

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

Group Report

17-072

Bachelor of Science Special (Hons) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

October 2017

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

Group Report

17-072

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of
Science Special (Hons) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

October 2017

DECLARATION

We declare that this is our own work and this dissertation 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 my 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.

Also we hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. We retain the right to use this content in whole or part in future works (such as article or books).

	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 the supervisor: Mr. Nuwan Kodagoda

.....

Signature of supervisor

.....

Date

ABSTRACT

Online Classified Advertising has been dominating the advertising industry over traditional newspaper advertising since the emergence of web technologies. The rapid growth of mobile technologies and the massive usage of mobile devices has further increased the boundaries of web content advertising over the years. The adoption of online web technologies for advertising has improved the credibility of the advertisers and also improved the speed of transactions, where the traditional way had the interested buyer having to go through dozens of magazine advertisements before making a sound decision on his / her purchase order. This rapid growth in the classified industry has put many companies seeking to gain a share in this ever-growing market, which has led to a number of new classified websites being popping up every year. This has also brought up huge competition between existing classified websites and newly emerging ones, forcing the organizations to update their web functionalities or even expand to new emerging technologies before the competitive rivals do. The time taken for adapting to the competition has directly affected these web based advertising businesses. Web Developers creating these web contents have not only been able to make changes to their existing websites, but also to adapt to new technologies that is continuously and inevitably changing. This raise in the learning curve for the developers has a direct effect on the time taken to bring the application or changes into the domain. This research project proposes a Middleware Framework that focuses on reducing the learning curve the developer has to face when developing classified based web applications. It will support the developers by having a developer friendly API that would reduce the learning curve of the developer for efficiently building classified based websites with core functionalities. This Framework supports many of the latest technologies needed to build a classified website from the scratch. It also facilitates the developers to integrate new technologies extending the framework, requiring only minor changes. This will highly reduce the development time of classified based web sites 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.

Keywords - advertising; classifieds; web development; service oriented architecture; advertising framework

ACKNOWLEDGEMENT

There are some individuals who have given their fullest support throughout our research project. It would not have been possible without the kind support and help of those individuals. We would like to extend our sincere thanks to our project supervisor Mr. Nuwan Kodagoda and our external supervisor Mr. Tharindu Edirisinghe for the kind guidance, inspiration and constant supervision as well as encouragement which support us in completion of this research.

We express our gratitude to Mr. Jayantha Amarakuchi for providing final years students with the knowledge that is necessary for writing and presenting effective project progress reports. Our grateful thank goes to all the personnel of the Sri Lanka Institute of Information Technology who contributed essential information. We are highly indebted to our examiner Mr. Indraka Udayakumara whose contribution in simulating suggestions and knowledge, helped us to coordinate our research.

We would like to express our special thanks to our parents, for their support and encouragement which help us in completion of this research.

The completion of this research could not have been possible without the participation and assistance of so many people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged.

TABLE OF CONTENTS

DECLARATION	i
ABSTRACT.....	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF TABLES	vi
LIST OF ABBREVIATION	vii
1 INTRODUCTION	1
1.1 Background Context.....	1
1.1.1 Background	1
1.1.2 Literature Survey.....	2
1.2 Research Gap.....	2
1.3 Research Problem.....	5
1.3.1 Difficulties in using numerous technologies for the development.....	6
1.3.2 Developing same functionalities repetitively	6
1.3.3 Restriction of using preferred database vendor.....	6
1.3.4 Difficulties in providing third party identity provider support	7
1.3.5 Difficulties in analysing web log data.....	7
1.3.6 Difficulties in identifying suspicious logins	7
1.4 Research Objectives	8
1.4.1 General Objective.....	8
1.4.2 Specific Objectives.....	8
2 METHODOLOGY.....	11
2.1 Methodology	11
2.1.1 Background Study	12
2.1.2 Requirement Analysis	13
2.1.3 Design	14
2.1.4 System Overview	14
2.2 Testing and Implementation	16
2.2.1 Implementation	16
2.2.2 Testing.....	33

2.3	Research Findings	34
3	RESULTS AND DISCUSSION	37
3.1	Results	37
3.2	Discussion	41
4	COMMERCIALIZATION OF THE PRODUCT	42
4.1	Position of the Market	42
4.2	Differentiate from the competition.....	42
4.3	Marketing Plan	43
4.3.1	Need of the Product.....	43
4.3.2	Marketing Strategy.....	44
4.3.3	Target Market.....	44
5	MEMBERS INDIVIDUAL CONTRIBUTION	45
6	CONCLUSION.....	48
	REFERENCES.....	49
	APPENDIX A: Class diagram for Database Abstraction	51
	APPENDIX B: Use case diagram for Authentication component.....	52
	APPENDIX C: Use case scenarios for Authentication Component	53
	APPENDIX D: Class diagram for Authentication component	54
	APPENDIX E: Test Cases for Authentication Component	55
	APPENDIX F: Use case diagram of Web analytics component.....	58
	APPENDIX G: Use case scenarios for Web analytics Component.....	59
	APPENDIX H: Class Diagram of Web Analytics Module.....	60

LIST OF FIGURES

Figure 2.1: Work Breakdown Structure	12
Figure 2.2: System Architecture	15
Figure 2.3: High level diagram of core framework.....	16
Figure 2.4: High level diagram of extensible database abstraction	17
Figure 2.5: Architecture diagram of database extension point	19
Figure 2.6: Data access interface	20
Figure 2.7: Sample use of query handler	20
Figure 2.8: A sample method in query handler class	21
Figure 2.9: Implementation of MySQL data access class.....	22
Figure 2.10: High level diagram of authentication module	24
Figure 2.11: High level diagram of OAuth process flow	25
Figure 2.12: High level diagram of extensible federated authentication	26
Figure 2.13: Code snippet of authentication abstract class	27
Figure 2.14: Facebook connector implementation.....	27
Figure 2.15: Linkedin connector implementation.....	28
Figure 2.16: Google connector implementation	28
Figure 2.17: Twitter connector implementation	28
Figure 2.18: Authentication configuration file.....	29
Figure 2.19: Web analytics process	31
Figure 2.20: Event listener architecture	32
Figure 2.21: Page visualization of the visitors	33
Figure 3.1: Login interface.....	38
Figure 3.2: Developer settings interface	38
Figure 3.3: OTP verification interface	39
Figure 3.4: Security Question verification	39
Figure 3.5: Ehcache configuration file.....	40
Figure 3.6: Visitor analysis screen	40

LIST OF TABLES

Table 1.1: Comparison of similar frameworks	5
---	---

LIST OF ABBREVIATION

Abbreviation	Description
MVC	Model, View, Controller
DBMS	Database Management System
API	Application Program Interface
SDLC	Software Development Life Cycle
UML	Unified Modeling Language
SOA	Service Oriented Architecture
CRUD	Create, Read, Update and Delete
JDBC	Java Database Connectivity
POJO	Plain Old Java Object
ORM	Object-Relational Mapping
OAuth	Open Authentication
XML	Extensible Markup Language
OTP	One-Time Password
CSV	Comma Separated Values

1 INTRODUCTION

1.1 Background Context

1.1.1 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.

1.1.2 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 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 has 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.

1.2 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.

There are numerous products available in classified domain for web development. Most of these products present themselves as web platforms, where the application developers can create classified based advertising websites consuming the features of that platform. After the creation of the application, the developer of the classifieds website needs to deploy the application on the same vendor's servers because these products are platform dependent. In addition to that, customizability is at a very poor level in many of the available products in the market.

Following are some existing frameworks available in the classified domain:

- Yclas

Yclas is a popular tool which enables the users who have little or no knowledge in web development, to create classified web sites. Deployment should be done on a platform that is provided by Yclas. Yclas can be used to create a website for real-estate, auto or jobs marketplace. It supports social logins such as Facebook, Google, Twitter. But it doesn't have the facility to extend the authentication point. It has full access to all needed functions to run a classified website; create categories, define locations, enable google map, add content and manage the admin panel. Yclas doesn't support multiple database vendors rather it only supports for mysql. Google analytics tool is integrated with the product. Initially users can get a free trial, but in order to get the full featured product, users has to select and pay for a pricing plan [4].

- Flynax

Flynax is a PHP script developed based on MVC architecture. They offer six PHP classified scripts for classifieds, auto classifieds, real estate agency, boat classifieds, pet classifieds and escort agency. It has 80 powerful plugins to boost basic functionality of the software. But users have to pay to get most of the plugins. Facebook connectivity is integrated with flynax. But doesn't have the facility to connect with other social logins. Category filter, similar adds, recently added ads functionalities are integrated with flynax. But it's expensive than other softwares in the classified domain. It doesn't have a web analytics module built within it [5].

- Oxy classifieds

PHP scripts based product that helps to build classifieds and this product is customizable up to some extent. Oxy classifieds define a number of classified ads templates such as General Classifieds, Auto Classifieds, Realty Classifieds and Boats Classifieds. Responsive templates, Location filter, email alerts, user groups and many functionalities are included with oxy classifieds. But it only integrated for the login with Facebook and Google. It doesn't provide any extensibility facility to extend the federated authentication. It doesn't support multiple database vendors as well. Oxy classifieds doesn't have web analytics module within it [6].

- Titan classifieds

This can create and cleanly manage classifieds professionally. Customizations can be done through vendor's developers. This product is now part of yclas product [7].

- Os class

This is a PHP script that allows you to quickly create and manage your own classifieds site free of charge. This has dozens of templates, themes and plugins to create classified web sites. Os class is fully customizable. It has various features such as adding categories and subcategories, custom fields, static pages and many more. It only has the connectivity for Facebook. And doesn't have an authentication extension for federated authentication. It doesn't support for extensible database abstraction. It doesn't include a web analytics module [8].

- Classipress

This is a word press plugin which helps to create and manage WordPress based classified websites. It's a theme used for classified ads for word press [9].

Considering the features of classified websites, we compared the existing products with the features we're going to provide with our framework.

Table 1.1: Comparison of similar frameworks

Features	Yclas	Flynax	Oxy classifieds	Os class	ClassiPress	Proposed framework
Open source	Yes	No	No	Yes	No	Yes
Free	Yes	No	No	Yes	Yes	Yes
Extensible database abstraction	No	No	No	No	No	Yes
Social login	Yes	Yes (only Facebook)	Yes	Yes (only Facebook)	Yes	Yes
Extensible federated authentication	No	No	No	No	No	Yes
Web analytics	Yes	No	No	No	No	Yes

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 solutions for those problems defined below:

1.3.1 Difficulties in using numerous technologies for the development

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. When it comes to the classified web development, even though the core functionalities are similar, developer has to go through different technologies and learn each of the relevant ones to implement the functionalities.

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

1.3.2 Developing same functionalities repetitively

One of the main problems faced by developers is to develop the same functionalities again and again due to the lack of classified framework. Most classified web sites have the same functionalities such as posting an advertisement, viewing an advertisement, deleting an advertisement, editing an advertisement etc. If the developer gets numerous classified web sites to work on, he/she has to develop all the functionalities from the scratch to fulfill the requirements of different classified websites. The developer burden can get decreased by a providing a proper template of functionalities to the developer to carry out the implementation.

1.3.3 Restriction of using preferred database vendor

Most of the development frameworks don't support multiple database vendors. Because of that, developer has to stick to the database vendor which the framework is supporting. If developer wants to get it connected to a different database vendor, he/she has to implement the support from the scratch using knowledge and programming skills. This leads to time consuming tasks and ultimately this leads to a cause of developer frustration.

1.3.4 Difficulties in providing third party identity provider support

Web development frameworks which are quite popular nowadays provide the facility of getting the development library externally to implement the identity provider support and integrate with the framework. But most of those external libraries don't provide the functionalities to many identity providers. Rather it provides just one or two popular social identity providers. For the developers who prefer to extend the identity provider support, could not be fulfill with the existing frameworks.

In the classified domain, identity provider support has become an essential part for a website. So, in the development process, if developer has to go through an external library to integrate with the framework in order to get the facility, it will consume more time and more effort.

1.3.5 Difficulties in analysing web log data

When it comes to classified web sites, the users who are using the website and data which are storing across the network is larger than normal websites. In order to optimize the web application to work perfectly without any interruption, network traffic should be minimized. For that, there should be a tool to analyze the data and get important decisions to maintain the website for future purposes. Most frameworks don't come up with in built web analytics module. Therefore, developers are forced to use analytics tool available in the internet to do the analyzing part. To use a new analytics tool, is risky and more time consuming than using a tool which developer has used before. Otherwise there will be a big learning curve around the developer to learn new tools.

