McCormick
Northwestern Engineering

EECS499 Report

Web Development

Qiyuan HU

Biao He

2015 winter

Content

I Vocabulary

Agile Method: a group of software development method based on iterative and incremental development, where requirement and solutions evolve through collaboration between self-organizing, cross-functional teams.

AJAX: Asynchronous JavaScript and XML, a technique for creating fast and dynamic web pages.

Apache: The HTTP Server

CSS: Cascading Style Sheets.

Deploy: All of the activities that make a software system available for use.

HTML: Hyper Text Markup Language.

IDE: Integrated Development Environment.

JavaScript: The scripting language of the Web.

PHP: A server-side scripting language designed for web development.

PostgreSQL: An open source object-relational database management system.

SVN: A software versioning and revision control system distributed as free software under the Apache License.

WebMatrix: The web development application for Windows.

II Introduction

This report is about our EECS 499 Individual Study project, we are involved in the web development team for the Smart Cap project.

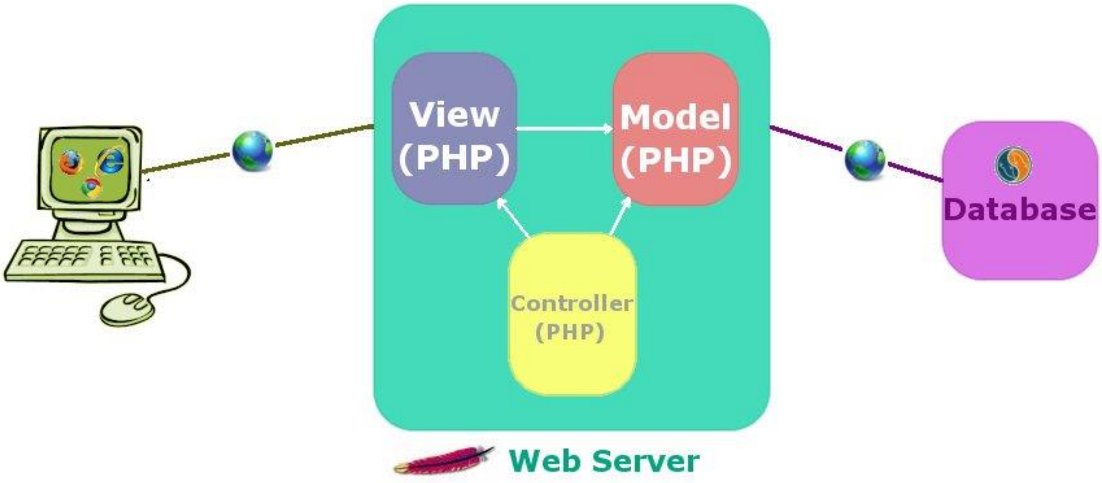
Smart Cap Project combines Hardware, Web Development, Machine Learning, Data Transferring and Mobile App Development. It is aimed to monitor the patient status of taking pills, and represent data in a visualized formation to doctors, patient relatives and patients themselves.

There is a small camera at the cap of the pill bottle, and it will take pictures which contains all the pills remain in the bottle. And Machine Learning will determine the pill that the patient took, and transfer the data to the database. After that, the Website and Android App will be able to show the data in a meaningful way.

We are working for the Web part, the framework is built by the former developer, so our tasks is to retrieve the “real data” from the database, and present them in some chart which are designed by the project manager.

