Liam Courtney

dundee and angus college  1625542

Simply rugby evaluation

Table of Contents

[Outline of Assignment 2](#_Toc166249419)

[Were the Original Requirements Met? 2](#_Toc166249420)

[Original Requirements 2](#_Toc166249421)

[Functional Requirements 2](#_Toc166249422)

[Non-functional Requirements 5](#_Toc166249423)

[Strengths and Weakness During the Project 6](#_Toc166249424)

[Strengths 6](#_Toc166249425)

[Weaknesses 8](#_Toc166249426)

[Recommendations for Future Development of the Project and Reasons for Recommendations 9](#_Toc166249427)

[Modifications Made to the Project Plan and Design During Development 10](#_Toc166249428)

[Skills and Knowledge Gained During the Project 11](#_Toc166249429)

[Planning 11](#_Toc166249430)

[Planning Process 11](#_Toc166249431)

[Visual Paradigm/UML Diagrams 11](#_Toc166249432)

[Rugby Knowledge 13](#_Toc166249433)

[Development 13](#_Toc166249434)

[Entity Framework 13](#_Toc166249435)

[WinUI3 13](#_Toc166249436)

[Microsoft Azure SQL Server 13](#_Toc166249437)

[Using Other 3rd Party Libraries 14](#_Toc166249438)

[Robust Testing 14](#_Toc166249439)

[How the Process of Completing the Project Could Have Been Improved 14](#_Toc166249440)

[Final Completed Plan 14](#_Toc166249441)

[References 16](#_Toc166249442)

Evaluation

# Outline of Assignment

‘Simple Rugby’ is a rugby organisation that currently has a paper-based system for storing and tracking all aspects of their organisation. Teams at both junior and senior levels contain players and coaches, as well as participating in matches and training sessions. The organisation also manages members not associated with a team e.g. secretaries. All this information is currently logged and stored physically through paper forms by the coaches and secretary.

Through careful planning and meetings with the organisation’s chairman, a new electronic system for managing all aspects of the organisation was proposed to alleviate the current paper-based system. During the development stage the proposed system was built and tested to meet the current requirements of the organisation, as well as additional features to upgrade the current system.

# Were the Original Requirements Met?

## Original Requirements

### Functional Requirements

#### **High Priority (Required)**

**Secretary:** Add teams at different age levels, such as junior and senior.

**Secretary:** Input and store non-playing members’ (coaches) personal details, including name, SRU number, address, postcode, phone number, and email address.

**Coach:** Input and view players’ details, including name, SRU number, possible positions, address, postcode, date of birth, phone number, email address, next of kin’s name and contact information, and finally, doctor’s name and contact information. Junior players are required to provide extra guardian information as well as a signed consent form each year.

**Coach:** Track and view individual player’s skills in different categories and subcategories, including:

* Passing:
  + Standard
  + Spin
  + Pop
* Tackling:
  + Front
  + Rear
  + Side
  + Scrabble
* Kicking:
  + Drop
  + Punt
  + Grubber
  + Goal

**Coach:** Keep viewable records of individual training sessions with information such as date, time, location, skills practised, activities undertaken, players present, accidents/injuries, and the coach organising the session.

**Coach:** Keep viewable records of matches played with information such as opposition, date, kick-off time, location, result, score, and first/second half performance.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Requirement Met** | **Comments** |
| **Secretary:** Add teams at different age levels, such as junior and senior. | Yes | *This was a mandatory requirement, and it was implemented as expected for both junior and senior teams.* |
| **Secretary:** Input and store non-playing members’ (coaches) personal details. | Yes | *This was a mandatory requirement, and it was implemented as expected.* |
| **Coach:** Input and view players’ details for both junior and senior players. | Yes | *This was a mandatory requirement, and it was implemented as expected for both junior and senior players.* |
| **Coach:** Track and view individual player’s skills in different categories and subcategories. | Yes | *This was a mandatory requirement, and it was implemented with a change to the UI design.* |
| **Coach:** Keep viewable records of individual training sessions. | Yes | *This was a mandatory requirement, and it was implemented as expected.* |
| **Coach:** Keep viewable records of matches played. | Yes | *This was a mandatory requirement, and it was implemented as expected.* |

#### **Medium Priority (Optional)**

**All Users:** Create an account with permissions based on the user’s role within the organisation.

**All Users:** Login to the user’s account with access based on their role within the organisation.

* **Secretary:** Admin account with access to the whole system.
* **Coach:** Account with access to their team’s players, training sessions, and matches.

