UNIVERSITY OF DUNDEE

FarmField Desktop Application

An agricultural business management application

John D Lamont

Science and Engineering

Computing

MSc Applied Computing

2019-20

Executive Summary

This project aimed to develop a desktop application to support users to manage their farming/agricultural business budget and, therefore, assist them to make informed management decisions. This is an area which have proven to be problematic at times due to poor organisational and time management skills, lack of understanding of the current market environment, poor record keeping and literacy issues which can impact on the understanding of info, market changes, usage of technology, difficulties in meeting deadlines for agricultural schemes and, therefore, financial issues. Thus, this project intends to assist users with their budgeting and access to applications for governmental support schemes which are an important source of income within this industry. Finally, the program anticipated to be user friendly with people with literacy difficulties which is a wide problem identified in the farming/agriculture community.

The initial planning stages involved research in the field, the development of questionnaires to complement this research and create the requirements followed by the UML diagrams. Subsequently, a prototype was created and developed to provide the writer a clear vision of how the application would work and identify potential weaknesses and improvements required at this initial stage of the design. C# was the language chosen to develop FarmField due to its compatibility with Windows Forms to develop the required GUI and its similarity with the language learnt by the writer during the MSc course (C++).

The final product was tested by the writer and sent to participants to be evaluated. This process produced positive feedback which reinforces the viability of the program in today’s agriculture and rural market.

Declaration

I declare that the special study described in this dissertation has been carried out and the dissertation composed by me, and that the dissertation has not been accepted in fulfilment of the requirements of any other degree or professional qualification.

Signed: John D Lamont Date: 07/09/2020

Certificate

I certify that John D Lamont has satisfied the conditions of the Ordinance and Regulations and is qualified to submit this dissertation in application for the degree of Master of Science.

Signed: Date:

Acknowledgements

I would like to express my gratitude to Dr Craig Ramsay for his advice over the course of the project. I would also like to thank those who participated in the questionnaire, prototype testing and the final evaluation test. Additionally, I would like to thank all my lectures and who have supported me throughout my MSc giving me the knowledge to change my career path and pave a new future. I would also like to thank Matty Souter of NCR for his extra tuition and guidance throughout my MSc. And finally I would like to thank my wife Marina for her support throughout this project.

Table of Contents

Index of figures

Index of Appendices

1 - Introduction

Budgeting skills and the organisation of essential paperwork, such governmental schemes which are a source of income in farming/agricultural businesses, are two essential factors which farmers/business owners require to master to be successful, run a profitable enterprise and face future challenges of agriculture in this industry. However, the path to a successful business is not always straightforward

The writer's previous experience as an Agricultural Inspector revealed that agricultural businesses often struggle to meet governmental schemes deadlines as a consequence of poor organisational and time management skills. If deadlines are missed, significant financial penalties could be imposed. The writer also identified that farmers/business owners often experience book keeping issues, which is mainly paper or nonexistent. This poor record keeping leads to inspectors find it difficult to collect appropriate information and farmers being financially penalised and, consequently, developing stress and anxiety problems.

In addition, literacy problems in the farming community, such as dyslexia, often cause problems with the interface between governmental websites and users who often find these websites overwhelming due to the amount of information displaying on screen.

Finally, financial issues are also a well-known reality within the farming community, especially for small businesses. This could potentially be due to poor financial management and knowledge of enterprise costs with farmers not being able to identify which part of the business is the most profitable and which part is causing financial hardship and, potentially, bankruptcy.

Based on the above, the writer aimed to develop an application to support users to make informed decisions involving their business budget and facilitate the access to governmental schemes applications which are an important source of income. The program was developed to be a user friendly, take into account existing literacy difficulties in the farming community and be easily adjustable to future needs and developments.

The purpose of this report is to demonstrate the development process of this application including the different software engineering stages (e.g. software life cycle, requirements, structure design such as classes, etc.), the code dopted to create the program (C# and Windows Forms) and UX elements associated with this project. Thus, it also aims to demonstrate the skills the writer gained during the MSc in Applied Computing and additional skills such the learning of a new code language (C#) outside the curriculum lectured during the two semesters.

This chapter (Chapter 1) provides an introduction to this project. Chapter 2 outlines the background of the project which involves the link between agriculture and the present governmental, business and technological environment, the impact of literacy issues in business management and usage of technology, the analyses of existing software in the market and the result of the questionnaires applied in the project. Chapter 3 demonstrates the design adopted (software engineering stages). Chapter 4 focuses on the development of the prototype. Chapter 5 introduces the language and the front end development tool (C# and Windows Forms) to create the application. Chapter 6 describes the testing strategies undertaken and the results achieved. Finally, chapter 7 presents the conclusions and considerations for future development.

2 - Background of the Project

2.1 Agriculture and the present business and technological environment

Over the past 20 years technology has become an essential part of agricultural businesses with the introduction of fully autonomous machinery, GIS, drones, real time analysis, management software, etc. However, individuals' lack of understanding of this association as a consequence of literacy issues and poor IT skills can impact on the successful application of technological developments in this field and, therefore, in the future of a profitable and sustainable agriculture sector. This is even more concerning when considering the importance of agriculture in today's life. For example, 38% of the total of European tax payments (€58bn) is spent in farming/agriculture (Weiss, 2019). At National level, there are £20 Million hectares of farm land in the UK. For every £1 of investment in the form of subsidies in the UK, farming delivers £7.40 back to the economy. The UK farming sector also spends around £13.5 Billion on goods and services and the agriculture tech services contribute £14.3 billion to the UK economy (BAYER,2019).

5.7 million of the 20 Million hectares are in Scotland which amounts for 73% of Scotland's total land area (7.8 Million hectares) (Scottish Government, 2016). Thus, agriculture is a significant industry in the Scottish landscape with farmers, crofters and growers producing an output worth around £2.9 billion a year. These stakeholders are also responsible for much of Scotland's £5 billion food and drink exports (NFU Scotland). It is estimated that 67,000 people are directly employed in agriculture (8% of the Scottish rural workforce) making agriculture the third largest employer in rural Scotland (first: services and public sector) (NFU Scotland). According to NFU, there are around 360,000 jobs indirectly related or dependent on agriculture making 1 in 10 of all Scottish jobs with 60% of the food consumed in the UK originated from British Farms.

Technology can support agricultural businesses to reduce costs and increase profit. This is very important in a landscape where 50% of farms no longer make a living from framing itself and 20% generated a loss before accounting for family labour and capital. As a consequence, the levels of borrowing within the farming community have doubled in the past 10 years. The volatility of the food/drink market prices adds another level of stress to this environment (Prince's countryside Fund, 2016). Thus, it is not surprising that European statics show that between 2003 and 2013, 25% of farmers in Europe went out of business (Weiss, 2019).

To overcome the financial hardship experienced by farmers, The Prince's Countryside Fund suggests the development of new skills thinking about the future such as business and financial management skills. The use of on-line tools to assist farmers to '*identify and evaluate the cost of their production and efficiency for their business*' (pp4) is also one of the main suggestions. Thus, this software program is framed within this environment.

2.2 The Government's role in agriculture

The agriculture in Scotland is regulated by the Scottish Government Rural Payments and Inspections Department (SGRPID). This department aims to deliver subsides and financial support as well as inspect if businesses are complying with the rules and regulations of the relevant schemes.

The writer's personal and professional experience have identified that Governmental inspections can be complex and stressful with farmers not having adequate paperwork, failing to meet deadlines and misunderstanding of what is required for the schemes which they have applied for. This is corroborated by Gailbraith, one of the biggest property and farming consultancy company in the country, which states that '*inspections can create stress for farmers who are juggling day-to-day tasks on their holdings with the demands that paperwork can place upon them'.* This can lead to mental health distress and financial loss with farmers paying to major companies such as Gailbraith to manage the run of the business and governmental schemes. This difficult position can be aggravated by the experience of literacy issues. Within this context a program which facilitates the access of governmental documentation beside easy budgeting calculation is an added value to farmers/business owners.

2.3 The implication of literacy issues and dyslexia in the farming community

Literacy difficulties could be caused by several factors such as poor cognitive difficulties, poor school attainment, specific language impairment (SLI) or attention deficit hyperactivity disorder (Hulme and Snowling, 2017). The most well-known disorder is dyslexia which affects 25% of the farming population (NFU). Most people identify dyslexia as a reading and writing problem; however, according to the British Dyslexia Association, although dyslexia primarily affects reading/writing skills, it also has an impact on the way that individuals see, hear, process and remember information. Dyslexia is also known for impacting on organisational skills, co-ordination, mental calculation and concentration.

According to BBC Scotland News, form filling is one of the most vital skills in farming, mainly as a consequence of budgeting needs which involve calculation, management and organisation skills and the application to governmental schemes. Thus, dyslexia or any other literacy issues can have a significant implication on the management of businesses. In its article, BBC showed a farmer who gave up on parts of his enterprise as a consequence of his dyslexia as this person was unable to read and deal with complex forms and processes. As a consequence, this farmer also avoided Governmental schemes which were a source of income to him and his business (BBC Scotland News 2015). Within this context, it is evident that literacy skills, such as dyslexia, can have a negative impact on decision-making. This software aims to aid farmers with this process by facilitating the access to form and providing a simple budgeting application. In technological terms, we call this accessibility.

2.4 The importance of technology and accessibility in the Agriculture Field – an HCI and UX perspective

HCI (Human Computer Interaction) and UX (User Experience) teaches us to look into technology through the optic of the service user. Thus, if this project aimed to develop an application which supports budgeting and the access to governmental schemes by adopting an user friendly interface, then HCI and UX are two concepts which required to be imbedded in its creation. Within this context, the concept of accessibility goes hand in hand with the development of ethical software design. This is supported by Milligan et. all (2019) for whom accessibility is a key component in developing fair and inclusive technology in order to avoid discrimination. With this in mind, the writer believes that by taking the concept of accessibility into account he will be facilitating business management skills. Thus, accessibility is not only about thinking how to overcome a disability, but also about how to give opportunities to people with disabilities. Some aspects of HCI and UX associated with dyslexia were adopted when developing this program. Some of these aspects were:

1. Visual preferences for dyslexia (fonts, colour, and text magnification),
2. Use of large fonts with plenty of leading,
3. Left justification,
4. Clear and concise content (use of headers, short sentences, and bulleted/numbered lists)
5. Preferred line length is 45 characters. Maximum line length should not exceed 100 characters.
6. Use bold text rather than italics or underlines for emphasis.

2.5 – Existing software

Research undertaken demonstrated that there are a considerable number of software management systems in the market. However, considering the knowledge acquired during the MSc, it is possible to identify issues in all the programs identified and analysed. For example, FarmWorks which is distributed and marketed by Shearwell, who are the Market Leaders in livestock tags, presents a program with an outdated interface which is difficult to navigate due to having a considerable amount of information in a single screen and design which do not take into account potential reading difficulties. Additionally, the colours used have very little contrast making it difficult to read and understand where to press to choose the desire icon. The two following screen shots demonstrate the aforementioned point:

Figure 1 – Home Screen for FarmWork by Shearwell

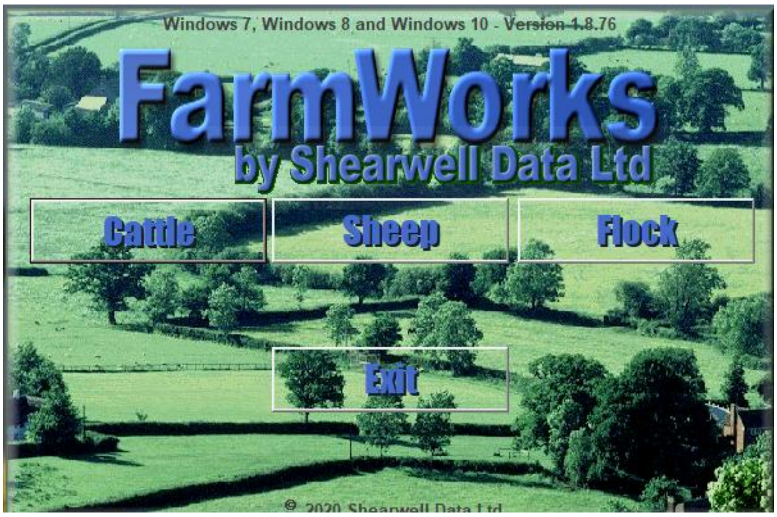
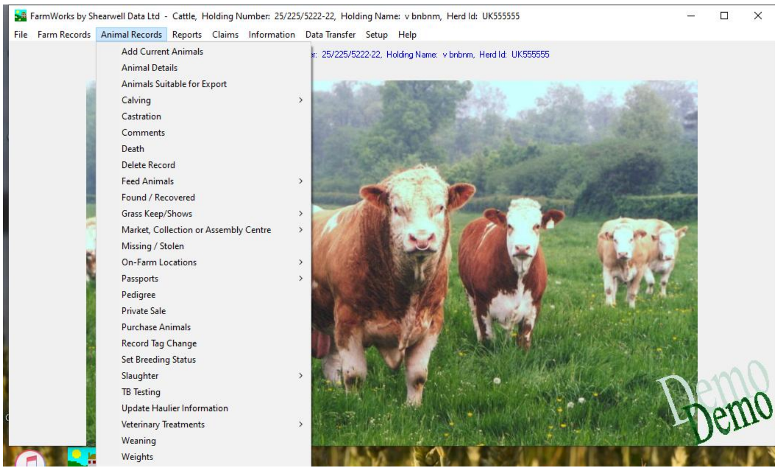

Windows 7, Windows 8 and Windows 10 - ~~Version 1.8.76~~
FarmWorks
by Shearwell Data Ltd
Cattle
Sheep
Flock
Exit
© 2020 Shearwell Data Ltd

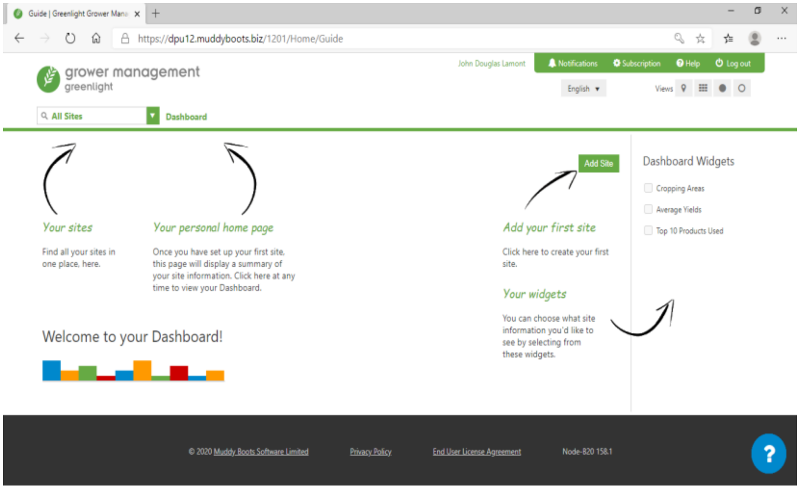
Figure 2 – Cattle menus for FarmWork by Shearwell


FarmWorks by Shearwell Data Ltd - Cattle, Holding Number: 25/225/5222-22, Holding Name: v bnbnm, Herd Id: UK555555

r: 25/225/5222-22, Holding Name: v bnbnm, Herd Id: UK555555

The current market leading product (Muddy Boots) offers a comprehensive program with a modern interface designed to meet a full range of needs of agricultural businesses. However, this comprehensiveness is also its downfall as the program is so vast that it is difficult to navigate in order to find what farmers actually need for their business. Additionally, it is a very expensive program due to its comprehensiveness which means that farmers may not use it because their business does not require all the features to justify the cost. The program also does not allow bespoke versions which farmers could tailor to their business. Finally, the program requests information that it is difficult for clients to calculate and therefore which might discourage them from using it. The following pictures present two screenshots of Muddy Boots.

Figure 3 – Home Screen of Muddy Boots



Your sites

Find all your sites in one place, here.

Your personal home page

Once you have set up your first site, this page will display a summary of your site information. Click here at any time to view your Dashboard.

Add your first site

Click here to create your first site.

Your widgets

You can choose what site information you'd like to see by selecting from these widgets.

Welcome to your Dashboard!

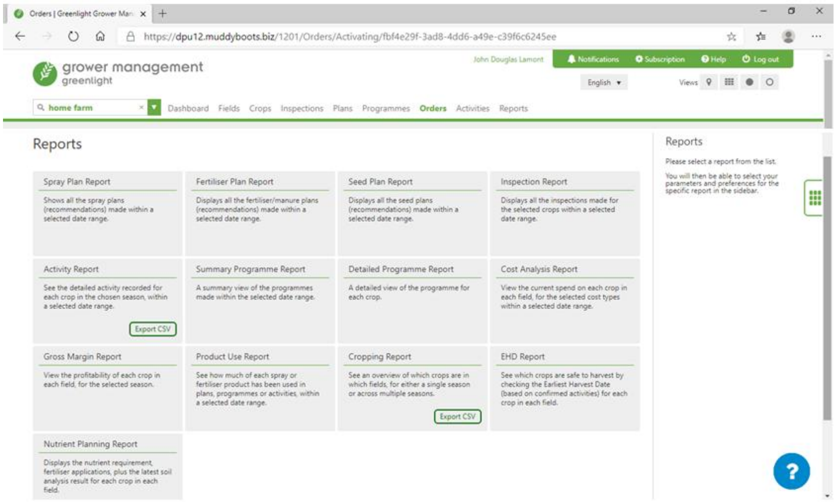
Dashboard Widgets

□ Cropping Areas

□ Average Yields

□ Top 10 Products Used

Figure 4 – Example of Menus for Muddy Boots



Reports

Spray Plan Report
Shows all the spray plans (recommendations) made within a selected date range.

Fertiliser Plan Report
Displays all the fertiliser/manure plans (recommendations) made within a selected date range.

Seed Plan Report
Displays all the seed plans (recommendations) made within a selected date range.

Inspection Report
Displays all the inspections made for the selected crops within a selected date range.

Activity Report
See the detailed activity recorded for each crop in the chosen season, within a selected date range.

Summary Programme Report
A summary view of the programmes made within the selected date range.

Detailed Programme Report
A detailed view of the programme for each crop.

Cost Analysis Report
View the current spend on each crop in each field, for the selected cost types within a selected date range.

Gross Margin Report
View the profitability of each crop in each field, for the selected season.

Product Use Report
See how much of each spray or fertiliser product has been used in plans, programmes or activities, within a selected date range.

Cropping Report
See an overview of which crops are in which fields, for either a single season or across multiple seasons.

EHD Report
See which crops are safe to harvest by checking the Earliest Harvest Date (based on confirmed activities) for each crop in each field.

Nutrient Planning Report
Displays the nutrient requirement, fertiliser applications, plus the latest soil analysis result for each crop in each field.

Reports

Please select a report from the list.

You will then be able to select your parameters and preferences for the specific report in the sidebar.

Finally, it is important to highlight that none of the programs analysed had a link to facilitate the access to the current Governmental documentation.

2.6 Questionnaires

In order to complement the information collated by the literature search and assist with the design of the software, the writer created a questionnaire in order to gather information to facilitate a better understanding of the field, corroborate personal and professional experience and ascertain the demand for a project of this nature and the requirements of the design of the software. The questionnaire was found to be an appropriate method of data collection for this study as farmers and business owners are normally busy managing their work and would require something they could do quickly and effectively (See Appendix 1 – End User Questionnaire Template). The questionnaires were written based on close and guided open questions due to the time scale allocated to its completion, easy reading and ensure focused answers. Within this context, entirely open questions could lead to the shopping cart problem where farmers could potentially undermine the project by wanting personal features or provide answers which were not relevant to this project.

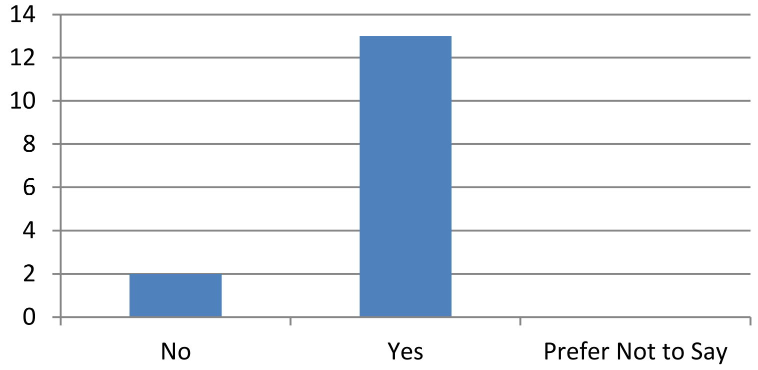
The questionnaire was applied to a sample of 15 individuals from several locations in Scotland. The sample involved young and enthusiastic farmers and elderly farms who have been in the field for decades and whom the writer has met through personal channels. The farmers were contacted by email or phone, a discussion was taken place regarding the project and the questionnaire before the forms were sent out. Confidentiality was explained including how the data would be analysed and stored. To respect the principle of confidentiality, participants were asked to complete the questionnaires anonymously, on return the questionnaires were saved by using a naming convention which ensured anonymity. The writer also explained to the participants that filling in the questionnaire was not mandatory, so if they wish to withdraw from the research they could opt not to send the form.

The variety of the sample allowed the writer to collect a representative sample of the heterogeneity existent in agricultural field in Scotland and to eliminate any potential bias.

The analysis of the questionnaires show:

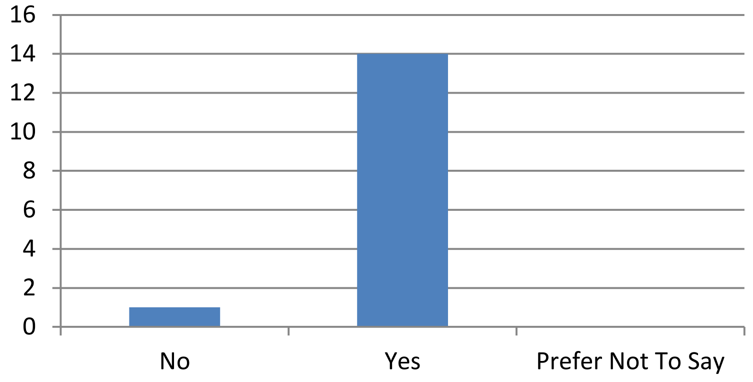
- Most participants were interested in having a simple budgeting program which could show them how each part of their business is performing financially. It is important to highlight that most participants focused their attention on the simplicity of the program, otherwise would not be potentially used.

Figure 5 - Number of participants interested in a simple budgeting program



- Most participants managed their own budgeting as the following graphic shows. Thus, a program which focuses on the budgeting of the farm could potentially be utilised by this community as the graphic bellow illustrates:

Figure 6 - Budgeting self-management



- Most of the participants have a computer as the following graphic shows. Additionally, most of the participants have computer running on Windows. Thus, it makes sense for the writer to focus on developing a software based on this operating system at this stage before considering others.