1 Technical Introduction

1.1 Architecture of the project

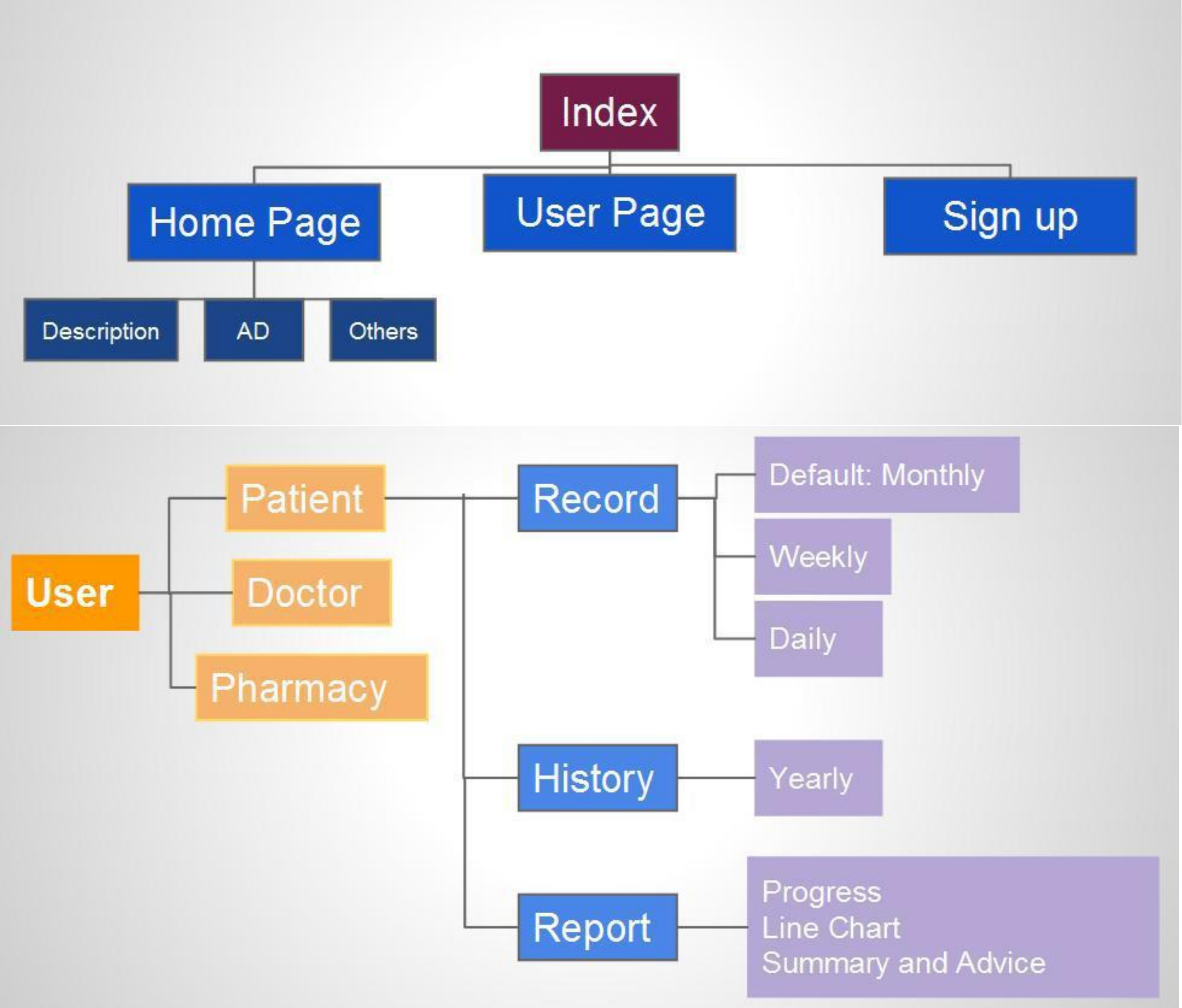


Picture 1.1.1

We are using a PHP based web design architecture where contains the following part:

- We choose PostgreSQL as our database not only because it is free & public database, but also our data will be transferred from another PostgreSQL database which associated with the hardware device directly in the future. So for the convenience of the developers who will take over the web part, we change the database back from MySQL to PostgreSQL, and modified every queries so that it will works with PostgreSQL.
- At the middle it is PHP, we deployed it using a MVC (Model-View-Controller) design pattern, well-developed MVC system should allow a front-end developer and a back-end developer to work on the same system without interfering, sharing, or editing files either party is working on:
  - Model: The model manages the behavior and data of applications, responds the request from controller (view) to deal things with data.
  - View: To display information.
  - Controller: It acts as the communicator sending requests and receiving responds. MVC is probably one of the most popular frameworks.
- Apache HTTP Server is our server, so that the web browser can display our page.

1.2 The Website Structure



Picture 1.2.1

The picture above is the Web page structure, the log-in page, Register page, Home page, User-Index Page and Calendar-records page are working correctly before. And at this quarter we added the History page and Report Page, and made all pages (included 5 pages before) work correctly with the ability querying real data from database.

1.3 Team & Working environment

This project is like a real software development practice project, we have two developers and three project managers who are the professors. When we develop the code, we tried make it using like an Agile Methodology. Agile method is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. We have weekly meeting with our project managers and get feedbacks from them, then working on the same page develop out the new functions when they have new ideas. We pretend to make new versions and release frequently to meet their requirement.

1.4 Version Control

Since we have two developers, we need find a way to control our versions, to make it easy to combine our works together and more important is to ensure the combined code works correctly.

