

**SNAG@JOB**

CS551 ASE Project Increment2 Report



March 18, 2015

UNIVERSITY OF MISSOURI-KANSAS CITY

CS551 ASE Project SP15

By

Surekha Dani-ID#12

Srikar Reddy Mallareddygari-ID#33

Lavanya Kumar Somu-ID#46

Sandesh Puppala-ID#42

Table of Contents

[Import Existing Services/API 2](#_Toc412670725)

[Detail Design of Services 4](#_Toc412670726)

[User Stories /Use Case (Using Scrum Do) 4](#_Toc412670727)

[Service description 7](#_Toc412670728)

[Sequence diagram 9](#_Toc412670729)

[Sequence Diagram Description 9](#_Toc412670730)

[Class diagram 10](#_Toc412670731)

[Class Diagram Description 11](#_Toc412670732)

[Design of Mobile Client Interface 12](#_Toc412670733)

[Implementation 13](#_Toc412670734)

[Implementation of REST services 13](#_Toc412670735)

[Implementation of user interface (Mobile Apps) 16](#_Toc412670736)

[Project Management 19](#_Toc412670737)

[Work completed 19](#_Toc412670738)

[Description 19](#_Toc412670739)

[Responsibility (Task, Person) 19](#_Toc412670740)

[Time taken 19](#_Toc412670741)

[Contributions (members/percentage) 19](#_Toc412670742)

[Work to be completed 19](#_Toc412670743)

[Description 19](#_Toc412670744)

[Responsibility (Task, Person) 20](#_Toc412670745)

[Time to be taken 20](#_Toc412670746)

[Scrum Do Link 20](#_Toc412670747)

[Summary 20](#_Toc412670748)

[Iteration1 20](#_Toc412670749)

[GitHub Link 20](#_Toc412670750)

[Source Code 20](#_Toc412670751)

[Documentation 20](#_Toc412670752)

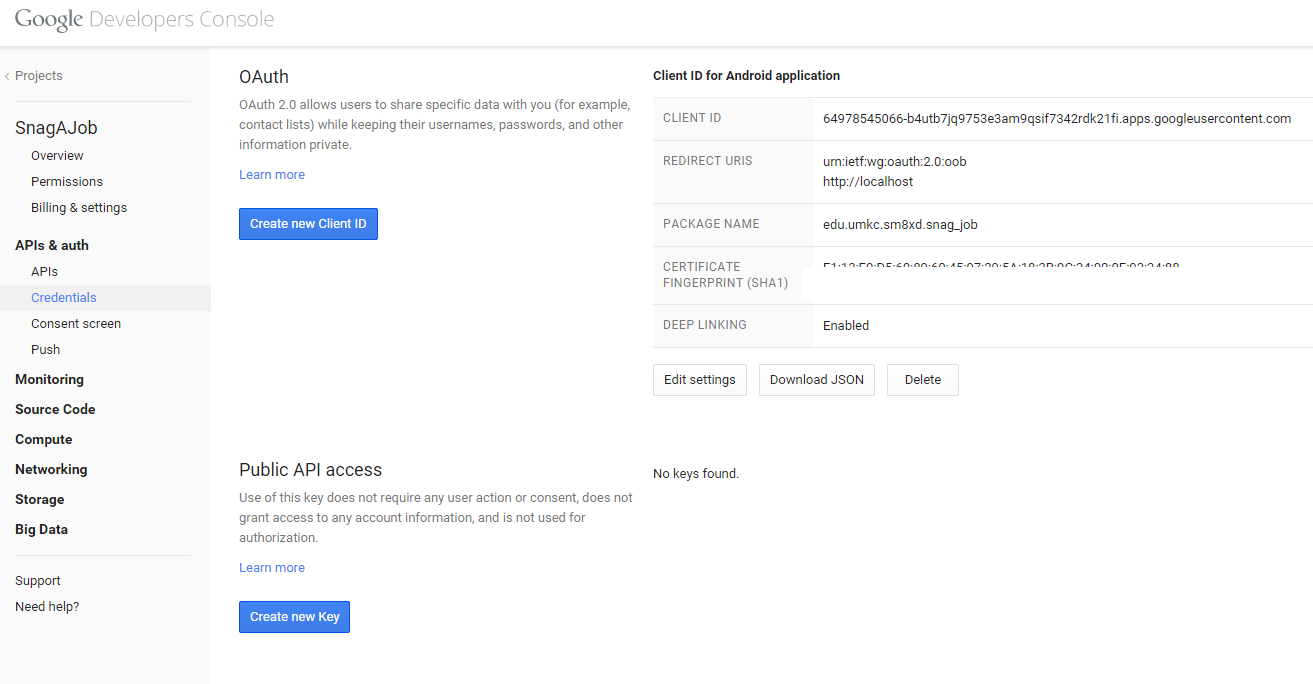
[Issues/Concerns 20](#_Toc412670753)

[References 21](#_Toc412670754)

