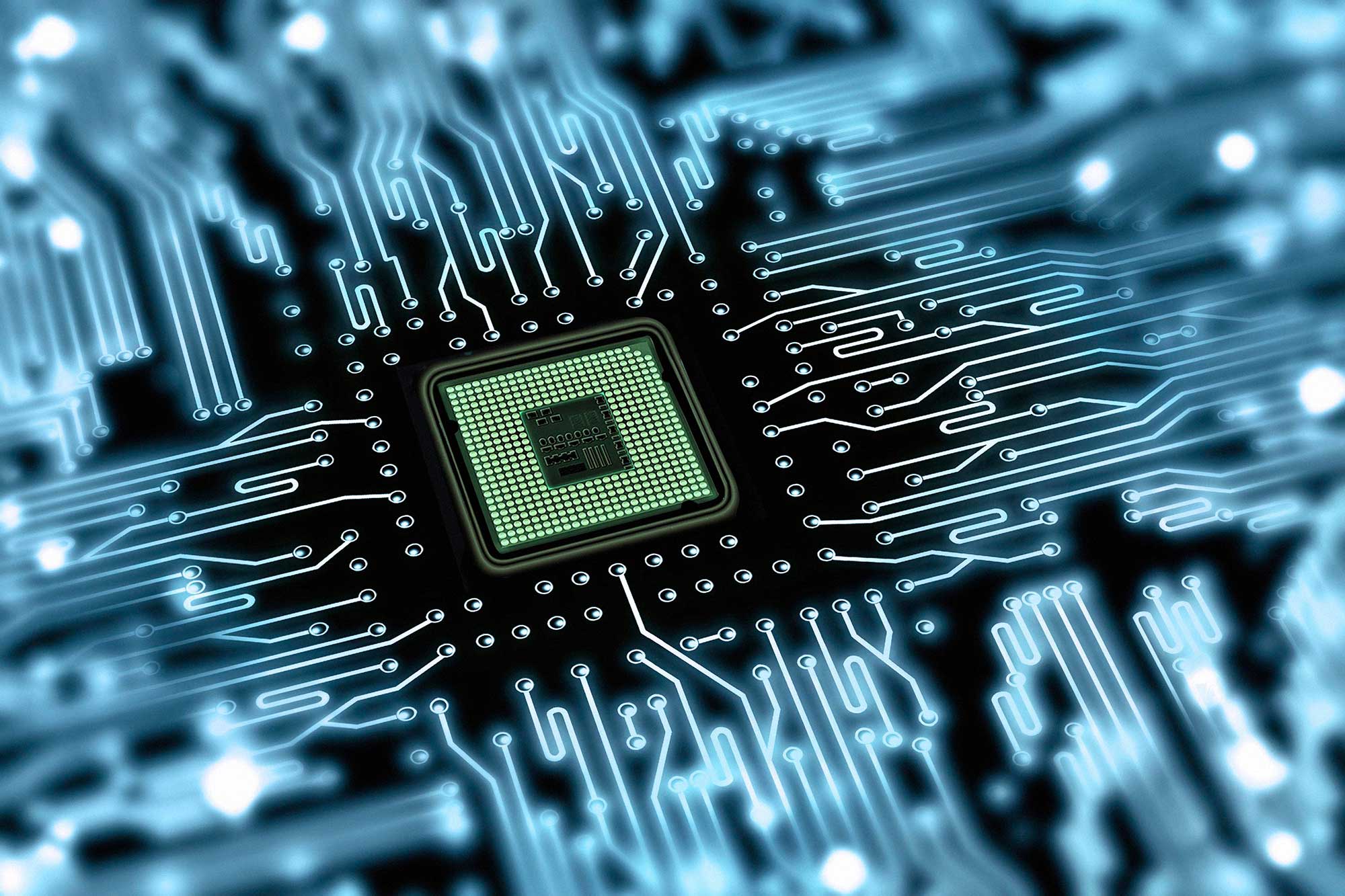
HND Computing:

Software Development

Graded unit 2

H48W35

Brodie Harkins

Evaluation

Contents Page

Contents

[Introduction 3](#_Toc104929018)

[Outline of the Assignment 4](#_Toc104929019)

[**Strengths & Weaknesses of the Practical Assessment** 5](#_Toc104929020)

[**Summary of Modifications (Unforeseen Events)** 7](#_Toc104929021)

[**Knowledge & Skills gained & how to improve project development** 8](#_Toc104929022)

[Inception Plan 8](#_Toc104929023)

[Solution Plan 8](#_Toc104929025)

[Implementation 9](#_Toc104929029)

[Test Documentation 9](#_Toc104929030)

[User Documentation 10](#_Toc104929031)

[Appendix 11](#_Toc104929032)

# Introduction

This evaluation will cover the overall development process of the simplyRugby application. Within this document I will cover:

* An outline of the assignment and to what extent the solution met the original requirements of the assignment brief.
* An assessment of the strengths and weaknesses of the outputs of the practical assignment.
* Recommendations for any future development of the solution and reasons for these recommendations.
* A summary of any modifications to the project plan, solution plan and/or implementation, which were made during the project. Including reference to any unforeseen events and how they were managed.
* Identification of any knowledge and skills which have been gained or developed while conducting the project assignment and how the actions/process of conducting the project could have been improved.

