Construction of Social CRM System based on WeChat Public Platform

Linjun Sun

School of Management, North China Institute of Science and Technology, Hebei, 065201, China

Keywords: WeChat public platform; CRM; Social CRM; MVC.

Abstract. The technical solution of Social CRM System was designed based on features of WeChat public platform. The main functions of the system were realized by utilizing Code Igniter framework, following an mvc pattern. The system is easy to be used and maintained. For enterprises, the application of the system can bring one-on-one marketing and service and rapid marketing information dissemination. It not only cuts down their cost on marketing but deepens interaction with customers and improves customer satisfaction.

1. Introduction

With the fast development of social media and mobile terminals, the traditional CRM is gradually evolving into Social CRM. Social media, like WeChat and microblog, has smashed the mode of communication and helped enterprises a lot in following up potential customers and improving their customer satisfaction. Now WeChat is becoming the most popular social apps because of its convenience and social functions. Since WeChat public platform was launched, many enterprises provide more personalized service to customers on this platform and WeChat has become their channels connecting with customers [2, 3]. Due to this, WeChat-based Social CRM solution comes into being. It can provide enterprises effective communication with their customers for the features of instant messaging, low cost and high precision of WeChat. Targeting on small and medium sized enterprises, a Social CRM solution based on WeChat public platform is proposed to meet their requirements of customer management. It can help them to rapidly establish deep connections with customers on marking and communication services at a low cost and bring them new business opportunities.

2. System function design

WeChat has released a series of APIs to encourage businesses to develop custom features for their accounts and the Social CRM system is developed on the APIs. After a WeChat user sends a message to the WeChat public platform server by HTTP protocol, the WeChat server will deliver it to a third-party message server which can receive and analyze messages. Next, the server will get a response message from the enterprise’s CRM application based on the type and content of the original message, pack it in the form of XML and send it to the WeChat server, which will finally reply the user. The interaction flow of the system is shown in Figure 1.

On the foundation of the analysis on the requirements of enterprises and customers and combining with CRM system’s common functions, the WeChat-based CRM system composes of client management, service management, marketing management and statistic analysis. The function of the system is shown in Figure 2.

Customer management can call the group management API to establish different groups and push messages accordingly. User's information and user locations can also be obtained from the user profile via Web API and user location API, respectively.

Customer service management makes full use the feature of one-on-one communication of WeChat and makes it convenient for customers to connect with enterprises at any time. It can be realized through calling user-defined menu and call back messages API.

Marketing management can make customers to attend their marketing campaigns through calling callback messages API, user-defined menu API and user location API. For example, people can send messages to check in a marketing activity or clicking relevant menu to enter the marketing webpage.

Statistical analysis can collect and analyze customers' information through the data such as the number of a message be read and shared, and therefore to make enterprises to push messages effectively and find their potential clients.

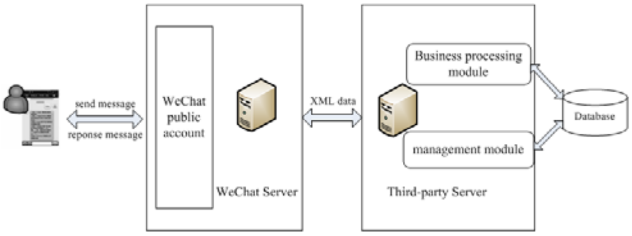

Figure 1. System architecture design

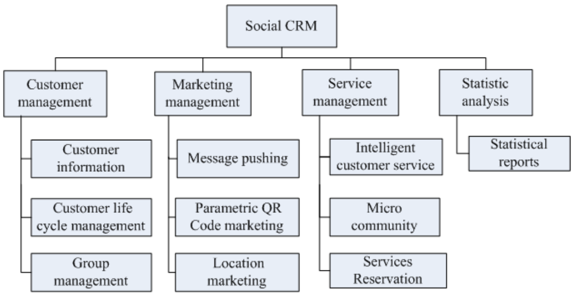

Figure 2. Functions of the system

3. Key technologies and realization

The server-side of the system is developed by utilizing CodeIgniter framework. CodeIgniter is a proven, agile & open PHP web application framework. It is a lean MVC framework. The development concentrates on making seamless connectivity between the WeChat APIs and the application, with the controller as the core which is responsible for major functions like receiving and processing messages.

