Description of Web Development

1. Website structure overview

There are 9 pages we design for the website and now we finished the interface design of log-in page, Register-Page, Homepage, User-index Page, Calendar-records page, history page and Weekly Records page.

We have fullfilled the function of log-in page, Register page, Home Page, User-Index Page and Calendar-records Page.

The structure of the website is shown in figure 1.1 and figure 1.2.

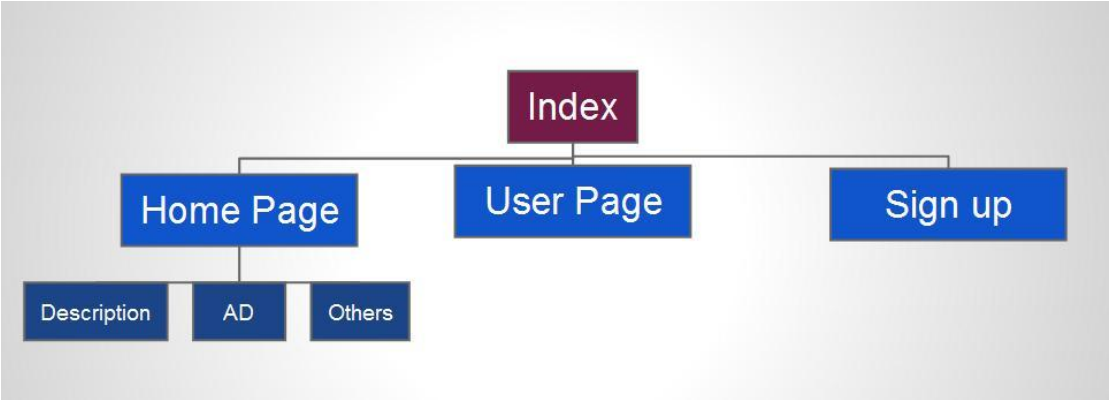


Figure 1.1 Structure of the Site

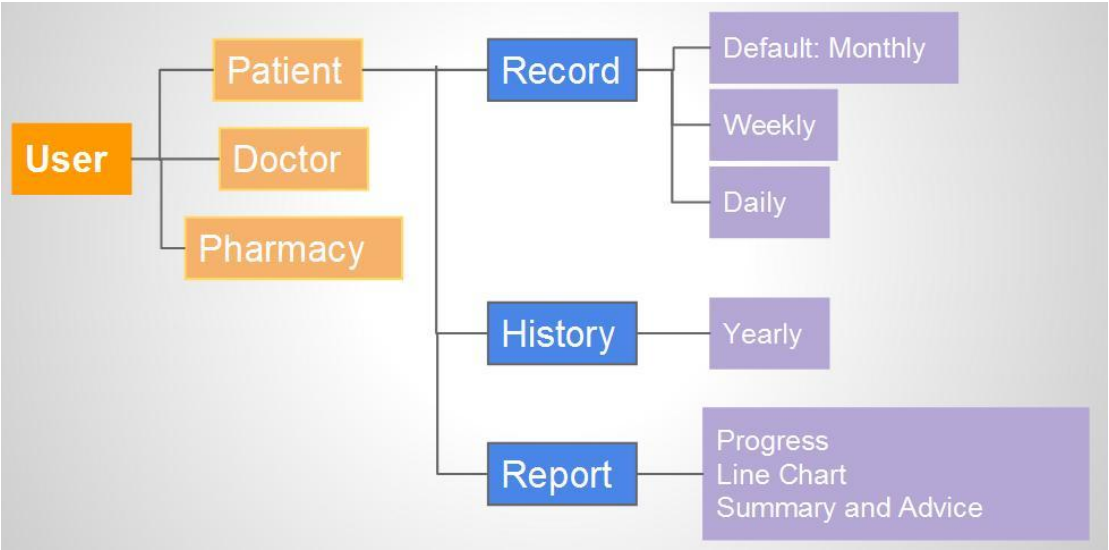


Figure 1.2 Structure of User Pages

Figure 1.3 and figure 1.4 show the all files we have.

