OPEN SOURCE SOA BASED MIDDLEWARE FRAMEWORK FOR CLASSIFIED BASED WEB DEVELOPMENT

Project Id :17-072

Project Proposal Report

Bachelor of Science Special (Hons) Degree in Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

27th March 2017

OPEN SOURCE SOA BASED MIDDLEWARE FRAMEWORK FOR CLASSIFIED BASED WEB DEVELOPMENT

Project Id :17-072

Project Proposal Report

K.S.D.A. Kulathunga – IT14025686
W.M.N. Radith – IT14011030
I.H. Liyanaarachchi – IT13137342
B.R.K.S. Kumari – IT14047152

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

27th March 2017

DECLARATION

We declare that this is our own work and this proposal 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
1	K.S.D.A Kulathunga (GROUP LEADER)	IT 14 0256 86	
2	W.M.N Radith	IT 14 0110 30	
3	I.H. Liyanaarachchi	IT 13 1373 42	
4	B.R.K.S. Kumari	IT 14 0471 52	

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

Online Classified Advertising has dominated over traditional newspaper advertising since the emergency of web technologies, and the enhancements in mobile devices has further projected the activities as well as number of contents (ads) now being advertised on classified advertising websites. This online tradition of advertising has improved the credibility of the advertisers and also improved the speed of transaction, where the traditional way had the interested buyer having to go through hundreds of magazine advertisements before making a sound decision on his / her purchase. This rapid growth in the classified industry has put many companies seeking to gain a share in this ever-growing market, which has lead to a number of new classified websites being created every year. This has also brought up huge competition between existing classified websites and new emerging ones, forcing the organization to update their web functionalities or even expand to new emerging technologies before the competitive rivals do. This time to adapt to the competition has directly affected these web businesses. Developers creating these web contents, have not only be able to make changes to their existing sites, but also to adapt to new technologies that is continuously changing. This raise in learning curve for the developer has direct effect on the time to bring the application or changes in to the domain. Our Middleware Framework focuses on reducing the learning curve the developer has to face when developing classified web site. It will support the developer by having a developer friendly API that will reduce the learning curve of the developer to building classified web sites. This Framework will support many of the technologies need to build a classified website from scratch. It will also allow the developer to integrate new technologies to the framework, requiring only minor changes. This will highly reduce the development time of classified web site which will also directly reduce the cost of bringing the application to market. This will also advance their chances to sustainability in a market where the time to adapt to technological changes has been a factor of survival in internet businesses.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Background

The origin of classified base 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 decreases the profitability of those printed media [1]. Online web format of classified service provides the 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].

The 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 base web development. This middleware handles the complexity of different technology layers, that are involve in the development process.

1.2 Literature survey

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 applications 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 developers 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 [3] 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 Gap

The Research gap is defined as the area or topic for which missing or insufficient information limits the ability to reach a conclusion for a question. This research is directly adopted with the development online classified ad domain. When considering this domain not much other researches done in implementing or identifying the relationship between the development ease and time to market the product. Although tools existing exists today in the market for developers to get started on developing classified websites, the tools available have limited support to new emerging technologies and thus making it hard for the developer to maintain or even adopt to new technologies. In the software development field reducing the development time is the secret of reducing the time to market and reducing the project development cost.

1.4 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:

1.4.1 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.

1.4.2 **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

2. OBJECTIVES

2.1 Main Objective

Main objective of this research is to develop an open source middleware framework for classified base 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

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

2.2 Specific Objectives

- Design and implement a component which supports multiple database manage systems (MySQL [4], MSSQL [5], Oracle [6] etc.) and has the capability to extend the database abstraction layer based on the developer's preference. For an example, if the developer needs to connects his/her website to a different database management system apart from the DBMSs listed in the framework, he/she will be provided an easy way of customizing the connection.
- Design and implement the architecture component to extend federated authentication. By default, federated authentication supports facebook, twitter, yahoo authentications. If developer needs to provide authentication from a different website, he/she will be provided a feature to add that authentication facility to the framework.
- Design and implement a component to use as a web analytic component to monitor and analyses the statistics of the website. Developer may give the opportunity to choose between existing analytical libraries and also, he/she will

be given the facility to develop an analytic library himself by following certain instructions.

- 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. 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.

3. METHODOLOGY

3.1 System Overview

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 the huge amount of time and effort. It's important to have a solution to boost this development process meanwhile reduce the effort.

To resolve this problem, the research group propose a middleware framework that can handle the complexity of development process. This middleware framework positioned between the application UI and the Database engine. Middleware exposes core functionality of classified application as RESTful web services, this conceals the complexity of development process. Since the middleware is an independent layer [7], the developer can use any front-end technologies to develop application front end.

3.2 System Architecture

The solution follows modularized SOA approach. The research group identifies 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. Figure 2 visualize the architecture of this middleware framework.