3.1 Receiving messages

The service that the system can provide is presented by message interaction on Wechat. The third-party message server is responsible for analyzing POST request from Wechat server, matching the message type and order and utilizing corresponding subsystem to process.

Message authentication. When the third-party message server receives a message from a user, it need to authenticate that the request is from WeChat by checking the signature firstly. If the request is verified, the third-party message server then processes the request message. The verification code is as follows:

```
public function valid(){
$echoStr = $_GET["echostr"];
if($this->checkSignature()){
echo $echoStr;
exit;}
}
private function checkSignature(){
$signature = $_GET["signature"];
$timestamp = $_GET["timestamp"];
$nonce = $_GET["nonce"];
$this->load->model(MODUAL_NAME.'/set');
$token = $this->set->getOneSet('token');
$tmpArr = array($token, $timestamp, $nonce);
sort($tmpArr);
$tmpStr = implode( $tmpArr );
$tmpStr = sha1( $tmpStr );
if( $tmpStr == $signature ){ return true;}
else{ return false;}
}
```

Message processing. The controller processes received messages and is responsible for business logic processing. It will call different service modules according to the message type. The code for message processing is as follows:

```
public function responseMsg () {
$postStr = $GLOBALS ["HTTP_RAW_POST_DATA"];
If (empty ($postStr))  Return;
$postObj = simplexml_load_string ($postStr, 'Simple XMLE lement', LIBXML_NOCDATA);
// getting the message data from the XML data package
$fromUsername = $postObj->FromUserName;
// judging the type of a message and then call the corresponding function
Switch ($postObj->MsgType) {
//event message
Case 'event': Return $this->handlerEvent ($postObj); break;
// text message
Case 'text': Return $this->handlerText ($postObj); break;
//location data message
Case 'location': Return $this->handlerLocation ($postObj); break;}
}
```

If a user follows or unfollows an official account, or click the user-defined menu, the controller can deliver corresponding messages or unlink the account. Program fragments are as follows:

```
function handlerEvent($obj) {
$fromUsername = $obj->FromUserName;
$toUsername = $obj->ToUserName;
$info = $this->getUserInfo($fromUsername);
$this->load->model(MODUAL_NAME.'/fan');
$isExist = $this->fan->isExist((string)$fromUsername);
if(!$isExist){
      $this->fan->addFans((string)$fromUsername, $info->nickname, $info->sex,
$info->province, $info->country);}
// following event
if($obj->Event=="subscribe"){
return $this->getFollowResponse($fromUsername, $toUsername); }
//unfollowing event
 else if($obj->Event=="unsubscribe" ) {
$this->fan->delFans((string)$fromUsername);    }
//user-defined menu click event
else if($obj->Event=="CLICK"){
$content = $obj->EventKey;
self::textRespont($fromUsername, $toUsername,$content);
$reponse = $this->getResponse($fromUsername, $toUsername, $content);
if ($reponse) { return $reponse; }
}
}
```

3.2 Messages Reply

After the received messages are processed, the controller returns reply messages in a specific XML structure in the response packet [1]. The reply message can be a text or image one. Each reply message has information such as receiver's WeChat ID, sender's WeChat ID, message creation time,etc. In the following code, if the server returns some image messages, before they are send to the WeChat server, they will be converted into XML data package by returnPic() function firstly.

```
function returnPic ($fromUsername, $toUsername, $picArray) {
$time = time ();
$count = count ($picArray);
$text = "<xml>
<ToUserName><! [CDATA [$toUsername]]></ToUserName>
<FromUserName><! [CDATA [$fromUsername]]></FromUserName>
<CreateTime>$time</CreateTime>
<MsgType><! [CDATA [news]]></MsgType>
<ArticleCount>$count</ArticleCount>
<Articles>";
Foreach ($picArray as $key=>$value) {
$pattern = '/^http/';
$url = preg_match ($pattern, $value ['picurl'])? $value ['picurl']: base_url (). $value ['picurl'];
$text. = "<item>
<Title><! [CDATA [". $value ['title']."]]></Title>
<Description><! [". $value ['description']."]]></Description>
<PicUrl><! [CDATA [". $url."]]></PicUrl>
<Url><! [CDATA [". $value ['url']."]]></Url>
</item>";
}
$text. = "</Articles></xml>";
Return $text;
}
```