1.3.6 Difficulties in identifying suspicious logins

Identifying suspicious or risky logins is important in the classified website domain as they have a huge audience of users using the website every day. Frameworks don't come up with a mechanism to identify suspicious logins. If the developer has to

implement the functionality from the scratch, it will be a complex and heavy task. The other option is to use an external product to identify the suspicious logins. Even for that it costs time and effort to integrate with the framework and if it's a new product, there will be a learning curve to learn the functionality beforehand.

1.4 Research Objectives

1.4.1 General Objective

General objective of this research is to overcome the issues mentioned under the research problem and develop an open source middleware framework for classified base web development. This middleware should handle the complexity of different technology layers, that are involve in the development process.

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.
4. Reduce the maintainability cost and time.

1.4.2 Specific Objectives

- Design and implement a component which supports multiple database management systems (MySQL [10], MSSQL [11], Oracle [12] 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 connect his/her website to a different database management system apart from the DBMSs listed in the framework, he/she will be provided an extension point to implement the connectivity. Sample queries for the common functions such as posting an advertisement, editing an advertisement, deleting an advertisement, viewing

advertisements should be handled according to the connected database management system.

- Design and implement the architecture component to extend federated authentication. By default, federated authentication supports Facebook, Twitter, Google, LinkedIn authentications. If a developer needs to provide authentication from a different identity provider, he/she will be provided a feature to add that authentication facility to the framework. So that a developer can easily implement the functions according to the particular identity provider by going through certain configurations. Developers should be able to easily enable or disable a connector, not going through a source code, but going through a configuration file. Main goal is to reduce the time and effort taken to customize the authentication point and implement the functionality more effectively.
- Design and implement a component to be integrated with the local authentication to identify suspicious and risk logins. This can be called as risk based authentication or adaptive authentication which considers certain risk levels to check whether the login attempt is risky or not. A developer should be able to customize the component according to his/her preference and get the output as the way he/she wanted. Mainly the component will consist with the risk level IP address, location, browser, device and time. If the calculated risk level is high, a developer can customize different verification levels to verify the user.
- Design and implement a component to use as a web analytic component to monitor and analyse the statistics of the website. A 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. These statistics can be used to make important decisions to maintain the website for future purposes. Graphs and charts can be used to show the change of statistics along with the time.

- 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 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.

2 METHODOLOGY

2.1 Methodology

In industry, there are many software development life cycle approaches which are used in the development process of the software. The system development life cycle (SDLC) also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing and deploying an information system. Software development life cycle consists with 6 phases.

1. Requirement gathering and analysis
2. Design
3. Implementation or coding
4. Testing
5. Deployment
6. Maintenance

As the methodology, this section explains each phase of the SDLC, with using the relevant process descriptions, up to the design phase. Implementation phase and testing phase will be covered in the next sub-section (i.e. 2.2). Work breakdown structure will explain the connection between each step of SDLC.



Figure 2.1: Work Breakdown Structure

2.1.1 Background Study

The research team has done a background study of the research to determine the feasibility of the project. Identifying the constraints, evaluating them and analyzing the capabilities of the proposed project are the main objectives of the feasibility study. It consists of,

1. Economic feasibility
2. Technical feasibility

Economic Feasibility

Determining the positive economic benefits that the proposed system would provide is the main purpose of the economic feasibility. As we were doing the background study, we realized that there was no research conducted in this domain and determined there will be a huge benefit by doing this project. Since it doesn't integrate with payable products to deal with, there were no external cost to determine to evaluate on benefits.

Technical Feasibility

Gathering knowledge about existing technologies and how they will be applicable for our proposed project is technical feasibility. The technical feasibility evaluates whether the project is able to be carried out with the existing equipment, skilled manpower and technologies and make sure the concept is technically possible.

2.1.2 Requirement Analysis

Requirement gathering and analysis is the crucial part of most projects. Before starting to implement the system, research group clarified whether the expected result can be achieved from the project. Since the research team is doing an external project belongs to WSO2 software company, research team met with the external supervisor several times and gathered the requirements.

And, research team has gone through followings to analyze and clarify the requirements.

1. Online Resources:

The research team has looked into the research papers regarding classified web development and determine the requirements to do the research project. As we went through the research papers, we realized there was no proper research projects carried out regarding classified web development. And we browse through the frameworks which can be used in classified web development.

2. Document analysis:

Since there were existing frameworks, we referred their documents to understand the functionalities and limitations of those frameworks. And also we referred the documents regarding the technologies we were expected to use and made sure those were the suitable ones for the implementation.

2.1.3 Design

One of the important phases in SDLC is design phase. The design stage takes approved requirement specification as its initial input. Output which are described in this phase are, features to build, its sub modules, how each module works and relationships between those sub modules. The design phase is started immediately after identifying all the requirements from the end user and to initiate the identification of user interfaces. For this purpose, we have designed set of UML diagrams such as use cases that describe how end users interact with the system.

User interfaces

Since it's a web development framework, there's no specific user interfaces to be built with the framework. But to make the development process easier, we're giving a set of interfaces related to the core functionalities and other module functionalities to build with the framework.

2.1.4 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 proposed 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 [13], the developer can use any front-end technologies to develop application front end.

System Architecture

The solution follows modularized SOA approach. The research group identifies four main components, those are listed below.

- Core framework module
- Extensible Database Abstraction
- Authentication module
- Analytics Generator module

This is an open source framework. Each module in this framework can be extensible or even customizable according to the developer's requirements. Figure 2.2 visualize the architecture of the implemented middleware framework.

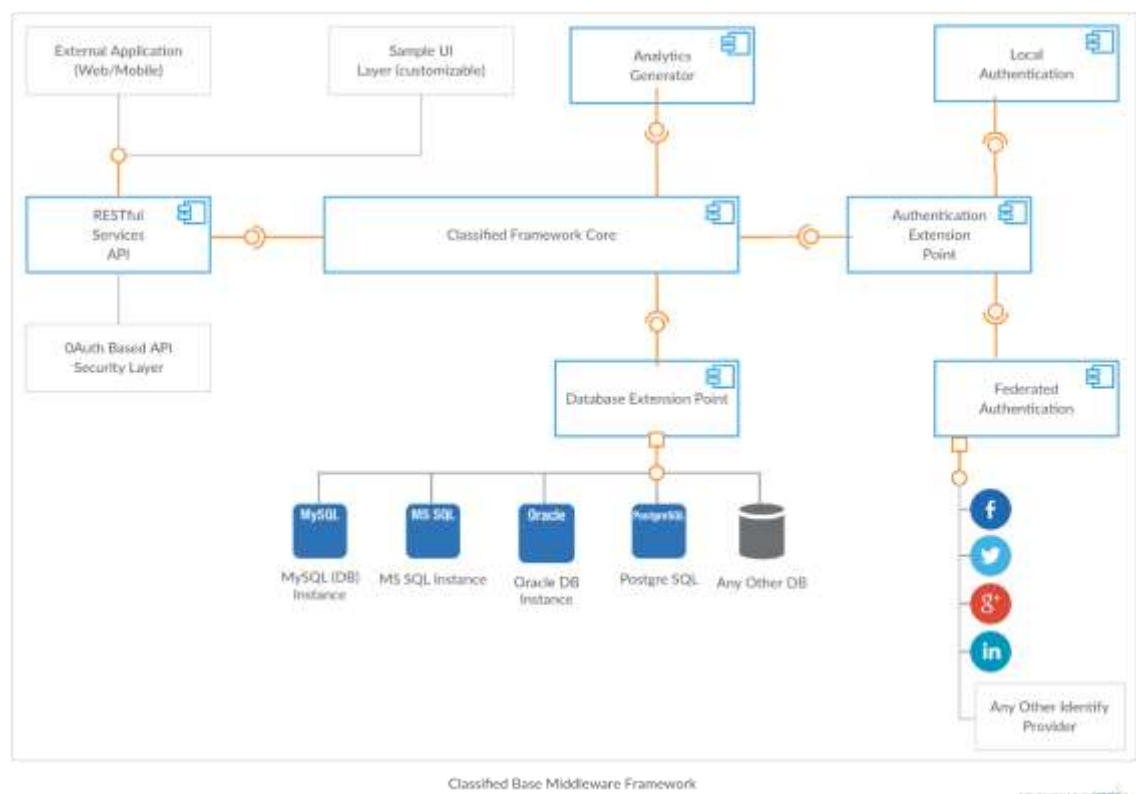


Figure 2.2: System Architecture

2.2 Testing and Implementation

2.2.1 Implementation

1. Core Framework

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).

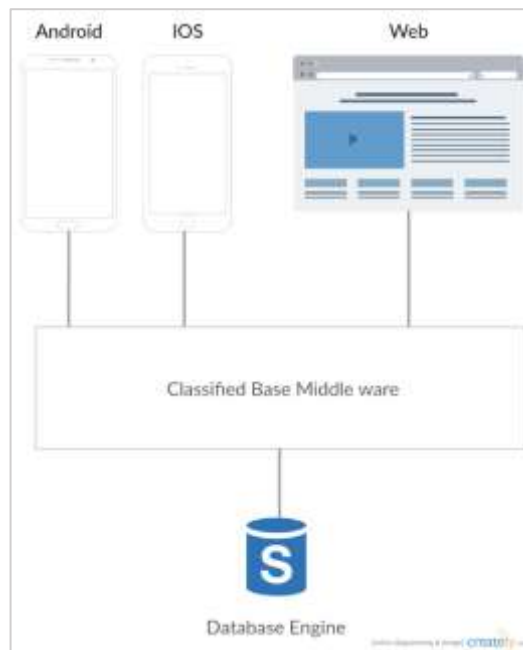


Figure 2.3: High level diagram of core framework

2. Database Extension Point

This module falls 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 2.4 shows a high-level architecture diagram of this module.

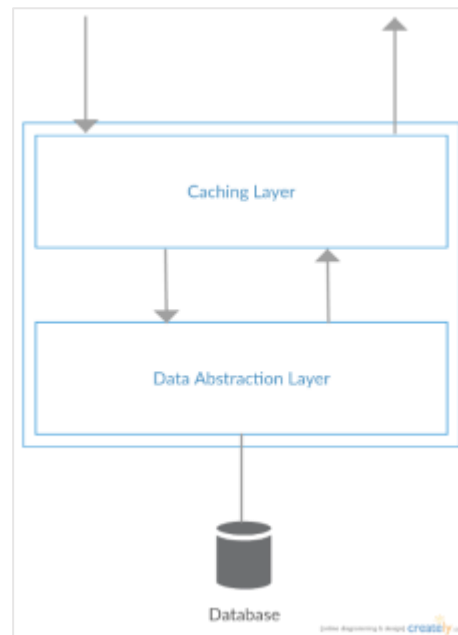


Figure 2.4: High level diagram of extensible database abstraction

- **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 [14]. 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 a 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 and this time can be configurable. If caching layer contains the response for requested search query then it directly response with that response. Otherwise, it will forward that query to data abstraction layer in order to get the result. When a result arrives, caching layer temporary store it for a given time period according to the configurations and forward that result as a response to the user.

Implementations of caching module done using ehcache caching library [15]. Ehcache is the most popular java caching library available at the moment. It's a commercial available caching library, and leading java frameworks such as Spring, Hibernate use this for caching. It also provides community level version as well, which we used to develop our caching layer.

Database vendors provide their own caching methodologies. Which means we can implement caching at data base level. Team neglect database level caching by considering two major facts which can make direct impact on our product.

- *Network traffic* - Production environments, database engines are deployed separately from the application layer. We can achieve high performance caching at the database level, but each data query application need to send request to the database server. This introduce network traffic, which can be caused to reduce the performance of overall framework.
- *Developer knowledge gap* – To configure database level caching developer required minimum intermediate level knowledge for a given database managements system. It is possible to give database level caching functionality with the middleware for the major vendors we support by default, although when someone is going to extend database supportability, they also required additional knowledge for that given database vendor as well, we consider this situation will kill developer usability of this framework.

- **Data Abstraction Layer**

One of main concern in this framework is extensible Database support. By default, framework support MySQL, MSSQL, Oracle and PostgreSQL. The architecture we followed for develop this layer is very simple as well as can be extensible quickly.

