Queen's University Belfast

CSC7054 - Web and Mobile App Development

Danske Bank Premiership Android App

Student 1 - 10000000
Student 2 - 40000000

Introduction to App

For this project we decided as a group to create an Android App using Eclipse Android Developer Tools. After discussing a number of different options we choose to develop an App which would provide users with information on the IFA Premiership, which is currently sponsored by Danske Bank. This is due to a lack of information available for the league online and although there is a free app available at present, we felt collectively that we could produce something that would be more marketable and provide more information to the user. The design and features of the App itself are discussed later in this report.

Requirements

Problem Statement

The original problem faced by the potential customer is to create an App which holds an equal amount of information on all teams within the IFA Premiership. For popular teams such as Linfield information is readily available, but this is not the case for smaller teams within the league. The information held must be current and the app should enable the user the search social media sources to ensure the most up to date information is captured. For new supporters of a team the App should give them information on how to contact the team of their choice (phone, email, etc.) and also find their ground, information which would also be very useful to travelling supporters attending away games. Statistics about players would also be beneficial and it is hoped that journalists reporting on the league could use the App as an accomplice for their reporting duties.

System Requirements

A requirement is defined as "a condition or capability to which a system must conform". A number of requirements for the App were extracted from the Problem Statement and are listed below;

No.	Requirement	Brief Description	Priority Weight
1.	Every team in IFA Premiership represented.	All 12 teams within the IFA Premiership are available for the user to select. These teams are; Ballinamallard United, Ballymena United, Cliftonville, Coleraine, Crusaders, Donegal Celtic, Dungannon Swifts, Glenavon, Glentoran, Linfield, Lisburn Distillery and Portadown.	

2.	Up to date team news.	All news provided for a specific team should be relevant and the most recent which is available, ensuring the App retains a cutting edge image and can be marketed as such.	
3.	Directions to team grounds.	A map should be provided along with an exact address, and possibly directions, which clearly describe the location of the team ground to the user.	
4.	Team contact information.	As well as an address (discussed previously) a phone number, email address and name of a point of contact with the club (e.g. administrator) must be present.	
5.	Player information and statistics	Relevant and up to date information and statistics will be available for at least the starting 11 for each team within the league. Information should include, but is limited to; Appearances, Goals, Yellow/Red cards, Man of the match, Clean sheets (where appropriate).	
6.	App is easy to navigate.	Usability is of paramount importance and at all times the user must be able to access the home screen and/or return to the previous page they were viewing.	
7.	Images on App are clear and of a high quality.	To improve the aesthetics of the App all images must be of a high quality and scaled appropriately for the particular platform.	
8.	Any text within App is clear and concise.	The App must be usable for all ages and to ensure this simple text used with only important information to reduce App size.	

Functional Requirements Specification

Stakeholders

People with possible interest in the system, in a stakeholder capacity, may include;

- IFA
- Danske Bank

Actors and Goals

As the App will be available to the all members of the general public they will be the main actors who will interact with this system/App. It is assumed that their goal will be to obtain some sort of information on some aspect of a particular team within the IFA Premiership. This may include any aspect of the System Requirements which will be included in the finished working App.

Use Cases

For the purpose of this exercise 'Glentoran' has been used as an example. The use cases listed are relevant to all teams contained within the IFA Premiership.

No.	Casual Description	Related Requirement(s)
1	User wants to find the contact number for Glentoran regarding purchase of tickets.	1, 5.
2	User wants to find a map to Glentoran's ground 'The Oval' for an upcoming away game.	1, 4.
3	User has highlighted 'Stuart Elliot' as their favourite player and wants to determine how many goals he has score this season.	1, 6, 7.
4	User wants to check what other fans have been saying on BBC about Glentoran's upcoming match.	1, 3.
5	User wants to be able to clearly see the Glentoran icon on the main screen.	1, 9, 10.
6	User wants to find out information on a player from Crusaders who will be signing for Glentoran next season.	1, 6, 7.
7	Users want to be able to see news on their favourite teams	1,2

Use Case Diagrams

Please see **APPENDIX B – Use Case Diagram**.

