University of Dublin

TRINITY COLLEGE

Football Management mobile application

ROBERT SQUIRE
B.A.(Mod.) Business & Computing

Final Year Project April 2014
Supervisor: DECLAN O’ SULLIVAN

School of Computer Science and Statistics

O'Reilly Institute, Trinity College, Dublin 2, Ireland
DECLARATION

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university

______________________________  _______________________
Name                                             Date
Abstract

The purpose of this report is to outline the implementation of an Android application which enables football managers in Ireland to organise their squad formation, create tactical animations of different sequences of play, manage fixtures /upcoming events and contact players through messaging functionality.

Applications and systems already well established have had a great influence on the creation of this application but none have delivered a total quality approach in visualisation of football management. This application delivers a high quality visualisation to football management creating clear representations of different plays.
Acknowledgements

I would like to thank my supervisor Declan O’ Sullivan for ongoing support and recommendations throughout the year. Ongoing meetings really aided the structure and scope of this mobile application and it could not have been delivered in the standard that it was without this structure. I would also like to thank my colleagues in Business and computing for their insightful ideas throughout different stages of development.

Finally I would like to add that the managers of Leister Celtic F.C and Broadford Rovers F.C have been an excellent aid in evaluating this mobile application and have added opinions and insights as to how they felt about the application and different functionality it demonstrates. I would like to thank them for taking the time out to evaluate this application and in there many suggestions for further development.
Contents

Abstract ................................................................................................................................................... ii

Acknowledgements ....................................................................................................................................... iii

List of Figures ............................................................................................................................................... 1

Introduction ................................................................................................................................................... 2
  1.1 Aims and objectives ................................................................................................................................. 2
  1.2 Motivation .................................................................................................................................................. 3
  1.3 Being new to Android ............................................................................................................................... 4
  1.4 Document Structure ................................................................................................................................... 5

Background ................................................................................................................................................... 6
  2.1 Existing Systems of Inspiration .................................................................................................................. 6
    2.1.1 FIFA 2014 .......................................................................................................................................... 6
    2.1.2 American football manager 2012 ......................................................................................................... 8
  2.2 Requirements gathering ............................................................................................................................... 9
    2.2.1 Initial Ideas ......................................................................................................................................... 9
    2.2.2 Short listing requirements ................................................................................................................... 10
  2.3 Installation and setup for development ....................................................................................................... 11
    2.3.1 Installation of Eclipse .......................................................................................................................... 11
    2.3.2 Installing the Android SDK (software development kit) .................................................................... 11
    2.3.3 Integrating the Android ADT plugin for Eclipse ................................................................................. 12
    2.3.4 Configuring the ADT plugin for Eclipse ............................................................................................... 12
    2.3.5 Optimising the Android Emulator ....................................................................................................... 13
Design.......................................................................................................................... 15

3.1 Problem Overview ....................................................................................................... 15
  3.1.1 Storing Player Coordinates and Player identification ............................................ 15
  3.1.2 Tactic Creation ...................................................................................................... 16
  3.1.3 Google Calendar Integration.................................................................................. 16
  3.1.4 Personal Contacts Integration ............................................................................. 16
  3.1.5 Problems Summarised .......................................................................................... 17

3.2 Conceptual Design of Application ................................................................................ 17
  3.2.1 Design Flow Chart .............................................................................................. 18
  3.2.2 FluidUI mock-ups .............................................................................................. 19

3.3 Initial Conceptual Design of Squad Screen ................................................................. 20
  3.3.1 Storing Player Coordinates and Identifying Players .............................................. 22
  3.3.2 User Interface Design .......................................................................................... 23

3.4 Initial Conceptual Design of Tactics Screen ............................................................... 25
  3.4.1 Original Concept Design ...................................................................................... 25
  3.4.2 Player Animation and the Complete Design ........................................................ 26

3.5 Google Calendar and Player Contacts Integration ....................................................... 27
  3.5.1 The Initial Calendar Design Mock-Up ................................................................. 27
  3.5.2 Design of Player Contacting Integration ............................................................. 28
  3.5.3 The final Design of Calendar and Contacts ........................................................ 28

3.6 About Screen Design .................................................................................................. 29

3.7 Design Summary .......................................................................................................... 30

Implementation .................................................................................................................. 32

4.1 Creating the skeleton of the application ...................................................................... 32
  4.1.1 Creating an Android Activity ............................................................................... 33

4.2 Implementing the Squad Screen .................................................................................. 34
4.2.1 Player Creation........................................................................................................35
4.2.2 The setDefaultSquad() method............................................................................35
4.2.3 Storing and Reading Coordinates of Players and their Identification ..............37
4.2.4 Integrating Save and Reset Buttons......................................................................39
4.2.5 Dragging Players to New Positions ........................................................................39
4.2.6 The Resulting Squad Screen Completed..............................................................39
4.3 Implementing the Tactic generation screen..............................................................40
4.3.1 Loading the latest saved Formation ......................................................................41
4.3.2 Register Player Clicks and Screen Clicks ..............................................................42
4.3.3 Arrow Creation and Trigonometric Rotation.........................................................42
4.3.4 Player Translation Animation ...............................................................................43
4.4 Implementing a Calendar for Fixtures .......................................................................43
4.5 Implementing Player Contact functionality ..............................................................45
4.6 Resulting Implementation .........................................................................................46

Evaluation..............................................................................................................................48
5.1 The SUS approach to Survey Design.................................................................48
5.2 Broadford Rovers F.C..............................................................................................49
  5.2.1 Player Identification with Initials or full names..................................................50
  5.2.2 Moveable Goal Keeper .......................................................................................50
  5.2.3 Corner / free-kick feature ..................................................................................50
  5.2.4 Survey Results ..................................................................................................51
5.3 Leister Celtic F.C........................................................................................................53
  5.3.1 Half Pitch Zoomed Feature ...............................................................................53
  5.3.2 Multi Formation Saves .....................................................................................53
  5.3.3 Passing and Receiving Functionality .................................................................54
List of Figures

1.2 The complicated whiteboard system currently in place for managers
2.1.1 FIFA 2014 squad management screen example
2.1.2 American Football Manager 2012 example
2.3.3 Integrating the ADT plugin for Eclipse
2.3.5.1 Android SDK manager icon example
2.3.5.2 HAXM Emulator Accelerator Installer

3.1.5 Use case scenario diagram example
3.2.1 Flow chart of application design
3.2.2 FluidUI Screen Mock-up Diagram
3.3 Sky Sports Match Analysis Example
3.3.1 Default 4-4-2 Formation and conceptual design of Squad screen
3.3.1.1 Sending Squad Formation Data Between two Activities
3.3.1.2 File writing Design Approach to Storing Player Coordinates
3.3.2 Squad Screen Design Interpretation
3.4.1 Initial Tactic screen Design
3.4.2 Sequence Diagram of actions to create a Tactical Player Animation
3.5.1 Initial Fixtures Calendar Design
3.5.2 Player Contacts Screen initial Design
3.5.3 Calendar and Contacts Integration with new Processes
3.6 The Instructions Screen Design

4.1.1.2 Splash Screen XML Layout Example
4.1.1.3 Android Manifest Activity Declaration
4.2.2 Set Default Squad Formation Method
4.2.3 Storing Player Coordinates by the WriteToFile method
4.2.3.2 Reading Player Coordinates From File
4.2.6 Squad Screen Result
4.2.6.1 Moving a Player
4.3.1 Loading Saved Formation Tactic Screen
4.3.3 Arrow Creation and Trigonometric Rotation
4.3.4 Multiple Player Animations
4.4.1 Calendar Activity Code
4.4.2 Google Calendar integration
4.6.1 Original Conceptual Design of the application
4.6.2 Resulting Screens of application
Chapter 1

Introduction

This chapter introduces the reader to the objectives set out for the duration of the project and provides a detailed chapter outline for the different sections contained within the report. Motivation for the development of such a project will be discussed as well as a background on the author in terms of Android development and motivations to develop in an area such as this.

