Lorem Ipsuma CRM System

Documentation

1) Overview

Lorem Ipsuma CRM system is a custom full-feature CRM system developed for a Marketing Startup based in Istanbul, Turkey from November 2016 to January 2018. Thanks to its well-designed architecture, this CRM system is a highly extensible and maintainable software system that can be easily and quickly adapted to the new and changeable requirements of a marketing company in a very competitive business environment. Its architecture is completely object-oriented and it has been fully developed based on MVC design pattern. Its core is based on TPS framework, a PHP framework which has been developed for implementing extensible, flexible, and highly customizable business applications.

This CRM system can handle many complexities including but not limited to Localization and Multi-Language Usage, Landing Page Integration, S2S Postback Integration with the Referrers, API based Integration, Affiliates Accounts and Management, Requests and Orders Management, Customers Management, Shipping Management, Agents Management, Contacts Management, Conversion Policies, Users Management, various types of Reports Generation Tools, Flexible Access Control Management, Events Handling and Reports for Orders, Customers, Users, and Products, and non-functional features including Scalability, Reliability, and Security.

Leads are generated in the system either directly through the integrated landing pages or are entered into the system by other third parties through the system API. Generally each lead consists of customer full name and phone number. The generated leads can be approved by administrators and converted to orders, and then the created orders can be assigned to different agents in order to start processing them.

Customers are separated from orders in the system. Each customer can have more than one order over the time, and a customer can be assigned to an order. When agents start processing a new order, they can use the “Find Customer” tool which helps them to find the customer of the order among all existing customers based on the lead full name and phone number. If there is already such a customer in the system, they can assign it to the order, and if there is not that customer information in the system, they can create it as a new customer. Customer profile information can be saved in the system in multiple languages that means each customer can have more than one profile in different languages. Each customer can have more than one contact information. Contact information can also be multi languages.

When an agent gets the approval of customers for their order so that the order items can be sent to the customers, if the shipping information of the customers does not already exist in the system, the agent has to create them as a new contact for them. Then the agent has to select one contact from the customer’s contacts for shipping purpose, the contact which has been approved by the customer. Then that contact will be assigned to the order and submitted to the shipping department for shipping related processes.

The quality control and shipping managers can check and inspect the orders approved and submitted by the agents for shipping, and if they can approve them, then the orders will be sent to the shipping queue in the system. In a predetermined time during each working day, the excel export of the orders existing in the shipping queue are created by the shipping manager and it will be sent to the shipping company, and then the status of those orders will be changed to shipped. The system can automatically get the shipping status of the shipped orders by connecting to the server of the shipping company through its corresponding API.

Then the final status of the shipped orders whether it is delivered or undelivered can be automatically obtained and set by the system.

Products are also created in the system as multi language entities and they can be organized in hierarchical groups. Product groups are also created in the system as multi language entities and they can be organized in a hierarchy. Multi language entities can have more than one profile in different languages. Customers and contacts as mentioned above are also multi language entities. Departments which can be assigned to the users of the system are also multi language entities which can be organized in a hierarchy. It is also possible to attach files and images to the profiles of customers, products, product groups, and departments. The attached images might be automatically cropped and optimized by the system based on the system configuration parameters before they are saved in the system.

For each order, the agents can easily find a product from a list to add it to that order. They can add more than one product with any amount to each order. The final subtotal of the order will be automatically computed by the system based on the price of the products. But the agents can also manually set the subtotal and discount of the order, and then the total of the order will be automatically computed by the system. The price of the products and the value of orders can be easily viewed and processed in multiple currencies. The main report generation tool will allow changing the currency in order to see the revenue and cost results in various currencies.

The web address of the integrated landing pages and also the address of the API leads are stored in the system if they are sent by the clients. If the lead requests contain some new referrer information, that referrer information will also be stored in the system. The system can generate various types of reports for every referrer or landing page.

