

**OPEN SOURCE SOA BASED MIDDLEWARE
FRAMEWORK FOR CLASSIFIED BASED WEB
DEVELOPMENT**

Project ID: 17-072

Final Report(Individual)

Bachelor of Science Special (Hons) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

October 2017

**OPEN SOURCE SOA BASED MIDDLEWARE
FRAMEWORK FOR CLASSIFIED BASED WEB
DEVELOPMENT**

Project ID: 17-072

Final Report(Individual)

K.S.D.A Kulathunga – IT14025686

Supervisor: Mr.Nuwan Kodagoda

Bachelor of Science Special (Hons) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

October 2017

DECLARATION

I declare that this is my own work and this report does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

	STUDENT NAME	STUDENT NO.	SIGNATURE
2	K.S.D.A Kulathunga	IT 140 256 86	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Name of supervisor: Mr. Nuwan Kodagoda

Signature of supervisor:

Date:

ABSTRACT

Design and implementation of Core Framework (Middleware framework core) including the restful Service API used by the web developer for classified web development (eg: ikman.lk, craigslist.org) and End Points to External Application (Mobile/Web), integration with other components of the framework, permissions and roles and security of the core framework, routing, services library that gives the developer the tools they need for modern web development. These endpoint include services the web application/site in its core functionality such as displaying the advertisements, user uploading the advertisements, to even user submitting a form will be provided with a uniform, high level API (Applications Programming Interface) to applications. This middleware framework will facilitate evolution, enhance the reusability and as well improve portability to new platforms. The framework includes other features such as routing and data type conversion, error detection and handling.

This middleware framework will save the developers learning curve by providing a common programming abstraction and by hiding low-level details and development time required to complete the project.

ACKNOWLEDGEMENT

TABLE OF CONTENTS

DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
1. INTRODUCTION	1
1.1. Background Context	1
1.2. Research Gap	2
1.3. Research Problem	2
1.4. Research Objectives	3
2.1. Methodology	5
2.1.1. Background Study	6
2.1.2. Requirement Gathering	6
2.1.3. Design	7
2.2. Testing and Implementation	8
2.2.1. Implementation	8
2.2.1.1. Error! Bookmark not defined.	10
2.2.2. Testing	11
2.2.2.1. Module Testing	12
2.2.2.2. Integration Testing	12
2.2.2.3. System Testing	12
2.3. Research Findings	12
3. Error! Bookmark not defined.	
4. REFERENCES	15
5. APPENDICES	16

LIST OF FIGURES

Figure 1: High level view of the complete system	3
Figure 2: High level architecture diagram of the middleware	4

LIST OF TABLES

Table 1: Abbreviations List	ix
Table 2: Existing product comparison table	18

LIST OF ABBREVIATIONS

Database	a structured set of data held in a computer, especially one that is accessible in various ways.
----------	---

API	a set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or another service.
Database Abstraction	A database abstraction layer is an application programming interface which unifies the communication between a computer application and databases such as SQL Server, DB2, MySQL, PostgreSQL, Oracle or SQLite
RESTful	A way of providing interoperability between computer systems on the Internet
Module	each of a set of standardized parts or independent units that can be used to construct a more complex structure, such as an item of furniture or a building.

1. INTRODUCTION

1.1. Background Context

The origin of classified based web development is classified advertising. Classified advertising is a form of advertising which is particularly common in newspapers, periodicals and online. Even today printed classified are exists although the online web format decrease the profitability of those printed media [1]. Online web format of classified service provide wide range of features comparatively to printed media. Advertisement can be longer, searchable even some companies offer free advertising facilities. Due to this flexibility, online web format classified market has become heavily fragmented. Today international range, domestic range even hometown range online classified advertising companies provide their services. Furthermore, there is an increasing emphasis on developing specialized classified websites over general classified websites for vertical markets and niche markets [2].

Solid classified website is a valuable service for different perspectives. Find or sell a product or service, a great profitable business respectively to consumer and company perspectives. Building such site from the scratch is much more complex than its sounds where developers can get frustrated and eventually product might be failed or cost overrun. Since this is a very competitive marketing segment it's very important to build the application with possible minimum time with competitive features with other competitors.