Picture 1.4.1

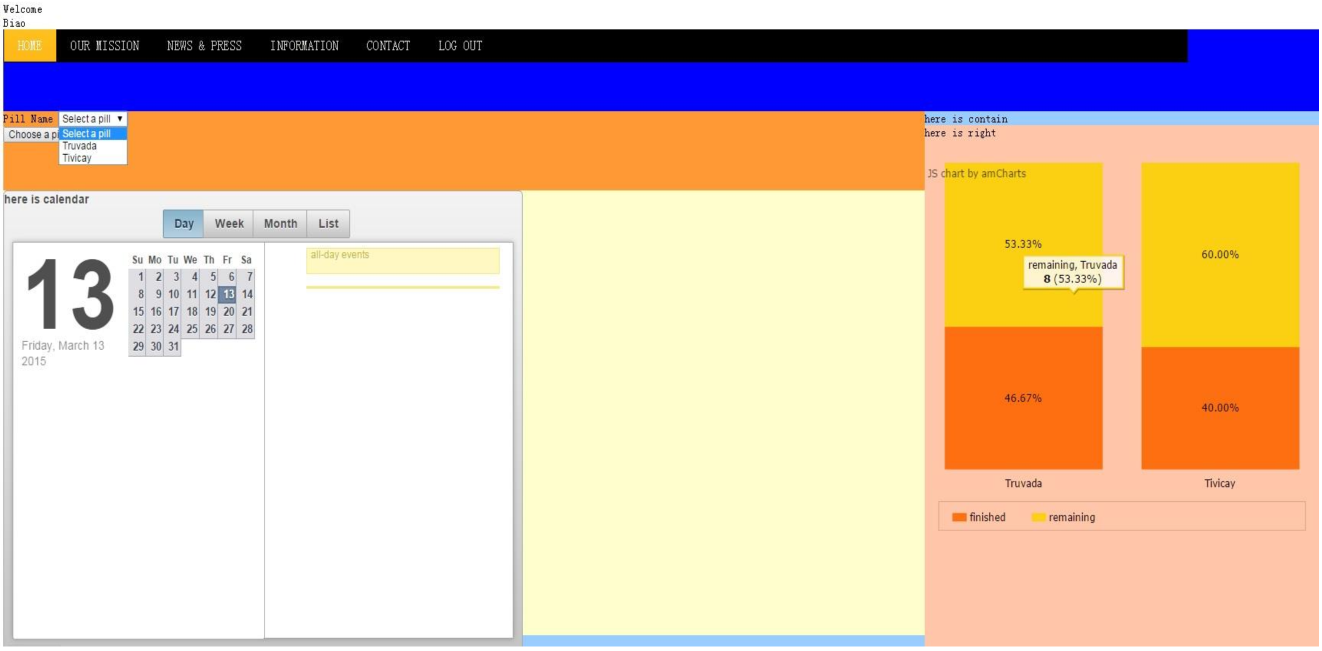
Apache Subversion (often abbreviated SVN, after the command name SVN) is a software versioning and revision control system distributed as free software under the Apache License. Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation.

2 Achievements

2.1 Patient Home Page

The previous developer has implemented the patient home page, but it is just a layout display that is the data are not real data from the database. Also, the session was not kept so when jumping to another page, the login information will be lost.

We added query, and let PHP model layer run the queries via PostgreSQL driver, and PHP controller layer (some works are done by AJAX in JQuery) to process the data we got. Finally display it in the calendar chart.