User Interface Specification

Preliminary Design

After a series of brainstorming sessions an initial design was sketched out with some ideas as to how the main screen of the App would appear and the functions we would want it to have. We decided upon a simplistic design to ensure the App would be user friendly and also placing aesthetics as a factor of paramount importance. These initial sketches are shown below;

Progression of Design

In order to obtain a clearer view on how our App may appear these sketches were then taken and some visions of potential outcomes were produced on Photoshop. This allowed us to visualise how the App may appear and provided at starting point from which we could build and think about possible additional features. These mock ups are shown below;

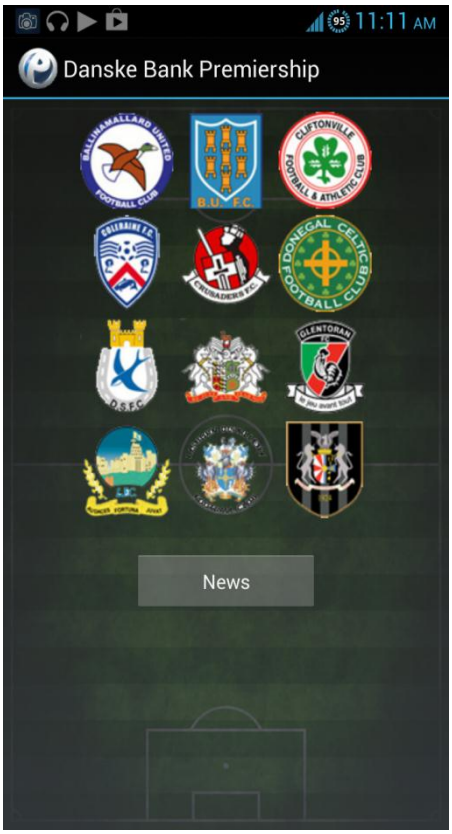
Design

Features

The App itself will provide information on each of the 12 teams within the IFA premiership. The initial screen the user is presented with shows each team badge as an icon which they can select. Once an icon is selected they are brought to a screen dedicated to their desired team with a range of current information. To achieve this club badges were obtained and scaled accordingly then placed in a grid layout. Below is a small piece of the code used to achieve the desired layout.

```
<LinearLayout
    android:id="@+id/mainback"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/back_xl"
    android:orientation="vertical" >

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp" >

        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal" >

            <ImageButton
                android:id="@+id/ballinamallard"
                android:layout_width="75dp"
                android:layout_height="75dp"
                android:background="@android:color/transparent"
                android:onClick="onClick_Ballinamallard"
                android:scaleType="fitXY"
                android:src="@drawable/ballinamallard" />

            <ImageButton
                android:id="@+id/ballymena"
                android:layout_width="75dp"
                android:layout_height="75dp"
                android:layout_marginLeft="5dp"
                android:background="@android:color/transparent"
                android:src="@drawable/ballymena" />

            <ImageButton
                android:id="@+id/cliftonville"
                android:layout_width="75dp"
                android:layout_height="75dp"
                android:layout_marginLeft="5dp"
                android:background="@android:color/transparent"
                android:src="@drawable/cliftonville" />
        </TableRow>
```

Xml code to demonstrate the table layout structure.



Screenshot to demonstrate the layout of the main activity screen.

For each team there is a range of features which are easy to navigate, while maintaining an aesthetic appeal throughout. These features include;

- Team Information/Player Information
- News Feed
- Photographs
- Contact Information/Map

- Database
- League News (only available using the main screen).

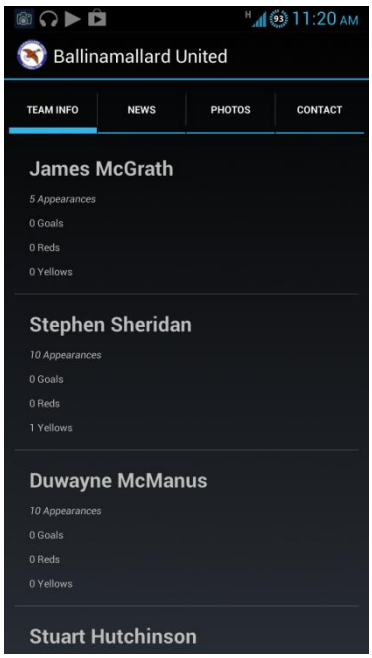
The relevant code for these features, along with screenshots, are described and depicted below.

