ITT DUBLIN
Institute of Technology Tallaght
Institiúid Teicneolaíochta Tamhlacht

UEFA Ranking System

Android App and Web Application

4th Year Computing Project

Supervisor – Gary Clynch

Submitted by

Daniel Fulham – X00090749

Table of Contents

Research Document

Detailed Discussion

The proposed project is the development of an Android Application that will function as a UEFA Ranking System utility.

The Application will be developed to do the following:

Web API – Pull down League Tables from an official UEFA affiliated Site. This will ensure up to date data for the Database and the Application to then correctly process the correct information for accurate Rankings.

The Web API will also pull down the UEFA Ranking for each team that is required. The data will be obtained through GET requests within the API. There are several API services for obtaining football information such as OPTA, ESPN, livescoreboard, xmlsoccer that are free services and will be used to retrieve live data for the application to use all necessary data required for its functionality.

There will be many permutations implemented to ensure the ranking is calculated and allocated correctly. This is vital to the system functioning correctly and therefore a very high level of attention to detail whilst programming will need to be performed on a constant basis.

The Application will offer the ability for the User to request a real time update on the current rankings. This will be implemented in the system through the Web API and Database as the latest information will need to be taken from the appropriate League and UEFA ranking page and then processed for an up to date rank.

A week by week 'main' update on the latest ranking and standings will be done. The application will have the ability to compare the rankings of that week with previous weeks. The capability of 'predicting' final rankings will be implemented by comparing changes in ranking standings over a series of weeks. The user will have the ability to perform these comparisons and see the predictions for themselves.

There will also be a section that will display all the teams in one large ranking system. Each team will be ranked as per their current UEFA Ranking. This will be quite dynamic because teams will change in rank on a weekly basis. A search function will be implemented and be made available for the user to use for easy navigation to find the current ranking position of their desired team.

When the user opens the Application initially, an option will be included to allow them to choose their own team and this will show them their team's current progress. The Ranking for Qualifying Rounds will be quite broad and easy navigation will be essential for good accessibility for Users trying to track their own team's progress.

A Country Ranking section will show the Ranking of Countries in Europe. There are several permutations once again that will be required for correct calculation. Country ranking has a direct impact by results of European matches from teams in their Country.

It should be noted the UEFA ranking system is quite complex and due to the several permutations involved it can be quite difficult for anyone trying to manually predict what Qualifying round their team will be in and whether they will be seeded or unseeded in that round. There is a massive gap in the market for such an application and will be of benefit to Fans, statistic reviewers but more importantly the clubs themselves. Considering the money involved for getting through qualifying rounds in Europe is so substantial for the board of a club to know in advance they will be seeded could have a major influence on their club budget.

For the actual Qualifying rounds teams are allocated to one of four rounds of qualifying. This is set by the country ranking. For example Countries ranked 49-54 in the Country rankings will have the club that wins their league go in to the first round of Qualifying.

Those ranked 17-48 will have the club that wins their league enter round 2. This continues until all clubs have been allocated a Round. Once again there are exceptions which will effect what round a club will go. These complex permutations will need to be implemented correctly otherwise the ranking will not be correct.

The Ranking of a team is Calculated over Five Years. The application will correctly calculate the latest ranking for each team based on the Round they get knocked out in. Further complexity will occur for teams in the Group stages as results there will affect the team rank. The application will need to be implemented to support this and give a correct ranking for that team.

The app will be developed with different screens for certain sections. There will be a general template screen that most pages will use as a display however the specific sections such as the Qualifying Rounds/Info section/Country Ranking will vary in size and density to accommodate the level of complexity that is required for the data it needs to display to the user.

The application will be internationalized to broaden the pool of users that can use the application. Advantages of rankings are that they are universal to readers though menu sections, titles will need to be altered in language. The languages for support will be English, French, German and Spanish for initial release.

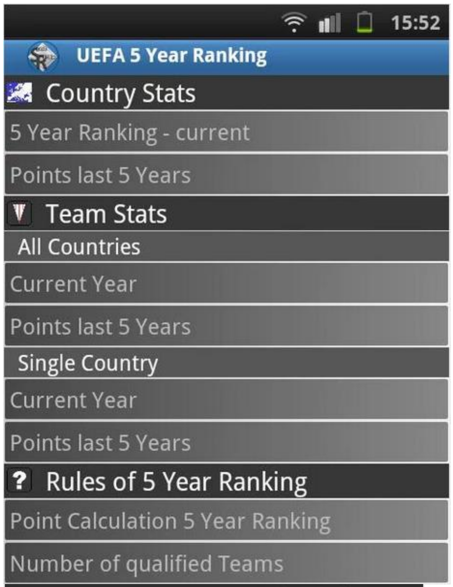
Existing Applications

Having researched both Android and Windows Store Markets there are some Applications with some relation to the proposed Idea however they are quite limited and do not provide much functionality with regards to the Ranking System.

Features of similar Applications

	UEFA-Ranking System (Proposed Idea)	UEFA Ranking (Windows phone)	5 Year Ranking of the UEFA (Android)
Team Ranking(Current)	☒	☒	☒
Country Ranking(Current)	☒	☒	☒
Team Ranking(Next Year)	☒	☒	☒
Country Ranking(Next Year)	☒	☒	☒
Information Section for Ranking	☒	☒	☒
Search Function	☒	☒	☒
Qualifying Rounds	☒	☒	☒
Real Time Updates	☒	☒	☒
Qualifying Predictions	☒	☒	☒
Personal User View	☒	☒	☒
Index and Contents section	☒	☒	☒

5 Year Ranking of the UEFA (Android)



Country Stats

- 5 Year Ranking - current
- Points last 5 Years

Team Stats

All Countries

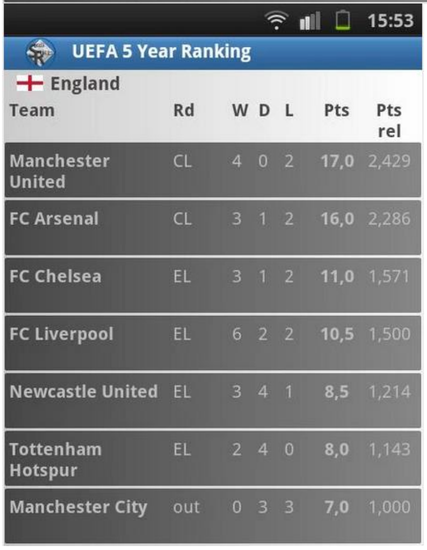
- Current Year
- Points last 5 Years

Single Country

- Current Year
- Points last 5 Years

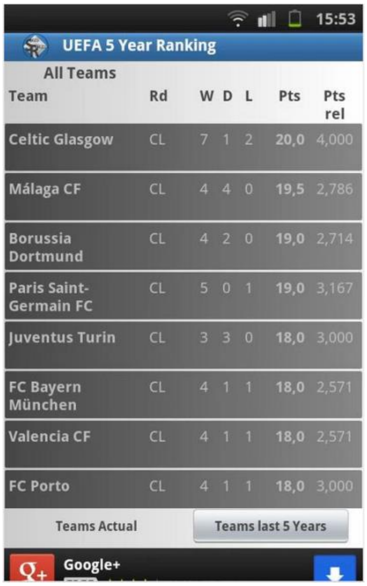
Rules of 5 Year Ranking

- Point Calculation 5 Year Ranking
- Number of qualified Teams



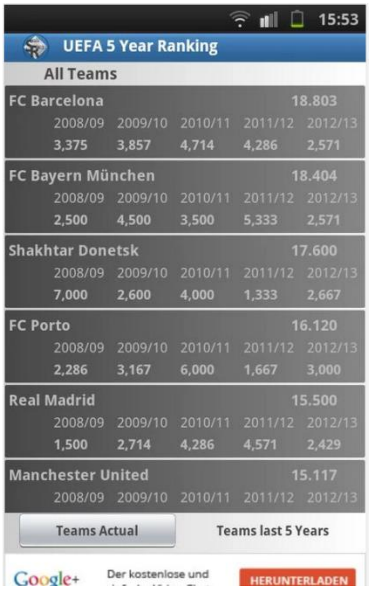
England

Team	Rd	W	D	L	Pts	Pts rel
Manchester United	CL	4	0	2	17,0	2,429
FC Arsenal	CL	3	1	2	16,0	2,286
FC Chelsea	EL	3	1	2	11,0	1,571
FC Liverpool	EL	6	2	2	10,5	1,500
Newcastle United	EL	3	4	1	8,5	1,214
Tottenham Hotspur	EL	2	4	0	8,0	1,143
Manchester City	out	0	3	3	7,0	1,000



All Teams

Team	Rd	W	D	L	Pts	Pts rel
Celtic Glasgow	CL	7	1	2	20,0	4,000
Málaga CF	CL	4	4	0	19,5	2,786
Borussia Dortmund	CL	4	2	0	19,0	2,714
Paris Saint-Germain FC	CL	5	0	1	19,0	3,167
Juventus Turin	CL	3	3	0	18,0	3,000
FC Bayern München	CL	4	1	1	18,0	2,571
Valencia CF	CL	4	1	1	18,0	2,571
FC Porto	CL	4	1	1	18,0	3,000

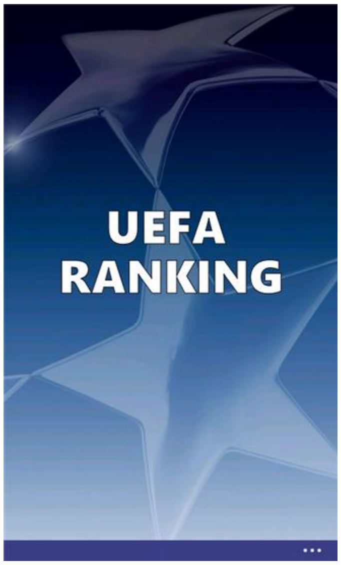


All Teams

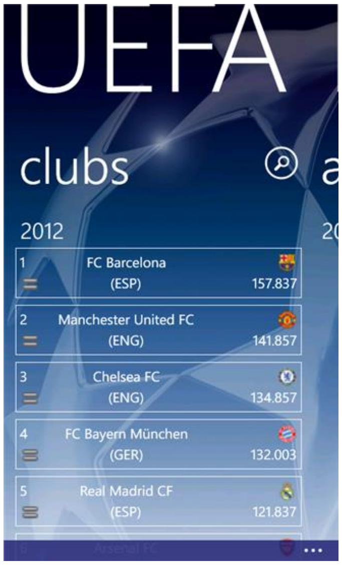
Team		2008/09	2009/10	2010/11	2011/12	2012/13
FC Barcelona	18.803	3,375	3,857	4,714	4,286	2,571
FC Bayern München	18.404	2,500	4,500	3,500	5,333	2,571
Shakhtar Donetsk	17.600	7,000	2,600	4,000	1,333	2,667
FC Porto	16.120	2,286	3,167	6,000	1,667	3,000
Real Madrid	15.500	1,500	2,714	4,286	4,571	2,429
Manchester United	15.117					

(Wahl, 2014)

UEFA Ranking (Windows phone)


UEFA RANKING

…


UEFA

clubs a

2012

1 FC Barcelona (ESP) 157.837

2 Manchester United FC (ENG) 141.857

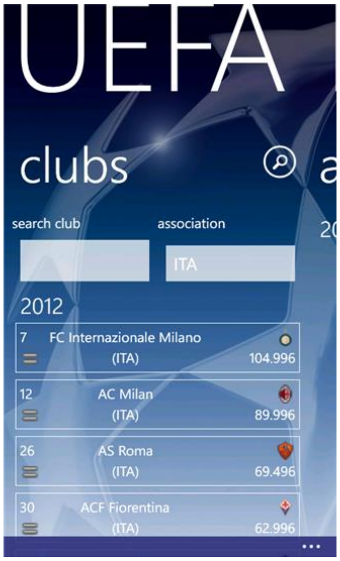
3 Chelsea FC (ENG) 134.857

4 FC Bayern München (GER) 132.003

5 Real Madrid CF (ESP) 121.837

6 Arsenal FC

…


UEFA

clubs a

search club association

ITA

2012

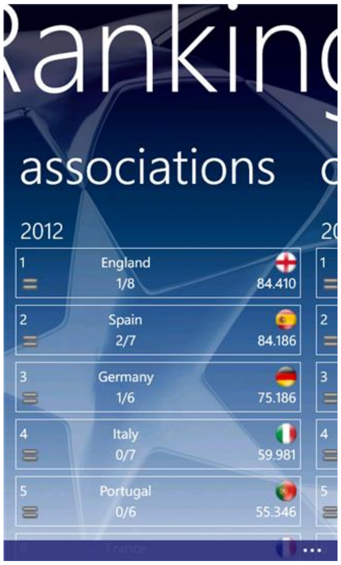
7 FC Internazionale Milano (ITA) 104.996

12 AC Milan (ITA) 89.996

26 AS Roma (ITA) 69.496

30 ACF Fiorentina (ITA) 62.996

…


Ranking

associations c

2012

1 England 1/8 84.410

2 Spain 2/7 84.186

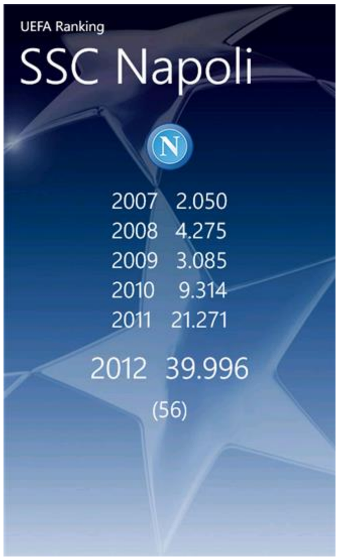
3 Germany 1/6 75.186

4 Italy 0/7 59.981

5 Portugal 0/6 55.346

6 France

…


UEFA Ranking

SSC Napoli

2007 2.050

2008 4.275

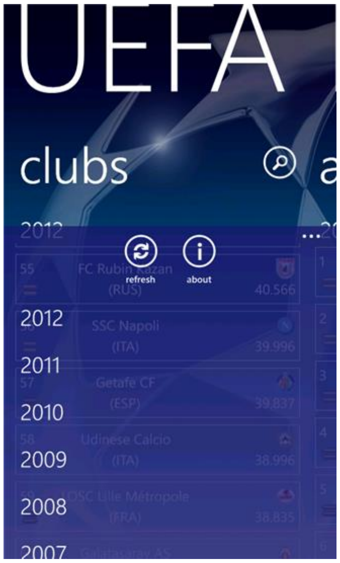
2009 3.085

2010 9.314

2011 21.271

2012 39.996

(56)


UEFA

clubs a

2012

…

refresh about

55 FC Rubin Kazan (RUS) 40.566

2012

56 SSC Napoli (ITA) 39.996

2011

57 Getafe CF (ESP) 39.837

2010

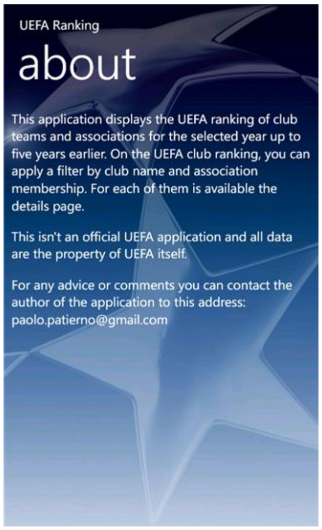
58 Udinese Calcio (ITA) 38.996

2009

59 LOSC Lille Métropole (FRA) 38.835

2008

2007 Galatasaray AŞ


UEFA Ranking

about

This application displays the UEFA ranking of club teams and associations for the selected year up to five years earlier. On the UEFA club ranking, you can apply a filter by club name and association membership. For each of them is available the details page.

This isn't an official UEFA application and all data are the property of UEFA itself.

For any advice or comments you can contact the author of the application to this address:
paolo.patierno@gmail.com

(Patierno, 2012)

Platform, Technologies and Libraries

Android

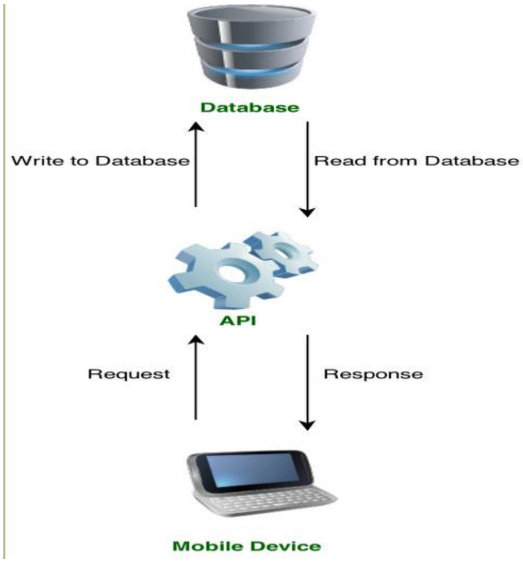
Android is a Mobile specific Operating System which has grown to become the leading Operating system on the market today. The Android market is currently the largest on the mobile marker. There are one billion android users currently with IOS users hitting half that figure(470 million). Android is a popular choice among users and the development of systems through Android is a preferred choice by many.

(Hay Newman, 2014)

An actual android Application is a mobile software application that is developed for use on Android devices powered by Google's Android platform. The Apps can be developed by anyone and also placed on the Google Play Store to be made available for over one billions devices to download. Apps that are developed can come free of charge to the user or given a premium price to which both the user(70%) and Google(30%) will gain proceeds from.

(Stroud, 2014)

An android application can be broken down in to the Application itself, the API and the database. When the application is functioning it can operate by requesting data from the API to which the API writes/reads from the Database, the API then executes a response back to the application.



(Johnson, 2013)

Web API 2

Web API is a framework for building web APIs on top of the .NET framework. It has the ability to work with mobile clients such as Android and Windows Phones. Web API 2 has the ability to be called from a Windows 8 phone and Android. On the foundation of things HTTP methods are the minimum required by the developer to perform a GET, POST, PUT and DELETE operation. Web API 2 can also support interactive applications that will allow that application to retrieve any information and data from the Web API Application.

Using Android for the Application for development calling from Web API is quite straight forward. In order to function correcting an operation of simply calling the API and then parsing the results is the basics to what is required. This idea will rely on requiring live data from the internet for essential functions and Web API 2 is very strong to being the catalyst in that regard.

(Wasson, 2014)

SQLite

SQLite is an Open source database that can support an Android application. It is incredibly efficient considering it only requires approximately 250Kbyte at runtime. It is embedded within every android device providing full compatibility. The database is automatically managed by the Android platform, the developer is only required to define SQL statements for creating and updating the database.

(Vogel, 2014)

SQL Azure

SQL Azure is the database supplied when using the Azure Mobile Services and other services provided. It allows relational queries against stored data, search, data analysis and data synchronization capabilities. Data transfer is done through an XML-based format.

Krishnaswamy, Jayaram (2010).

Azure Mobile Services

Azure Mobile Services has the ability to provide a cloud-based backend service to Applications of various operating systems. It provides compatibility not only to Windows phones but also to Android and IOS along with various others. Android Developer tools are required in order to pursue implementing the service. The backend can be implemented through both JavaScript and .Net. The service is offered for free with limitations and the Developer will need to be registered with Microsoft Azure. Azure services tutorials on Microsoft's site shows the web service is efficient and straight forward in terms of being implemented and working correctly with a Mobile application.