Figure 7 - Number of participants with computer

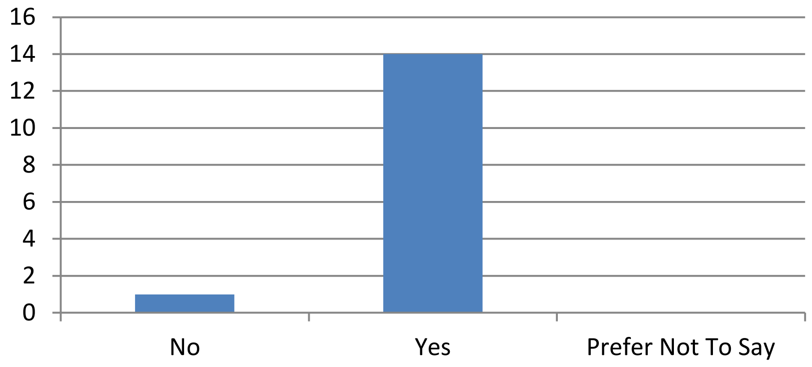
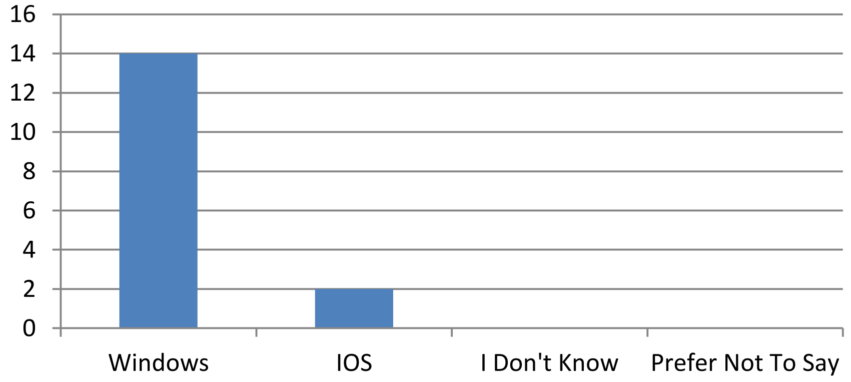
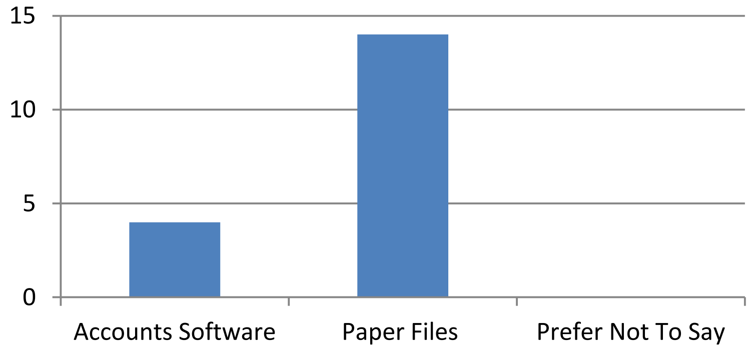


Figure 8 - Operating system utilised



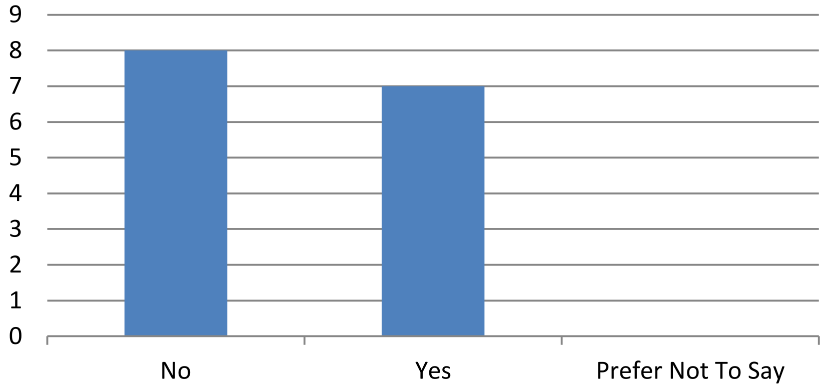
- Although most participants have a computer, it seems that they do not maximise the opportunity that technology could offer to them as budgeting appears to remain predominantly done by traditional paper methods as the following graphic illustrates.

Figure 9 – Usage of software vs paper files by participants



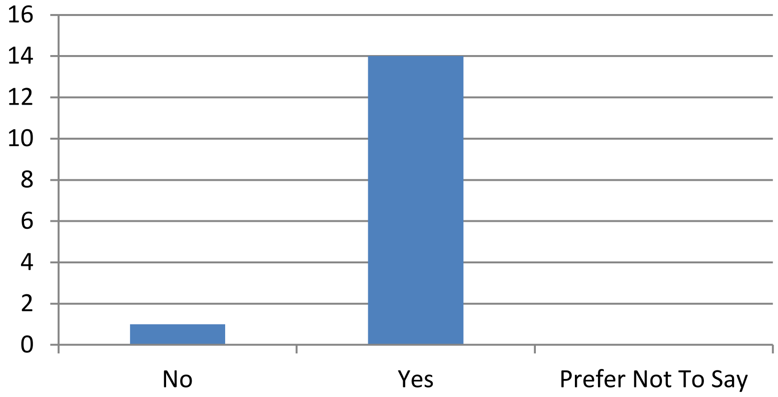
- There were a higher number of people not interested in the development of a mobile application as business and budgeting are often carried out during allocated time in the office. However, those who showed an interest on a mobile app could be explained by the ability to refer to calculations done on the software via mobile app, rather than use the mobile app to carry budgeting work.

Figure 10 - Number of participants not interested in a mobile app



- Interestingly, there are still some businesses that do not have internet and, therefore, a computer and form whom a computer budget system would not be suitable.

Figure 11 - Number of people without and with internet



The questionnaires were also useful to identify that the ARPID Scottish Government Website was considered by most participants as difficult to navigate. Finally, the questionnaires also provided information re the kind of content that the program should involve (please see Appendix 2 Graphic Representation of the Questionnaire Results for a more detailed information and the full graphic representation of the results).

2.7 - Conclusion

This section introduced the current agriculture background and existing technology in field. It provided some interesting data which shows the importance of agriculture in the economy of the country and how technology can contribute to sustainable agricultural businesses. Chapter 2 also identified the importance of budgeting in such volatile sector and the role and contribution of the Scottish Government to profitable agricultural practices. Information was also present regarding literacy issues, specifically dyslexia, and how these issues impact on the farming community and business management, and how technology and ethical software design allied to the concept of accessibility can support the future of farming and agriculture. This section also analysed two existing software programs in the field and highlighted some problems within these applications. Finally, this chapter presents the result of the questionnaires applied in the project and which corroborated the information collated from the literature search and established the foundation of the software design.

3 – Software Design: Approach, Requirements, Use Cases, Threat Model, Classes and Logo.

3.1 Introduction: Project plan and development

The writer planned this application based on the knowledge acquired in the Software Engineering Module of the MSc Applied Computing (Ramsay, 2020). This section will discuss the different stages of the design; however, the prototype and architectural tooling (language and front end environment) will be discussed within specific chapters due to the importance of these stages, though they are still part of software design.

After some consideration, the writer decided to adopt the Waterfall Approach instead of the Incremental Model to develop FarmField. The waterfall model, also named linear-sequential life cycle model, determines that *'all phases involved in this model are completed one by one in linear fashion'*1.

The writer felt that this model would allow a deeper learning and understanding of the different software engineering stages as each phase would required to be fully understood and completed with the necessary detail before moving onto another stage. This was also compatible with the different supervisions sessions carried out every fortnight by Professor Craig Ramsay. Although this model is associated with a higher risk due to the lack of flexibility in changing potential faults and requirements from previous stages, the relative small size of the project worked as a mitigation measure to reduce these risks. Additionally, the fact that the writer tried to spend enough time to achieve the success of each stage before moving onto a different phase also worked as a mitigation measure to reduce the aforementioned risks. Finally, the Waterfall Model felt to be more adequate due to the relatively small scale of this project as it allowed the application to be ready in time by the beginning of September 2020 without needing to change elements of its design.

As mentioned above, the waterfall model was chosen over the incremental model. The incremental model '*is divided into various sub development phases where corresponding testing phase for each development phase is practiced*'2. Although the Incremental Model brings great benefits to software design, such as the ability to accommodate changes in requirements and identify errors at early stages, it requires a more or less clear pathway at the beginning of the project which the writer did not have, especially the requirements which were only fully established followed the literature search and analyses of the questionnaires.

The stages adopted by the writer were as follows:

1) Primary Field Research and Proposal to Craig Ramsay (Project Supervisor);

[1] https://www.geeksforgeeks.org/difference-between-waterfall-model-and-incremental-model/
[2] https://www.tutorialspoint.com/difference-between-incremental-model-and-waterfall-model

2) Field Research (The writer used materials from the course, online search and books which were referenced during the MSc);

3) Definition of the aim of the program and specification;

4) Development and application of questioners for end users (the writer send the questionnaires via email as it was not possible to meet with participants as a consequence of the current COVID-19 lockdown. The email explained the aim of the project/program and the principle of confidentially);

5) Analysis of the information gathered from questioners;

6) Definition of Requirements and Use Cases;

7) Design - UML was adopted during this stage and visual paradigm was the program used to draw all the UML diagrams. It was logical for the writer to utilise his skills and experience using Visual Paradigm to produce UML rather than learning a new system which could required additional time and compromise the project.

8) Prototype

9) Learning of new language C# and front end environment Windows Forms.

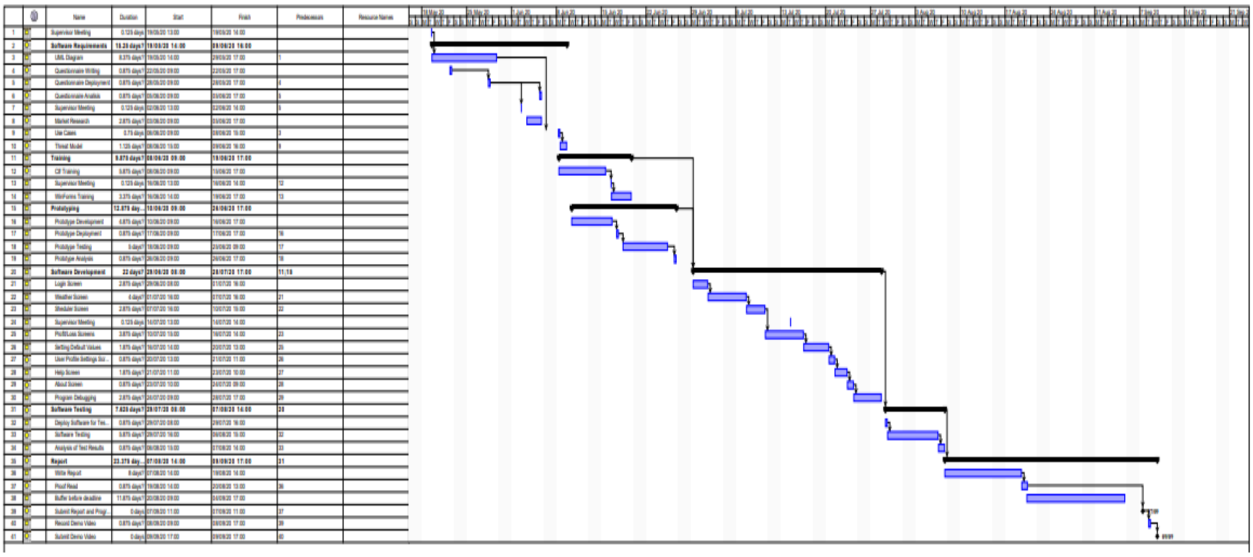
10) Creation of the application (hard coding)

11) Testing

12) On-going meetings with Craig Ramsay. These meetings focused on providing the writer with advice on the project, discuss potential obstacles and weaknesses, advice on the language used to write the program, support during the design program an coding when difficulties were experienced (Appendix 9)

These Stages can also be visualised in the following GANTT Chart:

Figure 12 – GANTT Chart



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		Supervisor Meeting	0.125 days	19/05/20 13:00	19/05/20 14:00		
2		**Software Requirements**	**15.25 days?**	**19/05/20 14:00**	**09/06/20 16:00**		
3		UML Diagram	8.375 days?	19/05/20 14:00	29/05/20 17:00	1	
4		Questionnaire Writing	0.875 days?	22/05/20 09:00	22/05/20 17:00		
5		Questionnaire Deployment	0.875 days?	28/05/20 09:00	28/05/20 17:00	4	
6		Questionnaire Analisis	0.875 days?	03/06/20 09:00	03/06/20 17:00	5	
7		Supervisor Meeting	0.125 days	02/06/20 13:00	02/06/20 14:00	5	
8		Market Research	2.875 days?	03/06/20 09:00	05/06/20 17:00		
9		Use Cases	0.75 days	08/06/20 09:00	08/06/20 15:00	3	
10		Threat Model	1.125 days?	08/06/20 15:00	09/06/20 16:00	9	
11		**Training**	**9.875 days?**	**08/06/20 09:00**	**19/06/20 17:00**		
12		C# Training	5.875 days?	08/06/20 09:00	15/06/20 17:00		
13		Supervisor Meeting	0.125 days	16/06/20 13:00	16/06/20 14:00	12	
14		WinForms Training	3.375 days?	16/06/20 14:00	19/06/20 17:00	13	
15		**Prototyping**	**12.875 day...**	**10/06/20 09:00**	**26/06/20 17:00**		
16		Prototype Development	4.875 days?	10/06/20 09:00	16/06/20 17:00		
17		Prototype Deployment	0.875 days?	17/06/20 09:00	17/06/20 17:00	16	
18		Prototype Testing	5 days?	18/06/20 09:00	25/06/20 09:00	17	
19		Prototype Analysis	0.875 days?	26/06/20 09:00	26/06/20 17:00	18	
20		**Software Development**	**22 days?**	**29/06/20 08:00**	**28/07/20 17:00**	**11,15**	
21		Login Screen	2.875 days?	29/06/20 08:00	01/07/20 16:00		
22		Weather Screen	4 days?	01/07/20 16:00	07/07/20 16:00	21	
23		Sheduler Screen	2.875 days?	07/07/20 16:00	10/07/20 15:00	22	
24		Supervisor Meeting	0.125 days	14/07/20 13:00	14/07/20 14:00		
25		Profit/Loss Screens	3.875 days?	10/07/20 15:00	16/07/20 14:00	23	
26		Setting Default Values	1.875 days?	16/07/20 14:00	20/07/20 13:00	25	
27		User Profile Settings Scr...	0.875 days?	20/07/20 13:00	21/07/20 11:00	26	
28		Help Screen	1.875 days?	21/07/20 11:00	23/07/20 10:00	27	
29		About Screen	0.875 days?	23/07/20 10:00	24/07/20 09:00	28	
30		Program Debugging	2.875 days?	24/07/20 09:00	28/07/20 17:00	29	
31		**Software Testing**	**7.625 days?**	**29/07/20 08:00**	**07/08/20 14:00**	**20**	
32		Deploy Software for Tes...	0.875 days?	29/07/20 08:00	29/07/20 16:00		
33		Software Testing	5.875 days?	29/07/20 16:00	06/08/20 15:00	32	
34		Analysis of Test Results	0.875 days?	06/08/20 15:00	07/08/20 14:00	33	
35		**Report**	**23.375 day...**	**07/08/20 14:00**	**09/09/20 17:00**	**31**	
36		Write Report	8 days?	07/08/20 14:00	19/08/20 14:00		
37		Proof Read	0.875 days?	19/08/20 14:00	20/08/20 13:00	36	
38		Buffer before deadline	11.875 days?	20/08/20 09:00	04/09/20 17:00		
39		Submit Report and Progr...	0 days	07/09/20 11:00	07/09/20 11:00	37	
40		Record Demo Video	0.875 days?	08/09/20 09:00	08/09/20 17:00	39	
41		Submit Demo Video	0 days	09/09/20 17:00	09/09/20 17:00	40	

The reader can verify the Gant Chart with more detail in Appendix 3.

3.2 Requirements and Use Cases

To define the software requirements, the writer started by establishing the stakeholders of this project which were the users (farmers/business owners), System and Scottish Government. By requirements, we mean a complete description of what the system needs to do such as the different stages of the program, different features and constrains (Ramsay, 2020). To identify the needs of the stakeholders, the writer looked into his personal and professional experience and used questionnaire technique (Appendix 1). All requirements were reviewed and analysed by taking into account FURPS +. The requirements were then incorporated into Use Cases. The Use Cases allows us to understand where the system starts, how the requirements fit together and how to perceive the user's perspective (Ramsay, 2020). The following text provides an example of some of the Requirements and Use Case established in this project. For the whole Requirements and Uses Cases information please refer to appendices 3 Project Requirements and 4 Project Use Cases. Diagram 1 introduces the Use Cases diagram.

Example of Functional and hierarchical requirements:

R2 Log in Procedure

Non functional requirements

R2.1 – User licence quantity

Description - The system should allow only one person to have access to the system at a given time.

Rationale – The system aims to be used by the business owner/manager who will be able to share the information/access with people who he/she chooses (normally the wife or son/daughter). This kind of task is normally managed by one or two people in the field. The access to one single authorised user also aims to guarantee security and concurrency.

R2.2 Password Structure
Description – The password should be definable by user
Rationale – Ensure security and allow the password to be memorable.

Functional Requirements

R2.3 First time access to the system
Description – The software shall ask for the service user to define a user name and a password to have future access.
Rationale – The usage of a user name and password provides security to the system.

R2.4 Definition of the business Profile
Description – Following the creation of the password, the system shall ask the user to introduce information associated with the business.
Rationale – This provides information which can be used for future calculations.

R2.5 Access to the system

Description - The system should allow access to one person by requesting the introduction of a user name and a password.
Rationale: This requirement is important as it involves the security of the system and to ensure that non-authorise people have access to the information of the business.

R2.6 - **Authorisation to Access the system**
Description - Once the service user has introduced the password, the system shall authenticate the details before allowing access.
Rationale – This requirement is important as it involves the security of the system and to ensure that non-authorise people have access to the information of the business.

R2.7- Input of wrong password and user name
Description – the program shall display a message informing that the incorrect details have been entered.
Rationale - keep the user informed as to what it is happening at various stages of the log in process.

Use Case and associated Alternative Flow

Use Case 1 - Login

Actors
The following actors are involved in this use case: User and System.

Brief Description
This use case describes the process of logging in to the system

Basic Flow of Events

The use case begins when the user selects the program from the computer icon.

User	Selects the FarmField icon on the computer desktop
System	Loads Program and displays Login screen
User	Enters Login details (username and password)
System	Authenticates login details
System	Allows access if Login Details correct (A1)

Alternative Flow A1 Incorrect Login details (username and password)

Brief Description

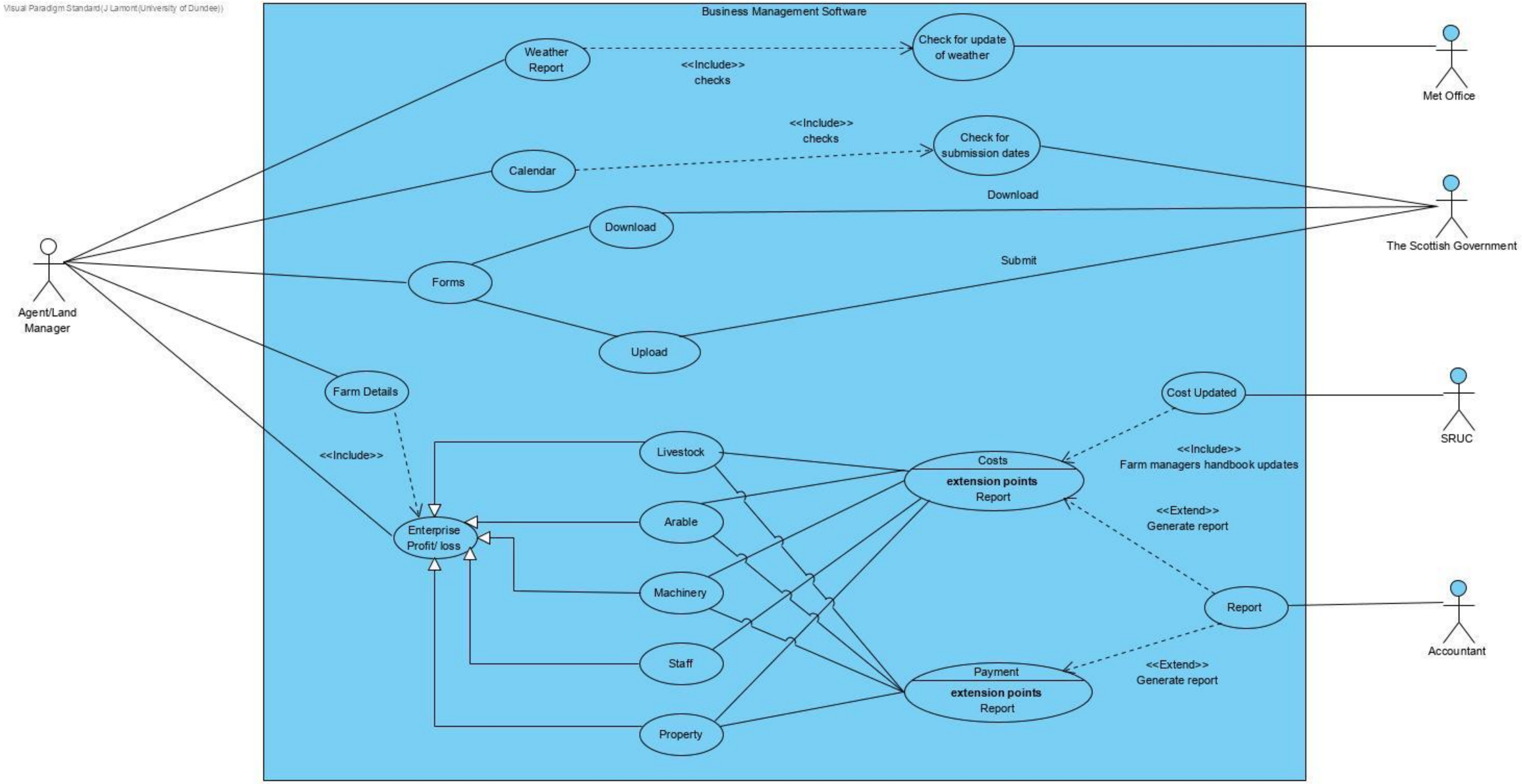
This describes the process of logging in to the system if incorrect details are entered

Basic Flow of Events

The use case begins when the user introduces incorrect details are entered

System	Authenticates login details
System	Incorrect details entered – Access denied
System	Request to re-enter username and password

Figure 13 – Use Case Diagram



The Use Case diagram demonstrates the relationships between Use Cases in this project. For example, the Use Case Weather Report has an Included Relationship with Check for Update of Weather Use Case. An Included Relationship shows dependency between a Base Use Case and an Included Use Case. When the Base Use Case is executed, the Included Use Case is also executed. The Base Use Case requires the information from the Included Use Case to be completed (Lucidchart.com). Thus, information about weather report will always require updated information from a weather forecast service.

The diagram also shows the existence of Extended Relationships. This is observed between the Report Use Case and the Profit Loss Calculation. Extended Relationships mean that when the Base Use Case is executed, the Extended Use Case is only executed if certain criteria are met (Lucidchart.com). In the context of this project, the Report Use Case is only completed if requested.

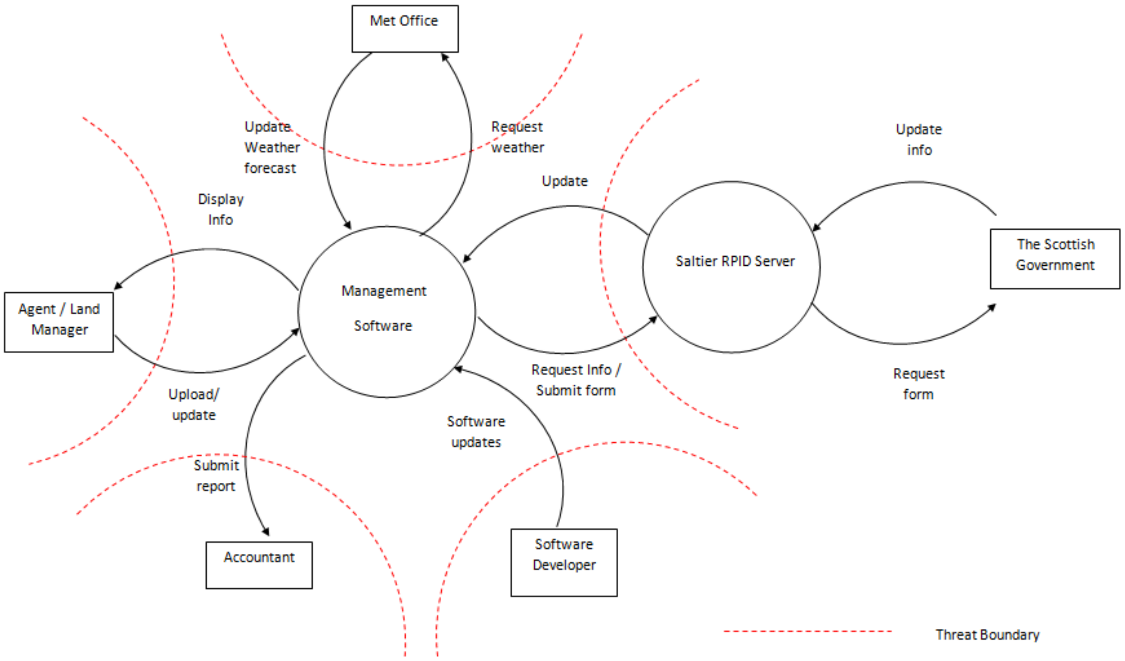
3.3. Threat Model

The development of a Threat Model by using STRIDE was the approach adopted in this project. The model was developed following the creation of the Use Cases to highlight the areas in which security threats were present. By adopting this model at this stage, the writer was able to identify early potential threats caused by the interface between the internal and external systems, incorporate security measures into the system and develop a more robust and well structural solution and, therefore, a better final product (based on Ramsay, 2020). The following example shows how some of the threat areas for the User Login were identified and analysed (see the full list of threats in to Appendix 5 Threat Model):

- **User login:** land manager, agent and business owner.
    - **Information disclosure** - While the authorised person has access to the system, other non-authorised personnel could see the username, password or sensitive information about the business if not appropriately secured. **Countermeasure:** masking password when entering the password details and provide the opportunity to change/update password as and when required.
    - ▪ **Elevation privilege** – Someone who should not have authorised access to the system has managed to get the username and password through unauthorised means (e.g. stolen password and username). This would be a violation of the employee’s moral conduct and lack of complacency with business safety polices. **Countermeasure:** masking password when entering the password details and provide the opportunity to change/update password as and when required.
    - ▪ **Elevation of Privilege** - an employee who has been dismissed from his duties may continue to hold the user name and password, which may result in the program being operated without authorization from the authorised person. **Countermeasure:** The program provides the opportunity to change or update password as and when required.