Team Information/Player Information

For the purposes of this report the java code to retrieve and display the database information on the players has been omitted. However, it can be viewed in any of the team activity files e.g Ballymena Activity. Further supporting code is also listed in the RequestTask and TeamActivity java files.

```
<ListView
    android:id="@+id/layoutTab1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" />
```

This is an example of the information that is retrieved from the database developed to support the app. It is currently displaying the players in the Ballinamallard team, listed by the number of times they have appeared for the team - they are also order from lowest appearances to highest appearances.



ListView to demonstrate the player information

News Feed

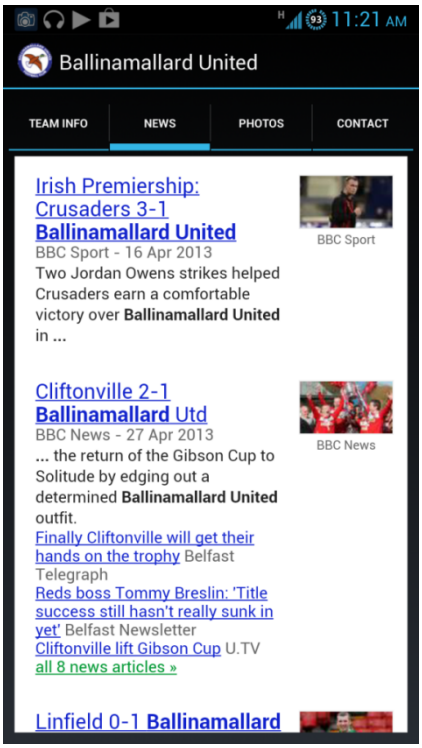
```
<WebView
    android:id="@+id/layoutTab2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" />
```

XML code to show a webview.

```
spec = tabHost.newTabSpec("news").setIndicator("News").setContent(R.id.layoutTab2);
WebView myWebView = (WebView) findViewById(R.id.layoutTab2);
myWebView.setWebViewClient(new WebViewClient());
myWebView.loadUrl("https://www.google.co.uk/search?hl=en&gl=uk&tbm=nws&q=ballinamallard+united
tabHost.addTab(spec);
```

Java Code to load a webview into the webview defined in the XML and the URL to load.

This is the news feed for the Ballinamallard United team. It loads a Google news page and has a predefined search of Ballinamallard United loaded in. The results are then displayed in the app by using the setWebViewClient line of code. As the search is only carried out at the time of loading the news tab, the information that is displayed is up to date and current. The WebViewClient is used as otherwise the default browser would be passed the URL and load externally to the app.



The News Tab for Ballinamallard.

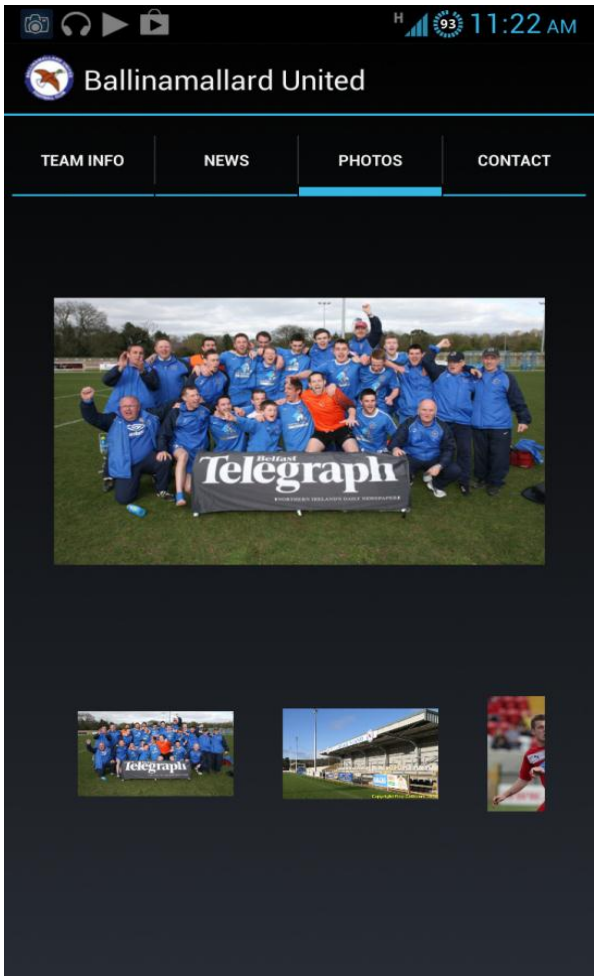
Photographs

```
<LinearLayout
    android:id="@+id/layoutTab3"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imagedisplay"
        android:layout_width="600dp"
        android:layout_height="300dp"
        android:layout_gravity="center"
        android:src="@drawable/ballinamallard1" />

    <HorizontalScrollView
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center" >

        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal" >

            <ImageView
                android:id="@+id/imagedisplay1"
                android:layout_width="125dp"
                android:layout_height="125dp"
                android:padding="15dp"
                android:src="@drawable/ballinamallard1" />
```