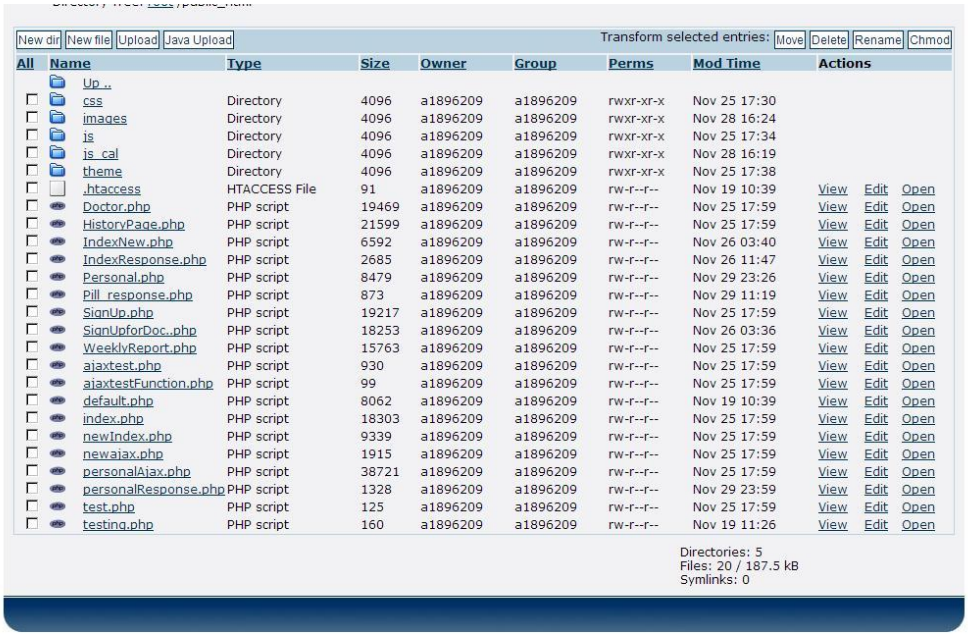


Figure 1.3 Files for the website

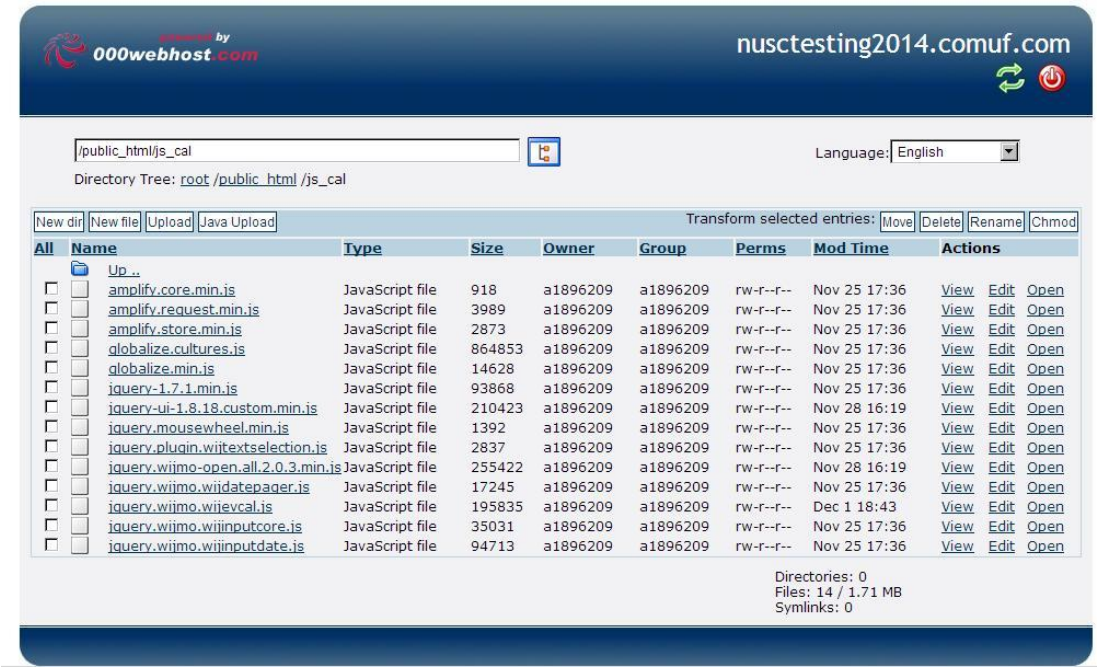


Figure 1.4 Js Files for calendar widget

2 Index Page

2.1 Operation

Users can log-in their account at this page. The index page is shown in Figure 3.1.