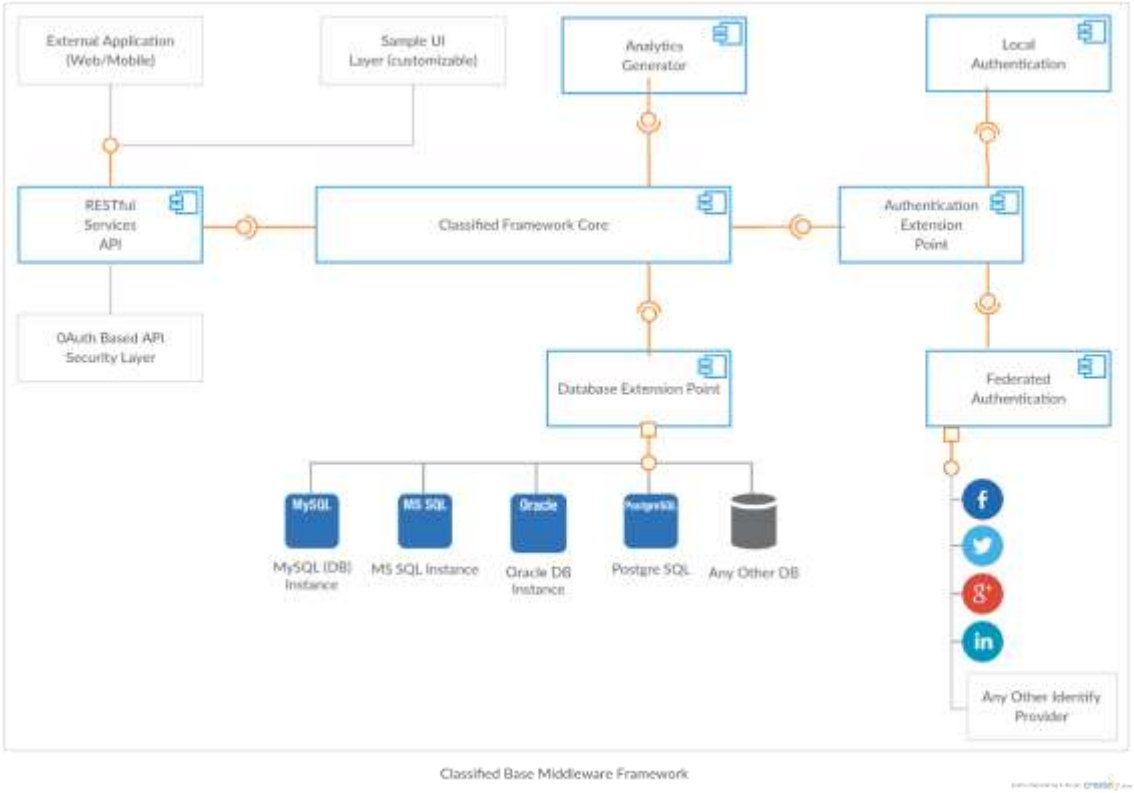


Figure 1 – Classified base Middleware Framework

3.2.1 Core Framework Module

The Core framework module responsibility is the design and creation of the high-level API that is exposed to developer and the other components of the system. This includes the routing of request as well as security and data conversion of the request. This will also include pluggability of the other Components, and extensibility to add new authentication as wanted by the developer and even to support multiple databases. Other objectives focused is the usability and development of interoperable API with other technologies. This will allow the developer to freely select the client technology to be used regardless of the platform (mobile/web).

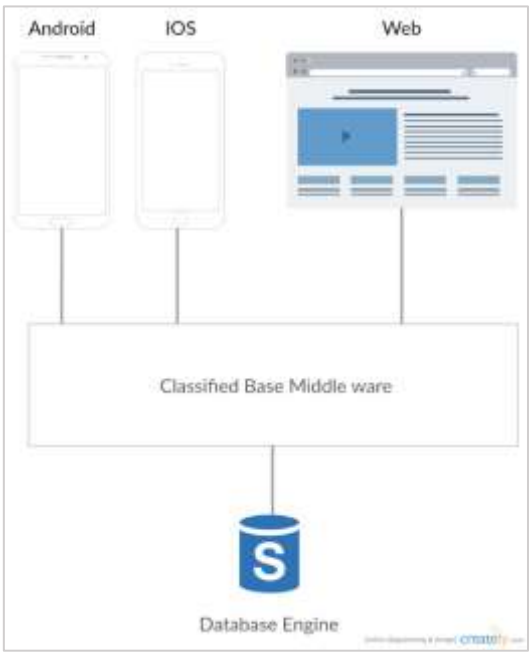


Figure 2- High level diagram

3.2.2 Extensible Database Abstraction

This module falling under software engineering research category. The main focus of this module is the development of extensible data abstraction layer and a caching layer. All database related activities are handled through this Module. Module exposes RESTful web services to other modules. Figure 3 shows a high-level Architecture diagram of this Module.

