|  |
| --- |
|  |

|  |
| --- |
|  |
| Post-Project Review |
|  |

Table of Contents

# Project Overview

page about what the project is and what I intended it to do when I started it

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 |

Bert Kaissies <http://kassiesa.home.xs4all.nl/bert/uefa/access2014.html>

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.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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) |  |
| 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.

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

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

1. 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 finalised 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 finalised 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.

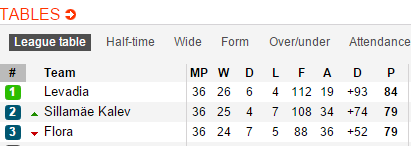
Seeded (3rd Round) - The current team ranking of the selected team will determine whether they are seeded or not. This is finalised by taking all teams in the pot for the 3rd round and arranging them by rank. Of the 20 teams in the pot the top 10 ranked are seeded, the remaining 10 are unseeded.

Data flow Example



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



Levadia have won their League therefore will enter the Champions Route



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



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.

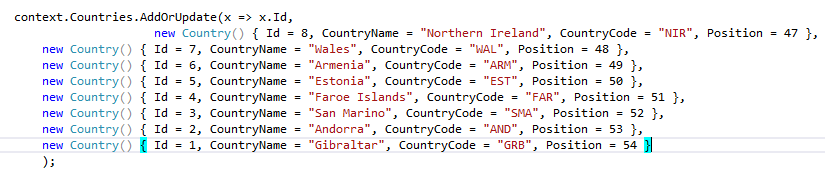
The data to be put in the database required over 700 single inserts alone. These were needed to accommodate the Team Ranking (508 inserts), Country Ranking (54 inserts), Group stage (32 inserts) and all the qualifying rounds (over 100 inserts depending on algorithm permutations).

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:



Country Ranking Initial Database Migration Inserts:

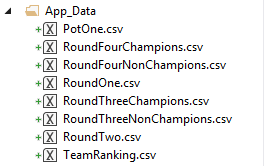


This was a major problem however a solution as proposed to solve this and that was using a CSV Helper to perform inserts through a CSV file for each context.

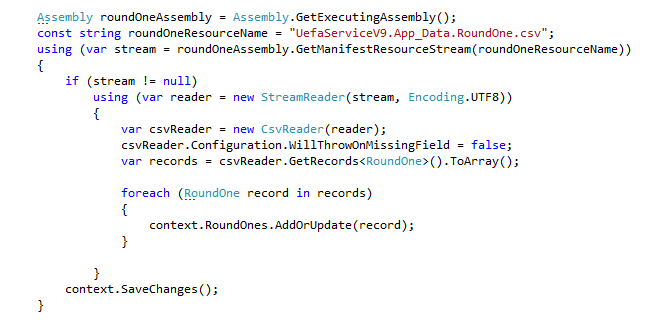
CSV Files were created for all Qualifying Rounds, Country Ranking and Team Ranking. This was an excellent solution due to the reduction in time constraints to have inserts going through a CSV.

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:



1st Qualifying Round Context ‘AddOrUpdate’:



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. An errors that ever occur are 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

Bert Kaissies