OPEN SOURCE SOA BASED MIDDLEWARE FRAMEWORK FOR CLASSIFIED BASED WEB DEVELOPMENT

Project ID: 17-072

Software Requirements Specification (SRS)

Bachelor of Science Special (Hons) Degree in Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

02nd May 2017

OPEN SOURCE SOA BASED MIDDLEWARE FRAMEWORK FOR CLASSIFIED BASED WEB DEVELOPMENT

Project ID : 17-072

Software Requirements Specification (SRS)

Author: K.S.D.A Kulathunga

Student ID: IT 14 256 86

Supervisor: Mr.Nuwan Kodagoda

Bachelor of Science Special (Hons) Degree in Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

02nd May 2017

DECLARATION

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

K.S.D.A Kulathunga Signature:

ABBREVIATIONS

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to fully describe the functional and nonfunctional requirements, design constraints, project approach and other factors necessary to provide an in-depth view of the project progress for the Open source SOA Middleware framework, Knowledge management and decision supporting systems. System requirements, functional requirements and nonfunctional requirements are stated explicitly and precisely for the software engineers who will be involved in the process of implementation and maintenance. This document is targeting the client, designers, developers and other stakeholders as its audience. A description of the problem in focus will be provided relating what the system would do in order to overcome the problem. This document will serve the purpose of providing the potential users to determine if the software specified meets their needs or how the software must be modified to meet their needs.

1.2 Scope

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

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

1.3 Definitions, Acronyms, and Abbreviations

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

1.4 Overview

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

1. Reduce developer burden, complexity and knowledge gap during the development process.

2. Reduce development time and cost of a classified based web development project.

3. Get sophisticated features as well as wide range of flexibility.

2. Overall Descriptions

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

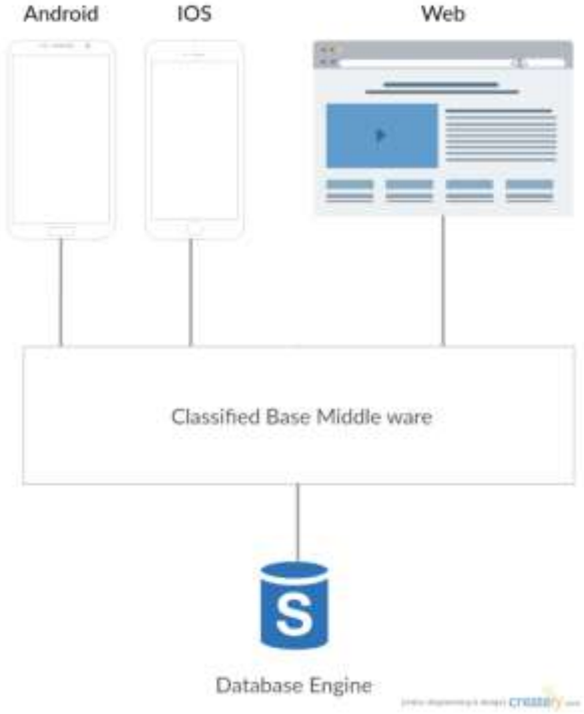

Figure 1: High level view of the complete system

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

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

This is an open source framework. Each module in this framework can be extensible or even customizable according to the user's requirements. The figure below visualize the architecture of this middleware framework