XML code for the Photos Tab

```
spec = tabHost.newTabSpec("photos").setIndicator("Photos")
        .setContent(R.id.layoutTab3);
display = (ImageView) findViewById(R.id.imagedisplay);
ImageView image1 = (ImageView) findViewById(R.id.imagedisplay1);
ImageView image2 = (ImageView) findViewById(R.id.imagedisplay2);
ImageView image3 = (ImageView) findViewById(R.id.imagedisplay3);
ImageView image4 = (ImageView) findViewById(R.id.imagedisplay4);
ImageView image5 = (ImageView) findViewById(R.id.imagedisplay5);
ImageView image6 = (ImageView) findViewById(R.id.imagedisplay6);
image1.setOnClickListener(this);
image2.setOnClickListener(this);
image3.setOnClickListener(this);
image4.setOnClickListener(this);
image5.setOnClickListener(this);
image6.setOnClickListener(this);
tabHost.addTab(spec);
```

Java code to assign images with locations and how to change main image.



Screenshot of the photos tab for Ballinamallard United.

The code above is used to generate the photos tab content as shown immediately to the left. There is a main photo which takes up a larger portion of the screen in comparison with the other images. The other images can be loaded by scrolling to them along a scroll bar and then tapping the desired image to be enlarged. The local storage of the photos has dramatically increased the size of the app overall, however a benefit of this method is that this feature can be enjoyed even when an active internet connection is absent.

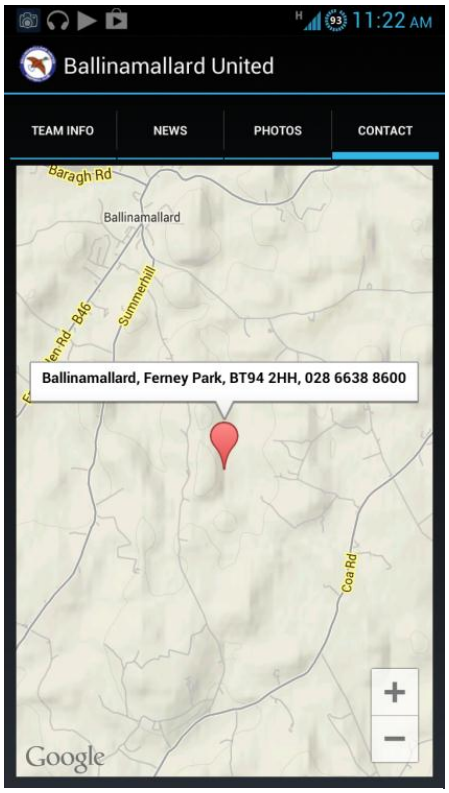
Contact Information/Map

```
<fragment
    android:id="@+id/layoutTab4"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_below="@+id/tableRow4" />
```

XML fragment used to generate the mapview.

```
spec = tabHost.newTabSpec("news").setIndicator("Contact")
        .setContent(R.id.layoutTab4);
map = ((MapFragment) getFragmentManager().findFragmentById(
        R.id.layoutTab4)).getMap();
map.addMarker(new MarkerOptions().position(LOCATION_BALLINAMALLARD)
        .title("Ballinamallard," + " Ferney Park, BT94 2HH,"
                + " 028 6638 8600"));
map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
map.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(54.4000,
        -7.5833), 13.0f));
```

Java code used to implement the map and the map default location.