Welcome
Biao

here is calendar

Friday, March 13
2015

here is contain
here is right

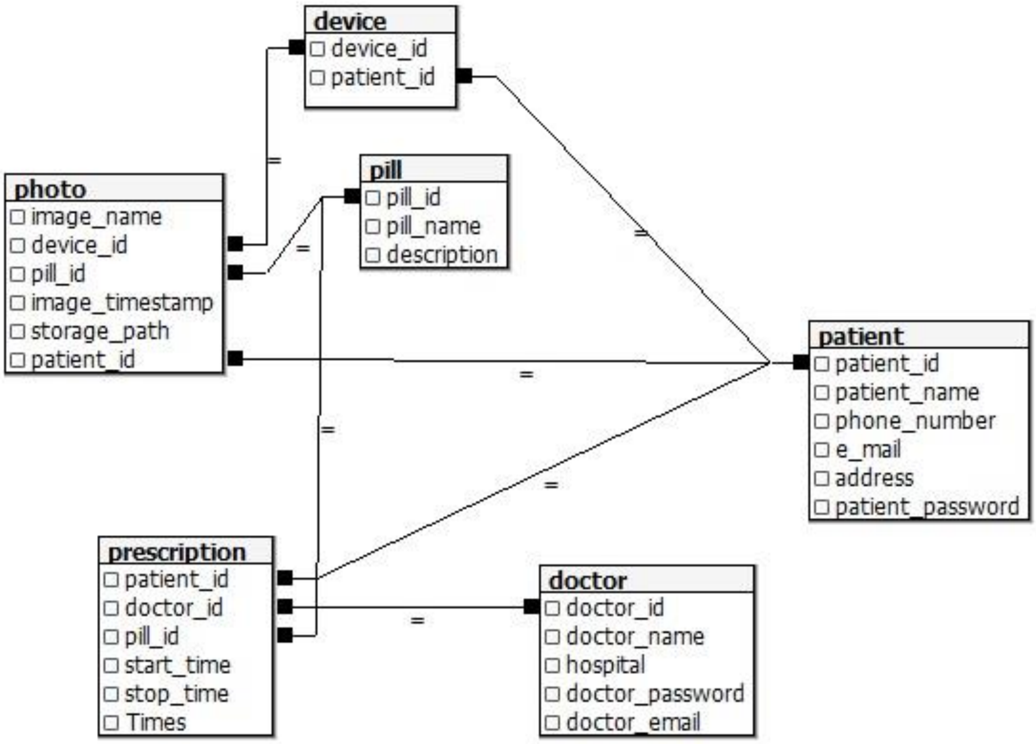
Picture 2.1.1

We use HTTP session to keep the login info for example:

```
$_SESSION['accountID']=$_SESSION["accountID"];
$_SESSION['type']=$_SESSION["type"];;
$_SESSION['password']=$_SESSION["password"];
```

2.2 Build PostgreSQL Database

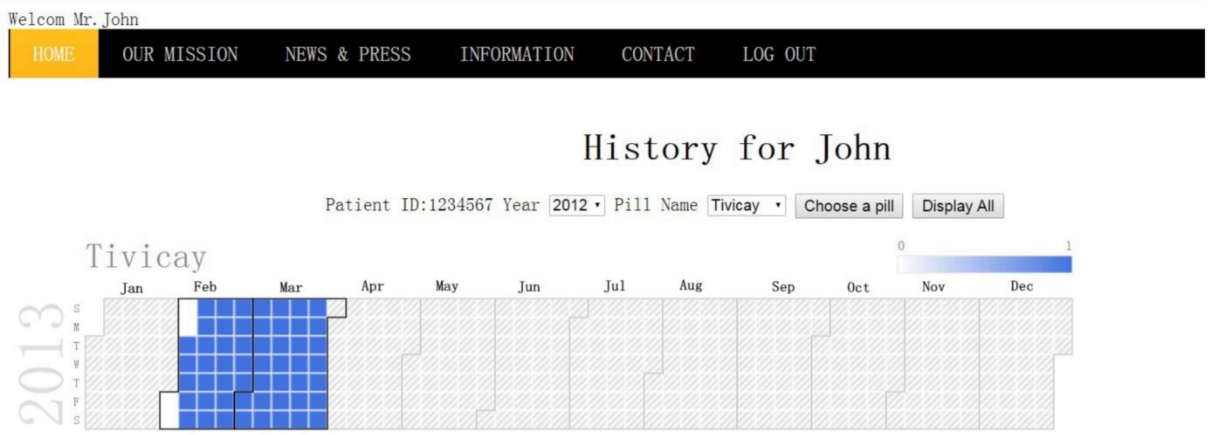
Since the old version database of this project is using MySQL, we need create new PostgreSQL database on our local and make it runnable when we query it from PHP. The ER model of the new version database is as follows:



Picture 2.2.1

2.3 History Report Chart

The History Report Chart will reflect the amount of number of pills that the patient forgot to take for a particular pill in a particular year range. Also the by the good using of the color, it will display more directly the serious level of the forgotten pills number. And it will let the doctors and patients find the relations of taking pills in time and the illness recover status.



Welcom Mr. John

History for John

Patient ID:1234567 Year 2012 Pill Name Tivicay

Picture 2.3.1

We designed it has two buttons: “Choose a pill” and “Display all”. The user needs choose the year and pill name for displaying the details of the pill in that year. If the user doesn’t choose, the default is the first result of the year and pill corresponded to the current user.