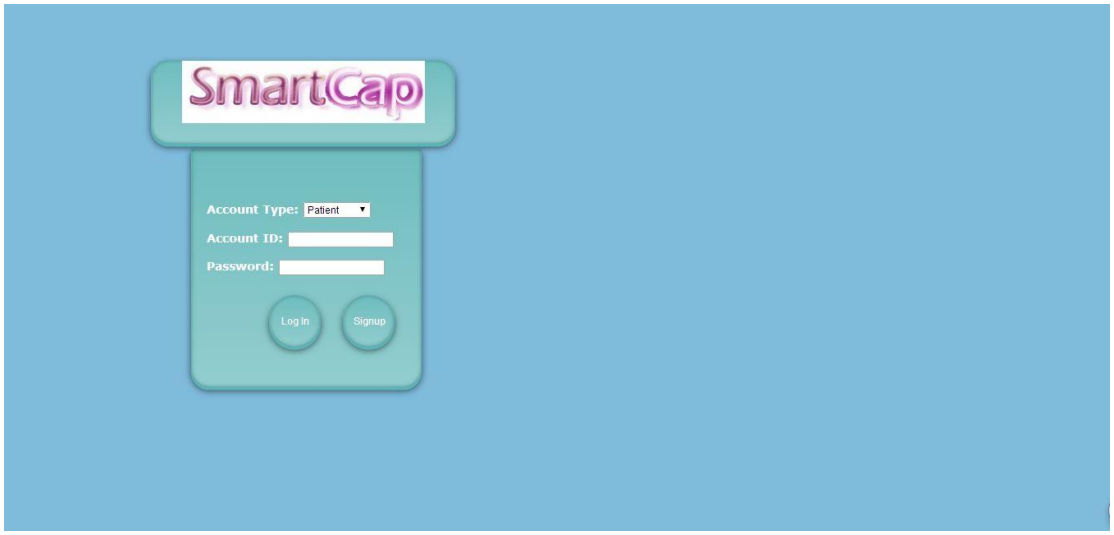


Figure 3.1 Index Page

2.2 Code Description

We use ajax to process data from front-end (the page) to backstage and fetch result from database. We have the file IndexResponse.php to support the page to fetch data from our database. We will post the data from the form in the website, including User types, User ID and Password. In the login page, we need to compare the password of a certain user stored in the database. If the password equal to the records in our database, it will be a valid user then the page will jump to the user-index page.

3 User page

3.1 User Index Page

User index page is a navigation page.

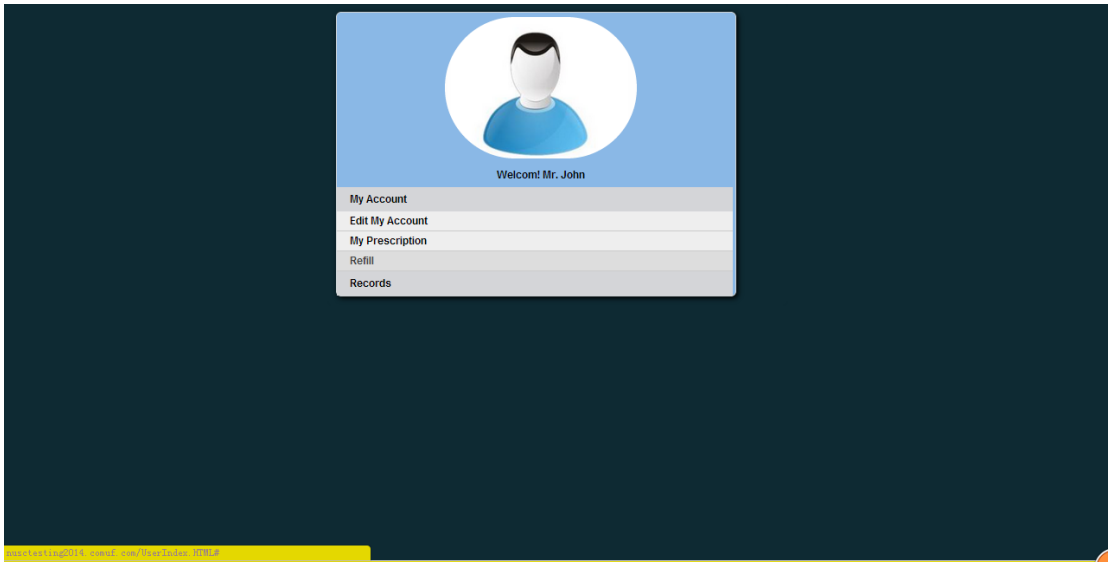


Figure 3.1 UserIndex Page

3.2 Calendar Page

The calendar page the primary page to show the records to our users. The calendar page is shown on figure 3.2.