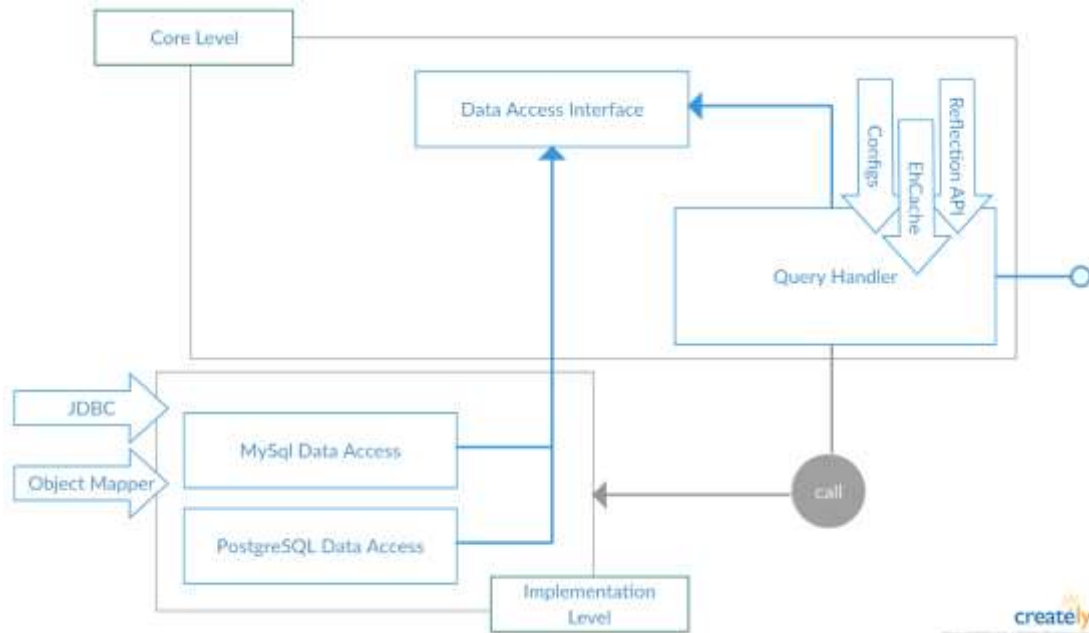
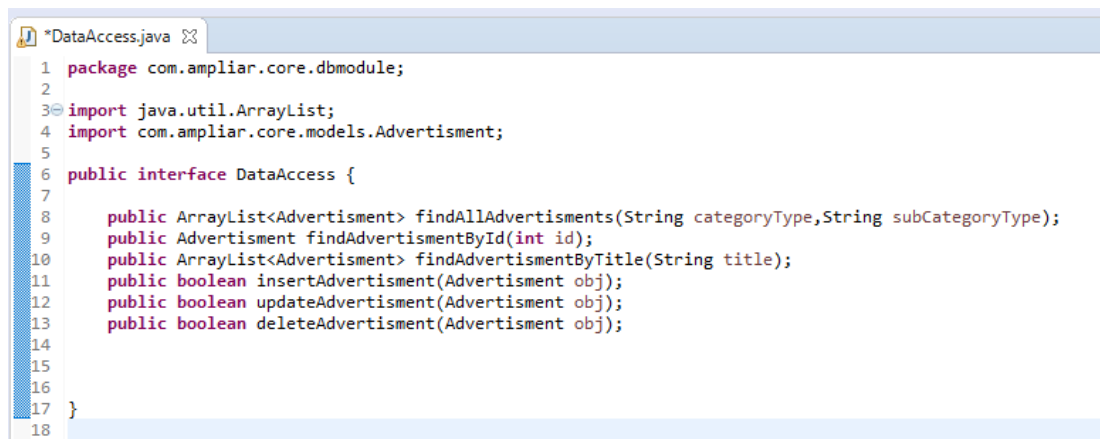


Figure 2.5: Architecture diagram of database extension point

We divide this architecture into two levels. One is the core level, this contains abstract level classes and definitions. We do not recommend any code modifications in this level. Here we can see one interface and one class. “Data Access” is an interface, it defines all for each operation. Any data CRUD process can identify as an operation in here. Therefore search, inserting, update, delete of advertisement or user are operations in this interface. Other class is the “Query Handler”. This class is an implementation of “Data Access Interface” and it expose this module services to outside. Basically, when another module or controller want data access, it can be achieved by calling a relevant method provided by this “Query Handler”.



```

1 package com.ampliar.core.dbmodule;
2
3 import java.util.ArrayList;
4 import com.ampliar.core.models.Advertisement;
5
6 public interface DataAccess {
7
8     public ArrayList<Advertisement> findAllAdvertisements(String categoryType,String subCategoryType);
9     public Advertisement findAdvertisementById(int id);
10    public ArrayList<Advertisement> findAdvertisementByTitle(String title);
11    public boolean insertAdvertisement(Advertisement obj);
12    public boolean updateAdvertisement(Advertisement obj);
13    public boolean deleteAdvertisement(Advertisement obj);
14
15
16
17 }
18

```

Figure 2.6: Data access interface

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

Figure 2.7: Sample use of query handler

Query Handler hide the complexity of framework performing various tasks. This class able to identify database vendor dynamically. First it reads the database configuration file and get dbms property. Using that dbms property value it will create instance of relevant DBMS vendor access class at the run time by using java reflection API. Since this is the main entry point of the whole module team apply caching library to this class. Each method in this class are wrapped by ehcache library. Ehcache automatically analyses each method when they invoked. If method previously called and available in cache pool, then ehcache omit the method execution, and return the result available in its cache pool, otherwise it will let the method to execute and then cache the result. By default, we provide ehcache with basic configurations, if developer wish to configure it more, ehcache configuration file available at config folder.

The second level of this module is implementation level. Each class at this level use JDBC for get database connections and Object mapper to crate POJOs using result set. We provide this level for to do customizations of the module. By default, vendor specific Data Access classes are implemented at here. We recommend implementation of all vendor specific data access classes at here. As previously described Data Access

Interface defined generic methods for generic database structure of a classified web application. In each class implements this interface should implements those methods. For example, MySQL Data access class implements Data Abstraction interface, then this class needs to implement all methods according to MySQL 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.

```

66 public Advertisement findAdvertisementById(int id) {
67     Class params[] = new Class[2];
68     params[0] = String.class;
69     params[1] = String.class;
70
71     Class<?> clazz;
72     try {
73
74         clazz = Class.forName("com.ampliar.dbmodule." + props.getProperty("dbms") + "DataAccess");
75         Method findAdvertisementById = clazz.getDeclaredMethod("findAdvertisementById", Integer.TYPE);
76         Object obj = clazz.newInstance();
77         return (Advertisement) findAdvertisementById.invoke(obj, id);
78
79
80
81     } catch (ClassNotFoundException e) {
82         // TODO Auto-generated catch block
83         e.printStackTrace();
84     } catch (NoSuchMethodException e) {
85         // TODO Auto-generated catch block
86         e.printStackTrace();
87     } catch (SecurityException e) {
88         // TODO Auto-generated catch block
89         e.printStackTrace();
90     } catch (InstantiationException e) {
91         // TODO Auto-generated catch block
92         e.printStackTrace();
93     } catch (IllegalAccessException e) {
94         // TODO Auto-generated catch block
95         e.printStackTrace();
96     } catch (IllegalArgumentException e) {
97         // TODO Auto-generated catch block
98         e.printStackTrace();
99     } catch (InvocationTargetException e) {
100        // TODO Auto-generated catch block
101        e.printStackTrace();
102    }
103
104    return null;
105
106 }

```

Figure 2.8: A sample method in query handler class

```

28 public class MySqlDataAccess implements DataAccess {
29
30     private Properties props;
31     private Connection con = null;
32     PreparedStatement pst = null;
33     ResultSet rs = null;
34
35     public MySqlDataAccess() {
36         System.out.println("mysql constructor executed");
37         if (this.con == null) {
38             getConnectionConfigurations();
39             try {
40                 // using mysql driver and configs create and return db connection
41                 Class.forName("com.mysql.jdbc.Driver");
42                 con = DriverManager.getConnection(
43                     "jdbc:mysql://" + props.getProperty("host") + ":" + props.getProperty("port") + "/"
44                     + props.getProperty("database"),
45                     props.getProperty("dbuser"), props.getProperty("dbpassword"));
46
47                 System.out.println("db connection established to mysql server");
48
49             } catch (ClassNotFoundException e) {
50                 // TODO Auto-generated catch block
51                 e.printStackTrace();
52             } catch (SQLException e) {
53                 // TODO Auto-generated catch block
54                 e.printStackTrace();
55             }
56         }
57     }
58
59     public ArrayList<Advertisement> findAllAdvertisements(String categoryType, String subCategoryType) {}
60
61     public boolean insertAdvertisement(Advertisement adv) {}
62
63     public boolean updateAdvertisement(Advertisement adv) {}
64
65     public boolean deleteAdvertisement(Advertisement adv) {}
66
67     private void getConnectionConfigurations() {}
68
69     public Advertisement findAdvertisementById(int id) {}
70
71     public ArrayList<Advertisement> findAdvertisementByTitle(String title) {}
72
73 }

```

Figure 2.9: Implementation of MySQL data access class

Previous paragraphs of this sub-chapter describe components and responsibilities of each components in depth. This paragraph describes the way those components work together. Process begins with the invoking of available methods in “Query Handler” class. First in controller or through another external component method all has to be done. With the method call “Query Handler” class executing begin. Since all selection methods are wrapped by ehcache library first ehcache will check given particular method previously invoked with given parameter values and the result is available in cache pool. If so ehcache will directly return the result by omitting “Query handler” class method call. Otherwise method execution starts. “Query handler” class create instance of particular DataAccess class by reading provided dbms configurations and java reflection API. Then invoke the same method which is available in the created DataAccess instance (since all DataAccess class as well as Query handler class implements Data Access Interface). With the creation of instance database connection

establish by using JDBC as well as relevant connections string dynamically generated via reading database config file. With the method invoke class access the database and get relevant data as a result set and return it to the Query handler class. Query handler converts this into a POJO or collection of POJOs and then return it to the controller or to the external module. Same time ehcache library insert this return values to its cache pool.

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 [16]. 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. Authentication Module

Authentication module consists with local authentication and federated authentication components. Federated authentication module deals with the third-party identity providers while local authentication deals with local logins to the website. To enhance the functionality of local authentication component, adaptive authentication functionality is integrated. Figure 2.6 visualize the high-level diagram of the authentication module.

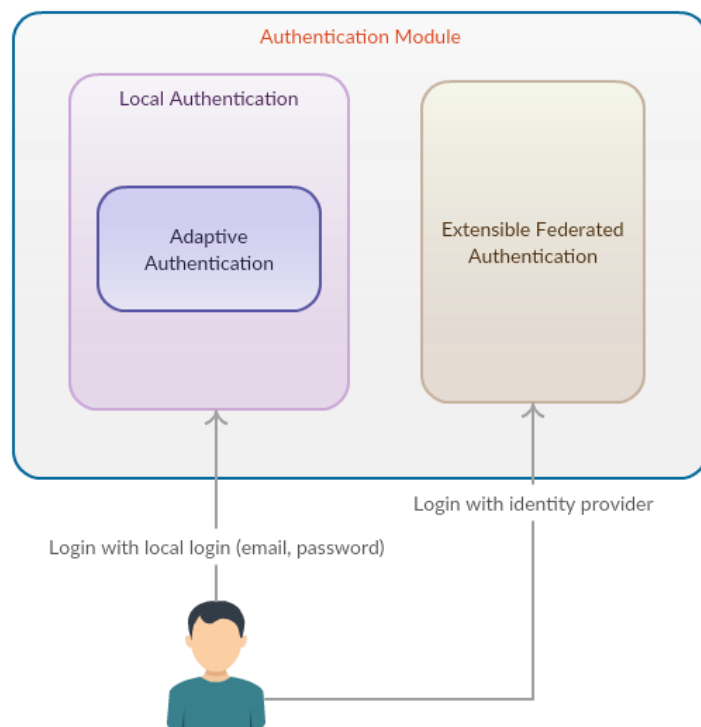


Figure 2.10: High level diagram of authentication module

- **Extensible federated authentication**

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" [17]. 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. Figure 2.7 shows the high-level diagram of OAuth process flow.