**Secretary:** Verify non-playing members’ SRU numbers manually before being allowed access to their accounts.

**Coach:** Calculating an average of each player’s skills in each category and the total average overall categories.

**Coach:** Track players’ previous skill development, showing the progression made by the player.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Requirement Met** | **Comments** |
| **All Users:** Create an account with permissions based on the user’s role within the organisation. | Yes | *This was an optional requirement, but it was implemented due to the robustness it gave the system.* |
| **All Users:** Login to the user’s account with access based on their role within the organisation. | Yes | *This was an optional requirement, but it was implemented due to the robustness it gave the system.* |
| **Secretary:** Verify non-playing members’ SRU numbers manually before being allowed access to their accounts. | Yes | *This was an optional requirement but was implemented due to the security it added to the system. This was achieved through the admin portal.* |
| **Coach:** Calculating an average of each player’s skills in each category and the total average overall categories. | Yes | *This was an optional requirement, but it was added due to the ease it gave coaches to quickly assess players.* |
| **Coach:** Track players’ previous skill development, showing the progression made by the player. | No | *This was an optional requirement, and it was not implemented due to the time it would have taken and the complexity it would have added to the system.* |

#### **Low Priority (Time-dependant)**

**All Users:** Keep a viewable record of previous seasons and all the information accompanying them.

**Coach:** Track players’ previous progression through different age levels.

**Coach:** Determine the ideal position based on the player’s skills.

**Secretary:** Track coaches’ skill levels of training players in each skill category and subcategory.

**Coach:** Submit an electronic consent form with online signatures.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Requirement Met** | **Comments** |
| **All Users:** Keep a viewable record of previous seasons and all the information accompanying them. | No | *This requirement depended on the available time left, and it was not implemented due to the time it would have taken and the complexity it would have added to the system.* |
| **Coach:** Track players’ previous progression through different age levels. | No | *This requirement depended on the available time left, and it was not implemented due to the time it would have taken and the complexity it would have added to the system.* |
| **Coach:** Determine the ideal position based on the player’s skills. | No | *This requirement depended on the available time left, and it was not implemented due to the time it would have taken. The average of the players’ skills was implemented which would help in developing this in future development.* |
| **Secretary:** Track coaches’ skill levels of training players in each skill category and subcategory. | No | *This requirement depended on the available time left, and it was not implemented due to the time it would have taken. The backend for the coaches’ skills was implemented which would only require a ‘User Interface’ (UI) for adding this in future development.* |
| **Coach:** Submit an electronic consent form with online signatures. | No | *This requirement depended on the available time left, and it was not implemented due to the time it would have taken.* |

After evaluating all the functional requirements, all the mandatory requirements have been met. Additionally, certain optional requirements such as the account system and different user access was added, as well as a security measure for the administrator to decide the appropriate access level before the user has any access to the system. This added a robustness to the system, allowing coaches to access the system instead of just administrators.

Finally, none of the additional optional requirements based on time restrictions were added, but for certain requirements, the system was built to allow for further development of these. For instance, the skills feature was added to the functionality of coaches, requiring only a UI to implement this feature fully. Additionally, functionality for averages were added to the skills, allowing for an ‘ideal position’ feature in future development.

### Non-functional Requirements

#### Design

* A blue and white colour scheme based on Scottish Rugby (Scottish Rugby, 2024).
* Design a logo based on the Scottish Rugby logo (Scottish Rugby, 2024).
* Minimalist design for more straightforward interactions to aid users with little technical knowledge.

#### Accessibility

* Larger font size for visually impaired users such as the secretary.
* Simple font choices for visually impaired users, such as the secretary.