Following Diagram, demonstrates a graphic representation of the threat model:

Figure 14– Threat Model Diagram



3.4 Class diagram and Sequence Diagrams

Following the Threat Model, the Class diagram which defines the Class and their associations was developed. The full Diagram Class can be seen in Appendix 6 Class Diagram. The diagram involves 61 Entity Classes which represent various aspects of the business such as Weather, Calendar and Profit Loss. The Class diagram also incorporates two Boundary Classes which handle the communication between the real world and the system (Met Office Weather Update and Scottish Government). Home Screen was developed as a Control Class as it coordinates events within the system. Home Screen also assumes a delegated control which means that each class that sequences the Home Screen Class takes a well rounded responsibility for performing tasks regarding information and performance. A Delegated Controlled Class was chosen as it facilities changes when required. This means that as a developer, the writer could change a class without affecting other parts of the program (based on Ramsay, 2020). Each Class has its own attributes which define the responsibility of each class. The attributes are focused on the purpose of the class.

The Diagram also incorporates different relationships and associations. For Example, Met Office has a reflexive relationship with Meat Office Weather Update as the Class Met Office was developed by incorporating a single association which involves only one class of object. There are also several associations by Composition such as Carrots which is a component of Root Crops which in turn is a component of Arable. This means that Carrots would not exist as an object without Root Crops and Arable Classes. These classes are represented by a filled diamond at the assembly end.

Harvesting had a unidirectional dependency with Fuel and Servicing Repair. This means that Harvesting comes temporarily in contact with Fuel and Servicing Repair at some point during the year, but not always.

The Diagram also shows a strong top down inheritance between classes with more generalised classes associated with more specific forms (children) which share some of the attributes of the parent class, but maintain their individuality. For example, Sheep Class is the child class of Livestock Class, thus Sheep inherits the attributes of Livestock (e.g. cost, bedding cost and medication), but also has it owns attributes (e.g. purchase price and sale price). There is also a strong polymorphism component as several classes have same methods/function, but specific to them.

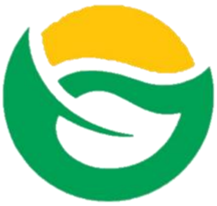
Following the Class Diagram, the Sequence Diagrams were also developed for each process. The Sequence Diagram describe the behaviour expected from Systems over time and provides a static view of the system structure such as the objects, classes, attributed, relationships and responsibilities attached to the different classes(Ramsay, 2020). The Sequence Diagram can be seen on Appendix 7 Sequence Diagrams.

3.5 Name and Logo

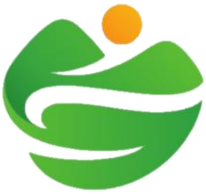
The name of the project was chosen following some reflection on potential words which were associated with the industry (e.g. farm, crops, field, environment, planning, sun, animals, etc). This reflection led the writer to adopt Farm and Field together as they were general worlds which represent several areas in the industry. The word field was also thought to be an appropriate one as it encompasses farming and technology: field as agricultural field and data field to which the user would be entering data as part of the program. Once the words were put together, the writer then googled the name of the project to ensure that there was no other software with a similar name. No other software was found with an identical name which then allowed the writer to adopt this name for his project.

In order to develop the logo, the writer looked for websites which would aid him to create a logo to represent the project. The website consulted was Brandcrowd.com. The name Farm Field was introduced in the search option of this website which provided a considerable number of potential logo candidates. Two logos were chosen due to the feeling of contemporary and modern design and their colours which represented adequately the concept of the project. The main colours of the logo were green due to be the predominate colour in agriculture and nature and orange which represented the sun (an important element in the agricultural field). These two colours together were also found to form the basis of a colour scheme which would be eye catching and appropriate to the field.

Figure 15 – Logos



Logo 1 Green leaf sun by Town



Logo 2 Leaf Mountain by Town

The two options were circulated among nine people who were asked to choose their favourite logo out of the two options. These participants were all known to the writer and they were all somehow related to farming/agriculture. Logo number two was selected due to a two to seven majority vote. This logo became the main Icon and its colours set the colour scheme of the whole program.

3.6 Conclusion

The design of this project reflects the knowledge acquired by the writer during the academic year 2019/20 in Software Engineering and his skills in this field. The software design was an important stage as it allowed the specification of the whole application including the principles and the architecture of the project which in turn facilitated the creation and implementation of the prototype, coding and front end environment as we will see in chapter 6. Specifically, this chapter demonstrated the rationale behind the creation of the requirements which were built based on the background research, personal and professional experience of the writer and the questionnaires implemented. It also highlighted the reasoning behind the Use Classes, Threat Model, and Class and Sequence diagrams. Concluding, this section will give the reader a comprehensive understanding of the engineering and architecture behind the real interface.

4 Software Design: Prototype

4.1 Introduction

Following the creation of the class diagram and the logo, the author proceeded to the development of a high fidelity prototype which allowed participants to trial the functionality of the program before committing to the final design. The development of the Prototype also assisted the writer to establish the architecture of the program and test some of the features initially thought in order to understand what should be included in the final application. This was a very useful stage as it helped to visualised the application before starting coding the final version. Although time was spent developing the prototype, this end up saving time during the construction of application itself as it led the writer to firmly establish his direction and vision of the application before progressing. This section describes this stage and the outcomes of trialling the prototype which led to some changes in initial design.

4.2 The creation of the prototype

Axure RP9 was chosen as the program to develop the prototype due to the author's previous experience with this software. No other prototype software was looked into due to the short time frame to develop the project to which the writer decided to use and maximise the knowledge already required. The next images show examples of the prototype pages:

Figure 16 – login page

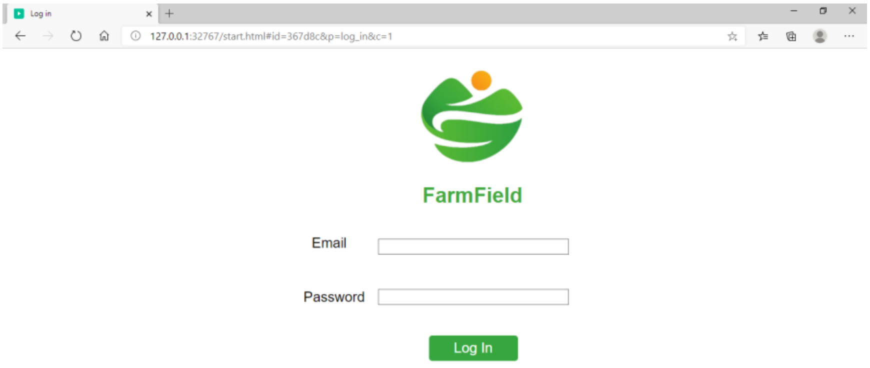
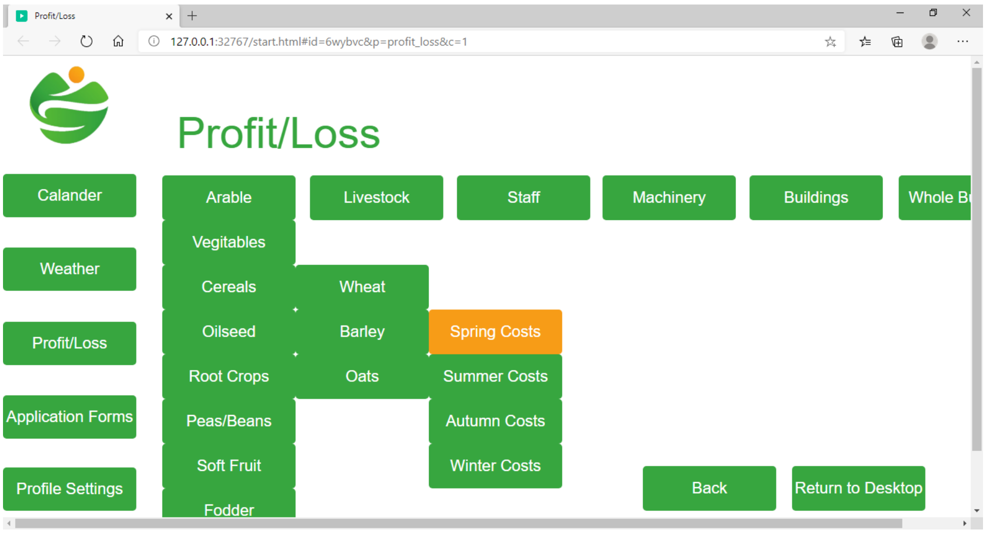


Figure 17 – Prototype Menu Page



Following completion, the prototype was firstly tested by the writer and no changes were made. However, in order for the prototype to be properly tested, the writer distributed the prototype to a sample of farmers based on different locations across Scotland in order to alleviate any bias. Seven people, who had also taken part in the questionnaires, tested the prototype. The prototype was distributed in an HTML format which could be viewed using the participants' web browser with the accompaniment of a demonstration video which showed how to use the software. Telephone support was also provided and contact was made by phone after the trial period of seven days. The writer decided to collect the information from the testing by phone as this felt to be the best way to ensure that participants would engage and not forget about the trial which could easy happen if they were left with a questionnaire to fill in. An informal conversation is always more desirable within the farming community rather than documentation. This strategy was also better suited as people would not be concerned with literacy problems and also gave control to the writer over the timescale to collect the information. Some of the comments provided by these participants were:

- Participants advised that a two day weather forecast would be more useful compared with the single day forecast presented in the prototype as this would allow some forward planning.
- Participants also advised that they would like the deadlines incorporated in the calendar itself as well as with support information available externally to the calendar. This would allow participants to visualise immediately what is happening during the month and when dead lines require to be submitted.
- Participants expressed that establishing the costs for each season for every crop was overcomplicated and confusing. Participants were more interested in the cost of the financial year per crop rather than season. This is due to the fact that although crops are harvested seasonally, their cost are generally carried over the year as farmers required to buy the seed, wait to plough the fields, apply pesticides, pay for agronomist and harvest and sell their product throughout several months (six months of more).
- Concerns were expressed if users forgot their username and password as the prototype did not have any recovery process for password and username. This was more of an oversight of the writer then an intended omission.

4.3 Conclusion

The aim of the prototype was to test the usability and functionality of the program. The writer recognises that the prototype has limitations as a consequence of the program Axure being used, the capabilities of the writer and also due to the time allocated to the project. However, the writer found this to be a very useful stage as by developing a functional prototype, the writer was be able to find how people were navigating the program and interface and how they perceived the initial concept of the project. Additionally, the prototype was also very useful to reflect on the structure of the program, including the usage of the language to be employed and potential changes which aimed to increase the functionality of the application.

5 – Software Design: Coding and Front End Environment

5.1 Introduction

This chapter introduces the choice of the language adopted by the writer and the front end development environment which was also chosen to complete this project. The section also provides an explanation to why the writer adopted C# and WinForms and identifies some of the challenges that the author of this program encountered during the development of the coding stage. It is important to highlight that this chapter only shows some parts of the code created. Thus, for a full appreciation of the work carried out in terms of coding it is advisable for the reader to look into the program itself.

5.2 The choice of the Language and Front End Environment

During the MSc course the writer learnt C++ as a Programming Language; however, within the context of a desktop application, other languages could be used more efficiently. C++ is an excellent language; however, as it was shown in class, it requires a high level of complexity involving pointers which are not always the best coding choice due to the memory required to define functions. This was corroborated in discussion with the project supervisor where the writer concluded that using C++ with a front end application could be very complex with additional code, memory overload and decreased efficiency; specially when there is less complex, but more functional and efficient languages which could be used with a front end application.

Following the advice acquired from supervision and research undertaken, the writer opted for the use of C# together with Windows Forms. This led the writer to learn a completely new language which although was a challenge due to the time frame of the project, was also a very interesting journey. During the research undertaken, C# revealed to be a more functional and effective language due to its rapid application development compared with C++ as it does not require the same amount of coding to achieve the same tasks due to in-built garbage handling and more extensive library. These features also allow easy maintenance and write code faster. However, C# is not without it disadvantages. For example, C# is known for being slower than C++; however, the compromise performance is not considered a major issue for FarmField as this program does not require tasks to be done fast due to the nature of the calculations developed which should be revealed to the user once the data has been manually introduced. Another disadvantage of C# is the lack of flexibility as functions not found in .Net Framework are difficult to implement; however, this was not problem in the development of FarmField as all the functions required were found without issues [3].

Windows Forms was chosen as the front end development environment. This open source graphical library allowed the writer to quickly and simply build a graphical user interface

[3] https://chiafong6799.wordpress.com/2006/07/11/advantages-and-disadvantages-of-c-as-compared-to-c/

which housed the program created. This is also an application which can be used with and without being linked to the internet. Additionally, Windows Forms could be easily used with Visual Study compared with WPF[4] which would require more work due to its complexity which could become an obstacle to the writer who was attempting to produce a program with a graphic interface with C# for the first time. Finally, the writer also felt that Windows Forms seems to have a better community where it was easy to find materials to learn and overcome doubts and obstacles. Although the writer recognises that by using WPF more functionality, better design and individual controls could have been achieved, the outcome of the project by using C# and Windows Form met the expectations of the writer.

5.3 The coding Process

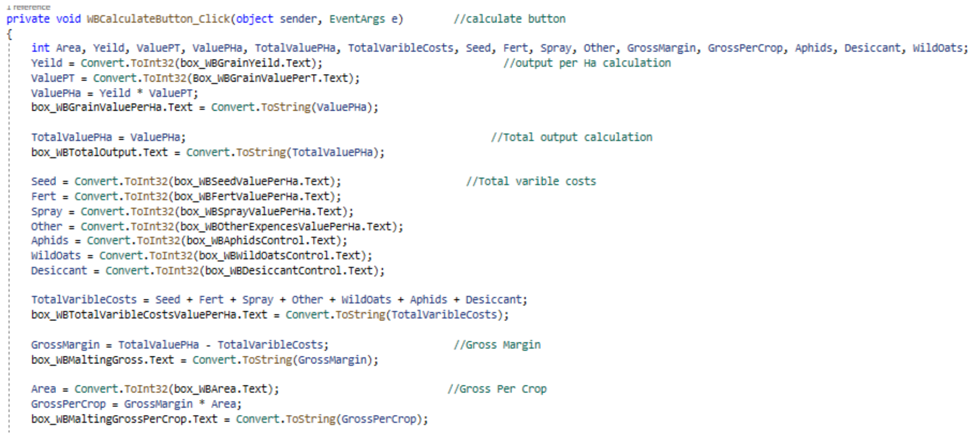
This section aims to explain the development of the coding itself. The program comprises 7 forms and 47 user controls. The writer started by trying to understand how to display multiple different interfaces within one base screen. To do so, the writer created a desktop form which would shape the base of the program. The desktop form contains the navigation buttons and an area to which user controls would be displayed. The buttons controls used were already incorporated into Visual Studio and Windows Forms. Following this, individual user controls were created to be displayed on this main screen form. These individual user controls were employed by using specific data fields which were made of label controls and textbox controls which were specific to each individual enterprise represented in the application. This process formed the basis of the user interface which would house the necessary coding to provide the desired functionality.

Once the forms and interface were created, the writer proceed by given functionality to each user control. The writer decided to target the areas which he felt to be more complex for him; those being the calendar and the weather forecast. In order to create the Calendar the writer used an alternative control sourced externally from Windows forms (please see challenges below) which allowed more specific coding to meet the requirements established by the participants who tested the prototype. Following this, the writer proceed with the Weather Forecast Control by researching how to establish the methodology to successfully display the weather forecast on the form. This was achieved by using JSON NuGet to transfer the data from a weather API to the program.

The calculations for the main feature of the program (Profit/Loss Calculation) were arguably the most straight forward procedure, although transferring the calculated values between user controls had its own challenges. The following figure provids an example of the code used for a calculation:

[4] https://www.wpf-tutorial.com/about-wpf/wpf-vs-winforms/

Figure 18 – Example of Coding for Calculation of Enterprises

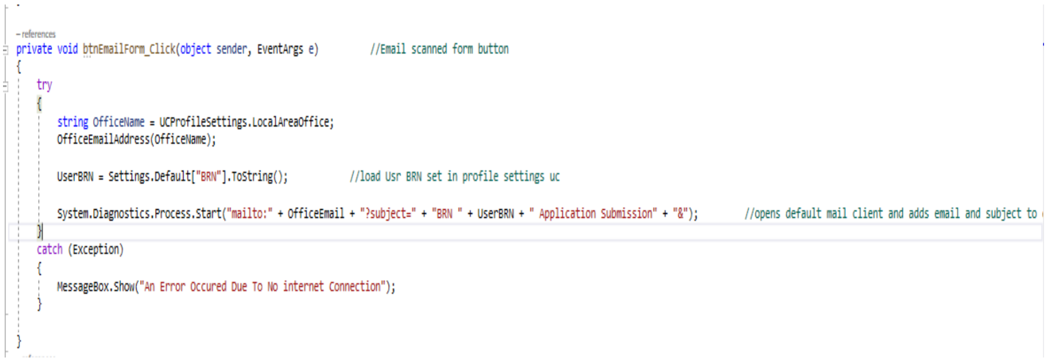


```
1 reference
private void WBCalculateButton_Click(object sender, EventArgs e)        //calculate button
{
    int Area, Yeild, ValuePT, ValuePHa, TotalValuePHa, TotalVaribleCosts, Seed, Fert, Spray, Other, GrossMargin, GrossPerCrop, Aphids, Desiccant, WildOats;
    Yeild = Convert.ToInt32(box_WBGrainYeild.Text);                                    //output per Ha calculation
    ValuePT = Convert.ToInt32(Box_WBGrainValuePerT.Text);
    ValuePHa = Yeild * ValuePT;
    box_WBGrainValuePerHa.Text = Convert.ToString(ValuePHa);

    TotalValuePHa = ValuePHa;                                                         //Total output calculation
    box_WBTotalOutput.Text = Convert.ToString(TotalValuePHa);

    Seed = Convert.ToInt32(box_WBSeedValuePerHa.Text);                            //Total varible costs
    Fert = Convert.ToInt32(box_WBFertValuePerHa.Text);
    Spray = Convert.ToInt32(box_WBSprayValuePerHa.Text);
    Other = Convert.ToInt32(box_WBOtherExpencesValuePerHa.Text);
    Aphids = Convert.ToInt32(box_WBAphidsControl.Text);
    WildOats = Convert.ToInt32(box_WBWildOatsControl.Text);
    Desiccant = Convert.ToInt32(box_WBDesiccantControl.Text);

    TotalVaribleCosts = Seed + Fert + Spray + Other + WildOats + Aphids + Desiccant;
    box_WBTotalVaribleCostsValuePerHa.Text = Convert.ToString(TotalVaribleCosts);

    GrossMargin = TotalValuePHa - TotalVaribleCosts;                             //Gross Margin
    box_WBMaltingGross.Text = Convert.ToString(GrossMargin);

    Area = Convert.ToInt32(box_WBArea.Text);                                    //Gross Per Crop
    GrossPerCrop = GrossMargin * Area;
    box_WBMaltingGrossPerCrop.Text = Convert.ToString(GrossPerCrop);
```

The calculations were all done in C# and once they were completed the writer linked these calculations to the Whole Business Calculation user control by using C#. Following this, the writer carried on by establishing the Application Forms user control. This user control allowed the user to have direct access to specific governmental application forms for agricultural/rural subsidies. This was a very straight forward coding procedure which involved calling the default web browser and insert a predefine URL which would be loaded by the browser.

Still part of the Application Forms user control, the method employed to allow the user to email an application form required the ability to call upon details predefined within the profile settings to insert relevant details into an email created by the default mail client. The next figure shows the code for this process:

Figure 19 – Email data transfer



```
- references
private void btnEmailForm_Click(object sender, EventArgs e)        //Email scanned form button
{
    try
    {
        string OfficeName = UCProfileSettings.LocalAreaOffice;
        OfficeEmailAddress(OfficeName);

        UserBRN = Settings.Default["BRN"].ToString();          //load Usr BRN set in profile settings uc

        System.Diagnostics.Process.Start("mailto:" + OfficeEmail + "?subject=" + "BRN " + UserBRN + " Application Submission" + "&");        //opens default mail client and adds email and subject to
    }
    catch (Exception)
    {
        MessageBox.Show("An Error Occured Due To No internet Connection");
    }

}
```

One of the final steps was the creation of the Help videos which was done by using Windows Media Player control and by linking this to MP4 files which contained various recordings. This involved C# coding to link the media Player to MP4 files to allow them to be played.

Finally, the last coding step in this application was focused on the password recovery process. This was an important step as participants who tested the prototype advised that they would want this feature incorporated within the program. To achieve this, the writer set up an email account dedicated to this process. This feature was a challenge for the writer which the reader will be able to understand deeper in the next section which describes the challenges faced by the writer during the design processes.

5.4 – The Challenges

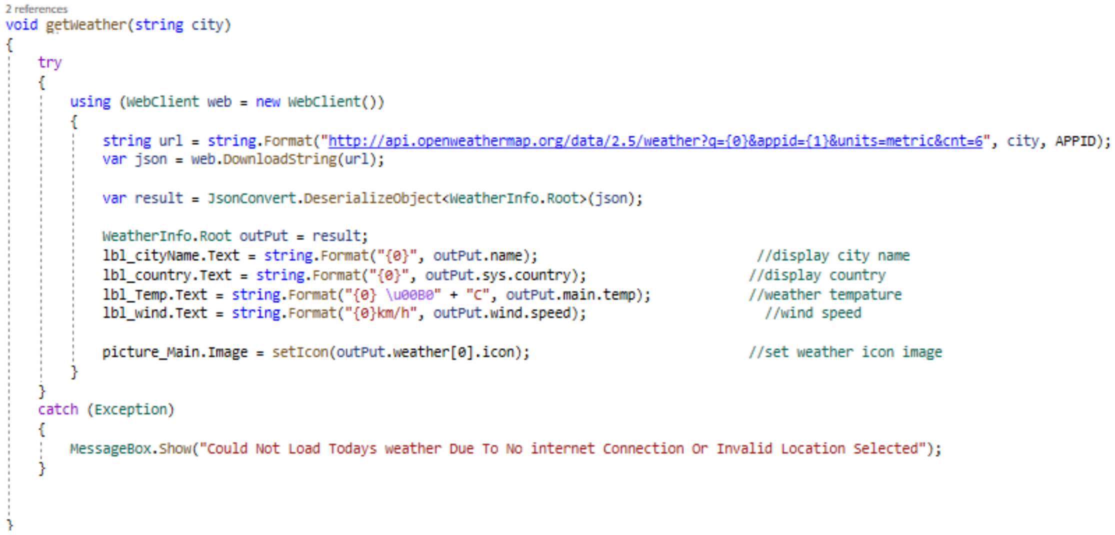
The main page is the desktop form where the main user controls are displayed. T

A different page was set for the dialogue box which displays notifications throughout the all program. The followings forms are related to the login process, four of which involve the management of the username and password.