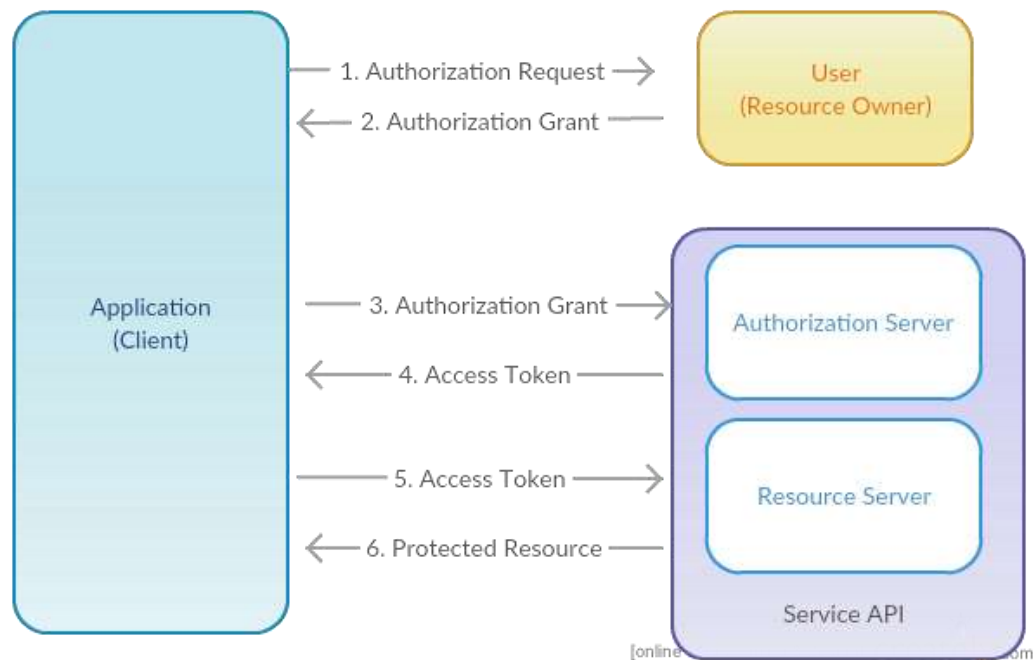


Figure 2.11: High level diagram of OAuth process flow

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

There is an authentication abstraction interface where we define the methods for the OAuth process flow. Those methods will be implemented inside the connectors according to the federated identity provider. We used JIT provisioning to create the users on the fly without creating user accounts in advance [18]. Following figure 2.8 shows the high-level diagram of Extensible federated authentication.

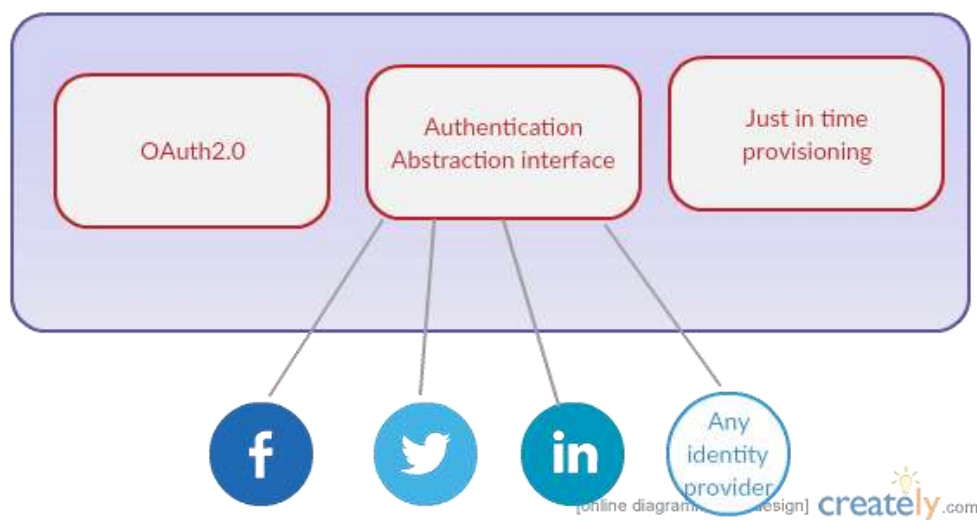


Figure 2.12: High level diagram of extensible federated authentication

For the implementation, we created an abstract class called authentication which define the main methods for the OAuth authorization flow. Those methods are to get the authorization code, get access token and get profile details respectively. Following is the code snippet of authentication abstraction class.

```

public abstract class Authentication {
    public static String TOKEN_ENDPOINT;
    public static String AUTH_ENDPOINT;
    public static String CLIENT_ID;
    public static String CLIENT_SECRET;
    public static String REDIRECT_URI;
    public static String ENABLE;

    public abstract String getAuthorizationcode(HttpServletRequest request);
    public abstract String getAccessToken(String output, HttpServletRequest request);
    public abstract void getprofiledetails(HttpServletRequest request, String UserInfoEndpoint, String accessToken);
}

```

Figure 2.13: Code snippet of authentication abstract class

Then if we want to implement a connector, we need to extend from authentication class and implement the methods according to the particular connector. Following are the connectors that we're providing.

```

public class Facebook extends Authentication {
    public static String GRANT_TYPE;
    public static String RESPONSE_TYPE;
    public static String SCOPE;

    public Facebook() {
        (...3 lines)
    }
    public static void assignvalues() {
        (...50 lines)
    }

    @Override
    public String getAuthorizationcode(HttpServletRequest request) { (...3 lines) }
    public void addAuthorizationHeader(HttpPost httpPost, String CLIENT_ID, String CLIENT_SECRET) {
        (...8 lines)
    }
    public HttpResponse makeAccessTokenRequest(HttpPost httpPost) throws IOException {
        (...6 lines)
    }
    public String handleAccessTokenRequest(HttpResponse httpResponse) throws IOException {
        (...7 lines)
    }
    @Override
    public String getAccessToken(String line, HttpServletRequest request) { (...21 lines) }
    @Override
    public void getprofiledetails(HttpServletRequest request, String FacebookUserInfoEndpoint, String accessToken) {
        (... lines)
    }
}

```

Figure 2.14: Facebook connector implementation

```

public class Linkedin extends Authentication{
    public static String GRANT_TYPE;
    public static String RESPONSE_TYPE;
    public static String SCOPE;
    public static String STATE;

    public Linkedin()
    {...3 lines }
    public static void assignvalues()
    {...53 lines }

    @Override
    public String getAuthorizationcode(HttpServletRequest request) {...3 lines }

    public String getAccessToken(String output,HttpServletRequest request) {...19 lines }

    @Override
    public void getprofiledetails(HttpServletRequest request,String linkedInUserInfoEndpoint,String accessToken)
}

```

Figure 2.15: Linkedin connector implementation

```

public class Google extends Authentication{
    public static String GRANT_TYPE;
    public static String RESPONSE_TYPE;
    public static String SCOPE;
    public static String APPROVAL_PROMPT;

    public Google()
    {...3 lines }
    public static void assignvalues()
    {...53 lines }

    @Override
    public String getAuthorizationcode(HttpServletRequest request) {...4 lines }

    public String makeAccessTokenRequest(String urlParameters,String tokenEndPoint) throws IOException
    {...13 lines }

    @Override
    public String getAccessToken(String line,HttpServletRequest request) {...5 lines }

    @Override
    public void getprofiledetails(HttpServletRequest request, String GoogleUserInfoEndpoint, String accessToken)
}

```

Figure 2.16: Google connector implementation

```

public class TwitterClass extends Authentication{
    public static String REQUEST_TOKEN_ENDPOINT;

    public TwitterClass()
    {...3 lines }
    public static void assignvalues()
    {...43 lines }

    @Override
    public String getAuthorizationcode(HttpServletRequest request) {...3 lines }
    public RequestToken getRequestToken(HttpServletRequest request, Twitter twitter) throws ServletException {...14 lines }

    public String getVerifier(HttpServletRequest request)
    {...4 lines }

    @Override
    public String getAccessToken(String verifier,HttpServletRequest request) {...14 lines }

    @Override
    public void getprofiledetails(HttpServletRequest request, String UserInfoEndpoint, String accessToken) {...3 lines }
}

```

Figure 2.17: Twitter connector implementation

Similar to these connector implementations, developer can extend and create a new connector by implementing all the abstract methods according to the way that particular connector works. Developer will be given a document to guide through the new connector creation. For the configurations related to connectors, developer will be given a configuration file, where he/she needs to modify the configurations according to the app they're creating on the developer website of a particular identity provider. Following is a screenshot of part of a XML configuration file which used in the extensible federated authentication.



```
<?xml version="1.0" encoding="windows-1252"?>
<connectors>
  <connector name="Linkedin">
    <enable>true</enable>
    <clientid>812j558k7kkgx</clientid>
    <clientsecret>KsshjjVz47p6lzcX</clientsecret>
    <authorization_endpoint>https://www.linkedin.com/uas/oauth2/authorization</authorization_endpoint>
    <token_endpoint>https://www.linkedin.com/uas/oauth2/accessToken</token_endpoint>
    <client_endpoint>http://localhost:8080/TestProject/Linkedincallback</client_endpoint>
    <grant_type>authorization_code</grant_type>
    <response_type>code</response_type>
    <scope>r_basicprofile r_emailaddress</scope>
    <state>123456</state>
  </connector>
  <connector name="Facebook">
    <enable>true</enable>
    <clientid>1741364576163534</clientid>
    <clientsecret>d23e5754aa74dad41e8eebfel6f3b764</clientsecret>
    <authorization_endpoint>https://www.facebook.com/dialog/oauth</authorization_endpoint>
    <token_endpoint>https://graph.facebook.com/oauth/access_token</token_endpoint>
    <client_endpoint>http://localhost:8080/TestProject/Facebookcallback</client_endpoint>
    <grant_type>authorization_code</grant_type>
    <response_type>code</response_type>
    <scope>public_profile email</scope>
  </connector>
  <connector name="Google">
    <enable>true</enable>
    <clientid>642110263906-sotsrsaeinlsudtfig5dhuvq97247ohe.apps.googleusercontent.com</clientid>
    <clientsecret>75fAn-mynWV6BAhRRCIHNPc5</clientsecret>
    <authorization_endpoint>https://accounts.google.com/o/oauth2/auth</authorization_endpoint>
  </connector>
</connectors>
```

Figure 2.18: Authentication configuration file

Developer only has to modify the configuration file, using his/her app configurations to get the functionality. If developer doesn't want any of identity providers he/she can disable that connector by making the enable configuration false. So that it will dynamically load the login page by reading the configuration file.

Once user logged in for the first time using an identity provider, an identification field will be selected (mostly email address) and user account will be created using the fields we're getting from the identity provider. A field called authenticator will be used to identify the users who are logged in using federated authentication.

- **Adaptive Authentication**

Adaptive authentication provides world-class security without impacting usability. That's because risk checks are done without users even being aware of it- and multi factor authentication is applied only risks are detected [19]. For the classified web development framework local authentication component, adaptive authentication is integrated considering following risk factors:

1. IP address
2. Geographical location
3. Device
4. Browser
5. Date/Time

To calculate the risk levels more effectively, a weighted scoring model was implemented under adaptive authentication component. Weight is divided among the risk factors and developer can decide which factors to give more priority and assign weights. By default, IP address, location and date/time risk factors will be given more priority. An average risk score will be calculated for every risk factor by going through the log records of past login attempts. If the total risk score is more than 70%, it will consider as a higher risk level and an OTP of 6 digits code will be generated and send to user's email address to verify the user. If the risk score is between 50% and 70%, it will consider as a middle level risk and pops up security questions which were given by the user at the registration stage. If the calculated total risk score is below 50%, it will consider as a minimal risk and redirect user to the home page.

Developer is given a user interface to configure the weight for the risk factors and also for the configurations related to location ranges and time ranges to consider when calculating risk scores.

4. Web Analytics Generator

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 and in modern days, more focus is given for fraud detection too based on web analytics [20], [21].



Figure 2.19: Web analytics process

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 etc [22], [23], [24], [25], [26].