The main goal of this project provides a solid feature rich middleware framework for classified based web development. This middleware handles the complexity of different technology layers, that are involved in the development process.

1.2. Research Gap

Since the Emergence of the Internet, classified ads have moved on from the old-fashioned way of newspapers to the internet, where the competition is now

higher than ever for classified web sites. More than 20+ classified websites are created and used within Sri Lanka itself. These Classified websites generate a huge amount of profit merely from the ads displayed on their websites. New classified websites are still being created in Sri Lanka to potentially capture or part take in the existing online market. These websites require certain web technologies to be implemented in order to create a successful classified web application for the end users. The continuous changes in existing technologies and arrivals of new modern technologies brings a burden to the developers of these classified websites.

A classified web developer have to face the problem of having to learn all the latest technologies from database all the way to authenticating the application needed to create a classified website. This learning curve bring up the time required to build the classified website as well as the cost of the project. This raise in time required to bring the project to market and the increase in the cost that's brought with it is a down side the developer or the organization has to face when carrying on such projects

1.3. Research Problem

Developers face problems of having to learn new emerging technologies that come out and changes to existing technologies already implemented in their current web applications. This brings on a learning curve which increase the time that the developer takes to implement the changes or build a new version of the system. The main requirement of this research is to find solution to the defined below:

Development of a Developer Friendly Middleware framework

One problem faced by developers is the number of different technologies needed to build a web/mobile application. This is also accompanied by the

core differences between these technologies and the differences that arise by the organization that have defined the way their technologies have to be used.

This knowledge gap of the developer not only increase the time to develop a system but also bring on changes to existing systems.

Developing Middleware Framework that can Integrate new technologies easily

One of the main problems faced by developers is how to integrate new technologies to existing system without having to make changes to existing version of the system. For Example: how to Integrate a new social authentication service to yahoo while already catering to Facebook, google and local authentication

1.4. Research Objectives

Main objective of this research is to develop an open source middleware framework for classified based web development. This framework should minimize the workload and the learning curve which a developer has to face throughout the development phase of classified websites. For the developers who are in the field of classified web development, should be able to

- Reduce developer burden, complexity and knowledge gap during the development process.
- Reduce development time and hence cost of a classified based web development project.
- Get sophisticated features as well as wide range of flexibility.

Specific Objectives

- Design and implement the core framework with the integration of other components and modules. Configurations including permissions, security, roles, routing, services libraries that gives the developer the tools they need

for modern web development. This middleware framework will facilitate evolution, enhance the reusability and as well improve portability to new platforms. The framework includes other features such as routing and data type conversion, error detection and handling.

- Design and implementation of Core Framework (Middleware framework core) including the restful Service API used by the web developer for classified web development (e.g.: ikman.lk, craigslist.org) and End Points to External Application (Mobile/Web), integration with other components of the framework, permissions and roles and security of the core framework, routing, services library that gives the developer the tools they need for modern web development. These endpoints include services the web application/site in its core functionality such as displaying the advertisements, user uploading the advertisements, to even user submitting a form will be provided with a uniform, high level API (Applications Programming Interface) to applications. This middleware framework will facilitate evolution, enhance the reusability and as well improve portability to new platforms. The framework includes other features such as routing and data type conversion, error detection and handling.

