MARK MCALLISTER

1741906

Abstract

The inception planning of a local rugby system from information gathered

Simple Rugby

Part A Inception Plan

Contents

[Project Brief 2](#_Toc36297251)

[Aims of the Project 3](#_Toc36297252)

[Functional Requirements 4](#_Toc36297253)

[Non-Functional Requirements 5](#_Toc36297254)

[Information Gathered 6](#_Toc36297255)

[Resources 7](#_Toc36297256)

[Project Schedule Plan 8](#_Toc36297257)

# Project Brief

The aim of the project is to design and build an object-oriented program to track and update rugby players from a local rugby club called Simple Rugby. This is to include the details and development of the players and their skill through training sessions and playing in matches. There will be information stored on players from two different levels determined by their age (junior or senior). Information will also be stored on the coaching staff who will be able to update the player details and skill levels, and the membership secretary who will have the ability to add coaches. Non-staff have the option join and become members to support the club and these members will be able to see how game results go.

# Aims of the Project

The main aims of the project are to create an easy to use, secure program that allows staff members of a club to add, track and update information on their players. They will start by adding a player, specifying their position and putting in their joining details. The details on skill levels will be discovered through training and in games and the player profile will get updated as a result of these. The program will implement a graphical user interface as a means of increasing user-friendliness and reducing effort required from the staff members. Member secretaries act as admins and can generate reports on players and add and remove coaching staff as required. Non-staff will be able to register to join the club as a member to support the club by using facilities and would be able to maintain their account through the application.

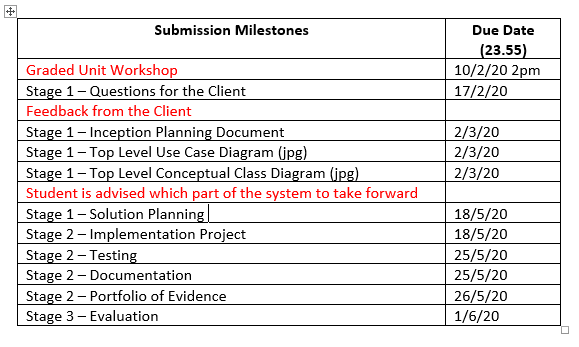
# Functional Requirements

The system will have 3 user types. These are the admin, the coaches and the non-staff members. When logging in, login validation will occur to ensure the correct user type has access to the correct information. Each user will have a unique user ID that they will use with an auto generated password that they must change on first login. The login validation will ensure that a correct user ID matches the password for the user and admin before proceeding. The admin account will be held to the membership secretary for the club. The admin account will be used for the creation of coach accounts and non-staff accounts.

The coaches for each team will be able to add players specifically for the team they are assigned only. They can perform create, read, update and delete operations on the information on their team. This includes the information on the players, the skill development of the player, the details of a match and training session details. Non-staff members will have the ability to view match information.

# Non-Functional Requirements

The first thing to decide for a requirement for the program is a milestone chart, marking important areas of the development and when they should be completed by:



The project created will take assumptions of low technology skill from the users so will need to be user friendly, laid out well and robust to aim to eliminate issues that any user could come across. It will be programmed in the language Java and it will implement a graphical user interface to aid in the user-friendliness of the program. It will also be important to ensure that GDPR is followed as well as good coding practices by using meaningful variable/function names and the program will be commented to aid in understanding of what is meant to be occurring.

# Information Gathered

I have obtained information relevant to the rules and how rugby is played by both watching games of rugby on television or physically being present for a game and from visiting a website that explains information on rugby union in more depth.

<https://www.rulesofsport.com/sports/rugby.html>

The rules of rugby are:

15 players play on field at a time for each team at a time in rugby union made of 8 forward positions and 7 back positions. Unique position names are fly half, scrum half, hooker, prop, flanker, number eight, winger, full back, second row, inside centre and outside centre. Distance of the pitch is measured in metres.

Points are scored in 4 different ways; trys, penalty kicks, drop goals and conversions with the points equating to 5, 3, 3 and 2 for each respectively. There are also different methods of kicking the ball. A player’s position, skill level, training record and game performance will be recorded and updated whenever any sessions, games or changes occur.

As well as obtaining information on the rules and regulations of the sport itself, it was important to gain information relevant to creating the program required. To do this, further information was requested from the client to elaborate what information will be obtained and tracked and how this is wished to be achieved. On top of this, research will be done using w3schools.com, stackoverflow.com and relevant previous work will be used in assistance on obtaining details on how to perform actions within the program.

# Resources

This program will be created in the language Java and will implement a graphical user interface for interaction by the user. This will be programmed using an application called Eclipse and will use the window builder plugin to aid in the creation of the JFrames used. Other software that will be required in the planning, building and testing of this will be:

* Windows operating system version running windows 10
* Microsoft office for documentation
* Google chrome for research
* StarUML for creating use-case and class diagrams

The hardware being used in the creation of this are:

* Processor: AMD FX8350 Eight-Core Processor
* Graphics card: Nvidia GeForce GTX 970
* RAM: 16GB
* Hard disk space: 2TB HDD

# Project Schedule Plan



The key shows the colour representing each phase of the project with blue being the planning, green the developing, and red the evaluating. The blocks that are coloured in within the chart are the weeks in which each part is intended to be started on, worked on and finished by.

From the project schedule, the aim is to work on the later stages of planning; the detailed class diagram, creating a model of views, designing sequence diagrams, and creating activity diagrams will be worked on whilst implementing the project. The test plan will be generated alongside implementation and the testing will be done as this is happening too. Once the project implementation is complete, the last of the testing will occur and all the documentation will be gathered together ready to be submitted. Once the project and portfolio are submitted, a self-evaluation will be completed to reflect on how each stage of planning and development went and what could have been improved.