?>
```

CustomSmarty.class.php

```
<?php
 //include Smarty
 require_once('Smarty/libs/Smarty.class.php');
 //include config file
 require_once('config.php');

 /**
 * Name:CustomSmarty
 * SuperClass:Smarty
 * creation date:2006/11/13
 * last modified:2006/11/13
 */
 class CustomSmarty extends Smarty{

   /**
```

```
    * name:__construct()
    * detail:コンストラクタ
    * creation date:2006/11/13
    * last modified:2006/11/13
    */
    public function __construct(){
      $this->smarty();
      $this->template_dir = DEV_HOME.'templates';
      $this->compile_dir = DEV_HOME.'templates_c';
    }

  }
?>
```

DatabaseBridge.class.class.php dev_check.php

```
<?php
  //コンフィグファイル
  require_once('config.php');

  //セッション開始
  session_start();

  if($_SESSION['auth'] != DEV_FLG && !isset($_SESSION['auth_key'])){
    header('Location: '.HOME_ADDRESS.'login.php');
  }
?>
```

MemberHandle.class.php

```
<?php
  //include config file
  require_once('config.php');

  /**
  * name:MemberHandle
  * detail:開発者関連の操作クラス
  * creation date:2006/12/15
  * last modified:2006/12/15
  */
  class MemberHandle{
    //開発者番号
    var $developer_number;
    //エラー内容
    var $error_detail;

    /**
    * name:__construct
    * detail:コンストラクタ
    * param:$dev_num(integer) 開発者番号
    * creation date:2006/12/15
    * last modified:2006/12/15
    */
    public function __construct($dev_num){
      $this->developer_number = $dev_num;
    }

    /**
    * name:initialize
    * detail:開発環境を作成する
    * return:true 作成成功時,false 作成失敗時
    * creation date:2006/12/15
    * last modified:2006/12/15
    */
    public function initialize(){
      //開発者ホームディレクトリを作成する
```

```
  if(!mkdir(WORK_SPACES.$this->developer_number,0777)){
    //失敗時にはエラー内容を取得する
    $this->error_detail = 'ディレクトリ作成失敗'.WORK_SPACES.$this->developer_n
    return false;
  }
  return true;
}

/**
* name:terminate
* detail:開発環境を削除する
* return:true 削除成功時,false 削除失敗時
* creation date:2006/12/15
* last modified:2006/12/15
*/
public function terminate(){
  //開発者ホームディレクトリへのリソース
  $resource = '';
  //開発者ホームディレクトリに含まれるファイル名
  $file_name = '';
  //開発者ホームディレクトリに含まれるファイルの配列
  $file_names = array();
  //ディレクトリ名取得
  $dir_name = WORK_SPACES.$this->developer_number;

  //開発者ホームディレクトリへのリソース取得
  if($resource = opendir($dir_name)){
    while(($file_name = readdir($resource)) !== false){
      array_push($file_names,$file_name);
    }
    for($loop = 0 ; $loop < count($file_names) ; $loop++){
      if(is_file($dir_name.'/'.$file_names[$loop])){
        if(!unlink($dir_name.'/'.$file_names[$loop])){
          //ファイル削除失敗時
          $this->error_detail =
          'ファイル削除失敗\n'.
          'ファイル名:'.$dir_name.'/'.$file_names[$loop].'\n'.
          'ファイルグループID:'.filegroup($dir_name.'/'.$file_names[$loop]).'\
          'ファイル所有者ID:'.fileowner($dir_name.'/'.$file_names[$loop]).'\n
          'ファイル許可属性:'.fileperms($dir_name.'/'.$file_names[$loop]).'\n'
          'ファイルタイプ:'.filetype($dir_name.'/'.$file_names[$loop]).'\n';
          closedir($resource);
          return false;
        }
      }
    }

    //ディレクトリ削除のためリソースを開放
    closedir($resource);

    if(!rmdir($dir_name)){
      //ディレクトリ削除失敗時
      $this->error_detail =
      'ディレクトリ削除失敗\n'.
      'ディレクトリ名:'.$dir_name.'\n'.
      'ディレクトリID:'.filegroup($dir_name).'\n'.
      'ディレクトリ所有者ID:'.fileowner($dir_name).'\n'.
      'ディレクトリ許可属性:'.fileperms($dir_name).'\n'.
      'ディレクトリタイプ:'.filetype($dir_name).'\n';
      return false;
    }
    return true;
```