#### Time

* The system planning should be completed by 23rd February 2024 to ensure enough time for development and testing of the system.
* The development should be completed by 19th April 2024 to allow enough time for appropriate testing and necessary changes.
* The final system must be completed by the chairman's 10th May 2024 deadline for the upcoming season.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Requirement Met** | **Comments** |
| **Design:** A blue and white colour scheme based on Scottish Rugby (Scottish Rugby, 2024). | Yes | *This was a mandatory requirement, and it was implemented as expected.* |
| **Design:** Design a logo based on the Scottish Rugby logo (Scottish Rugby, 2024). | Yes | *This was a mandatory requirement, and it was implemented as expected, used as the application logo.* |
| **Design:** Minimalist design for more straightforward interactions to aid users with little technical knowledge. | Yes | *This was a mandatory requirement, and it was implemented through a change of the ‘view’ pages to more align with the rest of the design.* |
| **Accessibility:** Larger font size for visually impaired users such as the secretary. | Half Met | *This was a mandatory requirement, but some text such as the input fields could have been larger, or an option to enlarge the text could have been added.* |
| **Accessibility:** Simple font choices for visually impaired users, such as the secretary. | Yes | *This was a mandatory requirement, and it was implemented as expected, using ‘Segoe Fluent Icon’ the default font for ‘Windows’ apps* (Microsoft, 2024)*.* |
| **Time:** The system planning should be completed by 23rd February 2024. | Yes | *This was a mandatory requirement, and it was met by the date required.* |
| **Time:** The development should be completed by 19th April 2024. | Half Met | *This was a mandatory requirement, and after an additional week was allocated, the new date was met.* |
| **Time:** The final system must be completed by the chairman's 10th May 2024. | Yes | *This was a mandatory requirement, and it was met by the date required.* |

After evaluating all the non-functional requirements, all the mandatory requirements have been met. While some requirements may not have been met entirely, such as the size of the text or the development time restraint, most of the requirements were met entirely. In future development, a feature to enlarge the text size would be advantageous, but it is acceptable in its current form. With the development time restraint, the possibility of a time extension was discussed, and as it did not affect the final time requirement, an additional week was allocated to development. The final time requirement was met for the 10th May as expected, so this additional time did not affect the project in any way.