1.1 Aims and objectives

The initial aims of this project were to have two core pieces of functionality which were considered most beneficial to the end user (a football manager) in terms of the value added to their managing duties. First and foremost was the idea of allowing the manager to be able to manipulate representations of players around the screen into different positions to produce a customisable formation for their squad. The second idea was enabling the user to create their own customisable tactics involving player animations to make a clear representation of player movement during a sequence of play. These were considered the core objectives of this project and were decided and set in stone as the ultimate requirements of this mobile application. They were named as follows:

1) Squad formation management
2) Tactic creation centre

It was not until later in the project after reaching these goals four weeks ahead of schedule with a substantial amount of development time remaining that it was decided that another two requirements were added to the list of objectives which encapsulated other functionalities which a manager could benefit from. These were:

3) Calendar integration – (to manage fixtures and events)
4) Player messaging – (to contact multiple players)
Later in the chapter entitled ‘background’ the initial ideas for development will be discussed and how these objectives were screened to arrive to the above list of requirements.

1.2 Motivation

Most football teams currently use a whiteboard system as a traditional artifact for displaying the squad formation or demonstrating tactics for offense or defence to members of the team. Managers become increasingly frustrated through using this current system as it may not always be a clear viable solution for explaining the sequence of a play. Most explanations only scratch the surface in terms of what a manager is trying to portray through a whiteboard and the train of thought is often lost as players focus on their own positional stance and not the team as a whole. It is important for the non-football minded reader to contemplate the complicated scenario of such a white board explanation. Each time the manager wants to display a player moving into a new position, they have to correlate this on the board by multiple variations of drawing arrows, erasing players, rewriting players and using dotted lines. This quickly becomes exponentially difficult to understand for every player that is explained through patterns of movement on the board (as seen in figure 1.2 below).

![Figure 1.2: The complicated whiteboard system currently in place for managers](www.shutterstock.com · 114681655)
It was a major motivation to conceptualise and visualise a solution which could aid all processes currently found difficult by a football manager. Inspirations for this project will also be drawn on in the ‘Background’ chapter. Most of these inspirations mentioned are very well credited for what they deliver and the target area they aimed for, but none of which currently deliver a solution in high quality visualisation or directed at this area of the market.

Another strong motivation for developing a project in an area such as this was the opportunity to work in the mobile applications development area. The market for mobile applications has expanded greatly over the past number of years and although the market may be saturated with applications it still seems like an excellent opportunity for a programmer or private development opportunities. The motivation around this area of development is in the shear amount of people that currently use mobile applications: 1.2 billion People were using mobile applications worldwide at the end of 2012 [1] and currently there are 1,195,932 android applications available on the Google play store [2].

1.3 Being new to Android

I was new to Android development before the commencement of this project. In the past I was involved in the development of an Android application but only acquired experience working on back-end development - interaction with a server that involved pushing and pulling of information between the server and the application. This being the case, experience was not obtained with direct programming of the application itself as responsibilities for backend development were:

- mySQL – an SQL database management system [3]
- PHP – A hypertext scripting language, used to query the database [4]

One of my personal goals was to work on mobile application development and more specifically graphical animation through these applications to create user friendly experiences.
1.4 Document Structure
The chapters of this report are broken down and delivered in a manor to provide clarity and coherence to the reader. The following chapters proceed in the order:

- Background – describing initial existing systems which have inspired the development of this application and initial ideas for the project

- Design – describes the methodology for how the application was designed from concept including the components required and diagrammatic breakdown of the inner workings of the application

- Implementation – How all the functionality was fitted together and how they were implemented on a technical level

- Evaluation – the SUS approach to survey design and the outside opinions gathered from local football clubs

- Conclusion – The overall experience gained throughout the project and final remarks on its development

The following report outlines in great detail the planning, organisation and development of this project from the idea concept generation stage through its incubation period and the developmental process. The report captures the approaches taken and decisions made to deliver the application in a quality driven manor.
Chapter 2

Background

The purpose of the chapter is to bring to light existing systems which have been of inspiration to the development of this mobile application through discussing their strengths and weaknesses. Also discussed are the requirements gathering process for development which outlines brainstorming of initial ideas and the screening process used to narrow down core requirements. Furthermore the topic of what was required to develop this project and the installation of these tools and features are discussed in great detail.

2.1 Existing Systems of Inspiration

Some of the following systems were never intended for commercial use by football managers but are a direct inspiration in the development of this project. Other systems which have attempted to make football manager duties more efficient were seen as lacking in quality. Some of these systems will now be discussed weighing up their pros and cons in terms of the system which was aimed to be delivered from ‘Aims and objectives’ in the introduction chapter.

2.1.1 FIFA 2014

EA Sports produce an array of well known games all over the world, one of which is entitled ‘FIFA’. It is an interactive football game which mimics real life football teams, with latest updates of players and high quality graphical depictions of real athletes and stadiums [5]. Within FIFA there is also the opportunity to manage your own squad in career mode. The user interface of the squad management screen contained in FIFA was a great inspiration for this project as it is extremely user friendly and is graphically state of the art. In game functionalities allow a user to create their own formations, manipulate players in different
positions and create customised tactics or use pre set default tactics. (See figure 2.1.1 below)

![FIFA 2014 squad management screen example](image)

**FIGURE 2.1.1** FIFA 2014 squad management screen example

The features this game provides are top quality visualisations not only in game play but in these specific management screens which this project is more concerned with. Fictionally the game can allow a user to play as any team in a league and provides the user with total control over that team whereby they have to try to win numerous trophies. No application was built with the same squad management quality benchmark in mind or near to the same graphical depiction of formations. It was such an inspiration for this project considering the detail to squad management but still lacks the graphical animation of players to display a sequence of a play in tactic creation.
2.1.2 American football manager 2012

This application is a widely used system for American Football management and portrays a great deal of functionality that is an inspiration for the conceptualising of how this mobile application was imagined to be and functionality it aimed to deliver [6]. The flaws found with this application are that the graphics are believed to be produced at an unprofessional level. It does not act as a clear solution; rather it replaces the current whiteboard protocol with a technological version which carries out the functionality of a virtual version. It does not particularly add value to squad management as it is an unclear solution and does not provide player animations which are considered a highly important goal of this project. (See Figure 2.1.2 below)

![American Football Manager 2012 example](image)

FIGURE 2.1.2 American Football Manager 2012 example

The application does however deliver basic functionality such as movement of players into different positions for formation management and delivers an element to tactic creation through the addition of arrows on screen. It is believed that without animation an
application such as this will certainly lose its core value proposition as acting as a solution to tactic generation.

This application caters for numerous other sports which may be seen as depriving it of value as it provides a “one fits all solution”. Certainly valuable information can be taken from this application in terms of aesthetics and propositions it makes and it is important to take these into account when designing an optimum solution.

2.2 Requirements gathering

From the early stages of this project there were many ideas documented as a result of brainstorming. In the scope of this project it was a difficult task in short listing the most important pieces of functionality as a ‘must have’ in the finished solution. The following contains an introduction to initial ideas and concepts thought up in the idea generation phase of the project and moves on to explain how these ideas were screened to arrive at a permanent list of requirements.