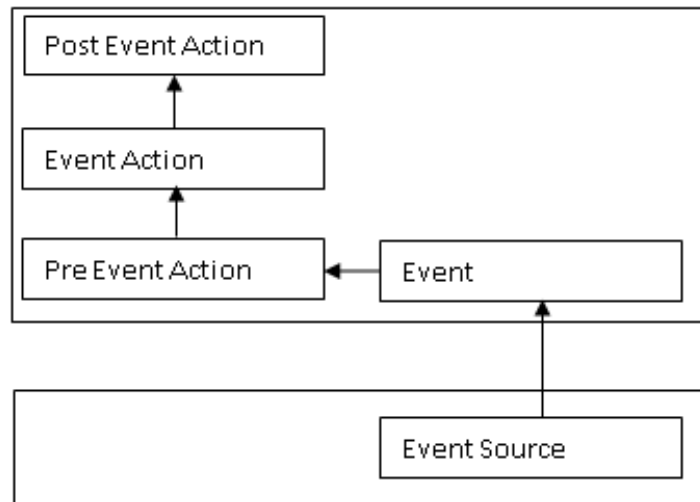


Figure 2.20: Event listener architecture

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.

To follow this architecture, first need to create a class for each and every event. Then those created even classes should extends the listener abstract class and then those classes can implement the pre-method and post method for the separate events. In the pre-method specify the actions need to be happened before the actual event is take place. For an example when we consider an event like posting an advertisement, the actual event is post the advertisement to the web site. So, before the customer is filling

the posting advertisement details. Therefore, the pre-method can call in that time. Inside that pre-method can validations of the form can be done. As well as the post method can call after the actual event takes place. Post-method can be used to publish an event and get the details of that event to the outside. Event details such as user IP, advertisement id, advertisement name, date, time, advertisement category can be recorded into a CSV file or recorded to the database for future analyzing purposes. Those files can be used for data mining and prediction techniques to predict about a particular event. Furthermore, this module is generating analytics reports regarding the advertisement views, advertisement posts, visitors and much more.

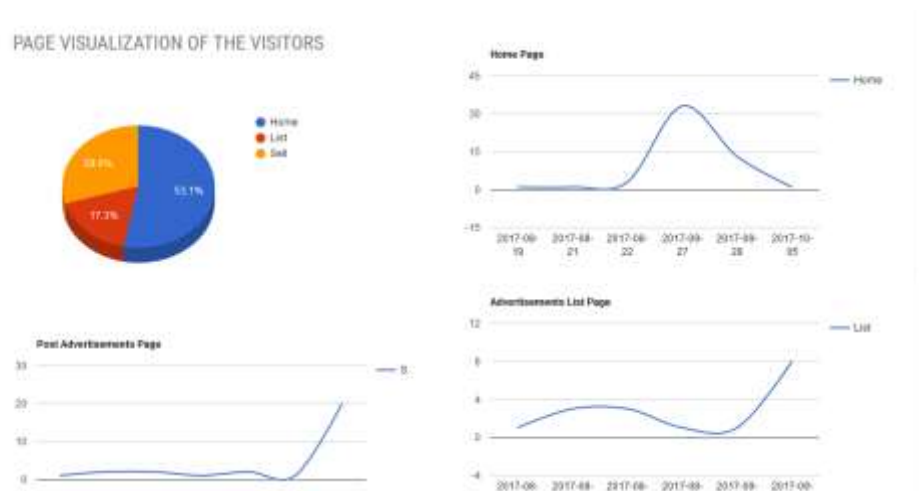


Figure 2.21: Page visualization of the visitors

2.2.2 Testing

Software testing is an important phase in the SDLC. Testing helps in identifying bugs and issues in the system and can take necessary actions to correct to fix those bugs. Testing has happened throughout the development starting from the requirement gathering phase. Testing has conducted in the implementation that always with line to the requirement specification. This can be used to provide information about the quality of the product under tests. There are recognized test levels can be identified under testing. They are unit testing, module testing, integration testing and the system testing.

Unit Testing

Unit testing is used to test the functionality of each module separately. They are tested alone and isolated. It tests the functionality of a specific section of code, at the function level. Some of the unit testing frameworks are, Junit and Nunit. Unit testing follows white box analysis.

Integration Testing

Integration testing combines all the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This tests how each module interacts with each other. Integration testing occurs after module testing and before validation testing. This takes the input as modules, which are unit tested.

System Testing

System testing tests the whole integrated system to evaluate the system's compliance with its requirements. We perform system testing to ensure that we have completed and achieved up to the expected quality and level of objectives.

2.3 Research Findings

During the research, team has identified that there were some classified web development frameworks with lots of limitations and weaknesses which cause low usage. The team found that existing classified web developing frameworks don't provide Extensible database abstraction, extensible federated authentication and built in web analytics generator facilities for the developer within the framework. "Ampliar" framework focuses more into extendibility of modules and reduce developer frustration.

This section covers the findings that are gained during our research, related to classified web development domain.

- **Extensible Database Abstraction**

Providing a feature to extend the database support functionality is one of the main features of the research. This module was developed using two levels, core level and implementation level. Developer was given the database support for major database vendors such as MySQL, MSSQL, Oracle and PostgreSQL. This module uses a query handler class to provide access point for other modules to use this data extension point. And module uses a dynamic class generator to dynamically identify the database vendor. When extending vendor support, developer will be able to work with a particular vendor. Developer can integrate a particular vendor through maven by going through certain configurations. Then developer can implement the data access class for the vendor by implementing the abstract methods.

- **Integration of Caching module to boost the performance**

Caching has become very important concept in distributed systems. Most of the times, in development environment, database server is deploying separately. In those situations, it's hard to get the database level caching benefits due to the network traffic introduced in between the application server and the database server. Again, if we setup caching at database level, when it comes to database extensions, developer is required the knowledge of that particular database vendor cache configurations. Because of that, the caching module was developed in the application level. And used ehcache as the caching library for the module.

- **Extensible Federated Authentication**

Research team has found out that many classified websites use different identity providers for the federated authentication and decided to provide an extensibility to the federated authentication where developer can easily extend the functionality. By going through different classified web sites, we have identified major identity providers and integrated the connectivity to the framework. Abstract class was provided with the methods which are going through the OAuth process flow and made the connectors implement the methods according to their process. Since the goal is to reduce the developer time and effort, configuration file was given to the developer to easily configure the connector configurations.

- **Weighted Risk Score model to use in adaptive authentication**

Nowadays risk based authentication is popular among websites as it identifies suspicious logins and block users from using the website until the user verifies about his/her identification. Risk scoring model was developed considering the risk factors IP address, device, browser, location and date/time. Since the effect of each factor has to be considered and priorities should be given, weighted concept was introduced. Using the weight developer can configure which risk factors to give more priority and which factors to give less priority. According to the calculated risk, decisions can be taken to verify the user. Developer can configure different verification levels based on the risk score.

- **Web analytics generator to analyze the web data**

Research team has found out that the classified website audience is huge and there will be many users accessing the website. Due to the number of visitors there can be a network traffic to access the resources from the website. Because of that, it's important to get web log data of certain point in time and analyze them to take decisions for future purposes of the website.

3 RESULTS AND DISCUSSION

3.1 Results

The main goal of this research is to provide 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.

Though the final outcome of the research project, external parties related to the project development can get an idea of the final outcomes and the quality of the final product, which includes verification and validation.

We have presented the concept and the techniques that have been used in the process of the development of the “Ampliar” framework. Through our research paper, we have clearly pointed out the problems which were faced by the developers in developing classified websites and these problems have been addressed by this solution. By accepting the research paper from “**National Information Technology Conference 2017**”, the domain experts and community have accepted this solution as a proper solution which will address major problem areas.

This subsection provides evidence to the implementation results and the solutions provided for the identified research problems. Since it’s a development framework, developers directly should interact with the configuration files. For some cases, developer is provided with a user interface to reduce the complexity of the functionality.

Following is the user interface provided for the login with local authentication and federated authentication.

The screenshot shows a web application interface for 'THE AUTOSTARS'. At the top, there is a dark header bar with a logo on the left, navigation links 'home' and 'developer settings' in the center, and 'login' and 'register' buttons on the right. The main content area is a light gray box titled 'sign-in into your account'. It contains a form with two input fields: 'Email' (placeholder: 'Enter your email address') and 'Password' (placeholder: 'Enter your password'). Below the password field is a checkbox for 'Remember me' and a red link for 'Forgot password?'. At the bottom of the form is a red button labeled 'LOG IN' and a link 'No account? Register'. Below the form are four social media login buttons: 'Login with Facebook', 'Login with Twitter', 'Login with LinkedIn', and 'Login with Google+'.

Figure 3.1: Login interface

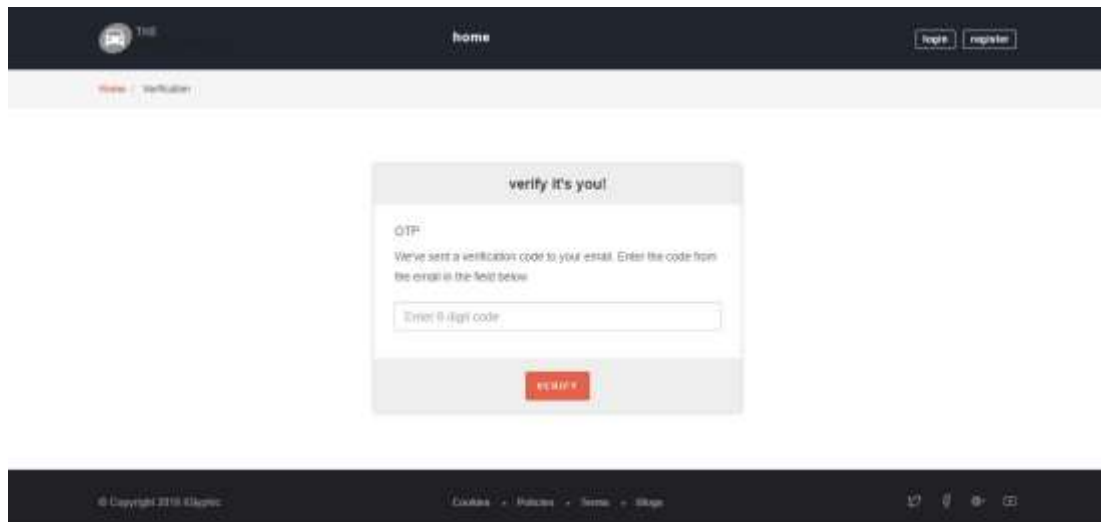
This login interface can be dynamically load by modifying the enable configuration in the authentication configuration file. If developer doesn't want the google connector to be integrated with the authentication, he/she only should do is modifying the enable configuration under google connector as false.

To configure the adaptive authentication, developer will be given a user interface to easily enter the configurations. So that, he doesn't have to go through the code to make changes in the configuration, this make the developer tasks more flexible.

The screenshot shows a web application interface for 'adaptive authentication settings- developer'. The interface has a dark header bar with a logo on the left, a 'home' link in the center, and 'login' and 'register' buttons on the right. The main content area is a light gray box titled 'adaptive authentication settings- developer'. It contains several configuration options, each with a label, an input field, and a unit indicator: 'IP address weight' (20.0 %), 'Browser weight' (20.0 %), 'Device weight' (10.0 %), 'Date/Time weight' (20.0 %), 'Time range' (3.0 hours), 'Location weight' (30.0 %), 'Latitude range' (2.0), and 'Longitude range' (1.0). There is also a 'To consider about the events that are established X hours before/after' label. At the bottom is a red button labeled 'SAVE'.