Some of the coding issues encountered were as follows:

1) Linking the weather app to an API became a challenge as the writer did not have the knowledge to carry out this task. The first attempts to write the code were not successful as the writer could not transfer data from the API to the program. However, after extensive research in the internet, the writer used a JSON NuGet to transfer the data from a weather API to the program. This successfully transferred the data wanted to the application. Following this the weather app became functional giving the possibility to the user to choose the weather location. The next piece of code, shows the implementation of the JSON NuGet Package:

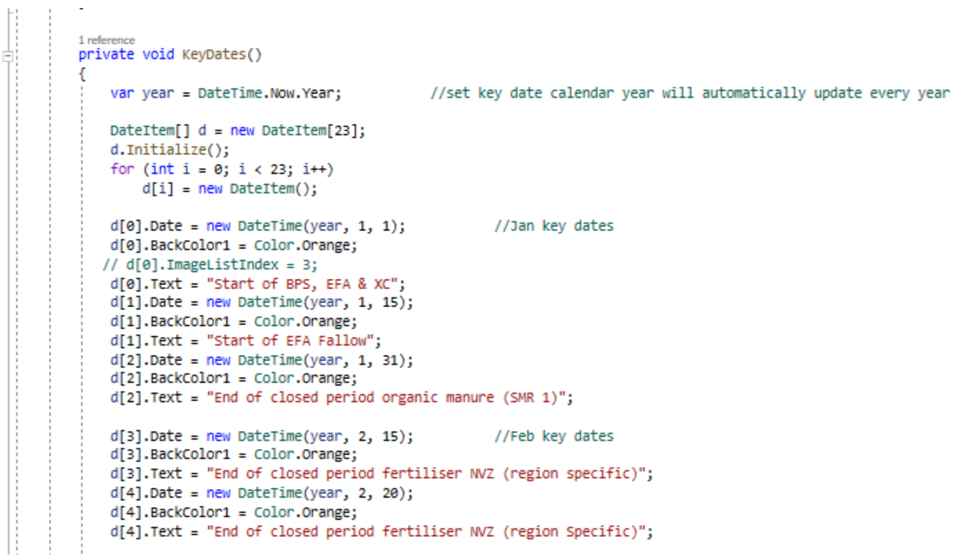
Figure 20 – Code for Weather Application



```
2 references
void getWeather(string city)
{
    try
    {
        using (WebClient web = new WebClient())
        {
            string url = string.Format("http://api.openweathermap.org/data/2.5/weather?q={0}&appid={1}&units=metric&cnt=6", city, APPID);
            var json = web.DownloadString(url);

            var result = JsonConvert.DeserializeObject<WeatherInfo.Root>(json);

            WeatherInfo.Root outPut = result;
            lbl_cityName.Text = string.Format("{0}", outPut.name);                    //display city name
            lbl_country.Text = string.Format("{0}", outPut.sys.country);             //display country
            lbl_Temp.Text = string.Format("{0} \u00B0" + "C", outPut.main.temp);     //weather tempature
            lbl_wind.Text = string.Format("{0}km/h", outPut.wind.speed);               //wind speed

            picture_Main.Image = setIcon(outPut.weather[0].icon);                    //set weather icon image
        }
    }
    catch (Exception)
    {
        MessageBox.Show("Could Not Load Todays weather Due To No internet Connection Or Invalid Location Selected");
    }


}
```

2) Another problem encountered was the limitations associated with the calendar control available in WindForms. This Calendar Control would not allow the functionality to have the deadlines displayed in the calendar itself, rather than just dates. As an alternative, PABO control was used which was downloaded from the Code Project[5]. This control required to be customised to meet the requirements of the FarmField in order to work as per requirements. The following piece of code shows. Some of this customisations involved updating the dates every year for the deadlines of important forms by using the DateTime Function from C#. The following piece of code shows the implementation of calendar dates and DateTime:

Figure 21 – Code for Calendar application

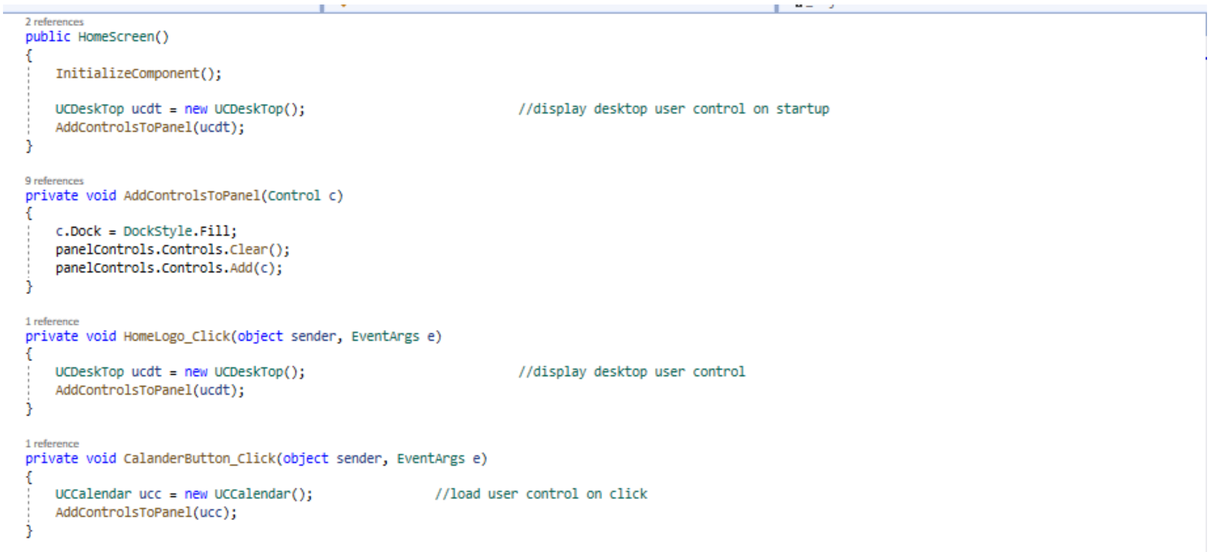


```
1 reference
private void KeyDates()
{
    var year = DateTime.Now.Year;          //set key date calendar year will automatically update every year

    DateItem[] d = new DateItem[23];
    d.Initialize();
    for (int i = 0; i < 23; i++)
        d[i] = new DateItem();

    d[0].Date = new DateTime(year, 1, 1);          //Jan key dates
    d[0].BackColor1 = Color.Orange;
   // d[0].ImageListIndex = 3;
    d[0].Text = "Start of BPS, EFA & XC";
    d[1].Date = new DateTime(year, 1, 15);
    d[1].BackColor1 = Color.Orange;
    d[1].Text = "Start of EFA Fallow";
    d[2].Date = new DateTime(year, 1, 31);
    d[2].BackColor1 = Color.Orange;
    d[2].Text = "End of closed period organic manure (SMR 1)";

    d[3].Date = new DateTime(year, 2, 15);         //Feb key dates
    d[3].BackColor1 = Color.Orange;
    d[3].Text = "End of closed period fertiliser NVZ (region specific)";
    d[4].Date = new DateTime(year, 2, 20);
    d[4].BackColor1 = Color.Orange;
    d[4].Text = "End of closed period fertiliser NVZ (region Specific)";
```

[5] https://www.codeproject.com/Articles/10840/Another-Month-Calendar

3) Another issue the writer encountered was the ability to display different screens on a standardise desktop screen. Initially the writer thought about creating different forms to achieve the main screen, but this would lead to a repetition of controls and bad coding practices. After some research, the writer decided to adopt user controls which allowed the desktop form to display different screens without the need for repetition. The following piece of code illustrates how the writer achieved this:

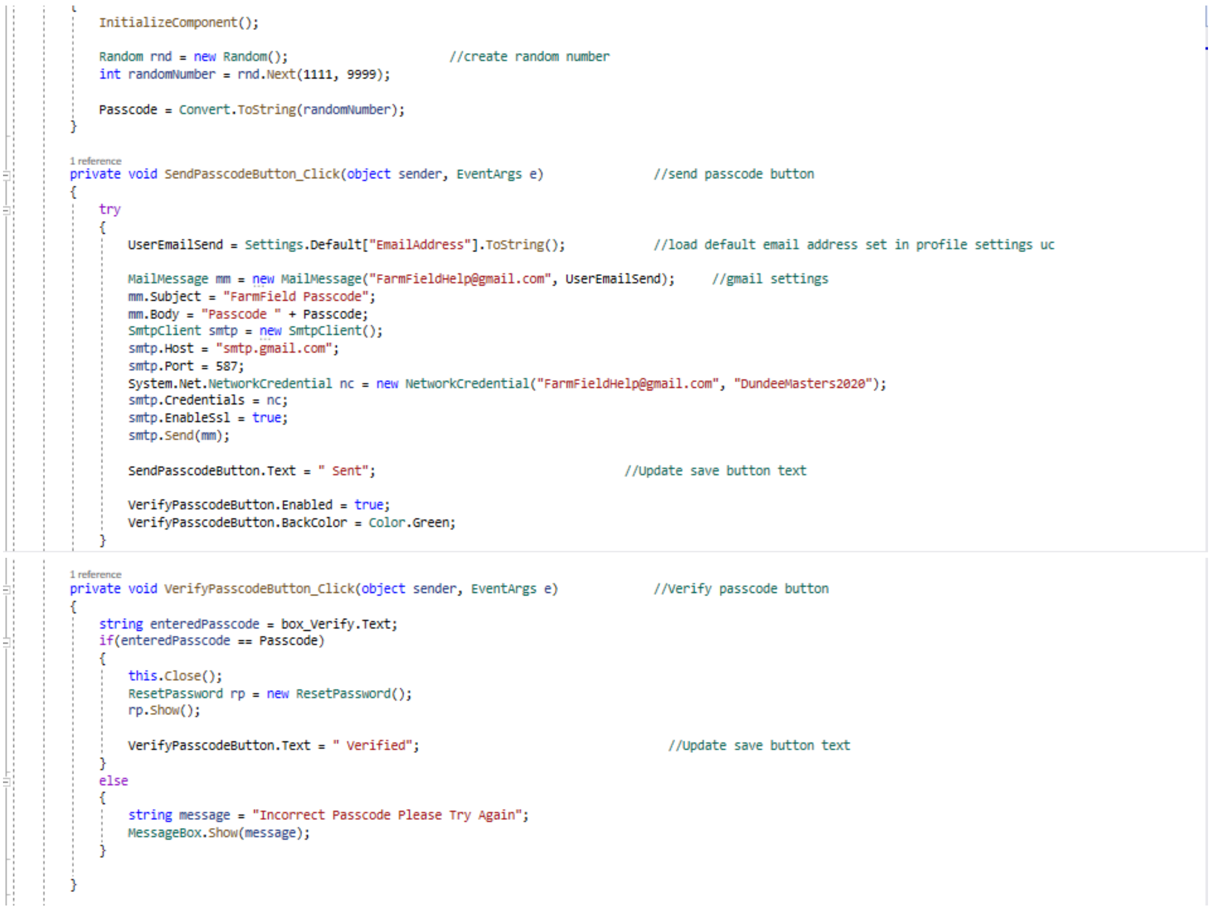
Figure 22 – Code to Display of a Standardise Desktop



```
2 references
public HomeScreen()
{
    InitializeComponent();

    UCDeskTop ucdt = new UCDeskTop();                       //display desktop user control on startup
    AddControlsToPanel(ucdt);
}

9 references
private void AddControlsToPanel(Control c)
{
    c.Dock = DockStyle.Fill;
    panelControls.Controls.Clear();
    panelControls.Controls.Add(c);
}

1 reference
private void HomeLogo_Click(object sender, EventArgs e)
{
    UCDeskTop ucdt = new UCDeskTop();                       //display desktop user control
    AddControlsToPanel(ucdt);
}

1 reference
private void CalanderButton_Click(object sender, EventArgs e)
{
    UCCalendar ucc = new UCCalendar();              //load user control on click
    AddControlsToPanel(ucc);
}
```

4) In response to the concerns raised by the participants who tested the prototype, the writer was required to create a system which would allow the recovery of the username and password if the user had forgotten their login details. This was achieve by an email authentication process by which a random number is created by the program and sent to the users' email who then enter this number in the program to authenticate their identity. This was a challenge as the writer was required to give some thought on how to find a safe process which did not impose security issues and also the method code for such a task. The final outcome was achieved by setting up a designated email account to send the user passcode and then this email account was hard coded into the program. The following piece of code shows how the writer achieve this task:

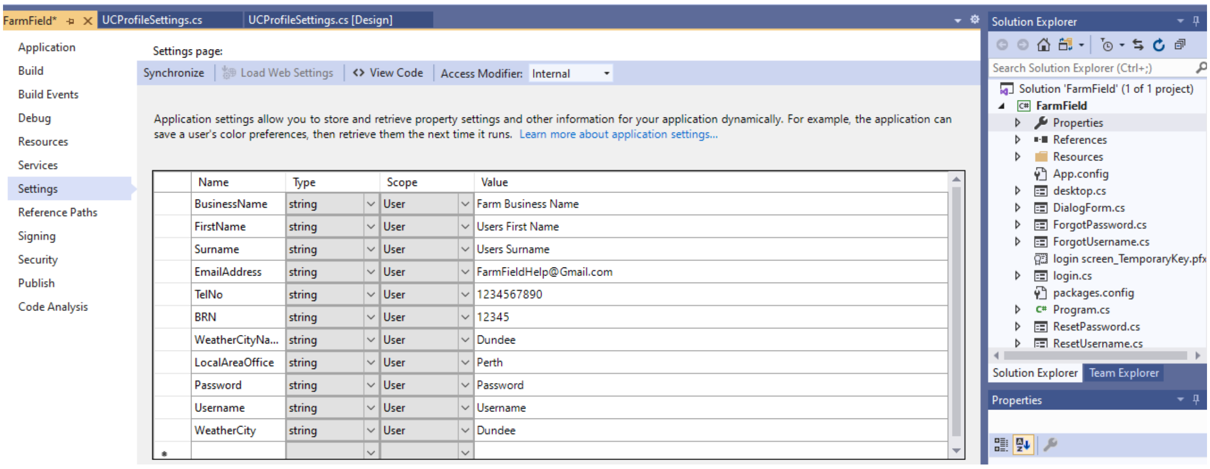
Figure 23 – Code to recover username and password



```
{
    InitializeComponent();

    Random rnd = new Random();                      //create random number
    int randomNumber = rnd.Next(1111, 9999);

    Passcode = Convert.ToString(randomNumber);
}

1 reference
private void SendPasscodeButton_Click(object sender, EventArgs e)              //send passcode button
{
    try
    {
        UserEmailSend = Settings.Default["EmailAddress"].ToString();           //load default email address set in profile settings uc

        MailMessage mm = new MailMessage("FarmFieldHelp@gmail.com", UserEmailSend);     //gmail settings
        mm.Subject = "FarmField Passcode";
        mm.Body = "Passcode " + Passcode;
        SmtpClient smtp = new SmtpClient();
        smtp.Host = "smtp.gmail.com";
        smtp.Port = 587;
        System.Net.NetworkCredential nc = new NetworkCredential("FarmFieldHelp@gmail.com", "DundeeMasters2020");
        smtp.Credentials = nc;
        smtp.EnableSsl = true;
        smtp.Send(mm);

        SendPasscodeButton.Text = " Sent";                                  //Update save button text

        VerifyPasscodeButton.Enabled = true;
        VerifyPasscodeButton.BackColor = Color.Green;
    }

1 reference
private void VerifyPasscodeButton_Click(object sender, EventArgs e)            //Verify passcode button
{
    string enteredPasscode = box_Verify.Text;
    if(enteredPasscode == Passcode)
    {
        this.Close();
        ResetPassword rp = new ResetPassword();
        rp.Show();

        VerifyPasscodeButton.Text = " Verified";                                //Update save button text
    }
    else
    {
        string message = "Incorrect Passcode Please Try Again";
        MessageBox.Show(message);
    }

}
```

5) In order for the program to be functional a series of default values were required to be created and updated by the user. This was an important aspect of the basic architecture of the program as these variables were integral part of the main functionally of the system. Due to the unfamiliarity with C# and WindForms, it took some time for the writer to research how to achieve this. Following some research, the writer accomplished the outcomes desired. This achievement was accomplished by setting values within the properties of the project and then calling upon these setting values within the code. The following figures show how the this was attained:

Figure 24 – Code for default values

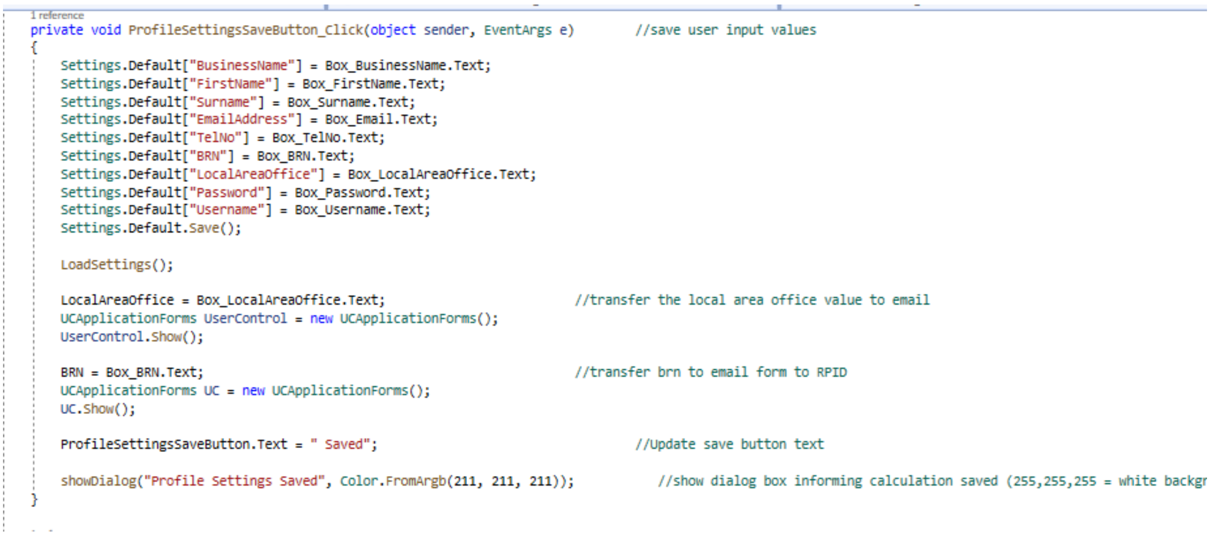


Settings page:

Application settings allow you to store and retrieve property settings and other information for your application dynamically. For example, the application can save a user's color preferences, then retrieve them the next time it runs. Learn more about application settings...

Name	Type	Scope	Value
BusinessName	string	User	Farm Business Name
FirstName	string	User	Users First Name
Surname	string	User	Users Surname
EmailAddress	string	User	FarmFieldHelp@Gmail.com
TelNo	string	User	1234567890
BRN	string	User	12345
WeatherCityNa...	string	User	Dundee
LocalAreaOffice	string	User	Perth
Password	string	User	Password
Username	string	User	Username
WeatherCity	string	User	Dundee

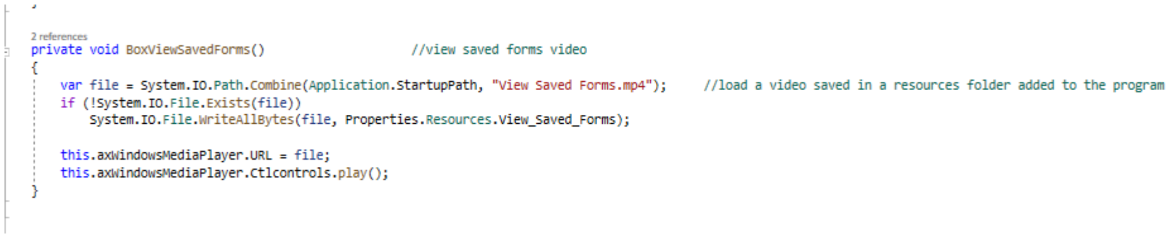
Figure 25 – Code default values



```
1 reference
private void ProfileSettingsSaveButton_Click(object sender, EventArgs e)        //save user input values
{
    Settings.Default["BusinessName"] = Box_BusinessName.Text;
    Settings.Default["FirstName"] = Box_FirstName.Text;
    Settings.Default["Surname"] = Box_Surname.Text;
    Settings.Default["EmailAddress"] = Box_Email.Text;
    Settings.Default["TelNo"] = Box_TelNo.Text;
    Settings.Default["BRN"] = Box_BRN.Text;
    Settings.Default["LocalAreaOffice"] = Box_LocalAreaOffice.Text;
    Settings.Default["Password"] = Box_Password.Text;
    Settings.Default["Username"] = Box_Username.Text;
    Settings.Default.Save();

    LoadSettings();

    LocalAreaOffice = Box_LocalAreaOffice.Text;                    //transfer the local area office value to email
    UCApplicationForms UserControl = new UCApplicationForms();
    UserControl.Show();

    BRN = Box_BRN.Text;                                            //transfer brn to email form to RPID
    UCApplicationForms UC = new UCApplicationForms();
    UC.Show();

    ProfileSettingsSaveButton.Text = " Saved";                             //Update save button text

    showDialog("Profile Settings Saved", Color.FromArgb(211, 211, 211));          //show dialog box informing calculation saved (255,255,255 = white backgr
}
```

6) Creating a help control was also of great interest in this program as it would give the user an easy method of problem solving. However, this control was also of interest to the writer as it would allow the practice of using MP4 files within a program. This was achieved by imbedding Widows Media Player into the program which would allow MP4 with help material to be played. The following image provides the code for this task:

Figure 26 – Code for Help Control



```
2 references
private void BoxViewSavedForms()                    //view saved forms video
{
    var file = System.IO.Path.Combine(Application.StartupPath, "View Saved Forms.mp4");     //load a video saved in a resources folder added to the program
    if (!System.IO.File.Exists(file))
        System.IO.File.WriteAllBytes(file, Properties.Resources.View_Saved_Forms);

    this.axWindowsMediaPlayer.URL = file;
    this.axWindowsMediaPlayer.Ctlcontrols.play();
}
```

5.5 Conclusion

This chapter explained the reasons why the writer adopted C# and WinForms to develop FarmField. This was a challenge for the writer who started the course with no knowledge of coding, learnt C++ during the course of a year and had three months to learn a new language and a new front end environment to complete this project. However, this challenge brought great knowledge to the writer, intellectual and coding challenges which required to be solved and a considerable amount of learning which will definitely be very useful in the future. Within this context the similarities of C# to C++ and the adaptability to of this language to WinForms were practicalities which led to the writer's choice. The challenges presented in the section also revealed the rationale and the work carried out by the writer to solve the problems encountered during this process.

6 Testing Strategy and Results

6.1 Introduction

This section introduces the testing carried out to establish the functionality of the program. This testing occurred at two levels. Firstly, the writer tested the entire program to confirm if all the functions operated correctly. Secondly, the program was sent to the participants who tested the prototype. The writer felt that it would be appropriate to contact the same participants who tested the prototype as this would provide continuity and ensure that the recommendation highlighted by these same participants were adequate. However, from the 7 participants contacted, only 5 advised to have availability to test the program by a given time slot (seven days). The other participants were not able to take part due to seasonal workload. Section 6.2 introducing the testing carried out by the writer while section 6.3 outlines the testing performed by the 5 participants.

6.2 Description of the testing results carried out by the writer

Test 1) The first test involved trialling the login process. If the user leaves the username and password fields blank and attempts to login, the outcome of this action leads to the display of a message stating that the username and password are incorrect and that the user should try to login again. The input of the incorrect values also leads to the display of the same message. By entering the correct details, then the system takes the user to the main page of the program. This all worked as expected.

Test 2) The writer tested the forgotten password and username recovery sequence by following the instructions provided by the system, which involved sending a code to the users' email. This worked as expected and the code was therefore introduced in the system leading to the successful change of the username and password.