To query the database and process the data, we have three individual quires and two nesting iterations. We firstly query the prescription to see in this year for that pill, when and how many pills the patient should take, by running:

```
 SELECT * from prescription JOIN pill on prescription.pill_id = pill.pill_id where
(prescription.patient_id = '$PatientID' and pill.pill_name ='$Selected_pillname')
and ((start_time <= '$lastDayofYear' and start_time >= '$firstDayofYear') or
(stop_time <= '$lastDayofYear'and stop_time >= '$firstDayofYear'));
```

Then for each prescription records, we count how many pills the patient actually took in each day in the prescription date range, with the query:

```
SELECT count(*), image_timestamp::DATE from photo where patient_id = '$PatientID'
and pill_id in (select pill_id from pill where pill_name= '$Selected_pillname')
and image_timestamp >= '$startDates[$prescriptionIndex]' and image_timestamp <=
'$endDates[$prescriptionIndex]' GROUP BY image_timestamp::DATE;
```

We need save the “actuals”, generate the “zero” data for those days he didn’t even take one pill.

To create the button and make the page refresh, we created several JavaScript functions which are useful for this page and all the page we developed after:

```
function getSelectedText(name){
        var obj=document.getElementById(name);
        for(i=0;i<obj.length;i++){
            if(obj[i].selected==true){
                return obj[i].innerText;
        }
        }
        }
```

This function enables to select the text of all HTTP element with the same name as the input.

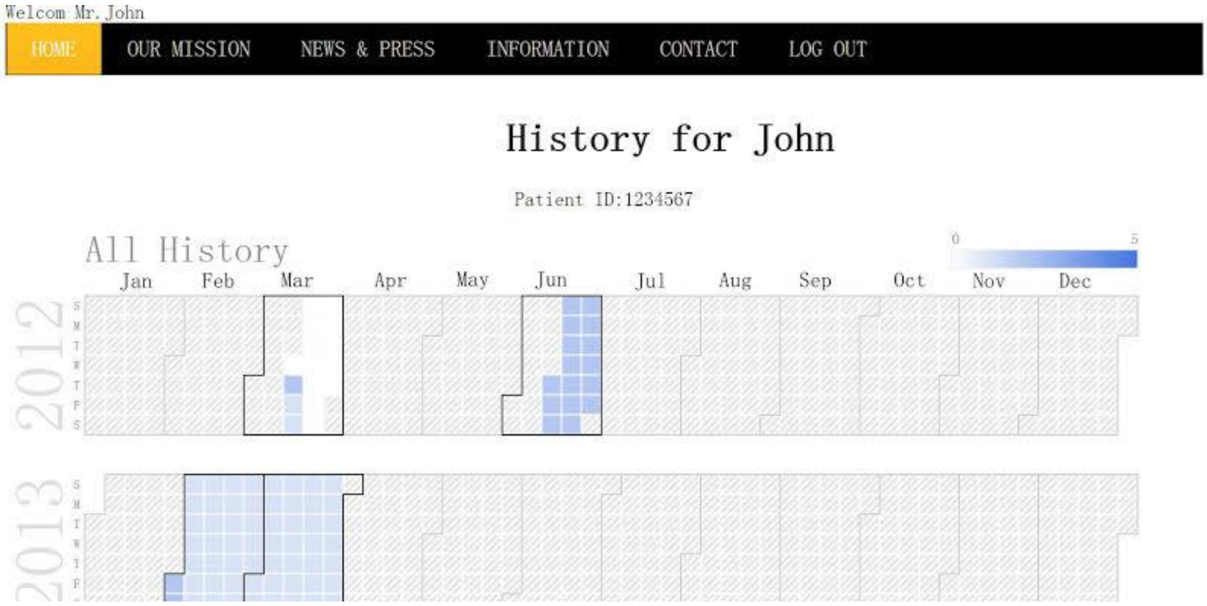
```
function PillAdding(year, name){
        alert(year + name);
        var postUrl = "./History.php";
        var postData = year;
        var msgData = name;
        var ExportForm = document.createElement("FORM");
        document.body.appendChild(ExportForm);
        ExportForm.method = "POST";
        var newElement = document.createElement("input");
        newElement.setAttribute("name", "PillSelection");
        newElement.setAttribute("type", "hidden");
        var newElement2 = document.createElement("input");
        newElement2.setAttribute("name", "YearSelection");
        newElement2.setAttribute("type", "hidden");
        ExportForm.appendChild(newElement);
        ExportForm.appendChild(newElement2);
        newElement.value = postData;
        newElement2.value = msgData;
        ExportForm.action = postUrl;
        ExportForm.submit();

        }
```

This function added a HTTP form with the POST function, to transfer data to next page.

