OPEN SOURCE SOA BASED MIDDLEWARE FRAMEWORK FOR CLASSIFIED BASED WEB DEVELOPMENT

Project Id :17-072

Software Requirements Specification

Bachelor of Science Special (Hons) Degree in Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

2nd May 2017

OPEN SOURCE SOA BASED MIDDLEWARE FRAMEWORK FOR CLASSIFIED BASED WEB DEVELOPMENT

Project Id :17-072

Software Requirements Specification

I.H. Liyanaarachchi – IT13137342

Supervisor : Mr. Nuwan Kodagoda

…………………………..

Bachelor of Science Special (Hons) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

2nd May 2017

DECLARATION

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

	STUDENT NAME	STUDENT NO.	SIGNATURE
1	I.H. Liyanaarachchi	IT 13 1373 42	

Table of Contents

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose of this document is described and a list of abbreviations and definitions are provided.

1.1 Purpose

The purpose of the Software Requirements Specification is to outline the user requirements, system requirements, functional requirements and non-functional requirements of the **Authentication extension** component. System requirements, functional requirements and non-functional requirements are stated explicitly and precisely for the software engineers who will be involved in the process of implementation and maintenance. Since this document provides the design documents of the system quality assurance engineers, maintenance engineers and project managers will be able to provide precise decisions. This document will describe the project perspectives and its functions. It would also give a detailed description on the operating environments as well as the assumptions that we have made. This document also will carry the interfaces assembled. The remaining sections of this SRS describe the functional requirements for the Authentication extension component.

1.2 Scope

The “Authentication extension” component has two major sub components local authentication and federated authentication. Local authentication sub component is used to authenticate end users locally (database, LDAP, Active directory etc.). Federated authentication sub component is there for the end users who do not wish to create a user account in the system. End users can use an already existing identity provider and get authenticated to the system.

Furthermore, federated authentication sub component will support major identity providers, such as facebook, twitter, linkedin, google. Apart from the major identity providers, there will be an extension point in the component to implement a connector to support any other identity provider of developer’s or customer’s choice.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
LDAP	Lightweight Directory Access Protocol
SRS	Software Requirement Specification
API	Application Programming Interface
TCP/IP	Transmission Control Protocol/ Internet Protocol
HTTP	Hypertext Transfer Protocol

1.4 Overview

The main goal of this “Authentication Extension” component is to provide a mechanism for the developer to extend the authentication extension module so that he/she can use different identity providers of his/her choice to attach to the authentication purpose.

Main tasks of this component,

1. Is to provide local authentication mechanism to get authenticated using database, LDAP, Active directory etc.
2. Is to provide federated authentication management for the major identity providers such as facebook, twitter, google, linkedin.
3. Is to provide a mechanism to extend the federated authentication sub component to connect any identity provider of developer’s choice.

Mainly this component will be used by developers for the developing purposes.

The remainder of this SRS document includes three sections and appendixes. The second section provides an overall view of the component functionality and interaction with other components. This section also discusses the specific requirements such as functional and nonfunctional requirements, design constraints and various approaches. Furthermore, this section also mentions the system constraints, User characteristics and assumptions about the product.

The third section provides the requirements specification in detail and a description of the different interfaces. Different specification techniques are used in order to specify the requirements more clearly for different audiences.

The rest of the sections that organized this document are Project perspective and descriptions, Different interfaces that the system consists, Requirements of the system, Summary of major functionality, Users and characteristics of the system and the background of the general factors affect the system.

2. Overall Description

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

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

There are several roles defined by OAuth.

- Resource owner
- Client
- Resource server
- Authorization server

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

Following diagram shows the architecture of Authentication extension component.