3.3 User-defined menu

The server returns a specific data structure in the response packet for POST request and a user-defined menu can be created on the WeChat Client. A user-defined menu can include up to three first-level menu items, and each of them can include five second-level sub-menus. The user-defined menu API can enables two button types: Click and View. Clicking a clickable button causes the WeChat server to push the click event to the third-party server through an event-type message API. Clicking a view-only button causes the WeChat client to open the URL on the button [B. Jiang, 2013]. The following code shows how the customerized menu is created.

```
function createMenu(){
$this->load->model(MODUAL_NAME.'/menuOrder');
$menu = $this->menuOrder->getCategory(1);
$i = 0;
foreach($menu as $key=>$value){
if($i == 3)  break;
$menu_arr['button'][$i]['name'] = urlencode($value['menu_name']);
if(isset($value['son'])){ //judge  whether there are  sub-menus
$n = 0;
foreach($value['son'] as $val){
if($n == 5) break;
$menu_arr['button'][$i]['sub_button'][$n]['name']=urlencode($val['menu_name']);
      $statu = preg_match('/(http|https):\/\//',$val['outurl']);
$style = $statu ? 'view' : 'click';
$s = $statu ? 'url' : 'key';
$menu_arr['button'][$i]['sub_button'][$n]['type'] = $style;
$menu_arr['button'][$i]['sub_button'][$n][$s]=urlencode($val['outurl']);
$n++;         }
}else{
$statu = preg_match('/(http|https):\/\//',urlencode($value['outurl']));
$style = $statu ? 'view' : 'click';
$s = $statu ? 'url' : 'key';
$menu_arr['button'][$i]['type'] = $style;
$menu_arr['button'][$i][$s] = urlencode($value['outurl']); }
$i++; }
//converting the menu data into JSON response
$str = urldecode(json_encode($menu_arr));
//sending the menu data to the Wechat server
$return = post_method($str,'/menu/create');
…
}
```

4. Security design of application

The code security of the appliction is mainly based on the following aspects:

i) SQL injection defense. SQL injection can be used to gain illegal control over websites. In the application, all the received messages is processed by the function request () to remove malicious code. The code for SQL injection defense is as follows:

```
function request($name){
$value = isset($_GET[$name]) ? $_GET[$name] : '';
if(isNull($value)){
$value = isset($_POST[$name]) ? $_POST[$name] : '';
//processing the data similar to SQL injection code
$value = str_replace("'", "''", $value);
return $value;
}
```

ii) Using session to store user login information and setting session lifetime to avoid cookies attack. The following code can be used to set the max lifetime.

```
ini_set('session.gc_maxlifetime',3600);   // set the max lifetime to 1 hour
```

iii) User verification is required before important data being processed. Anonymous user will be forced to jump to the login page.

iv) Limiting size and format of the files allowed to upload. Specifying the upload directory and setting read-only permission for all the files in the directory.

5. Conclusions

Through secondary development, the WeChat-based Social CRM system has realized messages management, messages reply,subscribers management, etc. With the help of the features of WeChat such as high arrival rate, high precision, highconvenience features, the system help businesses to rapidly spread the message about their products and services and maintain close communication and interaction with customers and provide an important channel to manage customer relationships.

Acknowledgements

The research work was supported by “the Fundamental Research Funds for the Central Universities” under Grant No.3142014105.

References

[1] B. Jiang, Y.M. Qin. Design and Implementation of Mobile Library APP Service System Based on WeChat. Journal of Modern Information, 33(6), p. 41-44, 2013.

[2] L. Zhou, D.Y. Dong. Construction of Library Information Service System Based on WeChat Public Platform. Journal of Modern Information, 34(8), p. 95-99, 2014.

[3] L.Y. Xia, L.F. Han & C. Q. Wang. Building mobile publishing platform of campus information with WeChat Masses platform. Information Technology, 2, p. 183-189, 2014.

[4] Wechat official account admin platform, admin.wechat.com/wiki/index.php