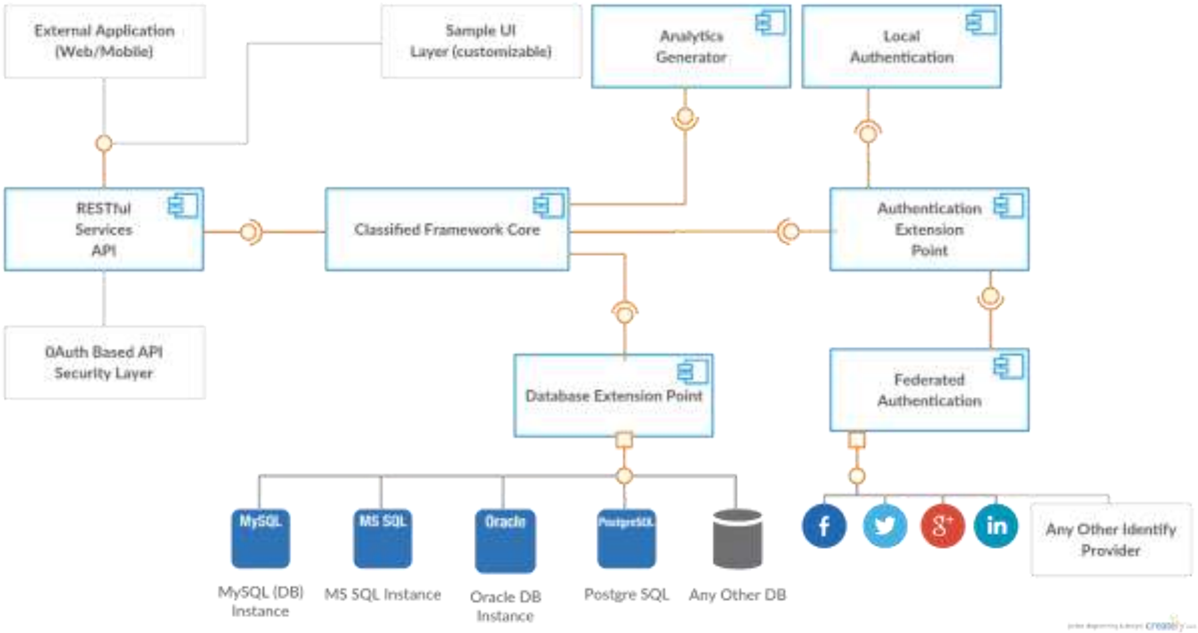

Figure 2: High level architecture diagram of the middleware

2.1 Product perspective

There are several existing solution to development of classified based web development. Majority of these are vendor specific platforms or frameworks where the user can create an classified based website using that framework and deploy on the vendor's platform. The main drawback is lack of customizability, and thus lack of extensible or require complex configurations. Listed below are some several popular available solutions.**YCLAS** [1]

Yclas is a tool which enables the users that are little or no knowledge in web development, to create classified web sites. Deployment should be done on a platform that is provided by Yclas.

FLYNAX [2]

It's a PHP script developed based on MVC architecture.

OXY CLASSIFIDES [3]

PHP script base product that helps to build classifieds, this product can be customizable for some extend.

TITAN CLASSIFIDES [4]

Can create and manage professional level clean classifieds. Customization can be done through vendors' developers. Also can't be used for free.

OS CLASS [5]

It is a PHP script that allows you to quickly create and manage your own free classifieds site.

CLASSIPRESS

It is a word press plugin which helps to create and manage WordPress base classified websites

(See appendix 1 for existing products and purposed product comparison table)

2.1.1 System Interfaces

This component doesn't have any system interfaces as it does not belong to any existing system

2.1.2 User Interfaces

This component doesn't have any direct user interfaces.

2.1.3 Hardware Interfaces

The component doesn't have any direct hardware interfaces. All hardware connections are managed by the operating system of the hosted web server.

2.1.4 Software Interfaces

Name	**Version**	**Purpose**
JDK	1.7 or higher	Execute sub-modules
Glassfish server	4.0	Deploy application
MySQL	5.7 or higher	Database operations
MSSQL	2012 or higher	
Oracle	12c	
PostgreSQL		
GIT	Latest stable version	Version controlling
Source tree	Latest stable version	GIT GUI client
Apache jmeter	3.2	Performance testing
Apache maven	3.3	Dependency Management
Jenkins	2.46	Continuous integration
Google Chrome	Latest stable version	Developments
Postman	Latest stable version	Test APIs
Junit	4.12	Unit testing
Intelij IDEA	16.0 or higher	Integrated development environment

FindSecBugs	Latest stable version	Security Testing
OWASP		
Dependency Check		
OWASP ZAP		
Ehcache	3.0	Caching layer implementations

2.1.5 Communication Interfaces

The core component exposes an extensible restful api services based on http protocol. The communication between the other component/modules are handled internally by the Java virtual Machine

2.1.6 Memory Constrains

The Minimum Configuration required is a Web server with 4GB RAM and 20GB hard drive.

2.1.7 Operations

The developer is required to make the below configuration for smooth operation of the system:

2.1.8 Site adaptation requirements

Adaption of the Core Component requires to meet the below requirements.

- JDK 1.7 or higher
- JDBC
- Glassfish server 4
- IDE (eclipse, IDEA).

2.2 Product functions

The core component is focused on Middleware framework core including the restful Service API used by the web developer for classified web development 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.