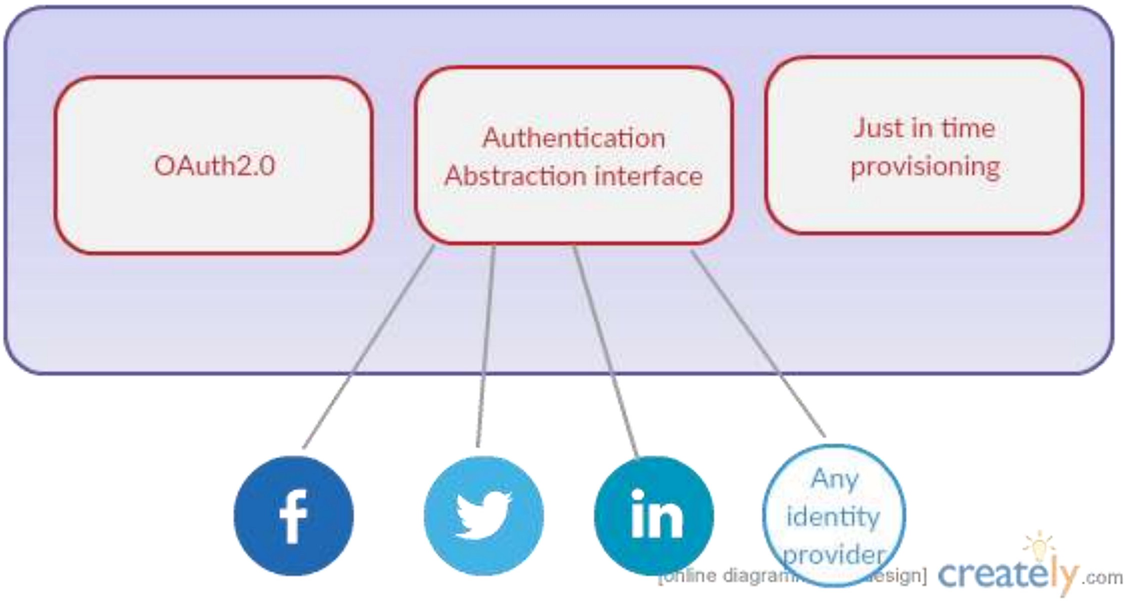

OAuth2.0
Authentication Abstraction interface
Just in time provisioning
Any identity provider
[online diagram design] creately.com

Figure 1: High level diagram of Authentication extension component

2.1 Product Perspective

Many web development frameworks use the concept of federated authentication. Out of those, we have focused on to the frameworks which are specially designed for classified web development.

Following table shows a feature comparison between those frameworks and tools.[1,2,3,4,5,6]

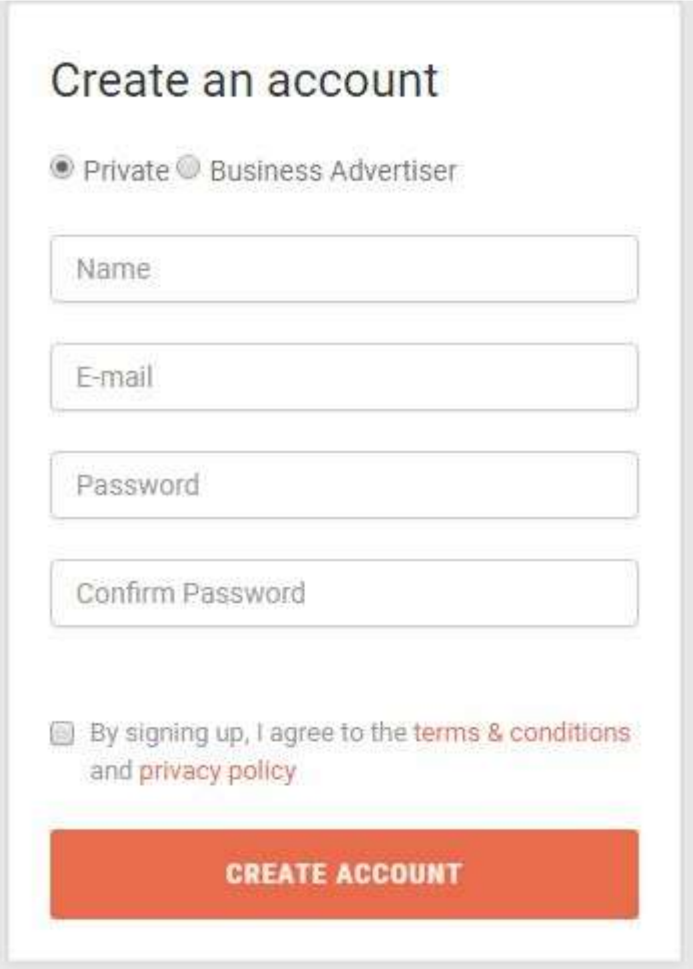
Features	Yclas (Open classifieds)	Flynax	Oxy classifieds	Titan classifieds	Os class	ClassiPress	Ampliar (Proposed framework)
Social login	Yes	Yes (Only facebook)	Yes	No	Yes (Only facebook)	Yes	Yes
Extensible federated authentication	No	No	No	No	No	No	No