(Azure.microsoft.com, 2014)

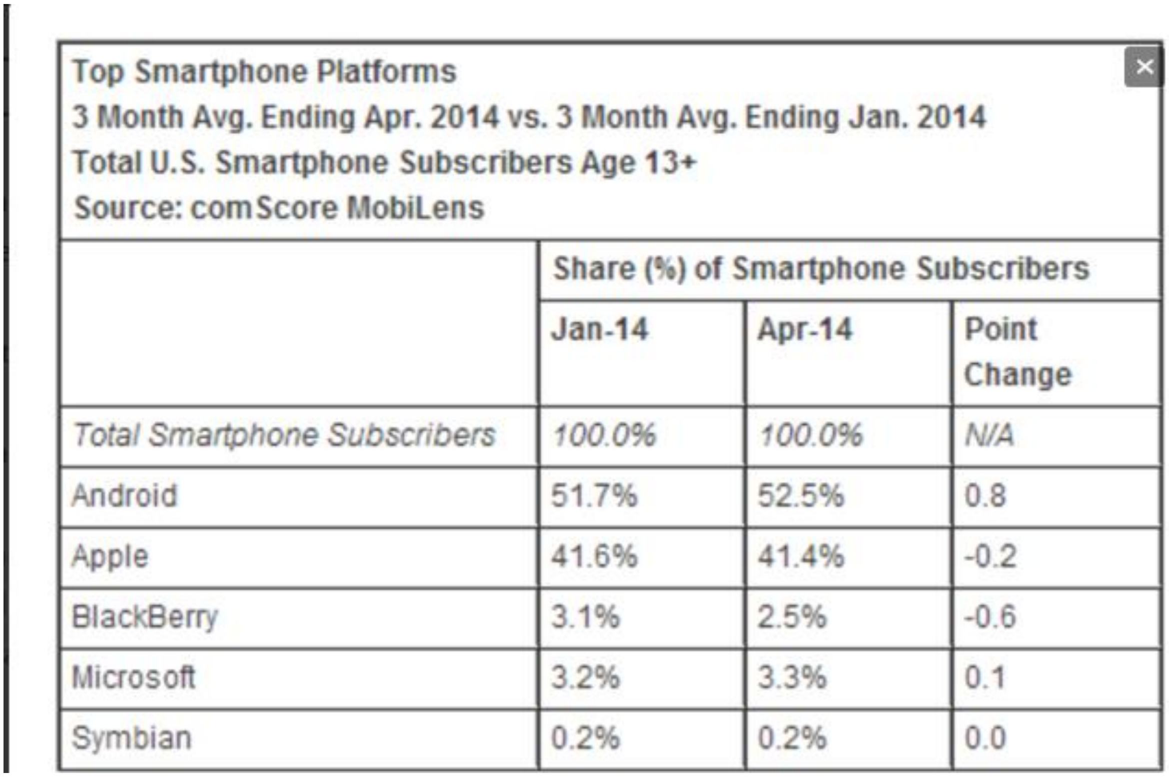
C#

C# is a programming language that offers the functionality to build native apps for iOS, Android and Windows Phone. It can be implemented through .Net and JavaScript. Many large applications such as the Microsoft Azure Service relies on C# for development. The actual interface is friendly with the majority of mobile development still being able to be implemented through Android Applications(Android Studio/Eclipse) and then the actual C# end of things being implemented through programs such as Visual Studio. C# is a strong force on the mobile market in terms of being used for development.

(Shackles, 2012)

Windows Phone

Windows Phone is a mobile specific operating system like Android and IOS designed and developed by Microsoft. It is an underdog in the market with Android and IOS users vastly overtaking the total users using a Windows Phone. It should be noted however it is on the rise and Microsoft have developed many things since such as the Azure Mobile Services, introducing better mobile devices and improving the Operating system itself to grow their total users in recent times.



Top Smartphone Platforms
3 Month Avg. Ending Apr. 2014 vs. 3 Month Avg. Ending Jan. 2014
Total U.S. Smartphone Subscribers Age 13+
Source: comScore MobiLens

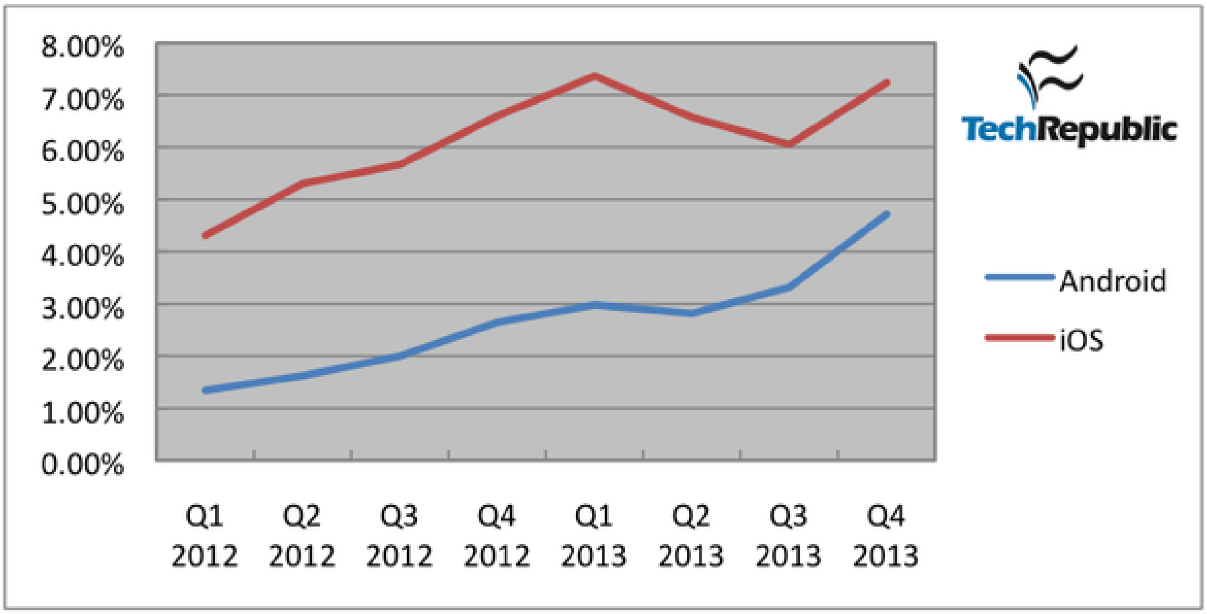
	Share (%) of Smartphone Subscribers		
	Jan-14	Apr-14	Point Change
Total Smartphone Subscribers	*100.0%*	*100.0%*	*N/A*
Android	51.7%	52.5%	0.8
Apple	41.6%	41.4%	-0.2
BlackBerry	3.1%	2.5%	-0.6
Microsoft	3.2%	3.3%	0.1
Symbian	0.2%	0.2%	0.0

(Surer, 2014)

In terms of choosing the windows phone for Application development the actual market size compared to IOS and Android isn't great. Windows have come a long way recently and their Azure Mobile Service is excellent however Android is where the market is, when developing an application hitting a large audience is essential.

IOS

IOS is the Mobile Operating System created, developed and offered by Apple on their IPhones. In order to develop IOS apps an environment of Apple devices are essential. A Mac computer, XCode and iOS SDK is required for initial development to create any Applications to work the with IOS operating system. Apple's official site offers a lot of documentation to their Operating System and also provides tutorials to learn how to develop an App of your own. Although there are double the amount of Android devices compared to IOS devices Apple have held their own on the market and the number of users with their phones is increases every year. Half the devices being used compared to IPhone's the amount of web usage is far exceeded by iOS users compared to Android. This is a very important factor to what platform to choose for App development because with more active iOS users there is actually a bigger market pool of active iOS users.



OS market share based on total web usage

Source: NetMarketShare

(Forrest, 2014)

Potential Risks

There are definitely a number of potential risks that have been identified. The proposed idea will require the use of the Database, Web Service and the Android Application to perform the majority of Functionality so it is essential none of those elements experience any issues.

The most complex functionality will be fully dynamic in that updates will be required from Websites in order to perform the algorithms for correct calculation of Ranking. The option for the user to execute Real Time updates could also propose many risks and again it is vital the coding is correct and the communication between all elements is correct for the Real Time update to be available at any time.

The Android Application will heavily rely on the web service and an internet connection in general. Limited functionality being available could result in large limitations and should be addressed so a user could possibly be provided with an offline offering of data to be able to use the majority of the Apps features.

Although the permutations are quite complex the actual programming of these should be done efficiently and correctly to avoid any bugs that could enter the frame due to the coding not being done in the best manner possible.

Upon research of the actual operation of Android Applications memory management and the management of running processes is very important for the Application to work right and for the Phone to run the Application in conjunction with shared services/resources. These will definitely be identified throughout development and any potential issues that may arise will put to action on immediately.

Security is a massive element to Development of any Application not only on Android devices. The majority of developed Android Applications will go on the Google Play store. There have been breaches to Apps on there with Malware the main threat that these apps have been hit by. It is also important the application is developed to the latest in date features, coding and implementation as a whole. Should an app be developed with outdated resources it can cause a threat to the phone itself due to security updates not being applied to deal with outdated apps since. A compliance policy could be introduced to the Application on release.

(Violino, 2013)

Technical Architecture

Use Cases

Title	Menu
Primary Actor	User
Scope	Category Navigation
Level	High Level
(Story)	User will see a display of different sections of the Application they can navigate to. These will include all the main sections including the Ranking, Tutorial, Qualifying round, team selection and about section.

Title	Team Selection
Primary Actor	User
Scope	Choose team for personal Ranking View
Level	Sub-Function
(Story)	User can select a team of preference and that will present them with the latest information for that team specifically (Ranking, League position).

Title	Ranking Update
Primary Actor	User
Scope	Update Current Rankings
Level	Sub-Function
(Story)	User can update the current rankings which will display all teams' positions in the current UEFA Ranking table.

Title	Tutorial
Primary Actor	User
Scope	Application Instructions
Level	Sub-Function
(Story)	User will be given a simple instruction of how they can use the application.

Title	Qualifying Rounds
Primary Actor	User
Scope	Qualifying Round Standings
Level	Sub-Function
(Story)	User can see the current standings of teams for the next Champions League qualifying rounds.

Software Components

The software will contain both a backend to perform the mobile services and a front end containing the android app development.

Back End

The backend is an Azure Mobile Service that uses .net framework. The Azure Mobile Service will be the cloud backend for the Application. It will be hosted through a .Net framework which will include 24x7 monitoring and management. The service can store data in to the Azure SQL and also allow Cloud-based sync. Through the .net framework it will have capabilities implemented to take data off both the UEFA Site and an appropriate site to pull off league tables. This data will then be used for a series of algorithms implemented through the Web API. The final data will then be provided to the front end (Java) for implantation of the actual android application.

The database to be used is Azure SQL. The will take a weekly storage of the current team ranking positions, qualifying rounds and information on positions for each league. This information can then be queried upon but this time to be used in a predictor algorithm for final ranking standings by taking standings from a series of weeks using the database.

Azure SQL includes excellent scalability, performance and security which are vital to the fundamental focus of the Application. It is self-managed so does not require much maintenance.

All functionality for the app will be developed here:

Examples –

1. Team Ranking – Algorithm for calculating the standings of **all** teams who have participated in UEFA competitions the previous five years. UEFA ranking is calculated over 5 years so an option to view the ranking for 2015 will be available. This will involve using the previous 4 years and the ranking points gained by each team this year (2014) to show rankings for 2015.

2. Country Ranking – Similar to the team ranking structure. A display of all countries in Europe by ranking position. Also calculated over 5 years, a similar algorithm will be implemented to calculate future years (2015).

3. Update current Rankings – An algorithm will be used to display the current ranking each week. A predictor algorithm will be introduced that will take a series of weeks of ranking and calculate (predict) what the final ranking tables could look like when all leagues finish.

4. Qualifying Round standings – Algorithms will be created to calculate the current standings in the qualifying rounds. The permutations for this are incredibly complex therefore an in depth level of coding will be required to do this correctly.

Front End

The front end will be a Java application developed in Android Studio. The Java coded will focus on taking in the final data from calculated algorithms in the API and then produce the different sections of the app with the data in a presentable view for the user.

A personal view will be developed to allow a user to login and choose a team of preference. Java will be coded to allow the users to have easy navigation to sections that will give them information on their team easily without have to manually find the information.

An instruction section will also be created that will present tutorials on how to use the application. Other sections include displays for Country Ranking, Team Ranking and the Qualifying rounds. Each is unique in their own display depending on the complexity of data to be presented to the user.

A section for displaying predicted final qualifying round standings will include some neat features such as color schemes for new teams that enter each week, teams that move to the seeded bracket for a qualifying round and teams that move to the unseeded bracket.

Platform Libraries

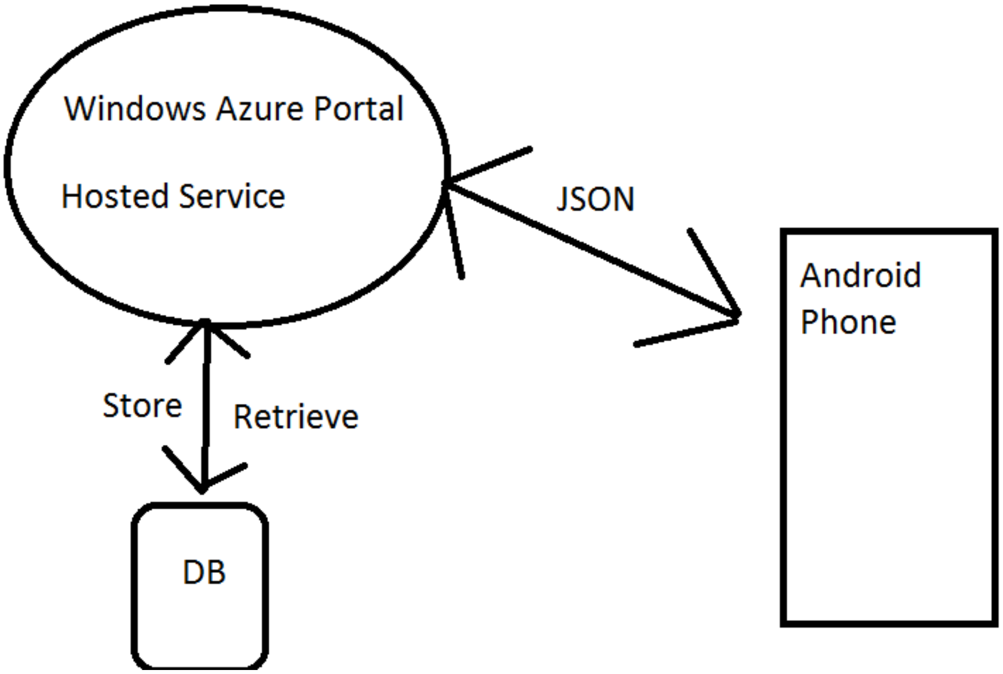
API –

- Web API 2.2
- ASP.Net

Languages –

- Java
- C#

Distribution and Deployment



The Application will function through the Azure Mobile Services Components.

The Azure will provide User Authentication, Push Notification capabilities, Scripts and Data. It will also connect to the Azure SQL Database for storage and retrieval.

The hosted service will retrieve information for the UEFA Ranking and League stats from the appropriate sites and provide final data for the Front End of Java development through JSON that is calculated through algorithms in the .Net framework.

The Backend is a .Net framework which is coded thoroughly with C#.

Security

Security will be vital for the Application and a high level of caution will be taken throughout development to prevent future breaches on release.

Application Layer – Android

1. Storage of sensitive data on the user's device will be very limited or non-existent. Whether the data is encrypted or not it can still be compromised and therefore essential storage will only be implemented with regard to sensitive data. The majority of data at run time will be stored in the RAM so any sensitive information will be cleared on application close.

2. Caching of the application data specifically the extent of it will be monitored throughout development. Should a registration section be implemented the caching of its URL history and page data for the actual web process could potentially be avoided. The Azure Mobile service has the capability of external login through Facebook/twitter, with a security element integrated for this case it could be beneficial for the user to login this way instead.

3. All query calls in the application must be protected. Should a query call be intercepted an attack could be executed that could breach the whole system. The .Net backend will use POST requests and temporary session keys which are secure for query calls and therefore will provide protection for any possible attacks.

4. Avoidance of crash logs occurring is crucial to the security aspect of the application. Logs provide data to the user and this could prove valuable in figuring out potential vulnerabilities in the coding. To prevent this ever occurring an extensive period of testing will be done throughout development to ensure the application does not crash. The application will be built without any warnings with assertion fails throughout should a potential error occur.

5. A Local Session Timeout will be employed not only for the protection of application data for the developer but also for the user. An easy breach of data includes the actual handset being used by unauthorised personnel and without a timeout this user can see all personalised data of the victim and perform all the permitted features they have when logged in. When the timeout occurs the user will be logged out and their personal team of preference for example will be cleared.

6. Debug logs like crash logs can store data which can be valuable to a user looking for breaches within the system. These debug logs could potentially be disabled for early releases to dilute the amount of information the user will obtain from a debug log.

7. The complexity of the code will be vital for security in the application. If the code is implemented in a simple way with regard to the functions and methods it can be easily understood and therefore altered for a hacker in breach. Reverse Engineering is easy for Android Apps and therefore it is vital the code is obfuscated to prevent malicious users making sense of it should they manage to obtain it.

8. The keyboard cache feature on an Android Operating System could become a security breach with regard to cached information. This is a user dictionary that is available to all applications and for potential sensitive key inputs the auto correct feature could provide an unauthorised user with the full keyword of what a previous user had entered on the App.

9. The simple copy and paste feature is available universally throughout the Android Operating System. For pages that require sensitive data to be entered the copy/paste feature will be disabled to avoid any exposure of potential data.

10. The selection of Third-party libraries will require thorough testing to ensure it can be implemented correctly in the code. Any third party can be used but some can contain vulnerabilities and weaknesses therefore the library selection is vital on the security side of things.

11. A snapshot preventer will be introduced for some pages with sensitive data to ensure they cannot be screenshotted/captured for viewing outside of the App itself. This will stop any easy means of taking the information for use other than on the Application.

12. For any files that need to be created (cache storage) it is essential there are no permissions on those files to be readable or writeable. Despite being stored in the private directory for that App it can still be accessed.

13. The implementation of intents is important and should be done in a careful manner. Starting services for example need to be done implicitly to make sure the right service is responding. Components accessed through intents should preferably be private for security.

14. Checking Activities is essential when implementing an app for security reasons. Despite being private a hacker can find ways around accessing authentication screens through an Activity in an Unlock state. With these potential scenarios in mind input validation is very important in case bad data gets inserted for these situations.

15. A series of Tamper checking will prevent an app being accessed through a back door or resigned by a hacker. A public key check that is read by the app's certificate can verify the app has been signed with the developer's private key.

16. Despite deleting files from the Application they can still be retrieved before they have been overwritten. This is important to note and another reason why sensitive data should be limited within local cached files.

17. Logic checks must be evident on the Azure Cloud to validate all potential input from the User. Attacks such as crash logs, buffer overflows can be done so it is vital checks are done to prevent any breaches.

18. Session Management will be done through the .Net framework to enhance security. Hosting a session outside of these session managements is risky and significant expertise would be required to have it done correctly.

(ViaForensics, 2014)

Prototype

Prototype Deliverable for week 8

The first prototype will be a basic functioning Android Application. The application will run in conjunction with an Azure Mobile Service and interact with the Azure SQL Database.

There will be a simple menu with hard coded examples of Ranking information. The actual GUI design (screens etc.) will set a very good gauge to how the proposed view will look for Release.

The testing strategy will involve complex data being inserted to the application for calculation and this will test the algorithms to ensure it runs smoothly. JUNIT testing could also be done through Android Studio to test different sets of data being inserted in to the algorithms to ensure efficiency, smooth scalability and correct output.

