**Bachelor of Science (Honours)**

**Computing**

**Semester 8**

**Module:**

**Enterprise Applications Development 2**

**Pair CA2**

**Android Project**

Lecturer: Gary Clynch

Submission Date: 31st March 2019

**Student Details: Word Count**

Aidan Dunne X00126748

659

Jordan Gardner X00126750

**Declaration**

The above-named students declares that the content of this Continuous Assessment project is solely the work of the individual whose name appears on this cover sheet. The work of any other authors has been cited and referenced in full.

# 

Table of Contents

[Introduction 3](#_Toc4935408)

[GitHub Repository 4](#_Toc4935409)

[Web Service URI Scheme 5](#_Toc4935410)

[Screen Shots of Android App 6](#_Toc4935411)

[Azure Deployment Settings 10](#_Toc4935412)

[Testing Report 11](#_Toc4935413)

# ****Introduction****

For the purpose of this assignment, We are to design and develop a RESTful web service and android app which then uses this web service. We have developed and deployed our RESTful web service to Azure, we have implemented swagger to make testing of the service much easier and our web service is using an SQL database to pull data from which is also deployed in Azure. Below we will supply a link to our GitHub repository containing all of our progress and source code. We will also provide our RESTful web service URI scheme, screen shots of our android app, our azure deployment settings and our testing report.

# GitHub Repository

The link to our GitHub Repository is:

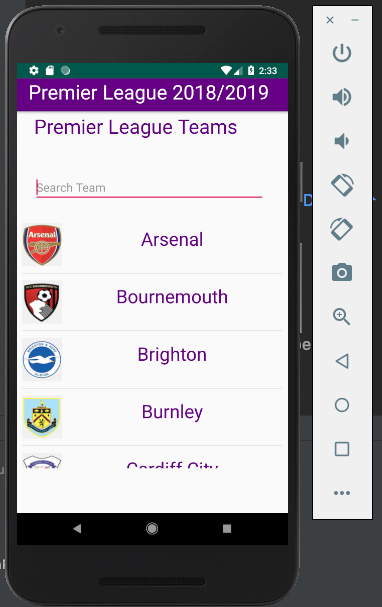
<https://github.com/aidandunne1892/CA2>

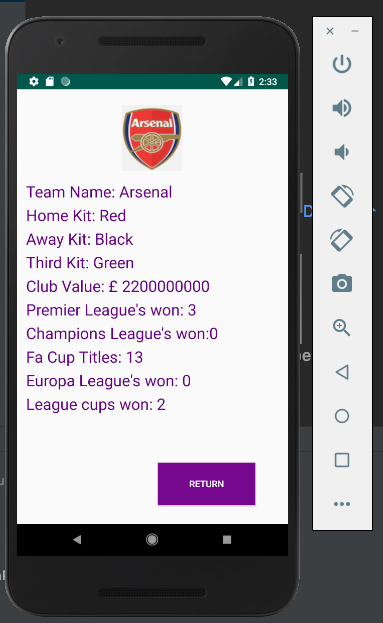
# Web Service URI Scheme

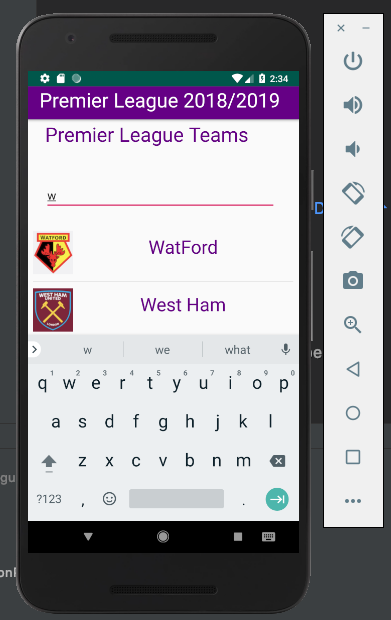
The URI scheme to our web service is :