2.1.1 System Interfaces

No specific system interfaces are there since this component is not going to interact with existing systems.

2.1.2 User Interfaces

The user interface (UI) comprises the logical face between software product and its users. This component deals with user interfaces for both local authentication and federated authentication. For the local authentication web application will have sign up user interface and login interface where federated authentication also embedded in login user interface.



Create an account

● Private ○ Business Advertiser

Name

E-mail

Password

Confirm Password

□ By signing up, I agree to the terms & conditions and privacy policy

CREATE ACCOUNT

Figure 2: Sign up user interface

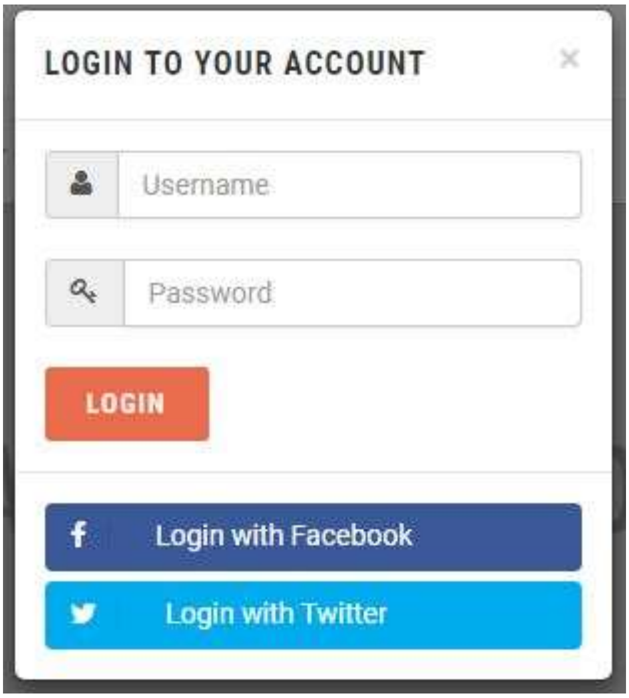


Figure 3: Login user interface

2.1.3 Hardware Interfaces

There’s no specific hardware interfaces used for this component. But in order to carry out the web development, developers need laptop or desktop computer.

2.1.4 Software Interfaces

For the local authentication process, to store user’s details and retrieve major database vendors such as MSSQL, MYSQL, Oracle etc will be used. For the federated authentication process, APIs (facebook, google, twitter, linkedin) will be used to integrate the connectors.

The framework should expose APIs for all functionality as RESTful web services. The services should be secured with OAuth protocol [7].

2.1.5 Communication Interfaces

- Client/ server based communication.
- TCP/IP protocol.
- To connect to the internet modem, router, dongle or any connection made up device is needed.
- Most of the communication will be handled over HTTP via an internet connection and the API services.

2.1.6 Memory Constraints

Database server should have at least minimum of 300MB to perform local authentication as well federated authentication. And also laptop or pc should have free space of 300MB and 512MB of RAM to carry out the development process.

2.1.7 Operations

The web application component of Authentication Extension will have following operations.

- Create user accounts locally.
- Login using local authentication.
- Login using federated authentication.
- Use just in time provisioning to create accounts on the fly.
- Add preference identity providers for the federated authentication.