```
      }
      else{
        //失敗時にはエラー内容を取得する
        $this->error_detail = 'ディレクトリへのリソース取得失敗';
        return false;
      }
    }

    /**
    * name:get_error
    * detail:環境作成時のエラー内容を取得する
    * return:error_detail(String) エラー内容
    * creation date:2006/12/15
    * last modified:2006/12/15
    */
    public function get_error(){
      return $this->error_detail;
    }

  }
?>
```

A.3 Developer source codes

dev_complete.php

```
<?php
  //ログインチェック
  require_once('../libs/dev_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //確認画面で戻るボタンが押された場合
  if(isset($_POST['return'])){
    header('Location: '.HOME_ADDRESS.'workspaces/dev_edit.php');
    exit;
  }

  //セッション入力値の操作
  if(isset($_SESSION['auth_name'])){
    $auth_name = $_SESSION['auth_name'];
    unset($_SESSION['auth_name']);
  }
  if(isset($_SESSION['auth_pass'])){
    $auth_pass = $_SESSION['auth_pass'];
    unset($_SESSION['auth_pass']);
  }

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //開発者情報更新の場合
  if(!isset($del_flg)){
    //開発者更新登録クエリ
    $query =
            'UPDATE
              auth_tab
            SET
              auth_name = \''.$auth_name.'\' ,
              auth_pass = \''.$auth_pass.'\' ,
              edit_date = \''.date('Y-m-d H:i:s').'\'
            WHERE auth_key = \''.$_SESSION['auth_key'].'\'';

    if($db->query($query)){
      $cs = new CustomSmarty();
      $cs->assign('massage','開発者情報の変更が完了しました。');
      $cs->display('dev_complete.tpl');
    }
    else{
      echo $db->get_error();
    }
  }
?>
```

dev_confirm.php

```
<?php
  //ログインチェック
  require_once('../libs/dev_check.php');
  //データベース連携クラスファイル
```

```
require_once('../libs/DatabaseBridge.class.php');
//Smarty拡張クラス
require_once('../libs/CustomSmarty.class.php');
//コンフィグファイル
require_once('../libs/config.php');

//セッション開始
session_start();

//戻るボタンが押された場合
if(isset($_POST['return'])){
  header('Location: '.HOME_ADDRESS.'workspaces/dev_info.php');
  exit;
}

$error_msg = '';

//セッションに入力値が保存されている場合
if(isset($_SESSION['auth_name']) && isset($_SESSION['auth_pass'])){
  $auth_name = $_SESSION['auth_name'];
  $auth_pass = $_SESSION['auth_pass'];
}
//POSTされた値のエスケープ
if(isset($_POST['confirm'])){
  $auth_name = pg_escape_string(trim($_POST['auth_name']));
  $auth_name = htmlspecialchars($auth_name);
  $auth_pass = pg_escape_string(trim($_POST['auth_pass']));
  $auth_pass = htmlspecialchars($auth_pass);
}

//エラーチェック
if($auth_name == ''){
  $error_msg .= "開発者IDが未入力です。<br>";
}
if(preg_match('/[^0-9a-zA-Z]/',$auth_name)){
  $error_msg .= "開発者IDは半角英数字で入力してください。<br>";
}
if(preg_match('/[0-9a-zA-Z]/',$auth_name) && strlen($auth_name) > 15){
  $error_msg .= "開発者IDは15文字以内で入力してください。<br>";
}
//同一IDの登録禁止
//データベース連携オブジェクト
$db = new DatabaseBridge();
//管理者情報取得クエリ
$query = 'SELECT
            count(auth_key) AS same_count
          FROM
            auth_tab
          WHERE
            auth_name = \''.$auth_name.'\' AND
            auth_key != '.$_SESSION['auth_key'].' AND
            admin_flg = \''.DEV_FLG.'\'';
if($db->query($query)){
  $result = $db->get_result();
  $assoc = pg_fetch_assoc($result);
  if($assoc['same_count'] != 0){
    $error_msg .= "既に同じ開発者IDが登録されています。<br>";
  }
}
else{
  echo $db->get_error();
}