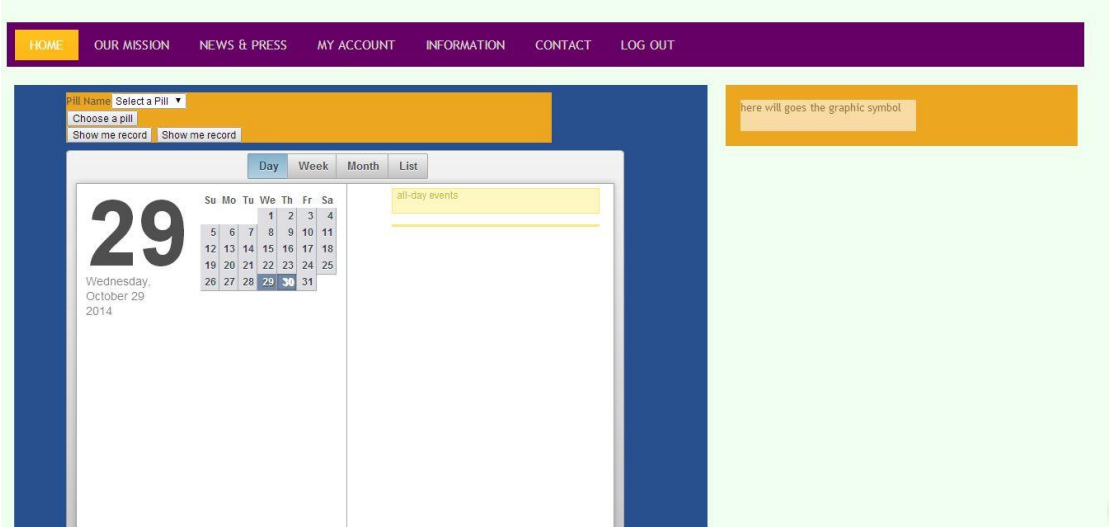


Figure 3.2 Calendar Page

There are 4 sub view of the calendar and they are day view, week view, month view and list view. We only use the first three view.

There are four files to support the implement the pages. They are Personal.php, pill_response.php, personalResponse.php and jquery.wijimo.wijevcal.js.

The Personal.php is the main file for the structure implement of the calendar web page. Here is the interface design and the basic operation implementation.

The jquery.wijimo.wijevcal.js is the most important file contains the jquery functions to implement the operation of calendar to show the records to users. They functions we used to show records are list in the following table.

Table 3.1

Function	Description
_onDayViewClick: function (){}	The function responses to the click of “Day” button of the calendar to show the records to user. Here we will change the right side to show the remainder from other users like doctors or pharmacy. We will show the pill-taking records to users in the calendar.
_onWeekViewClick: function (){}	The function responses to the click of “Week” button of the calendar to show the records to user. Here we will change the right side to show the progress of each pill taken this week. We will show the pill-taking records to users in the calendar.
_onMonthViewClick: function (){}	The function responses to the click of “Month” button of the calendar to show the records to user. Here we will change the right side to show the symbol of each pill. We will show the pill-taking records to users through the month.

The pill_response.php is used to get the pill names of a certain user from our database. The

functions list in the Table 3.1 call request data from the file via ajax and it will return the pill name of which are taken by the user.
The personalResponse.php is used to count times of a certain pill taken at a certain date.