Affiliate entities can be created in the system, and then some existing landing page addresses in the system can be assigned to them. For each landing page of an affiliate, a payout value can be set in any supported currency. That payout will be paid to that affiliate for each conversion occurring for the leads come from that landing page. Then the system can automatically compute the total revenues of the affiliates in various time periods and also separately for each landing page assigned to them.

User accounts can be created in the system in one of the two classes “Employee” and “Partner”. Also, a user group has to be assigned to each user that determines what information the user is authorized to access and what actions is authorized to do in the system. User profiles are also multi languages, and each user can have more than one profile in different languages. Also each user can have more than one contact information in different languages. Also users can attach some files and add their photo to their profiles. Some landing pages, a referrer, or some affiliate entities can be assigned to each partner account, and then that partner user can only see the leads and orders come from the assigned landing pages, or come from the landing pages of the assigned affiliate entities, or leads and orders referred by the assigned referrer. The partner users to whom some affiliate entities have been assigned can see their revenues and statistics in various time periods.

All events of any type including creation, change, and deletion in orders, customers, products, and users are recorded in the system and can be easily accessed and reviewed at any time in the future. So in the system always it is clear which user at what time has caused which changes in which data fields of orders, customers, products, and users. Also “Check in” and “Check out” actions have been implemented in the system to prevent editing orders or other entities such as customers and products by more than one user at the same time.

An almost full localization is supported by the system including languages, currencies, countries, zones, cities, and districts or areas. The system is multi language at its core, and new languages can be easily added to the system. New countries, zones, cities, and districts can be added to the system to be easily used in creating the contact information of customers or users in the system. New currencies can be added to the system to be easily used for setting the price of products and the value of orders.

This CRM System has been developed based on TPS Framework that is a highly extensible and maintainable software development framework. The system is highly scalable and has a good performance considering the large amount of data which has been being processed and stored in the system during about one year after the marketing company started to use this system.

2) Integration Guide

This Integration guide explains three types of integration with Lorem Ipsuma CRM system:

I. **S2S Postback Integration**
II. **API based Integration**
III. **Landing Page Integration**

2-1) S2S Postback Integration

Lorem Ipsuma CRM System supports S2S Postback Integration with referrers' systems. This CRM System is able to store three web request or referral parameters of the referrers when leads are generated by submitting a form on a landing page and that landing page has been configured to get the values of those parameters from the requests and submit them to the CRM system. The supported referral parameters are as follows:

1. **Transaction ID:** This is normally the unique id generated by the referrer for each referral link or request.
2. **Offer ID:** This is normally the unique id generated by the referrer for each offer representing a product which is promoted by the advertiser on some web pages including a landing page.
3. **Affiliate ID:** This is normally the unique id generated by the referrer for each of its affiliate publishers. These publishers or affiliates are responsible to create and send traffic towards the web pages of the advertiser for lead generation.

Lorem Ipsuma CRM System is able to store the names of those parameters for each referrer as specified and standardized by them to be used for sending traffic to the web pages of the advertiser. For example, transaction id parameter can be represented by different names such as transaction_id, click_id, or clickID by different referrers. This CRM system will store the names of those parameters separately for each referrer.

Also the core postback url or core conversion notifying url of each referrer will be stored in the system to be used for notifying the corresponding referrer automatically when a conversion occurs, or in other words, when generally the status of a lead is changed to sale. When a conversion occurs, the system automatically recognizes the referrer of that order, and creates a notifying or postback url based on the referrer’s core postback url and the aforementioned referral parameters, and send a request to the postback url to notify the referrer of the conversion. In this process, the referral transaction ID, offer ID, and affiliate ID of the order are sent to the referrer’s system to enable or help the referrer to recognize that referral transaction, and its corresponding offer and affiliate.

Normally a conversion occurs when the status of an order is changed from **Quality Control** to **Shipping Queue** (For the meaning of all the possible order statuses please refer to the next section i.e. API based Integration). The conversion information of all converted orders are static and unchangeable that means it is not possible that a converted order to be changed to non-converted order or its conversion time to be changed in the future. But the status of orders can be dynamic and frequently changed over the time.