The map for the Ballinamallard United homeground.

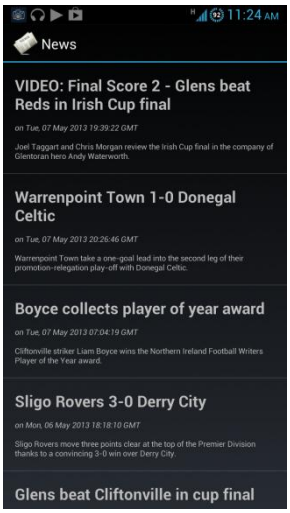
To implement maps, the relevant google libraries had to be imported for use in the project and API keys had to be obtained. Certain additional permissions were also necessary. Fragments were used to define the layout of the map relative to the screen. One such fragment can be seen in the XML code above. Fragments also had to be called in the java code for the map to display. The map type, location, marker and zoom levels are all defined in the preceding screenshot.

Database

A database was created to host the team and player information; the output of the database can be seen in the Team Info tab.

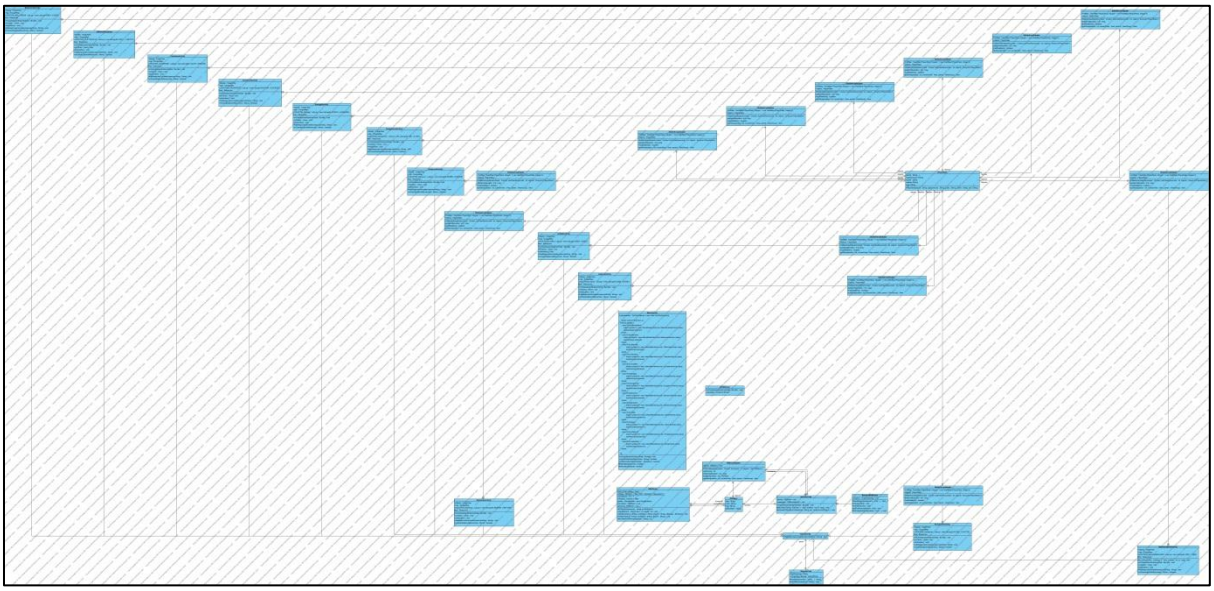
League News

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

        <ListView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:id="@android:id/list"/>
</LinearLayout>
```



The java file news activity has been omitted due to the constraints of writing this report. The code is located in the news activity file. The code retrieves the latest news regarding the league from a specific bbc news feed and then displayed the information on the screen.

Domain Model



See attached high resolution .jpg image for visual clarity.

