Croft Farm Foods software system proposal and evaluation

Owen Perry -	21905958
Luke Martin -	21610505
Callum O'Connell -	21904878
Jack Macey -	21913616

Due - Friday 21st May 2021 at 2:00 pm (+ 2 weeks for no detriment policy)

Contents

Project layout

Part 1

This part should document the team approach to the case study you are currently undertaking. You must include the following information in the report.

a) A justification of the Agile approach you are using. (Within this section you will need to critically evaluate the principles of the lean development methodology against the agile manifesto principles.)
b) The Agile Team will be required to produce a SCRUM based software development Project Plan based upon the given Case Study, to satisfy the requirements of the consortium that should include Vision Statement, Product Roadmap, Release Plan and Sprint Plan.
c) A statement of requirements compatible with the Agile approach your team has chosen. This should include:
    i) Proposed range of possible iterations(Sprints)
    ii) Identified features within these iterations
    iii) A selection of appropriate supporting user stories
d) Evidence of a workflow compatible with the agile approach your team has chosen. This should include consideration being given to:
    i) Prioritizing features
    ii) Estimating features
    iii) Monitoring progress - incorporating backlogs, burndown charts & velocity predictions
    iv) Change control
e) Evidence that the team has modelled the requirements within a suitable UML framework. This should include examples of:
    i) Use-Case diagrams
    ii) A detailed Use-Case description where relevant
    iii) Model system behaviours with Activity, State, Sequence and Communication diagrams
    iv) Document non-functional requirements
f) Evidence of Designing User Interfaces. This should include the following:
    i) Design and sketch a UI mock-up
    ii) Construct a UI prototype using a development tool
g) Evidence of team meetings and issues that have been raised
h) Evidence of risk management within the project

Part 2

This part will document your role within the project. You must include the following information.

a) Information on your role within the project. (It is also expected that you will critically evaluate the Agile Project Management roles of Product Owner. Scrum master and Development Team with regard to **Costs, Quality and Scope**.)
b) Critical reflection on how you performed in your perceived role. (It is anticipated that this section will highlight what measures you took to ensure that **Quality** was maintained throughout the development process.)
c) Critical reflection on the perception of the processes from the perspective of the other roles

Project content

Part 1

A

We will be using the Agile methodology for managing the project but more specifically the Scrum framework. We chose to use the Agile methodology over other methodologies such as Lean or Waterfall for multiple reasons. Firstly Agile works in iterations which means processes in the project are flexible, repeatable and can occur in any order or in parallel. This allows us to gather feedback from users, perform continuous testing and adapt the scope of the project during development. Both Lean and Waterfall separate each phase into a linear development line which makes it harder to adapt to change. Secondly, Lean requires more documentation to be produced than Agile which would cost us time and money in delivering the product. Thirdly Agile offers more structured practices than Lean such as defined roles, structured meetings, estimation techniques, systematic reviews and other project management techniques. Unlike Lean where trust and respect is relied upon to integrate the Lean workflow into an organisation. Fourthly Agile focuses more on developing new solutions for customers which works better for software development as every project varies. Whereas Lean is about improving the process to produce better quality products which does not work as well in software development because you are not working on the same product all the time. Fifthly Agile was designed with software development in mind which makes it more suitable for our project whereas Lean was founded in a factory to improve the production quality of cars. Finally the focus of Agile is to deliver the highest priority requirements within the budgeted cost and time. This allows Agile to adapt to new requirements or be able to re-prioritize requirements as the project moves forward. Lean on the other hand tries to produce a perfect value product for the customer through a perfect value production process. This does not work as well in software development as the customer is not looking for value but for a working product that meets their requirements. There are certain principles of both methodologies that are similar if not the same such as striving to produce a quality product. Another crossover is delivering the product promptly which helps to minimise the cost of the project and also prevents customers from changing their requirements. Lean requires that every process should be continuously inspected and adapted in order to improve it. Likewise Agile performs regular checks of the results of the working method in order to evaluate possible improvements.