2. METHODOLOGY

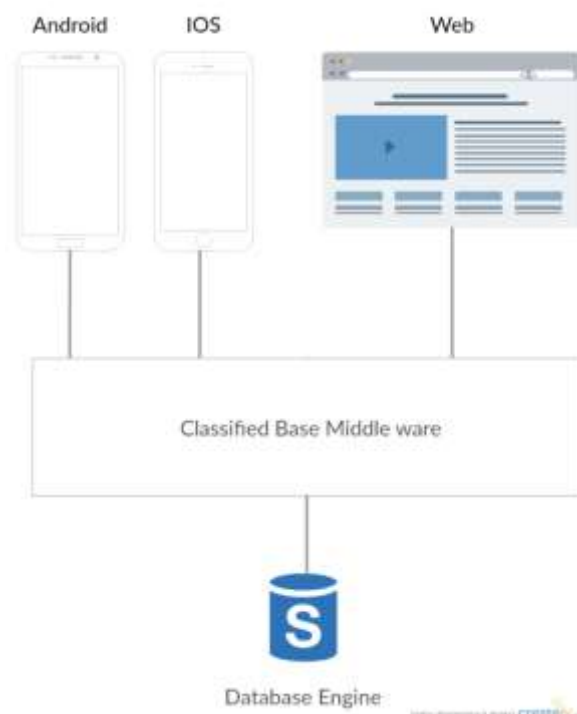
This section provides in-detail description of the research objective, the design and development of the core framework module. It will provide a set of guidelines or principles that can be tailored and apply for specific situation, as well as it also

includes specific approach, templates, forms, and testing used over the project life cycle. In Ampliar middleware all component configuration is setup through the core module. Ampliar is a classified based web development framework primarily focus on to increase the productivity and reliability of classified based web development projects. Primary goal of this product or research is provide efficient, developer friendly, extensible, modular and reliable middleware framework. This primary goal is a combination of four main objectives, those are listed below.

- Highly flexible core framework
- Extensible as well as minimal sql interaction database extension module
- Extensible authentication
- Extensible web analytics generator

From this point onwards, document present the methodology that used to achieve the Database extension point objective. All guidelines, specific approach, templates, forms, and testing are described in follow sub chapters in detail. This middleware framework that can handle the complexity of development process. This middleware framework positioned between the application UI and the Database engine.

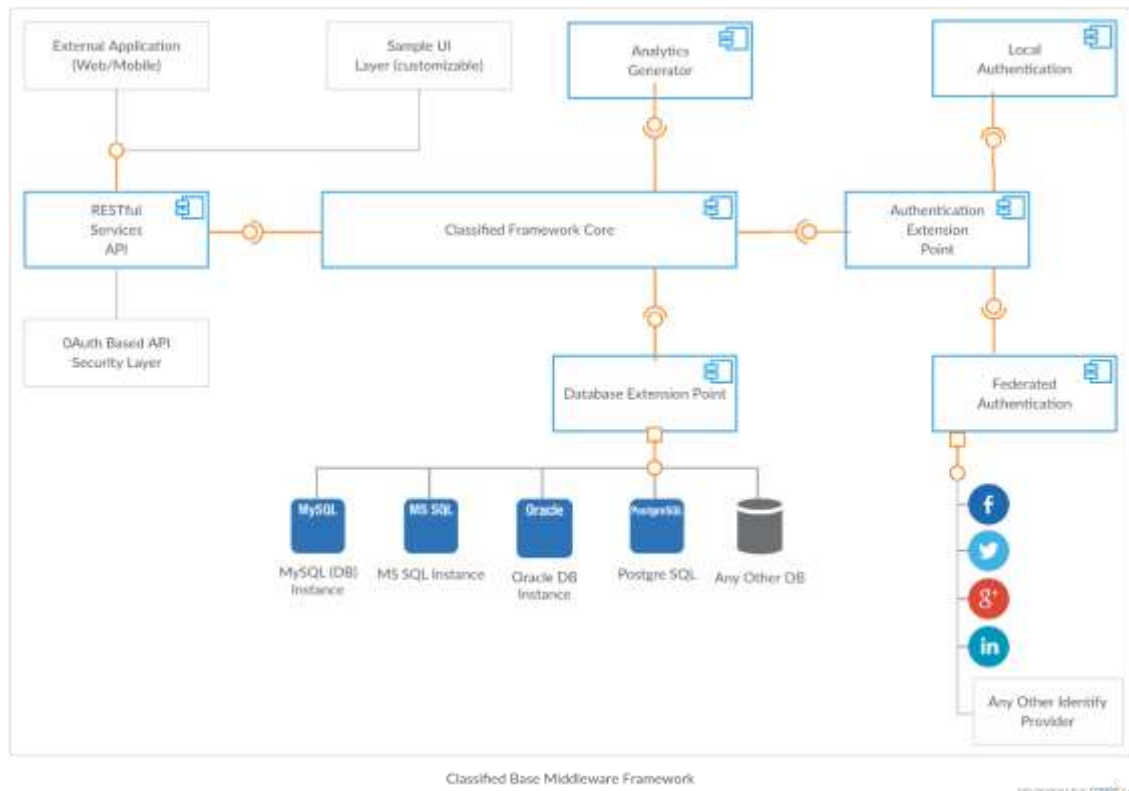
Middleware expose core functionality of classified application as RESTful web services, this conceal the complexity of development process. Since the middleware is an independent layer developer can use any front-end technologies to develop application front end present high level diagram of this middleware use case.