This chart is implemented based on the google JavaScript API, we need to make good query and process the data as the JavaScript data format, and put it in the chart one by one. This chart is called “visualization.calendar”, you can find the tutorials in: https://developers.google.com/chart/interactive/docs/gallery/calendar

We also designed a button for display all, by using this feature, our users can browser all records for all pills in any years recorded in the database by clicking one that button.



Welcom Mr. John

History for John

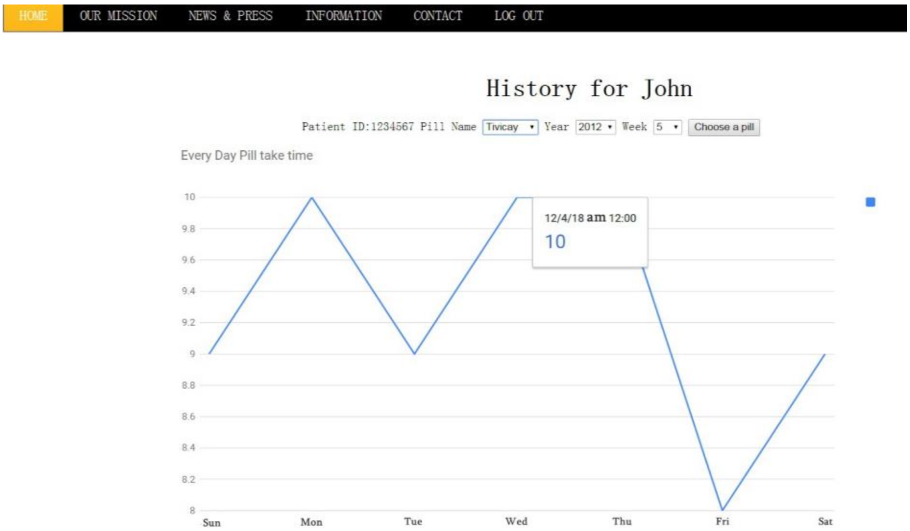
Patient ID:1234567

Picture 2.3.2

The color in the pictures above reflects the information of taking pills status, the deeper the color is the more pills the patient has forgotten to take. If you hover on the slot, it will display the actual data on it.

2.4 Weekly Taking Time report

This page enables our user browser the taking pill time of each days in a week, so that the doctor can monitor the consistency of taking pills, and find some relation between them.



History for John

Patient ID:1234567 Pill Name Tivicay Year 2012 Week 5

Picture 2.4.1

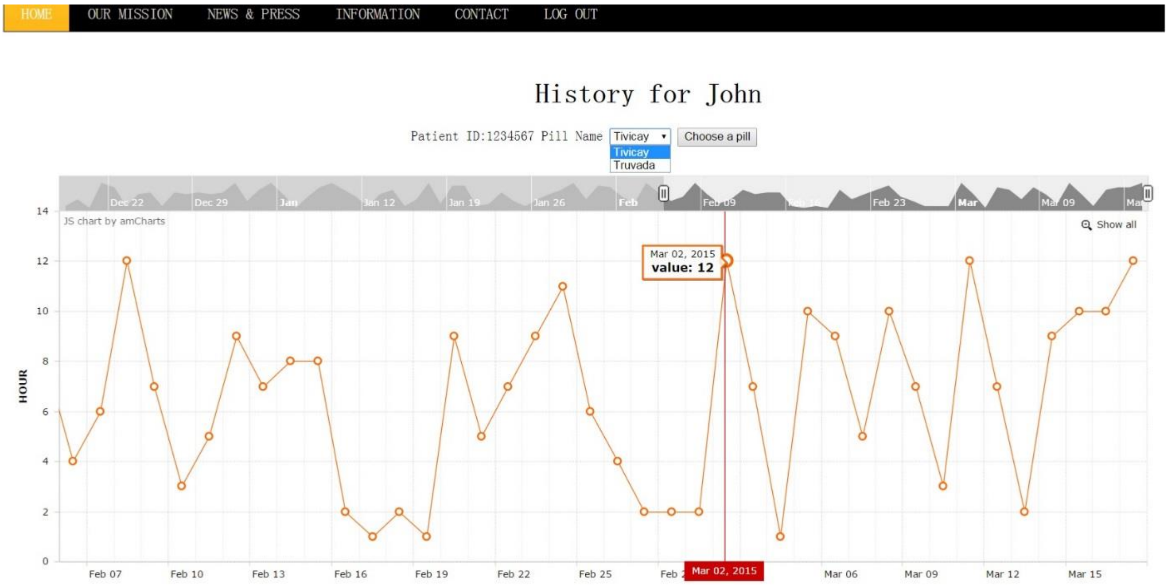
In this page, users need to choose the pill, the year, and the week of the year he want to see, then click the button, a line chart will be shown. The Y-axis is the first taking pill time in that day in X-axis.

