

# **Sri Lanka Institute of Information Technology**

## **Secure Software Development**

### **OAuth Framework Assignment 2**

Submitted by:

**IT18059878 Abdurrahmaan A.N**

**IT18019278 Hareeni V**

**IT18037920 Thenuwara T.B.K.P**

# 1.Introduction

Internet security became one of the major topics in this era due to the massive usage of the internet around the world. Billions of online transactions happen every second. Providing security methods for each of those transactions became mandatory and extremely complicated.

When it comes to internet security, authentication and authorization are basic facts that should be mostly aware by the developers and the security engineers. In simple terms, authentication is the method of verifying the user by receiving their credentials such as username password combinations and authorization are the method of access granting for an authenticated user to his/her resources by verifying the user's access rights for the system [1].

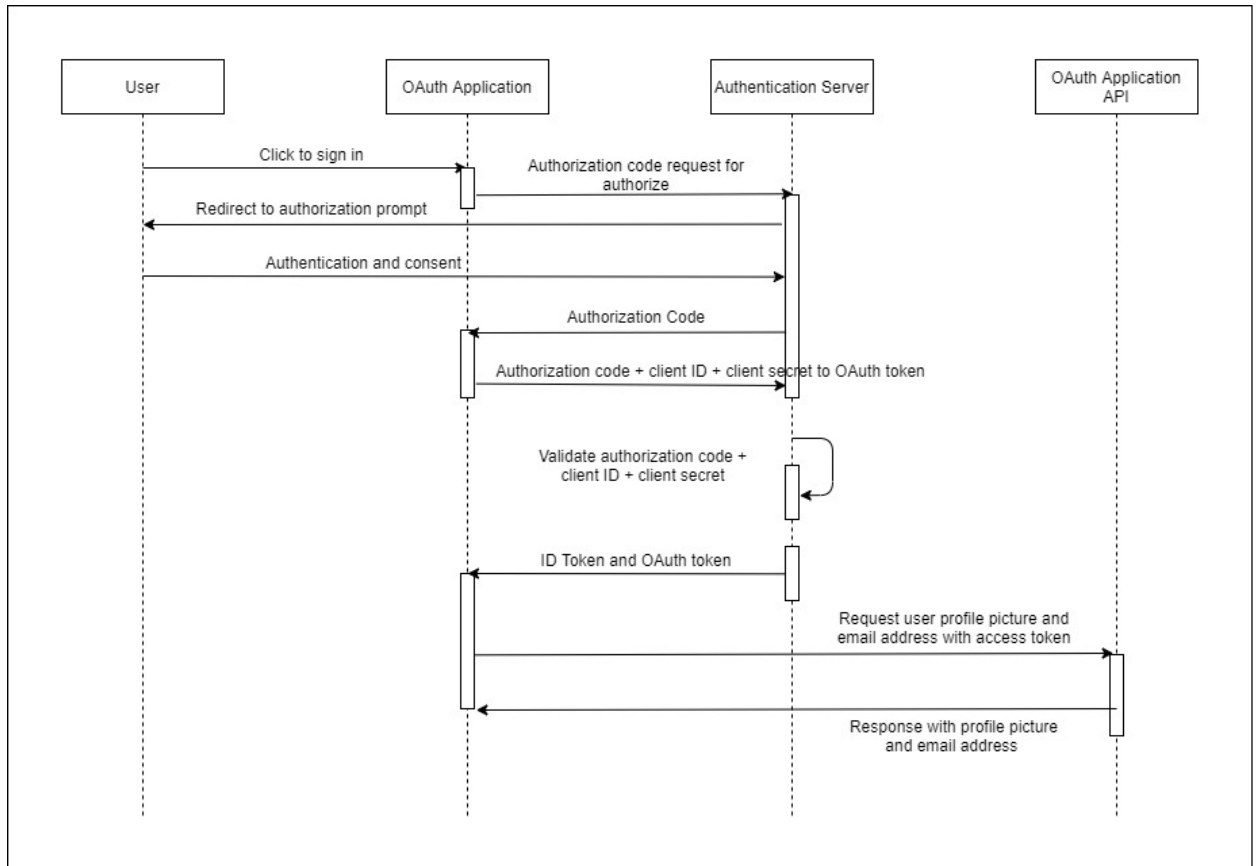
OAuth comes to play its role with regards to the fact about the authorization. It is an open standard for authorization which can be implemented by anyone. Apps use it to provide client applications with secure delegated access. It works over HTTPS and authorizes servers, applications, Application Programming Interfaces (APIs) and devices with access tokens rather than credentials. OAuth 2.0 is the most widely used version of OAuth between OAuth1.0a and OAuth 2.0 [2]. Utilizing OAuth, users are able to skip creating accounts and remembering passwords on each web application that they use on the internet.