Included and discussed within the document will be the original inception plan and solution document, Implementation, Testing document, and User Documentation. Those will be listed in the appendix

# 

# Outline of the Assignment

SimplyRugby, a rugby club, entrusted me with developing a prototype. They had stated that they were searching for a system that would enable coaches to track player information for each player on their team. Coaches will be able to keep track of games, player skill information, and training data as well. They also stated that they would have a membership secretary who would be able to maintain personal information on all club members and send emails to each member with any club news. After reading the club chairman's brief, we sent additional questions to our project manager, who will meet with the club chairman to discuss the questions that were submitted. The conception plan was created during this time. This gave me a clearer understanding of what was required within the prototype and what wasn't.

Following the conclusion of the discussion, it was decided to proceed with the prototype and develop the skill system that would track a player's skills. Other components specified that were not necessary for this prototype included the game and training session trackers, which were not to be developed on inside this prototype.

Following additional investigation using the solution from the original brief, I decided to include the player skills tracker as well as a login system to go along with it. I concluded that an administrative login would be required, allowing the secretary/Admin user to add and remove Junior and regular players and coaches from the system. The purpose for this was to provide the club chairman with a better grasp of how this prototype will function and what will be necessary when it is further developed for production.

# **Strengths & Weaknesses of the Practical Assessment**

STRENGTHS

One of the application's benefits, in my opinion, is its ease of use when viewing and browsing the system. Using a side bar that is always visible to the user and is suitably phrased. This enables the user to successfully choose what they want to do. Utilizing design tools for Swing and also combining this with the design pattern – Model, View, Controller. Swing was simple to utilise because of the broad variety of components available. For example, utilising a calendar or JSpinners to select and enter a value.

The use of the MVC design pattern was another strength of the practical assessment. Throughout the process, I felt confident in my comprehension and how it should be used within the prototype. I felt satisfied after creating the class diagram that I had identified the correct classes that will be used in the implementation. Once it was implemented, each class was used and the relationships between each classes worked and was used efficiently, and no errors or confusion was caused.

WEAKNESSES

During the application's development, I neglected to apply try catch error handling in places where it could have been used. This, however, does not disrupt or alter the application in any way and is a simple repair if the client decides to proceed with the prototype.

Another weakness that I noticed was the ability to test the application thoroughly. During the development stage I hadn’t tested the application enough. This resulted in the prototype being put forward to contain an error that had gone unnoticed until the test plan deadline was upon me. This meant that the test plan didn’t match up to what was delivered to the client.

RECOMMENDATIONS

I would recommend implementing a team system that would allow the admin user to add a team. When adding the new team, you would also add a coach to that team. Upon adding the new team, it will be made available to new players when they are being added to the system. This would mean when a coach is logging in it would display the players that are a part of the squad.

I would also recommend utilizing a database which would contain all information on players and users within the system. This would allow for an easier method of storing large quantities of data. I used serialization within the prototype to show how a coach/Admin might interact with the system.

Upon further development I would consider proper test plans to be carried out, to test every part of the application. This would ensure that the delivery of the application is up to a high standard, and it meets the clients requirements. An example of testing this application would be using automated testing. The tester can create different tests using this method to test different functionality of the application.

Another thing to consider for future improvement is a check on each player's SRU number. We are unable to check a player's SRU to ensure they have joined up in order to participate because we do not have access to the Scottish Rugby Union.

Finally, include a check to see if a player is under the age of five. I have included assessing a player's age to see if they are old enough to play for the senior team or if they should be in the junior squad.

# **Summary of Modifications (Unforeseen Events)**

During the creation of SimplyRugby, I was in a difficult predicament regarding how to display players and gather information about their skills. I had initially configured the lists to accept a string value for that player, such as Player name. This would allow the user to select a player and, if necessary, modify their information. Because of how I had configured the skills tracking system, I had to redo how I displayed players in a list.

I had decided to make the lists display an object of that player. This would allow me to track and save skill details to an individual player object.

How I made the lists display player objects was implementing a cellRendererComponent. This would get the list specified and create a JLabel with its text being set the value of the Players name. This way of rendering an object to a JList was then utilized throughout the application, providing an easier way of tracking a player skills, juniors, & personal details.

Due to this event happening it had caused a slight delay in completion of the Solution document, thus resulting in missing key details that should have been implemented into that planning document.

During development of the SimplyRugby app. I had run into an error where all player objects were being updated when a coach would save a player skills. I had done some testing into how I set up the skills within the controller where it would create a new instance of skill and set that skill to that player object. I had concluded that this was working as intended.

Once I had conducted that setting up the skills worked, I had then moved onto how the skills are saved. Upon testing this by removing all players and creating an individual player to test this. I would then test to make sure the skills are being saved to one player. This test was a success. After testing I added a new player and edited and saved their skills as well. This test failed and gave me an idea of what was causing the error.