We are using google JavaScript API here. This chart is called “visualization.line”, you can find the tutorials in:

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

2.5 Last Three Month Taking Time report

This page is an evolution of the weekly taking time report, because usually the last three months records is more useful. In this page, the user needs to choose a pill, and the page will show the last three months report from today that relate to the user and chosen pill.

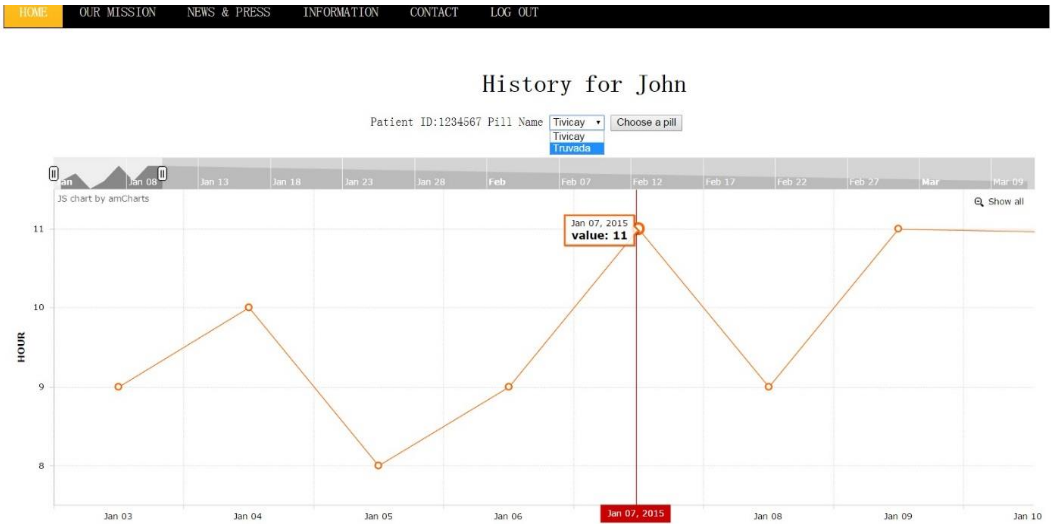


History for John

Patient ID:1234567 Pill Name Tivicay

Picture 2.5.1

In this chart, we designed to give users a dynamic chart, initially the chart will show a 15 days records, because some time the taking time is different showing only a limited date range will be more useful. Then if the user wants to see more records in a big view, he can just move the scroll in the upper bar, move forward the left one, or move backward the right one to select the date range.



History for John

Patient ID:1234567 Pill Name Tivicay

Picture 2.5.2

Moreover, if you want to see the very details of the record, you can zoom in and out the chart with mouse wheel, and if you hover your mouse on the point, it will tell you the date and value, also will highlight the date in the bottom. This chart is implemented by a more open-source amchart, unlike google, amchart give developer the opportunity to customize their own chart.

Also, we need let PHP model layer query the data and controller layer process the data. We query the time and date from database photo table:

```
SELECT DISTINCT ON (image_timestamp::DATE) image_timestamp::DATE, EXTRACT(HOUR
FROM image_timestamp::TIME) from photo where patient_id = '$PatientID' and pill_id
in (select pill_id from pill where pill_name= '$Selected_pillname') and
image_timestamp >= ' $startTimeThreeMonths' and image_timestamp <= ' $nowTime';
```