Prototype Deliverable for week 11

The second prototype will again provide basic functionality however it will work with JSON data that will be retrieved from appropriate web sources through the Azure Mobile Service. This data will contain ranking and team information that can be passed in to the algorithms for calculation of Rankings.

The testing strategy will involve different JSON data being inserted in to the Java application and in to Algorithms. Various levels of data will be inserted to fully test the capabilities of the implemented algorithms.

Risks

There are some potential risks that may affect the delivery of the project. When doing the calculation of ranking all of the permutations **must** be implemented correctly and to complete accuracy for the majority of the applications features to function. Should any issues arise with any of the permutations it could take up more time than proposed to solve and become a risk in affecting the delivery of the project.

The integration of security throughout the application is essential for the development. The Application should not be released without the proposed security features being included and running correctly. This could become a risk and affect the delivery of the project.

The GUI design for the Application is important and must be designed to suit a wide range of phones and facilitate their hardware specifications including screen sizes. Should the application be developed with a bad code design it can cause delays due to fixes being required to work with most phones and therefore affect the delivery of the project.

The GET operation must be configured correctly to pull the correct data from the chosen API. With the application being 'live' it is essential the correct data is received constantly as the algorithms cannot be calculated without the correct data. Should the data not be received correctly during development it could delay development and becoming a risk in affecting the delivery of the project.

Test Report

Overview

Overall there was quite an extensive set of tests conducted throughout the System to verify all functionality were running flawlessly, error free and bug free. The Testing was conducted via the following use-cases:

- Unit-Testing
- Integration-Testing
- System Testing

The majority of functionality was integrated on the Server side, more specifically within the API Controllers and the corresponding models. These were the core part of functionality and a lot of validation and algorithmic functions are found here. To test the API controllers and models a series of Unit Tests were performed to ensure all the methods worked correctly I conjunction with their relative models.

In total over 40 Unit tests were carried out across all the API controllers. The following tests were conducted:

- GET – Return Team with Same ID (comparison)
- GET – Return all Teams checking for Nulls and successful comparison
- POST – Create Team and return with comparison
- PUT – Update a Team with a specific ID (e.g. /5) ensuring no status codes are thrown and a comparison of the change is successful
- PUT Failure – Performing an update but with different ID’s. Comparison will be different and the Test should pass should a Bad Request be picked up.

The Unit Tests above covered the main API methods that were implemented throughout the project. It was important all of these were tested to ensure they were all functioning correctly. If no testing was done for these sections there may have been potential bugs within any element of the API and errors may have arisen that could have potentially broken the whole system or caused an incorrect calculation of the UEFA Ranking System.

The testing was quite extensive and a lot of time was allocated to generating them and covering all possible approaches that can be made to the API that could break it. The Unit testing was quite complex due to the fact the API works with an Entity framework. In order to perform Unit Testing on the API the Entire Entity Framework had to be mocked. All the controllers had to be modified to enable the passing of a context object for testing. Custom Test Controllers and a mock interface of the Database Context had to be created in order for the Unit Testing to be conducted. This was quite a complex piece of integration however it was essential the API was fully tested as it is the Core functionality to the whole System.

Testing was also conducted for the MVC end. All of the Model Classes contained validation and some regressive testing was conducted from the Administrator Web Client to attempt to break the Database through inserts. Several inputs were tried and were all picked up by the validation.

Tests were conducted against the SSL (HTTPS) ensuring the MVC Web Service could not be accessed through a normal HTTP connection. There were also tests done against a normal user and guest attempting to access the Admin only sections. The Admin has rights to all PUT, POST and DELETE methods within the API and the Web Service. Any sort of access from a normal user and guest was blocked from a series of tests. Currently all attempted access will redirect the guest/user to the login field.

System testing was conducted for the front end of the client. Several attempts were made in various ways to try breaking the Application.

The attempts to break the application included the following:

1. Disconnecting the Internet connection – Before opening the app, during an async call, when on a page containing Server Side data.
2. Multiple taps on Intents – Multiple selects on a single button or various buttons at once were attempted and no problems occurred.
3. Returning and Re-entering intent – Accessing the 1st round API and the previous menu over and over were done. Each time this was attempted the errors were caught and the user was returned to the previous screen.

The App has been regressively tested and no errors have been found anywhere. The app has been made available to 5 clients who have all tested the app without any errors occurring during their experiences.

Installation Manual

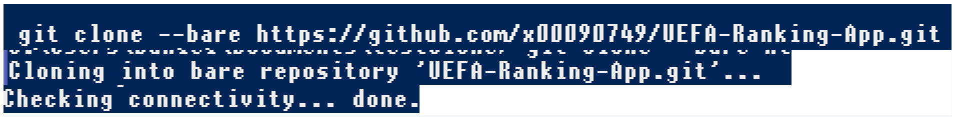
UEFA Ranking System Android Application Installation Manual

Download Project

The full project can be obtained through a GIT repository on GitHub. GitHub and its command Git Shell are required for installation.

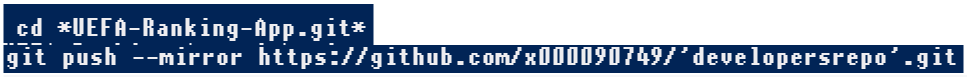
Steps to Clone Project:

1. To perform a fresh export of the data a bare clone of the repository is required for the specified local directory of the developers choosing:



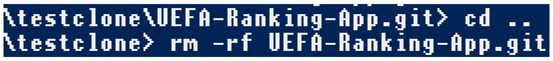
```
 git clone --bare https://github.com/x00090749/UEFA-Ranking-App.git
Cloning into bare repository 'UEFA-Ranking-App.git'...
Checking connectivity... done.
```

After successfully cloning the repository, the 'mirror' option can be selected to push the clone to the developer's new repository.



```
 cd *UEFA-Ranking-App.git*
git push --mirror https://github.com/x000090749/'developersrepo'.git
```

The temporary local repository can now be removed.



```
\testclone\UEFA-Ranking-App.git> cd ..
\testclone> rm -rf UEFA-Ranking-App.git
```

You have now successfully cloned the project, the next steps will focus on the different elements of development within the project.

Development

The project is effectively broken up in to two main components, Server side (Azure) and a Client Side (Android).

The main components of the server side consist of the following:

- ASP.NET Web API 2
- SQL Azure
- IIS Express
- JavaScript
- HTML
- Entity Framework 6
- ASP.NET MVC 5
- SSL
- OAuth 2 – Facebook, Google
- CSV Helper

The main components of the Client Side consist of the following:

- Android Studio IDE
- Java
- Android SDK (API Version 8+ supported)
- Apache HTTP Service Handler

All components listed above have been integrated within the projects and the majority is required for further development.

Database migrations through entity framework have been used to populate all initial data. The CSV files found in the Storage Git Folder are required. These can be implemented by currently inserting them in to the App_Data folder and called upon through the Migrations Configuration class.

Running Server and Client

Server Side

The Server side is fully implemented through Microsoft Visual Studio 2013 (Update4) using .Net framework 4.5.

Note: These are required to run the Project Solution for the UEFA Ranking System.

The server can then be started through IIS Express however many features such as Database and Authentication require an Azure Web Service Deployment of the Service.

The SQL Azure Database and its Initial Migrations must also be integrated when Deploying to an Azure Web Service. After doing this a fully working Server Side of the UEFA Ranking System will be running.

Client Side

The Client Side has been fully implemented through Android Studio 1.1.0 however Version 1.0 + should suffice for development.

The compile should target SDK Version 21 and a Gradle build is required to compile it. Internet permissions should **always** be declared in the Manifest file due to the Core data being called from the Azure Service (or IIS Express service for testing purposes only).

The App can now be successfully compiled and ran from through an Emulator or a connected device with USB debugging enabled.

For distribution, a Signed APK can also be generated at Build time and installed on any Android Device.

UEFA Ranking System

User Manual

Preface

About This Manual

The User Manual provided contains information about the functionality and services available with the UEFA Ranking System Mobile Android Application (which will be referred to as ‘the app’ throughout the document).

Audience

The Android App has been developed in conjunction with the UEFA Ranking System. The app is intended for those who want to use their Android Device to access elements of the Ranking System.

The app aims to tend to the needs of Football Clubs, Football Organizations (FAI, UEFA), and Statistical Reviewers. To a lesser extend it has also been designed and implemented for anyone with a remote interest in the Champions League and UEFA Rankings.

Overview of Application

The UEFA Ranking App is a complete replication of the UEFA Ranking System and contains various elements that are of use to anyone with an interest in the Ranking System itself.

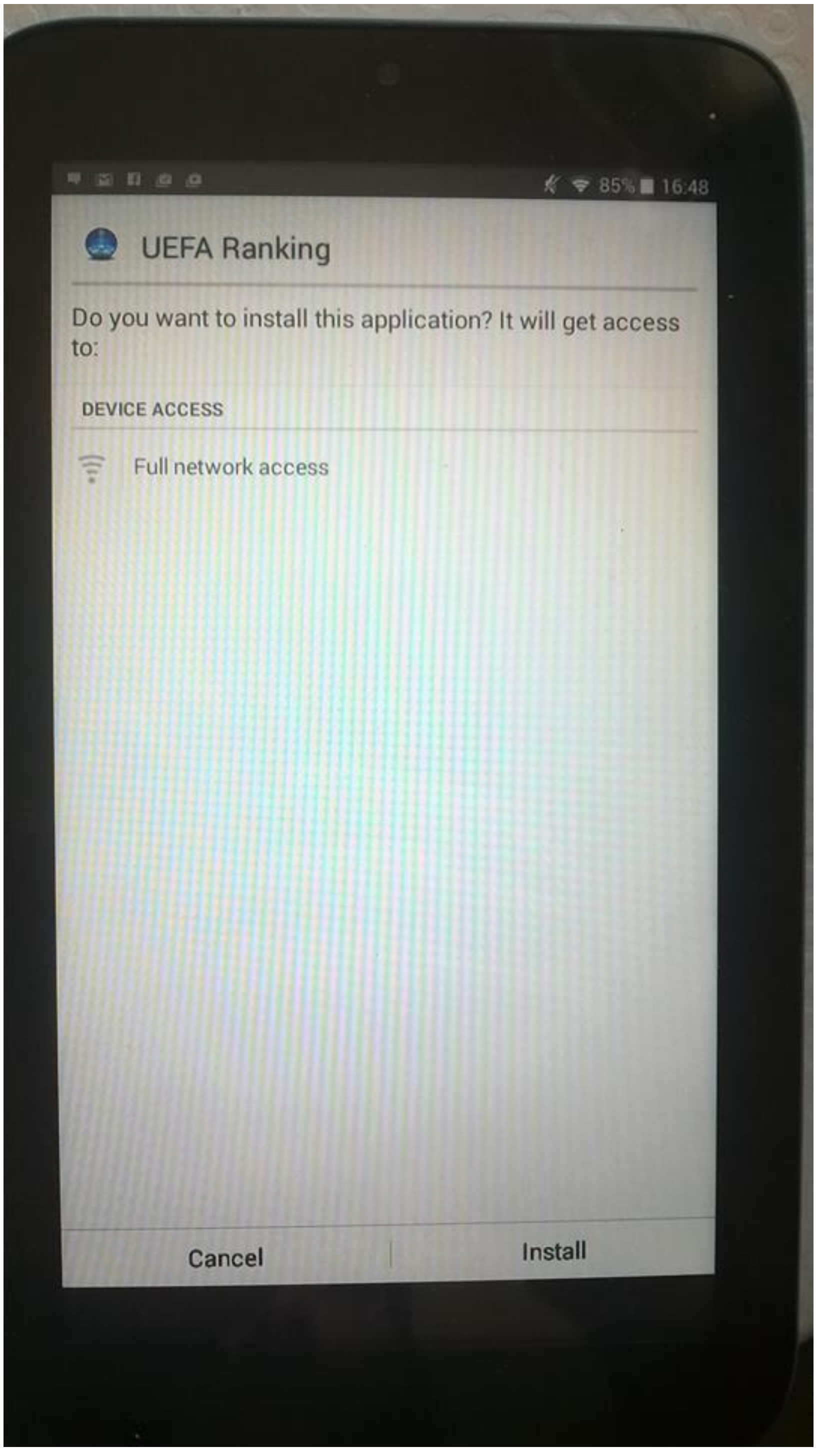
The App contains the following sections

- Main Menu
- Champions League Qualifying Rounds
- Champions League Group Stage
- UEFA Team Rankings
- UEFA Country Rankings
- Help Section
- About Section

Installing UEFA Ranking App

The application can be installed by opening the APK file for the version you have downloaded from the respective site.

Note: The App requires Internet Access to run all functionality.



UEFA Ranking

Do you want to install this application? It will get access to:

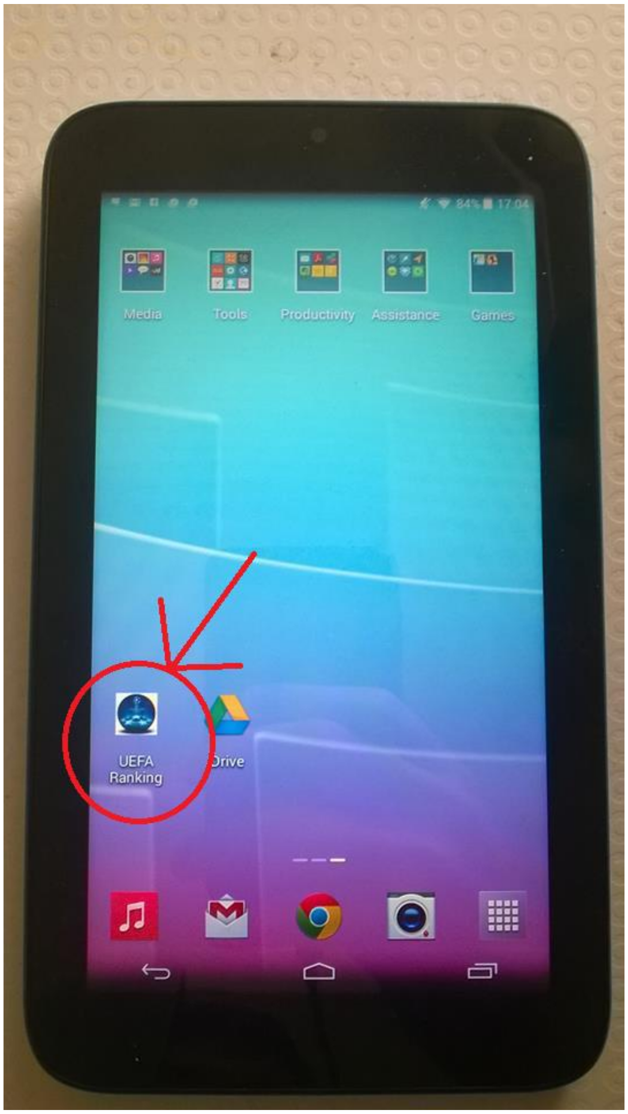
DEVICE ACCESS

Full network access

Opening the App for first use

To launch the app, tap the application icon on the Application screen of the Android Device you are using.

Please ensure you have WIFI or a Mobile Connection before opening the Application.

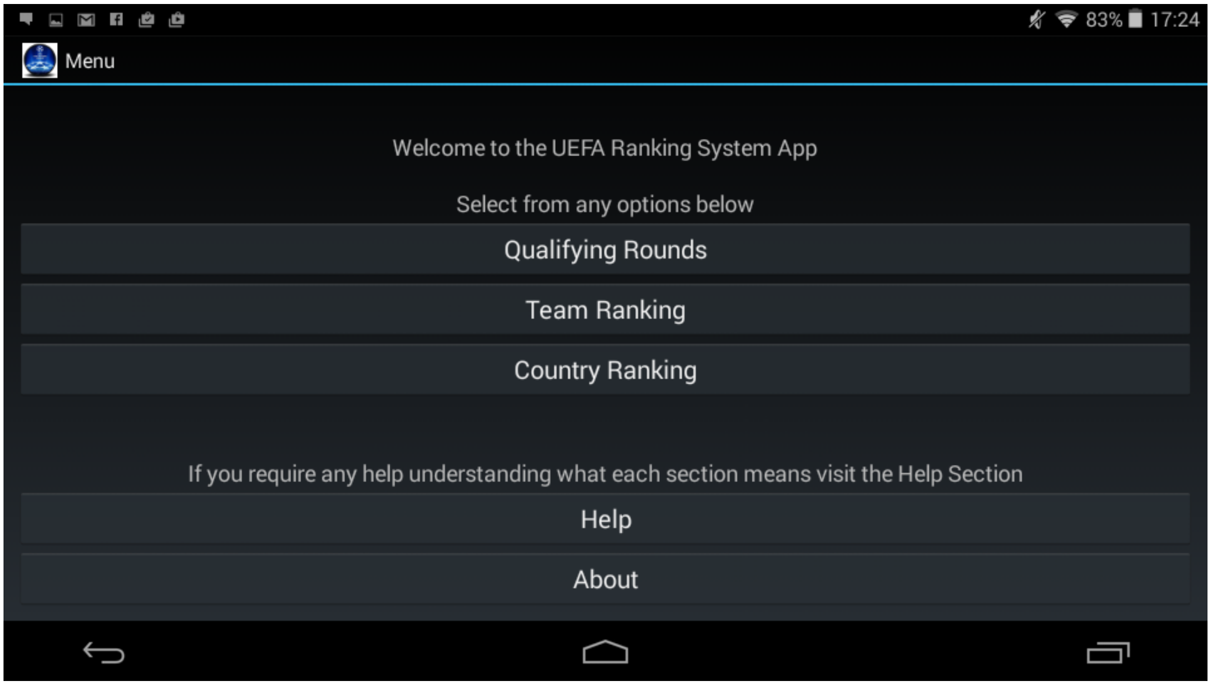


Main Menu

After opening the app and being greeted with a Title Screen you will be brought to the ‘Home’/Main Menu Screen.

The Menu contains Navigation options to all sections of the App.

Simply tap on any section to enter.