# Strengths and Weakness During the Project

## Strengths

|  |  |
| --- | --- |
| **Strength** | **Description** |
| Hardworking | *During this project, my ability to work hard and get tasks completed really helped with the outcome of the project, especially during the planning and development stages. Without this key strength the project would not have turned out how it did.* |
| Attention to Detail | *My attention to detail really helped in this project mostly in the planning stage but also during the development.*  *Having the small details completed correctly in the planning such as the ‘UML’ diagrams was especially advantageous when it came to the development, allowing time to focus on more minor details in the development of the project. This was most noticed in the development stage during the ‘User Guide’ section, as well as the internal documentation within the code.* |
| Creating and Following a Structured Plan | *My ability to create a very robust and meticulous plan, with lots of research and attention to detail, helped greatly in the planning of this project. Following the plan, was essential in the development stage, as it outlined a step-by-step process in developing the system.* |
| Openness to New Concepts/Ideas | *New concepts are essential in finding solutions to problems for both the client and the developer. During this project, I used many new concepts and libraries in development, which will be discussed later, but these new concepts really helped create a robust solution for the project.* |
| Willingness to Learn | *As with new concepts or ideas, learning is required, during the planning stage it was essential to find solutions for upcoming problems. My willingness to learn gave me an advantage in gaining the knowledge required to create solutions to problems.* |
| Research | *With learning, research is required, finding the right sources and asking the correct questions is essential. The ability to identify an issue and phrase it in a way that could be inputted into a search engine or another resource helped with this project. As well as finding credible sources in different media such as articles, documentation, videos etc.* |
| Asking for Assistance When Needed | *All the planning, research, and attention to detail in the world still leaves issues, it was essential to the project to ask for assistance when needed. This was in many areas, functionality, design, database queries etc. By asking the right people for assistance this helped identify and find a solution for several issues, mainly with the UI.* |
| Backup Procedures | *During all stages of this project, I constantly backed up everything, this was more beneficial than I could have imagined. In the development, I used several new concepts that led to mistakes, having local backups of the project at different stages of development, ready to load with one click, ‘saved the day’ on many occasions.* |
| Constant Testing | *During the development of this project my willingness to constantly test new features or functionality was essential in completing the project in time. Without constantly testing new functionality, I could have wasted a considerable amount of time on development that would have to be changed. The testing documentation was very concise and easy to follow.* |
| Problem Solving | *With anything in life problems will occur, my ability to identify and fix these problems was one of my key strengths in this project. During the planning stage, several problems were identified, solutions to these problems were soon put in place. My problem solving in the development stage was even more apparent, taking a step back and debugging helped greatly in the final solution and the time it took to implement it.* |
| Account System and Access Control | *During the development of the project, I created an account system for the first time, and it turned out surprisingly well. This had different types of accounts such as admin and coach, which all had different access levels. When a user first creates an account, they are assigned as a new user with limited access, needing an admin with full access over the system to assign them a role as either coach or admin. An admin gains full access to the system and a coach only has access to their associated team.* |
| Validation Procedures | *The validation procedures for this project were implemented very well, every input had a full validation procedure for length, format etc. as well as a profanity filter that adds a professional feel to the application.* |
| User Guide | *The user guide for this project was done incredibly well, it was very detailed and contained steps to complete every possible action in the application. This was also added to a help button inside the application, opening the document in the user’s default ‘PDF’ viewer.* |
| Custom Combo Boxes in the Style of List Views | *For the players that attended a training session and the positions selected for a player, I created custom combo boxes in the style of list views. These combo boxes allowed for multiple selections inside the combo box through checkboxes. The list views made the application look significantly worse and took up a large amount of space especially if the team had 20+ players.*  A screenshot of a computer  Description automatically generated |

## Weaknesses

|  |  |
| --- | --- |
| **Weakness** | **Description** |
| Perfectionist | *During all stages of this project, my tendency to always aim for perfection had a negative impact on the project. Spending unnecessary time and energy on trivial issues, when the overall work was ‘good enough’, led to stress and anxiety, meaning more time was spent trying to alleviate this stress and anxiety. Overall, the work was to a high standard and many issues were minor, having the ability to decide something was ‘good enough’ would have helped achieve more during this project.*  *Although this did lead to a more robust solution, it left a negative impact on my mental health and wellbeing, which could have been easily avoided.* |
| Overcomplicating Things | *Again, during all stages of this project, but mainly the planning, my tendency to overcomplicate simple ideas had a negative impact on the project. Some ideas are inherently simple and easy, but my focus on making them the best they can led me to overcomplicate them and use unnecessary time and energy.*  *This was noticed during the development stage, so changes were made to make some ideas simpler and allow for easier development, but some amount of time was already wasted on these ideas.* |
| Burning out Due to ‘Tunnel Vision’ | *Burning out due to ‘tunnel vision’, was a negative effect of my consistency to work hard and strive for perfection. This led to spending long hours, days at a time spent working on the project without breaks. This made me resent the project at times, which inadvertently required more breaks over longer periods.*  *The same amount of work easily could have been achieved if spread out over less hours and more days. The final project would have been the same, but I would have enjoyed the project more, and had less stress and anxiety.* |
| Scope of the Project | *Due to being a perfectionist and overcomplicating simple things, the scope of the project during the planning stage was more than necessary or possible. Although additional requirements were marked as optional or time-dependant, the thought of not meeting these requirements affected me negatively.*  *Also, the concept of adding additional features for ‘free’ didn’t occur to me. In future, a client or employer could possibly take advantage of this for free ‘work’.* |
| UI Design | *The design of the UI for the project was extremely simple due to my design capabilities. If I was more confident and open to trying different designs, the application could have looked and felt more professional. Although, for this project there wasn’t much scope for design due to accessibility reasons, but I could have experimented a bit more instead of focusing solely on the functionality.* |
| Using New Concepts without Full Knowledge | *A negative effect of my openness to trying new concepts was using them without full knowledge. Instead of creating test projects to practice and learn with, I opted to use some advanced concepts such as ‘Entity Framework’ on my original project. This led to several issues, some taking a large amount of time to correct. The use of backups did alleviate some of this stress, but if I had tested the concepts beforehand in a simple project, I wouldn’t have had as many issues as I did.* |