Figure 3.2: Developer settings interface

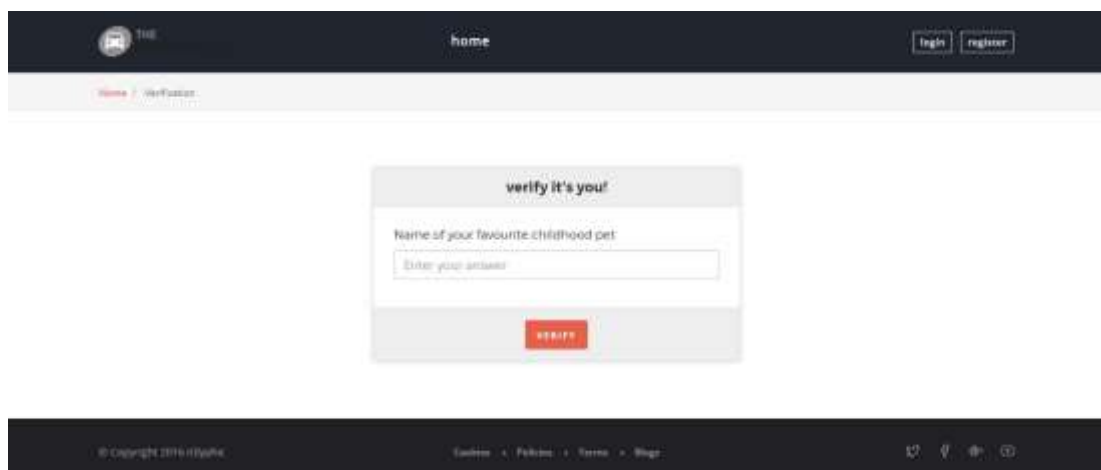
If the risk score calculated for the adaptive authentication is higher, OTP will generate and send to user's email. Following screen appears to enter the 6-digit code to verify the user.



The screenshot shows a web application interface for OTP verification. At the top, there is a dark navigation bar with a logo on the left, the word "home" in the center, and "login" and "register" buttons on the right. Below the navigation bar is a light gray breadcrumb trail showing "Home" and "Verification". The main content area features a white card with a gray header that says "verify it's you!". Below the header, the text "OTP" is displayed, followed by the instruction: "We've sent a verification code to your email. Enter the code from the email in the field below:". There is a text input field labeled "Enter 6-digit code" and a red "verify" button at the bottom of the card. The footer of the page is dark and contains copyright information "© Copyright 2016 Klyptec", a list of links "Contact", "Privacy", "Terms", and "Blog", and social media icons on the right.

Figure 3.3: OTP verification interface

If the risk score is in the middle level, user will be prompted the security question which was given by the user at the registration stage.



The screenshot shows a web application interface for security question verification. It has the same dark navigation bar and light gray breadcrumb trail as Figure 3.3. The main content area features a white card with a gray header that says "verify it's you!". Below the header, the text "Name of your favourite childhood pet" is displayed, followed by a text input field labeled "Enter your answer" and a red "verify" button at the bottom of the card. The footer of the page is dark and contains the same copyright information, links, and social media icons as Figure 3.3.

Figure 3.4: Security Question verification

Following is the sample screenshot of the ehcache configuration file. This is a XML base configuration file and located in src/resources directory. By default, component provide essential caching features of ehcache. Developers can configure this file the way they want.

```
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ehcache.xsd"
  updateCheck="true"
  monitoring="autodetect"
  dynamicConfig="true">

  <diskStore path="java.io.tmpdir" />

  <cache name="movieFindCache"
    maxEntriesLocalHeap="10000"
    maxEntriesLocalDisk="1000"
    eternal="false"
    diskSpoolBufferSizeMB="20"
    timeToIdleSeconds="300" timeToLiveSeconds="600"
    memoryStoreEvictionPolicy="LFU"
    transactionalMode="off">
    <persistence strategy="localTempSwap" />
  </cache>

</ehcache>
```

Figure 3.5: Ehcache configuration file

User analytics on page views and visitors are visualize by web analytics module and admin panel can view the analytics in following manner.

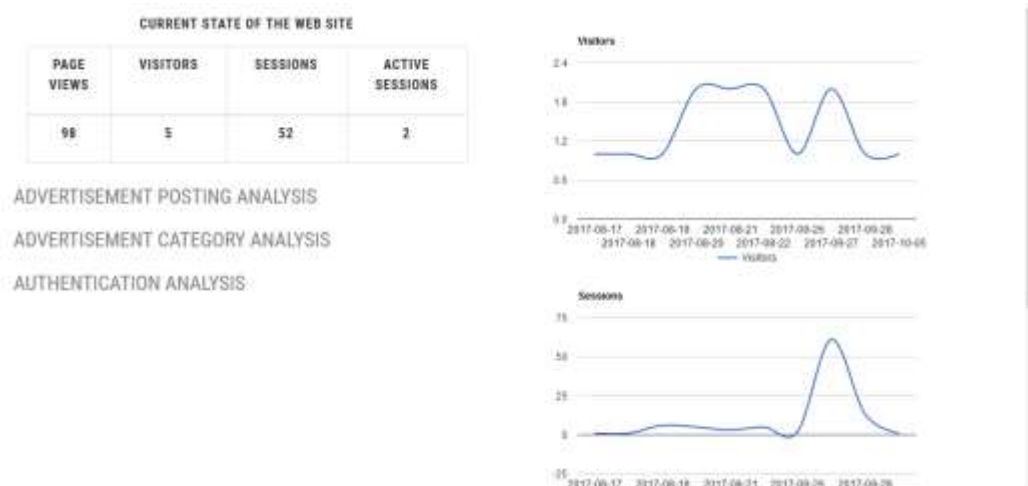


Figure 3.6: Visitor analysis screen

3.2 Discussion

This chapter includes a discussion regarding the total progress or the results shown in the previous chapter and the accuracy of the work done.

We have developed a classified web development framework which can be used by novice developers as well as professional developers. It consists of advanced features to support multiple database vendors, extensible federated authentication, web analytics generator and core functionalities which are identified by the research team as lack of the existing frameworks. These modules alone are providing the customizable features to customize the component in the way developer wants. This enhances the freedom and the flexibility that a developer can have when using the framework.

In order to use the framework, developers can just download the framework from the website and run in the java platform. To customize the functionalities developer can refer to the developer guides we're providing with the framework. These components can be further developed to enhance the functionalities and make the use of extendibility. Since this is an open source framework, others can also be part of the development and can contribute to the future development.

Framework was developed using object oriented concepts and have used standard web interfaces and configurations during the development. This framework also can be integrated with any other java web development frameworks, so that developers shouldn't worry about working with their preferred web development framework.

Anyone who is having little knowledge about programming languages can use this framework, since it doesn't directly interact with the coding. If a professional developer is using this framework, he/she can enhance the features and directly involve with the development if/she wishes.

4 COMMERCIALIZATION OF THE PRODUCT

4.1 Position of the Market

Online classifieds have been popular among users and organization for decades. Since traditional newspapers have limitations over the online classifieds, people tend to use online classifieds than traditional newspapers. Because of the huge audience that online classifieds have, many classified websites are developing among organizations as well as individual users. But there's no existing frameworks which can be used with the features of supporting multiple database vendors, multiple identity providers, adaptive authentication and in built analytical engine with a little knowledge of programming language. "Ampliar" classified web development framework mainly focuses on reducing the learning curve around the developer. 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.

4.2 Differentiate from the competition

Unlike the other frameworks, "Ampliar" framework supports for multiple database vendors. If developer wishes to work with a different database vendor, other than the integrated vendors, he/she can go through easy configurations and get the facility using our framework.

Unlike the other frameworks, "Ampliar" framework support for extensible federated authentication and adaptive authentication. If developer wishes to add a different identity provider than the ones that are integrated with the framework, he/she has to go through set of easy configurations and implement the connectivity for the website. And also if developer wishes to change the configurations to adaptive authentication, he/she can go through the developer user interface and do the necessary changes.

Unlike the other frameworks, “Ampliar” framework support for customizable web analytics generator. This component follows a listener architecture, which consists of pre-method listener and post-method listener. Every event can capture by this event listener architecture and publish the event details to outside for analyzing.

4.3 Marketing Plan

4.3.1 Need of the Product

Benefits for the developer

- Developer can easily use the core framework to develop the major functionalities, he doesn't have to develop from the scratch.
- Developer has the opportunity to choose his preference database vendor since ampliar framework supports multiple database vendors.
- Developer can easily plug a custom data store to the website.
- Developer can choose a different identity provider (apart from popular) for federated authentication.
- Developer can use the in -built analytical engine to develop the analytics for the website.
- Reduce developer burden, complexity and knowledge gap during the development process.
- Reduce development time and cost of a classified base web development project.

The classified web developing framework providing for the classified web developing organizations to maintain and reduce their time to develop their online classified web sites. This Ampliar framework offers the basic details regarding the classified web development.

4.3.2 Marketing Strategy

Since the component modules of framework can be used as plugins for other developments also, marketing them as separate component can also be benefited to market the framework as a whole. Because these features are a lack of every framework, not only in the classified web domain but in every other domain. So, for the developers who are finding a plugin for extensible database abstraction or extensible federated authentication or customizable adaptive authentication or customizable web analytic engine can be benefited by these modules and can increase the audience to market the “Ampliar” framework.

4.3.3 Target Market

The selected marketing strategy will be based mainly on making the right services available to the right target customer. The team will ensure that the prices of the services take into consideration organizations' and peoples' budgets, and that these people know that the team exist, appreciate the value of the services, and how to contact the team. The marketing will convey the sense of quality in every picture, every promotion, and every publication. The intension of the team will be to target those developers looking for reduce the developer burden, complexity and knowledge gap during the development process.

Pre-Launch Marketing

- Social Media marketing
- Referral marketing
- Blog

Furthermore, this framework can sell for free with self-hosting and we can give price ranges with hosting facility.

5 MEMBERS INDIVIDUAL CONTRIBUTION

Liyanaarachchi I.H.

- Develop an extensible federated authentication module for the framework

This module involves with integrating the major identity providers, Facebook, Twitter, LinkedIn and google as one component and provide extensibility to implement any identity provider connector. Since we use the OAuth 2.0 authorization framework, authorization process flow is divided to separate methods in the authorization abstract interface. When implementing a connector, what developer has to is to extend the from the abstract interface and implement the methods according to the identity provider functionality. A configuration file is maintained to store all the configurations related to the connectors such as, client id, client secret, authorization end point, client end point etc. In order to make the functionality work, what developer has to do is get the framework and change the configuration file elements into his/her developer application credentials. Furthermore, connector enable element in the configuration file is used to enable or disable the connector from the classified website.

- Develop a risk scoring model to calculate risk score for adaptive authentication

Adaptive authentication is a risk based authentication which is popular among social websites. Many products were developed which can be integrated with websites to calculate risks. This component is based on 5 risk factors, which are ip address, location, device, browser and date/time. For each risk factor, weight is assigned by the developer. When an average risk score is calculated for a risk factor, it is multiplied by the weight which was assigned to it. The total risk score is checked with the higher level, medium level and lower level to make sure it's a suspicious attempt. If the risk level is high, OTP will be sent to the user's email and user can use that verify his/her identity. For a medium risk, security question will be prompt to the user. These risk levels can be further expanded by the developer.

W.M N Radith

- Development of Data Extension point for the middleware framework

One of the most critical and impotent module the framework. Module contains two sub-modules as described in methodology chapter. Caching module developed using ehcache, which is the most popular caching library available today. This module contains septate configuration file which is located in the config folder for manage cache pool. By default, basic configurations are provided and can be customize according to developer needs. The next module is the database abstraction module, spend most of time on this sub-module for achieve possible simplest architecture. This also maintain separate configuration file for database configurations. All runtime object creations are centralized on that configuration file. All vendor base data access classes are independent from other classes, therefore extend can be easily done, in other words developer only needs worry about extensibility. All caching and other functions are handled by the module automatically. To store data in cache pool and return as POJOs, crate a separate Object mapping helper class again using java reflection. This will convert result set into POJOs. Please refer chapter 2.2.1 sub-topic 2 for get more in-depth details about the module implementation.

K.S.D.A Kulathunga

- Development of Core Framework for Ampliar

Design of Core Framework Architecture (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 (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.

B.R.K.S. Kumari

- Development of Web Analytics 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. This will be based on web log data mining. The main objective is to design and implement the web analytic component to monitor and analyze the statistics of the website. 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. The analytics component is focusing on the events handle by the user. It follows a listener architecture to develop the analytics event listener concept which can create listener for the events happening in the web site. Each event consists with pre, actual and post event. By using those event categorization developers will be capable of analyzing the event by calling relevant event listener when it's needed. Then developer may use the existing web analytics library or can be configured or extended with external existing analytics libraries to analyze their classified web site statistically. Moreover, reports can be generated using the generated analytics.

6 CONCLUSION

Analyzing the requirements, studying existing technologies to implement this system, gathering ideas from resourceful authorities and conducting literature reviews were done related to extensible database abstraction, extensible federated authentication, adaptive authentication and web analytical engines in order to achieve objectives of this research.