Welcome to the UEFA Ranking System App

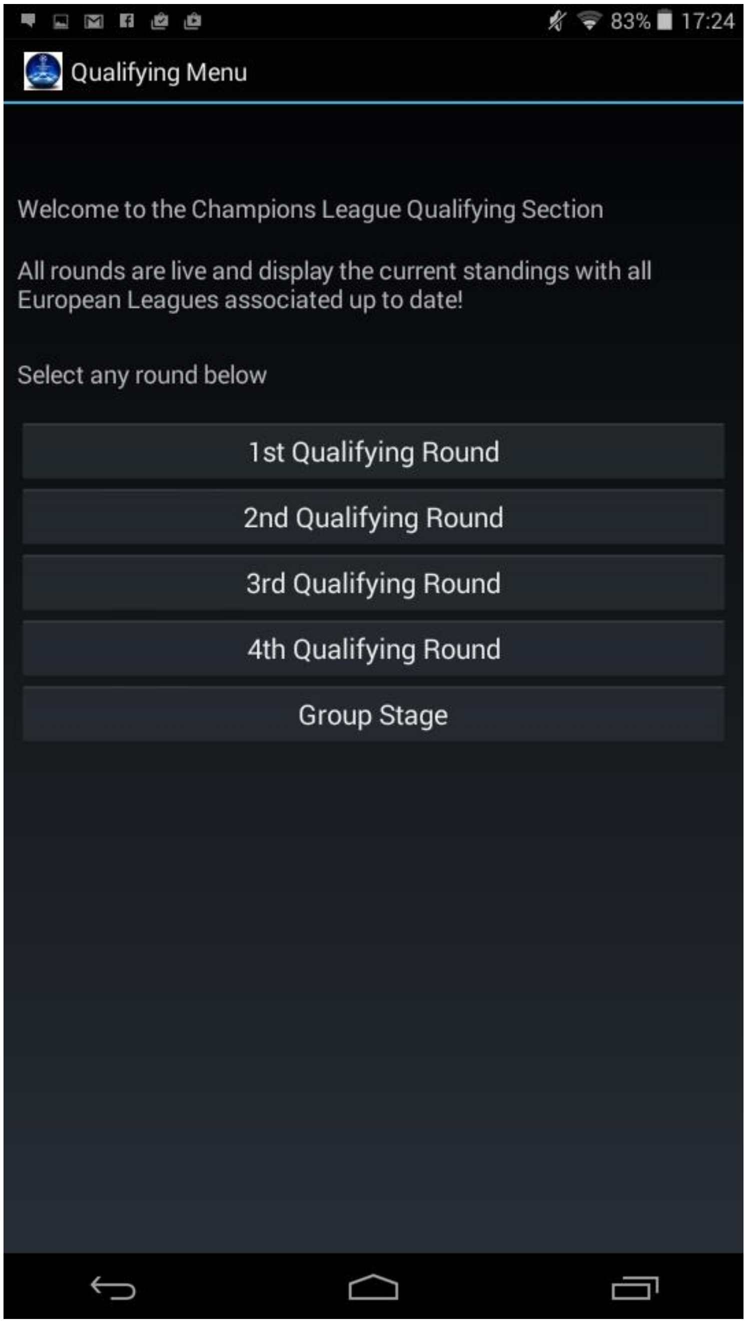
Select from any options below

If you require any help understanding what each section means visit the Help Section

Qualifying Rounds Menu

After selecting the Qualifying Rounds Option in the menu you will be greeted with various Qualifying Rounds and the Champions League Group Stage.

Each Round and the Group Stage can be accessed through this menu.



Welcome to the Champions League Qualifying Section

All rounds are live and display the current standings with all European Leagues associated up to date!

Select any round below

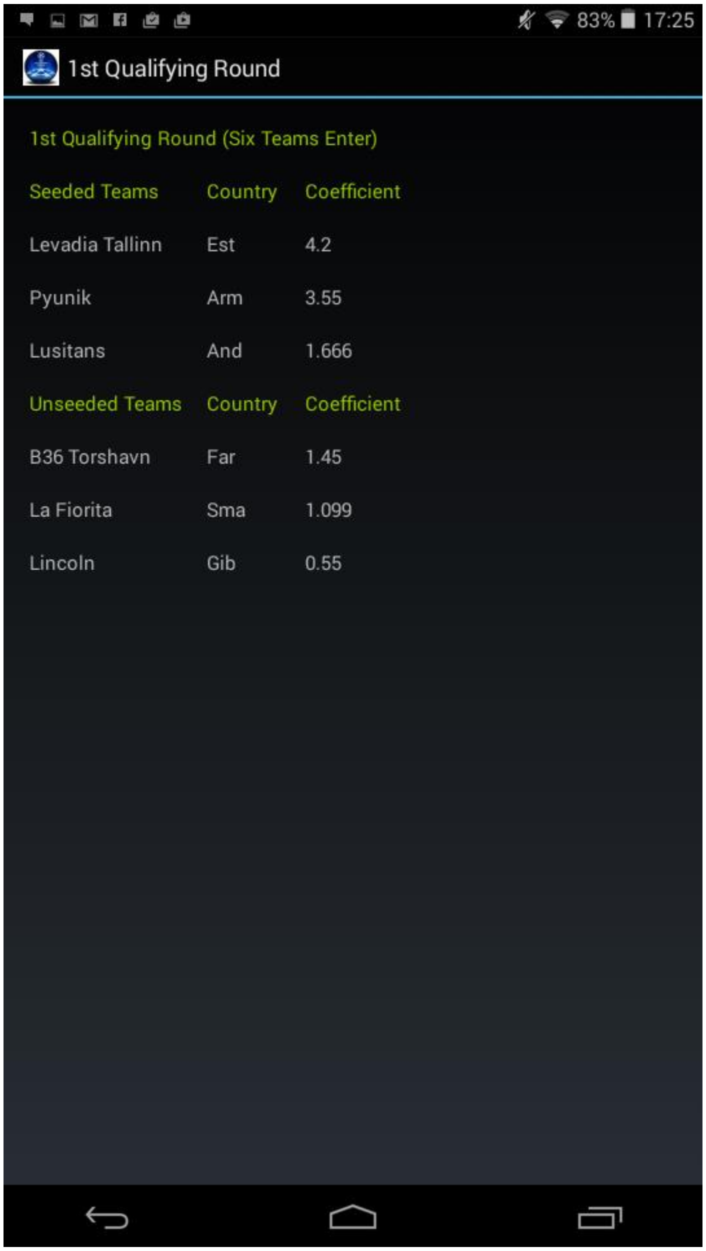
1st Qualifying Round

The 1st Qualifying Round provides the user with a live view of the teams entering the 1st round who are all currently top of their respective league.

The Round is then divided in to two, Seeded and UnSeeded.

In this instance 6 teams enter, 3 are seeded and 3 unseeded.

The Seedings are sorted based on each teams 'Team Ranking' (Can be cross checked in the Team Ranking Section). The 3 Teams entering with the highest Team Ranking are allocated a seeding spot.



1st Qualifying Round (Six Teams Enter)

Seeded Teams	Country	Coefficient
Levadia Tallinn	Est	4.2
Pyunik	Arm	3.55
Lusitans	And	1.666
Unseeded Teams	Country	Coefficient
B36 Torshavn	Far	1.45
La Fiorita	Sma	1.099
Lincoln	Gib	0.55

2nd Qualifying Round

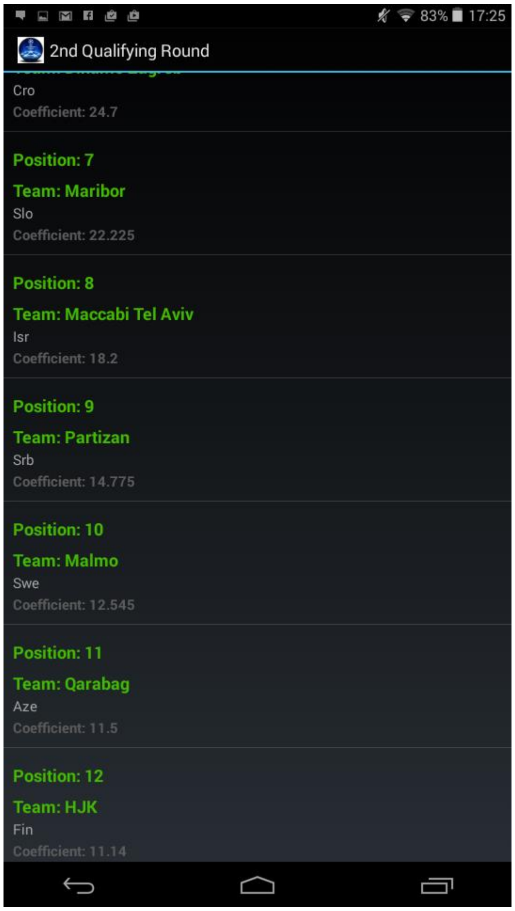
The 2nd Qualifying Round provides the user with a live view of the teams entering the 2nd round. Each team entering are all currently top of their respective league.

The Round is then divided in to two, Seeded and UnSeeded.

In this instance 34 teams enter the qualifying round (31 new entrants and 3 winners from the 1st round), the top 17 are seeded and bottom 17 unseeded.

Note: The top 3 ranked teams from the 1st Qualifying Round are included in the second qualifying round because regardless of the victor between teams the winner takes the ranked teams seeding rather than their own. Please see section 3.1 for further explanation.

The Seedings are sorted based on each teams 'Team Ranking' (Can be cross checked in the Team Ranking Section). The 17 Teams entering with the highest Team Ranking are allocated a seeding spot.



2nd Qualifying Round

Cro
Coefficient: 24.7

Position: 7
Team: Maribor
Slo
Coefficient: 22.225

Position: 8
Team: Maccabi Tel Aviv
Isr
Coefficient: 18.2

Position: 9
Team: Partizan
Srb
Coefficient: 14.775

Position: 10
Team: Malmo
Swe
Coefficient: 12.545

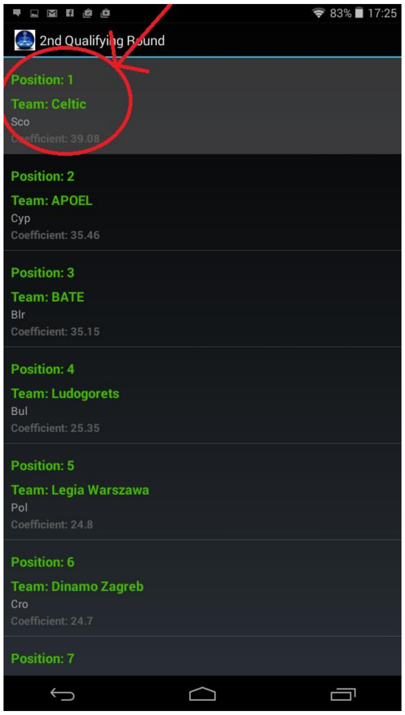
Position: 11
Team: Qarabag
Aze
Coefficient: 11.5

Position: 12
Team: HJK
Fin
Coefficient: 11.14

To view all teams in the round you can simply scroll up and down to view all 34 teams. The Position, Team name, Country and Coefficient is provided for each Team entering.

You can also select an individual team. After selecting a team you will be brought to a single screen containing specific information for the team you have selected.

Select Team of choosing:



83% 17:25

2nd Qualifying Round

Position: 1
Team: Celtic
Sco
Coefficient: 39.08

Position: 2
Team: APOEL
Cyp
Coefficient: 35.46

Position: 3
Team: BATE
Blr
Coefficient: 35.15

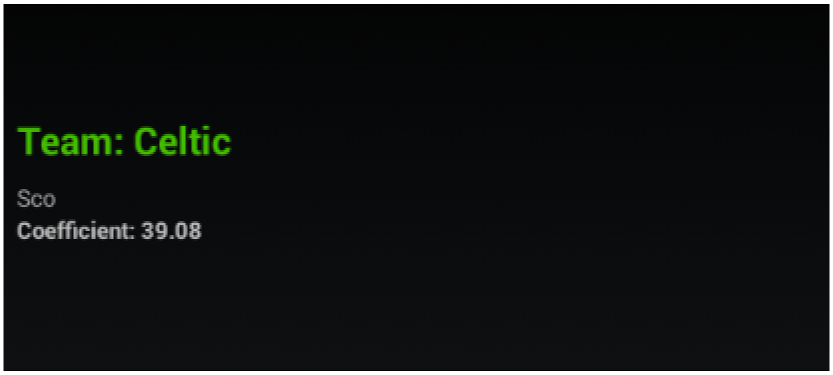
Position: 4
Team: Ludogorets
Bul
Coefficient: 25.35

Position: 5
Team: Legia Warszawa
Pol
Coefficient: 24.8

Position: 6
Team: Dinamo Zagreb
Cro
Coefficient: 24.7

Position: 7

Single Screen for selected Team



Team: Celtic

Sco
Coefficient: 39.08

3rd Qualifying Round Section

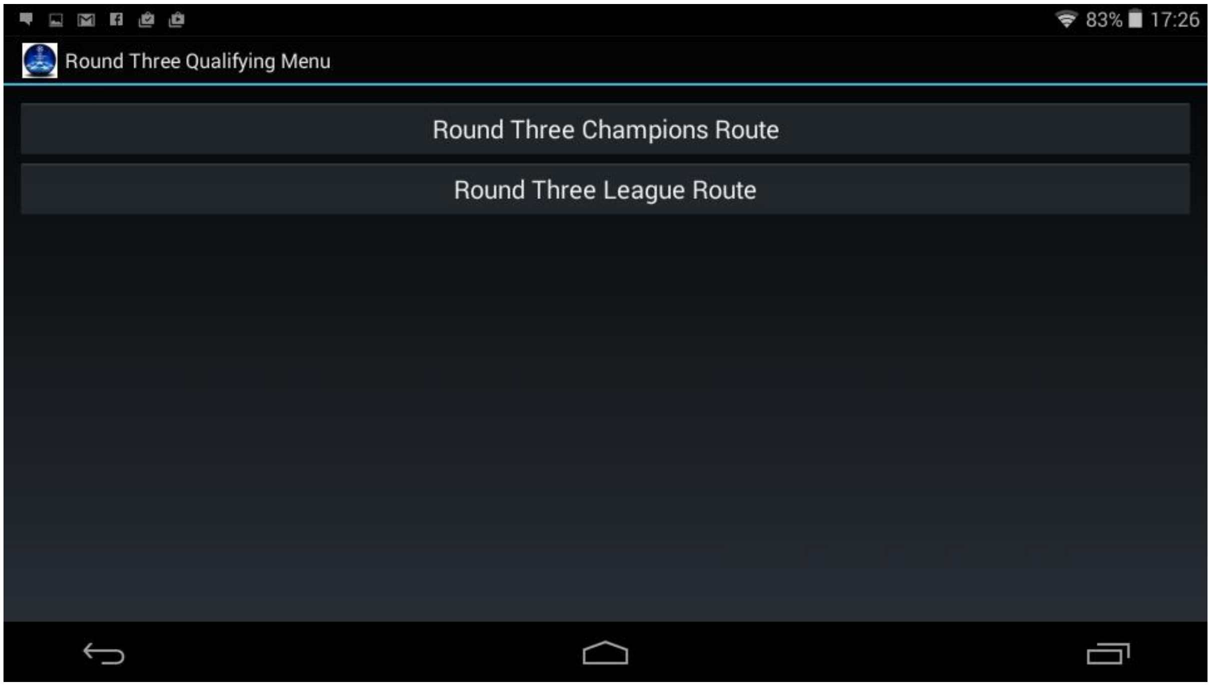
After selecting the 3rd Qualifying Round option from the Qualifying Rounds Menu you will be greeted with a 3rd Round Navigation Menu.

The Menu is comprised of 2 options to which you can select from.

The 3rd Qualifying Round is divided in to 2 sections:

- Round Three Champions Route

- Round Three League Route



3rd Qualifying Round – Champions Route

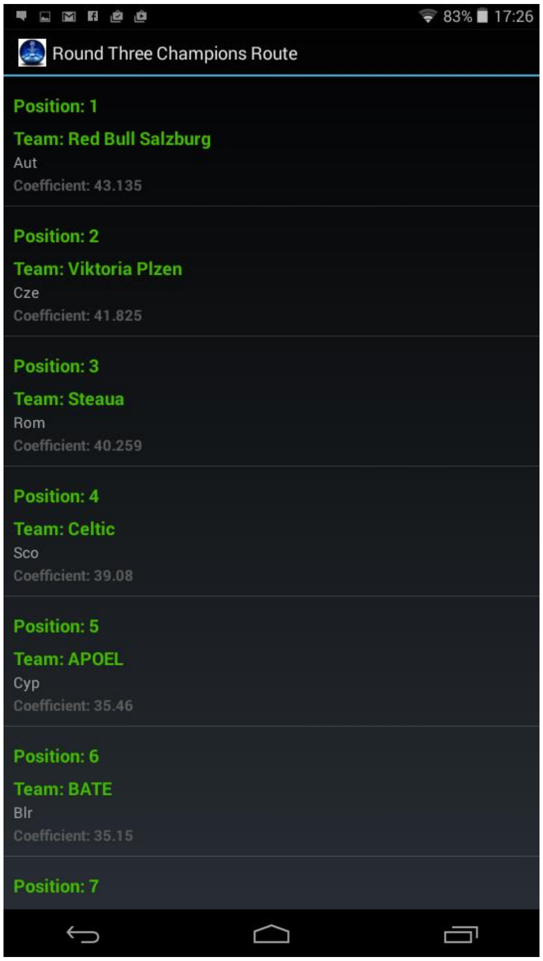
The Champions Route for the 3rd Qualifying Round provides the user with a live view of the teams entering the Champions Route. Each team entering is currently top of their respective league.

The Round is then divided in to two, Seeded and UnSeeded.

In this instance 20 teams enter the qualifying round (3 new entrants and 20 winners from the 2nd round), the top 10 are seeded and bottom 10 unseeded.

Note: The top ranked teams from the 2nd Qualifying Round are included in the third qualifying round champions route because regardless of the victor between teams the unseeded winner takes the seeded teams ranking rather than their own. Please see section 3.1 for further explanation.

The Seedings are sorted based on each teams 'Team Ranking' (Can be cross checked in the Team Ranking Section). The 10 Teams entering with the highest Team Ranking are allocated a seeding spot.



Round Three Champions Route

Position: 1
Team: Red Bull Salzburg
Aut
Coefficient: 43.135

Position: 2
Team: Viktoria Plzen
Cze
Coefficient: 41.825

Position: 3
Team: Steaua
Rom
Coefficient: 40.259

Position: 4
Team: Celtic
Sco
Coefficient: 39.08

Position: 5
Team: APOEL
Cyp
Coefficient: 35.46

Position: 6
Team: BATE
Blr
Coefficient: 35.15

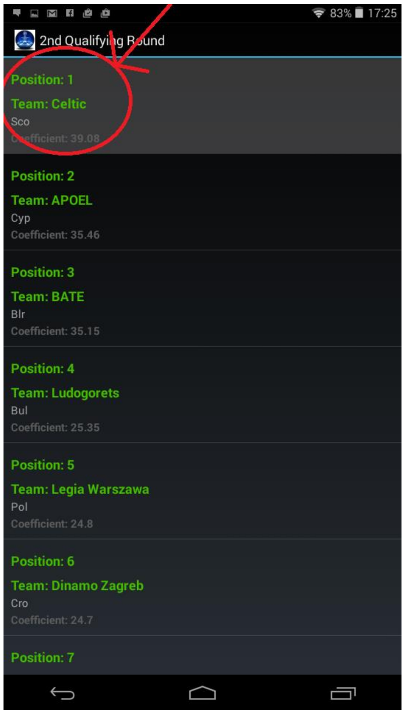
Position: 7

To view all teams in the round you can simply scroll up and down to view all 20 teams.

The Position, Team name, Country and Coefficient is provided for each Team entering.

You can also select an individual team. After selecting a team you will be brought to a single screen containing specific information for the team you have selected.