The structure of the function is shown as following.

```
/*Get the pill name from database for the user via ajax, requesting data from pill_response.php*/
$.ajax({
        url:'http://nusctesting2014.comuf.com/Pill_response.php',        type:'post',
        data: { PatientID: patient_id },
        async:false,
        success:function(result){
        //process the back value from pill_response.php
        //Here we get the pill name.
        },
        error:function(msg){
            alert('Error:'+msg);
        }
    });
/**************end of ajax*******************************/
//set content of right part here
            var element_add=// add the DOM tag here
            $(".right_part").html(element_add);//change the right side with the DOM tag
/**************add row*********************************/
for(p=0;p<j;p++){    //We use the loop to add ruler for each pill.
                    //p is the number of pills taken by the user
    ruler_pill_name="<div class=\"wijmo-wijev-timerulerinterval ui-widget-content
    wijmo-wijev-pill-0 wijmo-wijev-timerulerinterval-pillName\" style=\"height:50px\"
    id=\"";
    ruler_pill_name+=pill[p]+"\">";
    ruler_pill_name+=pill[p]+"</div>";//This is the format in html for ruler
    $("#pill-name-ruler").append(ruler_pill_name);..add the ruler to the calendar

/*****************add records*******************************/
    for(c=0;c<7;c++){ //The loop is used to add records for each pill
        //firstly we need to convert the class name into date type
        //For each element in the calendar, we insert the date represented by it
        column_class=cur[c].className;
        column_date=column_class.substring(39,43);
        column_date+="-";
        if(column_class.substring(44,45)=="1"){
                    if(column_class.substring(45,46)=="_"){
                        column_date+="02-";
                        column_date+=column_class.substring(46,48);
                    }
```

```
                    else{
                        switch (column_class.substring(44,46)){
                            case "10": column_date+="11";
                                    break;
                            case "11": column_date+="12";
                                    break;

                            default:break;
                           }
                        column_date+="-";
                        column_date+=column_class.substring(47,49);
                     }
                  }

               else{
                    switch (column_class.substring(44,45)){
                            case "0": column_date+="01";
                                    break;
                            case "2": column_date+="03";
                                    break;
                            case "3": column_date+="04";
                                    break;
                            case "4": column_date+="05";
                                    break;
                            case "5": column_date+="06";
                                    break;
                            case "6": column_date+="07";
                                    break;
                            case "7": column_date+="08";
                                    break;
                            case "8": column_date+="09";
                                    break;
                            case "9": column_date+="10";
                                    break;
                            default:break;
                     }
                     column_date+="-";
                     column_date+=column_class.substring(46,48);
               }
//Now we get the date represented by the element
/**********************Get records of taken time for the pill at that date.****************/
//The personalResponse.php is used to request.
  $.ajax({
          url:'http://nusctesting2014.comuf.com/personalResponse.php',
```

```
            type:'post',
            data: { curdate: column_date,pillname:pill[p] },
            async:false,
            success:function(result){

                times = result;//receive the times
                },
            error:function(msg){
                alert('Error:'+msg);
            }
        });

        record = record + times+"</div>";
        $(".wijmo-wijev-daycolumn").each(function(c){
            $(this).append(record); //add records into the block element
            }) ;
    }
}
```

3.2.1 Fetch array from ajax

Because we can only get a string from the ajax, a simple way to get an array form ajax is to convert the array into a string in a certain format and convert the string into array when we get the string.

Take the pill name fetch from database as an example.

We will request search in the file “jquery.wijimo.wijevcal.js” and file “pill_response.php” will response to the request.

In file “pill_response.php”, we get the array contains pill name get from database. We using the for-loop to fetch the result and save it into the array “$pill[]”. Then will build a string in format “pill1”“pill2”...“pilln”. The for-loop is shown as following.

```
for($i=0; $row = $stmt->fetch(); $i++){
            $pill[$i] = $row['pill_name'];
            $pillname=$pillname."\"".$pill[$i]."\"";
          }
```

In the call back of ajax, the success function, we rebuild the string into array using following for-loop.

```
success:function(result){
                    alert(result);
                    receive=result;
                    //data = result;
                    //alert(data);
                    //convert into array
                while(i<result.length){
                   pill[j]="";
                    if (result[i]="\""){
                        i=i+1;
```

```
                while(result[i]!="\""){
                pill[j]=pill[j]+result[i];
                i++;
                }

            j++;
            i++;

            }
        }
},
```

The outside for-loop is used for detect an element of array which is start from quote (“) and the inner for-loop is used for rebuilding the array by combining each letter from the string until meets another quote(”).

3.3 History and Record Page

We use API for chart building from google in these two pages. The API is called “chart gallery”. In our website, we need at least two type of charts. One is line chart. Another is a calendar chart. You can find a tutorial for using this API by the following website.

https://developers.google.com/chart/interactive/docs/gallery

The figure is shown in figure 3.3， figure 3.4 and figure 3.5.