Test 3) The Calendar Control was selected from the desktop and, as expected, the calendar displayed the deadlines and supporting information as requested by the participants who tested the prototype.

Test 4) The Weather Control was selected and, as expected, was displaying the current location weather and the two days forecast as requested by the participants who tested the prototype. The option of choosing a weather location was also tested with the city of Hong Kong which returned the weather for this location successfully. To further test this control, an invalid location was introduced which led to the return of a message box stating that it could not load the weather due to no internet connection or invalid location name. The program is able to recognised English and other languages. This was tested by using the word Lisbon and Lisboa (Portuguese name for the city). Both spellings returned the corrected weather for Lisbon/Lisboa as expected. It was also noticed that the lack of connection to the internet was making the weather control crash; this was later solved by displaying a message informing that the weather could not be displayed due to no internet connection. This is was achieved by using a Try {}Catch{} statement.

Test 5) The writer tested the Profit/Loss Calculation control. This was done by filling in the fields with information from the family farm. As expected, the program return the calculation as desired and was able to advise if the farm was profiting or not. The writer was also able to save the values introduced and open this document as a .txt file.

Test 6) The user control Application Forms was tested and, as expected, it took the writer to the page with several options of forms. The writer tested all forms and all controls took the writer to the desired download page.

Test 7) The user control Upload Forms was also tested. The email scan bottom was selected and the default email client was loaded as expected with the relevant local office email address loaded and the farm details entered in the Subject box. The upload RPID user control automatically leads the user to the login page of the Scottish government Website where the user can then proceed with his login details. Finally, the View Saved Forms user control was tested which led the writer to the documents folder in which the user would normally save their documents. This user controls all worked as expected.

Test 8) The Profile Settings user control shows the personalise details of the user which were passed onto the various user controls in the system successfully. Restore to default and save user controls were also tested and provided the expected results.

Test 9) The Help user control was also tested and worked as expected; however, the writer noticed that some of the interactive buttons did not perform in an expected manner as they were not exhibiting the colour change required when the mouse was over the button. This was fixed by adjusting some of the settings of the controls and started working as expected. The help videos were played by selecting the relevant subject and the videos also worked as expected.

Test 10) The writer tested if by clicking on the logo which is located in the top left hand side of the screen would take the user back to the desktop. This worked as expected.

Test 11) The About user control was also tested and, although it worked as expected, the message displayed was changed in order to read more clearly what the purpose of this program is.

Test 12) The logout user control was finally tested at the end of the testing stage. By clicking logout, a message box was displayed as expected asking for confirmation if the user wishes to continue to logout. By selecting no, the log out process was terminated. By selecting yes, the user was able to successfully logout from the program.

6.3 Methodology and Ethics – evaluation carried out by participants

Ethical issues are an important aspect of any piece of research due to the interest in protecting the participants and data. In this research, ethics was considering paramount as this would involve contacting people directly who could potentially have literacy issues or a lack of understanding over technology. Before carrying out testing, the writer discussed how to undertake this process in an ethical manner with supervisor Craig Ramsay. Both, writer and supervisor agreed that as long as no personal details were stored, all data was treated anonymously and the welfare of the participants was ensured, the writer could contact individuals to take part in the evaluation of the project and collect their opinions of the FarmField application. Following this, Craig Ramsay obtained a group consent from the Ethics Department - Dundee University. During the discussion with Craig Ramsay, the writer's plans to undertake the research felt within this scope of the Ethics Group's approval.

Based on the Ethics considerations discussed and similar to the prototype testing, the Writer contacted the seven participants who trailed the prototype. From these seven, two were unavailable to take part on the evaluation due to work commitments and their decision was respected. Five participants were happy to assist the writer during this stage. All farmers were approached by phone when asked to engage with the testing process. During this first phone call the following subjects were discussed:

1) The purpose of the test,

2) The writer discussed the testers' participation, consent form and future collection of data/feedback. A discussion about these subjects via telephone was preferable as the writer felt to be the best way to ensure that participants would engage and not forget about the trial which could easy happen during harvest time if they were left with a form explaining the application, a consent form to fill in and questionnaire to collect answers. Additionally, it is important to highlight that informal conversations are always more desirable within the farming community rather than documentation. This strategy was also better suited as people would not be concerned with literacy problems. Finally, some of these farmers were unable to meet in person due to living in different geographic locations, by being busy with their harvest and also due to Covid-19 restrictions.

3) The writer explained that the information collected would be treated anonymously, that the information would be stored in a computer (also anonymously) and deleted at the end of this project. All participants were happy to continue and provided verbal consent as long as their details were kept anonymous.

4) The writer advised that the application would be saved in a USB stick which would be sent via postal services. Due to the file size of the installation file (324mb) which exceeds the maximum email attachment size the writer decided to send the installation file to participants saved on a USB stick as in the previous prototype testing many participates struggled to access a shared download file.

5) The writer also informed and discussed the kind of tasks that they would be asked to do once they open the application. These tasks were:

- Simply navigate through the application
- Update their details in the profile settings
- Access the weather forecast for their location
- Access the calendar and observe the impending deadlines
- Perform a whole business calculation on either a selection of or all of their enterprises
- Download a Scottish Government subsidy form
- Attempt to send an email to the government using the Email form Function (But not to send)
- Watch a minimum of one help video
- Update either their password or Username via the "forgot username/password" options at login.

6) Authorisation for taking notes during phones call was also obtained.

7) Participants were also made aware that they had the right to stop the phone call at any time or not answer questions which they did not feel comfortable to answer.

8) Participants were informed that a week would be considered for the delivery of the USB stick and another week to carry out the testing. One participant was able to download the program from a shared Google drive folder.

9) The writer also establish a convenient time to make the second call a week later when he was aware that the USB stick had been delivered.

As agreed, a second call was made to check if farmers had received the USB stick and if they had managed to install the program. Although a video was sent to the participants with the instructions of the installation, this second contact became more a telephone guidance session on how to install the program for three participants; the other two were able to install the application without issues. It is also important to highlight that some participants started commenting on the application during this phone call. During this second contact, the writer explained that he would phone testers between the 27th and 29th of August to have a discussion about their feedback.

As planned, the writer contacted the participants a third time within the time scale pre-agreed to collect the feedback. The tasks mentioned during the first contact were discussed in an informal manner and feedback was collected throughout the conversation and noted in paper which was later shredded. The writer also noticed that participants were happy to discuss the application over the phone. No signs of stress or frustration were identified during the conversation; however, the writer felt that sometimes it was difficult to keep the farmers focused on evaluating the application and not to divert onto other farming subjects. The notes were then reflected which led the development of section 6.4 and Chapter 7 of this report

6.4 Evaluation testing - Description of the testing results carried out by participants

In order to test the program and evaluate if it met the expectations of the end users, the writer contacted the seven people who tested the prototype in order to provide some continuity to the testing and also to ensure that the changes highlighted by this participants during the prototype testing were successfully met. From the seven participants contacted, two were unable to take part on the evaluation testing due to seasonal work commitments.

The feedback from the testing was as follows:

1) Participants advised that the introductory video was too long and preferred a discussion by telephone to walk through the initial stages of installing the program on the desktop.
2) Testers advised that they found the option to see the password useful as some of them tried to enter the password without considering upper/lower case. The possibility to check this, allowed them to see where the mistake was.
3) Participants advised that the calendar was very useful, although the writing was too small. However, this was complemented with the support information bellow the calendar. Thus the small print was not considered a significant issue as they advised that they would probably use both forms of information to confirm their deadlines. Participants also suggested that they would like to be able to enter their own details and alerts in the calendar. A suggestion made by one participant was that countdown alerts/pop ups would be useful to indicate impending deadlines.
4) Participants with fields/farm in different locations found very useful to be able to check the weather forecast for the different locations; however, they would only check the weather through the program if they had the program already open. Thus, they would not use the program if they want to only check the weather forecast.
5) Participants found that by clicking on the desktop Icon was an easy way to return to the desktop.

6) Participants found particularly useful the already existing value set by the SAC Farm Managers Handbook as they do not always have this information to hand, but also gives them a tool to compare against predicted values offered to them. The ability to change these values to the price sold and not the actual values was also appreciated. One participant stated that this could influence the way they work in terms of putting the effort to have the crop up the standard required in order to sell at the prices stated by SAC Farm Managers Book states.
7) Participants advised that the program was very useful as it gave them the possibility to have a detailed account of their costs per enterprise, but also the full business profit and loss. In this way participants advised that by using FarmField they could make decisions about future crops and which ones could be more profitable, especially when they have already information from SAC Farm Managers Handbook. However, they mentioned that they would like to have the possibility to change the values once they have been saved rather than filling in the form again if there has been a change of circumstances.
8) The introduction of subsidies was also felt very useful, as some farmers may forget to take this income into account at times.
9) Having the opportunity to save the different yearly budgets was also thought to be a benefit as farmers could use this information to compare profit and losses in the future; however, they would like to be able to print the pages with this information from each enterprise budgeting page rather than needing to print from txt file.
10) Participants advised that they would like to save values as they go rather than introducing all the values and save at the end in case of program disruption.
11) Having the forms all compiled in one place with direct access to the Scottish Government Website and being able to send a form directly from the program was seen as saving time as sometimes can be difficult to find the right forms in government website.
12) The help section with different videos associated with each section was also mentioned to be preferable then written descriptions.
13) Participants also mentioned that it would be useful to be able to watch the help videos and follow the instructions at the same time by having the video displayed as a pop up window.

6.5 Conclusion

This section focused on the final stage of the software design cycle – Testing - which involved two different phases. The first stage was carried out by the writer who tested every form in the program. This revealed to be very important as it allowed the writer to understand if the program met the requirements established at the beginning of the project. As showed in this section, this testing was carried out without major problems and changes. The second stage of this process involved sending the program to five participants. This was a very important stage to validate the program and corroborate the testing carried out by the writer. Once again this process occurred without major problems. This evaluation testing was also important to delineate future recommendations as the next section will highlight. The chapter also highlights the methodology used to engage the participants with the evaluation of the program and the ethics employed during this stage.

7 – Final Conclusions and Considerations for Future Development

This project was developed in the context of the MSc Applied Computing Master's Degree – Dundee University. It aimed to develop a software program which supports users to manage their business budget and therefore make informed decisions, and also assist with their access to applications to governmental schemes which are an important source of income. The program was developed to be a user friendly application, to take into account literacy difficulties in the farming community and be easily adjustable to future needs and developments. To create this program the writer followed the software engineering cycle and UX principles which include the creation of a prototype, and used C# and Windows Forms to create the program itself. The choices made in relation to the creation of the application were mainly based upon the knowledge gained from the MSc; however, C# and Windows Forms were chosen due to the familiarity of the writer with C++ and time constrains (the writer felt that C# would be easy to learn within the time allocated to the project considering his knowledge on C++).

The onset of the project led the writer to undertake research in field (literature research/review and questionnaires) which confirmed that there is a place for this kind of technology and applications in the agricultural and rural industry as a consequence of farmers struggling to understand business budgeting. This project also took into account the high levels of literacy issues within the industry and which can be an obstacle to the usage of technology. This was achieved by building a simple program and by employing simple wording. The writer felt that this was achieved as participants felt no difficulties in understanding what was being asked of them during the evaluation testing.

Following the different software design stages was very constructive as it deepened the knowledge gained during the course, but also ensured that by adopting a Waterfall Model approach, the writer was able to minimise risks and ensure efficiency as each task would need to be fully completed before moving into the next stage. The creation of the requirements was felt to be one of the most important stages within the software design cycle as the writer felt this to be the base of the rest of the development process. Following this the different stages such as the development of use cases, threat model, class and sequence diagrams flowed without major setbacks.

The creation of the logo and the prototype was important as the writer was required to use his UX and HCI knowledge together with some creativity. Although the creation of the prototype was not a requirement, it was adopted in this project so that the writer could establish the correct path before investing a considerable amount of time in the project. The prototype was also tested by participants who advised the writer to change some aspects of the program before starting hard coding it.

Learning a new language and front end application was initially felt as an impossible mission due to the lack of knowledge and experience and the time allocated to the project.

However, after some research into C# and Windows Forms and advice provided by the supervisor and people within the programming industry, the writer felt this to be the best plan going forward. At the end of the project, the writer concluded that learning C# was more achievable then initially thought with problems experienced being a consequence of a lack of knowledge which was not covered within the MSc such as the development of GUI interfaces. However, research in the field allowed the writer to overcome these obstacles.

The final stage of the software engineering cycle was the testing and evaluation testing. Both stages occurred without major issues and the forms worked as expected. Nevertheless, the evaluation testing carried out by the participants offered the writer some future recommendations. These recommendations are as follows:

- Although the users can use upper and lower case, numbers and symbols, the application would benefit from an advice system informing how to create a strong password and the ability to analyse the users password with regard to its integrity.
- In relation to the calendar some redesign of the text incorporated within the calendar would be beneficial due to the small font displayed or provide a day or week view where the print would be displayed with a bigger font.
- Another future suggestion provided by the participants involving the calendar was the introduction of countdown deadline alerts/pop ups.
- Another recommendation involves the possibility to change the values on the budget forms once they have been saved rather than filling in the form again if there has been a change of circumstances and print these pages as they are displayed rather than needing to open a .txt file to print this information.
- Participants also mentioned that it would be useful to be able to watch the help videos and follow the instructions at the same time by having the video displayed as a pop up window.
- The Writer also felt that having an admin login which would allow access to the program in the event of the user to be completed lockout would be advantageous as currently a complete lockout would require a full reinstallation of the program.
- The writer also felt that a future mobile version would beneficial considering the current trend of technology.
- Finally, the possibility to disseminate the application with tailored features to specific businesses could also be a future benefit.

Another point relevant to this chapter is the writer personal's reflection in relation to its own work. The writer felt that completing the project with knowledge in UML and a different language from the one he learnt in the course was a benefit to his own development. The writer would also have liked to have involved more participants; however, Covid-19 restrictions, harvest time and the time allocated to the project hampered the ability to do this. Finally, if the writer had the opportunity to do this application again he would have considered the possibility to save the application information/calculations in the cloud where a mobile app version would be able to pull that information from rather than saving it directly to the documents folder of the user's computer.

Although there is space for improvement and further developments which cannot be dealt with in the current time constraints, the writer feels that the program has achieved its main goal: to be simple, understandable by people with literacy issues and usable to assist and support business budgeting. This was corroborated by the participants during their questionnaires and testing. Thus, Farmfield can be a future support to integrate technology in the farming and rural sector.

References

BBC Scotland News (2015), Dyslexia in farming highlighted by NFU in Scotland. Available at: https://www.bbc.co.uk/news/av/uk-scotland-31815442/dyslexia-in-farming-highlighted-by-nfu-in-scotland [Accessed:24/05/20]

Bayer Crop Science (2019), 19 intriguing UK farming facts. Available at: https://cropscience.bayer.co.uk/blog/articles/2019/10/farming-facts/ [Accessed 07/06/20]

British Dyslexia Association, What is Dyslexia?, Available at: https://www.bdadyslexia.org.uk/dyslexia/about-dyslexia/what-is-dyslexia [Accessed:24/05/20]

Hulme, C. and Snowling, M. (2017), Reading disorders and dyslexia, Nacional Library of Medicine, National Institute of Health, Published online 2017 Feb, Availabe at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5293161/ [Access 25/05/2020]

Lucid Chart, Types of UML Diagrams Available at https://www.lucidchart.com/blog/types-of-UML-diagrams [Accessed: 24/05/20]

Milligan, M. et. all (2019). Addressing Accessibility Issues in Institutional Review Board Policies. DSQ Vol. 39, No 1 (2019). Retrieved May 5th, 2020 from https://dsq-sds.org/article/view/6062/5188

Muddyboots Softwre, http://en.muddyboots.com/

NFU, Farming Facts. Available at: https://www.nfus.org.uk/farming-facts.aspx#:~:text=Some%2080%25%20of%20Scotland%27s%20land,billion%20food%20and%20drink%20exports. [Accessed: 07/06/2020]

Prince's countryside Fund (2016), The cash Flow crisis in farming and its implications for the wider rural economy. Available at: https://www.princescountrysidefund.org.uk/downloads/research/cash-flow-crisis-in-farming-report-april-2016.pdf [Accessed: 24/05/20]

Ramsay, C. (2020), Software Engineering Model, MSc Applied Computing Course, Department of Technology and Information, Dundee University.

Scottish Government (2016), AGRICULTURAL LAND USE IN SCOTLAND. Available at: https://www2.gov.scot/Topics/Statistics/Browse/Agriculture-Fisheries/agritopics/LandUseAll#:~:text=The%20total%20area%20of%20agricultural,area%20(7.8%20million%20hectares)%20 [Accessed: 07/06/20]

Shearwell, Management Formworks software https://www.shearwell.co.uk/management-software

Weiss, Kai (2019), The CAP doesn't fit – why the EU's farm subsidies are ripe for reform. Available at: https://capx.co/the-cap-doesnt-fit-why-the-eus-farm-subsidies-are-ripe-for-reform/ [accessed: 24/05/20]

Additional References

Web links used during the architecture of program

User controls https://www.youtube.com/watch?v=TeTuenCtvvk

Opening web pages https://www.youtube.com/watch?v=9C7J2Be07jE

Open file dialog https://www.youtube.com/watch?v=1oBHpE7H12A

Message box confirming exit of program https://www.youtube.com/watch?v=TkO8JQGoyrA

Button colour change https://www.youtube.com/watch?v=9nmAC-X86bg

Drop down menus https://www.youtube.com/watch?v=nOIR9_dHm1s

Display web pages https://www.youtube.com/watch?v=xEmDHNHivyw

Datetime function https://docs.microsoft.com/en-us/dotnet/api/system.datetime.today?view=netcore-3.1

Pass variables between forms https://www.youtube.com/watch?v=dWT43vdsUyk

Web link used to created recover username and password by using an Email

Setting up a email client https://stackoverflow.com/questions/20328598/open-default-mail-client-along-with-a-attachment

Web Link used to create the login

Login screen https://www.youtube.com/watch?v=tcmmCcMs8yU

Hide password https://www.youtube.com/watch?v=bXIP4uoy_FA

Send Passcode to registered email https://www.codegrepper.com/code-examples/csharp/c%23+send+email

Web Link used to create the Calendar

Calendar control https://www.codeproject.com/Articles/10840/Another-Month-Calendar

Web Link used to create the Profile settings

Save user settings	https://www.youtube.com/watch?v=P432z8q9iVE&app=desktop

FarmField Desktop Application

Appendices

John D Lamont

Science and Engineering

Computing

MSc Applied Computing

2019-20

Index of Appendices

Appendix 1 End User Questionnaire Template

As part of my MSc in Applied Computing I have decided to look into developing a simple Agri business budgeting program which can be run from a windows desktop computer.

Obtaining feedback from Farm managers/agents is vital to the review process and requirements to build this program. We would appreciate you taking the time to complete the following survey.

This questionnaire is divided into three sections and it should take around five minutes of your time to complete. Your responses are voluntary and will be treated anonymously and confidentially.

All responses will be compiled together and analyzed as a group. If you have any questions or concerns, please contact John Lamont, on 090016802@dundee.ac.uk.

Thank you,

John Lamont

Section 1 – General Background

1. Are you actively involved in farming?

   □ No □ Yes □ I Prefere Not To Say

2. Do you employ additional staff as part of your business?

   □ No □ Yes □ I Prefere Not To Say

3. If yes to Question 2 are the staff seasonal workers, part time or full time?
   Tick all that apply:

   □ Full time □ Part time □ Seasonal □ I Prefere Not To Say

4. Do you manage your business budgeting?

   □ No □ Yes □ I Prefere Not To Say

5. If yes to Question 4 do you use existing software or paper files

   □ Accounts Software □ Paper Files □ I Prefere Not To Say

Section 2 – Technology Preferences

6. Do you own a computer?

   □ No □ Yes □ I Prefere Not To Say

7. Out of five how would you rate your computer skills? (1 basic - 5 advanced)

   □ 1 □ 2 □ 3 □ 4 □ 5 □ I Prefere Not To Say

8. What operating System does your computer run?

   □ Windows □ IOS □ I don't Know □ I Prefere Not To Say

9. Would a mobile phone version be of interest?

   □ No □ Yes □ I Prefere Not To Say

10. Do you have an internet connection?

    □ No □ Yes □ I Prefere Not To Say

Section 3 – Reporting/Budgeting

11. Is there any particular task you struggle with when managing your business paperwork? Please describe here:

12. Do you feel it would be beneficial to your business to have a simple budgeting program associated with financial government scheme support?

    □ No □ Yes □ I Prefere Not To Say

13. How do you consider the accessibility of information related to form deadlines on the Scottish Government (RPID) website? (1 Easy - 5 difficult)

    □ 1 □ 2 □ 3 □ 4 □ 5 □ I Prefere Not To Say

14. Do you have any other problems when completing your business forms and record keeping? Please describe here:

15. Please tick which of the following Government schemes you claim/ if any?

□ BPS

□ Agri-Environment Climate Scheme

□ Scottish Suckler Beef Support Scheme

□ Scottish Upland Sheep Support Scheme

□ Forestry Grant Scheme

□ Less Favoured Area Support Scheme

□ Crofting Agricultural Grant Scheme

□ Croft House Grant Scheme

□ Small Farms Grant Scheme

□ New Entrants grants

□ I Prefere Not To Say

Other Please specify:

16. Which of the following livestock do you have on your farm?

□ Sheep

□ Beef Cattle

□ Dairy Cattle

□ Goats

□ Pigs

□ Poultry

□ No livestock

□ I Prefere Not To Say

Other Please specify:

17. Which of the following Arable crops do you grow if any?

 □ Wheat

 □ Barley

 □ Oats

 □ OSR

 □ Linseed

 □ Vine peas

 □ Combinable peas

 □ Beans

 □ No Arable

 □ I Prefere Not To Say

 Other Please specify:

18. Which of the following Root crops do you grow if any?

 □ Carrots

 □ Potatoes

 □ Turnips

 □ No Root Crops

 □ I Prefere Not To Say

 Other Please specify:

19. Which of the following Vegetable crops do you grow if any?

□ Broccoli

□ Cauliflower

□ Brussels sprouts

□ Cabbage

□ No Vegetable crops

□ I Prefere Not To Say

Other Please specify:

20. Which of the following Fodder crops do you grow if any?

□ Beet

□ Silage

□ Hay

□ Grazing Grassland

□ Kale

□ Turnips

□ No Fodder crops

□ I Prefere Not To Say

Other Please specify:

21. Which of the following soft fruit crops do you grow if any?

□ Strawberries

□ Raspberries

□ Gooseberries

□ Blueberries

□ Black currents

□ No Soft fruit

□ I Prefere Not To Say

Other Please specify:

22. Which of the following building types do you have as part of your business?

□ Domestic Properties (long term let)

□ Domestic Properties (holiday let)

□ Industrial Properties

□ I Prefere Not To Say

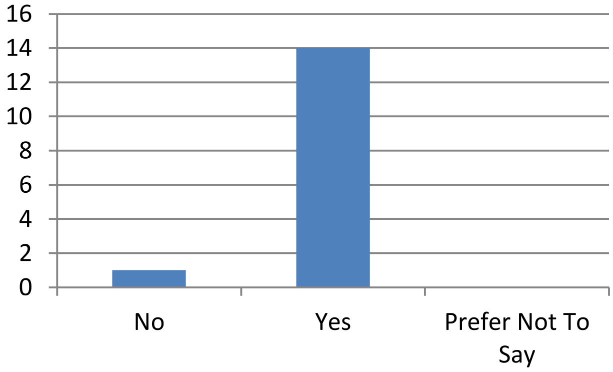
Other Please specify:

23. Is there anything else you would like to add which could be of benefit to the development of a business management program?
i.e. How it could specifically address your own needs.
Please describe here:

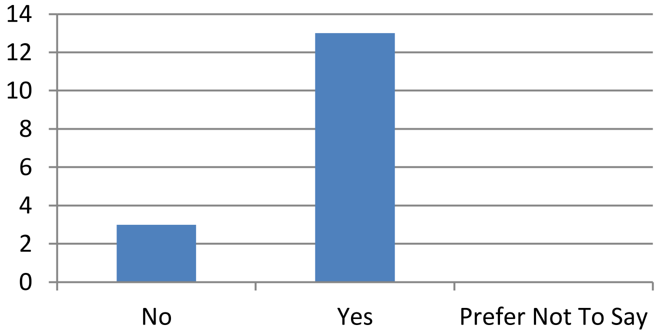
Appendix 2 Graphic Representation of the Questionnaire Results

Questionnaire results

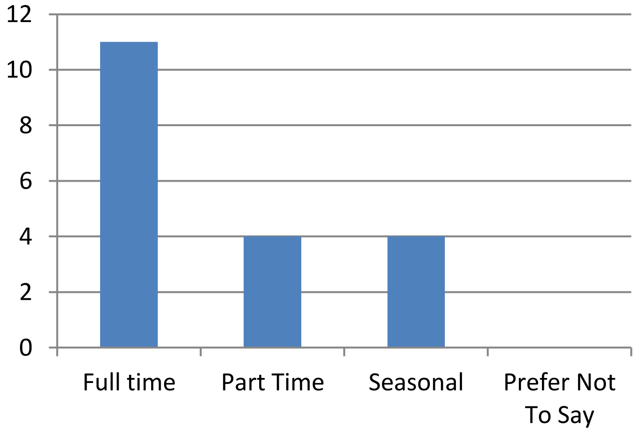
1. Are you actively involved in farming?