2.2.1 Initial Ideas

The following ideas were initially recorded as possibilities to enter into the list of requirements for the duration of the project. These features all revolve around the area of squad management and aim to increase efficiency in a manager’s duty driving the team towards success:

- Display of squad on screen in formation with options to move players around and save players in updated positions
- Create tactics through player animation displaying different routes they should take in different sequences of play
- Store all player details on the device and provide access to these details on user touching a player on screen
- Option to add a photo to a player on screen with link to camera on device
- Log injuries of players and their durations
- Ranking features of determining overall player performance
- Sandbox area for managers to freely edit any formations or tactics with no detrimental effects unless saved
- Attendance functionality and integration with another system where notifications can be sent to players and they can then accept or reject from their side of the system providing manager with a head count of attendees.

The above ideas were brainstormed during the early stages of the project before development began and it was clear given the scope of the project and the timeline in place that a number of these had to be shortlisted as ideas which were more important than others. This short listing process was based around judgements of which tasks a football manager had which were considered more important than others.

2.2.2 Short listing requirements

Ideas initially brainstormed were arbitrary and endless to imagination around the thought of what the duties of a football manager entail. There are many duties a manager has regarding player subscriptions and other off the pitch duties such as notifications mentioned above. At this stage of the project these different ideas had to be screened by weighing up the pros and cons to decide which should be a part of the resulting solution given the amount of time available. The concluding functionality perceived to be most important were:

- Squad formation customisability
- Tactic creation through player animation

It was not until late stages in the project when development was almost complete that another two further requirements were added due to completion of the above first two objectives before the estimated delivery time:

- Google Calendar integration
- player contacting integration
These were considered the next top key pieces of functionality which would ultimately make the duties of a manager much easier around organisation of events and contacting members of the team.

2.3 Installation and setup for development

This subsection contains all the necessary steps to install relevant components which all brought together aided the implementation of this project and are basic requirements for Android development. The following steps were undertaken and will be explained in detail below:

1. Installation of the Eclipse Integrated Development Environment (IDE)
2. Installation and unpacking the Android Software Development Kit (SDK)
3. Integration of Android Development toolkit plugin (ADT)
4. (optional) optimisation of the Android Emulator – (but highly recommended)

2.3.1 Installation of Eclipse

Eclipse is an Integrated Development Environment (IDE) for programmers developing applications primarily in Java but also has numerous plugins for developing and accessing different libraries [7]. It is considered the fundamental (IDE) for developing in Android as Android classes are compiled from Java byte code and converted to Dalvik files [8]. Eclipse also has a plugin available to develop Android applications known as the (ADT) plugin (Android development toolkit) this provides access to classes and implementations contained within a downloaded Android SDK (software development kit).

2.3.2 Installing the Android SDK (software development kit)

The Android SDK is:

“A kit that enables developers to create applications for the Android platform. The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications”.[9]
These are the fundamental tools for Android development and so they must be downloaded and unpacked to a directory on your machine. The ADT plugin we will later see requires a reference to where the Android SDK is unpacked on your computer. The SDK is unpacked as an executable file and when run checks your computer for the correct Java Development kit (JDK) and installs it if necessary. Finally the installer saves the Android SDK to a default location or one preferred by the user. The URL of the location of the SDK is required for the configuration of the ADT plugin so it should be stored in a known, easily accessed location. A step by step guide is also referenced in this document for installing the android SDK [10]

2.3.3 Integrating the Android ADT plugin for Eclipse

To integrate the ADT plugin for Eclipse one must simply navigate their way in Eclipse to the help tab in eclipse and then select “new software”. A window then opens allowing you to copy the URL of the new software you want to install. Here the following URL must be placed:

https://dl-ssl.google.com/android/eclipse/

FIGURE 2.3.3 Integrating the ADT plugin for Eclipse

The user must then click ok and follow the necessary steps for installation of the ADT plugin. A step by step guide is also included in the references section of this document for installing and integrating the ADT plugin [11].

2.3.4 Configuring the ADT plugin for Eclipse

Once the Eclipse IDE restarts after installing the ADT plugin it will then prompt the user to configure the plugin. In the available text field within the dialogue box, the location URL of the Android SDK should be input or the user can browse the file system and locate the Android SDK manually. Once next is clicked the Environment is now set up to develop Android applications.
2.3.5 Optimising the Android Emulator

This step is not required for Android development, however it is certainly recommended as it speeds up compilation and emulation start up time. Intel has a step by step guide to optimising the Android Emulator [12] and a pre-requisite to completing this step is having installed the Android SDK previously.

To begin one must simply open the Android SDK manager. This can be done from within Eclipse by clicking the android icon below which is located in the toolbar of Eclipse.

![Android SDK manager icon example](image1.png)

Otherwise you can search for “SDK MANAGER” on your computer and open that. The following appears:

![HAXM Emulator Accelerator Installer](image2.png)
You must then tick the box within the “Extras” folder above for “Intel x86 Emulator Accelerator HAXM (Hardware Accelerated Execution Manager)”. Although it will say “installed” when this action is complete the task has only copied the (.exe) file for installing (HAXM) to your computer and you must now locate and run this file to complete this step for optimising the Android Emulator.
Chapter 3

Design

This chapter attempts to outline the conceptual design of all features delivered in this project. It also introduces the different software technologies used which was used to deliver these features.

3.1 Problem Overview

This section provides the reader with an overview of the approach taken to deliver this mobile application. The break-down of the design of the application is as follows:

1. Player coordinates to stored in memory and unique identification for these players
2. New Tactic creation using arrows rotated to the correct direction and player animation
3. Google Calendar integration to allow fixtures/events to be added to the Calendar
4. Personal contacts integration to allow multi-messaging to players

3.1.1 Storing Player Coordinates and Player identification

Player coordinates are stored to file in memory by first collecting every individual player coordinates directly from the player, then writing all of this information to a file stored in memory. Player coordinates are only written to this file in two separate cases. The first case is where the user wants to save their current formation after they have manipulated their team into new positions and the second is when they want to reset their formation to the default stored in memory which is a standard 4-4-2 formation.

Another important aspect to creation of players is identifiers which would help the user tell the difference between players on screen. This idea was initially thought to be something such as numbers or abbreviated characters.
3.1.2  Tactic Creation
When a new Tactic is supposed to be created, the idea here is that the last saved squad formation will be loaded and it will then be used as the current formation for tactic creation. Player coordinates are taken from memory and each player is assigned to their position on screen. The functionality of this part of the application requires the user to touch an outfield player (not including the goalkeeper as he should never leave his goal) and then touch somewhere else on screen. An arrow then appears on screen where the user has clicked and is rotated at the correct angle based on the start point of the players’ location and the end point where the user has clicked. The player then animates repeatedly to and from their start location and their end location indicating the motion of running during a particular play.

3.1.3  Google Calendar Integration
This opens the local Google Calendar application of the device which starts as a new Android Activity. The design of this calendar functionality uses content providers as will be talked about in the technologies used section below. The content provider allows data interchange between this Android application and the Google Calendar application on the device. Once the Calendar process begins the user can then add and delete events from their Google Calendar.

3.1.4  Personal Contacts Integration
This feature works similarly to the Google Calendar integration but instead of using a content provider a specific new messaging intention is started. Through this Activity already well known to users they can then begin to add new contacts to the recipients list and type a body message to be sent to all recipients.
3.1.5 Problems Summarised

The design of the above problems are summarised below in the UML use case diagram for the football manager as an actor in this football management application system. (See Figure 3.1.5)

![Use case scenario diagram example](image)

FIGURE 3.1.5 Use case scenario diagram example

3.2 Conceptual Design of Application

This section outlines what the initial conceptual design of this application prior to the development stage. The conceptual design of the application are comprised of images below of the design flow chart which displays the flow of activity of the application from opening to close through and a number of user interface (UI) mock-up diagrams of what the screens were initially intended to look like.