if($auth_pass == ''){
```

```
    $error_msg .= "パスワードが未入力です。<br>";
  }
  if(preg_match('/[^0-9a-zA-Z]/',$auth_pass)){
    $error_msg .= "パスワードは半角英数字で入力してください。<br>";
  }
  if(preg_match('/[0-9a-zA-Z]/',$auth_pass) && strlen($auth_pass) > 15){
    $error_msg .= "パスワードは15文字以内で入力してください。<br>";
  }

  if($error_msg != ''){
    $cs = new CustomSmarty();
    $cs->assign('auth_name',$auth_name);
    $cs->assign('auth_pass',$auth_pass);
    $cs->assign('error',$error_msg);
    $cs->display('dev_edit.tpl');
  }
  else{
    //セッションに入力値の保存
    $_SESSION['auth_name'] = $auth_name;
    $_SESSION['auth_pass'] = $auth_pass;

    $cs = new CustomSmarty();
    $cs->assign('auth_name',$auth_name);
    $cs->assign('auth_pass',$auth_pass);
    $cs->assign('massage','この内容で更新しても宜しいですか？');
    $cs->display('dev_confirm.tpl');
  }
?>
```

dev_edit.php

```
<?php
  //ログインチェック
  require_once('../libs/dev_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['auth_name']) && isset($_SESSION['auth_pass'])){
    $cs = new CustomSmarty();
    $cs->assign('auth_name',$_SESSION['auth_name']);
    $cs->assign('auth_pass',$_SESSION['auth_pass']);
    $cs->display('dev_edit.tpl');
  }
  else{
    //データベース連携オブジェクト
    $db = new DatabaseBridge();

    //管理者情報取得クエリ
    $query = 'SELECT
                *
              FROM
                auth_tab
              WHERE
                auth_key = \''.$_SESSION['auth_key'].'\'
              AND admin_flg = \''.DEV_FLG.'\'';

    if($db->query($query)){
```

```
      $result = $db->get_result();
      if($assoc = pg_fetch_assoc($result)){
        $cs = new CustomSmarty();
        $cs->assign('auth_name',$assoc['auth_name']);
        $cs->assign('auth_pass',$assoc['auth_pass']);
        $cs->display('dev_edit.tpl');
      }
    }
    else{
      echo $db->get_error();
    }
  }
?>
```

dev_home.php

```
<?php
  //ログインチェック
  require_once('../libs/dev_check.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  $cs = new CustomSmarty();
  $cs->display('dev_home.tpl');
?>
```

dev_info.php

```
<?php
  //ログインチェック
  require_once('../libs/dev_check.php');
  //データベース連携クラスファイル
  require_once('../libs/DatabaseBridge.class.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');

  //セッション開始
  session_start();

  //セッションに入力値が保存されている場合
  if(isset($_SESSION['auth_name']) && isset($_SESSION['auth_pass'])){
    unset($_SESSION['auth_name']);
    unset($_SESSION['auth_pass']);
  }

  //データベース連携オブジェクト
  $db = new DatabaseBridge();

  //管理者情報取得クエリ
  $query = 'SELECT
              *
            FROM
              auth_tab
            WHERE
              auth_key = \''.$_SESSION['auth_key'].'\'
            AND admin_flg = \''.DEV_FLG.'\'';

  if($db->query($query)){
    $result = $db->get_result();
```

```
    if($assoc = pg_fetch_assoc($result)){
      $cs = new CustomSmarty();
      $cs->assign('auth_name',$assoc['auth_name']);
      $cs->assign('auth_pass',$assoc['auth_pass']);
      $cs->display('dev_info.tpl');
    }
  }
  else{
    echo $db->get_error();
  }
?>
```

file_edit.php

```
<?php
  //
  require_once('../libs/dev_check.php');
  //Smarty鬣
  require_once('../libs/CustomSmarty.class.php');
  //
  require_once('../libs/config.php');
  //譯 鬣鬣
  require_once('../libs/utils.php');

  //逾
  session_start();

  //  熙萃
  $path = WORK_SPACES.$_SESSION['auth_key'];
  //       霹
  $files = get_files($path);
  //鬟皈魘
  $error_msg = '';