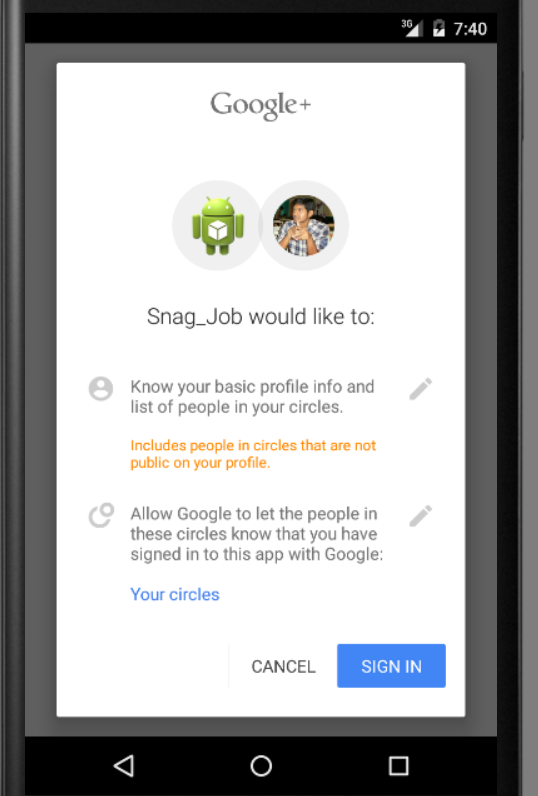
# Import Existing Services/API

**Screenshots for Google API integration**

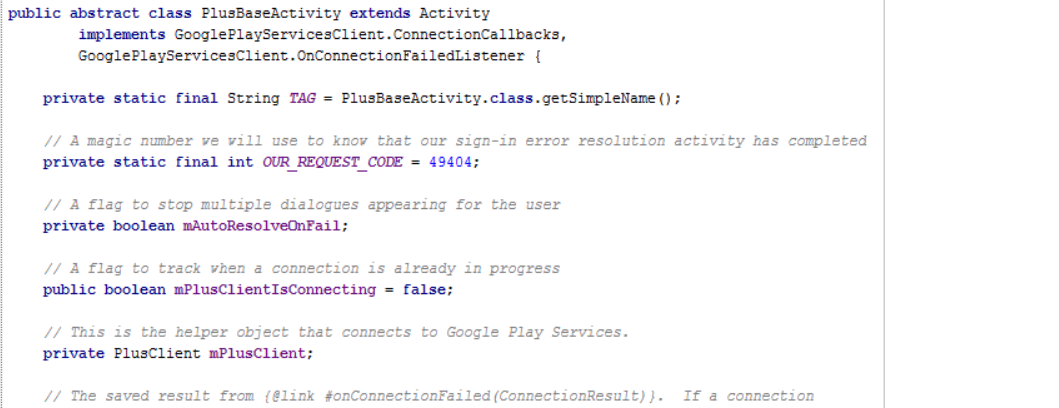
* OAuth Access Application Access for Google+ Sign In



* Google Sign In Validation for Mobile App



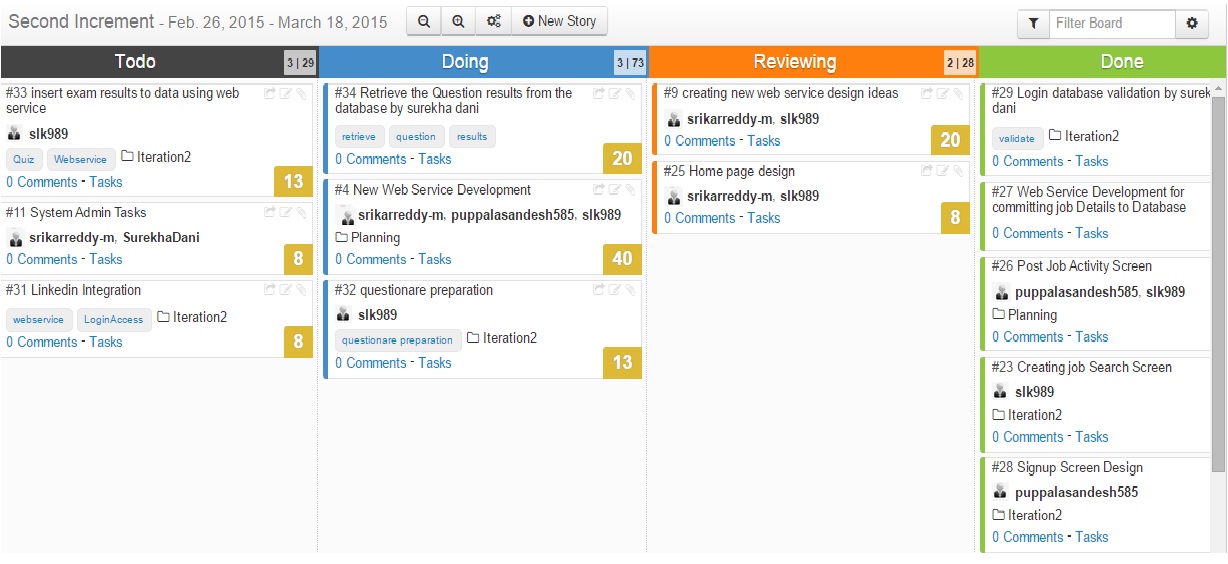
* Login Activity Class from Android Studio

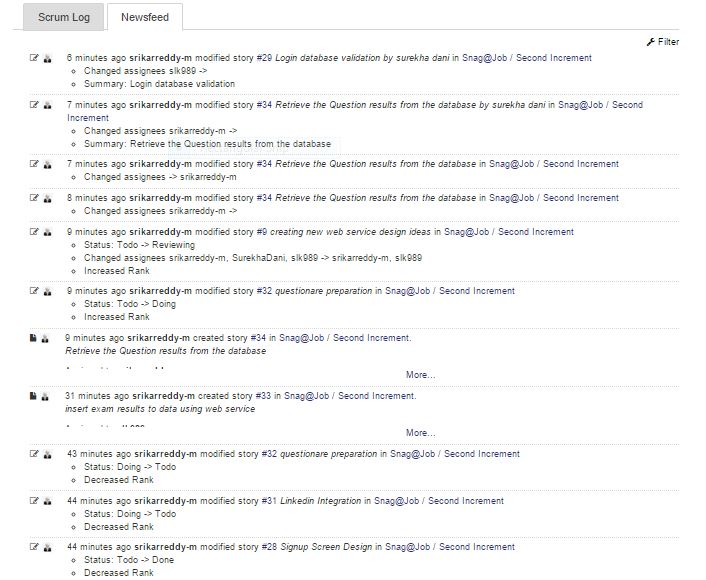
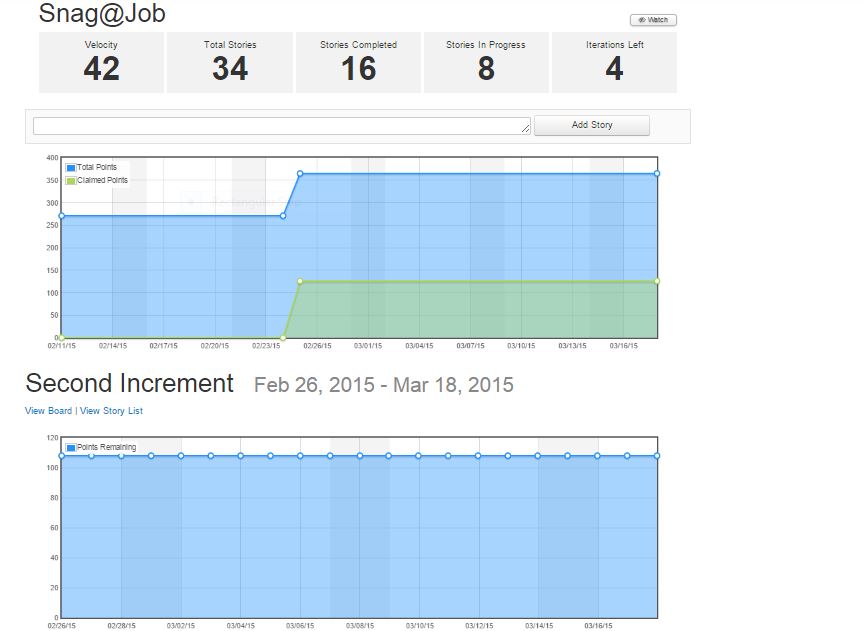