The conceptual Design of the application was realised on paper at first and then recreated through the use of FuidUI - a mobile prototype workflow creation tool [13] and Gliffy – a web based diagram editor [14].
3.2.1 Design Flow Chart

The Design flow chart which can be seen below describes the initial understanding of what was being developed. The diagram begins with a splash screen on the applications start-up which is simply a loading screen. After the splash screen is the main Menu. Within the Main Menu would be the list of options to the user incorporating a conventional layout of other systems which would emerge the user in a world of familiarity and ease them into understanding how the application works. (See Figure 1.2.1 below)

![Flow chart of application design](image)

Upon entering the Menu and then choosing the Squad screen, the user should be then provided with a screen containing their current Squad. This Squad should be moveable – meaning allowing the user to move a player by a dragging motion. The options to save and update the current formation were a foreseeable piece of functionality to have on this screen.

Going back to the main Menu – if the user clicked on Tactics, this would then bring them to another screen which looked similar to the Squad screen. It was anticipated in designing the application that this screen must use the latest saved squad formation from the squad
management screen; otherwise it was meaningless for the user to have a different formation which they were going to create Tactics with. In the Tactics screen the ultimate benefit to the user here was that they could click on a player and then click somewhere else on screen – the player would then animate to and from the point clicked on screen in a repetitive motion indicating where they are suppose to be at a given time. Similar sequences of events were idealised for the Fixtures screen which would open a new Google Calendar Activity or in player contacts a new messaging window would be started. Mock-ups were also created in fluidUI which materialised the aesthetics to the app and gave a great insight as to how it would ultimately be with the different activities linked together.

3.2.2 FluidUI mock-ups

The following is a fluidUI mock-up diagram of the different screens conceptualised during the project planning stage and was roughly sketched to paper before development began. Each screen was developed using fluidUI and then the following diagram of design was then completed using Gliffy. (See Figure 3.2.2)
3.3 Initial Conceptual Design of Squad Screen

Designing the Squad screen was a natural process considering all of the influences in the real world today. If one is watching a premier league match of two teams, half-time analysis or post match analysis always demonstrate different features of functionality that this application aimed to deliver. These examples of functionality – all serve a different purpose when seen demonstrated by match analysts as they are moving players on screen for different purposes – to show the match viewers were a squad player was during a particular play. The initial Design of the Squad screen was to work off the premise of moving players around the screen but in this case to serve a managers unnoticed need for formation management through a system such as this. (See Figure 3.3 below)

![Sky Sports Match Analysis Example](image)

**FIGURE 3.3**  Sky Sports Match Analysis Example

The formation layout idea was imagined as beginning with a 4-4-2 default formation for the user on entry to the Squad management screen for the first time. (See Figure 3.2.2 below)
The reason for this is that every football manager is familiar with this formation as it is considered the industry standard. The application was thought out to be designed as a system which allowed changes to the formation but once these changes were saved a record of all player positions must then be recorded so that in the case of exiting the screen, on re-entry the manager could then view their last saved formation. In the case where no changes were saved then there was no need to update the player’s positions in memory and exit the screen as usual.

FIGURE 3.3.1  Default 4-4-2 Formation and conceptual design of Squad screen
3.3.1 Storing Player Coordinates and Identifying Players

Research for this design was carried out and it was prominent that two valid options were available for storage of player locations. Player positions could either be recorded in a file in memory – which would then entail providing a reading and writing to the file using rules and orderings so that data could be read properly free from errors. The second option was to somehow pass data between the Squad Activity and the Tactic Activity. The second option seemed less feasible for this situation as if a crash occurred and no information about the formation of the squad was received it would be difficult to manage the case where it was required to roll back to the last successful save.

The first case had a stronger argument in that if something went wrong at one stage in the application, provided the last save was successfully committed, then even if the application crashed and reopened the file would still be available in memory for the last saved formation. This was the chosen approach for storing player coordinates. Another way of saving the formation considered was writing to the file every time the user successfully moved a player. This was soon deemed inefficient and too strenuous for the application because of multiple File access and writing to the file these of which are considered

![Diagram](image-url)
relatively expensive operations to execute. (See the File writing approach below – Figure 3.3.5)

FIGURE 3.3.1.2 File writing Design Approach to Storing Player Coordinates

It was also important not only to store these players but to have a unique way of identifying individual players. It was conceptualised that each player should have a unique identifier associated with them to display on screen for the user, allowing football managers to tell the difference between players.

3.3.2 User Interface Design

In Terms of User interface layout for this screen, the most optimal solution was to have as minimal buttons as possible available to the user. The reason for this being to keep everything visible clear on what the user was able to do and to reduce complexity and increase value provided to the user experience. The screen was aimed to basically just be the team displayed on screen with the addition of two buttons to reset or save the formation. Other than these options what else was required for the user? Button placement had not been decided at this stage of designing the screen but player placement and
everything else required to move players around was conceptualised as something like the following screen:

![Squad screen](image)

**FIGURE 3.3.2 Squad Screen Design Interpretation**

As we seen from this above depiction of how the Squad screen should look, it is visibly clear and reduces any form of complication for the user- (keeping in mind that the majority of Irish football managers were born long before the introduction of smart technology). The colour scheme for background was decided as green and player colours to be a strong dark colour temporarily set to RED. This design was the fundamental design followed for the duration of the project.

After choosing the file writing approach to storing player coordinates it was decided that the file should be stored in the src directory of the project making it readily available to the Tactics Activity so it could easily implement and display the latest formation. On opening of the squad screen activity the design was decided that a read from the file stored in memory would occur re-initialising all players with the latest coordinates from the file. In the case where no file existed, one would be created and the default formation (4-4-2) would be written to it for future instances. This therefore required a hardcoded default formation to be embedded in the application. This default formation would also handle the case where an error occurred such as a crash, there would be a backup formation which could be easily accessed from the src directory.
3.4 Initial Conceptual Design of Tactics Screen

The Tactics screen was designed as the ideal Activity which a manager would require which basically allowed them to create new tactic plays which would encapsulate player movement during a particular turn of play. It was understood immediately that this screen required the latest saved formation but the design for the squad screen above carried out most of the work in building this formation and it was just a matter of reading the latest formation.

3.4.1 Original Concept Design

In the early stages of idea generation it was thought that allowing the manager to draw arrows on screen would be the approach to take with tactic creation, however this just replaced the existing complex whiteboard approach with a virtual complex solution. This would not aid the manager; it only creates a new way of carrying out the existing task of tactic creation at the same level. (See Figure 3.4.1 early Tactic design below)

![Figure 3.4.1 Initial Tactic screen Design](image)
3.4.2 Player Animation and the Complete Design

The screen was altered to incorporate player animations through a combination of using arrows to indicate direction of player movement by arrow rotation and repeated translation animations to indicate where a player is running to and from. The screen was then altered with design of these ideas in mind. (See Figure 3.4.2 below)

![Sequence Diagram of actions to create a Tactical Player Animation](image)

**FIGURE 3.4.2** Sequence Diagram of actions to create a Tactical Player Animation

All players on this screen were different from the Squad screen in the sense that the user could not drag and update their positions. This was to separate functionality between the screens and keep one screen for doing one task and the other for looking after Tactic creation alone. The concept of creating a new tactic is seen in the figure above which demonstrates the sequence of actions required to implement a tactical animation. These actions are as follows:

1) Click a player to animate
2) Click the point to animate to and from on screen
3) Information about start and end points are recorded for arrow rotation
4) Arrow is added to the screen with correct rotated direction
5) Player animation is registered and begins animating

Player animations was considered the ultimate value in the application being that if it was executed to a high quality standard it would provide excellent clarification to the user and would rule out the requirement for a whiteboard during training sessions or half-time intervals. To have multiple players animating at the same time a system was required to be put in space which will be later seen in the chapter entitled ‘Implementation’. This was an entity which contains all registered animations and allowed for new creations to be added to the list.