Plan of Work

Please see attached Gantt chart in Appendix C.

References

1. Google (2013) Developers - Design, Available at: http://developer.android.com/design/index.html
2. Google (2013) Developers – Develop, Available at: http://developer.android.com/develop/index.html
3. StackOverFlow Available at: http://stackoverflow.com/
4. Vogella (2011) Available at: http://www.vogella.com/3
5. Various Guides to Demonstrate functionality

APPENDIX A

Individual Contributions Breakdown

Scores out of 30

Student 1 - 10000000

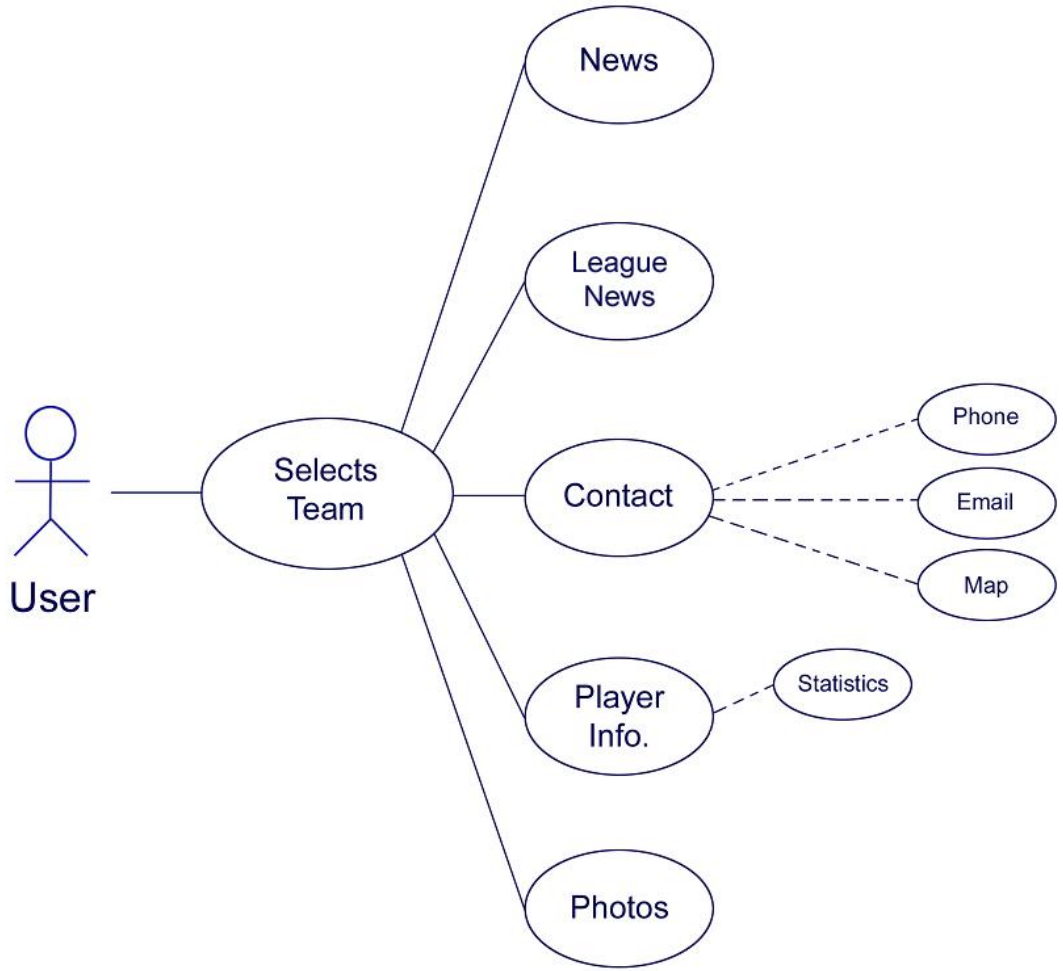
21/30

Student 2 – 40000000

22/30

APPENDIX B

Use Case Diagram



APPENDIX C

Project Gantt Chart

	Timescale	Week 1	Week 2	Week 3
Project Deliverables				
Main Activity Design and XML				
Team Activities Design and XML				
Creation Java Code for Activities and Intents				
Setting up of Photo Scroll Pane in Team Activities				
Creation of News Feed in the Team Activities				
Implementation of Maps in Team Activities				
Collect required photos for Photo Tab				
Embed photos in corresponding locations				
Develop database for teams				
Creation of PHP script to poll database				
Implementation of JSON Parser and corresponding display code				
Implementation of BBC news feed				