Google, Facebook and Twitter are some of the reputed web clients who use OAuth 2.0 authorization framework. Basically OAuth 2.0 simplifies the previous version of the protocol and facilitates interoperability between different applications. Resource owner, resource server, client and authorization server are the four roles that are defined by OAuth 2.0. Furthermore, there are two types of tokens in OAuth 2.0 such as access token and refresh token. Tokens are the random string that is generated by the authorization server [1].

For this assignment, we have implemented a third-party web application that uses the user's Gmail as the access token for the authorization and the user's Facebook as the access token for the authorization. Then the application has the capability of using that access token to reach users data such as profile picture, hosted in the resource server.

## 2.Methodology

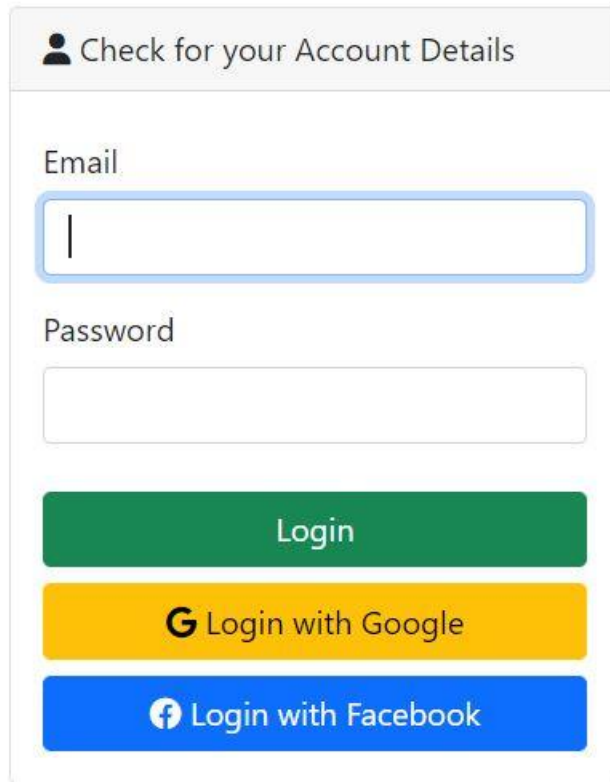
### Basic Message Flow



The above image shows the basic message flow overview of the implemented web application. The complete explanation on each step of the message flow is mentioned as below

### Step 1 – Click Sign in

The user reaches the first interface of the web application which is developed using OAuth 2.0 standard protocol and clicks the sign in button. In order to render the Sign In page, the system uses an endpoint