The solution follows modularized SOA approach. The framework is composed of four main components, those are listed below.

- Framework Core module
- Extensible Database Abstraction
- Authentication Extension module
- Analytics Generator module

This is an open source framework. Each module in this framework can be extensible or even customizable according to the user's requirements. The figure below visualize the architecture of this middleware framework and the Core Frame Module connectivity to other Components of the System.



2.1. Methodology

This subsection provides the development life cycle, each procedures, techniques, implementation, best practices and testing in-detail carried out through database extension point.

2.1.1. Background Study

According to marketing statistics as well as new trends, classified web development extends from generic developments to specialized developments. This increase the number of classified base development projects. As discussed, this development process takes huge amount of time and effort. It's important to have a solution to boost this development process meanwhile reduce the effort. Currently in this domain we can see multiple frameworks and plugins provide facilities for classified based web development. Almost all these products contain different levels of limitations when we compare those products with classified base web development domain needs and requirements. One of the highlighted limitation is extensibility of those product. Almost all free products are limited in extensibility. Most of the cases those products only support one database vendor, also its very hard to extend without having deep knowledge about product. Those extension leads different issues in production environments. Which is ultimately create developer frustrations as well as waste of valuable resources. Although some products are customizable unfortunately those products are vendor preparatory products. In order to customize the product, customer needs to depends on the vendor. Most cases customization request as well as huge amount of money need to spend on those vendors. At the end of the day this type products make customer totally depends on those vendors.

2.1.2. Requirement Gathering

In get started we went through existing frameworks and plugins to find the problems and weakness of those, we used most of the existing frameworks to get an idea of the differences of the architecture of each available framework. We even reached out on Professional developers using the existing framework in the given scope and even developers using different platforms to build Classified Websites to find features and functionalities of those products. We also went through the provided API documentations of each of these products. Throughout this process we are able to find strength of those products as well, which is help us to make our product / middleware as a next step of classified based web development frameworks. Meanwhile we also researched on modern techniques and tools available not only on development of these web application but also security and performance of these tools and techniques.

2.1.3. Design

The core framework component defines an interface and configurations common web elements and services API. This also provides HTTP security config and Data Conversion Abstraction for HTTP request data conversion API. Developers can implement the pre-defined interfaces according to the given specifications or even extended the provided Abstract classes to make necessary modifications for the system. The core framework component also provides with configuration files for Integration of the other components, or even 3rd party component integration. The RequestHandler class provides the developers with implemented REST API methods that they can directly use, otherwise developers are able to extent the given class as they wish to use. If the developer wishes to customize the project architecture, it is also possible with minimum configuration and modification to relevant modifications to interface as well as the implemented classes.