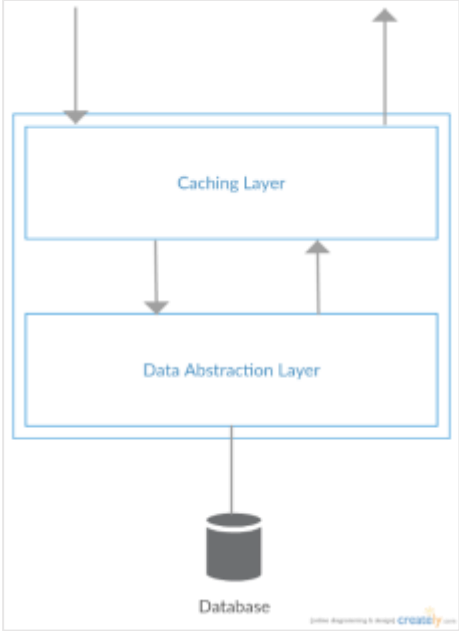


Figure 3 - Extensible Database Abstraction Architecture

3.2.2.1 Caching Layer

These types of domains user request various type of search queries. If there is only few considerable number of users requesting the same query, then it's possible to query the database on each request. Although classified base web applications manage a vast number of users. It can be a major performance issue if each identical search query directly forwards to the database. As a result, response might be delay, application might freeze worst case application might crash [8]. It's important to maintain good performance even in business perspective, because the competitive nature of the classified web market.

To overcome this issue research group proposed caching technique. Caching is a major concept when it comes to performance. This layer cache each search query for a constant amount of time given by the developer, this time can be configurable. If caching layer contains the response for requested search query then it directly response for that, otherwise, it will forward that query to data abstraction layer in order to get the result. When result arrived caching layer temporary store it while for given time period according to configurations and forward that result as a response to user.

Out of the box, java offers direct methods for caching. Java Hash Map, Hash Table and JNDI are some of those methods that contain in Java caching System. The major drawback is none of those default methods are provide a mechanism for removing the cache object from the memory when it's no longer need or automatically creation of the objects when its accessed after expiration. [9]

To overcome those, major drawback that we discussed above, there are several open source and commercial frameworks are available. JBoss [10], OSCache, EhCache [11] are some open source fireworks where SpritCache, Coherance [12] are commercial frameworks.

Team consider below listed factors in order to decide the optimum caching framework

- Support general purpose caching
- Minimal dependencies
- Documentation level
- Production performance
- Open source

The team left SpritCache and Coherance [12] because those are not from open source world, there for we cannot use those for open source development.

EcCache [11] was the best selection according to factors that we considered. It's an open source, well-documented framework with minimal decencies. Also, lot of leading java frameworks used EcCache [11] for caching. Spring [13], Hibernate [14] are few of those. [8]

3.2.2.2 Data Abstraction Layer

One of main concern in this framework is extensible Database support. By default, framework support MySQL [4], MsSQL [5], OracleSql [6] and PostgreSQL. [15] Although proposed architecture designed in extensible manner where developer can develop and plug another DBMS components. Through configuration options developer can configure DBMS configurations. Figure 4 shows the high-level architecture of Data Abstraction Layer.

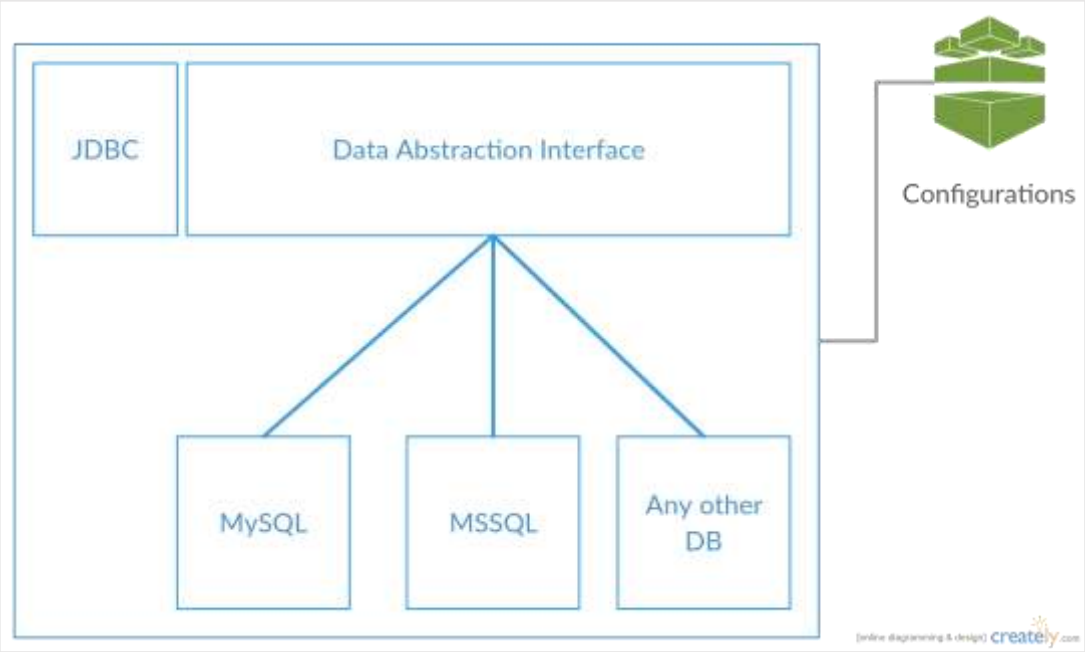


Figure 4- High-level Architecture Diagram Data Abstraction Layer

Data Abstraction layer contains JDBC component and Data Abstraction interface. JDBC is a specification from Sun Microsystem that provides stranded abstraction(API) for java applications to communicate with different databases. JDBC has 4 type of drivers, [16]