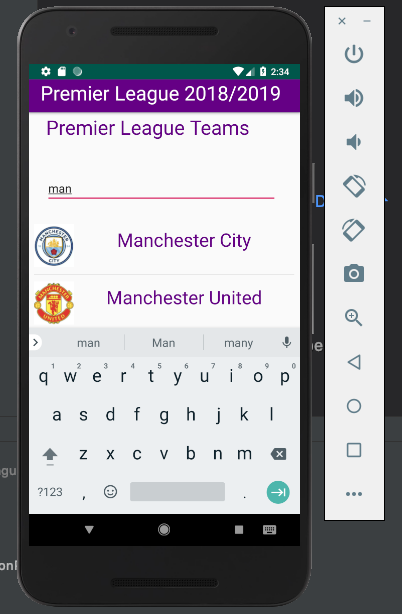
https://applicationbackend.azurewebsites.net

# Screen Shots of Android App





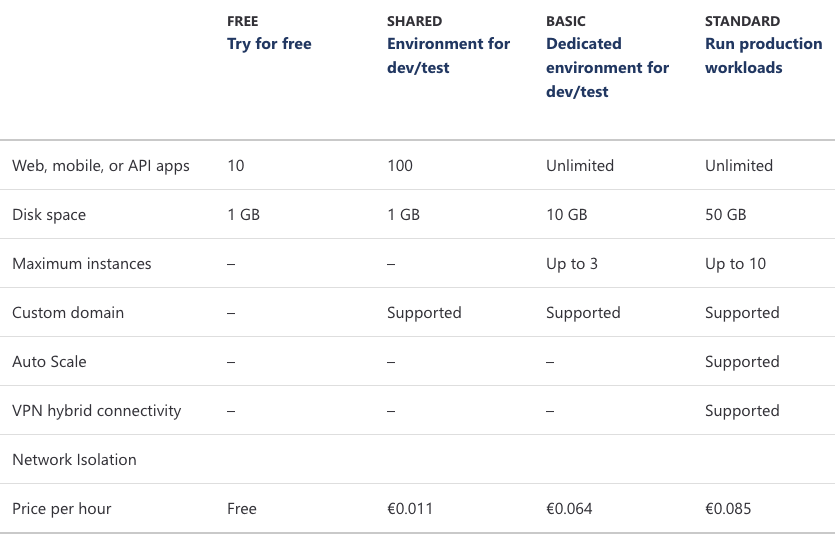




# Azure Deployment Settings

To deploy our RESTful web service to azure we used the standard approach of publishing from visual studio 2017 and then made some manual configuration changes on the azure dashboard.

As a team we selected the S1 pricing tier as we felt this provided the sufficient resources that our web service needed. The main resources in this tier are, 1 instance as a starting point, 100 total ACU which is Azure compute units, 1 core processing unit, 1.75GB of RAM and 50GB of storage for the app service. I have also enabled auto scale for our web service, if our web service CPU reaches a utilization percentage of greater than 70% another two instances are added to help spread the load.



# Testing Report

For the testing of the application, we tested using User Testing and Espresso UI Testing. For the user AI we decided as a team that people who aren’t tech savvy, would be the best people to give it to as they would give us feedback, that we would not have considered for example one user said they would like a way to look for a team without having to scroll this was then implemented and giving back to see if there were any problems.

For the espresso testing, we decided to have two test classes one for the user not interacting with the search bar and the other for the user interacting with the search bar.

In the first class we have 5 tests, the first test checks that the teams that the user would expect to see are displayed upon opening the app. The second test checks that the user can select a team and press return to get back to home screen. The third test checks that the user can select a team and press the back button to get back to the home screen. The forth test checks that the user can press a team then get back to the home Screen via return button then select another team. The fifth test checks that the user can press a team then get back to the home screen via the back button then select another team.

The second class has only two tests. The first test checks that the user can search and select team, then return via the return button and finally select another team. The second test

checks that the user can search and select team, then return via the back button and then the user selects another team.