In this report, we have discussed about the use of our framework as a classified web development framework which will reduce the time, effort and knowledge gap of developers in developing classified web sites. Moreover, how developers can extend the functionalities and enhance the framework was discussed in the report.

By using our framework, developers can get the flexibility in using any database vendor and any identity provider support to be integrated with the framework. Instead of using third-party analytical engine, framework gives the feature of a built in analytical engine which analyze data and generate reports. None of the existing frameworks support these functionalities as well as the customizability.

We have achieved the level of sophistication we wanted to achieve in our framework and even surpassed it. Through our framework we want to provide the best possible service to the developer in classified web development and “ampliar” wants to make sure that they’re able to consume the provided services for efficiently developing their projects with a minimum learning curve and lead their projects towards successful delivery.

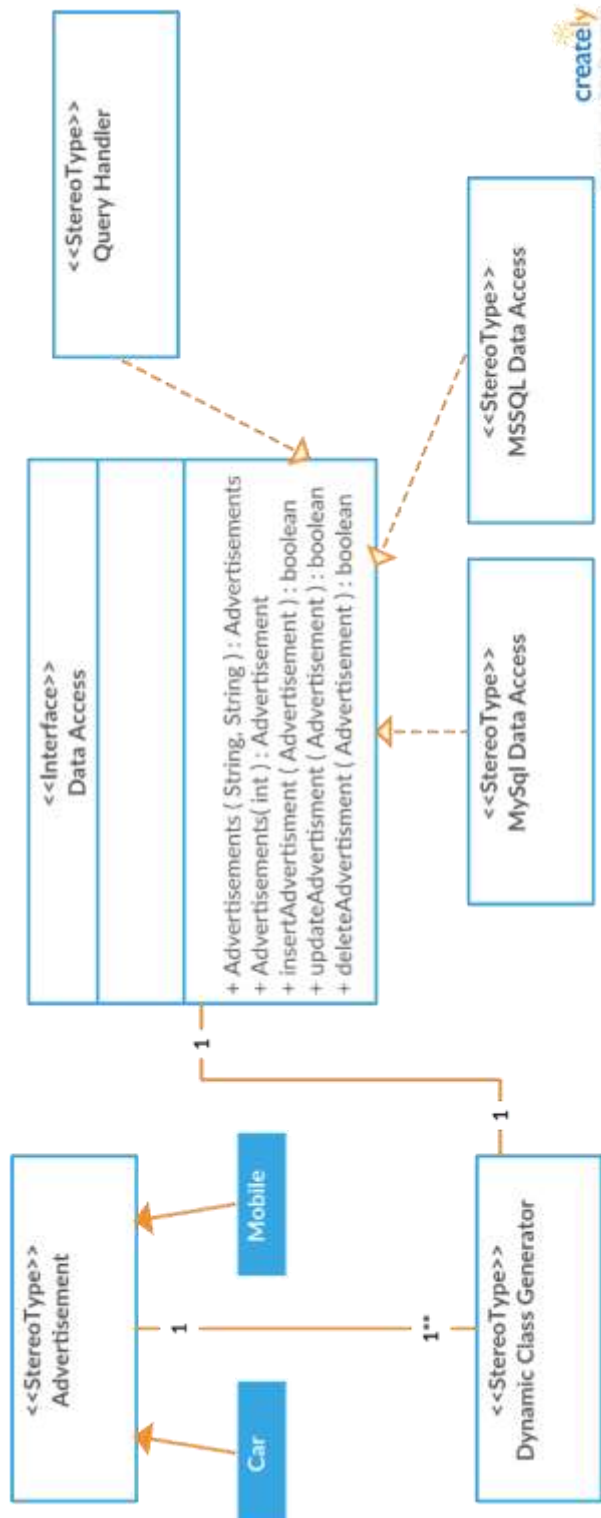
This framework features can be further developed to provide more advanced functionalities for the developer in later stages.

REFERENCES

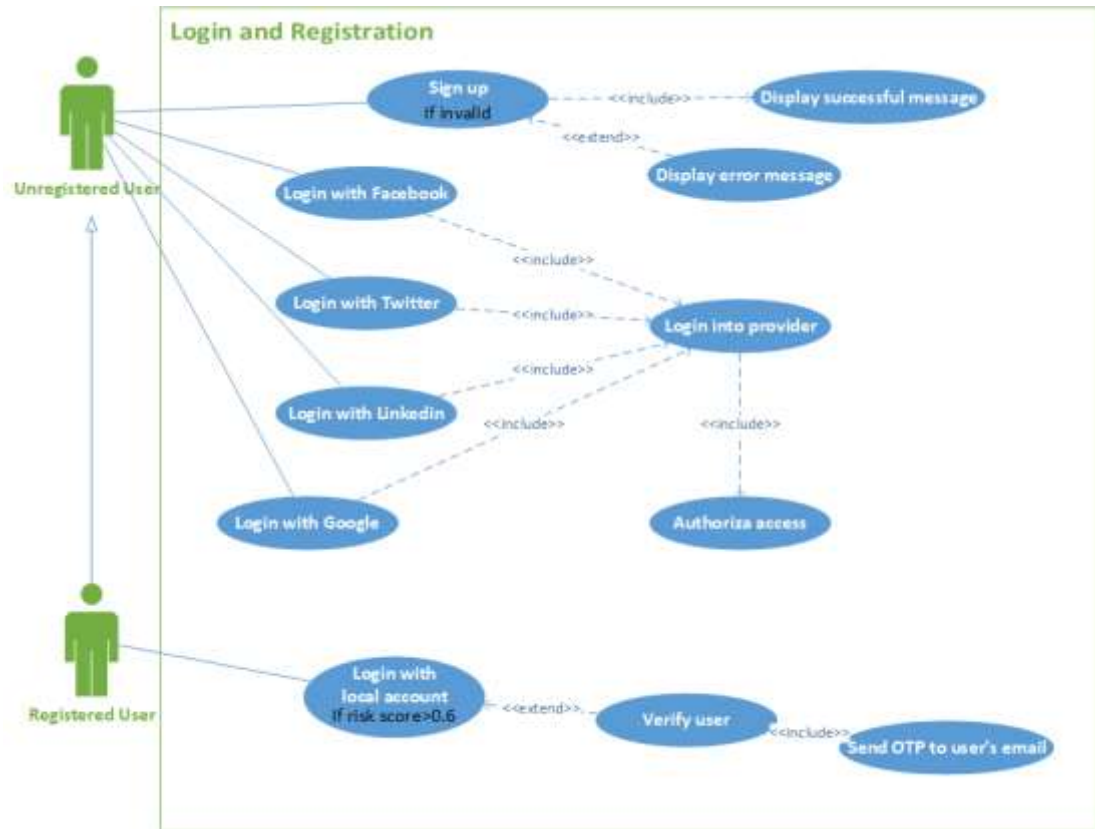
- [1] S. Diaz, "On the Internet, A Tangled Web Of Classified Ads," Washington Post, Friday August 2007. [Online]. Available: <http://www.washingtonpost.com/wp-dyn/content/article/2007/08/30/AR2007083002046.html?hpid=sec-tech>.
- [2] "Classified advertising," wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Classified_advertising#cite_note-4.
- [3] T. W. Lin and E. J. Blocher, Cost Management: A Strategic Emphasis, Tata McGraw-Hill Education, 2006.
- [4] "Yclas Home," [Online]. Available: <https://yclas.com/>. [Accessed 15 February 2017].
- [5] "Flyntax Home," [Online]. Available: <https://www.flynax.com/>. [Accessed 17 February 2017].
- [6] "Oxyclassifieds Home," [Online]. Available: <https://www.oxyclassifieds.com>. [Accessed 17 February 2017].
- [7] "Titan Classifieds Home," [Online]. Available: <http://www.titanclassifieds.com/>. [Accessed 17 February 2017].
- [8] "Oscass Home," [Online]. Available: <https://osclass.org/>. [Accessed 17 February 2017].
- [9] "Classipress Home," [Online]. Available: <https://wordpress.org/plugins/tags/classipress/>. [Accessed 17 February 2017].
- [10] M. Commiunity, "MySQL," MySQL, [Online]. Available: <https://www.mysql.com/>.
- [11] Microsoft, "SQL Server," Microsoft, [Online]. Available: <https://www.microsoft.com/en-us/sql-server/sql-server-2016>.
- [12] Oracle, "Oracle Database," Oracle, [Online]. Available: <http://www.oracle.com/technetwork/database/database-technologies/sql/overview/index.html>.
- [13] A. PUDER, K. ROMER and F. PILHOFER, DISTRIBUTED SYSTEMS ARCHITECTURE A Middleware Approach, Elsevier, 2006.

- [14] 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>.
- [15] "EHCACHE Home," [Online]. Available: <http://www.ehcache.org/>. [Accessed 7 February 2017].
- [16] 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.
- [17] "The OAuth 2.0 Authorization Framework," [Online]. Available: <https://tools.ietf.org/html/rfc6749>. [Accessed 5 February 2017].
- [18] "Design and implement just-in-time provisioning with SAML 2.0," [Online]. Available: <https://www.ibm.com/developerworks/library/se-jitp/>. [Accessed 12 February 2017].
- [19] "Adaptive Authentication," [Online]. Available: <https://www.secureauth.com/products/secureauth-idp/adaptive-authentication>.
- [20] B. Plaza, "Monitoring web traffic source effectiveness with Google Analytics: An experiment with time series," *Aslib Proceedings*, vol. 61, no. 5, pp. 474-482, 2009.
- [21] J. T. S. Q. a. M. Sriganesh, "Real-time credit card fraud detection using computational intelligence," *Expert Systems with Applications*, vol. 35, p. 1721–1732, 2008.
- [22] "Google Analytics API," [Online]. Available: <https://developers.google.com/analytics/>. [Accessed 7 February 2017].
- [23] "Woopra Home," [Online]. Available: <https://www.woopra.com>. [Accessed 17 February 2017].
- [24] "Metrics," [Online]. Available: <https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-metrics.html>. [Accessed 15 February 2017].
- [25] "Mint Home," [Online]. Available: <https://haveamint.com/>. [Accessed 17 February 2017].
- [26] "Chartbeat Home," [Online]. Available: <https://chartbeat.com/>. [Accessed 15 February 2017].