- **Type 1 driver –** it's act like a bridge between JDBC and other DB connectivity mechanisms such as ODBC, therefor its depends on ODBC driver which makes application indirectly depends on ODBC [16]
- **Type 2 driver -** it's a native API driver, its need native binaries installed in running environment [16]

- **Type 3 driver –** it's a middleware base driver, it converts JDBC call into database specific calls using middleware, which caused to increase the network traffic [16]
- **Type 4 driver –** it's a database protocol driver, this driver converts JDBC call into Database call using database specific protocol. This protocol is proprietary protocol for DBMS vender [16]

Type 2 driver is the fastest one among those four, but it needs specific binaries in order to communicate with the database. Type 3 driver is the most independent one it lack of performance due to vast network communication. Almost all DBMS vendors provide protocols today where which lead us to select type 4 driver as JDBC driver compare to other drivers. [16]

Data Abstraction interface defines common methods for each operation. For example, CREATE TABLE, EXECUTE QUERY. Interface defied generic methods for generic database structure of a classified web application. In each class implements this interface should implements those methods. For example, MySQL [4] Data access class implements Data Abstraction interface, then this class needs to implements all methods according to MySQL [4] specifications. This concept can apply for any DBMS vendor, if the framework by default do not support for particular vendor, then developer can develop and plug specific Data Access class implementing Data access interface provided by the framework, according to that particular vendors' specification. This ensure the extensibility of the Extensible Database Abstraction.

Java world we can see few ORMs which are type of Data Abstraction Layers. For example, Hibernate API. The main drawback of those, they reduce the performance of application comparatively to non-orm layer as well as hard to maintain. For example, changes that involve in ORM code are often scattered across many components. [17] Other benefit is it reduce the development complexity, which is handy when it comes to another developer to understand and extend framework modules. Therefore, team decided not to use any ORMS to this development process.

3.2.3 Authentication Extension Module

Nowadays users are widely using internet to fulfil their needs and requirements. In order to achieve their requirement, users may need to create user accounts in various websites. If you use different user credentials for different websites, you will definitely face the trouble of remembering passwords for the websites you're rarely using. To overcome this situation modern websites make use of OAuth protocol with the concept of "Identity federation" and "Delegated Authorization". If a website is providing a facility to use a different identity provider such as facebook, twitter, linkedin etc. you can simply sign up using that identity provider, rather than wasting your time by filling a lengthy form to create an account. But most of the frameworks doesn't come up with in built APIs for federated authentication. When it does, it will be only for the popular identity providers. There will be no mechanism for the developer to use different identity provider of his/her preference. This component mainly focuses on providing a facility to extend the federated authentication in order to make the development tasks easier for the developer.

Following is the generic way of how authorization should work.

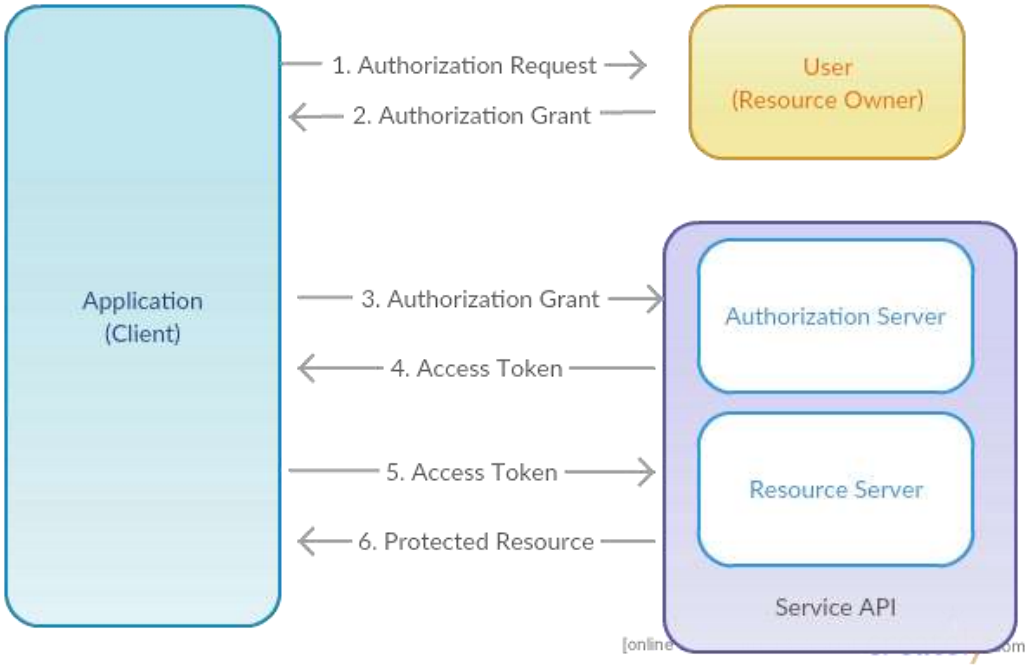


Figure 5-Generic process of authorization

Following is an example of what exactly happens during the federated authentication process.