3.5 Google Calendar and Player Contacts Integration

Originally Calendar functionality was conceptualised as a basic Calendar which would allow football managers to add new events which would represent training sessions for the team or upcoming match events. This would provide managers with the extra benefit of having a versatile application which also managed the organisation of off the pitch activities such as the above mentioned.

3.5.1 The Initial Calendar Design Mock-Up

The initial design for the fixtures screen of the application was designed as any regular calendar representing days with events on and assumed functionality of clicking an empty slot to add a new event. (See Figure 3.5.1 below)

![Fixtures screen](image)

**FIGURE 3.5.1 Initial Calendar Design**

As stated previously in a chapter entitled ‘background’ this calendar functionality was not a core requirement of the delivery of the project. It was only until a number of weeks before
the code freeze date that it was decided to be an included feature. The most optimum way of delivering calendar integration would be to have this football management application start a new activity of the locally stored Google Calendar application.

3.5.2 Design of Player Contacting Integration
Player contacts integration was also seen as a similar design as the calendar integration mentioned above. This was the second functional requirement mentioned in the ‘background’ chapter of this report which was added as a requirement before the code freeze due to an overestimation of time to deliver the two pieces of core functionality of squad movement and tactic creation. The initial design of the player contact screen was seen as the conventional messaging screen of any Android device of today. (See Figure 3.5.2 below)

3.5.3 The final Design of Calendar and Contacts
Delivering the two pieces of functionality which were added as late requirements to this project as new creations were considered astronomical tasks each in their own merit.
Integrating them however with existing functionalities already existing on the device would solve the problems which these objectives aimed to achieve. The resulting design for both integrations was a solution which would start new activities of the existing Google Calendar application, and the second was a new activity of the local messaging service. Both of these processes are external to the application and a diagrammatic form of these processes is captured below. (See Figure 3.5.3 below).

![Diagram of Calendar and Contacts Integration with new Processes](image)

**FIGURE 3.5.3** Calendar and Contacts Integration with new Processes

This design was followed through the delivery of both of these newly added requirements to be delivered and they are explained in detail in the chapter entitled ‘Implementation’

### 3.6 About Screen Design

The about screen for the application which was later renamed as the ‘Instructions’ screen is a simple screen, one which provides the end user with instructions of how each area within the application works. It entails a simplistic design one of which displays just text and should allow the screen to scroll up and down in the case of a device with not enough pixels to display all the textual information. (See Figure 3.6 below).
As seen above, it is a straight forward screen displaying all relevant textual information for the end user. This design approach was followed for the implementation of this screen and will be discussed further in the chapter entitled ‘Implementation’.

3.7 Design Summary
To summarise the overall design of this application the following list of development requirements had to be met:

squad screen

1. Storing of player coordinates to a file and identification of players for the user
2. Updating these coordinates through save and reset
3. Reading these coordinates stored in the file to load latest squad formation

Tactic screen

4. Register player initial and final positions
5. Use these to determine arrow rotation direction
6. Place arrow on screen in end location
7. Animate player to and from this position
Fixtures

8. Launch Google Calendar Activity

Player Contacts

9. Launch local messaging activity

About Screen

10. Create textual instructions to be displayed for end user

All of these above requirements for the individual screens are discussed in detail in the chapter entitled ‘Implementation’ and described in great detail how they were delivered.
Chapter 4

Implementation

This section aims to describe how design of the application was implemented and the different process undertook to deliver the functionality of the application. It is compiled into a number of different subsections:

1. Creation of base skeleton of the application
2. Implementation of the Squad screen
3. Implementation of the Tactic screen
4. Google Calendar Integration
5. Player Contacts Integration

Each of these topics will be described in great detail of how they were implemented and the different technologies used to build and deliver these requirements.

4.1 Creating the skeleton of the application

The skeleton of an application is the structure by which the initial screens are created and their ability to switch between one another. It was created from the flow chart design contained within the chapter entitled ‘Design’ of this report and attempted to mimic the basics of what was contained in the fluidUI scenarios also contained in that chapter. In order to build a basic structure for the application an understanding of the basics of Android was required. This was researched to understand how Activities were created and what they were made up of (an Activity is essentially an Android screen). For this an excellent video to aid in understanding the fundamentals of Android was sourced to research and understand [15]. This video explains in great detail the course of creating a new Android application and the importance resources used in development.
4.1.1 Creating an Android Activity

In order to create an Android Activity one must extend the class Activity provided by the Android SDK.

“Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView(View)”[16]

An Activity is a Java file which extends the Activity and has a number of different resources linked with to be completed:

- Java implementation of the Activity – the onCreate() method
- XML layout file which it accesses
- Android Manifest reference

4.1.1.1 onCreate()

The onCreate() method is a method which is inherited from the Activity class after extending it with a customised subclass. This method initialises the Activity and conventionally sets the current window to one previously defined in memory as an XML layout file. The XML layout file defines everything which is supposed to be displayed in a given window and is set using the method: setContentView(int). It takes an id reference to the XML layout in the form of an (int value) and then sets the current screen to the passed layout. The onCreate() method also initialises any other elements within the Activity and can be used to call helper methods within the class which will carry out any tasks based on user input or touches on screen.

4.1.1.2 XML LAYOUT

An XML layout in Android defines everything which will be visible on screen for an Activity. The most basic XML file within this application and is an excellent example of how XML layouts work would be the splash screen XML layout. This is a simple layout as it only defines a background image for the screen and a progress loading widget. The splash screen in essence is a loading screen between the application start up and the main menu. (See Figure 4.1.1.2 below)
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/sviva"
    android:orientation="horizontal">
    <ProgressBar
        android:id="@+id/progressBar1"
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"/>
</RelativeLayout>

FIGURE 4.1.1.2 Splash Screen XML Layout Example

### 4.1.1.3 ANDROID MANIFEST

Every application in Android requires an Android Manifest file. Any new Activities created or features which an application uses must declare this in this Manifest file. When creating a new Activity a declaration of the new Activity must be added into the Manifest file. (See Figure 4.1.1.3 below for an example of an activity declared in the manifest file)

```xml
<activity
    android:name="com.example.fyp_footballmanager.SplashScreen"
    android:label="@string/app_name"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

FIGURE 4.1.1.3 Android Manifest Activity Declaration

### 4.2 Implementing the Squad Screen

The approach of this implementation tackles the core functionality outlined within the Design chapter of this report. The individual tasks which had to be solved and collaborated together to work as an effect solution to formation management and player movement are as follows:
1. Player Creation
2. Storing coordinates of players and player identification
3. Integrating save functionality
4. Player Movement

These above challenges are the core makeup of the functionality of what is to be provided as a solution to Football managers in managing the dynamic formations of a Football team and are talked about further below.

4.2.1 Player Creation

In implementing a solution to create visible players on screen the approach taken was to add images to create general functionality of player creation and addition to the screen. In doing so a new XML layout was created to encapsulate the contents of what would be displayed in the Squad management screen entitled “Squad”. Player creation and addition to this XML was foreseen as Android Image Buttons which are a class provided by Android used in the creation of Buttons with backgrounds set to a user chosen image:

“Displays a button with an image (instead of text) that can be pressed or clicked by the user” [17]

Later it was discovered in order to display identities associated with players they were adopted to be Android Buttons to setText() of the Button on screen. The Squad.java File carries out all of the operations such as: ability to drag players to a new position, ability to reset the squad to default 4-4-2 formation and finally a save functionality which updates the stored coordinates with latest after changes have been made to player positions.

4.2.2 The setDefaultSquad() method

This method is used in two different cases:

1. There is no existing player formation saved in memory (such as first time opening the application)
2. In the case where the reset functionality is called to return all players to default positions (4-4-2) formation

The setDefaultSquad() method provides hardcoded information for player coordinates in the case where the default formation is called upon. The method is based on simple mathematic calculations for each individual player positions which takes screen height and width into account on any given device. This method makes use of a facilitating class called PlayerCoordinate.java. This class is not complicated and is only used to create objects representing ‘X’ and ‘Y’ positions on screen by the use of variables titled: ‘xPos’ and ‘yPos’.

The setDefaultSquad() method creates eleven (number of players on a football team) hardcoded PlayerCoordinate objects and passes these into an array of PlayerCoordinates. This array is then used to collect all X and Y coordinates from the playerCoordinates array within a for loop. After this loop completes a method known as: writeToFile() is called which writes the default formation to memory. (See Figure 4.2.2 below)