  //     霹
  if(is_numeric($_GET['num'])){
    //
    $file_number = $_GET['num'];
    //
    $file_name = $path.'/'.$files[$file_number]['name'];
    //     Ł
    if($file_pointer = fopen($file_name , 'rb')){
      if(filesize($file_name) > 0){
        //  簣 霹
        $contents = fread ($file_pointer , filesize($file_name));
      }
      else{
        $contents = '';
      }
      //
      fclose($file_pointer);
    }
    else{
      //   Ł
      $error_msg .= "     Ł   <br>";
    }
  }
  else{
    //GET
    if(!is_numeric($_GET['num'])){
      $error_msg .= "  讀   <br>";
    }
  }
```

```
  if($error_msg != ''){
    $cs = new CustomSmarty();
    $cs->assign('files',$files);
    $cs->assign('message',$error_msg);
    $cs->display('file_list.tpl');
  }

  $cs = new CustomSmarty();
  $cs->assign('source',$contents);
  $cs->assign('file_number',$file_number);
  $cs->assign('developer_number',$_SESSION['auth_key']);
  $cs->assign('file_name',$files[$file_number]['name']);
  $cs->display('file_edit.tpl');
?>
```

file_list.php

```
<?php
  //ログインチェック
  require_once('../libs/dev_check.php');
  //Smarty拡張クラス
  require_once('../libs/CustomSmarty.class.php');
  //コンフィグファイル
  require_once('../libs/config.php');
  //ユーティリティライブラリ
  require_once('../libs/utils.php');

  //セッション開始
  session_start();

  //開発者ディレクトリへの絶対パス
  $path = WORK_SPACES.$_SESSION['auth_key'];
  //存在するファイル情報取得
  $files = get_files($path);
  //エラーメッセージ初期化
  $error_msg = '';

  //ファイル新規作成時
  if($_POST['regist']){
    //ファイル名
    $file_name = htmlspecialchars($_POST['file_name']);
    //エラーチェック
    if($file_name == ''){
      $error_msg .= "ファイル名が未入力です。<br>";
    }
    if(preg_match('/[^0-9a-zA-Z.]/',$file_name)){
      $error_msg .= "ファイル名は半角英数字で入力してください。<br>";
    }
    if(preg_match('/[0-9a-zA-Z.]/',$file_name) && strlen($file_name) > 40){
      $error_msg .= "ファイル名は40文字以内で入力してください。<br>";
    }
    for($loop = 0 ; $loop < count($files) ; $loop++){
      if($files[$loop]['name'] == $file_name){
        $error_msg .= "存在するファイル名は使用できません。<br>";
      }
    }
    if(!isset($_POST['file_type'])){
      $error_msg .= "ファイルタイプを選択してください。<br>";
    }
    if($error_msg == ''){
      //ファイル作成時
      if($_POST['file_type'] == 1){
        $file_pointer = fopen($path.'/'.$file_name,'w');
```

```
        fclose($file_pointer);
      }
      else if($_POST['file_type'] == 2){
        mkdir($path.'/'.$file_name,0777);
      }
    }
  }

  //ファイル削除時
  if($_POST['delete']){
    //ファイル番号
    $file_number = $_POST['file_number'];
    if(is_file($path.'/'.$files[$file_number]['name'])){
      if(unlink($path.'/'.$files[$file_number]['name'])){
        $error_msg .= $files[$file_number]['name'].'を削除しました。<br>';
      }
      else{
        $error_msg .= $files[$file_number]['name'].'の削除に失敗しました。
<br>';
        $error_msg .=
        'ファイルグループID:'.filegroup($path.'/'.$files[$file_number]).'<br>'.
        'ファイル所有者ID:'.fileowner($path.'/'.$files[$file_number]).'<br>'.
        'ファイル許可属性:'.fileperms($path.'/'.$files[$file_number]).'<br>'.
        'ファイルタイプ:'.filetype($path.'/'.$files[$file_number]).'<br>';
      }
    }
    else if(is_dir($path.'/'.$files[$file_number]['name'])){
      if(rmdir($path.'/'.$files[$file_number]['name'])){
        $error_msg .= $files[$file_number]['name'].'を削除しました。<br>';
      }
      else{
        $error_msg .= $files[$file_number]['name'].'の削除に失敗しました。
<br>';
        $error_msg .=
        'ファイルグループID:'.filegroup($path.'/'.$files[$file_number]).'<br>'.
        'ファイル所有者ID:'.fileowner($path.'/'.$files[$file_number]).'<br>'.
        'ファイル許可属性:'.fileperms($path.'/'.$files[$file_number]).'<br>'.
        'ファイルタイプ:'.filetype($path.'/'.$files[$file_number]).'<br>';
      }
    }
  }