# Detail Design of Services

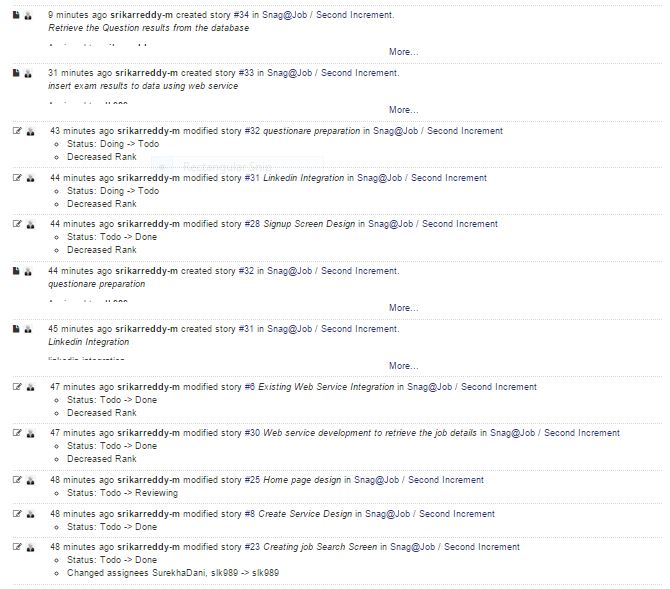
## User Stories /Use Case (Using Scrum Do)

## Scrum Board:



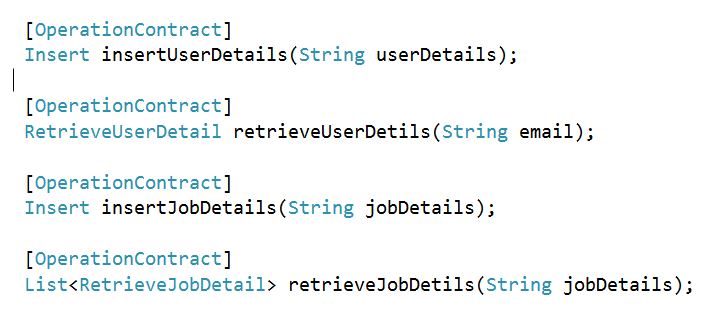


## 



## Service description

* **Snag job Web service:** A REST Web service is created Using Microsoft Visual Studio. It consists of two namely
  + - **insertUserDetails**
    - **retrieveUserDetails**
    - **insertJobDetails**
    - **retrieveJobDetails**



These two methods are useful to perform user registration and user login activities on a mobile application.

*InsertUserDetails:* This method performs storing of the user profile details to the MySQL server 2008 database.

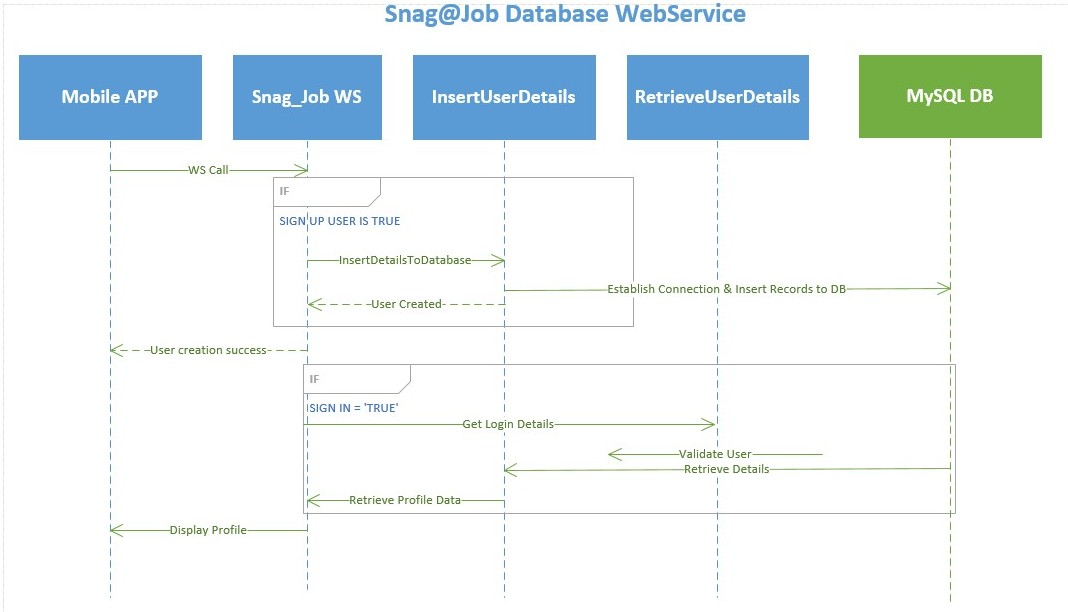
*RetrieveUserDetails:* This method is useful to retrieve the user profile details after successful validation of login credentials.