2.1.8 Site adaptation requirements

The system will display the information only in English language. In later versions, it will be implemented to use Sinhala and Tamil languages.

2.2 Product Functions

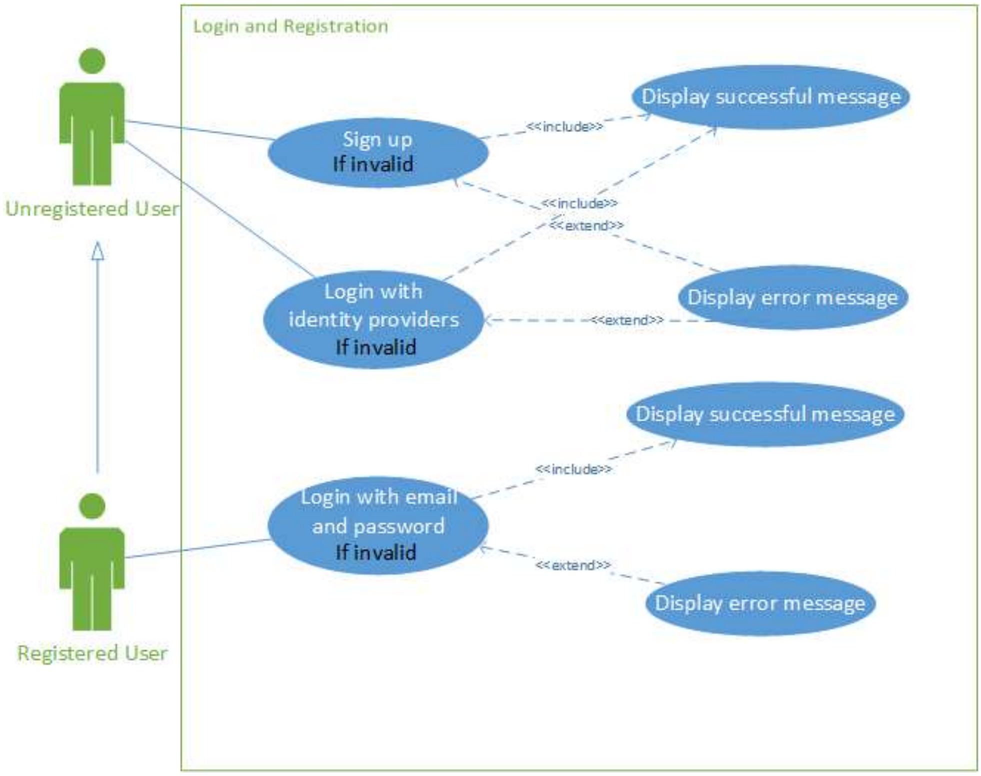


Figure 4: Use case diagram

Use Case ID	UCS_001
Use Case Name	Sign up
Pre-condition	User should have a proper internet connection
Primary Actor	Unregistered User
Main Success Scenario Steps	1.0 User selects sign up operation 2.0 The system prompt for unregistered user to enter the name, email address and password. 3.0 User enters name, email address and password. 4.0 System prompts to confirm password. 5.0 User confirms password. 6.0 System validates details. Include::Display successful message
Extensions	3.a.1 If entered email address is not a correct one, system prompts user to give a valid email address. 5.a.1 If the confirm password does not match with the original password, system prompts user to enter correct password.

Use Case ID	UCS_002
Use Case Name	Login with email and password
Pre-condition	User should have a valid user account
Primary Actor	Registered User
Main Success Scenario Steps	1.0 User selects login operation

	2.0 System prompts user to enter the email address and password. 3.0 User enters email address and password. 4.0 System validates email address and password.
Extensions	3.a.1 If entered email address is not a correct one, system prompts user to give a valid email address. 3.a.2 If the entered password does not match with the account password, system prompts user to re-enter password.