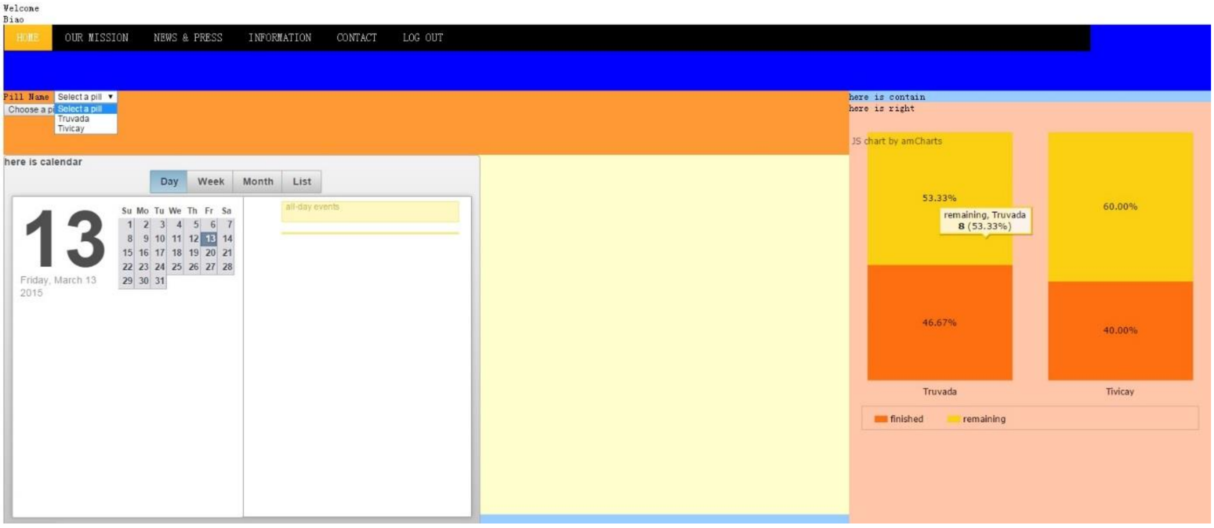
Then create a PHP array, PHP array is like a hash table, it provide the key value matching function which make things easier. We use this hash table to store time and date for displaying. Another thing need to be mention is the transformation from PHP time to JavaScript Time:

```
$displayDay = strtotime($actualDates[$displayIndex]);
$displayDay=date($displayDay*1000 + 24*60*60*1000);
```

PHP uses unix time, that count the seconds from the utc first day, but JavaScript uses different way to calculate time, which makes things so wired when we first plot out the chart.

2.6 Pill Taking Progress Bar

To make the pill taking progress available, we added a series progress bar in the personal view home page (The page just after logged in).



Picture 2.6.1

As what is shown in picture 2.6.1, we added the bar on the right part to show the percentage of how many pills have been taken, how many are remaining, for each pill that correspond to the current logged in user.

The orange part is the finished pill amount, and the yellow area shows the pills which are reaming in the bottle. If you hover your mouse on any of these area, a detail information will display saying the exact number of pills.

We are querying only the date period of the pill bottles that are corresponded to the current date, that is to say: if currently this patient is having Truvada, we query the prescription whose start time, and stop time includes the current date. Also we count the taking times from the Photo table that only in that time period.

As described above, we have two queries, the first is:

```
SELECT *,(start_time::DATE),(stop_time::DATE) from prescription pp, pill pi where
pp.pill_id = pi.pill_id and patient_id='$PatientID' and stop_time >= ' $nowTime'
and start_time <= ' $nowTime';
```

And the query for counting in Photo is:

```
SELECT count(*) from photo where patient_id = '$PatientID' and pill_id
='$pillidarray[$index]' and image_timestamp >= '$pillStartTime[$index]' and
image_timestamp <= '$pillStopTime[$index]';
```

We calculate the taking pill days by using the milliseconds level representation of date:

```
$startDay = strtotime($pillStartTime[$index]);
$endDay = strtotime( $pillStopTime[$index]);
$continueDays = $endDay - $startDay;
$continueDays = $continueDays/(60*60*24);
$pillAmount = $continueDays * $pillTimes[$ii];
$pillRemaining[$ii] =  $pillAmount -  $pillFinished[$ii];
```

And multiple the days with the times that the patient should take per day, is the amount of pills in that bottle.

For the chart, we need only two layers which are remaining and finished, we defined and created them in JavaScript:

```
"graphs": [{
            "balloonText": "[[title]], [[category]]<br><span style='font-
size:14px;'><b>[[value]]</b> ([[percents]]%)</span>",
            "fillAlphas": 0.9,
            "fontSize": 11,
            "labelText": "[[percents]]%",
            "lineAlpha": 0.5,
            "title": "finished",
            "type": "column",
            "valueField": "finished"
        }, {
            "balloonText": "[[title]], [[category]]<br><span style='font-
size:14px;'><b>[[value]]</b> ([[percents]]%)</span>",
            "fillAlphas": 0.9,
            "fontSize": 11,
            "labelText": "[[percents]]%",
            "lineAlpha": 0.5,
            "title": "remaining",
            "type": "column",
            "valueField": "remaining"
        }],
```