2.2. Testing and Implementation

2.2.1. Implementation

- **Restful Service Abstract**

RestHandler Class is a Abstract class which contains the core Restful service required for the functionality of the application and hides the complexity of framework Services functionality. This class can be extended by the user to add additional functionality or modifications.

```
public class RestController {
    Gson gson = new Gson();

    @RequestMapping(value = "/api-get-all-cars", method = RequestMethod.GET, produces = "application/json")
    public String getAllCarsApi() {
        ArrayList<Advertisement> tempAds = new ArrayList<Advertisement>();
        tempAds = new QueryHandler().findAllAdvertisements(null, null);
        return gson.toJson(tempAds);
    }

    @RequestMapping(value = "/api-insert-mobile", method = RequestMethod.POST)
    public String apiInsertMobile(HttpServletRequest request, @RequestParam("files") MultipartFile[] files) {
        ArrayList<AdvertisementImage> adimage = new ArrayList<AdvertisementImage>();

        for (MultipartFile multipartFile : files) {
            adimage.add(new AdvertisementImage(multipartFile));
        }

        Category cat = new Category(1, request.getParameter("Category"), 1);
        SubCategory subcat = new SubCategory(1, 1, request.getParameter("SubCategory"), 1);
        District dis = new District(1, request.getParameter("District"), 1);
        DistrictLocalArea disLocal = new DistrictLocalArea(1, 1, request.getParameter("DistrictLocalArea"), 1);

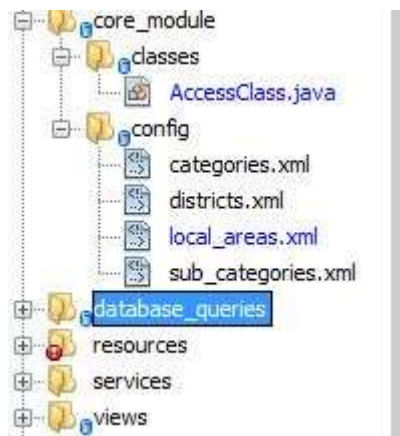
        Mobile mobile = new Mobile(1, request.getParameter("title"), adimage, cat, subcat, dis, disLocal,
            Double.parseDouble(request.getParameter("price")), 1, request.getParameter("condition"), request.getParameter("has"),
            request.getParameter("model"), request.getParameter("authenticity"), request.getParameter("bluetooth"),
            request.getParameter("camera"));

        new QueryHandler().insertAdvertisement(mobile);
    }
}
```

The class can be used as such to provide the service required.

- **UI component Generation**

Common UI components have been Identified an a API service been Implemented for the Component generation. With this a configuration is provided for each UI element as given below.



This will save the Developer time in writing the UI component and making modifications. The Services are Implemented as given below.

```

<?page import="org.w3c.dom.*, javax.xml.parsers.*" %>
<%
    DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
    Document doc = docBuilder.parse("http://localhost:8080/emp1st/core_module/config/districts.xml");
%>

<%!
    public boolean isTextNode(Node n) {
        return n.getNodeName().equals("#text");
    }
%>

<select class="form-control" id="district-select" onchange="loadLocalAreaSelect()" ">
    <option value="">Select</option>
    <%
        Element element = doc.getDocumentElement();
        NodeList personNodes = element.getChildNodes();

        for (int i = 0; i < personNodes.getLength(); i++) {
            Node emp = personNodes.item(i);
            if (isTextNode(emp)) {
                continue;
            }

            NodeList NameDOBCity = emp.getChildNodes();
        }
    <%
    <option>
        <%
            for (int j = 0; j < NameDOBCity.getLength(); j++) {
                Node node = NameDOBCity.item(j);
            }
        %>
    </option>
    </select>

```

2.2.2. Testing

Testing was carried out on the usability of this component and the whole framework itself. It was also to identify and investigate the quality of the product and each of the module. Testing helped identify the potential risk and reliability of the framework. Through the testing we were able to find errors and other defects which helped us to maximize the quality of the product at the production environment. The core framework component was compared with other existing frameworks of similar platform to develop classified web site. Comparing the development times and the amount of technical skills and knowledge (learning curve) required by the developer to complete a given task.

2.2.2.1. Module Testing

Module testing involve testing and verify the functionality of each functions inside a module. In this core framework module, module testing carried out by performing test on each REST API service, Configuration of the UI components and the Configuration of the system files. We tested to identify the breaking points of the API and security of the API services. Testing was also carried out to identify any potential system errors and to handle the appropriate exceptions and errors without causing the system to fail at production.