# Recommendations for Future Development of the Project and Reasons for Recommendations

There are several recommendations that I would make for future development of this project, these include:

1. **Try not to focus on making every little detail perfect,** completing work to a high standard is important, but not everything has to be one hundred percent perfect all the time. This takes extra time and creates more stress and anxiety than needed. It is better to identify if the time and energy taken to make something perfect could be better spent adding an additional feature or fixing another issue. Of course, if this is not the case, use the time and energy as needed.
2. **If something seems too simple, it is probably meant to be simple.** Not everything needs to be super complex, sometimes a solution can be as simple as it seems. There is no need to waste time and resources trying to overcomplicate trivial issues when the time and resources can be best used elsewhere. Of course, making solutions robust, secure etc. is essential but this doesn’t mean they need to be super complex or difficult.
3. **Take breaks when needed, try not to overwork yourself.** Working hard is a great characteristic, but sometimes spending too long without appropriate breaks doing something else you enjoy, can lead to burning out and resenting the project. Breaks and taking a step back can also benefit the project by giving you a different outlook, or a chance for you to subconsciously think of a solution.
4. **Be adventurous with the design,** of course, the design should still follow Jakob Nielsen’s ‘10 Heuristics’ and any other appropriate principles but try new things and see how they look. It is important try new designs and ideas, otherwise every project will look and feel the same.
5. **Don’t jump straight in with new concepts,** this is especially important with advanced concepts such as ‘Entity Framework’. Creating a quick test project to try things out and see what works, can greatly help in the long run. Issues with these concepts can become apparent further down the line when time is limited, leading to an increased workload, and added stress.
6. **Enjoy the project more, try not to focus on what still must be done, and try to focus more on what has already been done.** Large projects like this can be daunting and may seem like there is no ‘light at the end of the tunnel’, but taking a step back on focusing on what has been achieved will lead to more enjoyment and fulfilment from the project.
7. **Change the colour of the error text to something more readable,** the colour of the text is currently a shade of grey, this differentiates error messages from standard text, but the text isn’t as readable as it could be. This was an oversight during development and should be promptly fixed in future development.

Taking these recommendations onboard in future development, will lead to easier development and less stress or anxiety. Also, more time will be available for additional features if it isn’t wasted trying to make thing unnecessarily perfect. Finally, trying new designs could give the project a more professional look and feel or even make the application easier to use for the end user.

# Modifications Made to the Project Plan and Design During Development

During the development stage, it was decided to use **‘Entity Framework’** for the database operations, as it wasn’t stated during the planning stage. This change was due to security issues with explicitly writing ‘SQL’ code in the application code, it was also an easier way to manage relationships, as well as using dynamic queries instead of static prewritten ones (Tikhonov, 2021).

**“**The last version of EF Core that supported UWP was version 3.1. We recommend using .NET and the Windows App SDK instead” (Microsoft, 2024). This recommendation comes from the ‘Microsoft’ documentation, following this advice I switched to using **‘WinUI3’** instead of ‘UWP’. The development of the project was almost identical as ‘UWP’, but I was able to use the latest version of ‘Entity Framework’.

An extra validation step was added through a 3rd party library called **‘ProfanityChecker’,** this stopped any profanity from being used in the input boxes by the user. This didn’t really change the development in any way it only required an extra step in the validation process. This step was decided before the validation code had been implemented, so it was a simple addition that added a professional feel to the application.

I removed the **home page**; the login page now serves as the loading page. The user guide made the home page redundant and added extra pages in the navigation when it wasn’t necessary. This led to easier development, by having one less page to create and add to the navigation.

I changed the **skills/activities** and the **injuries/accidents** to strings to be more in-line with the original paper forms. After preliminary usability testing by myself and others, it was found that selecting individual members, skills etc. then adding user input made it more complicated than it needed to be. This saved a lot of time and stress during the development, as my initial plan was to create individual injury reports etc. which was overcomplicating a simple concept.