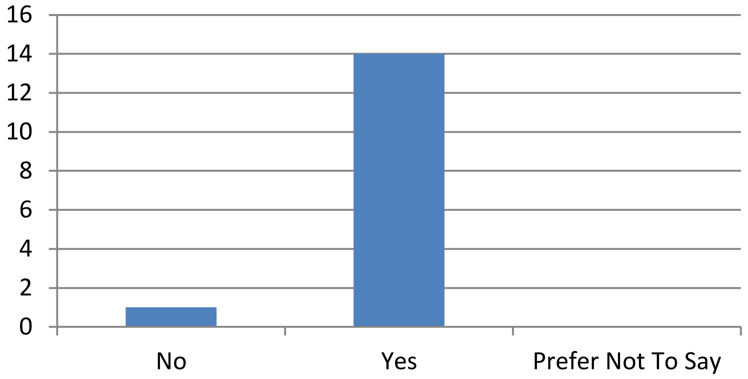
2. Do you employ additional staff as part of your business?



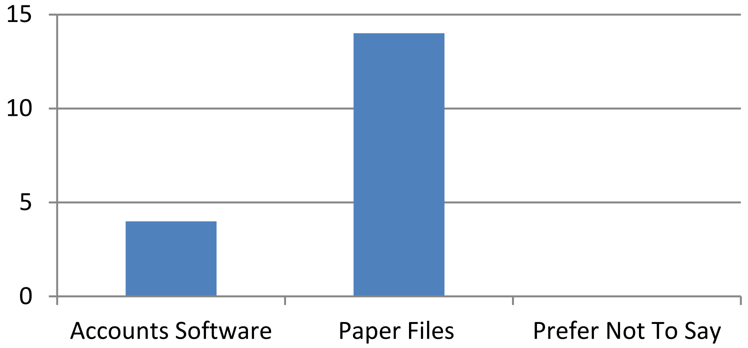
3. If yes to Question 2 are the staff seasonal workers, part time or full time?



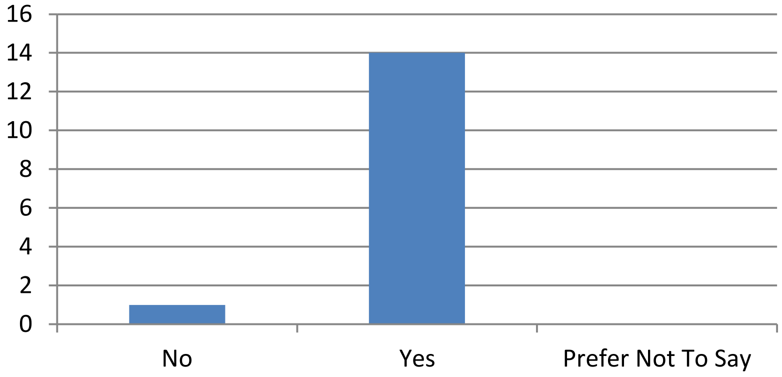
4. Do you manage your business budgeting?



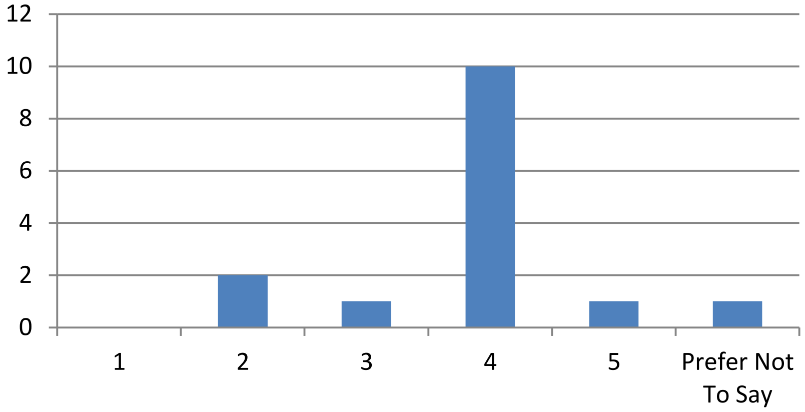
5. If yes to Question 4 do you use existing software or paper files



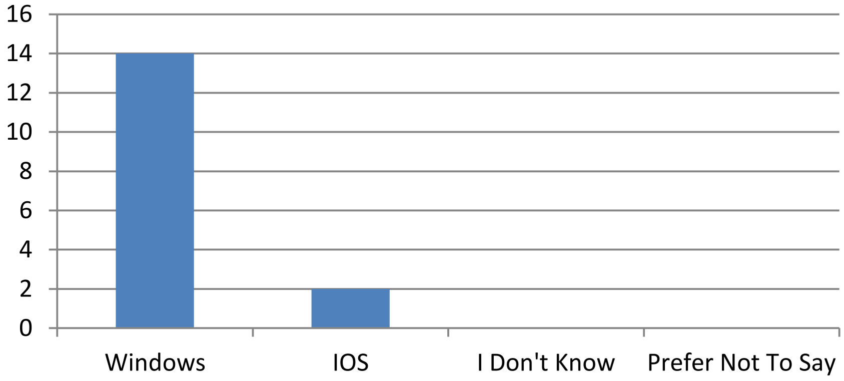
6. Do you own a computer?



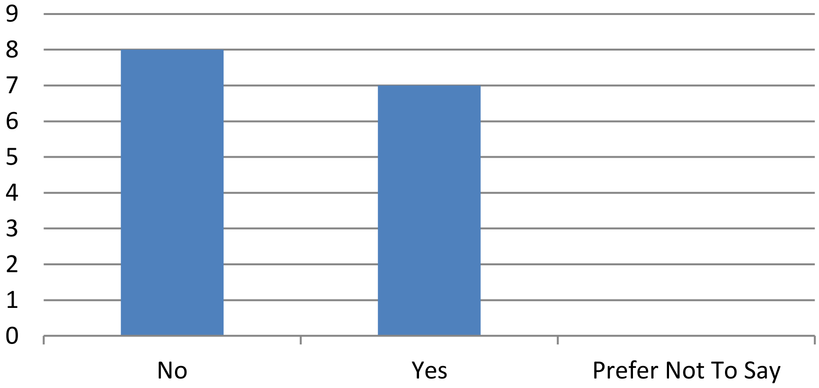
7. Out of five how would you rate your computer skills? (1 basic - 5 advanced)



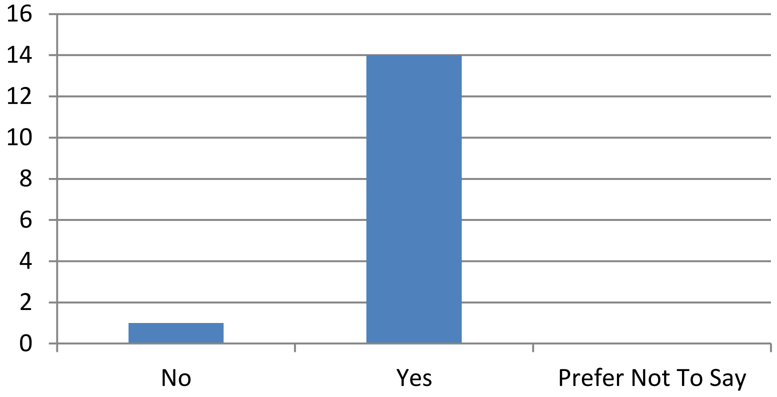
8. What operating System does your computer run?



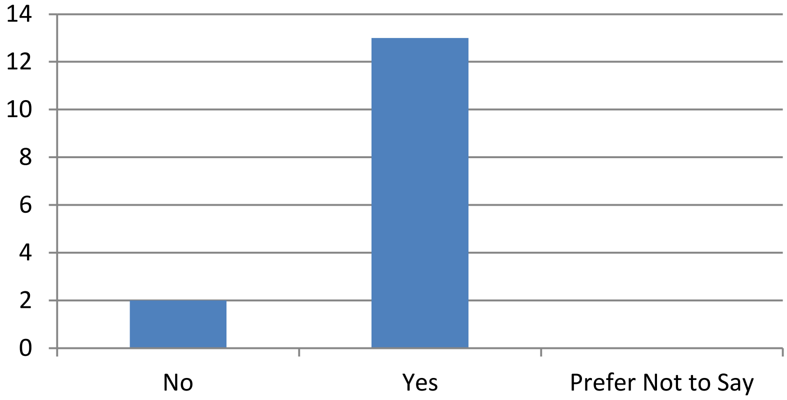
9. Would a mobile phone version be of interest?



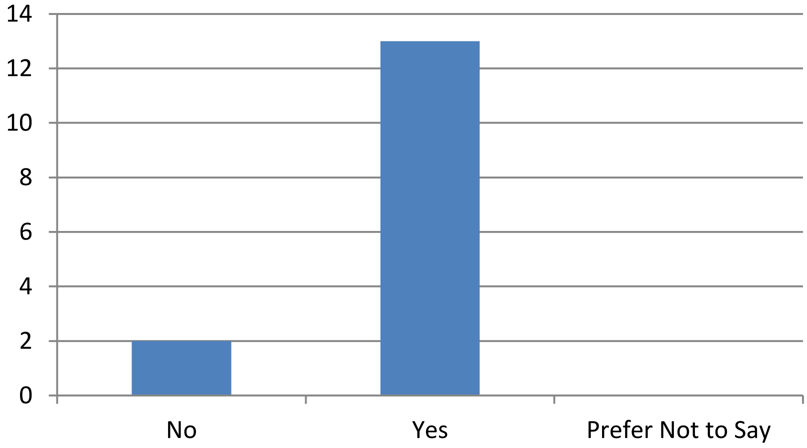
10. Do you have an internet connection?



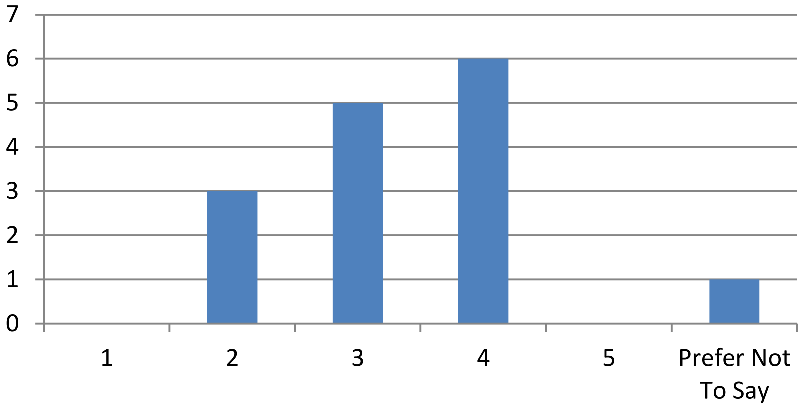
11. Is there any particular task you struggle with when managing your business paperwork?



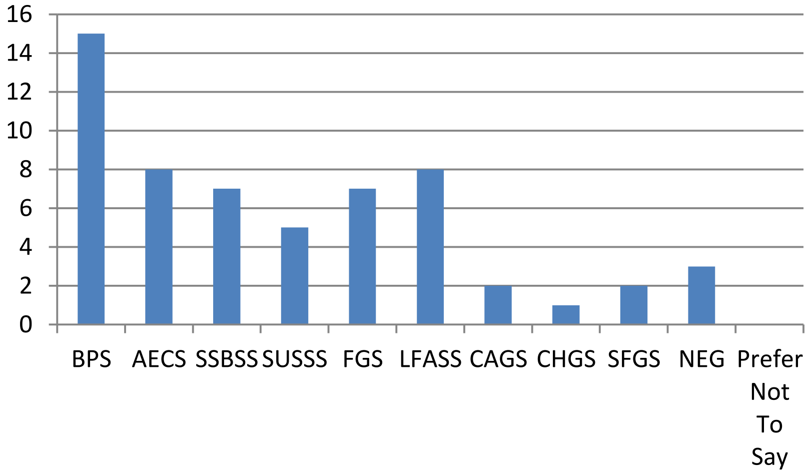
12. Do you feel it would be beneficial to your business to have a simple budgeting program associated with financial government scheme support?



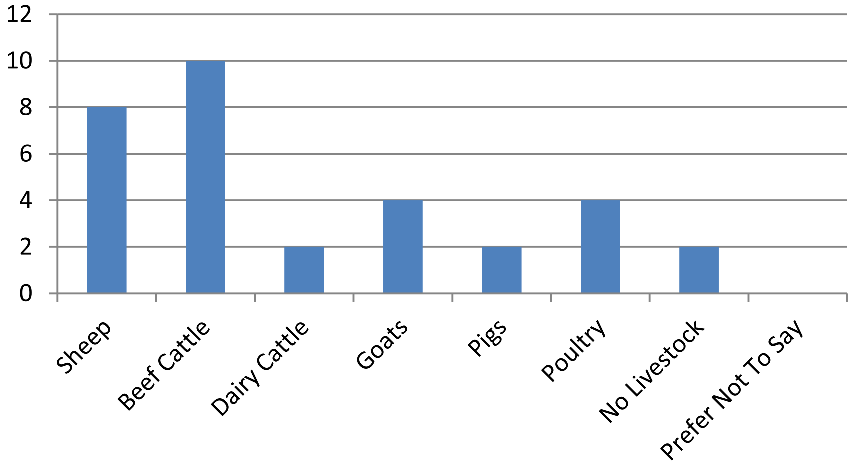
13. How do you consider the accessibility of information related to form deadlines on the Scottish Government (RPID) website? (1 Easy - 5 difficult)



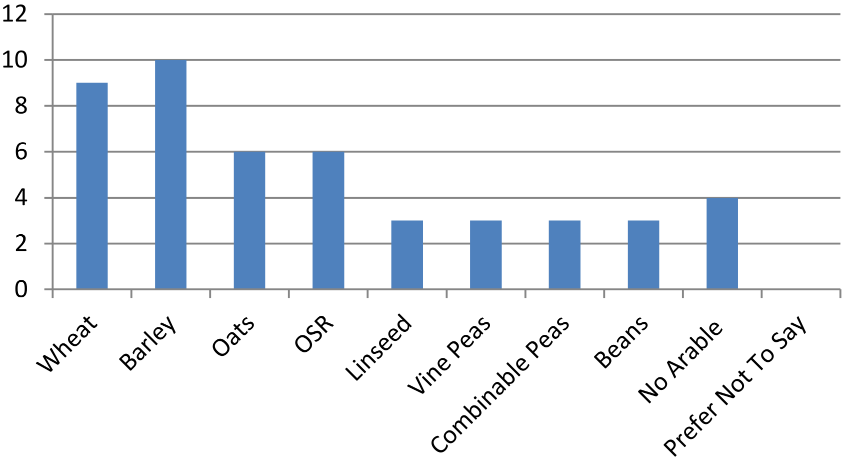
14. Please tick which of the following Government schemes you claim/ if any?



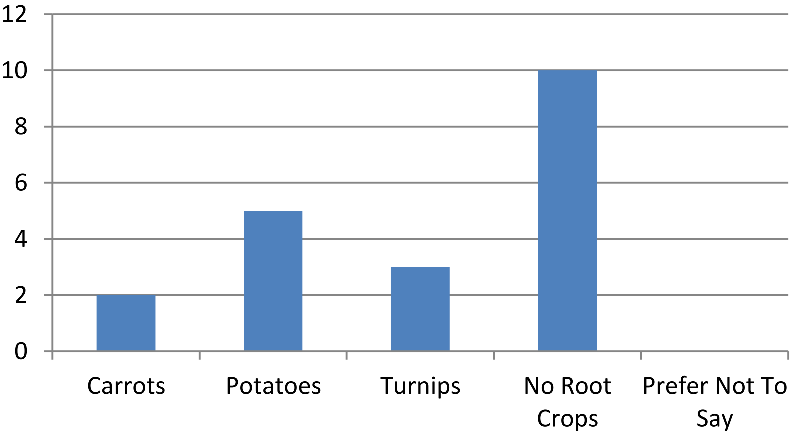
15. Which of the following livestock do you have on your farm?



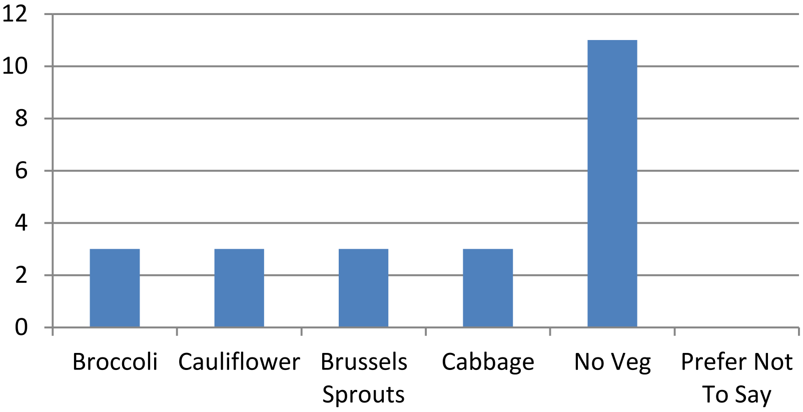
16. Which of the following Arable crops do you grow if any?



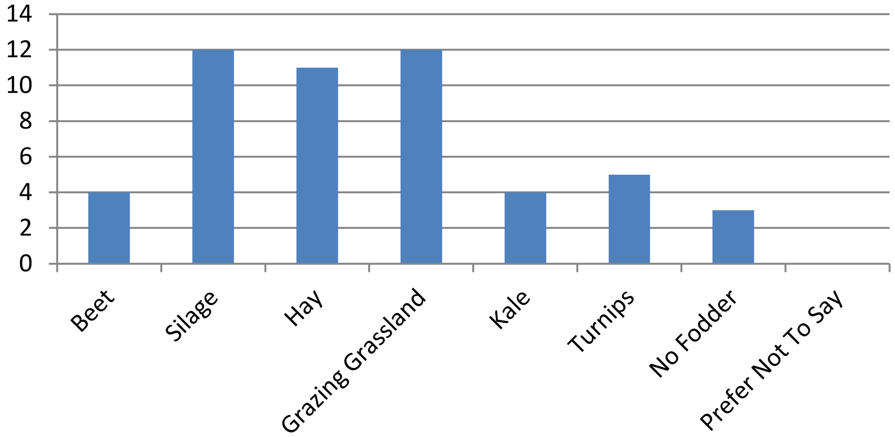
17. Which of the following Root crops do you grow if any?



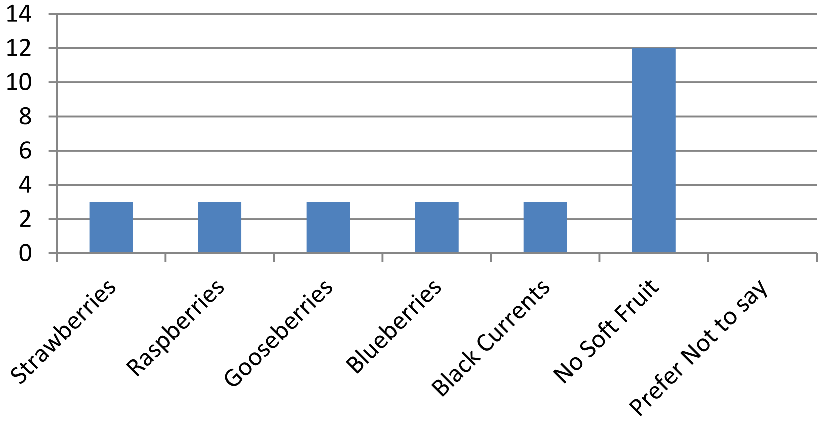
18. Which of the following Vegetable crops do you grow if any?



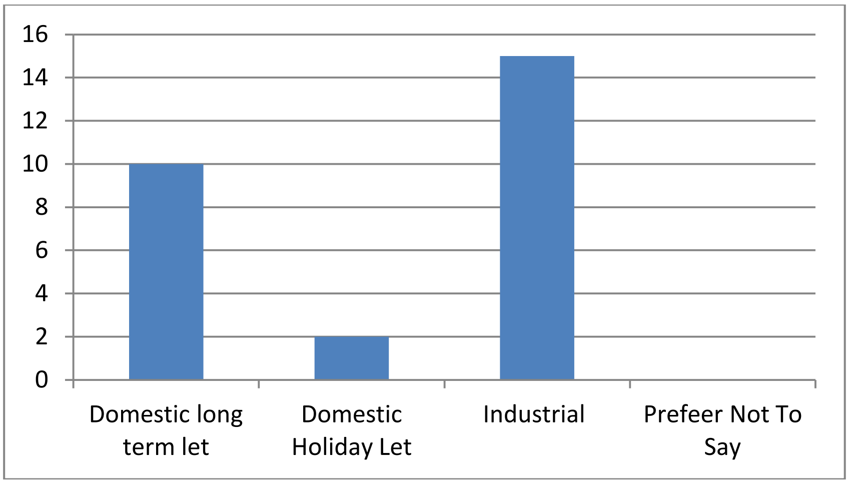
19. Which of the following Fodder crops do you grow if any?



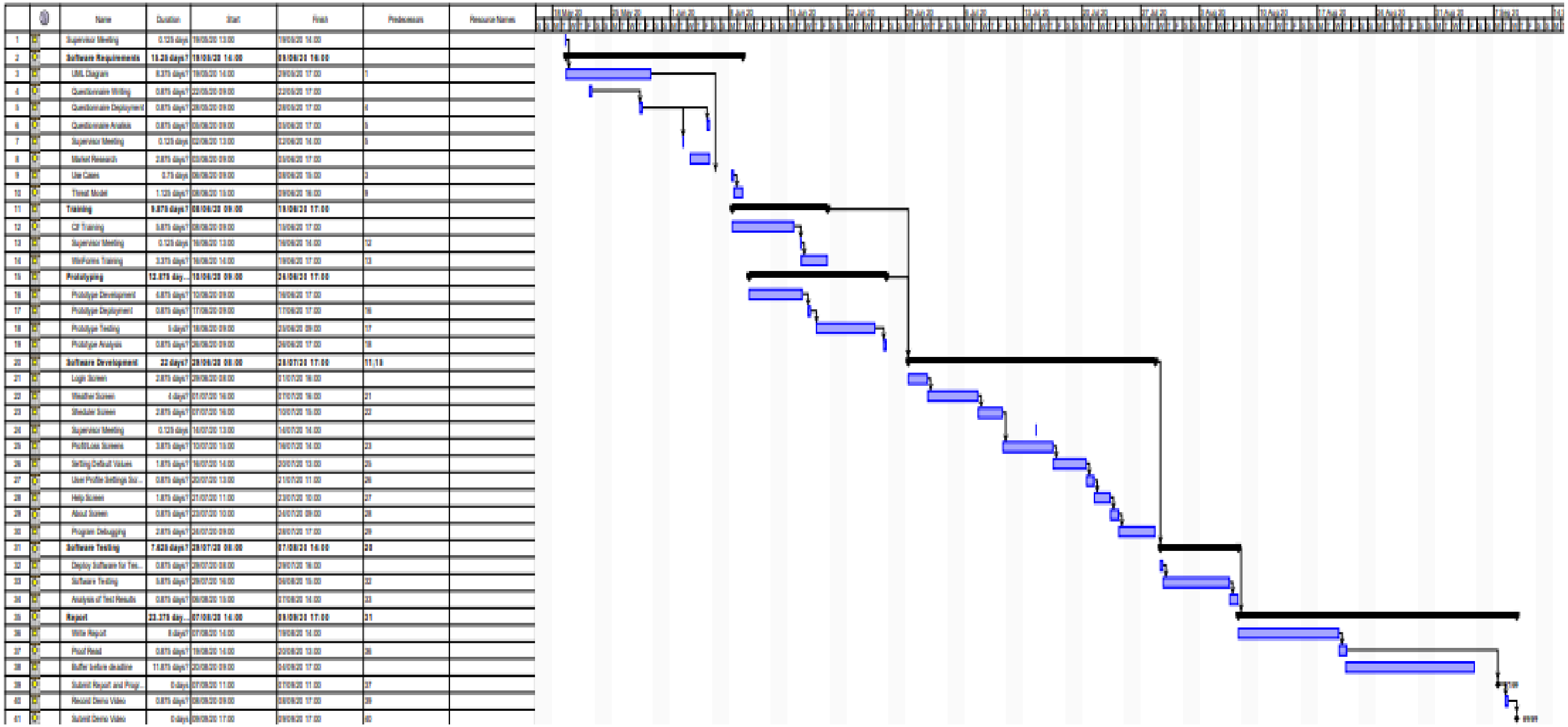
20. Which of the following soft fruit crops do you grow if any?