```java
public void setDefaultSquad() {
    PlayerCoordinate gk = new PlayerCoordinate((w/2.2),(h/10)* 7.3);
    PlayerCoordinate rb = new PlayerCoordinate((w/10)*7.6,(h/10)*5.5);
    PlayerCoordinate cb = new PlayerCoordinate((w/10)*5.6,(h/10)*6);
    PlayerCoordinate cb2 = new PlayerCoordinate((w/10)*3.6,(h/10)*6);
    PlayerCoordinate lb = new PlayerCoordinate((w/10)*1.6,(h/10)*5.5);
    PlayerCoordinate rm = new PlayerCoordinate((w/10)*7.6,(h/10) * 3.5);
    PlayerCoordinate cm = new PlayerCoordinate((w/10)*5.6,(h/10) * 4);
    PlayerCoordinate cm2 = new PlayerCoordinate((w/10)*3.6,(h/10) * 4);
    PlayerCoordinate lm = new PlayerCoordinate((w/10)*1.6,(h/10) * 3.5);
    PlayerCoordinate rf = new PlayerCoordinate((w/10)*5.6,(h/10) *1.8);
    PlayerCoordinate lf = new PlayerCoordinate((w/10)*3.6,(h/10) *1.8);

    PlayerCoordinate[] playerPositions = {gk,rb,cb,rb2,lb,rm,cm,cm2,lm,rf,lf};
    allPositions = playerPositions;

    String squadData = "";
    for(int i=0; i<allPositions.length; i++) {
        squadData += allPositions[i].xPos + " " + allPositions[i].yPos + ",";
    }
    writeToFile(squadData);
}
```

FIGURE 4.2.2  Set Default Squad Formation Method
As seen from the above code snippet it is a straight forward method which creates 11 new PlayerCoordinate objects each entitled with the abbreviations of player positions such as:

“gk” = goal keeper, “cb” = centre back, “rm” = right midfield, “rf” = right forward.

The calculations passed into each new PlayerCoordinate constructor is based on the mathematical division of the screen. This concept entails the division of the screen into a 10 x 10 grid in which case player positions can be worked out.

Once the array ‘allPositions’ is set to all of these individual player coordinates, the for loop can then iterate over each player coordinate and concatenate the ‘xPos’ and ‘yPos’ values of each coordinate to a String result entitled ‘squadData’. Finally this String is passed into the writeToFile() method which updates the formation in memory, which will be later used for Button addition to the screen representing players.

4.2.3 Storing and Reading Coordinates of Players and their Identification

It was decided for the implementation that player coordinates would be stored in a file within the application as outlined in the ‘Design’ chapter. This idea of writing coordinates to a file revolves around a simple method which takes a string of data as a parameter and then writes this data to a text File entitles: “squad.txt”. (See Figure 4.2.3 below)

```java
private void writeToFile(String data) {
    try {
        OutputStreamWriter outStreamWriter = new OutputStreamWriter(openFileOutput("squad.txt", Context.MODE_PRIVATE));
        outStreamWriter.write(data);
        outStreamWriter.close();
    } catch (IOException e) {
        log.e("Exception", "File write failed: " + e.toString());
    }
}
```

FIGURE 4.2.3 Storing Player Coordinates by the WriteToFile method

As seen from the above snippet an outputStreamWriter is used to write the data passed in to an opened File: “Squad.txt”. It is a simple method but core to the functionality of the objectives of the project in the sense of storing player coordinate information.
In the case of reading player Coordinates another method was required to be implemented to collect information from the text file stored in memory: “squad.txt” to read all of the player coordinate information regarding X and Y coordinates on screen. This method required opening the, reading and concatenating the file line by line to a result String and then returning this String of information for the use of initialising player positions on screen. (See Figure 4.2.3.2 below)

```java
/*reads from the squad text file of player positions*/
private String readFromFile() {
    String ret = "";
    try {
        InputStream inputStream = openFileInput("squad.txt");

        if ( inputStream != null ) {
            InputStreamReader inputStreamReader = new InputStreamReader(inputStream);
            BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
            String receiveString = "";
            StringBuilder stringBuilder = new StringBuilder();