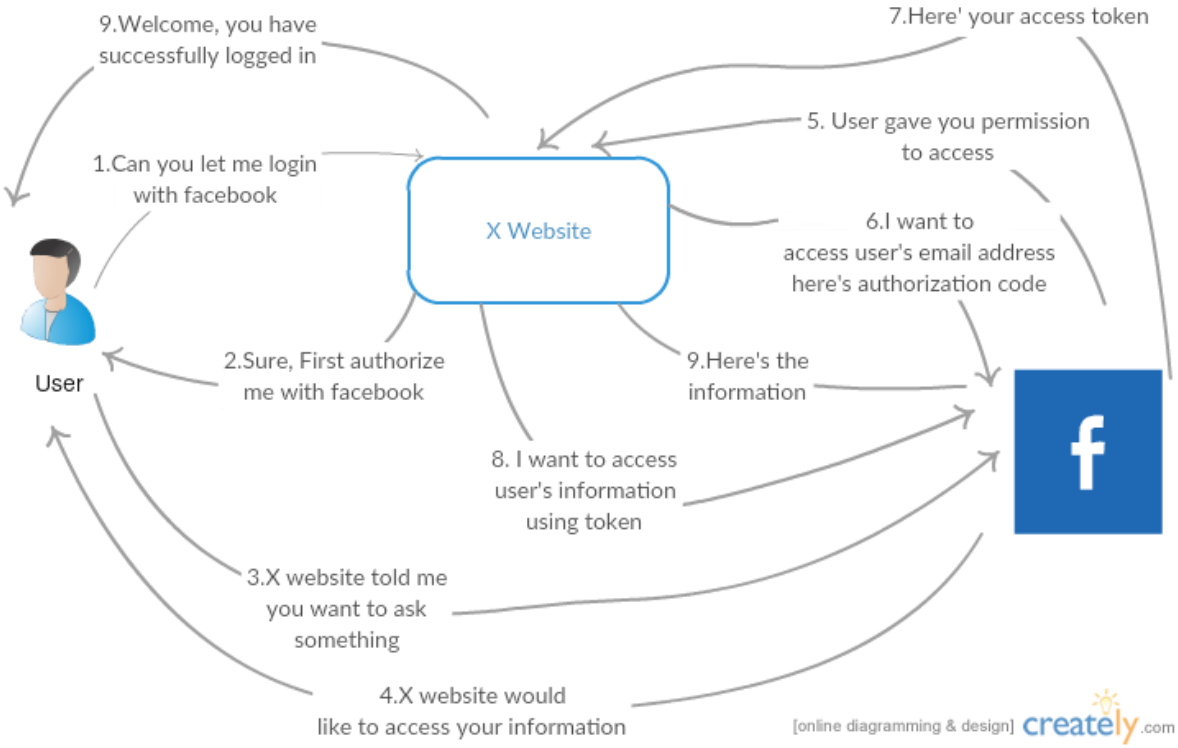


Figure 6- Example of federated authentication process

For the authentication purposes, we use OAuth 2.0 authorization framework. OAuth 2 is an authorization that enables application to obtain limited access to user accounts on an http service, such as facebook, github, linkedin etc.

There are several roles defined by OAuth.

- Resource owner
- Client
- Resource server
- Authorization server

For the users who are logged in for one time access, we use just in time provisioning mechanism to create users on the fly without creating user accounts in advance. We use internal identification numbers to keep track of them, if they visit again.

Following diagram shows the component architecture.