21. Which of the following building types do you have as part of your business?



Appendix 3 Gant Chart



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		Supervisor Meeting	0.125 days	19/05/20 13:00	19/05/20 14:00		
2		**Software Requirements**	**15.25 days?**	**19/05/20 14:00**	**09/06/20 16:00**		
3		UML Diagram	8.375 days?	19/05/20 14:00	29/05/20 17:00	1	
4		Questionnaire Writing	0.875 days?	22/05/20 09:00	22/05/20 17:00		
5		Questionnaire Deployment	0.875 days?	28/05/20 09:00	28/05/20 17:00	4	
6		Questionnaire Analsis	0.875 days?	03/06/20 09:00	03/06/20 17:00	5	
7		Supervisor Meeting	0.125 days	02/06/20 13:00	02/06/20 14:00	5	
8		Market Research	2.875 days?	03/06/20 09:00	05/06/20 17:00		
9		Use Cases	0.75 days	08/06/20 09:00	08/06/20 15:00	3	
10		Threat Model	1.125 days?	08/06/20 15:00	09/06/20 16:00	9	
11		**Training**	**9.875 days?**	**08/06/20 09:00**	**19/06/20 17:00**		
12		C# Training	5.875 days?	08/06/20 09:00	15/06/20 17:00		
13		Supervisor Meeting	0.125 days	16/06/20 13:00	16/06/20 14:00	12	
14		WinForms Training	3.375 days?	16/06/20 14:00	19/06/20 17:00	13	
15		**Prototyping**	**12.875 day...**	**10/06/20 09:00**	**26/06/20 17:00**		
16		Prototype Development	4.875 days?	10/06/20 09:00	16/06/20 17:00		
17		Prototype Deployment	0.875 days?	17/06/20 09:00	17/06/20 17:00	16	
18		Prototype Testing	5 days?	18/06/20 09:00	25/06/20 09:00	17	
19		Prototype Analysis	0.875 days?	26/06/20 09:00	26/06/20 17:00	18	
20		**Software Development**	**22 days?**	**29/06/20 08:00**	**28/07/20 17:00**	**11,15**	
21		Login Screen	2.875 days?	29/06/20 08:00	01/07/20 16:00		
22		Weather Screen	4 days?	01/07/20 16:00	07/07/20 16:00	21	
23		Sheduler Screen	2.875 days?	07/07/20 16:00	10/07/20 15:00	22	
24		Supervisor Meeting	0.125 days	14/07/20 13:00	14/07/20 14:00		
25		Profit/Loss Screens	3.875 days?	10/07/20 15:00	16/07/20 14:00	23	
26		Setting Default Values	1.875 days?	16/07/20 14:00	20/07/20 13:00	25	
27		User Profile Settings Scr...	0.875 days?	20/07/20 13:00	21/07/20 11:00	26	
28		Help Screen	1.875 days?	21/07/20 11:00	23/07/20 10:00	27	
29		About Screen	0.875 days?	23/07/20 10:00	24/07/20 09:00	28	
30		Program Debugging	2.875 days?	24/07/20 09:00	28/07/20 17:00	29	
31		**Software Testing**	**7.625 days?**	**29/07/20 08:00**	**07/08/20 14:00**	**20**	
32		Deploy Software for Tes...	0.875 days?	29/07/20 08:00	29/07/20 16:00		
33		Software Testing	5.875 days?	29/07/20 16:00	06/08/20 15:00	32	
34		Analysis of Test Results	0.875 days?	06/08/20 15:00	07/08/20 14:00	33	
35		**Report**	**23.375 day...**	**07/08/20 14:00**	**09/09/20 17:00**	**31**	
36		Write Report	8 days?	07/08/20 14:00	19/08/20 14:00		
37		Proof Read	0.875 days?	19/08/20 14:00	20/08/20 13:00	36	
38		Buffer before deadline	11.875 days?	20/08/20 09:00	04/09/20 17:00		
39		Submit Report and Progr...	0 days	07/09/20 11:00	07/09/20 11:00	37	
40		Record Demo Video	0.875 days?	08/09/20 09:00	08/09/20 17:00	39	
41		Submit Demo Video	0 days	09/09/20 17:00	09/09/20 17:00	40	

Appendix 4 Project Requirements

R1 General Software Requirements

Non-functional requirements

R1.1 – Software completion time scale
Description - Software system should be completed by the beginning of September.
Rationale – this deadline has been set in order to meet the assignment deadline.

R1.2 – Software programming language
Description - Software shall be written in C# using visual studio IDE.
Rationale – C# is a robust language capable of integrating with existing Windows systems and allows the development of a user friendly interface.

R1.3 – Future proof
Description – Software should be updatable to allow future requirements and demands in the field.
Rationale – forms and file types can change overtime, and the software should be allowed this through updates.

R1.4 – File standard formats
Description – Software shall conform to standardised file format set by Scottish Government and basic desktop applications.
Rationale – In order for this system to integrate with existing system, file types must be shared.

Functional Requirements

R1.5 Purpose of the software
Description - The software supports users to manage their agricultural business budget and their applications to governmental schemes in order to ascertain enterprise planning.
Rationale - The program aims to be a user friendly application which takes into account existing literacy and IT skills difficulties and be easily adjustable to future needs and developments.

R1.6 Software updates Procedure
Description - If a new version of the software or an update is available an alert should be sent to the user**.** Updates shall be managed by software developer/ manager.
Rationale – having an updated system is important to ensure the functionality of the software and the business.

R2 Log in Procedure

Non functional requirements

R2.1 – User licence quantity
Description - The system should allow only one person to have access to the system at a given time.
Rationale – The system aims to be used by the business owner/manager who will be able to share the information/access with people who he/she chooses (normally the wife or son/daughter). This kind of task is normally managed by one or two people in the field. The access to one single authorised user also aims to guarantee security and concurrency.

R2.2 Password Structure
Description – The password should be definable by user.
Rationale – Ensure security and allow the password to be memorable.

Functional Requirements

R2.3 First time access to the system
Description – The software shall ask for the service user to define a user name and a password to have future access.
Rationale – The usage of a user name and password provides security to the system.

R2.4 Definition of the business Profile
Description – Following the creation of the password, the system shall ask the user to introduce information associated with the business.
Rationale – This provides information which can be used for future calculations.

R2.5 Access to the system
Description - The system should allow access to one person by requesting the introduction of a user name and a password.
Rationale: This requirement is important as it involves the security of the system and to ensure that non-authorise people have access to the information of the business.

R2.6 - **Authorisation to Access the system**
Description - Once the service user has introduced the password, the system shall authenticate the details before allowing access.
Rationale – This requirement is important as it involves the security of the system and to ensure that non-authorise people have access to the information of the business.

R2.7- Input of wrong password and user name
Description – the program shall display a message informing that the incorrect details have been entered.
Rationale - keep the user informed as to what it is happening at various stages of the log in process.

R2.8 – Reset password
Description – The software shall provide the user with the possibility to change the password.
Rationale – This ability to change password is important if the user forgets the password or an unauthorised person has got access to the user name and password details.

R2.9 Desktop page
Description – Following the authentication of password and user name, the system should allow access to the main desktop where the user can choose from clear and defined options.
Rationale – Definition of a simple pathway to start using the system menu by taking into account potentially literacy and IT skills problems. The system aims to be simple to use.

R3 Main menu

Non Functional Requirements

R3.1 – Access to main menu
Description – The access to the main menu should be simple and without complications.
Rationale – To allow people with literacy and IT skill problems to use the software.

R3.2 – Display of main Menu
Description – The main menu should be simple, clear and easy to access.
Rationale - To allow people with literacy and IT skill problems to use the software.

R4 Main Menu - Software Functions

Non-functional Requirement

R 4.1 Weather application
Description – Weather application may be tied to the farm profile which allows a swift display of the weather at the location of the farm.
Rationale – This may be a desirable feature as day-to-day tasks often depend on the weather.

R 4.2 Source of weather forecast
Description – The source of the weather forecast should be from a recognised professional body.
Rationale – Provide an accurate weather forecast.

Functional Requirements

R4.3 Weather notification and updates
Description – The user should be able to access the weather information via the desktop
Rationale - Weather information can be useful on the day-to-day running of a agricultural busyness as a consequence of the impact of the weather on agricultural tasks.

R4.4 Calendar application
Description – The Calendar should be easy to use and read, and be displayed as a month view format.
Rationale – Month view format allows both clear planning and visibility of deadlines.

R4.5 information introduced on the calendar
Description – The calendar should be populated and updated to represent dated and changes on these dates established by the Scottish Government.
Rationale – This allows the user to have updated deadlines without them requiring to manually up-date the system themselves. This is a very important requirement as farmers

can be dealing with several tasks at the same time in one day and forget to update their systems. Failing deadlines can have a financial impact on the business.

R4.6 Management of application forms
Description – Users should be able to download any application form directly from the Scottish Government website. This is an extra and useful functionality of this program which can assist the whole business.
Rationale – These forms can be difficult to find within the Governmental Website, thus this software provides a direct link and a simple pathway to the forms.

R4.7 Management of completed application forms
Descriptions – The user should be able to store previous and completed applications and/or forwarded to the Scottish Government via this software.
Rationale – Keeping records together can facilitate inspections and farm management tasks.

R4.8 Enterprise profit/loss calculations
Description – The software main function should allow the user to simply and effectively calculate profit loss of each of the enterprisers within the business.
Rationale – Many farmers/business owners struggle to calculate the profit/loss of each of the enterprises individually. Farmers/business owners normally tend to look at the farm as a whole, rather than maximise each enterprise.

R4.9 Individual Enterprises
Description – The software should allow the user to choose among different enterprises. This navigation should be simple and straightforward.
Rationale - Many farmers/business owners struggle to calculate the profit/loss of each of the enterprises individually. Farmers/business owners normally tend to look at the farm as a whole, rather than maximise each enterprise.

R4.10 Calculations of enterprises
Description – The user should be able to introduce the figures throughout the season in order to obtain a running total.
Rationale – The ability to introduce manually takes into account the different prices throughout the seasons, years, regions and buyers.

R4.11 Introduction of Calculations
Description – the introduction of calculations should be simple and based on profit and costs. Thus, users should be able to introduce production cost of a specific item and the selling price - this provides the margin profit of a specific enterprise per season/year. This should be developed taking into account literacy issues and low IT skills.
Rationale - Many farmers/business owners struggle to calculate the profit/loss of each of the enterprises individually. Farmers/business owners normally tend to look at the farm as a whole, rather than maximise each enterprise.

R4.12 Calculations not requiring manual input
Description – The information for these calculations should be taken from the current Farm Manager Handbook published every year by SRUC. These calculations will be updated by the software manager who manages software updates.
Rationale – some calculations may be difficult to input manually by users due to the impossibility of farmers and business owners to have a detailed cost allocated to specific tasks. For example, farmers do not know accurately how much diesel they spent feeding

their cows per day if that tractor is used in other multiple tasks, but they will know how much fuel they buy per year. This software aims to support them with these calculations by automatically calculating difficult associations.

R4.13 Whole Business Calculation
Description – The profit/loss calculation should also be able to calculate the whole business profit and loss.
Rationale – This allows users to have a clear understanding of their business as whole.

R4.14 Calculation reports
Description – The software should allow the user to print reports per enterprise and for the whole business.
Rationale – This is an important document which could be shared with accountants.

R5 Logout

Description – Logout will automatically occur when the user closes the window.
Rationale – This aims to facilitates the logout, tackle literacy issues and ensure security in a easy and simple way.

Appendix 4 Project Use Cases

Login

Actors

The following actors are involved in this use case: User and System.

Brief Description

This use case describes the process of logging in to the system

Basic Flow of Events

The use case begins when the user selects the program from the computer icon.

User	User selects the FarmField icon on the computer desktop.
System	Loads Program and displays Login screen
User	Enters Login details (username and password)
System	Authenticates login details.
System	Allows access if Login Details correct (A1)

Setup of farm details

Actors

The following actors are involved in this use case: User and System.

Brief Description

Following successful login, the user will be able to set up the farm profile details.

Basic Flow of Events

The use case begins at the point that a success login has occurred

User	Selects profile settings from the Home Screen.
System	Loads profile settings and display
User	Enters details into data fields and selects save
System	Saves details and display save notification (A2)

Weather Report

Actors -The following actors are involved in this use case: User, System and Met Office

Brief Description - Following successful login, the user will be able to choose the weather icon to check weather forecast

Basic Flow of Events - The use case begins at the point that a success login has occurred

User	Selects Weather Icon.
System	Loads weather page and requests weather status update from Met Office
Met Office	Loads and sends weather update
System	Displays details about the weather

Calendar

Actors - The following actors are involved in this use case: User, System, and Scottish Government

Brief Description - Following successful login, the user will be able to choose the Calendar icon

Basic Flow of Events - The use case begins at the point that a success login has occurred

User	Selects calendar icon
System	Loads calendar and request update from Scottish Government
Scottish Government	Sends updated information required re deadlines
System	Displays calendar with application deadlines

Download of Forms

Actors - The following actors are involved in this use case: User, System, and Scottish Government

Brief Description - Following successful login, the user will be able to choose the Application Forms icon followed by Download Icon to choose the relevant scheme forms.

Basic Flow of Events – it begins at the point that a success login has occurred

User	Selects Application Forms icon
System	Loads application forms page
User	Selects Download
System	Displays drop down menu
User	Selects desired scheme
System	Requests Scottish Government relevant scheme page
Scottish Government	Display Scheme Guidelines

Upload Forms – Email Scan Paper Version

Actors - The following actors are involved in this use case: User, System, and Scottish Government

Brief Description - Following successful login, the user will be able to choose the Application Forms icon, followed by Email Scan Paper Version icon from the dropdown menu to upload forms which the user wants to send to the Scottish Government by email

Basic Flow of Events - The use case begins at the point that a success login has occurred

User	Selects Application Forms icon
System	Loads application forms page
User	Selects upload
System	Displays drop down menu
User	Selects Email Scan Paper Version icon
System	Loads page for Email Scan Paper Version
User	Introduces details required and selects Attach File Icon
System	Loads file where form are saved
User	Selects form
System	Attaches file and display attached message
User	Selects Send Icon
System	Sends forms to Scottish Government Email which is predefined in the system and displays Email Sent message
Scottish Government	Receives the email

Upload Forms – Upload to RPID Account

Actors - The following actors are involved in this use case: User, System, and Scottish Government

Brief Description - Following successful login, the user will be able to choose the Application Forms icon, followed by RPID to access their RPID account.

Basic Flow of Events - The use case begins at the point that a success login has occurred

User	Selects Application Forms icon
System	Loads application forms page
User	Selects upload
System	Display drop down menu
User	Selects RPID Button
System	Loads Scottish Government page
Scottish Government	Displays Login Screen
User	Introduces details
Scottish Government	Authenticates details and allow access if details are correct

Upload Forms – Save Copy to the Computer

Actors - The following actors are involved in this use case: User and System

Brief Description - Following successful login, the user will be able to choose the Application Forms icon, followed by Save Copy to the Computer which allows to scan or digitally save a form or document to the local computer.

Basic Flow of Events - The use case begins at the point that a success login has occurred

User	Selects Application Forms icon
System	Loads application forms page
User	Selects upload
System	Display drop down menu
User	Selects Save Copy to the Computer
System	Loads Save Copy to the Computer and displays screen
User	Fills in detail required
User	Selects Attach File
System	Loads the scan file folder
User	Selects forms or document which he/she wishes to upload
System	Uploads files and displays file attached notification
User	Selects save
System	Displays form saved notification

Enterprise – Profit/loss Calculation

Actors - The following actors are involved in this use case: User and System

Brief Description - Following successful login, the user will be able to choose the Profit Loss Calculation button which allows the user to choose the desire enterprise where the profit/cost calculation is desired. These calculations are based on a set of calculation pre-determined in the system and values which the user requires to input.

Basic Flow of Events - The use case begins at the point that a success login has occurred

User	Selects the Profit Loss Calculation icon
System	Loads the Profit Loss Calculation page
User	Selects enterprise where profit/loss calculation is desired
System	Displays screen of profit/loss calculation enterprise chosen
User	Introduces information required
System	Calculates profit/loss
User	Can choose to save the profit/loss calculation by pressing save
System	Saves Enterprise Profit/loss value and display details saved message

Logout

Actors - The following actors are involved in this use case: User and System

Brief Description - Following successful login and chosen action, the user can choose to log out the program

Basic Flow of Events - The use case begins at the point that a success login has occurred

User	Selects Logout Button at the bottom of the screen.
System	Logs out of program and displays You Have Successfully Logged Out message

Index of Alternative Flows

A1 – Incorrect Login details (username and password)

Brief Description - This describes the process of logging in to the system if incorrect details are entered

Basic Flow of Events - The use case begins when the user introduces incorrect details are entered

System	Authenticates login details
System	Incorrect details entered – Access denied
System	Request to re-enter username and password

A2 – Edit profile details

Brief Description - This describes the process of editing a profile details

Basic Flow of Events - The use case begins when the user wishes to change the information they have previously saved

User	User selects profile settings from the Home Screen.
System	Load profile settings and display
User	Enters new details into data fields and selects save
System	Saves details and display save notification

Appendix 5 Threat Model

Complete Threat Model

- **User login:** land manager, agent and business owner.

    - **Information disclosure** - While the authorised person has access to the system, other non-authorised personnel could see the username, password or sensitive information about the business if not appropriately secured. **Countermeasure:** masking password when entering the password details and provide the opportunity to change/update password as and when required.
    - **Elevation privilege** – Someone who should not have authorised access to system has managed to get the username and password through unauthorised means (e.g. stolen password and username). This would be a violation of the employee's moral conduct and lack of complacency with business safety polices. **Countermeasure:** masking password when entering the password details and provide the opportunity to change/update password as and when required.
    - **Elevation of Privilege** - an employee who has been dismissed from his duties may continue to hold the user name and password, which may result in the program being operated without authorization from the authorised person. **Countermeasure:** The program provides the opportunity to change or update password as and when required.
    - **Spoofing** - Someone who should not have access to system has managed to get the username and password details and change information in the system and/or sent information (potentially false) to the Scottish Government. This is done by assuming the identity of the authorised person. **Countermeasure:** User can have only access to the system with a password.
    - **Tampering** – an unauthorised person who has access to the system could insert false information, modify the existing information and upload and download documents. **Countermeasure:** User can have only access to the system with a password and information can be changed by authorised person to correct false information introduced.
    - **Repudiation** - The user could claim that the program failed to complete tasks correctly when in reality the user did not complete the task. **Countermeasure:** confirmation systems information that tasks have been successfully completed.
    - **Repudiation** – The user could think that they have completed a task or submitted a form when they have not. **Countermeasure:** confirmation systems information that tasks have been successfully completed.
    - **Denial of service** (Network infrastructure) – bad connection to the internet, some of the hill farms/businesses may suffer from poor internet connection. **Countermeasure:** The calculation of profit/loss can be done off line and developer could provide advice on how to

improve internet connection. The program will have an effective cookie service which will have this threat into consideration.

- **Connection to Met Office**

  - **Denial of service** (Network infrastructure) – bad connection to the internet, some of the hill farms/businesses may suffer from poor internet connection. **Countermeasure:** The program will have an effective cookie service which will have this threat into consideration.
  - **Denial of Service** – If Met Office server is down the connection may cause the system to crash or the information may not be loaded. **Countermeasure:** The program will have an effective cookie service which will have this threat into consideration and will display a message informing user that update is not available rather than allowing the program to crash.
  - **Denial of Service** – information may take time to upload due to problems in the network infrastructure. **Countermeasure:** The program will have an effective cookie service which will have this threat into consideration and will display a message informing user that update is not available rather than allowing the program to crash.

- **Connection to Scottish Government**

  - **Information disclosure** - While the authorised person has access to the Scottish Government Login and personal page, other non-authorised personnel could see these logging details or sensitive information about the business if not appropriately secured. **Countermeasure:** masking password when entering the password details and provide the opportunity to change/update password as and when required in the Farm Field Program. Repetition of these tasks when logging in the Scottish Government Website (two-phase login).
  - ▪ **Elevation Privilege** – Someone who should not have authorised access to Scottish Government system has managed to get the username and password through unauthorised means (e.g. stolen password and username). This would be a violation of the employee’s moral conduct and lack of complacency with business safety polices. **Countermeasure:** masking password when entering the password details and provide the opportunity to change/update password as and when required in the Farm Field Program. Repetition of these tasks when logging in the Scottish Government Website (two-phase login).
  - **Spoofing** - Someone who should not have access to system has managed to get the username and password details and change information in the system and/or sent information (potentially false) to the Scottish Government. This is done by assuming the identity of the authorised person. **Countermeasure:** User can have only access to the system with a password.

  - **Tapping** – The information sent to the Scottish Government requires travelling over the network and so there is a threat to the safety of the information at both points. **Countermeasure:** In order to combat this, encryption of information (at the point at which it is entered into the machine) will be used.
  - **Denial of service** (Network infrastructure) – bad connection to the internet, some of the hill farms/business may suffer from poor internet connection. **Countermeasure:** The program will have an effective cookie service which will have this threat into consideration.
  - **Denial of Service** – If Scottish Government server is down the connection may cause the system to crash or the information may not be loaded. **Countermeasure:** The program will have an effective cookie service which will have this threat into consideration.
  - **Denial of Service** – information may take time to upload due to problems in the network infrastructure. **Countermeasure:** The program will have an effective cookie service which will have this threat into consideration.
  - **Tampering** – an unauthorised person who has access to the system could and send the wrong information or forms, change business details in the Scottish Government business page. **Countermeasure:** User can have only access to the system with a password and information can be changed by authorised person to correct false information introduced.
  - **Repudiation** – Scottish Government may claim that they have never received the submitted application form. **Countermeasure:** Program displays confirmation message when something has been completed.
  - **Repudiation** – Users may claim that the program does not have the updated submission dates if Scottish Government changes the information on their website. **Countermeasure**: In the event of an update after login, a pop up window will appear at the home screen notifying the user of any changes.