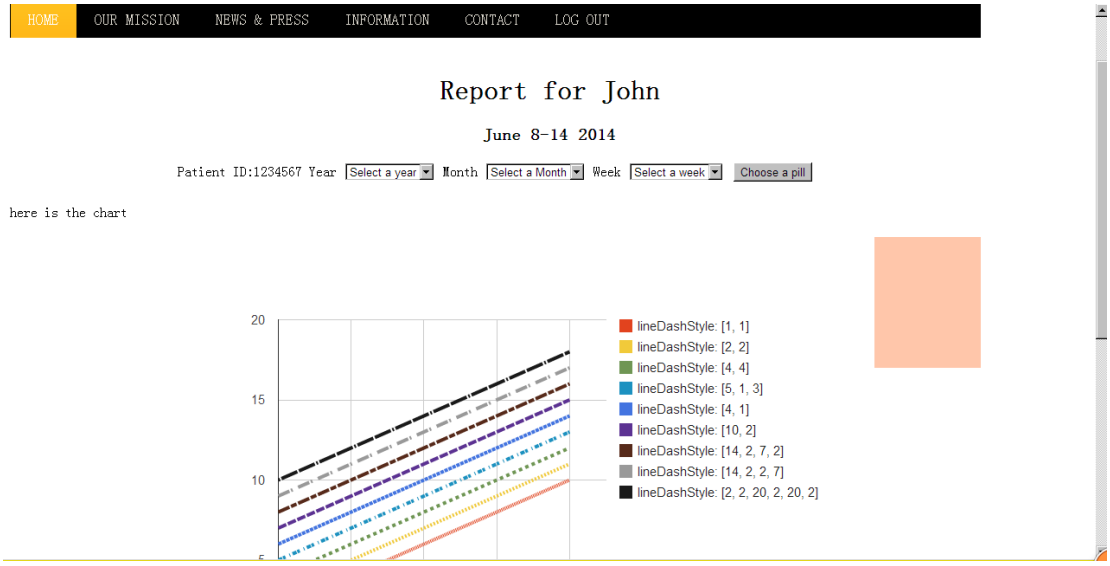
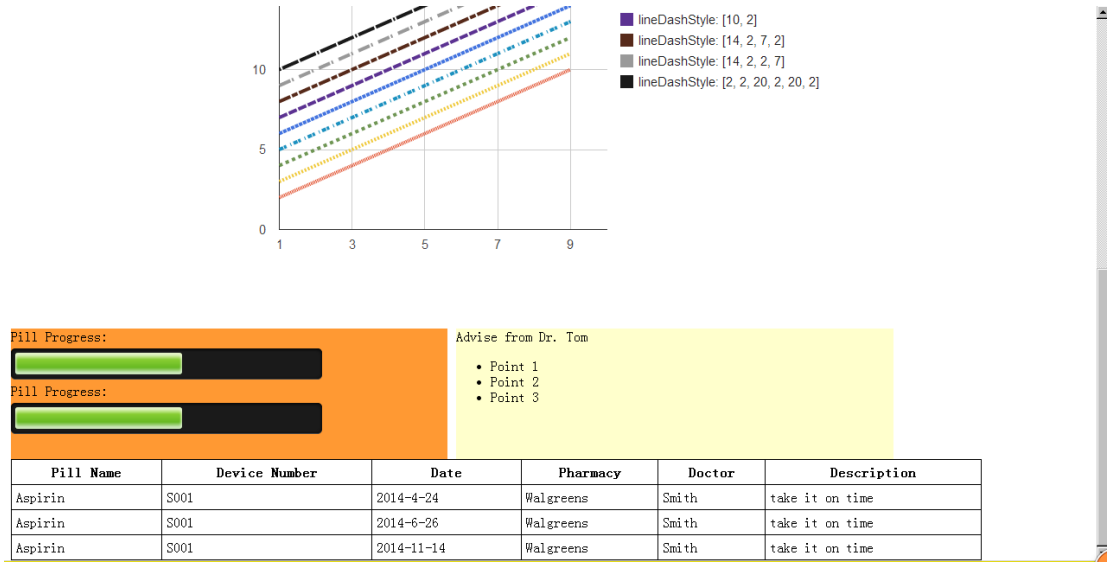


Figure 3.3 Line chart to trace weekly pill taken



Pill Name	Device Number	Date	Pharmacy	Doctor	Description
Aspirin	S001	2014-4-24	Walgreens	Smith	take it on time
Aspirin	S001	2014-6-26	Walgreens	Smith	take it on time
Aspirin	S001	2014-11-14	Walgreens	Smith	take it on time