            while ((receiveString = bufferedReader.readLine()) != null ) {
                stringBuilder.append(receiveString);
            }
            inputStream.close();
            ret = stringBuilder.toString();
        }
    } catch (FileNotFoundException e) {
    } catch (IOException e) {
    }
    return ret;
}
```

FIGURE 4.2.3.2 Reading Player Coordinates From File

One can see from the above code snippet that this method uses a try and catch block to open a connection to the squad.txt File and then uses a while loop to collect all information in the File while there is a line left in the File to read and then returns the String of concatenated information which can be later used to update player Coordinates.

These two methods are the building blocks in providing read an storage functionality for player coordinates and later we will see how once they are integrated with Save and Reset buttons their duties are performed excellently.
Player identification was carried out by setting the Tags associated with each player Button to a predefined tag within a tag list. This way in identifying which player was which on screen, the buttons original tag set was used to setText() of the button on screen.

4.2.4 Integrating Save and Reset Buttons
From the above methods already explained, the task of implementing save and reset formation functionality was a task which only required linking of Buttons been clicked to calling required methods to updating positions to the file in memory.

First of all, both save and reset buttons were entered into the XML layout at the bottom of the squad screen. To complete their functionality necessary methods had to be called.

4.2.5 Dragging Players to New Positions
Dragging players to new positions occurs when the user a drag motion on one of the image buttons representing a player and releases the player in a new position. This ability was implemented by implementing the onTouchListener class which allows the programmer to override the onTouch method provided therein. The concept behind this functionality is:

1. If a player is being dragged, first change drawable resource to selected indicating drag of the player
2. The X and Y coordinates of raw position of touch on screen are collected and while the action of the user is moving on screen the player is updated in accordance with the user finger. (Simulating dragging of the player)
3. When the motion of touch has stopped, the player is left in their new position and their background drawable resource is set to normal

4.2.6 The Resulting Squad Screen Completed
The squad screen when completed delivered all of the necessary requirements as outlined in the chapter entitled ‘Background’ and also delivered them consistently with design patterns outlined in the chapter entitled ‘Design’. (See Figure 4.2.6 and 4.2.6.1 below)
In order to implement the Tactics screen, a number of tasks need to be fulfilled. These were comprised of:

1. Loading the latest saved formation from the squad.txt file
2. Allowing the user to click a player and then the screen to register start and end points for the translation animation
3. Create an arrow at the end point, and rotate it accordingly from the players start point
4. Animate the player repeatedly from start and end points

These are the core requirements for implementing the tactic screen and allowing the user to create new tactical animations for their players.
4.3.1 Loading the latest saved Formation

This functionality had already been delivered for the squad screen as noted above with the method titled: readFromFile(). (Seen above already in Figure 4.2.3.2). This method need only be implemented again in the Tactics screen with no changes required as it reads every coordinate from memory and then all that was required was to iterate over an array of image buttons to provide the initial screen for tactic creation. (See below Figure 4.3.1)

Squad screen updated player  Tactic Screen result

![Squad screen updated player](image1.png)  ![Tactic Screen result](image2.png)

FIGURE 4.3.1 Loading Saved Formation Tactic Screen

The above squad screen on the left demonstrates how once the player is moved and the save button is pressed, the tactic screen reads the latest formation from the squad.txt file in memory.
### 4.3.2 Register Player Clicks and Screen Clicks

In order to tell if a player is ready to be animated the premise must be tested that:

1. A player has been selected
2. Another place on screen has been selected (while a player was selected)

This functionality was implemented using onClickListeners for all of the players on the Tactics screen. Once one of these players were clicked, another method which checks for screen touches then checks to see if a player is currently selected. If there is a player selected then continues with tactic creation.

### 4.3.3 Arrow Creation and Trigonometric Rotation

The point at which the user clicked on screen after selected a player is considered the centre point for the arrow to be added to the screen. The Arrow is of type imageButton which uses as drawable resource to display a red arrow on screen. Once this arrow is created it is then rotated by the rotateArrowToDirection method. (See Figure 4.3.3 below)

**FIGURE 4.3.3 Arrow Creation and Trigonometric Rotation**

From the above example it is seen how once you can tell which quadrant the second touch from the player selected is located, the angle that this second touch makes with the vertical Y-axis the angle that the arrow must be rotated. This is due to the arrow’s initial starting direction points directly down.
4.3.4 Player Translation Animation

To deal with multiple animations an AnimationSet is used. When a new Translation Animation is created it is firstly added to the animationSet which acts as a list of all occurring animations. Once added, the id of the clicked player whom this animation is associated with is then added to an Integer list of ids. Finally the animatePlayers method is called which starts every appropriate player according to the list of ids with their associated animations, which results in the ability of a manager to have multiple animations of players at the same time. (See Figure 4.3.4 below)

![Multiple Player Animations](image)

FIGURE 4.3.4 Multiple Player Animations

4.4 Implementing a Calendar for Fixtures

Integrating Google Calendar functionality into this application was completed through the use of Android content providers. Android provides many different types of content providers one of which is the CalendarContract content provider. The code for creating and accessing Google Calendar can be seen below. (See Figure 4.4.1)
The above code creates a new Gregorian Calendar which when passed to a URI Builder with the current time, a new Intent can then be started passing the URI builder which opens Google Calendar on the device as a new process allowing managers to view their current fixtures or add new events such as training or matches. (See Figure 4.4.2 below)
4.5 Implementing Player Contact functionality

Player Contact functionality is implemented similarly to Calendar integration as talked about previously. The difference between these two processes is that no information is required for player contact integration. A new intent can be started at any time for messaging or other Android Activities that start external processes to an application. Here is how it was implemented for this application. (See Figure 4.5.1 below)