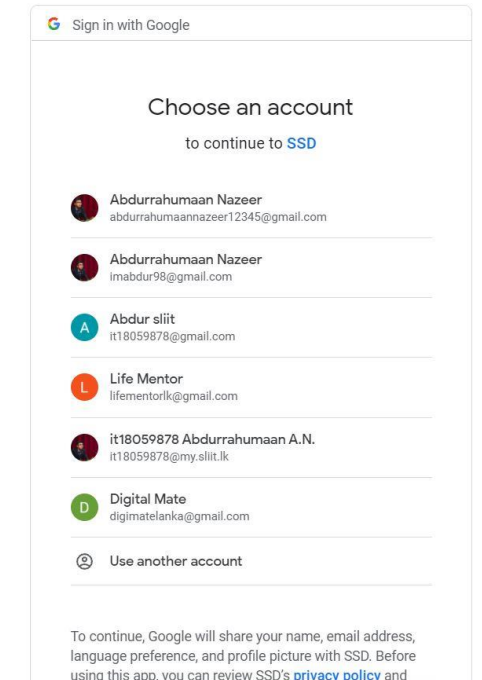
The image shows a login form with a light gray header bar containing a person icon and the text "Check for your Account Details". Below the header, there are two input fields: "Email" and "Password". The "Email" field is highlighted with a blue border and contains a vertical cursor. Below the "Password" field, there are three buttons: a green "Login" button, a yellow "G Login with Google" button, and a blue "f Login with Facebook" button.

### Step 2 – Send Authorization Code Request

OAuth application sends the authorization code request to Google Authorization Server

### Step 3 – Redirect to Authorization Prompt

In order to get the authorization code, Google Authorization Server redirects the user to the authorization prompt



#### Step 4 – Authorization and Consent

In this step the user selects the Gmail account that he/she is willing to sign in for the application. With that system becomes capable of authenticating and consenting this user.

#### Step 5 – Send Authorization Code

After the authentication of the user in the previous step, Google Authorization Server sends the authorization code to the OAuth Application.

#### Step 6 - Send Authorization Code, Client ID and Client Secret

In this step, OAuth Application sends the authorization code, client id and the client secret via the OAuth token to the Google Authorization Server

```
'google' => [
  'client_id' => '1068830446573-r698q2mkr9hh0jtnueod3k6n1rn0oqd8.apps.googleusercontent.com', //USE FROM GOOGLE DEVELOPER ACCOUNT
  'client_secret' => 'a3hL3xMPQTIbkiS01MSFmG', //USE FROM GOOGLE DEVELOPER ACCOUNT
  'redirect' => 'http://127.0.0.1:8000/google/callback/'
],

'facebook' => [
  'client_id' => '884791332409556', //USE FROM FACEBOOK DEVELOPER ACCOUNT
  'client_secret' => 'f6ef2fc402aeb722b527512c1494aee1', //USE FROM FACEBOOK DEVELOPER ACCOUNT
  'redirect' => 'http://localhost:8000/auth/facebook/callback',
],
```

### Step 7 - Validate Authorization Code, Client ID and Client Secret

Google Authorization Server validates the received authorization code, client id and client secret from the OAuth Application.

### Step 8 – Send ID and Access Tokens

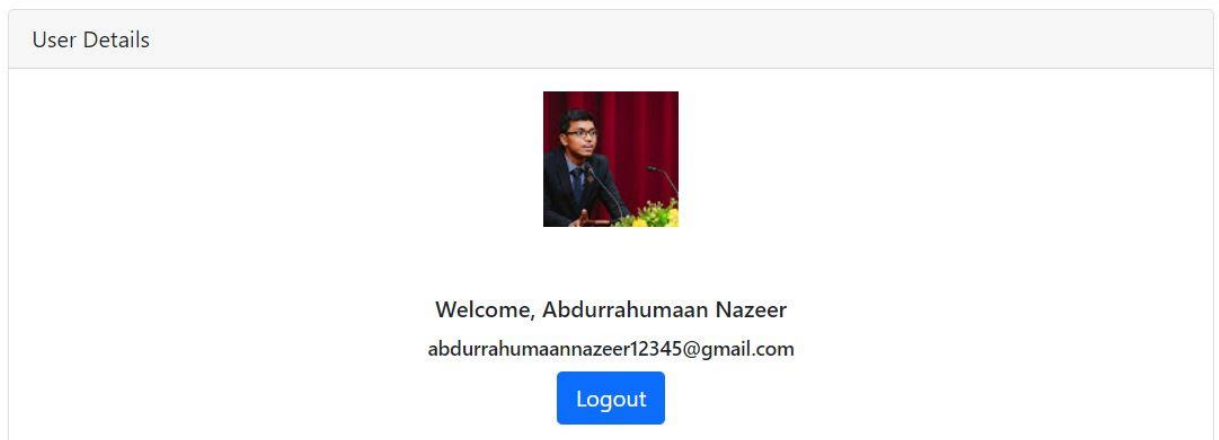
After the validation of the authorization code, client id and the client secret in the Google Authorization Server, it sends the ID token and the Access token to the OAuth Application.