Developer:

Integration and Configuration

- Integration and Configuration DBMS Component to the core of the framework
- Integration and Configuration of Analytical Component to the core of the framework
- Integration and Configuration of Federated Authentication to the core of the framework
- Extensibility to add plugins to the Core framework and Configuration
- Security Configurations of the core framework
- Extensibility to other endpoints to External Application
- Security Configuration of the core routing

Restful API Framework (extensible)

- Publish Classified advertisement
- Edit Classified advertisement
- Search Classified advertisement
- Sort Classified advertisements

DBMS Integration and Configurations to Core

Use Case Number	01
Use Case	DBMS Integration and configurations to Core
Actor	Developer
Pre-Condition	User should open the relevant configuration file using a text editor User should move the relevant package to the system
Flow of Event	User sets the configurations
Post-Condition	Database component successfully connected to core framework
Alternatives	Prompt error to user

Table 1: DBMS Integration and Configurations to Core

Analytics Component Integration and Configurations to Core

Use Case Number	02
Use Case	Analytics Component Integration and configurations to Core
Actor	Developer
Pre-Condition	User should open the relevant configuration file using a text editor User should move the relevant package to the system
Flow of Event	User sets the configurations
Post-Condition	Analytics component successfully connected to core framework
Alternatives	Prompt error to user

Table 2: Analytics Component Integration and Configurations to Core

Federated Authentication Component Integration and Configurations to Core

Use Case Number	03
Use Case	Federated Authentication Component Integration and configurations to Core
Actor	Developer
Pre-Condition	User should open the relevant configuration file using a text editor User should move the relevant package to the system
Flow of Event	User sets the configurations
Post-Condition	Federated Authentication component successfully connected to core framework
Alternatives	Prompt error to user

Table 3: Federated Authentication Component Integration and Configurations to Core

Extensibility to add plugins to the Core framework and Configuration

Use Case Number	04
Use Case	Extensibility to add plugins to the Core framework and Configuration
Actor	Developer
Pre-Condition	User should open the relevant configuration file using a text editor User should add the plugins to the system
Flow of Event	User adds the plugins to the system User sets the configurations

Post-Condition	Plugin successfully connected to core framework

Table 4: Extensibility to add plugins to the Core framework and Configuration

Security Configurations of the core framework

Use Case Number	05
Use Case	Security Configurations of the core framework
Actor	Developer
Pre-Condition	User should open the relevant configuration file using a text editor
Flow of Event	User sets the configurations
Post-Condition	Secure core framework
Alternatives	Prompt error to user

Table 5: Security Configurations of the core framework

2.3 User characteristics

This core component is directly used by developers in creating the classified website. As the framework is based on Java Technology they are required to have a certain level of knowledge of java and its workings. The API are based on restful service so a in depth knowledge of the SOA concepts would be an advantage. User's knowledge in the below development tools would be an added advantage:

- Maven
- Git
- Jerkins
- EhCahe

2.4 Constrains

- Server should be able to handle multiple requests
- Server resource should be increased with respective to the user amount increase
- Developers should follow guidelines in the documentation when customizing the module otherwise it may cause performance, security, integration issues
- In development environment should be set up with all required tool installed in the system
- In production environment, relevant tools and applications should be installed on the system
- Both user types need be confident in the English language

2.5 Assumptions and dependencies

Assumptions

- Future version of Java or any other dependency will allow running applications developed using existing Java version and its dependencies.
- No data lost during the communication between server and client

Dependencies

- Developer must have the appropriate tools installed and configured for development using the framework application.
- The technology used is Java, as such all required dependencies of java for operation must be present.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The module does not have any direct user interfaces.

3.1.2 Hardware Interfaces

- For Production Server should have minimum configuration of
    - 10GB RAM,
    - 500GB disk space hard drive
    - Installed glassfish server
    - Installed JDK 1.7 or higher required for executing all modules.
- Developer must have pc with the at least the minimum configuration of
    - 4GB RAM
    - 20GB disk space hard drive
    - 1.6 GHz Processor
- For development and testing requires a browser as well as an internet connection.

.