```java
Intent smsIntent = new Intent(Intent.ACTION_SENDTO, Uri.parse("smsto:"));
smsIntent.putExtra("sms_body", "");
startActivity(smsIntent);
```

**FIGURE 4.5.1** Player Messaging Implementation

Basically this starts a new intention which is specified by the above: Intent.ACTION_SENDTO and takes an argument of contacts which is currently blank. The result of starting such an Intent can be seen (Figure 4.5.2 below)

**FIGURE 4.5.2** Player Contacts Screen Result.
4.6 Resulting Implementation

The Resulting implementation is considered a success as it adhered to conceptual designs previously outlined in the chapter entitled ‘Design’. (See Figure 4.6.1 below). The projects solution remained consistent with design outlines and requirements and the comparison result can be seen below (See Figure 4.6.2).

**Original Design FluidUI mock-up screens**

![Original Conceptual Design of the application](image)

**FIGURE 4.6.1 Original Conceptual Design of the application**
FIGURE 4.6.2  Resulting Screens of application

As seen from both diagrams first with the conceptual fluidUI outline of how the application was designed and finishing with the resulting screens, it is clear that the application stuck to its requirements throughout its development and delivered all core functionalities aimed for.
Chapter 5

Evaluation

Organising an Evaluation was seen as a key part of this project. In terms of delivering a Football management application it was important to gather valid data from the right sources. In this case the potential end users were: football managers around Ireland. This chapter of the report describes how the process for collecting data in evaluating the application was carried out and the results which were found.

Local football clubs in the area were contacted and asked for their participation in a research session which involved a survey of questions about the application and also a short discussion afterwards about usability, appearance and application functionality. Two local football teams responded excellently offering up their time for research and evaluation and both clubs are known as:

1. Broadford Rovers F.C
2. Leister Celtic F.C

For the purpose of the evaluation, managers were contacted and dates were set in stone for a meeting point. The approach to the survey design was the SUS approach, and results found will be discussed in this section of the report.

5.1 The SUS approach to Survey Design

The SUS (System Usability Scale) approach was taken in design of the survey of questions to be put to managers:

“The System Usability Scale (SUS) is a lightweight and scientifically validated questionnaire and an effective tool to assess perceived usability” [18]

This approach aims to calculate usability of a system such as an application and uses rating averages to calculate an overall percentage result. Ratings per question on the survey
usually reside between 1-5. The SUS approach was incorporated in this survey but also included other questions and added space for the user to apply comments towards the end. Not every question in the survey provided the opportunity to rate the item discussed between 1-5 as it was also a goal to collect valuable information about detail of the application.

The following sub sections discuss the visits with local football teams: Broadford Rovers and Leister Celtic and discuss the results found about the application.

5.2 Broadford Rovers F.C

Upon arrival and meeting with the manager of Broadford Rovers F.C, a brief account as to what the application does was given to the football manager: Greg Dumbrell of the under 16’s team. The following description was given of the application:

“As a manager you have many different duties when organising your team, I have outlined these as:

1. Arranging your squad in different formations
2. Generating different Tactics and conveying these to the team as clear as possible
3. Organise dates for training and matches
4. Contacting players about Training and matches”

“This application intends to make your job as a manager easier by allowing you to visualise your squad formation, design tactical plays, is integrated with Google Calendar to remind you of important dates and provides the ability to message players in the team”. At this point in time the application was then given to the manager and a walk though explanation was provided from the main menu screen. The manager was then permitted to freely use the app to get a feel for how it performed. The following suggestions and comments were made:

1) Can I Identify players by names or initials?
2) Can you make the goal keeper moveable?
3) Corner/free kick added tactic feature

5.2.1 Player Identification with Initials or full names
The first suggestion made by the manager was that it would be a much more personal application if the manager could identify which players were which by initials or names on screen. Comments Greg made were:

“Can I enter in player names for this screen, or provide their initials so I can tell which player is which”

This he explained creates a clear understanding of what player is moving where during tactic creation or if the formation is changed.

5.2.2 Moveable Goal Keeper
The second suggestion Greg made was that the Goal Keeper should be a moveable player as of the others. He commented: “I would prefer to be able to move the goal keeper into a different location to demonstrate positioning during defending scenarios”. Another point he made about restricting the Goal Keeper in movement: “if you are going to allow the goal keeper to move, you should only restrict him to his box and not allow the user to move him further than that”. This is a great suggestion as it was never anticipated during the development of the application that it would be used for defensive tactics.

5.2.3 Corner / free-kick feature
The manager demonstrated the boards they use for current tactic creation which are mounted on the wall of the football club room. These boards are magnetised and have numerous magnets which represent players. On the other side there is an up close version of the goal which encapsulates corner kicks and free-kick tactics. He commented: “Even the way you have it is fine...but I don’t want to have to move every player to the top of the screen for a corner play”, “If you could adjust the screen to zoom in for a corner and have the full width of the tablet for a corner tactic it would work really well because we always
use the second side of the magnetic boards to demonstrate corner plays” These findings demonstrate new ideas which can be brought forward into development in the future.

5.2.4 Survey Results

1) How well do you naturally understand the App between 1-5?
   score given: 5
   comments:
   “Does exactly what it says, you only explained it to me in less than 20 seconds and I could pick it up and use it”

2) Would you change anything about the App and why?
   comments:
   “I think it would be great to have names appear identifying players, or even initials so I know who is in what position”
   “I would also like to be able to move the goal keeper for positioning purposes”
   “It would be great to have separate instances for tactics where the screen zooms in close on oppositions goal for corners or free-kicks”

3) Rate the App out of 5 in terms of its ease of use
   Score given: 5
   Comments:
   “self explanatory, even most of the older technophobic managers would understand how to work this”

4) Rate the APP out of 5 in terms of the functions it fulfils
   score given: 4
   comments:
   “Makes managing player formations and tactics creation simple. Sure it only takes 2 clicks on the Tactic screen for a player to start running, there is no multiple menus or clutter of buttons – it’s straight forward”
5) Do you think the squad screen delivers good functionality in changing your formation?
comments:
“Older managers are brutal with technology and most of the managers out there can’t make heads or tails with these applications – this makes it simple, even they could use it”

6) Does the Tactics screen make a clear visualisation of player movement? Rate out of 5
Score: 5
Comments:
“Sure it only takes 2 presses and you see players moving. It's easy to explain it to a young lad only starting football where you want him to be because they don’t understand at that age”

7) On a rating between 1-5 could you see yourself using an App such as this?
Score given: 2
comments:
“Under 16’s are too young for this to be used, they don’t take football serious enough. I think it is more suited to young lads because they don’t know where to stand. Senior players who take the game serious may also find it really useful. But it does work really well”
“Personally I feel I would want a bigger tablet to work with this on but that is just my opinion”

8) Any further comments or questions you might have?
“Within contacting players it would be excellent to have a group of contacts of all the players so every time I want to text the lads about training or a match it would only take a second”
5.3  Leister Celtic F.C

Manager Gerry Geraghty of the under 17’s football team was the application evaluator for Leister Celtic F.C. The same approach as of Broadford Rovers was taken in the sense of how the meeting proceeded. Starting out providing the manager with a description of the application and then allowing him to use the app in his own time was the approach taken.

Some interesting points were noted and suggestions the manager had are recorded and discussed as follows:

1) Suggested half-pitch zoom feature
2) Multi formation saves
3) Passing and Receiving functionality

5.3.1  Half Pitch Zoomed Feature

The half pitch zoom feature was an excellent suggestion by Gerry. He commented that it would be an excellent feature to have to see just the attacking players of your team and explain to them where they are supposed to be at a given time for example during a counter attack. It was mentioned to Gerry about suggestions of zoomed in feature for a corner or a freekick as of above from Broadford, but he commented:

“I think corner or a free kick zoom would be too concentrated, I would rather have half the pitch either top or bottom if I want to discuss attacking or defending, because every player is not involved in every tactic”

This was an excellent comment to make and could possibly add substantial value to the application.

5.3.2  Multi Formation Saves

When Gerry realised that he could not save multiple formations and then draw on different ones for different situations he commented:
“It would be an excellent feature to allow a manager to load saved formations from let’s say 4 weeks ago when the manager knows that that formation worked really well. I would like the opportunity to load a number of last saved formations”

This suggestion certainly would add value to the application in providing the manager with the opportunity to draw on different formations throughout the season.

5.3.3 Passing and Receiving Functionality
Another suggestion was made in the direction of teaching young players how to play football technically. Gerry commented:

“It would be great if you had the opportunity to show a young lad to pass to a player who is free and then run into space away from the opposition”

This functionality would provide an excellent opportunity for possibly a new application in the area of teaching football rather than football management.

5.3.4 Survey Results
The following survey results were recorded during the visit to Leister Celtic.

1) How well do you naturally understand the App between 1-5?
   score given: 5
   comments: “Easily understood by anyone with coaching experience”

2) Would you change anything about the App and why?
   comments: “half pitch zoom recommendation and multi saves for formations”

3) Rate the App out of 5 in terms of its ease of use
   Score given: 4
   Comments: “People not I.T savvy would understand how to work this”
4) Rate the APP out of 5 in terms of the functions it fulfils
   score given: 5
   comments: “Does everything that it aimed to do”

5) Do you think the squad screen delivers good functionality in changing your formation?
   comments: “excellent moving of players to create new formations”

6) Does the Tactics screen make a clear visualisation of player movement? Rate out of 5
   Score: 5
   Comments: “It’s sufficient for what it was built for”

7) On a rating between 1-5 could you see yourself using an App such as this?
   Score given: 5
   comments: “Certainly, and I think younger teams starting out could use this also”

8) Any further comments or questions you might have?
   Comments: “I think definitely kids would understand passing, receiving and running into space better if it was added”

5.4 Evaluation Conclusions
Through both visits to local football teams there is a clear interest in this application as managers seem to feel that it makes current systems redundant. There is certainly some substantial scope for further development as numerous suggestions from both managers tend to be aimed towards football learning for young players starting out and more advanced tactical creations such as corner, free kick, half pitch for offence and defence and much more which certainly could add substantial value to an application such as this.
Chapter 6

Conclusion

This report documents the successful implementation of an Android application which allows the end user (a football manager) to:

1. Manage player positions in their team formation
2. Create new visual tactic animations of multiple players synchronously
3. Manage events such as training sessions and matches with Google Calendar integration
4. Contact multiple players through player contacts messaging functionality

In the delivery of this project it has succeeded through delivering these objectives to an excellent standard as provided through feedback evaluations of two local football clubs. This is not to say by any means that the application is complete. An application such as this has great potential to flourish as it provides a strong base structure for future development.

5.5 Problems Encountered

There were a number of problems encountered throughout the development of this project around storing player coordinates originally to a text file in memory. Originally on file creation and writing to this file, it was not appearing in the project structure in the IDE. To overcome this problem a function was written to collect data from the file to see if it did exist. This approach worked as the output was a match for the test input. It is quite possible that Android does not provide direct access to newly created files from an outside application possibly for data integrity.

Other problems arose during developing of the multi player animation feature. Originally the approach was taken of: if a player was clicked and a place on screen was clicked, animate this player. If another player was clicked and then an animation was attempted
with this second player, it would not work with this approach as the first player animation would cease to exist and the second player would only animate. A reasonable assumption for this in Android is that a sequence of animations can occur and once a new sequence is registered the old sequence stops. In Android the above approach are seen as two separate sequence of animations each containing one translation animation. The problem was overcome by creating an Animation Set and adding every newly created Animation to this Animation Set. A call to invoke all Animations in the Animation Set then fulfilled the nature of having multiple player animations at the same time.

5.6 Final Remarks
In building this project new skills have been attained in computer science and certainly excellent experience has been gathered through exposure to gathering requirements, utilising Android technologies, and carrying out ongoing organisational tasks for the project. It was both an interesting and invaluable experience to have of mobile application development and there seems to be a gross requirement within the football management sector for an application such as this. Opportunities for the expansion of this product to further benefit the end user are prominent in features suggested during the chapter entitled ‘Evaluation’ by football managers and successful prospects lie in the near future for the course of this application.
References


10. ‘Setting up an existing IDE’, available at: https://developer.android.com/sdk/installing/index.html, accessed [14/03/14]


13. ‘FluidUI’, available at: https://www.fluidui.com/, accessed [13/03/14]