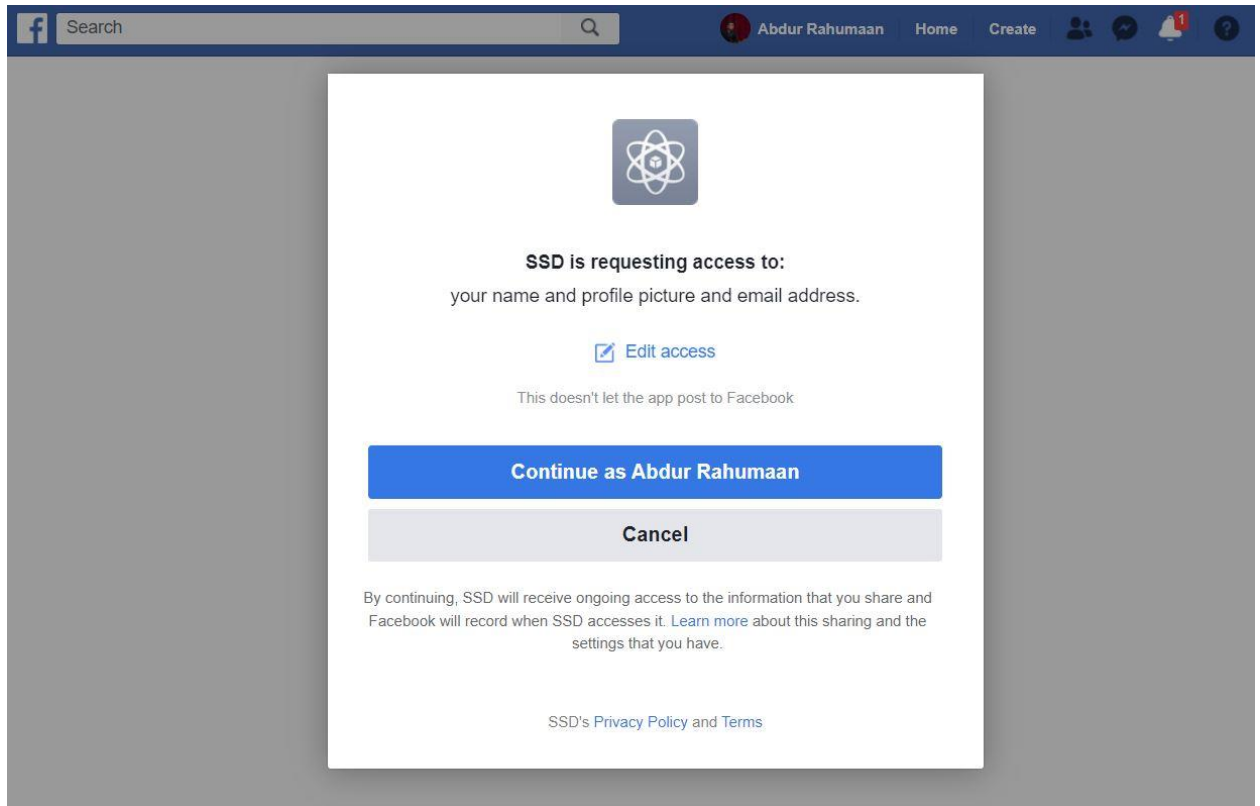
### Step 9 – Request Resource Using Access Token

Since OAuth Application has the access to the resource in the signed in google account, it sends a request by invoking the resource server API (OAuth Application API) asking for the user profile picture and email address by utilizing the access token.