Select Team of choosing:



2nd Qualifying Round

Position: 1
Team: Celtic
Sco
Coefficient: 39.08

Position: 2
Team: APOEL
Cyp
Coefficient: 35.46

Position: 3
Team: BATE
Blr
Coefficient: 35.15

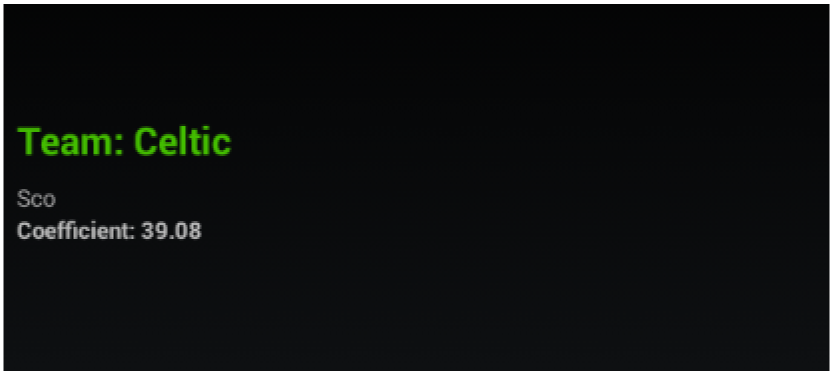
Position: 4
Team: Ludogorets
Bul
Coefficient: 25.35

Position: 5
Team: Legia Warszawa
Pol
Coefficient: 24.8

Position: 6
Team: Dinamo Zagreb
Cro
Coefficient: 24.7

Position: 7

Single Screen for selected Team



Team: Celtic
Sco
Coefficient: 39.08

3rd Qualifying Round – League Route

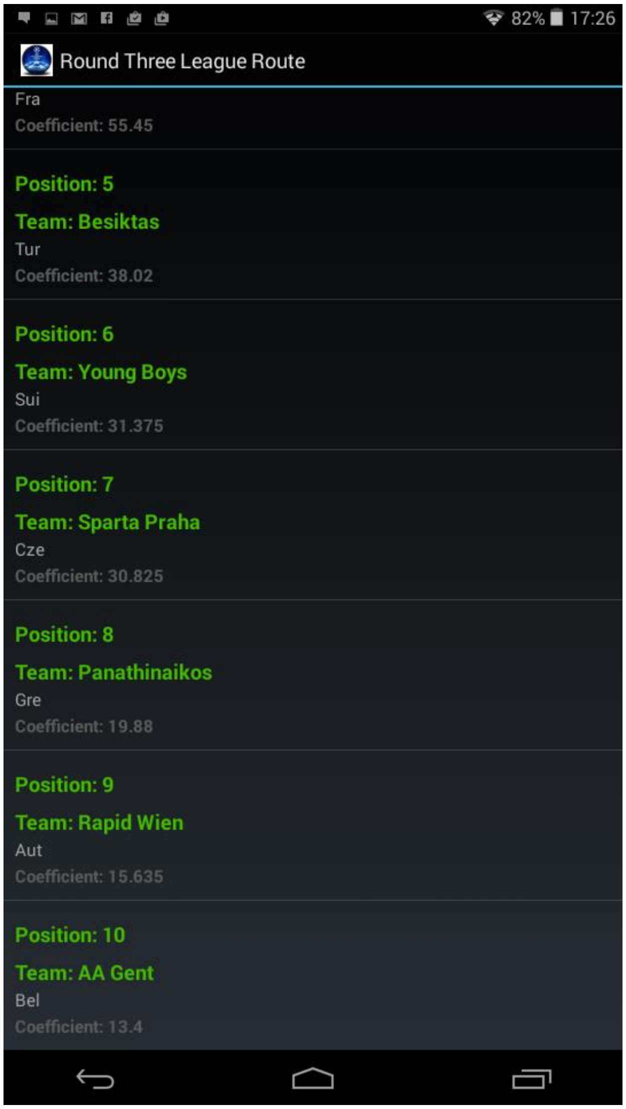
The League Route for the 3rd Qualifying Round provides the user with a live view of the teams entering the League Route. Each team entering has finished runner up in their respective league

The Round is then divided in to two, Seeded and UnSeeded.

In this instance 20 teams enter the qualifying round, the top 10 are seeded and bottom 10 unseeded.

You can browse through all teams in the round via scrolling.

The Seedings are sorted based on each teams 'Team Ranking' (Can be cross checked in the Team Ranking Section). The 10 Teams entering with the highest Team Ranking are allocated a seeding spot.



82% 17:26

Round Three League Route

Fra
Coefficient: 55.45

Position: 5
Team: Besiktas
Tur
Coefficient: 38.02

Position: 6
Team: Young Boys
Sui
Coefficient: 31.375

Position: 7
Team: Sparta Praha
Cze
Coefficient: 30.825

Position: 8
Team: Panathinaikos
Gre
Coefficient: 19.88

Position: 9
Team: Rapid Wien
Aut
Coefficient: 15.635

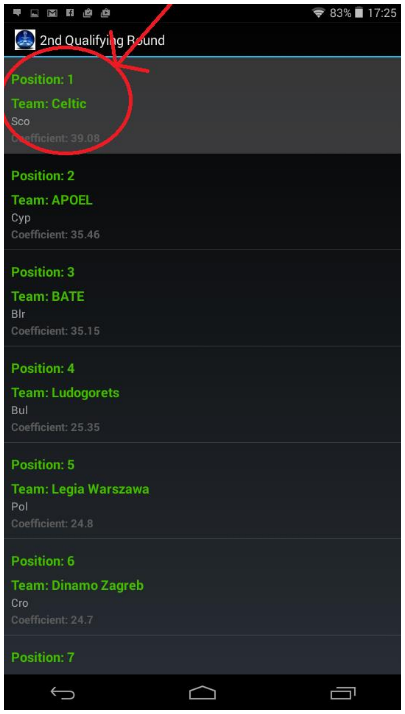
Position: 10
Team: AA Gent
Bel
Coefficient: 13.4

To view all teams in the round you can simply scroll up and down to view all 20 teams.

The Position, Team name, Country and Coefficient is provided for each Team entering.

You can also select an individual team. After selecting a team you will be brought to a single screen containing specific information for the team you have selected.

Select Team of choosing:



2nd Qualifying Round

Position: 1
Team: Celtic
Sco
Coefficient: 39.08

Position: 2
Team: APOEL
Cyp
Coefficient: 35.46

Position: 3
Team: BATE
Blr
Coefficient: 35.15

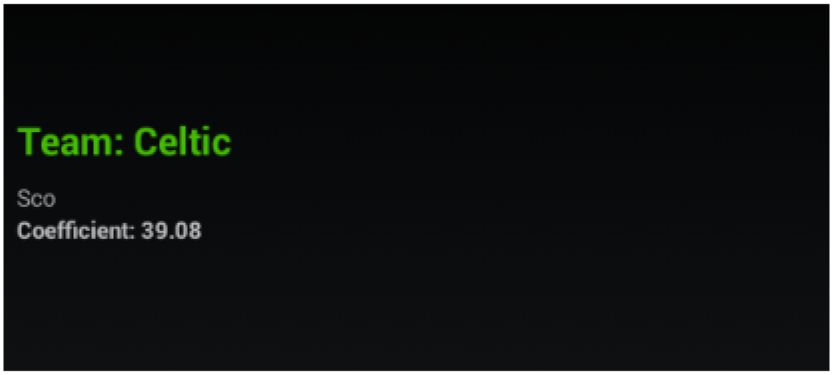
Position: 4
Team: Ludogorets
Bul
Coefficient: 25.35

Position: 5
Team: Legia Warszawa
Pol
Coefficient: 24.8

Position: 6
Team: Dinamo Zagreb
Cro
Coefficient: 24.7

Position: 7

Single Screen for selected Team



Team: Celtic
Sco
Coefficient: 39.08

4th Qualifying Round – League Route

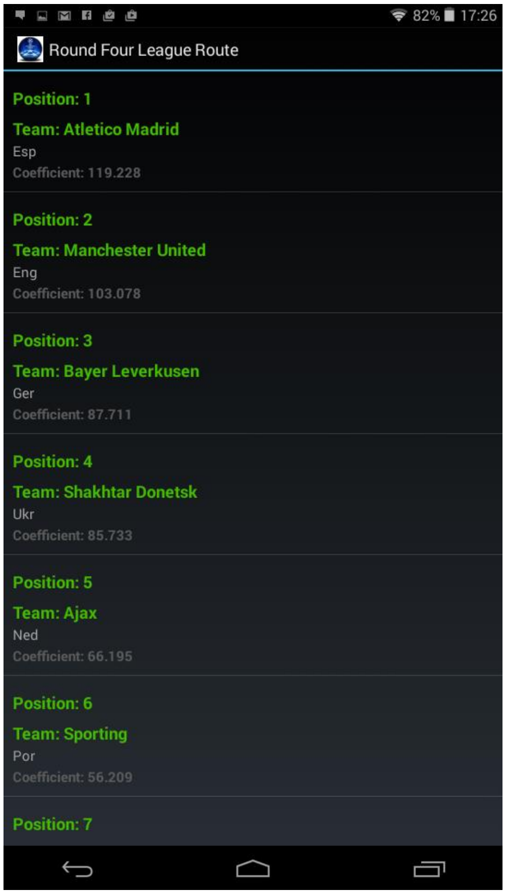
The League Route for the 4th Qualifying Round provides the user with a live view of the teams entering the Round. Each team entering has finished runner up in their respective league or 3rd and 4th in some cases (e.g. 4th spot in the English Premier League).

The Round is then divided in to two, Seeded and UnSeeded.

In this instance 20 teams enter the qualifying round (10 new entrants and 10 winners from the 3rd round), the top 10 are seeded and bottom 10 unseeded.

Note: The top ranked teams from the 3rd Qualifying Round League Route are included in this round however this is purely a prediction. After the 3rd round games finish the seedings are recalculated based on who advanced in that round.

The Seedings are sorted based on each teams ‘Team Ranking’ (Can be cross checked in the Team Ranking Section). The 10 Teams entering with the highest Team Ranking are allocated a seeding spot.



Round Four League Route

Position: 1
Team: Atletico Madrid
Esp
Coefficient: 119.228

Position: 2
Team: Manchester United
Eng
Coefficient: 103.078

Position: 3
Team: Bayer Leverkusen
Ger
Coefficient: 87.711

Position: 4
Team: Shakhtar Donetsk
Ukr
Coefficient: 85.733

Position: 5
Team: Ajax
Ned
Coefficient: 66.195

Position: 6
Team: Sporting
Por
Coefficient: 56.209

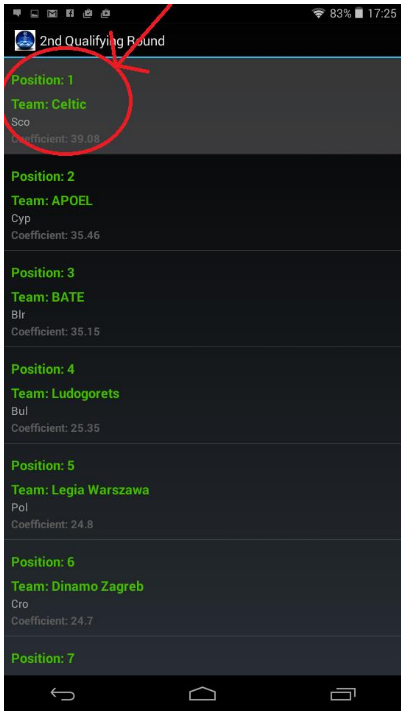
Position: 7

To view all teams in the round you can simply scroll up and down to view all 20 teams.

The Position, Team name, Country and Coefficient is provided for each Team entering.

You can also select an individual team. After selecting a team you will be brought to a single screen containing specific information for the team you have selected.

Select Team of choosing:



83% 17:25

2nd Qualifying Round

Position: 1
Team: Celtic
Sco
Coefficient: 39.08

Position: 2
Team: APOEL
Cyp
Coefficient: 35.46

Position: 3
Team: BATE
Blr
Coefficient: 35.15

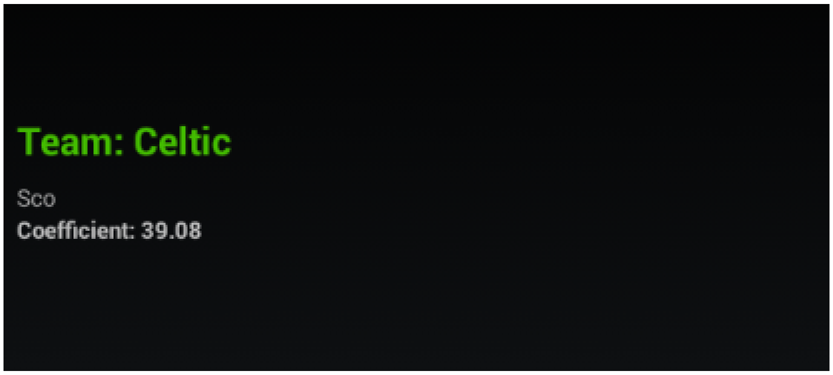
Position: 4
Team: Ludogorets
Bul
Coefficient: 25.35

Position: 5
Team: Legia Warszawa
Pol
Coefficient: 24.8

Position: 6
Team: Dinamo Zagreb
Cro
Coefficient: 24.7

Position: 7

Single Screen for selected Team



Team: Celtic

Sco
Coefficient: 39.08

Group Stages

The Group Stage section provides a live view of the teams that will enter the Champions League Group Stage. Each Team entering has qualified for the group stage either by winning, finishing runner up in their respective league, or qualifying through the Qualifying Round League Route and Champions Route.

The Group Stage is divided in to four sections:

- Pot 1 – 8 Teams
- Pot 2 – 8 Teams
- Pot 3 – 8 Teams
- Pot 4 – 8 Teams

In this instance 32 teams enter the Group Stage, they have entered through the following paths:

- 5 Winners from the Champions Route
- 5 Winners from the League Route
- 12 Champions of each league with their Country Ranked 1-12 (See Access List*)
- 6 Runners Up of each league with their Country Ranked 1-6*
- 3 third placed teams of each league with their Country Ranked 1-3*

The Seedings are sorted based on each teams 'Team Ranking' (Can be cross checked in the Team Ranking Section). The 32 teams entering are allocated to each pot based on their Team Ranking. For example, the 8 teams with the highest ranking of the 32 enter Pot One with the next 8 highest entering Pot 2 and so on.

To view all teams in the Group Stages you can simply scroll up and down to view all 32 teams.

The Position, Team name, Country and Coefficient is provided for each Team entering.

The following teams enter the following pots based on their position:

Teams with the following “Position” will be allocated to each respective Pot for the group stage Draw:

“Position” **1-8 – Pot One**

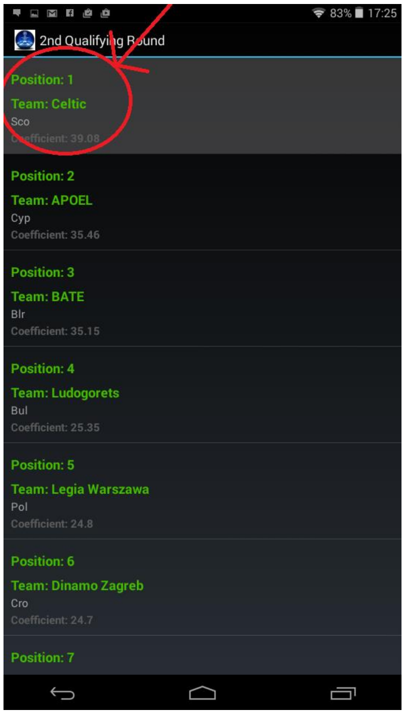
“Position” **9-16 – Pot Two**

“Position” **17-24 – Pot Three**

“Position” **25-32 – Pot Four**

You can also select an individual team. After selecting a team you will be brought to a single screen containing specific information for the team you have selected.

Select Team of choosing:



83% 17:25

2nd Qualifying Round

Position: 1

Team: Celtic

Sco

Coefficient: 39.08

Position: 2

Team: APOEL

Cyp

Coefficient: 35.46

Position: 3

Team: BATE

Blr

Coefficient: 35.15

Position: 4

Team: Ludogorets

Bul

Coefficient: 25.35

Position: 5

Team: Legia Warszawa

Pol

Coefficient: 24.8

Position: 6

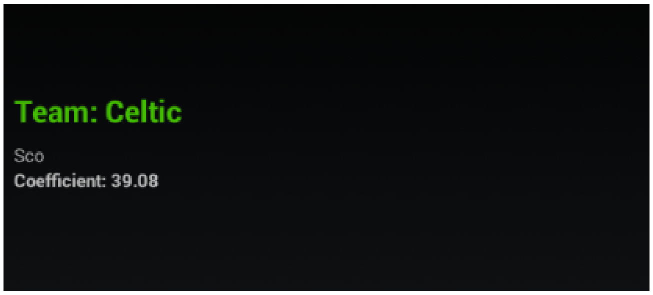
Team: Dinamo Zagreb

Cro

Coefficient: 24.7

Position: 7

Single Screen for selected Team



Team: Celtic

Sco

Coefficient: 39.08

Team Ranking Section

The Team Ranking section can be accessed by selecting the ‘Team Ranking’ option from the Main Menu Section.

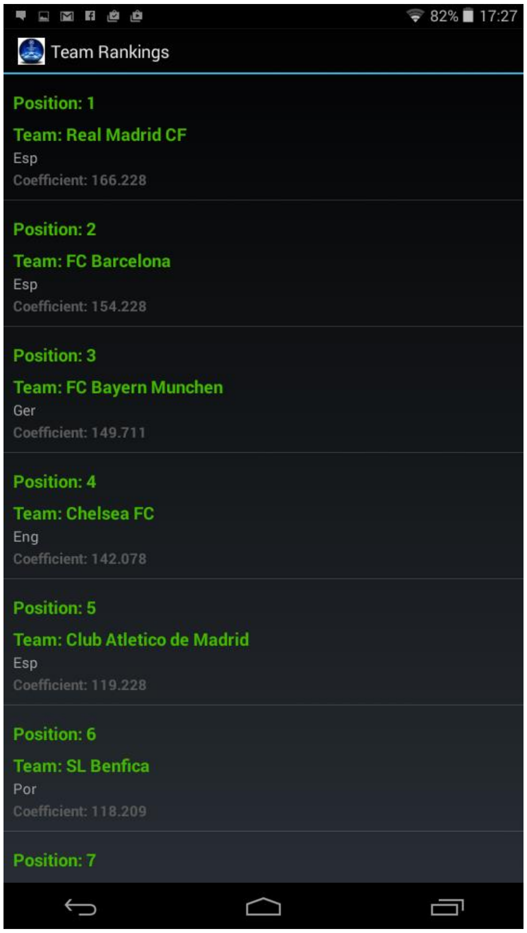
The Team Ranking section contains the full set of UEFA Rankings of 508 teams.

The Team Rankings are sorted by the Coefficient* of each team, in descending order of the team with the highest Coefficient.

The ‘base’ Ranking of Countries is also included within the Team Ranking section. The purpose of the ‘base’ ranking is for Teams of the Country in question that don’t have a Team Ranking. For qualification purposes a team with no ranking takes the Base Ranking of their country, this is why it is included.