B

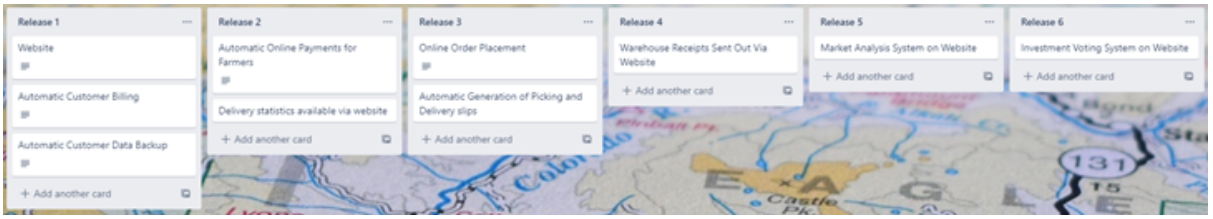
SCRUM Software Development Plan

Vision Statement

Our aim is to provide you with the highest quality product based on your criteria.

Product Roadmap and Release Plan

The product roadmap is a plan for how the product will evolve over time. The roadmap outlines future features that will be implemented into the product. A release plan sets out when certain functionality of a product will be delivered to the customer. A release plan can help to answer certain questions within a project such as when will we get the product? Which features can I get by the end of the year? How much will this cost? It also allows the team to identify important dates and be able to set milestones. Below is our roadmap and release plan combined into one.



Sprint Plan

For the sprint planning meeting the team sat down with the product owner to decide on appropriate sprints based on the requirements of the customer. We looked at the requirements and grouped them together into sprints which we thought were achievable. In the image below you can see the requirements that we have grouped into sprints.

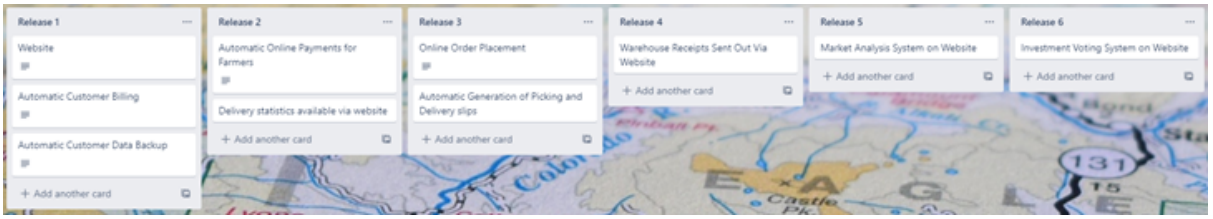
Number	Title	Description	Priority	Estimation
1.1	Register new user	The user can register a new account on the website	High	3 Days
1.2	User login	The user can login to access their account on the website	High	3 Days
1.3	Place order	The customer can place a order for the amount of the product they would like	High	4 Days
1.4	Put order in database	The order will be placed into the database which can be viewed on the website	Medium	2 Days
1.5	Generate customer invoice	The website will automatically generate an invoice for the customer and email it to them	Medium	1 Day
1.6	Send payment to farmer	The farmers should be automatically paid every Thursday. The amount they are paid will be calculated by the website which will look at the deliveries made by that specific farmer and calculate the appropriate amount.	High	3 Days
1.7	Order adjustment	The customer can make modifications to their order via the website	Low	2 Days
1.8	View delivery information	Both the customer and farmer will be able to view information on their deliveries via the website	Low	4 Days
1.9	Generate picking slip	The website will automatically produce picking slips when a customer order comes in	Medium	3 Days
2.1	Generate delivery slip	The website will automatically produce a delivery slip once a customers order has been picked	Medium	3 Days
2.2	Print slip	A slip can be printed upon request by a user	Medium	1 Day
2.3	Generate delivery note	The clerk can enter the delivery details into a tablet which will create a delivery note	High	5 Days
2.4	Add stock	Warehouse manager can add new stock to the current stock amount	High	3 Days
2.5	Remove stock	Warehouse manager can remove stock when stock is removed from the warehouse	High	3 Days
2.6	View stock	Warehouse manager can view the current stock levels for each item in the warehouse	High	3 Days
2.7	Vote for investments	The farmer can vote on investment properpostions presented by the consortium	High	4 Days
2.8	News emails	Farmers will be sent emails when a new piece of news such as a new proposal is available	Low	2 Days