2-2) API based Integration

There are two methods for interacting with Lorem Ipsuma CRM API, namely “lead” and “enquiry”. “lead” method can be used for submitting lead or order data to the CRM, and “enquiry” can be used for enquiring the status of the orders from the CRM. Both types of requests must be sent to the CRM API at http://[crm-domain]/api.php. The only difference is that “lead” requests must be sent by HTTP POST method, but “enquiry” requests can be sent by HTTP POST or GET methods both. In both cases, the type of request must be specified by “task” parameter. The task=lead request parameter will cause that the request to be processed as “lead” request, and the task=enquiry request parameter will cause that the request to be processed as “enquiry” request.

In both types of requests, the format and language of the response can be specified by “format” and “api_language” parameters respectively. The format=xml request parameter will cause that the response to be in xml format, and the format=json request parameter will cause that the response to be in json format. The api_language=en-GB request parameter will cause that the messages of the response to be in English language, and the api_language=tr-TR request parameter will cause that the messages of the response to be in Turkish language.

In both types of requests, the “ref_address” and “ref_key” parameters must be specified. “ref_address” parameter is normally the web address or domain of the referrer that sends that request, without any scheme, subdomain or path parts, for example “adcombo.com”. “ref_key” is a static api key or password which is exchanged between the referrer system and the CRM system for validating the requests.

Generally, the CRM API responses can have four major fields:

1) status: It informs whether the request processing was successful or not. Its value is either “success” or “error”.

2) message: It usually contains a description of the result of the request processing.

3) error: If the request processing was not successful, then this field will appear and contain the error trace in processing the request.

4) data: In “enquiry” requests, this field contains the orders’ data resulted from the enquiry.

❖ “lead” Request

In “lead” requests, in addition to the aforementioned parameters, the following parameters have to be specified:

1. **full_name:** The full name of the customer. (Required)
2. **phone:** The phone number of the customer. (Required)
3. **product_id:** The id of the product in the CRM system. The referrer will be informed of the valid product ids. (Required)
4. **ref_lead_id:** The id of the lead or order in the referrer’s system. This lead id can be then used in “enquiry” requests for enquiring the status of the order. (Required)
5. **customer_note:** The customer’s note or the referrer’s note for this order. This parameter can be important in some types of orders where the

customers have to specify their requirements or required products with more details. (Optional)

6. **address:** The web address of a sample web page promoting that product. It must be a simple domain name like www.example.com. (Optional)
7. **api_key:** The CRM System API Key that must be used usually when a new referrer address (i.e. ref_address parameter) or a new landing page address (i.e. address parameter) is submitted to the CRM system. For repeated ref_address or address values, using api_key is not necessary unless it is forced by the CRM system to only accept leads coming with a valid API Key. Usually the CRM system does not force using api_key in every lead requests. But using api_key in lead requests which contain a new ref_address or address value is necessary.

- **Sample responses in XML format:**

```
<response>
  <status>success</status>
  <message>Lead saved successfully</message>
  <request_id>47660</request_id>
</response>
```

```
<response>
  <status>error</status>
  <message>Failed to save the lead.</message>
  <error>
    <![CDATA[
<div class="exceptions"><div class="warnings"><div
class="warning">Warning 1300: LI_Model::saveRequest =>
Invalid Referrer Key!</div></div></div>
    ]]>
  </error>
</response>
```

- **Sample responses in JSON format:**

```
{"status":"success","message":"Lead saved
successfully","request_id":47661}
```

```
{"status":"error","message":"Failed to save the
lead.","error":"<![CDATA[<div class=\"exceptions\"><div
class=\"warnings\"><div class=\"warning\">Warning 1300:
LI_Model::saveRequest => Invalid Referrer
Key!<\/div><\/div><\/div>]]>"}
```

❖ "enquiry" Request

In "enquiry" requests, in addition to the aforementioned parameters, the following parameters have to be specified:

1. **ref_lead_ids:** The ids of the being enquired leads or orders in the referrer's system separated with comma ,. (Required)

The API response contains the status and conversion information of the enquired orders. For an order, conversion=1 means the order has been converted to sale according to the conversion policy of the CRM system, and conversion=0 means the order has not been converted to sale. Also the conversion_time field will contain the conversion date and time of the order. Also the response will contain the other date time information of the enquired orders including request_time, call_later_time, called_time, approved_time, shipped_time, and delivered_time if they have been set for those orders, otherwise empty or null date time i.e. '0000-00-00 00:00:00' will be returned.