Each Team is labeled with their Position, Team Name, Country and Coefficient*

*Team Ranking



Position: 1
Team: Real Madrid CF
Esp
Coefficient: 166.228

Position: 2
Team: FC Barcelona
Esp
Coefficient: 154.228

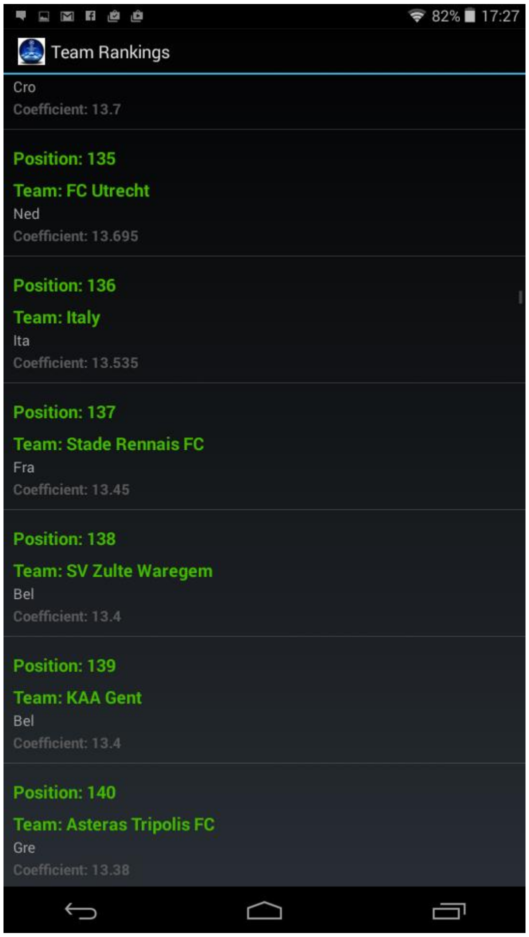
Position: 3
Team: FC Bayern Munchen
Ger
Coefficient: 149.711

Position: 4
Team: Chelsea FC
Eng
Coefficient: 142.078

Position: 5
Team: Club Atletico de Madrid
Esp
Coefficient: 119.228

Position: 6
Team: SL Benfica
Por
Coefficient: 118.209

Position: 7



Cro
Coefficient: 13.7

Position: 135
Team: FC Utrecht
Ned
Coefficient: 13.695

Position: 136
Team: Italy
Ita
Coefficient: 13.535

Position: 137
Team: Stade Rennais FC
Fra
Coefficient: 13.45

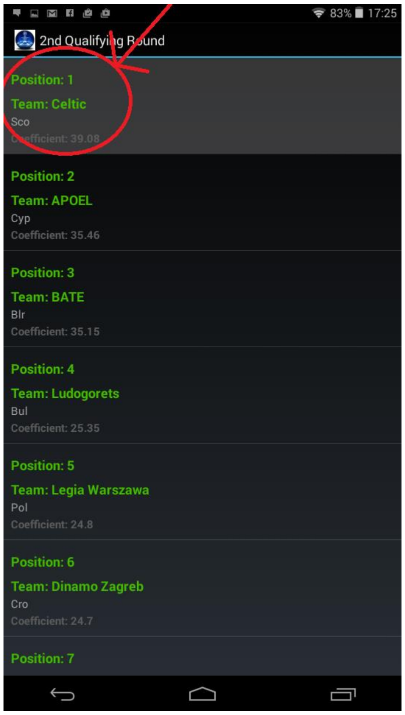
Position: 138
Team: SV Zulte Waregem
Bel
Coefficient: 13.4

Position: 139
Team: KAA Gent
Bel
Coefficient: 13.4

Position: 140
Team: Asteras Tripolis FC
Gre
Coefficient: 13.38

You can also select an individual team from the Full Team Ranking. After selecting a team you will be brought to a single screen containing specific information for the team you have selected.

Select Team of choosing:



2nd Qualifying Round

Position: 1
Team: Celtic
Sco
Coefficient: 39.08

Position: 2
Team: APOEL
Cyp
Coefficient: 35.46

Position: 3
Team: BATE
Blr
Coefficient: 35.15

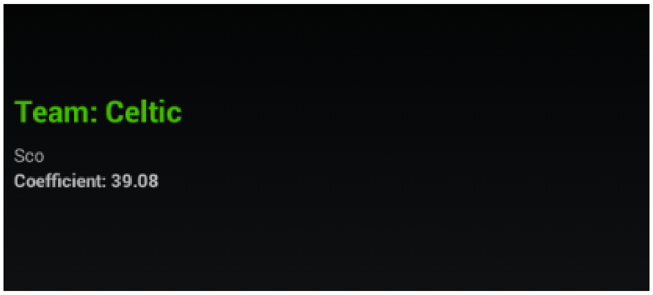
Position: 4
Team: Ludogorets
Bul
Coefficient: 25.35

Position: 5
Team: Legia Warszawa
Pol
Coefficient: 24.8

Position: 6
Team: Dinamo Zagreb
Cro
Coefficient: 24.7

Position: 7

Single Screen for selected Team



Team: Celtic
Sco
Coefficient: 39.08

Country Ranking Section

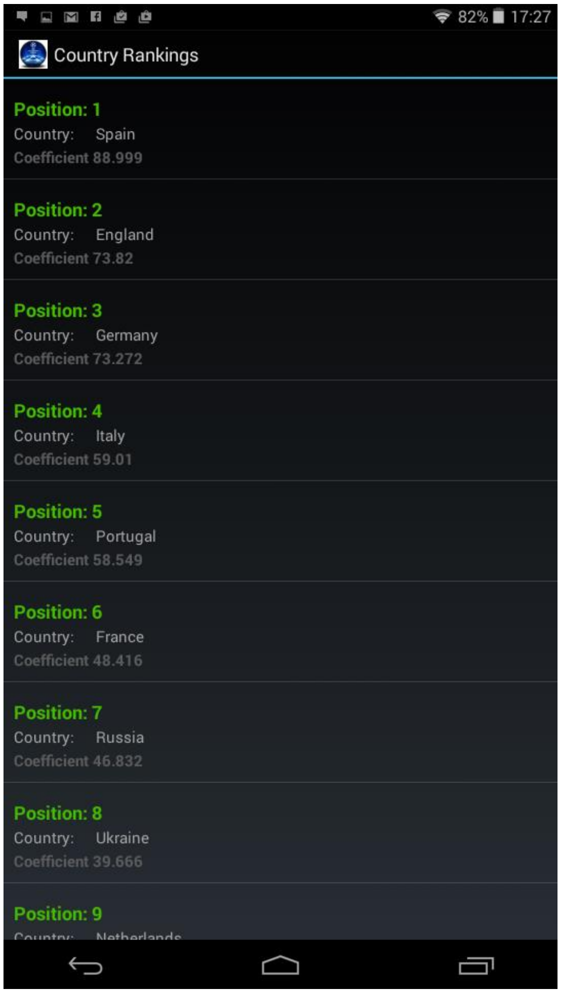
The Country Ranking section can be accessed by selecting the ‘Country Ranking’ option from the Main Menu Section.

The Country Ranking section contains all

The Country are sorted by the Coefficient* of each country, in descending order of the country with the highest Coefficient.

Country Team is labeled with their Position, Country Name and Coefficient*

*Country Ranking



Country Rankings

Position: 1
Country: Spain
Coefficient 88.999

Position: 2
Country: England
Coefficient 73.82

Position: 3
Country: Germany
Coefficient 73.272

Position: 4
Country: Italy
Coefficient 59.01

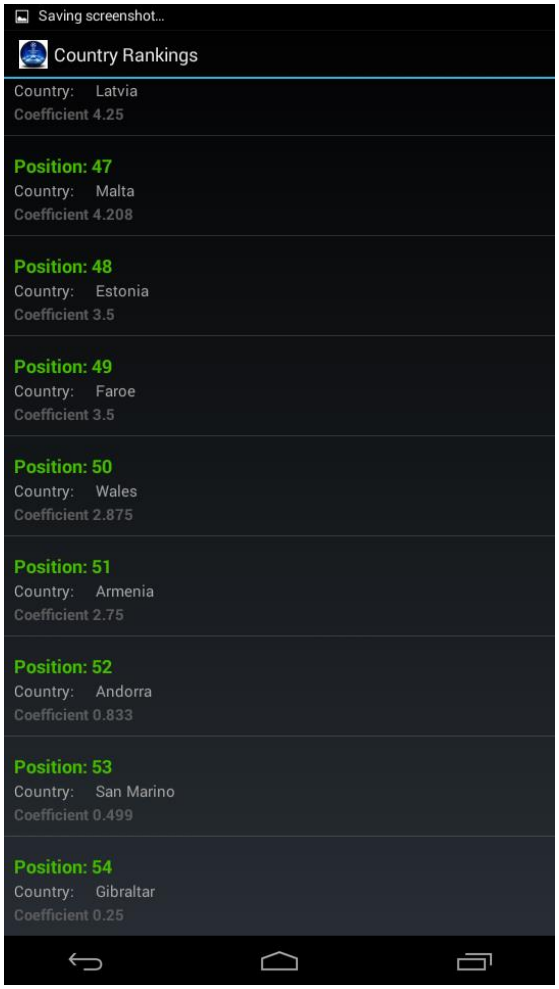
Position: 5
Country: Portugal
Coefficient 58.549

Position: 6
Country: France
Coefficient 48.416

Position: 7
Country: Russia
Coefficient 46.832

Position: 8
Country: Ukraine
Coefficient 39.666

Position: 9
Country: Netherlands



Country Rankings

Country: Latvia
Coefficient 4.25

Position: 47
Country: Malta
Coefficient 4.208

Position: 48
Country: Estonia
Coefficient 3.5

Position: 49
Country: Faroe
Coefficient 3.5

Position: 50
Country: Wales
Coefficient 2.875

Position: 51
Country: Armenia
Coefficient 2.75

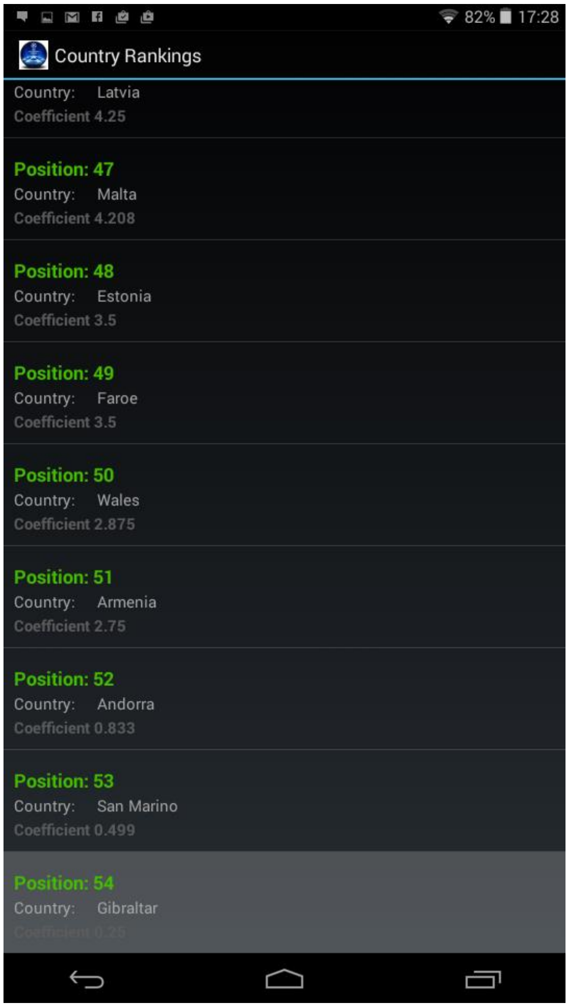
Position: 52
Country: Andorra
Coefficient 0.833

Position: 53
Country: San Marino
Coefficient 0.499

Position: 54
Country: Gibraltar
Coefficient 0.25

You can also select a country from the full Country Rankings. After selecting a country you will be brought to a single screen containing specific information for the country you have selected.

Select Country of choosing:



82% 17:28

Country Rankings

Country: Latvia
Coefficient 4.25

Position: 47
Country: Malta
Coefficient 4.208

Position: 48
Country: Estonia
Coefficient 3.5

Position: 49
Country: Faroe
Coefficient 3.5

Position: 50
Country: Wales
Coefficient 2.875

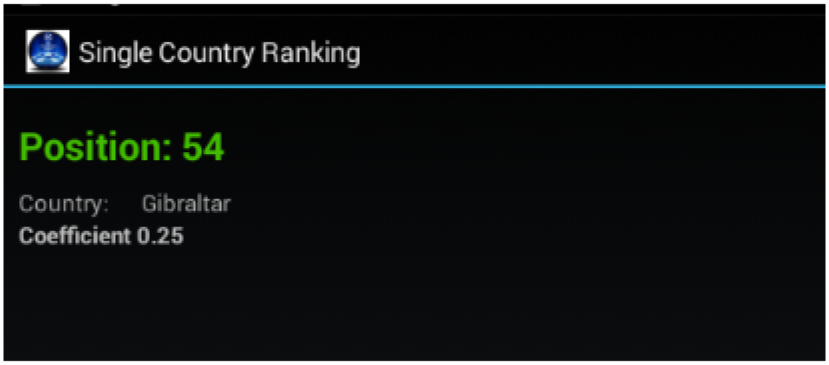
Position: 51
Country: Armenia
Coefficient 2.75

Position: 52
Country: Andorra
Coefficient 0.833

Position: 53
Country: San Marino
Coefficient 0.499

Position: 54
Country: Gibraltar
Coefficient 0.25

Single Screen for selected Team



Single Country Ranking

Position: 54

Country: Gibraltar
Coefficient 0.25

Additional Sections

About Section

The About section can be accessed by selecting the ‘About’ option from the Main Menu Section. To return to the menu simply select the standard Back Button

Help Section

The Help section can be accessed by selecting the ‘Help’ option from the Main Menu Section. To return to the menu simply select the standard Back Button

Troubleshooting

The following table lists some common issues and their resolutions:

Symptoms	Reasons	Resolutions
The Mobile Apps Core Functionality does not load or throw errors	Not connected to WIFI or Mobile Network. Although Connected you may not have Internet data incoming and outgoing at present.	Verify WIFI is enabled and connected. Also check if other internet based application are running correctly.
Application won’t open or freezes	System CPU and Ram may be used up	Check current CPU and RAM usage and close down other application if necessary and try again.

Post-Project Review

Project Overview

Looking at the project in its entirety it has been largely successful in what I first set out to achieve, a full functioning replication of the UEFA Ranking System for the Champions League.

There were many design, development and implementation changes throughout however the building blocks from the initial design were kept in the scope and implemented throughout the development of the Project. Some of the decisions that were made were major functionality changes however they were the correct solutions to problems faced during development that restricted originally planned functionality from being able to be developed.

When initially proposing ideas for my project selection I felt my idea was not only a good pick but it fell in to a comfort zone for me personally having known the full algorithm of all elements to the UEFA Ranking System. Personally knowing exactly what I needed to, what was required and how it should be implemented was quite a head start especially for the initial planning phase.

The UEFA Ranking system is quite a complex piece of architecture that has been designed by UEFA to calculate coefficients and rankings to create a format for the qualification and seedings for teams in the Champions League. For anyone who hasn’t seen the calculations before it is quite difficult to grasp and understand.

The whole purpose of this Project was to develop a User-Friendly app that would make it easier for users to understand the Ranking System. I believe the app I made is a perfectly suited Mobile Application to demonstrate all elements of the Ranking System. The additional functionality I have included such as live standings of each round and the group stage is unique to anything else that has been released.

It has to be stressed there is a specific target audience for the use of this app. In general, those concerned with the UEFA Ranking System would be Football Clubs that are or could potentially be involved in any Qualifying Round. For a team in this category they would be concerned with what round they are entering but most important if they will be seeded or unseeded.

From a Business and Marketing perspective the seeding status of a club is incredibly important considering the financial rewards for advancing through a round. Being seeded means you will more than likely face weak opposition in contrast to weak. For some teams half of their annual income is from the financial awards of advancing through a round. The app I have implemented gives teams a very good guidance to their seeding status therefore the unique functionality I have implemented is something that would be incredibly beneficial for Football Clubs to use.

Throughout the development lifecycle of the project I consistently corresponded with my supervisor in order to gain valuable feedback and guidance. I felt our meetings and regular communication through emails kept me on course to achieving what I set out to implement. There were times when I was under pressure to have certain deliverables completed (at person set deadlines) but I never fell behind throughout the development phase. I felt the feedback and guidance from my supervisor aided to the success of what now is a successful Project with a fully implemented UEFA Ranking System.

There are endless possibilities for extensions that can be integrated to the project. I feel further development could see the app becoming a major competitor on the Android Market among any UEFA or data statistic related application. The project itself is one that is unique and has never been done before. There is a massive gap for something like this and I believe this is a niche product chosen with entrepreneurial vision that has a lot of potential to be successful.

Possible Extensions

There is a wide range of possibilities to expanding my UEFA Ranking System Project. The project has been implemented in such a way there are several opportunities for extensions.

Implementation of the Europa League

The UEFA Ranking System project currently implements the Champions League Qualifying Rounds and Group stage. There is without questions this is a major implementation in itself however there is definitely an opening for the Europa League to be integrated in to the project as an extension also. The Europa League carries different permutations for teams entering and on a much larger scale however the foundation to entering rounds and a group stage is still the same.

At present the project has been designed for those concerned with the Champions League only but there is a massive gap there for those that would be interested in the Europa League also and a strong case is there for the Europa League to be implemented further down the line. It would be a similar process to that of the Champions League with Qualifying Rounds, Group Stages and several permutations that affects the final view of each round.

For a Complete design of the UEFA Ranking system to be done the inclusion of the Europa League would be a definite requirement and increase the target audience size largely on a whole scale.

Live Updates through Football API Service

Initially it was proposed the teams entering each round for some leagues would be updated through a Football API Service. For Example, the Service could provide live updates of standings in the English Premier League and should a change occur in teams occupying the Champions League spots, this would be replicated in the database and changed on the API.

Unfortunately the original plan for this to be implemented had to be scrapped due to developer licensing issues with the provider. The Football API used included League standings for 13 countries that were updated through a times Azure Worker Role.

This is definitely an extension that should be taken in to serious consideration. There will be whole scale changes required migrating to accommodate live changes through an API however the Administrator Web Client and initial database migrations would still be included. The original layout to accommodate an external API is still integrated within all the models.

Example:

Sports Radar was the Soccer API Service Company originally used. An API was provided with League Standings for 13 leagues.

How it would be implemented:

The League Standings would be accessed through an API key and taken for use through a Visual C# Console Application using .NET HTTP and System Threading async run tasks. A comparison with the current SQL Azure Database will be performed and any changes with a team currently top of a specific League would be checked. Should a different team be found on the Sports Radar API, an update would be done on the database. After this is done there will be full replication of this change on the Administrator MVC end, the Server API and the Android Client itself.

For the Project to be seen as completely ‘Live’ it would be essential to use a Football API service to update the App instantly or on timed intervals. Currently the Administrator Web Service has been implemented to suffice such changes.

A sample for the league standings can be found here:
http://developer.sportradar.us/files/soccer_v2_standings.xml

Note: The reference with the current database would be made through the ‘country_code’ variable implemented on all Model Classes (Qualifying Rounds, Group Stages, Team Ranking and Country Ranking)

Weekly Round Seedings Comparison

Another possible extension to the project would be a weekly comparison of Seeding in Each Qualifying Rounds and the Group Stages. With the project developed at the end of the UEFA European Leagues' cycle (leagues approaching their end) a comparison would have been limited however with the Majority of Leagues already underway for the 2016/2017 Champions League draw and the rest beginning in August a comparison could be implemented.

A comparison would give a much clearer view early on to what teams are on course to finish top of their respective leagues and ultimately find themselves in their respective qualifying round when all leagues finish. There are multiples scopes to the comparison. An algorithm that checks how many weeks a team is top of their league could be taken in to account and on a more advanced case the points difference between a tea currently first compared to the team second could be taken and then a predictor formed for what teams look set to win their respective leagues.

This would work well in conjunction with the Live API update extension and if that proposed extension is carried out the Comparison should also be implemented for additional functionality.

Mobile Device Support – IPhone, Windows

Currently the Project has been designed and developed specifically to function and run on Android Devices. With the target audience in mind having limited the mobile client to Android there is a certain percentage of potential users of the application not even being hit due to users owning alternative Mobile Operating Systems such as Windows 8.1 and IOS.

Taking the design of the IPhone and Windows App aside the Azure Service is currently implemented to fully support alternative Mobile Operating Systems using the service. This is definitely a possible extension that will be looked at and potentially fully integrated in the future.

User Web Service & Windows Platform Support

Currently the project has a Web Service for an Administrator to alter various elements of the UEFA Ranking System. There is possible extension for the Service to be made available for all users, admin or guest. The main obstacle to tackle would be security restrictions however the Client view on the Android Device could be replicated on the Web Service so the design is more user friendly compared to the current Administrator Views.