Use Case ID	UCS_003
Use Case Name	Login with identity providers.
Pre-condition	User should have a valid user account in the selected identity provider. (facebook, twitter, google etc)
Primary Actor	Unegistered User
Main Success Scenario Steps	1.0 User selects login with facebook operation 2.0 Facebook application opens up and ask for the email and password. 3.0 User enters email address and password. 4.0 Facebook application validates email address and password.
Extensions	3.a.1 If entered email address is not a correct one, application prompts user to give a valid email address. 3.a.2 If the entered password does not match with the account password, application prompts user to re-enter password.

2.3 User characteristics

Developers

Mainly the users who interacts with the classified based web development framework are developers. Developers should have basic knowledge in java web and OOP concepts. And also developers should have some understanding in using federated authentication processes. Novice to professional developers will be able to deal with the authentication extension component since the development effort and knowledge needed for the implementation is very less.

2.4 Constraints

- Developers should have a proper internet connection to test and run federated authentication process.
- Server should be able to handle multiple requests at a time.
- The response should be generated within minimum time constraint.

2.5 Assumptions and dependencies

Assumptions

- We assumed that the users are much likely to use social logins to login to different websites as in it has proved by many articles in the internet.
- We assumed that the different identity providers which developer choose will have proper security mechanisms.

Dependencies

- Federated authentication depends on the identity providers the website use. If the identity provider server goes down, there's no way of recovering data which used earlier.

2.6 Apportioning of requirements

The requirements described in sections 1 and 2 of this document are referred to as primary specifications; those in section 3 are referred to as requirements (or functional) specifications. The two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In the event that a requirement is stated within both primary and functional specifications, the application will be built from functional specification since it is more detailed.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Sign up user interface

Name of item	Sign up user interface
Description of purpose	Register users locally
Source of input or destination of output	
Valid range, accuracy and/or tolerance	100%
Units of measure	-
Timing	-
Relationships to other inputs/outputs	-
Screen formats/organization	Screen is organized in a web view
Window formats/organization	-
Data formats	Alphanumeric

- Login interface

Name of item	Login user interface
Description of purpose	Login to the application using locally registered credentials or login with social logins
Source of input or destination of output	-

Valid range, accuracy and/or tolerance	100%
Units of measure	-
Timing	Until user redirects to the home page
Relationships to other inputs/outputs	-
Screen formats/organization	Screen is organized in a web view
Window formats/organization	-
Data formats	Alphanumeric

3.1.2 Hardware Interfaces

PC/laptop to carry out the development for the developers. As mention before, there’s no specific hardware interfaces needed.

3.1.3 Software Interfaces

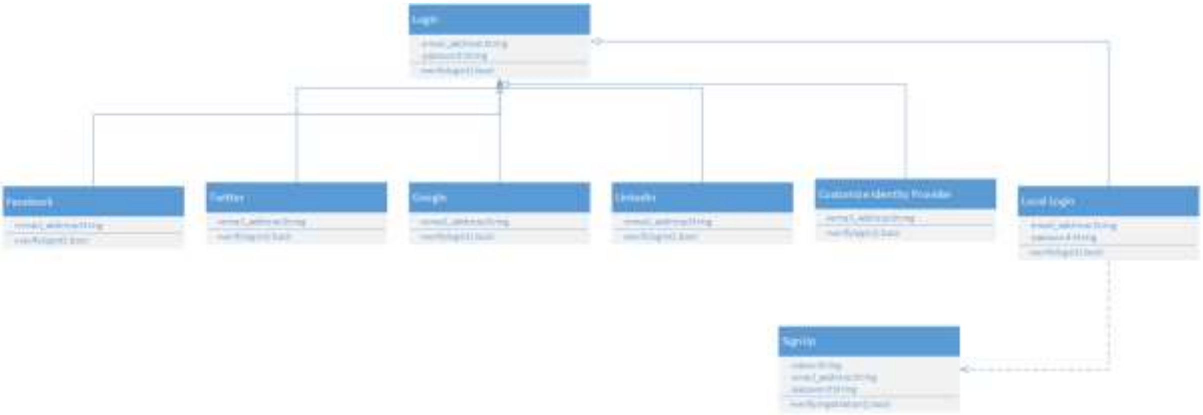
For the designing purposes team used visio 2013 and creately to design UML diagrams such as use case diagram, class diagram.