These two methods are useful to insert and retrieve job details as discussed below.

*InsertJobDetails:* This method is useful to insert the job details in to the mysql server.

*retrieveJobDetails:* This method is useful to retrieve the job details based on the search criteria.

## Sequence diagram

****

## Sequence Diagram Description

* A Web service named Snag\_Job is developed to perform a user’s registration and retrieval of the users profile data.
* When a user tries to Sign up in to the mobile application, the user details will be stored in to the MySQL DB using a Web service call.

<http://localhost:60838/Service1.svc/insertUserDetails/>

* When user tries to login in to the application, User credentials are authenticated through a Web service call. For authenticated users corresponding profile data will be displayed to the presentation layer of the mobile application.
* When employer logins into the app, he can post the jobdetails for a particular job.This is accomplished by call the insertJobDetails web service.

<http://localhost:60838/Service1.svc/insertJobDetails/>

* When applicant logins into the app and tries to apply for any job,the list available jobs are retrieved from the database by calling the retrieveJobDetails service

[http://localhost:60838/Service1.svc/retrievelogin/](http://localhost:60838/Service1.svc/retrievelogin/dani)

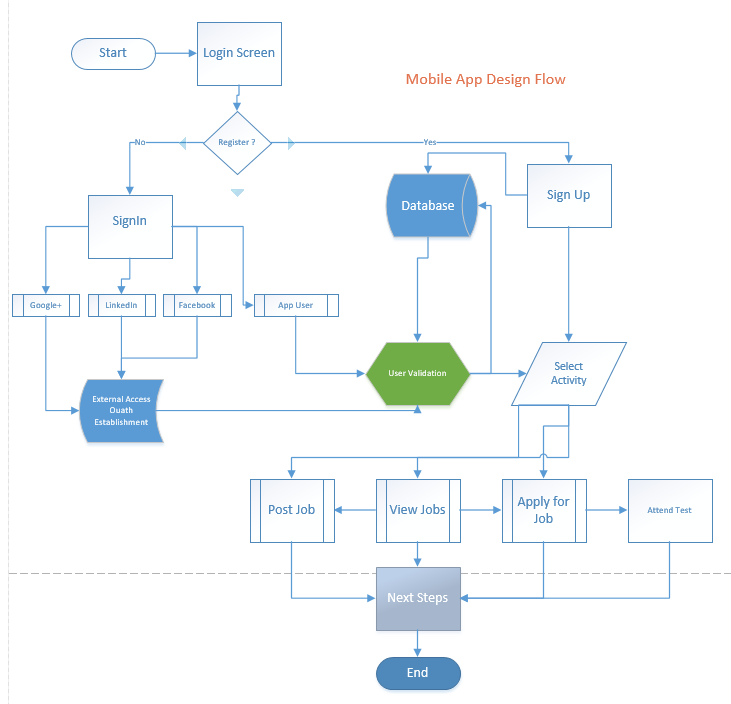
## D:\ASE\Class Diagram_Cropped.jpgClass diagram



## Class Diagram Description

* As shown in the above class diagram, we have implemented three different classes namely UserDetailsActivity, LoginDetailsActivity, SnagJobService.
* UserDetailsActivity class consists of various fields like first Name, last Name, phone, address, state, city and zip code along with their getter and setter methods.
* LoginDetailsActivity class consists fields email, password long with setter and getter methods.
* SnagJobService class contains two different methods of which once is responsible for insertion of user details while the other one is useful for the retrieval of details.
* PostJobDetailsActivity class consists of the job details to be stored into the database whenever employer post any job.
* ViewPostedJobsActivity class is used to retrieve the job details from the database when the employer wishes to view/edit the jobs he posted.

# Design of Mobile Client Interface

****

# Implementation

## Implementation of REST services

* Web Service Call for insertuserdetails:



* Web Service Call for retrieveuserdetails



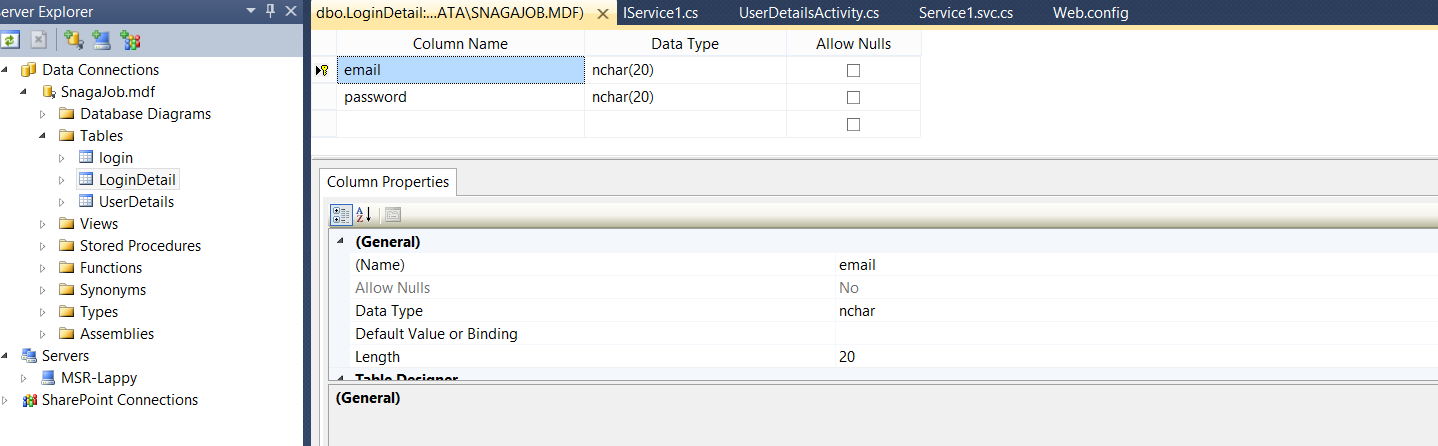
* Web Service Call for insertjobdetails



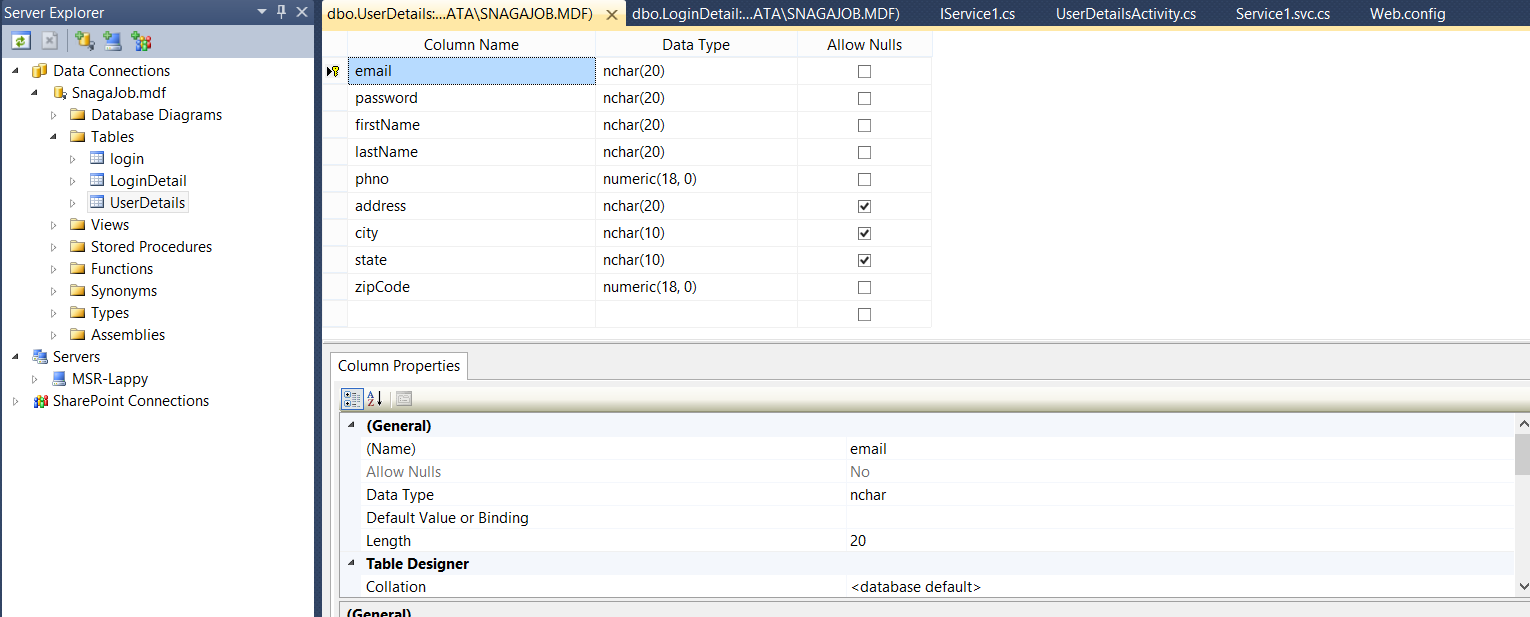
* Web Service Call for retrievejobdetails



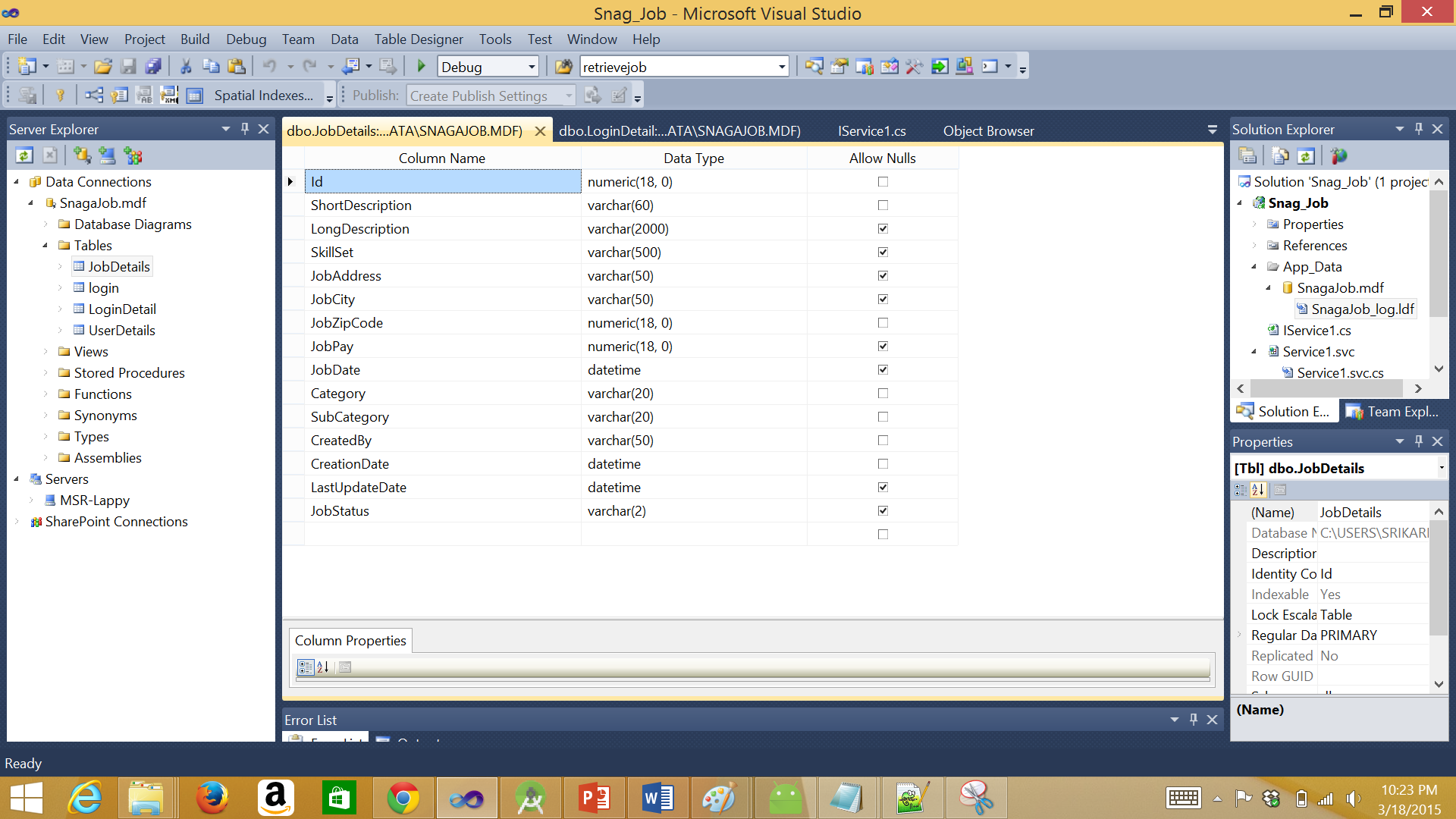
* Database Table Data Definition
  + Login Details Table:



* + User Details Table:

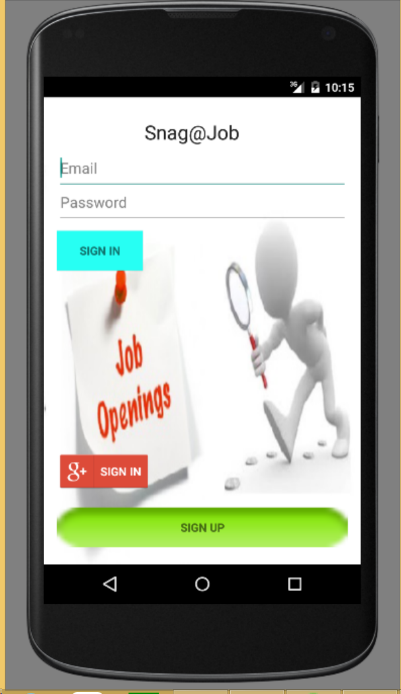


* + Job Details Table:

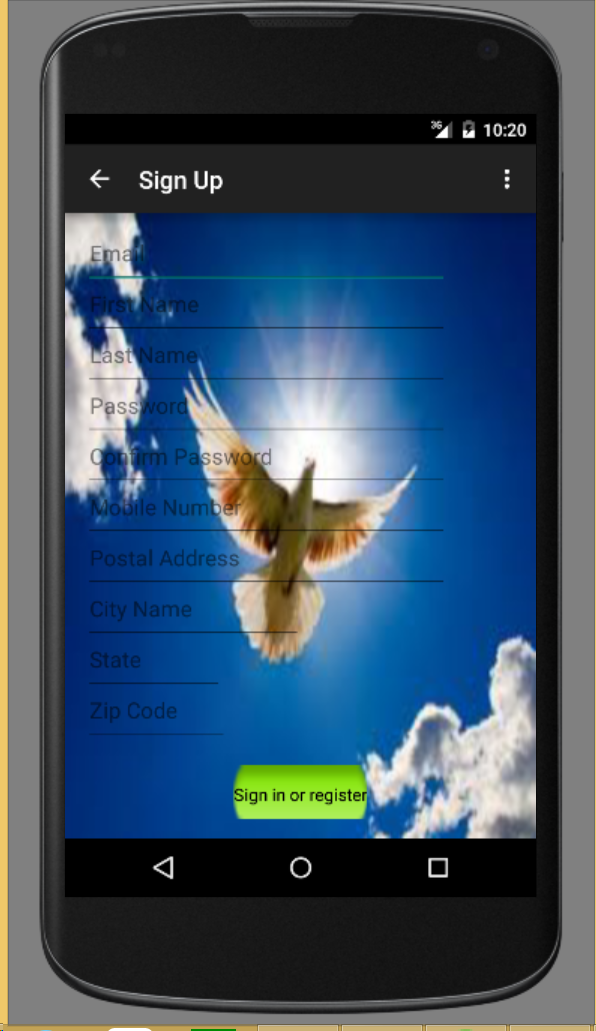
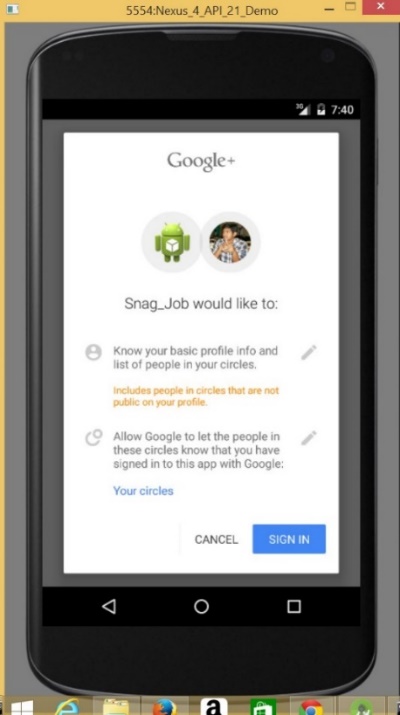


## Implementation of user interface (Mobile Apps)

* Login Activity:

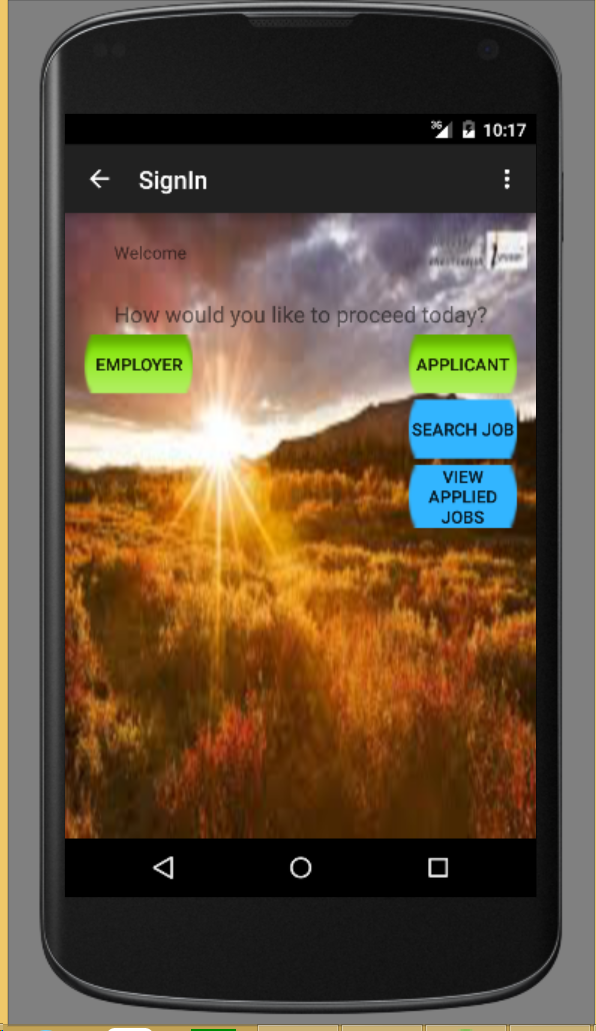
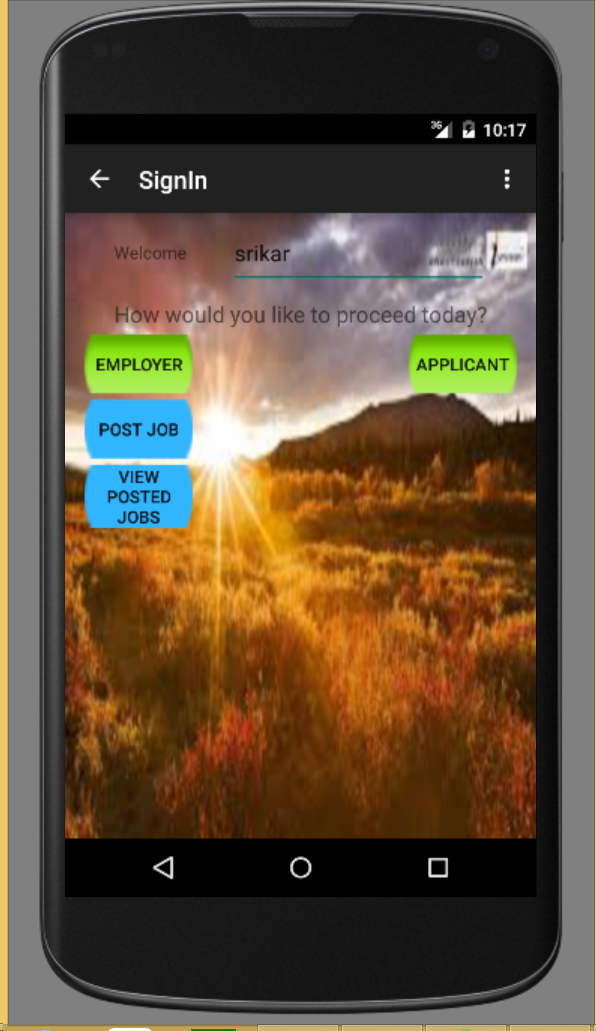
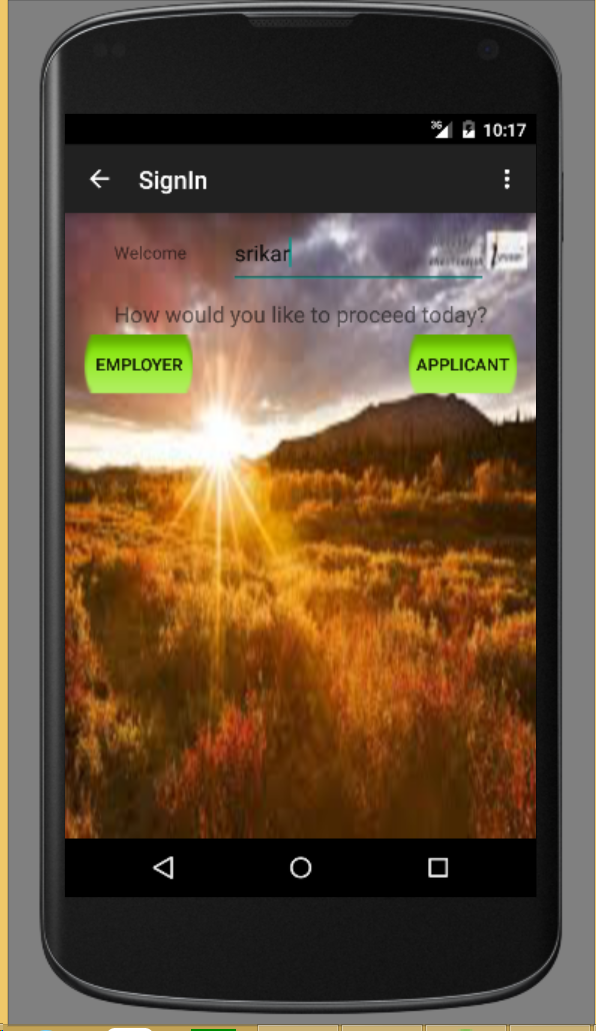