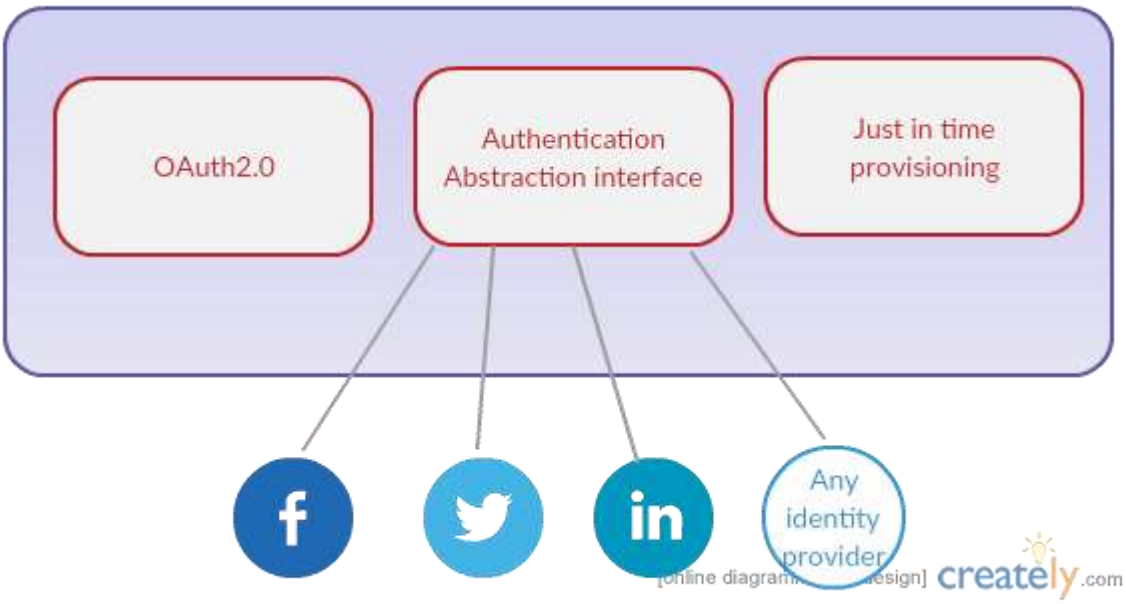


Figure 7 - Extensible authentication architecture

This component will integrate the most popular APIs into one and give the facility to use any identity provider of developer's preference.

3.2.4 Analytics Generator Module

Web analytics is measurement, collection, analysis and reporting of web data for purpose of understanding and optimizing web usage. It mainly focuses on the process of analyzing the behavior of visitors to a website.

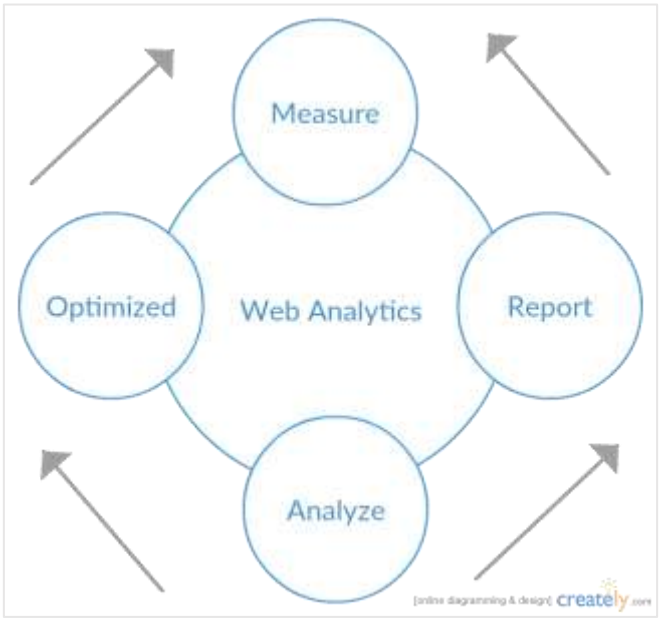


Figure 8

This will be based on web log data mining. Every web server keeps a log of page requests that can include visitor ip address, date and time of the request, request page, and referrer. This component will monitor and analyze the statistics of the website. Web analytics basically happens by considering the following metrics:

- **Visitor Type** – who is accessing the web site (returning, unique, etc.)
- **Visit Length** – The total amount of time a visitor spends on the web site
- **Demographics and System Statistics** – The Physical Location and information of the system used to access the website
- **Internal search information** – Information on keywords and results pages viewed using a search engine embedded in the website
- **Visitor Path** – The route a visitor uses to navigate through the Website

- **Top pages** – The pages that receives the most traffic

Through this component developer may get the opportunity to choose between existing analytical libraries and also will be given the facility to develop an analytic engine by himself. Can used the available web analytics engine based on the requirement of the developer. Such as Google analytics, spring metrics, Woopra, Clicky, Mint, Chartbeat and etc.