WPF Windows Client Application is another potential extension that again will provide all functionality present on the android application. By extending the platform choices across both Windows platform and Mobile devices the majority of the Target audience should have an eligible device to use the UEFA Ranking System Application.

Review of Research, Analysis and Design Phases

The Research, Analysis and Design Phases were all carried out to quite an extent ensuring the requirements for implementing the UEFA Ranking System were realized. There is quite a large complexity to the Ranking system therefore it was essential a lot of work was carried out in these areas before carrying out development.

The research phase was conducted to explore the UEFA Ranking System, the similar Applications already released on mobile, how the algorithm worked and how it could potentially be implemented in development. This was carried out in depth die to the complexity of the system proposed.

The various Mobile platforms had been researched in depth and a decision was made to choose Android as the preferred mobile client due to the percentage of users using the device on the mobile market and the future growth of Android in general.

It was understood after the research the Project would need to be comprised of a web service, database and the client itself. The implementation of all 3 running in conjunction with each other was taken in to heavy consideration and the extensive research suggested the best approach for the eventual development. Elements of Android such as memory management, service capabilities and allocation of resources were researched also to ensure compatibility was ever present for all functionality that would be integrated to the application.

Although Security was a large element for the Design phase some initial research was conducted to explore the various security breaches that can occur for such a System and what Security could be integrated across the whole system to ensure it is safe and anti-malware proof.

Multiple Platforms, Technologies and Libraries such as Android, Web API 2, SQLite, SQL Azure, Azure Mobile Services, C-Sharp, and Windows Phone were all researched to explore multiple options for how the project would be developed. It was important to explore as many proposed elements to ensure the best Technologies, platforms and libraries were chosen.

Along with the research conducted a large portion of analysis was done partaken to understand the requirements of employing the UEFA Ranking Algorithm correctly. The core analysis was looking at the algorithm. It was found the ranking system was quite complex due to the several permutations involved which had made it quite difficult to understand without any assistance from someone who understood the algorithm inside out. In order to implement the Ranking correctly analysis on the permutations, were taken in to huge consideration.

The delicate issue that was found was the Ranking System had to be calculate in a specific manner, any errors or personal changes to the algorithm meant the final ranking, qualifying rounds and group stages would not be a true reflection of the correct results. To ensure the algorithm was to be correctly implemented analysis was done to explore all aspects of the Ranking System.

In depth analysis was partaken on the following elements of the UEFA Ranking System:

1. Allocation of a team to a qualifying rounds
2. Calculation of the Country Rankings
3. Calculation of the Team Rankings
4. Exceptions to Above calculations
5. Affect Country rankings has on what round each team enters

These were initial permutations explored and throughout the project additional analysis research was conducted to explain and understand the algorithm during implementation.

The Design Phase tied in with the Research and Analysis conducted. It was essential the layout of the Mobile App and the Admin Web client tied in with the design of the model and API/MVC implementation.

Originally it was decided the Application would consist of a Server side (Web API 2.2, SQL Azure) and Client side (Android) design with regard to the architecture of the System with C-Sharp and Java being the coding languages used.

The Android Application was designed to originally include the following sections:

- Team Ranking
- Country Ranking
- Qualifying Round Standings

Throughout the development phase the sections were kept as originally designed which was vital to ensure the core elements of the UEFA Ranking System were available for the User.

One key factor to the original design was taking Security in to consideration throughout. A full set of research and analysis was taken exploring the various security elements that could be integrated to the project. It was found that the actual design of the system was important to allow the security functions to be employed correctly. A lot of analysis was done for Android security however this became largely redundant due to the CRUD methods being made available only on the server and Admin Client end.

The Android application was always set out to be designed to be user friendly, fast and provide the major functionality within 3 clicks/selects. After finishing development this design has been implemented correctly and having various users test the application the general feedback suggested the application was user friendly.

It is clear there were various design changes made after the original design plan. Further analysis was also conducted which can be found in the Additional Information section below.

Additional Research Conducted

Due to the Football API service becoming unavailable due to licensing issues, other areas were explored to solve the issue of altering the Database to replicate changes for the Qualification section of the project. Various areas were explored but the chosen area was ASP.NET MVC 5. Through MVC 5 a Web service could be implemented to provide an Administrative Web Client where the full database could be viewed, deleted, additional content added (teams) but ultimately teams in a round could be edited.

After extensive research in to MVC 5 areas required for a successful administrator client were looked in to. It was found that security elements such as HTTPS (SSL), OAUTH 2 and many other features could be integrated to the MVC 5 Service to ensure a 'lock down' of the administrative side was possible.

After finishing the additional research MVC 5 was chosen and HTTPS (SSL) was implemented with OAUTH for Facebook and Google included also. A register/login User Role was also included for normal logins with a secure lock down on permissions that only allowed Administrators to edit content.

The additional research was essential to solve the problem of losing the Football API Service and maintain the original planned functionality of the Project and the app itself.

Additional Analysis

Following on from the initial analysis partaken exploring the UEFA Ranking System Algorithm for all its elements and the design for how it should be implemented, a lot of additional analysis was partaken give explanations and understanding to the algorithm and how it should be employed in the project.

Access List

UEFA's access list was analyzed, this was used to initially understand and employ the algorithm. The access list contained the Country Ranking however also showed for each position how many teams a country in that position could have in the Champions League and what round they would enter.

Rank / Association		**Champions League**						
		Group	**NCQ4**	**NCQ3**	**CHQ4**	**CHQ3**	**Q2**	**Q1**
		CL-TH (5 x NCQ4) (5 x CHQ4)	EL-TH (5 x NCQ3)		(10 x CHQ3)	(17 x Q2)	(4 x Q1)	
1	Esp	CH,N2,N3	N4					
2	Eng	CH,N2,N3	N4					
3	Ger	CH,N2,N3	N4					
4	Ita	CH,N2	N3					
5	Por	CH,N2	N3					
6	Fra	CH,N2		N3				
7	Rus	CH		N2				
8	Ned	CH		N2				
9	Ukr	CH		N2				
10	Bel	CH		N2				
11	Tur	CH		N2				
12	Gre	CH		N2				
13	Sui			N2		CH		
14	Aut			N2		CH		

15	Cze			N2		CH		
16	Rom						CH	
17	Isr						CH	
18	Cyp						CH	
19	Den						CH	
20	Cro						CH	
21	Pol						CH	
22	Bls						CH	
23	Sco						CH	
24	Swe						CH	
25	Bul						CH	
26	Nor						CH	
27	Srb						CH	
28	Hun						CH	
29	Slo						CH	
30	Svk						CH	
31	Mol						CH	
32	Azb						CH	
33	Geo						CH	
34	Kaz						CH	
35	Bos						CH	
36	Fin						CH	
37	Isl						CH	
38	Lat						CH	
39	Mon						CH	
40	Alb						CH	
41	Lit						CH	

42	Mac						CH	
43	Irl						CH	
44	Lux						CH	
45	Mlt						CH	
46	Lie							
47	Nir							CH
48	Wal							CH
49	Arm							CH
50	Est							CH
51	Far							CH
52	Sma							CH
53	And							CH
54	Gib							CH
79 x CL		**22** **(32)**	**6** **(10)**	**10** **(10)**	**-** **(10)**	**3** **(20)**	**30** **(34)**	**8** **(8)**

Keys:

TH = title-holder

CH = domestic champion

N2 = domestic league 2nd-placed club

N3 = domestic league 3rd-placed club

N4 = domestic league 4th-placed club

Q = qualifying round

CHQ = qualifying round for champions

NCQ = qualifying round for non-champions

(Kassies, 2015)

By cross checking the Access List the teams that are eligible to enter each round will be determined. Taking Countries ranked 47-54, they are marked to have one team enter the 1st qualifying round. The teams representing each country in this round are the team currently top of their respective league.

		Champions League						
		Group	**NCQ4**	**NCQ3**	**CHQ4**	**CHQ3**	**Q2**	**Q1**
Rank / Association		CL-TH (5 x NCQ4) (5 x CHQ4)	EL-TH (5 x NCQ3)		(10 x CHQ3)	(17 x Q2)	(4 x Q1)	
47	Nir							CH
48	Wal							CH
49	Arm							CH
50	Est							CH
51	Far							CH
52	Sma							CH
53	And							CH
54	Gib							CH
79 x CL		**22** **(32)**	**6** **(10)**	**10** **(10)**	**-** **(10)**	**3** **(20)**	**30** **(34)**	**8** **(8)**

Additional Permutations

There are some exceptions that affect the overall algorithm.

1. Liechtenstein does not enter the Champions League. They are currently ranked 46th and an exception will need to be integrated to ignore this country when calculating.

2. The Winner of the Champions League 2014/2015 is entitled to automatic entry to the group stages. This will cause an influence to every team that enters the group stages. For example should the team in question also qualify for the group stages through their league position, all teams will move up a position in the qualifying rounds.
   - I.e. Highest ranked country entering round 1 moves to round 2.
   - Highest ranked country entering round 2 moves to round 3.

The above permutations were realized after the original analysis and were taken in to consideration before the implementation of the algorithm. There are various permutations such as the above throughout the UEFA Ranking system and it was important to employ all these as ignoring any will affect the round results and the whole system will be incorrect and not be a replication of the current correct Ranking.

Initial Algorithm

After analyzing the access list the initial algorithm below was documented for the first iteration of the Project.

The beginning of the algorithm checks the access list.

a. The access list contains a ranking list (1-54). Each rank coincides with the country ranking. The rank determines how many teams enter the Champions League from that country based on their current rank. The most essential part is the round to enter which is determined by the Access List.

b. After checking the rank in the access list the rank number maps to the country ranking list. For example, rank 54 in the access list checks with the country ranked 54th. The country now contains its name, number of teams to enter, what round the team will enter and some additional details though not necessary for now.

c. The Country then performs a check on its own league and who is first. The First place team takes the Champions league spot. The top team is found on the database and has been determined through a live Football Data API that updates the current positions of teams in each league.

d. With the correct team now they currently have information on the Round they enter but now need their own individual ranking to determine whether they will be Seeded or UnSeeded. This is performed **after** all teams are allocated each round. A sorting method will arrange them based on their ranking.

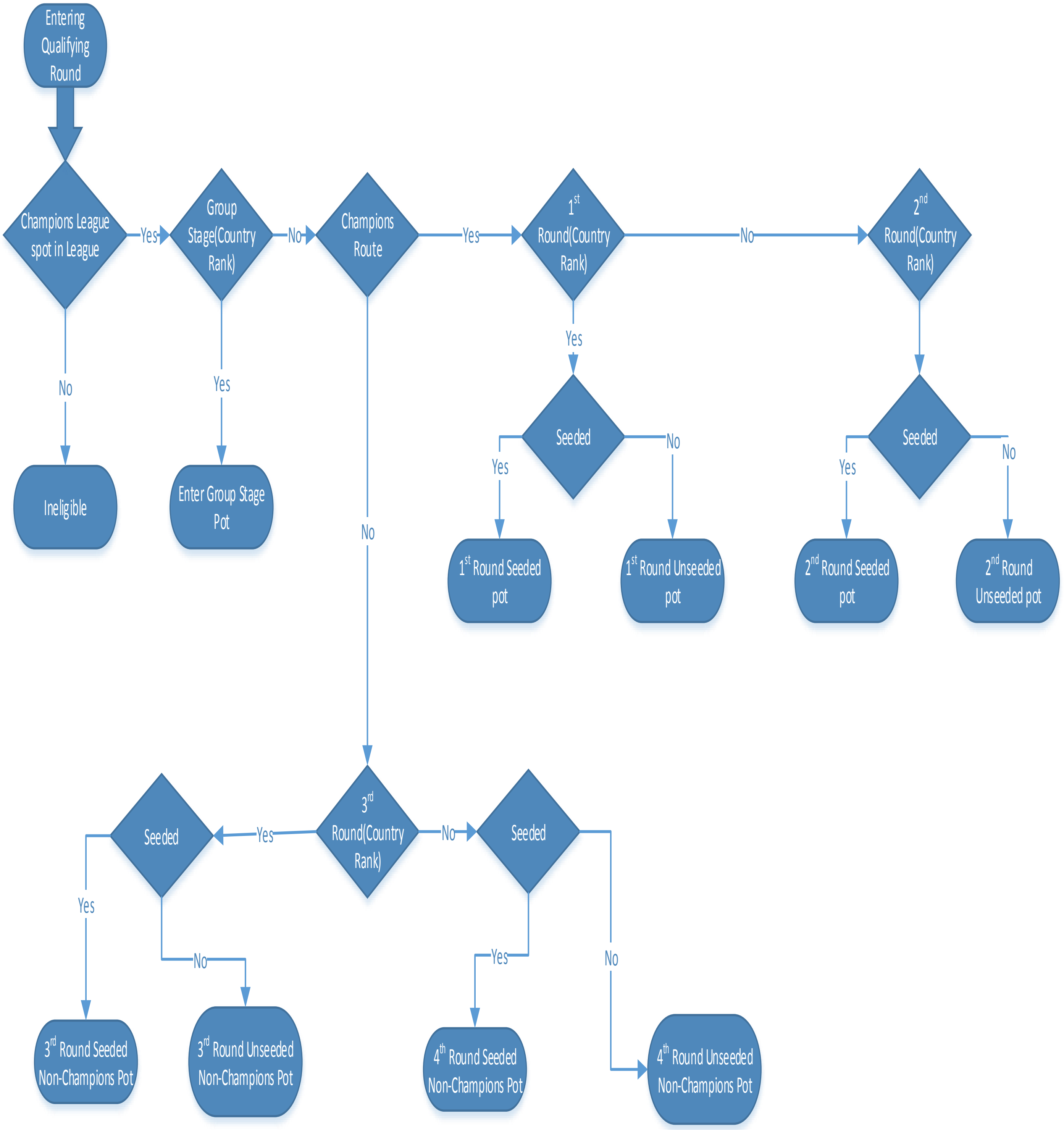
1. Check Access List -> 54
2. Country check -> Gibraltar
3. Number of teams to enter -> 1
4. Round to enter -> Round 1
5. Team Check -> Lincoln Red Imps
6. Insert to Round 1

Repeat 1-6 until all teams have been allocated to round 1

7. Perform sorting method to arrange Seeding for round (based on each teams ranking)

Permutations for Teams entering the Champions League

Data Flow Diagram for Team



Data Flow Diagram Steps

The final additional analysis carried out was the steps taken to determine where a team entering the Champions League will be placed. The Access List and Initial Algorithm are used and many parts taken in to consideration for making the decision throughout each step to determine what path the team in question is eligible for.

Steps

Enter Qualifying Round – Chosen team will enter the algorithm process that will deem where they should be placed.

Champions League Spot in League – Check done to see if the selected team has finished in a champions league spot in their respective league. If not in the right spot they are ineligible.

Group Stage (Country Rank) – The program will check the country ranking for the teams country. If the country is ranked 1-13 they will Enter the Group Stage Pot. Otherwise they enter the qualifying route.

Champions Route – If the team has won their league they will enter the Champions Route. If they are not champions but are eligible for the Champions League they will enter the Non-Champions Route.

3rd Round (Country Rank) – The Country Rank for the team will determine if they enter the 3rd or 4th qualifying round.

Seeded (3rd round) – The current team ranking of the selected team will determine whether they are seeded or not. This is finalized by taking all teams in the pot for the 3rd round and arranging them by rank. Of the 10 teams in the pot the top 5 ranked are seeded, the bottom them are unseeded.

Seeded (4th round) – The current team ranking of the selected team will determine whether they are seeded or not. This is finalized by taking all teams in the pot for the 4th round and arranging them by rank. Of the 10 teams in the pot the top 5 ranked are seeded, the bottom them are unseeded.

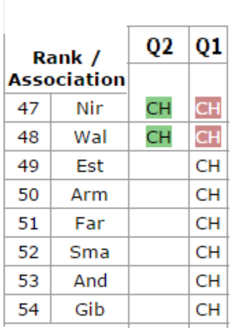
1st Round (Country Rank) - The Country Rank for the team will determine if they enter the 1st round. Otherwise they will either enter the 2nd or 3rd round.

Seeded (1st Round) - The current team ranking of the selected team will determine whether they are seeded or not. This is finalized by taking all teams in the pot for the 1st round and arranging them by rank. Of the 8 teams in the pot the top 4 ranked are seeded, the remaining 4 are unseeded.

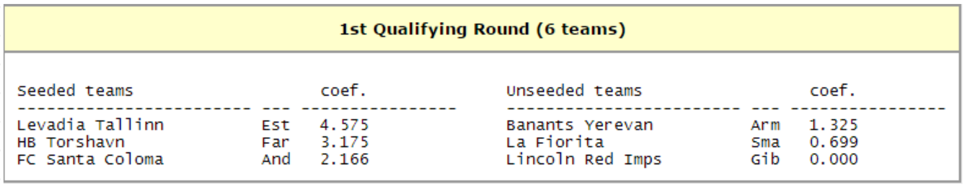
2nd Round (Country Rank) - The Country Rank for the team will determine if they enter the 2nd round. Otherwise they will either enter the 3rd round.

Seeded (2nd Round) - The current team ranking of the selected team will determine whether they are seeded or not. This is finalized by taking all teams in the pot for the 2nd round and arranging them by rank. Of the 34 teams in the pot the top 17 ranked are seeded, the remaining 17 are unseeded.

Seeded (3rd Round) - The current team ranking of the selected team will determine whether they are seeded or not. This is finalized by taking all teams in the pot for the 3rd round and arranging them by rank. Of the 20 teams in the pot the top 10 ranked are seeded, the remaining 10 are unseeded.



Rank / Association		Q2	Q1
47	Nir	CH	CH
48	Wal	CH	CH
49	Est		CH
50	Arm		CH
51	Far		CH
52	Sma		CH
53	And		CH
54	Gib		CH

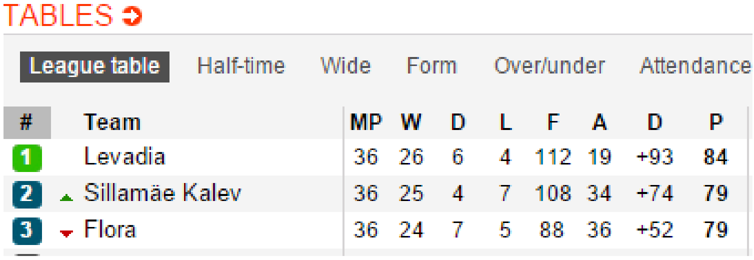


1st Qualifying Round (6 teams)

```
Seeded teams                      coef.              Unseeded teams                    coef.
------------------------- --- ----------------     ------------------------- --- ----------------
Levadia Tallinn           Est   4.575              Banants Yerevan           Arm   1.325
HB Torshavn               Far   3.175              La Fiorita                Sma   0.699
FC Santa Coloma           And   2.166              Lincoln Red Imps          Gib   0.000
```

Note: Due to Special circumstances 2 teams from round 1 were promoted to round 2. Therefore the seeding was calculated for 6 teams rather than 8. (Benfica Winning the Europa League)

Example: Levadia Tallinn

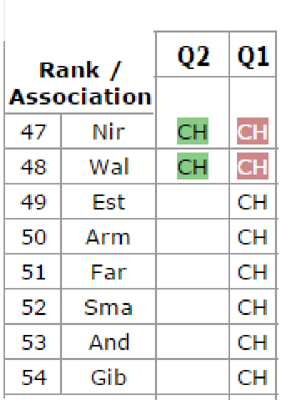


TABLES

League table | Half-time | Wide | Form | Over/under | Attendance

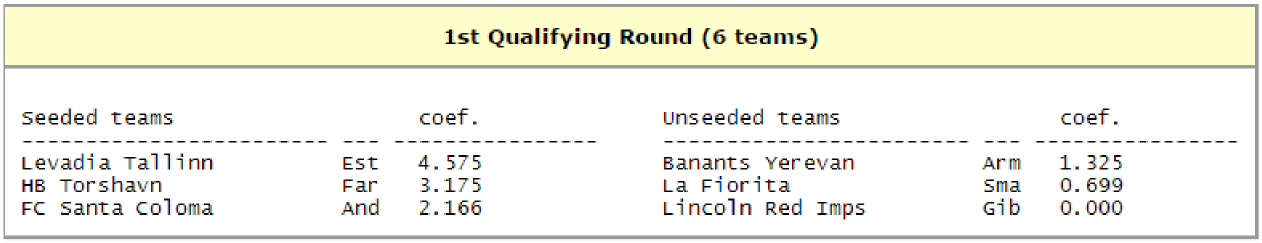
#		Team	MP	W	D	L	F	A	D	P
1		Levadia	36	26	6	4	112	19	+93	84
2	▲	Sillamäe Kalev	36	25	4	7	108	34	+74	79
3	▼	Flora	36	24	7	5	88	36	+52	79

Levadia have won their League therefore will enter the Champions Route



Rank / Association		Q2	Q1
47	Nir	CH	CH
48	Wal	CH	CH
49	Est		CH
50	Arm		CH
51	Far		CH
52	Sma		CH
53	And		CH
54	Gib		CH

Levadia's Country Estonia are currently ranked 49th in Europe therefore the team who wins their league enters the first round.