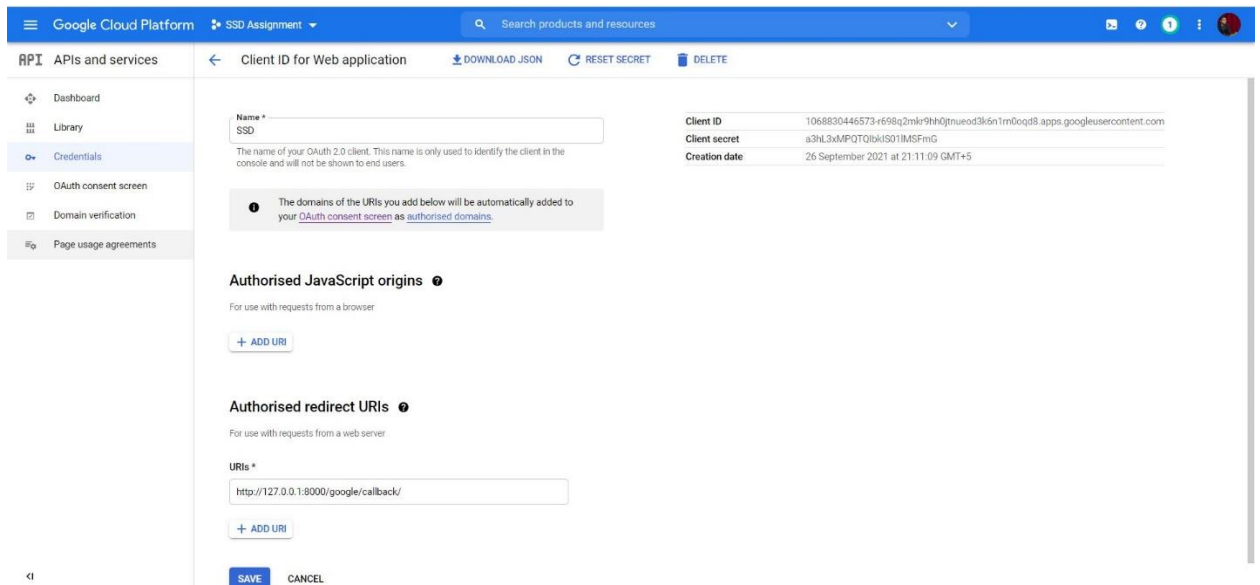
### Step 10 – Send the Response to the Request

OAuth Application API sends the response to the OAuth Application, along with the profile picture and the email address. Since this is the final step of the message flow, the final result of the implemented application.





## Google Cloud Platform



# Facebook Developer Platform

FACEBOOKfor Developers

DocsToolsSupportMy Apps

Search developer documentation

SSDApp ID: 884791332409556In developmentHelp

Dashboard

Settings

Basic

Advanced

Roles

Alerts

App Review

Products

Facebook Login

Activity Log

Activity Log

App ID

884791332409556

App Secret

\*\*\*\*\*

Show

Display Name

SSD

Namespace

App Domains

Contact Email

abdurrahumaannazeer@yahoo.com

Privacy Policy URL

https://www.privacypolicies.com/live/740e346e-e8d6-43e5-b20a-...

Terms of Service URL

https://icomtechnologies.lk/termsandconditions

App Icon (1024 x 1024)

1024 x 1024

Category

Education

Find out more information about app categories here



### **3. Conclusion**

In this assignment we were able to widen our knowledge on the OAuth 2.0 framework which includes the basic idea behind OAuth, versions of OAuth, different roles that were defined by the OAuth 2.0, purpose of the access and refresh tokens, how OAuth makes user's life easy by avoiding the usage of thousands of username and password combinations. Finally, we were able to implement a web application using OAuth 2.0 standard protocol, that has the capability to handle google single sign-on and display google account's protected resources such as profile picture and the email.

## References

- [1] S. Bandara, "An Introduction to OAuth 2.0", Medium, 2020.
- [2] M. Raible, "What the Heck is OAuth?", Okta, 2017.