  //ファイル変更時
  if($_POST['rename']){
    //ファイル番号
    $file_number = $_POST['file_number'];
    //変更後のファイル名
    $file_name = htmlspecialchars($_POST['file_name']);
    //エラーチェック
    if($file_number == ''){
      $error_msg .= "変更するファイルを選択してください。<br>";
    }
    else{
      if($file_name == ''){
        $error_msg .= "ファイル名が未入力です。<br>";
      }
      if(preg_match('/[^0-9a-zA-Z.]/',$file_name)){
        $error_msg .= "ファイル名は半角英数字で入力してください。<br>";
      }
      if(preg_match('/[0-9a-zA-Z.]/',$file_name) && strlen($file_name) > 40){
        $error_msg .= "ファイル名は40文字以内で入力してください。<br>";
      }
      if($files[$file_number]['name'] == $file_name){
```

```
        $error_msg .= "ファイル名が同じです。<br>";
      }
      for($loop = 0 ; $loop < count($files) ; $loop++){
        if($file_number != $loop && $files[$loop]['name'] == $file_name){
          $error_msg .= "存在するファイル名は使用できません。<br>";
        }
      }
    }
    if($error_msg == ''){
      if(rename($path.'/'.$files[$file_number]['name'] , $path.'/'.$file_name)
        $error_msg .= $files[$file_number]['name'].'を変更しました。<br>';
      }
      else{
        $error_msg .= $files[$file_number]['name'].'の変更に失敗しました。
<br>';
        $error_msg .=
        'ファイルグループID:'.filegroup($path.'/'.$files[$file_number]).'<br>'.
        'ファイル所有者ID:'.fileowner($path.'/'.$files[$file_number]).'<br>'.
        'ファイル許可属性:'.fileperms($path.'/'.$files[$file_number]).'<br>'.
        'ファイルタイプ:'.filetype($path.'/'.$files[$file_number]).'<br>';
      }
    }
  }

  //開発者ディレクトリに含まれるファイル名の配列
  $files = get_files($path);

  $cs = new CustomSmarty();
  $cs->assign('files',$files);
  if($error_msg != ''){
    $cs->assign('message',$error_msg);
  }
  $cs->display('file_list.tpl');
?>
```

save_file.php

```
<?php
  //ログインチェック
  require_once('../libs/dev_check.php');
  //コンフィグファイル
  require_once('../libs/config.php');
  //ユーティリティライブラリ
  require_once('../libs/utils.php');

  //受信時
  $start_time = get_microtime();

  //開発者ディレクトリへの絶対パス
  $path = WORK_SPACES.$_SESSION['auth_key'];
  //存在するファイル情報取得
  $files = get_files($path);

  //編集データを受信
  $source =
    preg_replace(
      "/&file_number=".$_GET['file_number']."&source=/","",$_SERVER[QUERY_STRI
      );

  //URLデコード
  $source = urldecode(stripslashes($source));

  //データのエンコード
  $source = mb_convert_encoding($source , 'EUC-JP' , 'UTF-8');
```

```
  //ファイル情報取得
  if(is_numeric($_GET['file_number'])){
    //ファイル番号
    $file_number = $_GET['file_number'];
    //ファイル名
    $file_name = $path.'/'.$files[$file_number]['name'];
    //ファイルポインタをオープン
    if($file_pointer = fopen($file_name , 'wb')){
      //ファイル内容を書き込む
      fwrite($file_pointer , $source);
      //ファイルポインタをクローズ
      fclose($file_pointer);
    }
    else{
      //ポインタをオープンできなかった場合
      $error_msg .= ' ファイルポインタをオープンできませんでした。<br>';
    }
  }
  else{
    //GET 文字列が数値文字列で無い場合
    if(!is_numeric($_GET['file_number'])){
      $error_msg .= ' ファイル番号が含まれていません。<br>';
    }
  }

  //完了時
  $end_time = get_microtime();

  //動作時間
  $diff_time = $end_time - $start_time;

  if($error_msg == ''){
    echo $files[$file_number]['name'].'の手動更新完了['.date('Y-m-d H:i:s').']';
  }
  else{
    echo ' エラー発生<br>';
    echo $error_msg;
  }
?>
```