APPENDIX A: Class diagram for Database Abstraction



APPENDIX B: Use case diagram for Authentication component

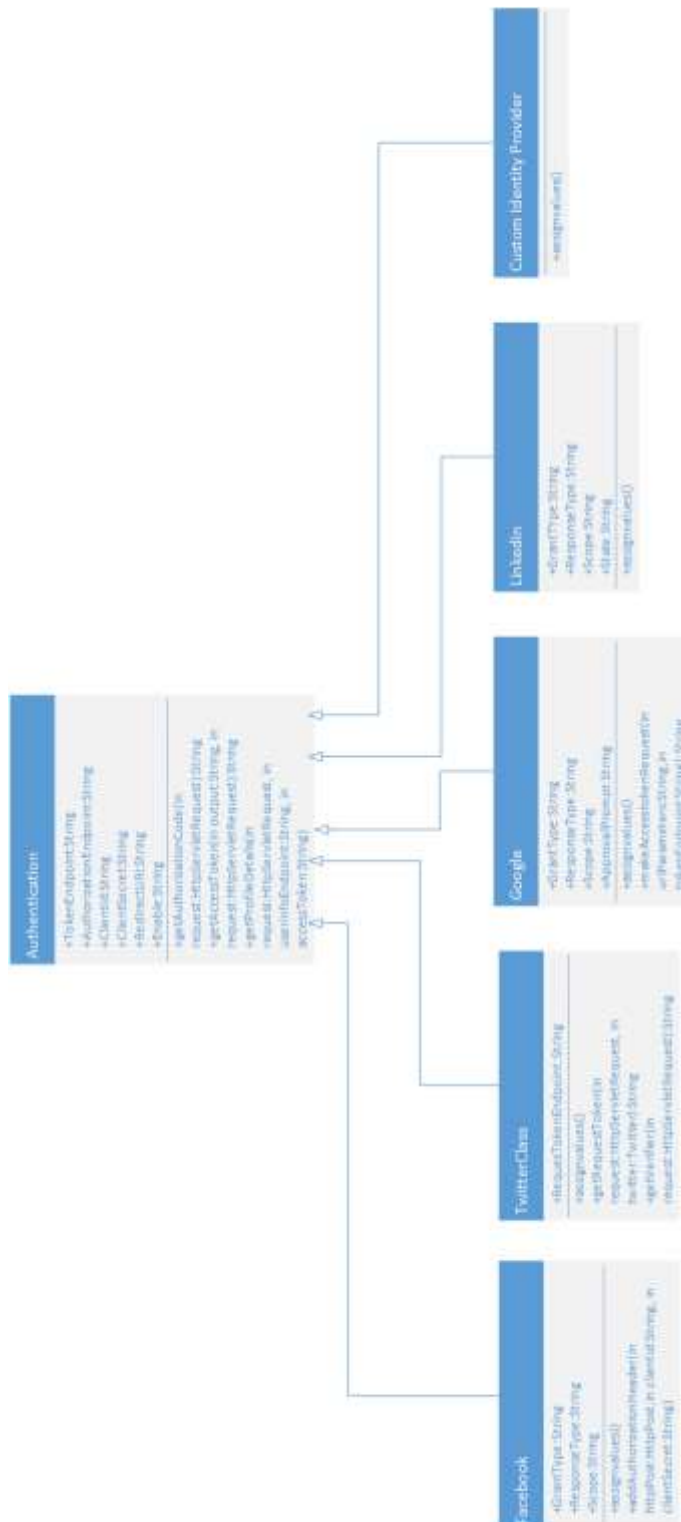


APPENDIX C: Use case scenarios for Authentication Component

Use case 1	User registration												
Goal	To register user in the classified web site												
Pre-condition	None												
Actor	User												
Success Scenario	<table border="1"> <thead> <tr> <th>Step</th><th>Action</th></tr> </thead> <tbody> <tr> <td>1.0</td><td>Go to registration page</td></tr> <tr> <td>2.0</td><td>Provide user details</td></tr> <tr> <td>3.0</td><td>Validate user details</td></tr> <tr> <td>4.0</td><td>Store details in database</td></tr> <tr> <td>5.0</td><td>Display successful message</td></tr> </tbody> </table>	Step	Action	1.0	Go to registration page	2.0	Provide user details	3.0	Validate user details	4.0	Store details in database	5.0	Display successful message
Step	Action												
1.0	Go to registration page												
2.0	Provide user details												
3.0	Validate user details												
4.0	Store details in database												
5.0	Display successful message												
Alternative flow	<table border="1"> <thead> <tr> <th>Step</th><th>Action</th></tr> </thead> <tbody> <tr> <td>3.0</td><td>If user details are invalid, system gives an error message</td></tr> </tbody> </table>	Step	Action	3.0	If user details are invalid, system gives an error message								
Step	Action												
3.0	If user details are invalid, system gives an error message												

Use case 2	Login with local authentication										
Goal	To register user in the classified web site										
Pre-condition	User should be pre-registered										
Actor	User										
Success Scenario	<table border="1"> <thead> <tr> <th>Step</th><th>Action</th></tr> </thead> <tbody> <tr> <td>1.0</td><td>Go to Login page</td></tr> <tr> <td>2.0</td><td>Provide user details</td></tr> <tr> <td>3.0</td><td>Validate user details</td></tr> <tr> <td>4.0</td><td>Display home page</td></tr> </tbody> </table>	Step	Action	1.0	Go to Login page	2.0	Provide user details	3.0	Validate user details	4.0	Display home page
Step	Action										
1.0	Go to Login page										
2.0	Provide user details										
3.0	Validate user details										
4.0	Display home page										
Alternative flow	<table border="1"> <thead> <tr> <th>Step</th><th>Action</th></tr> </thead> <tbody> <tr> <td>3.a.1</td><td>If user credentials are invalid, system gives an error message</td></tr> <tr> <td>3.a.2</td><td>If user credentials are invalid and risk score is high, email is sent to user telling suspicious login activity</td></tr> <tr> <td>3.a.3</td><td>If user credentials are valid, but risk score is higher, verification page will be loaded.</td></tr> </tbody> </table>	Step	Action	3.a.1	If user credentials are invalid, system gives an error message	3.a.2	If user credentials are invalid and risk score is high, email is sent to user telling suspicious login activity	3.a.3	If user credentials are valid, but risk score is higher, verification page will be loaded.		
Step	Action										
3.a.1	If user credentials are invalid, system gives an error message										
3.a.2	If user credentials are invalid and risk score is high, email is sent to user telling suspicious login activity										
3.a.3	If user credentials are valid, but risk score is higher, verification page will be loaded.										

APPENDIX D: Class diagram for Authentication component



APPENDIX E: Test Cases for Authentication Component

Test Case #	TC_01
Description	Login with Facebook
Pre-conditions	There should be an active internet connection and an active Facebook account.
Test Steps	<ol style="list-style-type: none">1. Click on login with Facebook button on Login page.2. Enter the email address and password.3. Give permission to access your profile
Input	Email: ishanivv@gmail.com Password: ishani
Expected Output	Successfully logged in
Actual Output	Successfully logged in
Status	Pass

Test Case #	TC_02
Description	Login with Twitter
Pre-conditions	There should be an active internet connection and an active Twitter account.
Test Steps	<ol style="list-style-type: none">1. Click on login with Twitter button on Login page.2. Enter the email address and password.3. Give permission to access your profile
Input	Email: ishanivv@gmail.com Password: ishani
Expected Output	Successfully logged in
Actual Output	Successfully logged in
Status	Pass

Test Case #	TC_03
Description	Login with LinkedIn
Pre-conditions	There should be an active internet connection and an active LinkedIn account.

Test Steps	<ol style="list-style-type: none"> 1. Click on login with LinkedIn button on Login page. 2. Enter the email address and password. 3. Give permission to access your profile
Input	Email: ishanivv@gmail.com Password: ishani
Expected Output	Successfully logged in
Actual Output	Successfully logged in
Status	Pass

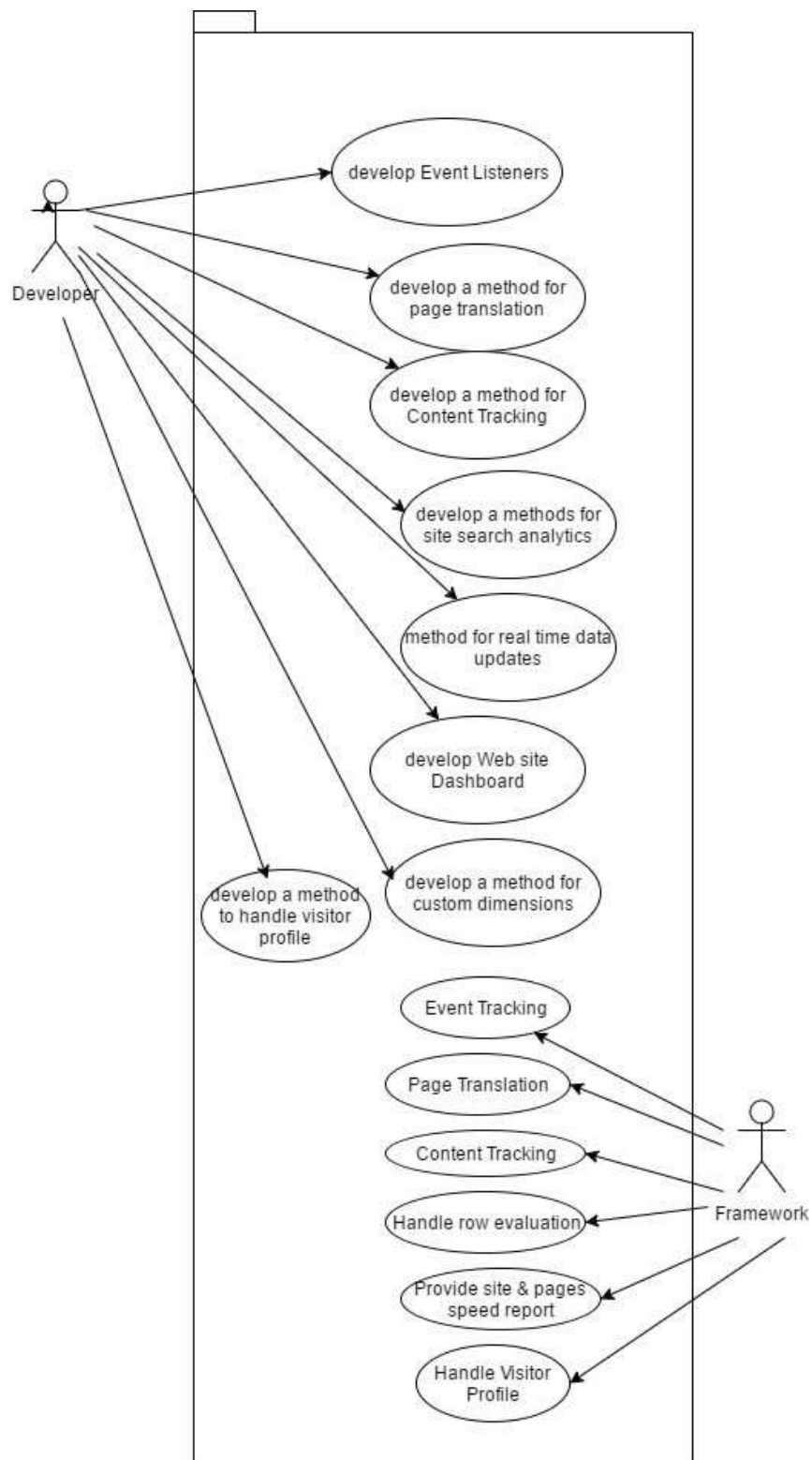
Test Case #	TC_04
Description	Login with Google
Pre-conditions	There should be an active internet connection and an active Google account.
Test Steps	<ol style="list-style-type: none"> 1. Click on login with Twitter button on Login page. 2. Enter the email address and password. 3. Give permission to access your profile
Input	Email: ishanivv@gmail.com Password: ishani
Expected Output	Successfully logged in
Actual Output	Successfully logged in
Status	Pass

Test Case #	TC_05
Description	Login with local authentication using previous login IP, device, browser, location and date/time
Pre-conditions	There should be an active internet connection and user should be registered with the website.
Test Steps	<ol style="list-style-type: none"> 1. Enter the email address and password. 2. Click login button
Input	Email: ishanivv@gmail.com Password: ishani
Expected Output	Successfully logged in
Actual Output	Successfully logged in
Status	Pass

Test Case #	TC_06
Description	Login with local authentication using different IP address
Pre-conditions	There should be an active internet connection and user should be registered with the website.
Test Steps	<ol style="list-style-type: none"> 1. Enter the email address and password. 2. Click login button
Input	Email: ishanivv@gmail.com Password: ishani
Expected Output	Email is sent with the OTP. Redirect to the verify page to enter the OTP
Actual Output	Email is sent with the OTP. Redirect to the verify page to enter the OTP
Status	Pass

Test Case #	TC_07
Description	Login with local authentication using different IP address and invalid password
Pre-conditions	There should be an active internet connection and user should be registered with the website.
Test Steps	<ol style="list-style-type: none"> 1. Enter the email address and password. 2. Click login button
Input	Email: ishanivv@gmail.com Password: ishaniii
Expected Output	Email is sent with the logged in information. Invalid credentials message is displaying
Actual Output	Email is sent with the logged in information. Invalid credentials message is displaying
Status	Pass

APPENDIX F: Use case diagram of Web analytics component



APPENDIX G: Use case scenarios for Web analytics Component

Use case No	01
Use Case	Event Tracking
Actors	Framework
Pre-Conditions	Should be an event
Flow of Event	Processing the Event
Post Conditions	Finishes the event

Use case No	02
Use Case	Page Translation
Actors	Framework
Pre-Conditions	What did before come to the page
Flow of Event	Went through page by page
Post Conditions	What did after use the page

Use case No	03
Use Case	Develop a methods to Handle Visitors
Actors	Developer
Pre-Conditions	Develop event listener
Flow of Event	Develop a Visitor class Create methods to handle the visitor
Post Conditions	Calling the created method in necessary events

Use case No	04
Use Case	Provide site & pages speed report
Actors	Framework
Pre-Conditions	User done some events in the website
Flow of Event	User click on view ad button The sever searching the result for the request Display the requested details
Post Conditions	Output the report

APPENDIX H: Class Diagram of Web Analytics Module