- **Accountant**

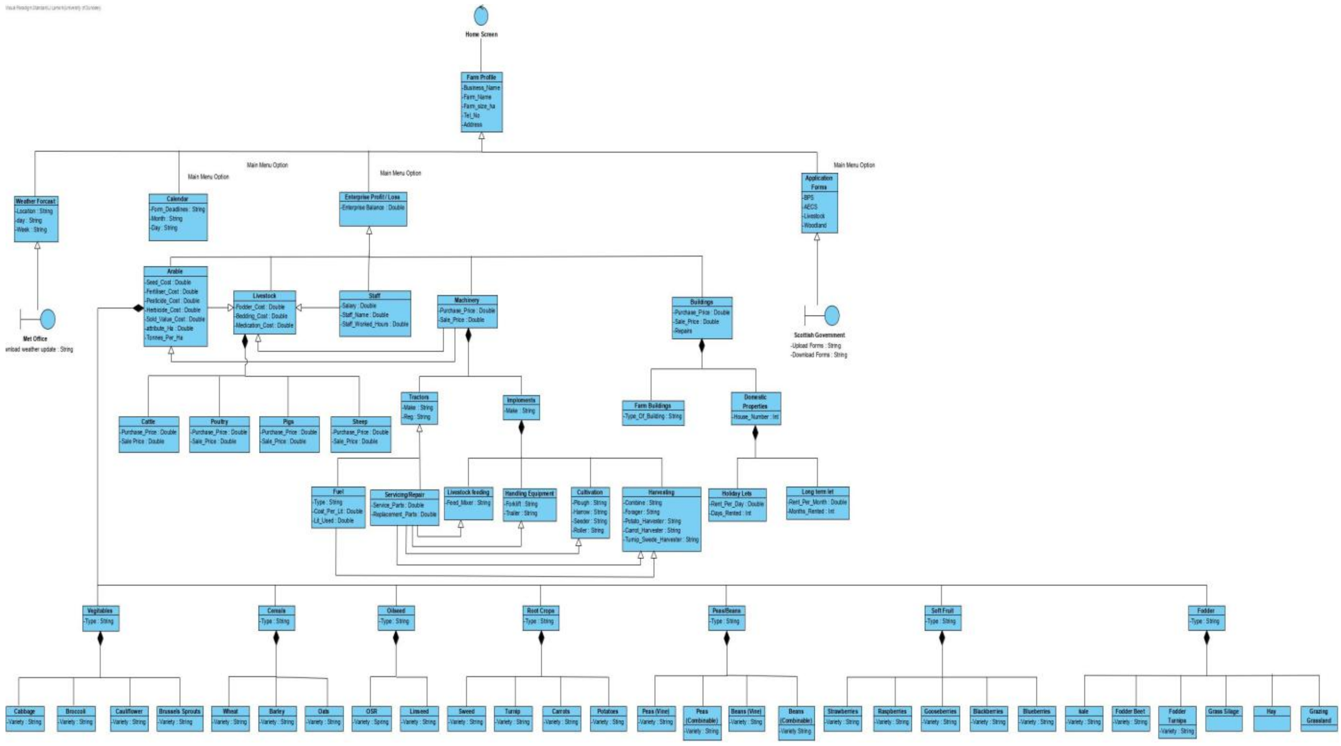
  - **Repudiation** – the accountant may state that information was not sent to him/her. **Countermeasure:** Program displays confirmation message when something has been completed.
  - **Tapping** – The information sent to the accounts requires travelling over the network and so there is a threat to the safety of the information at both points. **Countermeasure**: In order to combat this, encryption of information (at the point at which it is entered into the machine) will be used.

- **Program, updates and software developer**
  - Although unlikely, there are some potential threats that may come from the software developer himself. These could involve any of the following threat categories: lack of management and updates in the

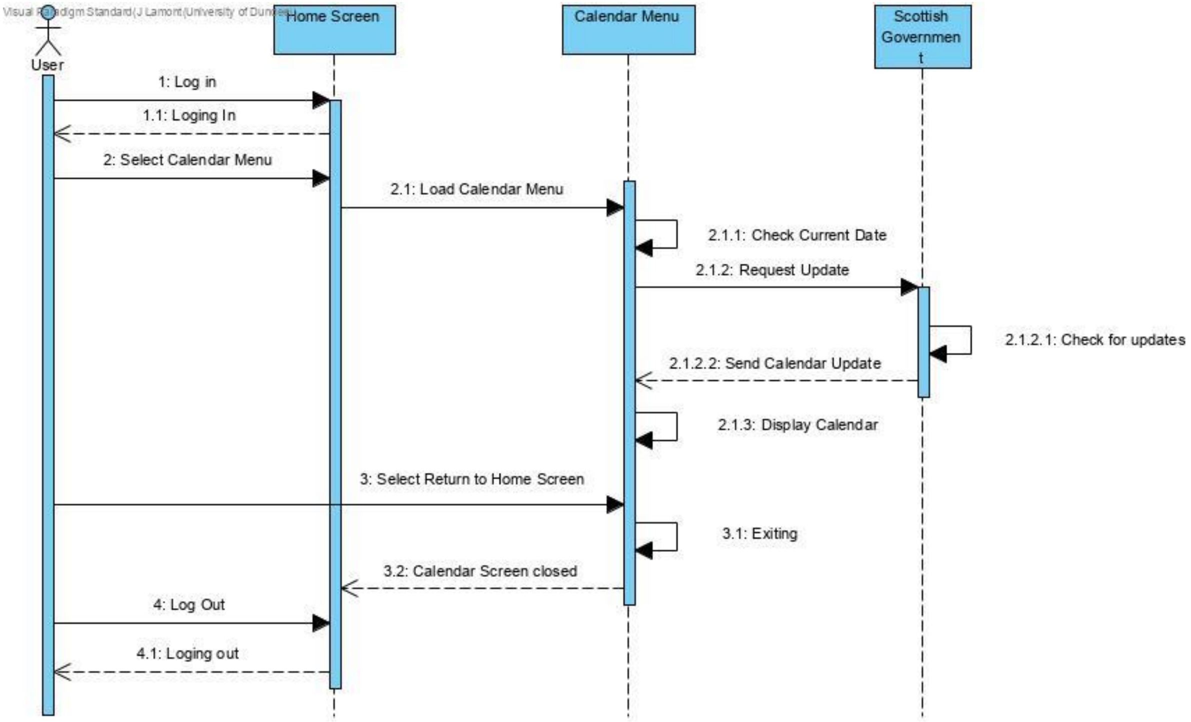
program, Tampering, Repudiation, Information disclosure, and Elevation of privilege. **Countermeasure**: Any updates must be reviewed before distribution. Developer does not have access to the information inserted by user.

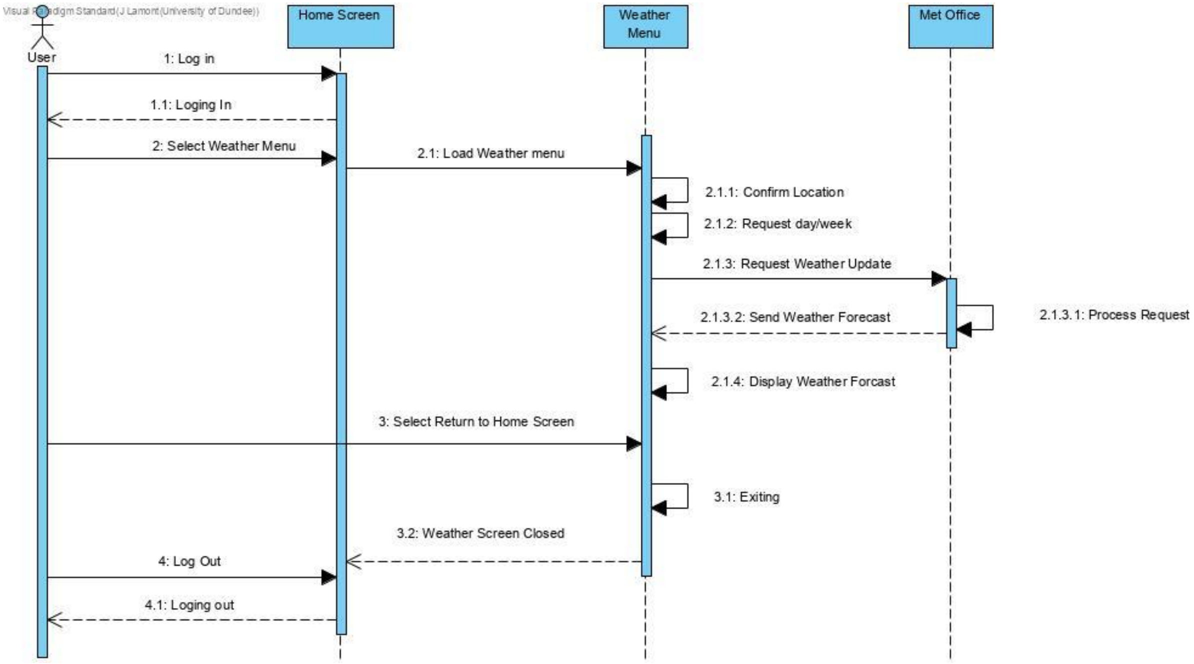
- Issues with the construction and development of the program. **Countermeasure**: Any updates must be reviewed before distribution.

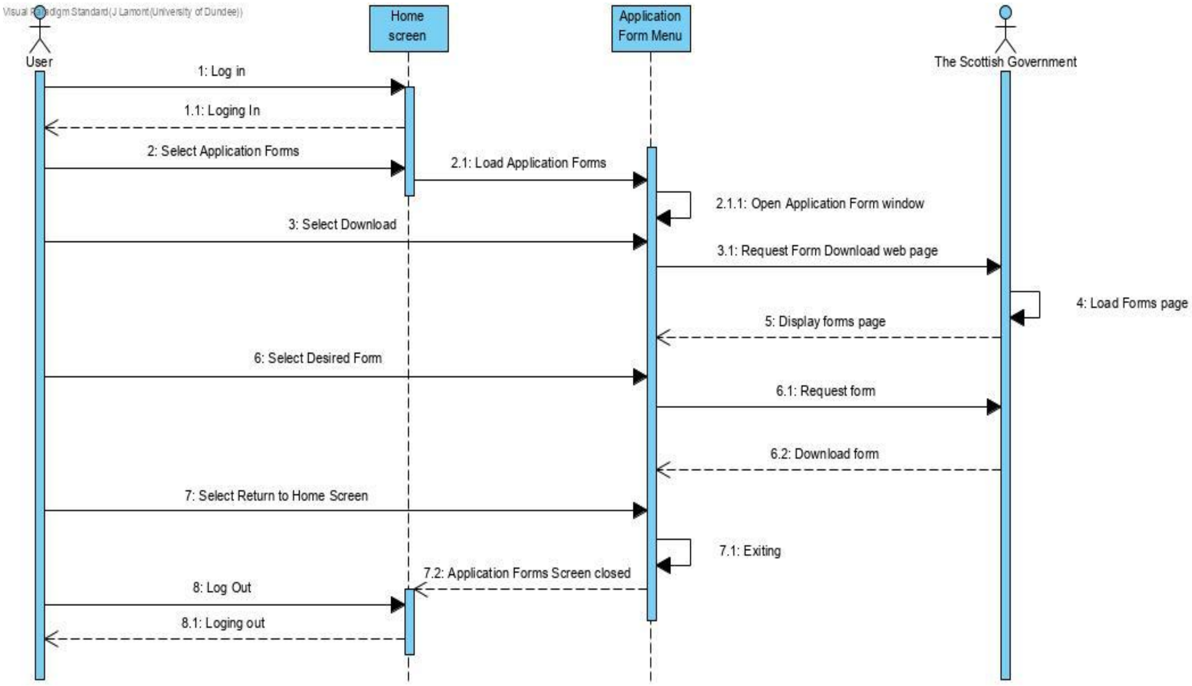
Appendix 6 Class diagram



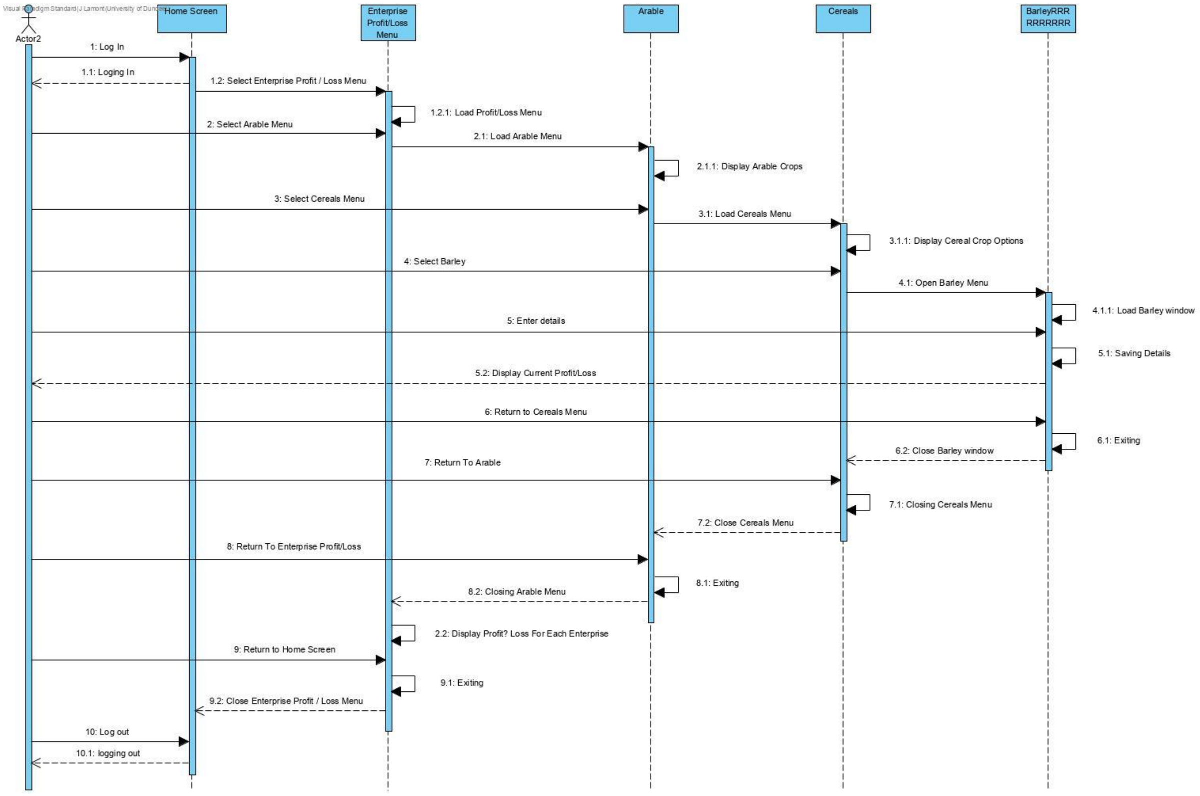
Appendix 7 Sequence Diagrams



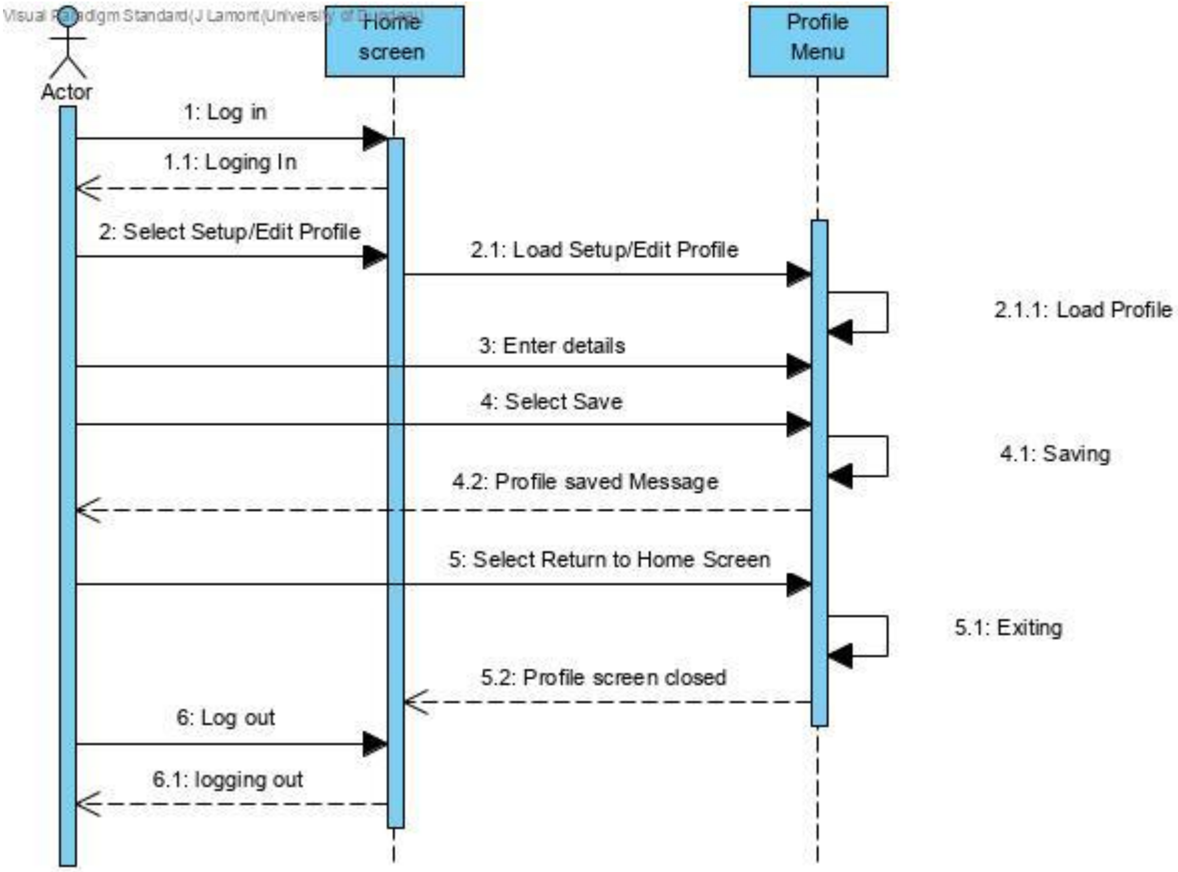


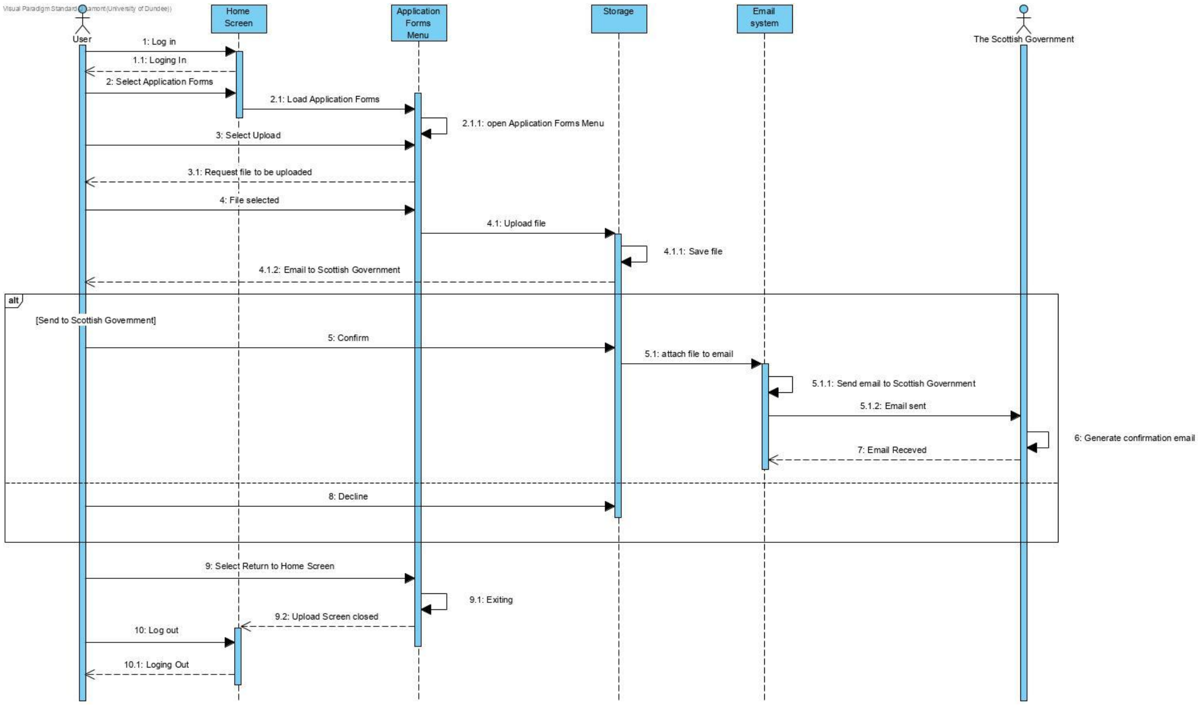


Visual Paradigm Standard(J Lamont(University of Dundee))
User
Home screen
Application Form Menu
The Scottish Government
1: Log in
1.1: Loging In
2: Select Application Forms
2.1: Load Application Forms
2.1.1: Open Application Form window
3: Select Download
3.1: Request Form Download web page
4: Load Forms page
5: Display forms page
6: Select Desired Form
6.1: Request form
6.2: Download form
7: Select Return to Home Screen
7.1: Exiting
7.2: Application Forms Screen closed
8: Log Out
8.1: Loging out



Visual Paradigm Standard(J Lamont(University of Dundee))
Actor2
Home Screen
Enterprise Profit/Loss Menu
Arable
Cereals
BarleyRRR RRRRRRR
1: Log In
1.1: Loging In
1.2: Select Enterprise Profit / Loss Menu
1.2.1: Load Profit/Loss Menu
2: Select Arable Menu
2.1: Load Arable Menu
2.1.1: Display Arable Crops
3: Select Cereals Menu
3.1: Load Cereals Menu
3.1.1: Display Cereal Crop Options
4: Select Barley
4.1: Open Barley Menu
4.1.1: Load Barley window
5: Enter details
5.1: Saving Details
5.2: Display Current Profit/Loss
6: Return to Cereals Menu
6.1: Exiting
6.2: Close Barley window
7: Return To Arable
7.1: Closing Cereals Menu
7.2: Close Cereals Menu
8: Return To Enterprise Profit/Loss
8.1: Exiting
8.2: Closing Arable Menu
2.2: Display Profit? Loss For Each Enterprise
9: Return to Home Screen
9.1: Exiting
9.2: Close Enterprise Profit / Loss Menu
10: Log out
10.1: logging out


Visual Paradigm Standard(J Lamont(University of Dundee))
Actor
Home screen
Profile Menu
1: Log in
1.1: Loging In
2: Select Setup/Edit Profile
2.1: Load Setup/Edit Profile
2.1.1: Load Profile
3: Enter details
4: Select Save
4.1: Saving
4.2: Profile saved Message
5: Select Return to Home Screen
5.1: Exiting
5.2: Profile screen closed
6: Log out
6.1: logging out


Visual Paradigm Standard(J Lamont(University of Dundee))
User
Home Screen
Application Forms Menu
Storage
Email system
The Scottish Government
1: Log in
1.1: Loging In
2: Select Application Forms
2.1: Load Application Forms
2.1.1: open Application Forms Menu
3: Select Upload
3.1: Request file to be uploaded
4: File selected
4.1: Upload file
4.1.1: Save file
4.1.2: Email to Scottish Government
alt
[Send to Scottish Government]
5: Confirm
5.1: attach file to email
5.1.1: Send email to Scottish Government
5.1.2: Email sent
6: Generate confirmation email
7: Email Receved
8: Decline
9: Select Return to Home Screen
9.1: Exiting
9.2: Upload Screen closed
10: Log out
10.1: Loging Out

Appendix 9 - Summary of supervision meeting

19/05/2020 Meeting 1

First meeting topics discussed:

- Project as a whole and what the aims are
- Market research, questionnaires and existing solutions to similar problems.
- C# discussed as a suitable programming language

02/06/2020 Meeting 2

Topics discussed:

- Questionnaire draft suitability/regulations
- C# and windows forms agreed as the method of developing project program

16/06/2020 Meeting 3

Topics discussed:

- Prototype development
- Prototype testing – delivery to testers via teams, html file attached to email
- Prototype instructions delivered by teams, video link or pdf
- Best sources of learning material for windows forms – YouTube, Dundee university library

14/07/2020 Meeting 4

Topics discussed:

- Current documentation in report – more details on the questionnaire results in appendences required
- Menu panel logic – implementation of methods to clear other menus in the onclick function of each button.
- project deadline – beginning of September

4/08/2020 Meeting 5

Topics discussed:

- The inclusion of a final evaluation of the program done by original people used for the questionnaire and prototype.
- Saving previous calculations