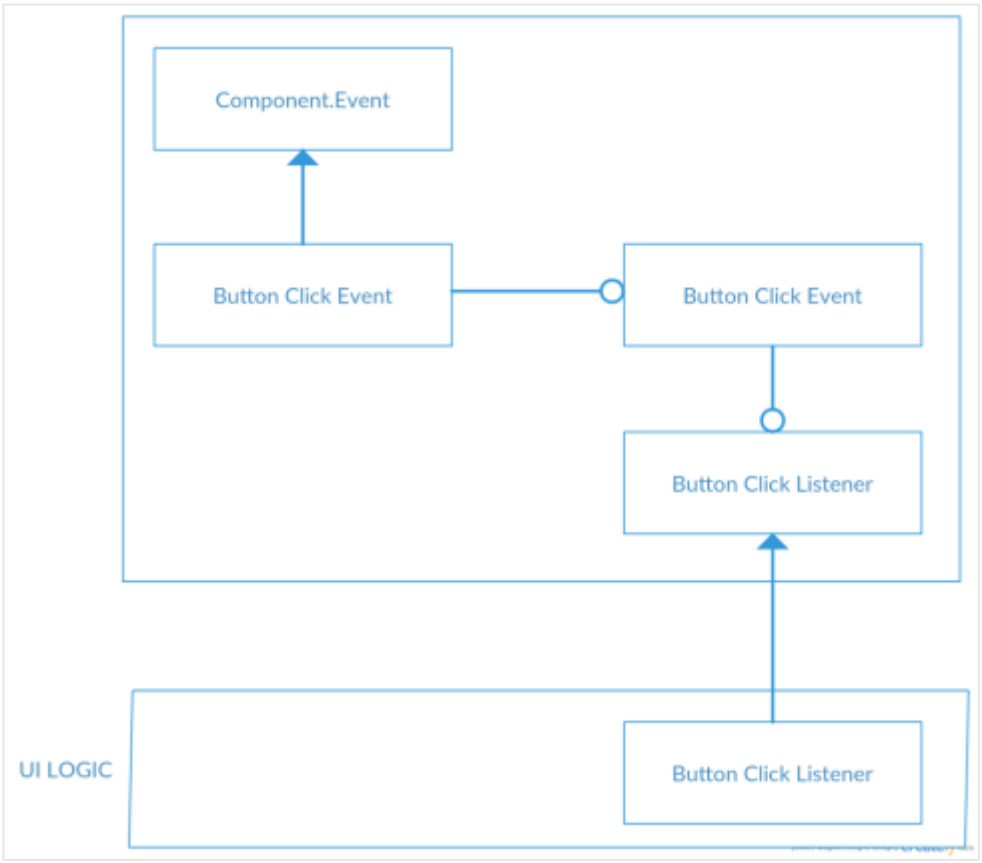
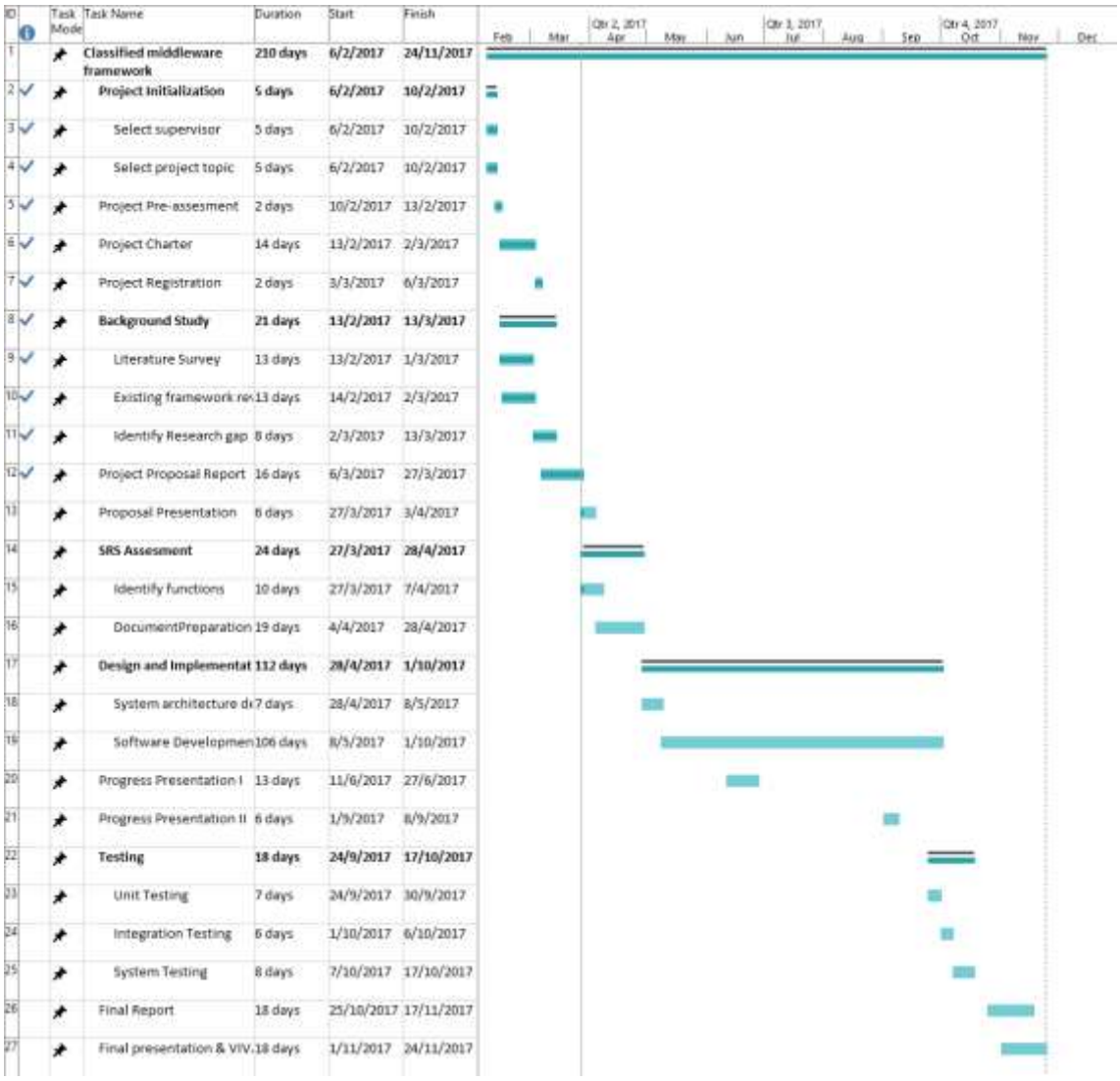


Figure 9

The analytics component is mainly focusing on the events handle by the user. It follows a listener architecture to develop the analytics generator. Each event consists with pre, actual and post event. Therefore, this component totally based on Event Listener Concept which can create listener for the events happening in the web site separately. By using those event categorization developers will be capable of analyzing the event by calling relevant event listener when it's needed.