The **designs** of the **‘view’ pages** were changed to match the ‘add’ pages. During the same preliminary usability testing it was found that it made it simpler for the user, especially for editing fields as they remained in the same place. Again, this saved a lot of time and stress during the development, as the pages were simpler and easier to create.

I also changed **which data was shown in the view pages**, every page only shows the data associated with it, this was most noticeable in the view team page. Again, during the same preliminary usability testing it was found that it kept it simpler for the user to match the design of the ‘add’ pages. This saved a large amount of time during development and led to easier pages to create.

The **design of the player skills page** was also changed due to an oversight in the planning, I had forgotten to add an area for comments for skills. This required a rework of the design in the same format of the rest of the pages. This added some time on to the development, as I had already created the page for player skills, but it meant that the project met the necessary requirements.

Finally, I added an option to **reset the password** for the user’s account, and the ability for an admin to reset any user’s password. This was added for obvious reasons, in the likely event of a user forgetting their password, also for security purposes, if a password is compromised elsewhere. Again, this added some time on to the development, as well as the stress of having to create some new buttons, popups, and functionality for resetting/changing the password.

# Skills and Knowledge Gained During the Project

## Planning

### Planning Process

The knowledge of the entire planning process for the software development cycle was gained during this project. This involved finding the project requirements through research and interviews, creating profiles for users, as well as planning different tasks a user would complete. The planning stage also involved creating a meticulous plan with task priority and scheduling. Finally, both the business models and view models were designed through ‘Unified Modelling Language’ (UML) diagrams and wireframes. This knowledge is greatly useful going forward in future projects as it provides an answer for how and why things in the development are done.

### Visual Paradigm/UML Diagrams

Through the planning process the skill of creating ‘UML’ diagrams was gained through using ‘Visual Paradigm’s ‘UML’ software. This was used to create multiple types of ‘UML’ diagrams, such as class diagrams, activity diagrams, use case diagrams, sequence diagrams, and an entity relationship diagram (ERD). This skill is essential in creating robust business model designs, allowing for better planning in future projects.

*Some examples of the ‘UML’ diagrams created are displayed below.*

*A diagram of a computer program

Description automatically generated*

*A diagram of a company

Description automatically generated*

*A diagram of a computer

Description automatically generated*

### Rugby Knowledge

Before this project I had no prior knowledge of rugby as a sport, I have a strong background in other similar sports such as football, but this was a completely new area to me. Through thorough research, I gained a basic understanding of rugby, including number of players, match lengths, scoring system etc. This knowledge was imperative in the planning stage especially, but going forward this serves little purpose. The knowledge is there if I ever need to call upon it in the future, but it appears to only be useful for this project.

## Development

### Entity Framework

‘Entity Framework’ is an object-relational mapper (ORM) used for database operations in ‘.NET’, this is a key part of ‘.NET’, and served as the database layer for this application (Microsoft, 2022). In past projects I had used prewritten ‘SQL’ code inside the application code, which is a security issue and a novice practice. So, this skill was gained through researching the ‘Microsoft’ documentation and watching online videos from content creators. Going forward, this skill will be incredibly useful as it is the standard tool for database operations in ‘.NET’ (Microsoft, 2022).

### WinUI3

Previously I had created multiple applications in Universal ‘Windows Platform’ (UWP), but ‘Entity Framework’ doesn’t work with ‘UWP’ applications, ‘WinUI3’ is needed. This is a similar experience to ‘UWP’, but it is the most recent UI framework for creating ‘Windows’ applications (Microsoft, 2023). This skill was gained through trial and error following the ‘Microsoft’ documentation. By being the most recent UI framework for creating ‘Windows’ applications, this skill will serve as a key part of my ‘.NET’ knowledge (Microsoft, 2023).