Normally a conversion occurs when the status of an order is changed from **Quality Control** to **Shipping Queue**. The conversion information of all converted orders are static and unchangeable that means it is not possible that a converted order to be changed to non-converted order or its conversion time to be changed in the future. But the status of orders can be dynamic and frequently changed over the time.

- **Possible Order Statuses and their meanings:**

1) **Waiting:** The lead is still waiting for processing.
2) **Trash:** The lead is not a valid lead usually because of its invalid customer name or phone number, or because it contains repeated data.
3) **Process:** The processing of the lead has just been started.
4) **Lead:** The lead is considered as a normal lead and can be called by the agents.
5) **Invalid:** The lead is considered as an invalid lead and may not be called by the agents.
6) **Unreachable:** The lead phone number was called, but it is unreachable.
7) **Canceled:** The lead phone number was called, but the customer canceled their order.
8) **Suspicious:** The order is considered as suspicious, and has to be investigated more.
9) **Call later:** The customer was called, but either the customer or line is busy, or the customer did not answer, or has asked for calling them in a later time.
10) **Called:** The order’s customer was called, but the result has not still been specified by the agent.
11) **Approved:** The agent has approved the order because the order’s customer has approved their order.

12) **Quality Control:** The agent has approved the order, and the order is in the quality control process.
13) **Shipping Queue:** The quality of the order has been approved, and the order is waiting in the shipping queue that its items to be shipped to its corresponding customer.
14) **Shipping Company:** The order information has been sent to the shipping company.
15) **Shipped:** The shipping process has been started for the order.
16) **Delivered:** The order products have been delivered to its corresponding customer.
17) **Undelivered:** The order products could not be delivered to its corresponding customer.
18) **Rejected:** The order products were rejected by its corresponding customer in the delivery time.
19) **Refund:** The customer has asked for the refund, and the order has been refunded.
20) **Unknown:** The status of the order is unknown, usually because there is not such an order in the system, or that order does not belong to the requester referrer.

Note:

Please take it into account that you will get the lead or order statuses in lower case without any space between their characters. For example, the Quality Control status will be sent to you as “quality_control”. You might need to create a function in order to convert our order statuses to those of yours according to your own business language.

- **Sample responses in XML format:**

```
<response>
   <ref_key>ddddddddddd-oooooooo</ref_key>
   <status>success</status>
   <message>The status of the orders enquired successfully</message>
   <data>
      <item>
         <id>Nak16DY3</id>
         <info>
            <status>trash</status>
            <conversion>0</conversion>
            <request_time/>
            <call_later_time/>
            <called_time/>
            <approved_time/>
            <conversion_time/>
            <shipped_time/>
            <delivered_time/>
         </info>
      </item>
      <item>
         <id>o0nrwE0d</id>
         <info>
            <status>shipping_queue</status>
            <conversion>1</conversion>
            <request_time>2017-12-19 05:29:25</request_time>
            <call_later_time>0000-00-00 00:00:00</call_later_time>
            <called_time>2017-12-19 10:27:40</called_time>
            <approved_time>2017-12-19 10:29:41</approved_time>
            <conversion_time>2017-12-19 10:36:38</conversion_time>
            <shipped_time>0000-00-00 00:00:00</shipped_time>
            <delivered_time>0000-00-00 00:00:00</delivered_time>
         </info>
      </item>
      <item>
         <id>p0ZBYA0l</id>
         <info>
            <status>invalid</status>
            <conversion>0</conversion>
            <request_time>2017-12-15 21:46:55</request_time>
            <call_later_time>0000-00-00 00:00:00</call_later_time>
            <called_time>2017-12-19 09:47:18</called_time>
            <approved_time>0000-00-00 00:00:00</approved_time>
            <conversion_time>0000-00-00 00:00:00</conversion_time>
            <shipped_time>0000-00-00 00:00:00</shipped_time>
            <delivered_time>0000-00-00 00:00:00</delivered_time>
         </info>
      </item>
   </data>
</response>
```

```
<response>
  <ref_key>dddddddddddd-oooooooo</ref_key>
  <status>error</status>
  <message>Failed to enquiry the status of the orders.
  </message>
  <error>
    <![CDATA[<div class="exceptions"><div
class="warnings"><div class="warning">Warning 1300:
LI_Model::enquiryOrderStatus => Invalid Referrer
Key!</div></div></div>
     ]]>
  </error>
</response>
```