* This is the login screen of the mobile application.
* A user can sign in to the application using the google+ sign in, general sign in and a new user can sign up.
* A user require an email id and password to login to the application

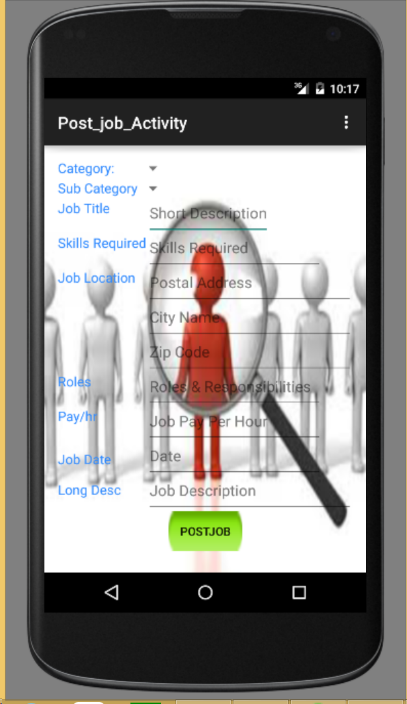
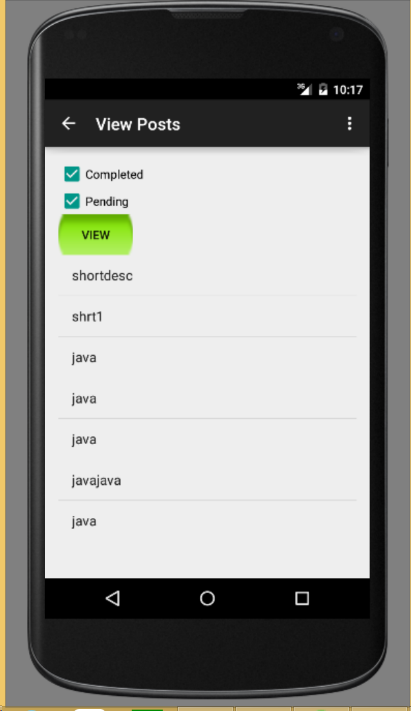


Sign in with google+ sign in tab.

Sign up Screen of the User.

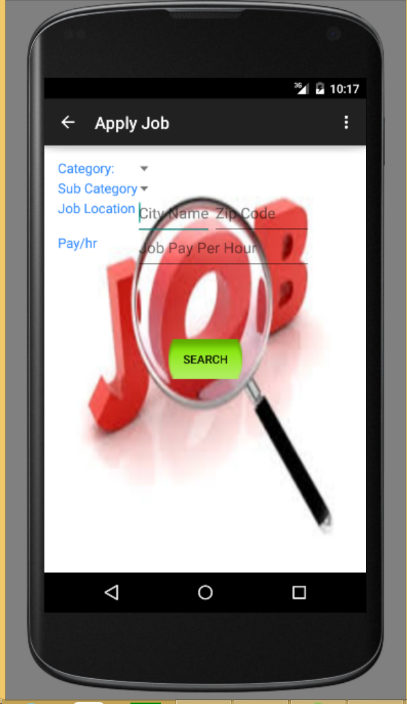


Home Page- Sign in Screen

Post Job Screen

View Posted Jobs Screen



Search Jobs Screen

* Implementation of test cases

-TBD- (As of now, we have issues in accessing the project class code using nunit tool)

Project Management

## Work completed

### Description

* Successfully created a REST Web service which is useful for the insertion and retrieval of users profile data over Webservice calls.
* Successfully created database tables using Microsoft Visual Studio.
* Successfully developed the Login Screen, Signup Screen and Post job User Interfaces.

### Responsibility (Task, Person)

* Login Screen, user profile table, Web service call for data insertion – Srikar Reddy Mallareddygari
* Signup Screen, Webservice call for data insertion, Class Diagram, Documentation – Lavanya Kumar Somu
* Post Job Screen, login details table, Web service call for data retrieval, Documentation - Surekha Dani
* Database Connection, Webservice call for data retrieval, Sequence diagram – Sandesh Puppala

### Time taken

140 Man Hours

### Contributions (members/percentage)

Srikar Reddy Mallareddygari – 25%

Lavanya Kumar Somu - 25%

Surekha Dani - 25%

Sandesh Puppala – 25%

## Work to be completed

### Description

Task1: Submit the post job details to database using Webservice, Develop Job search screen.

Task2: Retrieve and display the job search results based on search criteria.

Task3: Develop Questionnaire, Develop screens to conduct exam, Retrieve exam questions based on selected technical job criteria.

Task 4: Evaluate the exam, Display and store the results in database.

### Responsibility (Task, Person)

Task 1: Srikar Reddy Mallareddygari

Task 2: Lavanya Kumar Somu

Task 3: Surekha Dani

Task 4: Sandesh Puppala

### Time to be taken

300 Man Hours

# Scrum Do Link

## Summary

<https://www.scrumdo.com/projects/project/snagjob1/summary>

## Iteration1

<https://www.scrumdo.com/projects/project/snagjob1/iteration/121673>

# GitHub Link

## Source Code

<https://github.com/srikarreddy-m/CS551-ASE/tree/src/Project/Iteration2/Snag_Job>

## Documentation

<https://github.com/srikarreddy-m/CS551-ASE/tree/src/Project/Snag%40Job/documentation>

# Issues/Concerns

* Unable to add class Library under the snag job web service project. Currently we are working on this issue. If it gets resolved will add the test cases for the web services.
* Poor performance of android studio software
* Requires access to the centralized database for storing and retrieving data globally
* Slow performance of the web service call

# References

<https://developers.google.com/+/mobile/android/getting-started#step_1_enable_the_google_api>

<https://developer.linkedin.com/apis>

<https://social.msdn.microsoft.com/Search/en-US?query=how%20to%20insert%20into%20c%23&emptyWatermark=true&ac=5>

<https://developer.android.com/tools/studio/index.html>