Figure 3.4 Lower part for Weekly Report Page

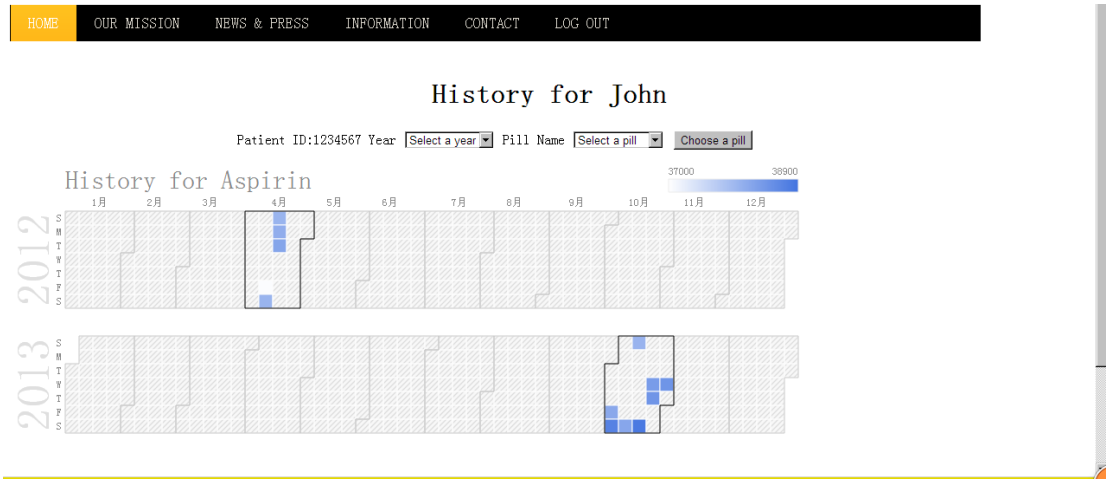


Figure 3.5 Calendar chart to trace the yearly pill taken

For the WeeklyReport page, we need to make a summary for a certain user of a week. The statistic data used for line chart and progress bar should be generated from our database.

The basic idea is that we get data from database by php and use the data in the API. Also we can use ajax for it.

There is an example of how to get data from database and use it in the API. This part of code is commented in WeeklyReport.php.

```
<?php
echo"<script type=\"text/javascript\">";
        echo"var date=new Array();
                var pill=new Array();
                var times=new Array();";
                $r = count($targetDate_array);
                $index=$r-2;
                while ($r-1){
```

```
echo                    " var t=".$index;
echo                ";
                            date[t]=\"".$targetDate_array[$index];
echo            "\";";
$r--;
$index--;
}

 $t = count($targetTimes_array);
            $i=$t-2;
            while ($t-1){

echo                    " var t=".$i;
echo                ";
                            times[t]=".$targetTimes_array[$i];
echo            ";";
$t--;
$i--;
}

      echo "    google.load(\"visualization\", \"1\", {packages:[\"corechart\"]});
         google.setOnLoadCallback(drawChart);
         function drawChart() {


           var data = google.visualization.arrayToDataTable([
            ['Day', 'Aspirin'],";
             $n=2;
             while($n>0){

                  echo "[date[".$n;
                  echo "], times[".$n;
                  echo "]],";
                     $n--;
                  }
                  echo "[date[0],times[0]]



           ]);

           var options = {
              title: 'Daily Pill Taking'
           };
```

```
        var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
        chart.draw(data, options);
    }
</script>";
?>
```

The history page is like the WeeklyReport page. The code is commented in History.php.

3.4 Message Page

Users can manage and review message from doctors, pharmacies and other users at this page. We use CSS to implement the display.

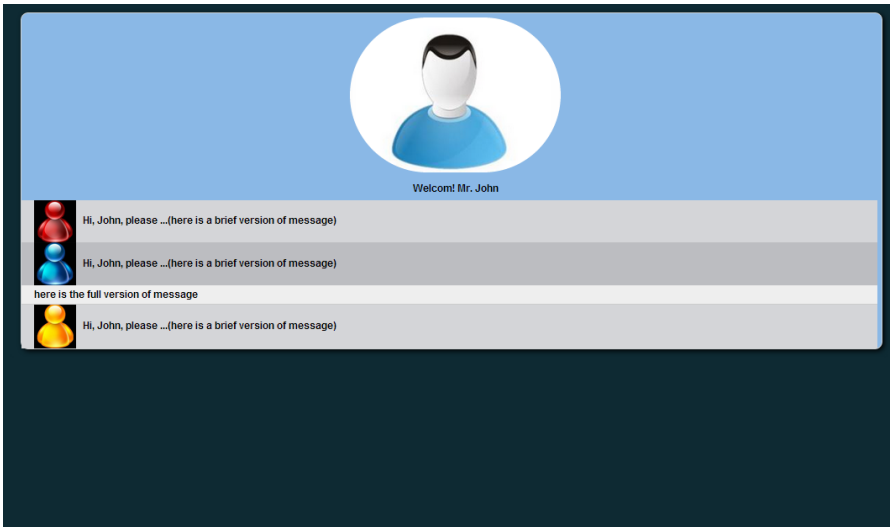


Figure 3.6 Message Page

4 To be continued

What we have done till now is,

- The format and interface of each page we designed.
- The basic function to show records on calendar page.
- The basic function of history page.
- The basic function of weekly record page.