2.2.2.2. Integration Testing

Module testing ensured the functionality of each module at the atomic level or isolated level. Integration testing with other modules was carried out related by integrating a single module at a time and carrying out system functionalities with the Integrated component Changes were made to the system configuration file and testing was carried out to identify and fix errors and exceptions. This was carried out similarly with other components of the system. Mainly Integration testing was carried out after each module had made modification or a new feature was added, testing was carried out in the Integration system at that time (Staging Version) to

make sure each component was functioning properly as an Individual and as a Independent Module.

2.2.2.3. System Testing

This is the final level of testing a system. Integration testing verify the communication quality between components. In system testing more focus on testing the whole system. This was carried out by our Team as well as other developers using the system during the development stage. This ensured us that the System was functioning properly. System testing was done also to verify the security, performance and usability objectives of the middleware frameWork.

2.3. Research Findings

The research helped us to explore the domain of classified based web development. We are able to identify what are the requirements of this domain, users of this domain , the business process and the challenges faced by the developer building a classified based website, with every advancement of technology, and a high competition to bring the application to market developer are facing a challenge. With all those facts we are able compare and analyses existing products with the classified based web development domain to find problems as well as weakness of those products. Major drawbacks found in those products are limited to one platform in most cases. These products need to deploy at vendor's platform. Next issue was lack of customization facilities. None of those products support customizations at end user level (developer level). Developers need to use products directly the way vendor provide it. Also, none of those products are support easy extensibility. Identifying those weakness and team use those as strengths of our product. Basically, our product use core java, which means these modules are able use with any existing java frameworks. Support for major database management systems if developer wants extend it's also possible with few amounts of codes. No vendor base deployments, developer can deploy choose environment if they wish to deploy. All source codes

are open source; therefore, anyone can modify or extend the framework according to their preferences, also they can offer their contribution to this framework for further enhancements and improvements.

3. CONCLUSION

The middleware framework provides an effective and efficient way of developing classified based website. The web development community in Sri Lanka is rapidly growing, and this framework can boost their work allowing them to deliver their product at a much lesser time, allowing them to focus their resources and time for the business process itself and adding value to them. Because of the current

unavailability of such a customizable framework in Sri Lanka, the market segment can easily be captured.

The middleware framework by the authors will be an open source project that can be worked by the community to be used by the software developers in Sri Lanka and worldwide. Documentation and guidelines will be made available to the developers and the middleware framework which they can use to build their applications on top of that. With time more features can be added which enlarges the user base of this middleware framework.

4. REFERENCES

- [1] "Yclas Home," Yclas, [Online]. Available: <https://yclas.com/>.
- [2] "flynax home," flynax, [Online]. Available: <https://www.flynax.com/>.
- [3] "OXYCLASSIFIEDS Home," OXY CLASSIFIEDS, [Online]. Available: <https://www.oxyclassifieds.com/>.
- [4] "TITAN CLASSIFIEDS Home," TITAN CLASSIFIEDS, [Online]. Available: <http://www.titanclassifieds.com/>.

- [5] "OSCLASS Home," OS CLASS, [Online]. Available: <https://osclass.org/>.
- [6] J. Pan, "Software Reliability," [Online]. Available: https://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/.
- [7] A. Biswal, "Java caching frameworks for enterprise applications," LinkedIn, 13 March 2015. [Online]. Available: <https://www.linkedin.com/pulse/java-caching-frameworks-enterprise-applications-anshuman-biswal>.
- [8] J. Community, "Ehcache," [Online]. Available: <http://www.ehcache.org/>.
- [9] P. Community, "PostgreSQL," PostgreSQL, [Online]. Available: <https://www.postgresql.org/>.
- [10] M. Community, "MySQL," MySQL, [Online]. Available: <https://www.mysql.com/>.
- [11] J. Community, "Hibernate," [Online]. Available: <http://hibernate.org/>.

5. APPENDICES

Table of comparison between existing products and proposed product

Features	Yclas	Flynax	Oxy classifides	Titan classifides	C
Open Source	<i>YES</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	
Free	<i>YES</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	
Extensible Database Integration	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	
Social Login	<i>YES</i>	<i>YES</i> <i>(only facebook)</i>	<i>YES</i>	<i>NO</i>	<i>(or)</i>
Extensible Federated	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	

Authentication					
Web Analytics	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	