Case Study
Web Based Affiliate Network System
Client Requirement

Sale Servant is a web application with an aim to affiliate network solution for small companies. Sale Servant is a platform aimed at small to medium sized designers, fashion brands and online retailers. It helps to drive traffic, increase sales and provide the tips and knowledge to build your brand and sell more of your fashion products.

Sale Servant is low-cost affiliate solution to drive more traffic to retailer’s website from various publishers’ sources. The publisher can earn % of sales by driving traffic to a particular website.

The Sale Servant mainly has 2 different types of users.

1. Retailer
2. Publisher

The Retailer who want to get traffic and drive sales from the third-party, get register themselves on this platform and manages the various aspect of their account. The Retailer can manage their profile elements like personal information, business information, type of business, country of residence etc. The Retailer can also decide and manage their association with Publishers. The Retailer has to approve the request raised by the Publisher in order to get associated with the Publisher. Retailer requires integrating tracking code on their platform manually or with the use of provided app. After approval and confirmation from admin for tracking code integration, Retailer can be an active part of the platform. Retailers can provide their product feeds on the platform which will be converted to required XML format for Publisher to use on their website or blogs. The Retailer has to verify commission manually in order to consider it as a “Confirmed Sale” done on their platform. Based on the approved transactions, Retailer can pay Publisher their % commission.

Publisher who actually drives traffic and sales to the Retailer’s website, get register themselves on this platform and manages various aspect of their account. The Publisher can manage their profile elements like personal information, business information, banking or pay pal account information etc. The Publisher can also manage their relationship with Retailer and can view transaction reports, click reports and payment reports from their dashboard area.

Challenges

- Incorporating web usability principles while developing numerous features designed for enhancing the users’ experience.

- Security of user’s transactional data is the key challenge of the platform.

- Platform supports multiple currencies for calculating publisher’s and platform commission.

- This was the very first project for Symfony 2.x.

- Rapid application development was the main target behind building this application. As a part of this, we used Twitter Bootstrap Package to manage the UI part.

- Utilizing community available bundles and customize/override them according to the requirement was one of the most challenging part of the development cycle.
- Customization of FOS user bundle for supporting multiple roles along with managing separate dashboard area was key part of the main system technical architecture.

- Commission payment and roll over monthly payment management was technically challenging for the core part of the platform.

- Ensuring the more complex relation between data and their consistency. Each entity or information is being used in various places in conjunction with other information. Showing most relevant information at various places by clear way of presentation is the main challenge as almost all information is similar to the different entities.

### Technologies

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Tools &amp; Environments</td>
<td>PHP5+, Symfony 2.3 Framework, Microsoft Visio, NetBeans IDE, AJAX, Java Script, HTML5, CSS3, SVN etc.</td>
</tr>
<tr>
<td>Database</td>
<td>MYSQL Database Server, DB Clustering, DB Optimization, Master Slave Replication, Query Optimization, Scheduler for Backups</td>
</tr>
</tbody>
</table>

### Man Power

<table>
<thead>
<tr>
<th>Role</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Leader</td>
<td>1</td>
</tr>
<tr>
<td>Developers</td>
<td>2</td>
</tr>
<tr>
<td>Designers</td>
<td>Nil</td>
</tr>
<tr>
<td>Quality Assurance Testers</td>
<td>1</td>
</tr>
</tbody>
</table>
The following four-tier development approach was adopted to equip the site with numerous features and functionality mentioned below:

- The Database layer containing MYSQL Server Database, Tables, Triggers and so on.
- The Data Access layer managed by Doctrine 2.x containing the Data Access DLL responsible for accessing data from the database.
- The Business Logic layer consisting of all the business logic procedures for various modules.
- The User Interface layer which forms the Graphical User Interface of the website. This part/aspect of the platform is built using Symfony’s View Layer and Twitter Bootstrap Package.

Development Highlights

The design approach was built around Symfony 2.3 and MYSQL SERVER 5 due to the affiliate nature of the website. Symfony 2.3 is used in order to follow Rapid Application Development and Do Not Repeat Yourself (DRY), Keep It Simple Stupid (KISS) principle. Symfony 2.x have many community contributed bundles available which helps to make the development really faster, efficient and more reliable for platform building. Symfony 2.x has been used in order to keep platform more secure, stable and easy to maintain. In order to most effectively access the database in an object-oriented context, an interface translating the object logic to the relational logic was used to communicate with the relational databases in an object-oriented manner.

Doctrine 2.0 is used for database interaction layer. An intermediary abstraction layer was created for accessing data from the database. Triggers, Stored Procedures and Custom MYSQL functions were used only for complex calculation of data from multiple tables and were utilized with heavy conditional syntax to ensure the smooth performance of the website. The UI layer was kept free of any business logic with images, applications and data being called from their respective servers.

The web usability guidelines provided and supported by the Twitter Bootstrap bundle/package were strictly followed during development and the interface was made easily navigable through the judicious use of AJAX, CSS3 and HTML5 controls. With respect to UI part, the platform must support any smart phone device or tablet so is the main reason for using Twitter Bootstrap Bundle. The site was developed and fully functional within a span of 3 months.