What are remained to be done are,

- The history page is needed to get the real data from our database and we need to show the frequency of pill taking by color. The basic function for connecting database is commented in the code.
- The report page is needed to get the real data from our database. The basic function for connecting database is commented in the code.
- The function of message page should be implemented.
- The human computer interaction modified.

5 000Webhost Description

The 000Webhost is a free online web server. Currently, we use the server for our website testing.

The account information of 000webhost:

Account:xiaoran1026@hotmail.com

Password:Smartcap.2014

The account information is shown as following.

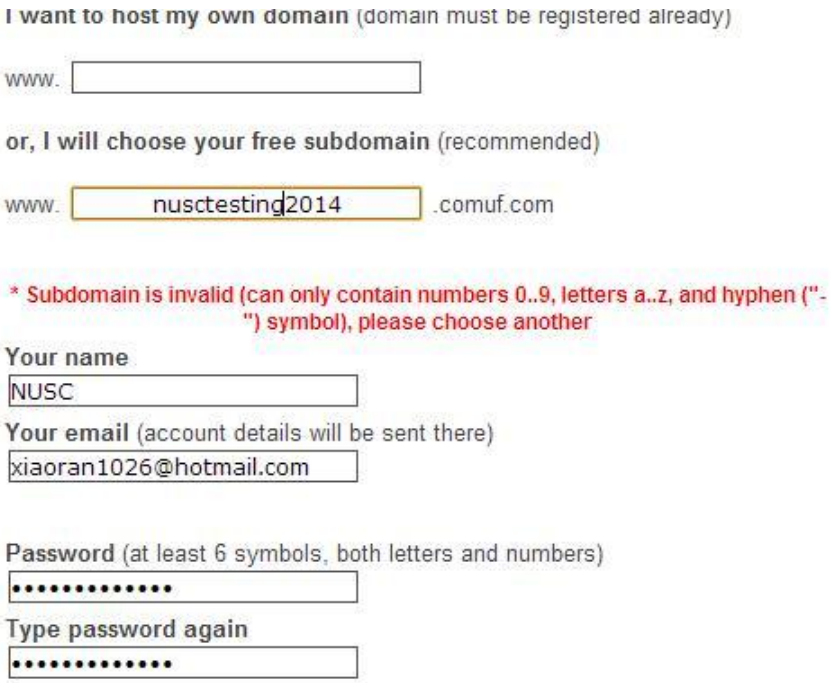


Figure 5.1 Account Information

In the 000webhost, we use MySQL as the database. The account information is showin figure 2 and figure 3.

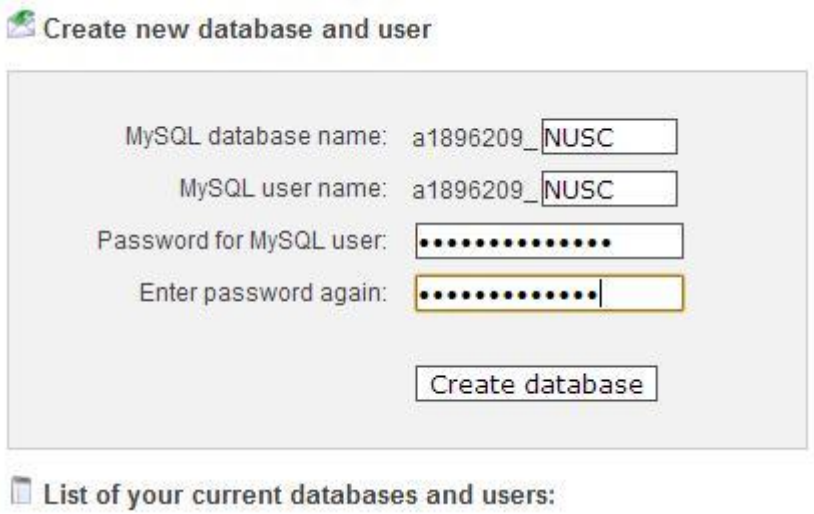


Figure 5.2 MySQL Account Information

Figure 3 MySQL Information