3.3 Gantt Chart



ID	Task Mode	Task Name	Duration	Start	Finish
1		**Classified middleware framework**	**210 days**	**6/2/2017**	**24/11/2017**
2		**Project Initialization**	**5 days**	**6/2/2017**	**10/2/2017**
3		Select supervisor	5 days	6/2/2017	10/2/2017
4		Select project topic	5 days	6/2/2017	10/2/2017
5		Project Pre-assesment	2 days	10/2/2017	13/2/2017
6		Project Charter	14 days	13/2/2017	2/3/2017
7		Project Registration	2 days	3/3/2017	6/3/2017
8		**Background Study**	**21 days**	**13/2/2017**	**13/3/2017**
9		Literature Survey	13 days	13/2/2017	1/3/2017
10		Existing framework re	13 days	14/2/2017	2/3/2017
11		Identify Research gap	8 days	2/3/2017	13/3/2017
12		Project Proposal Report	16 days	6/3/2017	27/3/2017
13		Proposal Presentation	6 days	27/3/2017	3/4/2017
14		**SRS Assesment**	**24 days**	**27/3/2017**	**28/4/2017**
15		Identify functions	10 days	27/3/2017	7/4/2017
16		DocumentPreparation	19 days	4/4/2017	28/4/2017
17		**Design and Implementat**	**112 days**	**28/4/2017**	**1/10/2017**
18		System architecture d	7 days	28/4/2017	8/5/2017
19		Software Developmen	106 days	8/5/2017	1/10/2017
20		Progress Presentation I	13 days	11/6/2017	27/6/2017
21		Progress Presentation II	6 days	1/9/2017	8/9/2017
22		**Testing**	**18 days**	**24/9/2017**	**17/10/2017**
23		Unit Testing	7 days	24/9/2017	30/9/2017
24		Integration Testing	6 days	1/10/2017	6/10/2017
25		System Testing	8 days	7/10/2017	17/10/2017
26		Final Report	18 days	25/10/2017	17/11/2017
27		Final presentation & VIV	18 days	1/11/2017	24/11/2017

Figure 10- Gannt Chart

4. DESCRIPTION OF PERSONAL AND FACILITIES

Member	Component	Task
K S D A Kulathunga	Middleware Framework Core Component	• Development of the core framework routing • Design and Development of the developer friendly high level API
W.M N Radith	Database Abstraction Component	• Design and Implement of data caching layer • Design and Implement of Extensible Data Abstraction layer
Liyanaarachchi I.H.	Authentication extension module	• Integrate popular authentication APIs • Implement and Extensible mechanism to expand the authentication.
B.R.K.S. Kumari	Analytics Generator component	• Design and implement the listener architecture • Implement the basic analytic generator

5. REFERENCES LIST

6.

[1] S. Diaz, "On the Internet, A Tangled Web Of Classified Ads," Washington Post, Friday August 2007. [Online]. Available: http://www.washingtonpost.com/wp-dyn/content/article/2007/08/30/AR2007083002046.html?hpid=sec-tech.

[2] "Classified advertising," wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Classified_advertising#cite_note-4.

[3] T. W. Lin and E. J. Blocher, Cost Management: A Strategic Emphasis, Tata McGraw-Hill Education, 2006.

[4] M. Commiunity, "MySql," MySql, [Online]. Available: https://www.mysql.com/.

[5] Microsoft, "SQL Server," Microsoft, [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-2016.

[6] Oracle, "Oracle Database," Oracle, [Online]. Available: http://www.oracle.com/technetwork/database/database-technologies/sql/overview/index.html.

[7] A. PUDER, K. ROMER and F. PILHOFER, DISTRIBUTED SYSTEMS ARCHITECTURE A Middleware Approach, Elsevier, 2006.

[8] 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.

[9] S. Penchikala, "J2EE object-caching frameworks," JavaWorld, 31 MAY 2004. [Online]. Available: http://www.javaworld.com/article/2072734/open-source-tools/j2ee-object-caching-frameworks.html.

[10] J. Commiuinity, "JBoss," [Online]. Available: http://www.jboss.org/.

[11] J. Community, "Ehcache," [Online]. Available: http://www.ehcache.org/.

[12] Oracal, "Oracal Coherence," Oracal, [Online]. Available: http://www.oracle.com/technetwork/middleware/coherence/overview/index.html.

[13] Pivotal, "Spring," Pivotal , [Online]. Available: https://spring.io/docs.

[14] J. Community, "Hibernate," [Online]. Available: http://hibernate.org/.

[15] P. Commiunity, "PostgreSQL," PostgreSQL, [Online]. Available: https://www.postgresql.org/.

[16] K. L. Solutions, Java Server Programming Black Book, Dreamtech, 2014.

[17] T.-H. Chen, W. Shang, J. Yang, A. E. Hassan and M. W. Godfrey, "An Empirical Study on the Practice of Maintaining Object-Relational Mapping Code in Java Systems," p. 12, 2016.