1st Qualifying Round (6 teams)

```
Seeded teams                    coef.                Unseeded teams                  coef.
------------------------ --- ----------------        ------------------------ --- ----------------
Levadia Tallinn          Est   4.575                 Banants Yerevan          Arm   1.325
HB Torshavn              Far   3.175                 La Fiorita               Sma   0.699
FC Santa Coloma          And   2.166                 Lincoln Red Imps         Gib   0.000
```

Levadia are to be seeded in round one. This is because of the 6 teams in the round, Levadia are ranked in the top 3

Levadia are just one example for how the algorithm works, each team have their own round and seeding determined based on the algorithm working with the access list, country ranking and their own individual team ranking to determine seeding.

Review of Implementation Phase

The Implementation Phase was successful throughout however there were various problems/obstacles encountered along the way with solutions integrated for each issue.

The original Design was kept to a certain extent however some changes had to be made to accommodate certain elements that became redundant due to the integration and issues outside of the developers control such as licensing issues.

The biggest issue during the Implementation phase was the Integration and use of the Football Data API Service. A lot of issues occurred early on acquiring the developer key and correct Schema to use the data required to successfully make live changes to the database.

Initially it was proposed the teams entering each round for some leagues would be updated through the Service. For Example, the Service could provide live updates of standings in the Spanish League and should a change occur for teams occupying the Champions League spots, this would be replicated in the database and changed on the API.

Unfortunately the original plan for this to be implemented had to be scrapped due to developer licensing issues with the provider. The Football API used included League standings for 13 countries that were updated through a times Azure Worker Role.

The original implementation checked the League Standings accessed through an API key and the database updated through CRUD using a Visual C# Console Application that was deployed to Azure using .NET HTTP and System Threading async run tasks. A comparison with the current SQL Azure Database was performed and any changes with a team currently top of a specific League were checked. When a different team was found on the Sports Radar API, an update was replicated on the database. A full replication of this change was then replicated on the Administrator MVC end, the Server API and the Android Client itself.

The Core functionality of the Application was to make it ‘Live’ and with this function unavailable because of licensing issues due to Web deployment another alternative had to be implemented instead.

The proposed change was to fully implement an MVC 5 Web Service for an Administrator to view the all sections of the Application and make appropriate changes. In some ways this actually provided further functionality to the Project with further security such as OAUTH and SSL (HTTPS) integrated in to the system. The original plan was not to implement such a service however due to the problem occurring this was the solution proposed which was the perfect solution.

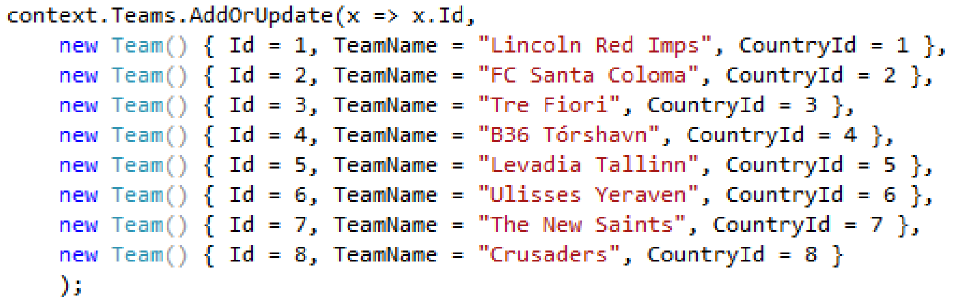
Initial Code First Migrations

Originally it was proposed the initial data would be entered through context inserts but problems arose due to the fact such large amounts of data was required. During the early stages of development the database consisted of Test Teams comprising of no more than 5-10 at a time. After the algorithm was implemented and all the API/MVC Controllers were integrated the full amount of data was required for the Application to become fully functional.

The data to be put in the database required over 700 single inserts alone. These were needed to accommodate the Team Ranking (508 inserts), Country Ranking (54 inserts), Group stage (32 inserts) and all the qualifying rounds (over 100 inserts depending on algorithm permutations).

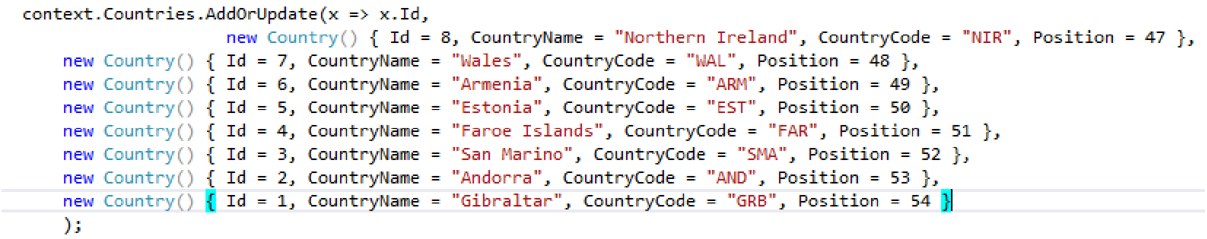
Below are the original entries used that became time consuming due to the amount of data that would have been required:

1st Qualifying Round Initial Database Migration Inserts:



```
context.Teams.AddOrUpdate(x => x.Id,
    new Team() { Id = 1, TeamName = "Lincoln Red Imps", CountryId = 1 },
    new Team() { Id = 2, TeamName = "FC Santa Coloma", CountryId = 2 },
    new Team() { Id = 3, TeamName = "Tre Fiori", CountryId = 3 },
    new Team() { Id = 4, TeamName = "B36 Tórshavn", CountryId = 4 },
    new Team() { Id = 5, TeamName = "Levadia Tallinn", CountryId = 5 },
    new Team() { Id = 6, TeamName = "Ulisses Yeraven", CountryId = 6 },
    new Team() { Id = 7, TeamName = "The New Saints", CountryId = 7 },
    new Team() { Id = 8, TeamName = "Crusaders", CountryId = 8 }
    );
```

Country Ranking Initial Database Migration Inserts:



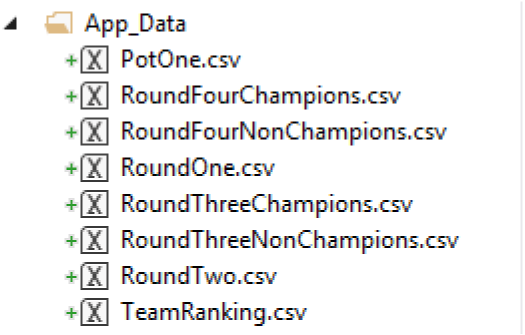
```
context.Countries.AddOrUpdate(x => x.Id,
                    new Country() { Id = 8, CountryName = "Northern Ireland", CountryCode = "NIR", Position = 47 },
    new Country() { Id = 7, CountryName = "Wales", CountryCode = "WAL", Position = 48 },
    new Country() { Id = 6, CountryName = "Armenia", CountryCode = "ARM", Position = 49 },
    new Country() { Id = 5, CountryName = "Estonia", CountryCode = "EST", Position = 50 },
    new Country() { Id = 4, CountryName = "Faroe Islands", CountryCode = "FAR", Position = 51 },
    new Country() { Id = 3, CountryName = "San Marino", CountryCode = "SMA", Position = 52 },
    new Country() { Id = 2, CountryName = "Andorra", CountryCode = "AND", Position = 53 },
    new Country() { Id = 1, CountryName = "Gibraltar", CountryCode = "GRB", Position = 54 }
    );
```

This was a major problem however a solution as proposed to solve this and that was using a CSV Helper to perform inserts through a CSV file for each context.

CSV Files were created for all Qualifying Rounds, Country Ranking and Team Ranking. This was an excellent solution due to the reduction in time constraints to have inserts going through a CSV.

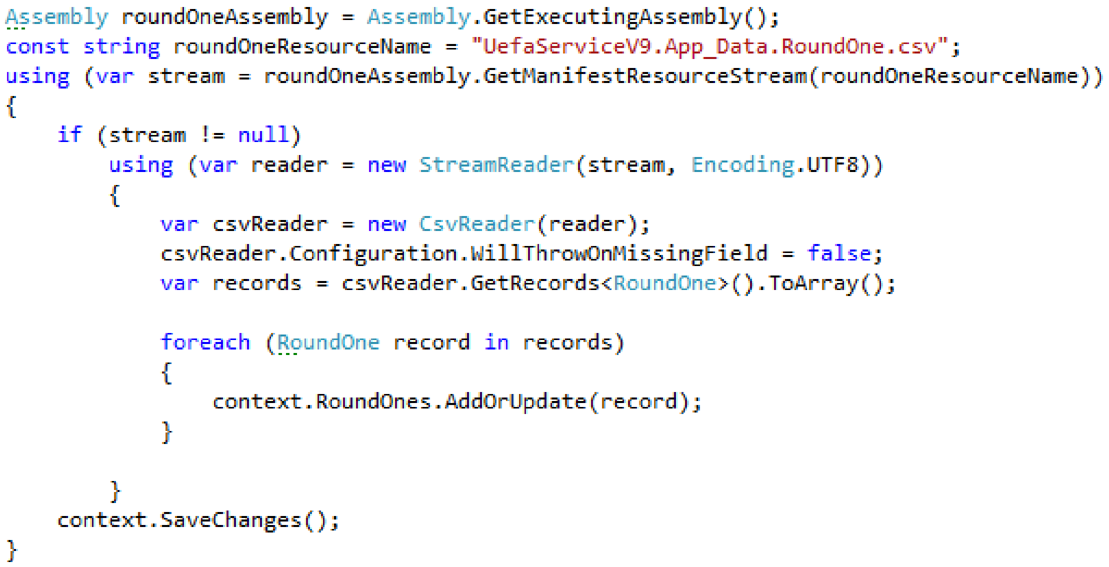
The Context ‘AddOrUpdate’ for the two above examples were altered to the following using CSV files instead that were defined as ‘Embedded Resources’

CSV Files:



```
App_Data
    PotOne.csv
    RoundFourChampions.csv
    RoundFourNonChampions.csv
    RoundOne.csv
    RoundThreeChampions.csv
    RoundThreeNonChampions.csv
    RoundTwo.csv
    TeamRanking.csv
```

1st Qualifying Round Context ‘AddOrUpdate’:



```
Assembly roundOneAssembly = Assembly.GetExecutingAssembly();
const string roundOneResourceName = "UefaServiceV9.App_Data.RoundOne.csv";
using (var stream = roundOneAssembly.GetManifestResourceStream(roundOneResourceName))
{
    if (stream != null)
        using (var reader = new StreamReader(stream, Encoding.UTF8))
        {
            var csvReader = new CsvReader(reader);
            csvReader.Configuration.WillThrowOnMissingField = false;
            var records = csvReader.GetRecords<RoundOne>().ToArray();

            foreach (RoundOne record in records)
            {
                context.RoundOnes.AddOrUpdate(record);
            }

        }
    context.SaveChanges();
}
```

Overall the original problem encountered was matched and bettered with an excellent solution that handled the large data populating the database flawlessly.

The implementation of the Android Client went quite well overall however there were several obstacles hit along the way with solutions proposed for each instance. The main issue was getting the data from API’s on the server side and parsing them in to JSON format that could then allow the various amounts of required data to be used on the App for user viewing.

The solution proposed was to use the Apache HTTP Components in order to successfully get data from the API and implement it throughout the app. A service handler was then developed within the code to make a HTTP call but with try and catches allowing any issues that may have originally occurred through a basic http call to be caught reducing errors to a minimum. Having fully tested this Service handler and the HTTP calls the level of errors occurring are less than 1% and even when errors occurred they were due to network issues on the mobile itself. Any errors that ever occurred were captured regardless.

With regard to the design of the Android App there were some layout problems that cause different screen sizes to not be compatible with the originally developed layout code. The App was to be compatible with all screen sizes including Tablets.

The solution proposed to solve this was creating separate resource files that would specifically set different layouts depending on the screen size of the device. This was replicated throughout all the screens and the issue was solved.

Minutes of Meetings

Tuesday, 27th January 10.45am

Discussed Algorithm of the UEFA Ranking System and made proposals to implement the algorithm using sample C# Console Applications

Discussed in detail ideas for initial implementation of the Server Side

Tuesday, 3rd February 10.45am

Talked about selection of API's proposed to use for Live Data updates on the Server API

Discussion made on Client side choices. Potential development for either Android or Windows Mobile Devices

Tuesday, 10th February 10.45am

We discussed issues relating to calling an API through a Client for initial testing of League Standings that can potentially be implemented in the project.

It was suggested to allow the Web Service behave as a client to the API Service as well as a Service itself.

Tuesday, 17th February 10:45am

We discussed the overall distribution and deployment of the project.

It was agreed the System would work through a Server Side (Azure with SQL Azure Database) and Android as the Mobile Operating System Client.

Further explanations of the algorithm were done. Supervisor suggested drafting up the algorithm to provide further clarification in Project Meeting.

Tuesday, 24th February

Meeting Postponed

Tuesday, 3rd March

We had a discussion on the current progress of the Project which was very constructive with progress being steadily made. Suggestions were made with regard to security that could be implemented, taken note of.

It was suggested by the supervisor that for the panel meeting another draft of the algorithm would be created to show on the day. A PowerPoint presentation of what had been implemented already and was needed to be implemented was suggested to be shown on the day of the panel meeting also.

Tuesday, 10th March

We discussed the current progress of the Server Side. Supervisor pleased with current progress of Server side with an API fully functioning with Test Data on a SQL Azure database using Entity Framework Code First Migrations.

The client side was also discussed with initial implementation underway. The Football API service was marked a doubt due to client issues taking down the data and working with the database.

Tuesday, 17th March

Meeting Postponed

Tuesday, 24th March

We discussed the current progress of both Server and Client. There were issues regarding the Android client getting the data from the Server API and this issue was addressed at the meeting with a suggestion from the supervisor to follow the Apache method to Parse the data on the API in to JSON.

The current progress of the project was good however a working version of all elements pieced together was suggested by the supervisor to release soon to avoid problems closer to the project deadline.

Tuesday, 31st March

We had a general discussion on the current progress of the project. Progress from the client side marked a slight concern due to issues still present for the Client HTTP calls.

Some issues were arising with the Azure service and proposed suggestions from myself to the supervisor as to how to come up with a solution were made and agreed upon.

Tuesday, 14th April

We had a detailed discussion on the current progress of the project. All the functionality was implemented on the server side. An issue had arisen regarding code first migrations due to the large amounts of data being inserted. The solution of using CSV files for the code first migrations instead was taken note of and agreed upon being a reasonable solution.

The progress from the client side was seen as good from the supervisor due to the issue of API data being called through HTTP resolved over the Easter Break. Some functionality was left to be implemented and it was suggested it should be done a.s.a.p.

This was the final meeting and it was suggested to have a fully working demo for the Demonstration the following Wednesday with all functionality working and fully implemented.

Bibliography

Research Document

Wahl, C. (2014). *5 Year Ranking of the UEFA*. Germany: Sports Relive App https://play.google.com/store/apps/details?id=fjw.table&hl=en

Patierno, P. (2012). *Uefa Ranking*. Italy. http://www.windowsphone.com/en-ie/store/app/uefa-ranking/3357abc1-84fa-48b0-a38c-1f3edc0da7b4

Hay Newman, L. (2014). Android Users Won't Drop Money on Just Any App. [online] Slate Magazine. Available at: http://www.slate.com/blogs/future_tense/2014/06/26/there_are_twice_as_many_android_users_as_ios_but_ios_users_spend_double.html [Accessed 3 Oct. 2014].

Stroud, F. (2014). *What is an Android App? Webopedia*. [online] Webopedia.com. Available at: http://www.webopedia.com/TERM/A/android_app.html [Accessed 3 Oct. 2014].

Johnson, L. (2013). Android SDK: Making Remote API Calls - Tuts+ Code Tutorial. [online] Code Tuts+. Available at: http://code.tutsplus.com/tutorials/android-sdk-making-remote-api-calls--mobile-17568 [Accessed 3 Oct. 2014].

Wasson, M. (2014). Getting Started with Web API 2 (C#). [online] The Official Microsoft ASP.NET Site. Available at: http://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api [Accessed 3 Oct. 2014].

Bibliography: Vogel, L. (2014). Android SQLite database and content provider - Tutorial. [online] Vogella.com. Available at: http://www.vogella.com/tutorials/AndroidSQLite/article.html [Accessed 3 Oct. 2014].

Azure.microsoft.com, (2014). Get Started with Azure Mobile Services for Android apps. [online] Available at: http://azure.microsoft.com/en-us/documentation/articles/mobile-services-dotnet-backend-android-get-started/ [Accessed 3 Oct. 2014].

Shackles, G. (2012). Cross Platform Mobile Development in .NET. 1st ed. Sebastopol, CA: O'Reilly & Associates.

Surer, S. (2014). Windows Phone grows US share by 400K new users. [online] WMPoweruser. Available at: http://wmpoweruser.com/windows-phone-grows-us-share-by-400k-new-users/ [Accessed 3 Oct. 2014].

Forrest, C. (2014). Apple v. Google: The goliath deathmatch by the numbers in 2014. [online] TechRepublic. Available at: http://www.techrepublic.com/article/apple-v-google-the-goliath-deathmatch-by-the-numbers-in-2014/ [Accessed 3 Oct. 2014].

Violino, B. (2013). A clear-eyed guide to Android's actual security risks. [online] InfoWorld. Available at: http://www.infoworld.com/article/2609338/android/a-clear-eyed-guide-to-android-s-actual-security-risks.html [Accessed 3 Oct. 2014].

Microsoft SQL Azure Enterprise Application Development. Packt Publishing. p. 420. ISBN 978-1-84968-080-6.

Technical Architecture

viaForensics, (2014). viaForensics. [online] Available at: https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/ [Accessed 16 Oct. 2014].

Post-Project Review

Kassies, B. (2015). Access list for European Cup Football 2015/2016. [online] Kassiesa.home.xs4all.nl. Available at: http://kassiesa.home.xs4all.nl/bert/uefa/access2015.html [Accessed 1 May 2015].