For the implementation, we’re going to use java development kit 1.8 and java runtime environment 1.8. And as the IDE we’re going to use IntelliJ IDEA. As mentioned before, for the federated authentication we will have to use different APIs to get them integrated with the component. MYSQL, MSSQL, Oracle and many different database vendors are going to deal with the component.

3.1.4 Communication Interfaces

Modem/Router or dongle will provide access to internet when it needs and HTTP/HTTPS protocols will be used.

3.2 Classes/Objects



3.3 Performance requirements

In order to minimize the delay between requests and responses, PC or laptop should have a minimum of core i3 processor. As the web application deals with huge amount of data and store those in the database for the future processing, it needs around 20GB capacity.

3.4 Design constraints

Since the component is a part of a classified based web development framework, the interfaces should not be complex. Because the users who deals with the classifieds web sites can be professional and novice users. So the interfaces should be simple and user friendly for the users to understand easily.

3.5 Software system attributes

3.5.1 Reliability

Reliability is one of most important attribute of measuring software quality. It is the probability of failure-free software operation for a time period. Software failures may be due to errors, ambiguities or misinterpretation of the specification that the software is supposed to satisfy, carelessness or incompetence in writing code, inadequate testing, incorrect or unexpected usage of the software or some other undesired problems. Other than that, there should be more reliability with the web servers to provide the best services. As well as when some system failure occurs there should be a way to maintain the reliability by providing the automated backup system.

3.5.2 Availability

Availability is the other most important attribute, which should have with the system. Mainly there must be ability to multiple users and the user levels to use the system whenever they need. Therefore, this type of decision support system should have high availability. Otherwise, it will lose the interest and importance of the decision support system.

3.5.3 Security

Security is the other major attribute, which should be there with any software solution. In here mainly to maintain good security condition, it should be provided various access levels as privileges. According to the authentication that each user will be obtained from the system, they can work with the system with their limitations. It will cause high security of the system.

- The development team must consider about the security of the user’s data. Because all the users provide their private data to the system. There for the database security must be on a higher place.
- Since for the federated authentication we use OAuth token based mechanism, all the details of the users will be secure with OAuth protocol.

3.5.4 Maintainability

Maintainability is another important attribute which cause to provide a best performed system. That means the proposed system can be maintained easily if there is some modification without happening any damage or interrupt to other system functionalities. As well as modifications can be done through the low cost solutions. It is also a somewhat important feature to having high maintainable system.

4. Supporting Information

4.1 Appendices

4.2 References

[1]"Classifieds Scripts Comparison - Open Classifieds", *Open-classifieds.com*, 2017. [Online]. Available: http://open-classifieds.com/comparison/. [Accessed: 01- Apr- 2017]

[2]"Classifieds Software - PHP Classified Ads Script", *Flynax Classifieds Software*, 2017. [Online]. Available: https://www.flynax.com/. [Accessed: 01- Apr- 2017]

[3]"OxyClassifieds - Powerful and Flexible Classified Ads Software", *Oxyclassifieds.com*, 2017. [Online]. Available: https://www.oxyclassifieds.com/. [Accessed: 01- Apr- 2017]

[4]"classipress — WordPress Plugins", *Wordpress.org*, 2017. [Online]. Available: https://wordpress.org/plugins/tags/classipress/. [Accessed: 01- Apr- 2017]

[5]"Titan Classifieds - Classifieds Software", *Titanclassifieds.com*, 2017. [Online]. Available: http://www.titanclassifieds.com/. [Accessed: 01- Apr- 2017]

[6]"Osclass, the free classifieds script", *Osclass.org*, 2017. [Online]. Available: https://osclass.org/. [Accessed: 01- Apr- 2017]

[7]"OAuth 2.0 — OAuth", *Oauth.net*, 2017. [Online]. Available: https://oauth.net/2/. [Accessed: 01- Apr- 2017]