- **Sample responses in JSON format:**

```
{"ref_key":"dddddddddddd-oooooooo","status":"success",
"message":"The status of the orders enquiried successfully",
"data":{"Nak16DY3":{"status":"trash","conversion":0,
"request_time":null,"call_later_time":null,"called_time":null,
"approved_time":null,"conversion_time":null,"shipped_time":nul
l,"delivered_time":null},"o0nrwE0d":{"status":"shipping_queue"
,"conversion":1,"request_time":"2017-12-15 12:22:12",
"call_later_time":"0000-00-00 00:00:00","called_time":"2017-
12-15 12:25:45","approved_time":"2017-12-15 12:28:12",
"conversion_time":"2017-12-15 12:33:24","shipped_time":
"2017-12-15 12:58:11","delivered_time":"2017-12-18 16:41:40"},
"p0ZBYA0l":{"status":"invalid","conversion":0,"request_time":
"2017-12-15 21:46:55","call_later_time":"0000-00-00 00:00:00",
"called_time":"2017-12-19 09:47:18","approved_time":
"0000-00-00 00:00:00","conversion_time":"0000-00-00 00:00:00",
"shipped_time":"0000-00-00 00:00:00","delivered_time":
"0000-00-00 00:00:00"}}}
```

```
{"ref_key":"dddddddddddd-
oooooooo","status":"error","message":"Failed to enquiry the
status of the orders.","data":[],"error":"<![CDATA[<div
class=\"exceptions\"><div class=\"warnings\"><div
class=\"warning\">Warning 1300: LI_Model::enquiryOrderStatus
=> Invalid Referrer Key!<\/div><\/div><\/div>]]>"}
```

2-3) Landing Page Integration

- ✓ Upload the “client.php”, “setup.php”, “post.php”, and “thanks.html” files, and also the languages folder to the root directory of the landing page
- ✓ Add the code of “call-me-form.php” to the landing page, and give it the desired HTML structure if needed
- ✓ Change the configuration parameters in “setup.php” if needed
- ✓ Edit the “After call me form submission” section in “post.php” if needed
- ✓ Edit the “thanks.html” file if needed

The following inputs are supported by this CRM system for generating a lead:

1. **full_name:** The full name of the customer. (Required)
2. **phone:** The phone number of the customer. (Required)
3. **address:** The web address of the landing page or a sample web page promoting that product. It must be a simple domain name like example.com, otherwise it will be converted to a simple domain name by the system itself. (Required)
4. **customer_note:** The customer’s note or the content of the lead generation form submitted by the customer on the landing page. This parameter can be important in some types of orders where the customers have to specify their requirements or required products with more details. (Optional)
5. **product_id:** The id of the product in the CRM system. (Optional)

2) Requests and Orders Management

Not ready yet …

3) Customers and Contacts Management

Not ready yet …

4) Shipping Management

Not ready yet …

5) Affiliates and Referrers Management

Not ready yet …

6) Products and Product Groups Management

Not ready yet …

7) Users Management

Not ready yet …

8) Report Generation Tools

Not ready yet …

9) Settings and System Configuration

Not ready yet …