### Microsoft Azure SQL Server

Previously I had used ‘XAMPP’ to host a local ‘MySQL’ database for past projects, but in this project, I wanted to use an online database. ‘Microsoft Azure’ has several features, one of them is the ‘SQL Server’ hosting a ‘SQL Database’, this can then be accessed by anyone with verified credentials and ‘IP Address’. This skill was also gained through the ‘Microsoft’ documentation and online video tutorials, this was a steep learning experience, but it was worth it for the knowledge. In the future, this skill will be used often as well as ‘Entity Framework’ and ‘WinUI3’, by being part of the ‘Microsoft Development Stack’.

### Using Other 3rd Party Libraries

A few 3rd party libraries were used during this project such as ‘ProfanityChecker’, ‘EmailValidator’, and ‘BCrypt’ (Password Hashing). These were easy to add to the project and then use by reading the documentation in the ‘GitHub’ repositories, and all served an essential purpose. The knowledge of these libraries will be useful in the future, but what will be more valuable is the knowledge of integrating 3rd party or my own libraries in future projects.

### Robust Testing

The types of testing that I used in this project, unit testing, functionality testing, usability testing, and integration testing, aren’t a new concept to me. Previously I have used these types of testing, but in a more relaxed way. The attention to detail and thoroughness in the testing in this project really opened my eyes to how crucial testing is for the software development cycle. In future I will be applying the same detail that I applied in the testing of this project in everything that I do.

# How the Process of Completing the Project Could Have Been Improved

On a final note, the development process could have been improved in several different ways, these include:

* The use of a **second display** would be indispensable to the project, in my current working environment there is only access to one display built into my laptop. This added confusion and extra steps when swapping between windows, pages etc. where an additional display would have proved useful. There were some chances to use a second display when working in an office environment, but because in my home setting I was so used to one display, I didn’t always take advantage of these situations. Investing in a cheap display for my home working environment would have been incredibly advantageous in this project.
* If I wasn’t so focused on **achieving perfection in every detail** no matter how small or insignificant, the project would have been a lot easier and ultimately more enjoyable. This could have resulted in additional features, or a quicker development time.
* **Creating a basic isolated test environment,** would have helped regarding experimenting with how advanced concepts integrate with each other. ‘Entity Framework’ and ‘Microsoft Azure’ led to some issues, if these were tested first before working on the master project, this could have alleviated most issues.
* **Experimenting more with the UI design,** would have made the application feel different than past projects. Instead of focusing primarily on functionality, trying new designs would have encouraged future creative endeavours.

Overall, there were some things I would have improved, but in the most part, a lot of things were done well. The use of ‘Microsoft Stack’ concepts such as ‘WinUI3’, ‘Entity Framework’, and ‘Microsoft Azure’ really elevated this project and resulted in gaining new skills for future project. Additionally, the technical and user documentation was completed to a high standard, which benefited the project greatly.

# Final Completed Plan

*All the tasks for the project have finally been completed, and this marks the end of the project.*

A screenshot of a computer

Description automatically generated

# References

Microsoft, 2022. *Entity Framework.* [Online]   
Available at: https://learn.microsoft.com/en-us/aspnet/entity-framework  
[Accessed 09 05 2024].

Microsoft, 2023. *Windows UI Library in the Windows App SDK (WinUI 3).* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/apps/winui/winui3/  
[Accessed 09 05 2024].

Microsoft, 2024. *.NET implementations supported by EF Core.* [Online]   
Available at: https://learn.microsoft.com/en-us/ef/core/miscellaneous/platforms#universal-windows-platform  
[Accessed 10 05 2024].

Microsoft, 2024. *FontIconSource.FontFamily Property.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/windows-app-sdk/api/winrt/microsoft.ui.xaml.controls.fonticonsource.fontfamily?view=windows-app-sdk-1.5  
[Accessed 07 05 2024].

Scottish Rugby, 2024. *Scottish Rugby.* [Online]   
Available at: https://scottishrugby.org/  
[Accessed 12 02 2024].

Tikhonov, D., 2021. *Writing SQL in C# or When You should not use ORM.* [Online]   
Available at: https://itnext.io/writing-sql-in-c-or-when-you-should-not-use-orm-5800a868db43  
[Accessed 10 05 2024].