3.1.3 Software Interfaces

Name	Version	Purpose
JDK	1.7 or higher	Execute sub-modules
Glassfish server	4.0	Deploy application
MySQL	5.7 or higher	Database operations
MSSQL	2012 or higher	
Oracle	12c	
PostgreSQL		
GIT	Latest stable version	Version controlling
Source tree	Latest stable version	GIT GUI client
Apache jmeter	3.2	Performance testing
Apache maven	3.3	Dependency Management
Jenkins	2.46	Continuous integration
Google Chrome	Latest stable version	Developments
Postman	Latest stable version	Test APIs
Junit	4.12	Unit testing
Intelij IDEA	16.0 or higher	Integrated development environment

FindSecBugs	Latest stable version	Security Testing
OWASP		
Dependency Check		
OWASP ZAP		
Ehcache	3.0	Caching layer implementations

Table 6 : list of software interfaces

3.1.4 Communication Interfaces

The core component exposes an extensible restful api services based on http protocol. The communication between the other component/modules are handled internally by the Java virtual Machine.The Client application is connected via a RESTful interface to connect with an API server that provides services for those applications..

3.2 Performance Requirements

Classified website engage with thousands of users daily and their respective requests. Thus a high performance is crucial for its successful operation.

- Optimum the response for the request must be given in less than a second
- The Server should be able to handle multiple request simultaneously.
- There should a fast Internet connection
- The server must have a fast processor and minimum of 4GB RAM.

3.3 Design constraints

In designing this framework core, we have to consider about the designing of the extensible restful API and the ability to plugin other components to the core of the framework. This will include designing of the framework architecture as well as routings of the system.

3.4 Software system attributes

3.4.1 Reliability

Reliability is one of the most important attributes of measuring software quality. It is the probability of failure-free software operation for a given time period. [6]

- The component needs to continuously work without crashes or errors.
- Data should not get corrupted
- For any failure there should be notice given prior.

3.4.2 Availability

Availability of a system is the possibility that a system will work as required during a given point of time, and it should be able to deliver the requested service. In other words, services should be available with minimal system downtime, or without having any system failures for a long-time period.

- Framework should be available to be used with any client application.
- Other modules of the middleware should be able to use this module at any given time period

3.4.3 Security

Security is another major attribute in this component. Setting up security condition, the system should be provided with various access levels as privileges.

- Maintains strong server-side controls
- All external exposed API used HTTPS protocol
- User separate API security layer

3.4.4 Maintainability

Maintainability of a system is the ability of the system to handle with new requirements in order to enhance the performance and capabilities of the application and making sure that new errors shall not be prone in the system because of the changes. That means the proposed system can be maintained easily if there is some modification without happening any damage or interrupt to other system functionalities. The frameworks maintenance is performed by the development team.

3.5 Other requirements

<u>Proper documentations and help and support</u>

The Middleware framework will need a documentation and guidelines for developer to reference

References

[1] "Yclas Home," Yclas, [Online]. Available: https://yclas.com/.

[2] "flynax home," flynax, [Online]. Available: https://www.flynax.com/.

[3] "OXY CLASSIFIDES Home," OXY CLASSIFIDES, [Online]. Available: https://www.oxyclassifieds.com/.

[4] "TITAN CLASSIFIDES Home," TITAN CLASSIFIDES, [Online]. Available: http://www.titanclassifieds.com/.

[5] "OS CLASS Home," OS CLASS, [Online]. Available: https://osclass.org/.

[6] J. Pan, "Software Reliability," [Online]. Available: https://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/.

[7] A. Biswal, "Java caching frameworks for enterprise applications," Linkedin, 13 March 2015. [Online]. Available: https://www.linkedin.com/pulse/java-caching-frameworks-enterprise-applications-anshuman-biswal.

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

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

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

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

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

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

[14] 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.

[15] Q. Luo, S. Krishnamurthy, C. Mohan, H. Pirahesh, H. Woo, B. G. Lindsay and J. F. Naughton, *Middle-Tier Database Caching for e-Business,* p. 12.

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

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

APPENDICIES

Appendix 1: Table of comparison between existing products and proposed product

Features	**Yclas**	**Flynax**	**Oxy classifides**	**Titan classifides**	**Os class**	**Classipress**	**Proposed Framework (amplier)**
Open Source	*YES*	*NO*	*NO*	*NO*	*YES*	*NO*	*YES*
Free	*YES*	*NO*	*NO*	*NO*	*YES*	*YES*	*YES*
Extensible Database Integration	*NO*	*NO*	*NO*	*NO*	*NO*	*NO*	*YES*
Social Login	*YES*	*YES* *(only facebook)*	*YES*	*NO*	*YES* *(only facebook*	*YES*	*YES*
Extensible Federated Authentication	*NO*	*NO*	*NO*	*NO*	*NO*	*NO*	*YES*
Web Analytics	*NO*	*NO*	*NO*	*NO*	*NO*	*NO*	*YES*

Table 7: existing product comparison table