I'd decided to look into how an array of objects is produced. When making a static array, like I had done. When you save an object's data to a static array, it saves all of its attributes to every other object in that static array. I fixed this by removing the static modifier and replacing it with a private array with a getter to access skill data. Following additional testing, the skill system was effectively saving individual player skills.

# **Knowledge & Skills gained & how to improve project development**

## Inception Plan

I improved my ability to look for Functional and Non-functional needs when constructing an inception plan. Where I could have done better is to provide more detailed explanations of why or how these things are required.

I also gained extra knowledge into developing use case diagrams. I managed to create use case diagrams that covered the required areas on how a user might interact with a specific part of an application. I did find I need to practice more on developing my skills and knowledge that goes into creating a Use Case Diagram.

# We were entrusted with creating a project strategy when establishing the Inception plan. This would provide us with an outline of how the development stage might proceed, as well as deadlines and a time frame to work with. I was able to develop a clear and realistic plan with obvious dependencies. Unfortunately, I overlooked a few important points in this plan. Milestones and a more detailed breakdown of significant tasks such as the implementation and testing stages Again, researching and honing my skills in generating a succinct and professional project plan would have aided in the creation of a solid project plan to work from.

## Solution Plan

The solution plan's development progressed well, and I managed to create a proper solution plan that met the requirements. I ultimately fell short again with my Use Case Diagrams, the readability of my wireframes, Data Binding, and covering the HCI decisions that were made when developing the GUI for users.

# When developing a solution strategy, I should ensure that the wireframes are understandable to all readers of this document. Importing a PDF file including all images of wireframes or any other type of documentation such as Use Case Diagrams, Extended Class Diagrams, and so on would be one solution to this problem.

# Incorporating data binding into the solution plan would benefit me since I would know what each aspect of the GUI will do. This will help with getting an idea of how each element will interact with the user.

# My knowledge and skills in building a Use Case Diagram for this application were adequate for addressing the program's functions but were not adequately designed. Conducting extensive research and utilising my skills in producing a clear and simple Use Case would aid in the creation of a well-structured diagram.

## Implementation

Overall, the implementation stage was a success. I was able to develop a working application that users can utilise. I added error handling and improved my understanding of designing a Swing application. I also investigated displaying an object within a list. This is how I came across the cellRendererComponent. This enables the inclusion of selectable objects within a list.

I could have included conformation boxes on my app. This would allow a user to select logout and be greeted with a confirmation box allowing the user to logout or remain logged in. This would avoid an unintentional logout. The same is true for adding, erasing, and preserving any modifications made to a player's information.

Another feature that may have been introduced to improve usability is the ability to edit administrative users. This would enable other admin users to remove out-of-date users who are no longer members of the club.

## Test Documentation

Stage 3 testing was a success, with only a few minor tweaks needed if the customer decides to move forward with this proposal. I tested each system thoroughly, including adding, removing, and editing players, juniors, administrators, and coaches.

I provided a plethora of test cases in the test documentation, covering anything from different situations on how a user can overlook aspects of the system to external user testing.

This section can be improved by ensuring that I have thoroughly tested the application for probable exceptions. After the implementation deadline had past, I discovered an exception. I noted this problem in the test manual and demonstrated how I would fix it. If the client chooses to proceed with the prototype, this problem will no longer exist, and the appropriate error message would be displayed.

As previously said, I also included user testing. I had my father and a friend who also studies software development test the programme by following a series of instructions. This resulted in a properly tested programme that was a success because no faults were discovered.

I could improve this stage by obtaining feedback from each external test. This feedback can contain questions which can ask the tester:

* How simple was it to navigate the GUI?
* How easy was it to add a player?
* How easy was it to login?
* How easy was it to edit the player details?
* Give your thoughts on what can be improved from the prototype?
* What worked well?

These questions are just a sample of what can be asked. This would give an idea of what could be introduced in the future and what GUI modifications could be done to make navigation easier for new users.

Also, having a wider ranger of external testers that aren’t aware of my techniques and the application itself in order to get unbiased feedback allowing for a more constructive form of criticism.

## User Documentation

I had never developed a user manual before, so creating one was new territory for me. I decided to try to convey every piece of functionality in a formal manner, ensuring that the user understands what each portion of the programme will perform. Including screenshots and descriptions for each page will assist me in writing a clear and simple User Manual for new users. Overall, I learned more understanding about making a user manual, and with further practise, I will be able to further expand my newly discovered abilities and expertise.

# Appendix

Inception Plan:

<https://drive.google.com/drive/folders/1ZS2ORvPdkjvvPgWED_CZs0WPRSWFrSMS?usp=sharing>

Solution Plan:

<https://drive.google.com/drive/folders/11MtPJFK_7vN_u2667Sqq3O6cK7yNrSU2?usp=sharing>

Implementation:

<https://drive.google.com/drive/folders/1uGvIBGsWdVlVB1wEA-E546P0UYBRwRfZ?usp=sharing>

Testing/User Doc:

<https://drive.google.com/drive/folders/1GQ1w99Sa3fwzhP_YKkwwSChSmLxIUoEE?usp=sharing>