Sprint Key
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6

C

User stories

Epic 3 Manage personal accounts	Epic 2 Make orders on the website	Epic 3 Automate the ordering process
User story 1 - As a user I want to be able to create an account so that i can make purchases with saved information.	User story 1 - As a user i want to be able to choose the quantity of products I buy with an order so I dont get too much or too little of a product.	User story 1 - As a client I want to be able to not have to manually print picking slips on orders so that I can speed up the shipping preperation process
User story 2 - As a user I want to be able to log into my existing account so that i can access it on any device from wherever i want.	User story 2 - As a user I want to be able to modify my order online so I can fix any mistakes that I may have made when making an order.	User story 2 - As a client I want to be able to not have to manually print a delivery slip so that I can speed up the sh ipping preperation process
User story 3 - As a user i want to be able to delete my account so that I have more control over my personal information that is stored on the website	User story 3 - As a user i want to be able to view all of my orderso on the website so I can keep track of previous purchases more easily.	User story 3 - As a client I want invoices to be automatically sent to customers by email so that they get the infromation as soon as possible without me habing to spend extra time doing so.



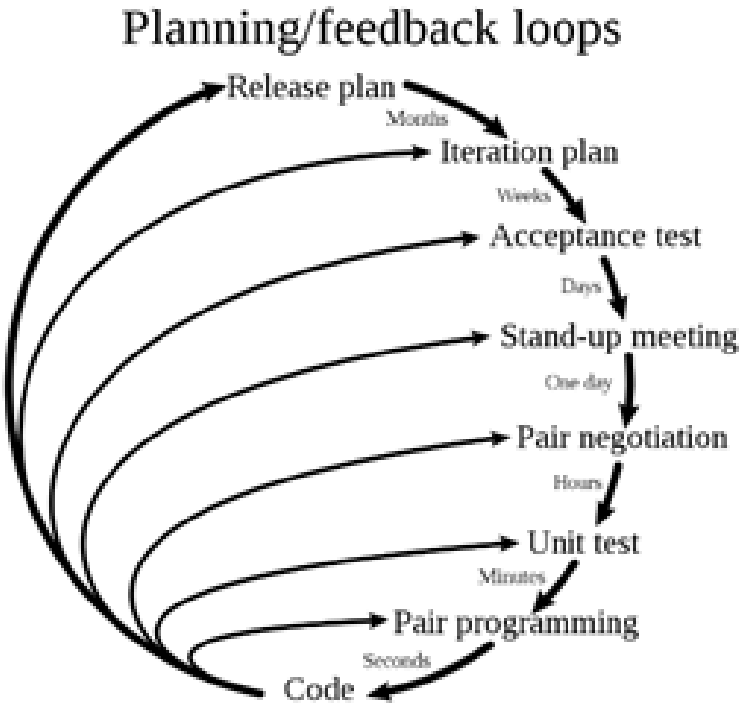
Number	Title	Description	Priority	Estimation
1.1	Register new user	The user can register a new account on the website	High	3 Days
1.2	User login	The user can login to access their account on the website	High	3 Days
1.3	Place order	The customer can place a order for the amount of the product they would like	High	4 Days
1.4	Put order in database	The order will be placed into the database which can be viewed on the website	Medium	2 Days
1.5	Generate customer invoice	The website will automatically generate an invoice for the customer and email it to them	Medium	1 Day
1.6	Send payment to farmer	The farmers should be automatically paid every Thursday. The amount they are paid will be calculated by the website which will look at the deliveries made by that specific farmer and calculate the appropriate amount.	High	3 Days
1.7	Order adjustment	The customer can make modifications to their order via the website	Low	2 Days
1.8	View delivery information	Both the customer and farmer will be able to view information on their deliveries via the website	Low	4 Days
1.9	Generate picking slip	The website will automatically produce picking slips when a customer order comes in	Medium	3 Days
2.1	Generate delivery slip	The website will automatically produce a delivery slip once a customers order has been picked	Medium	3 Days
2.2	Print slip	A slip can be printed upon request by a user	Medium	1 Day
2.3	Generate delivery note	The clerk can enter the delivery details into a tablet which will create a delivery note	High	5 Days
2.4	Add stock	Warehouse manager can add new stock to the current stock amount	High	3 Days
2.5	Remove stock	Warehouse manager can remove stock when stock is removed from the warehouse	High	3 Days
2.6	View stock	Warehouse manager can view the current stock levels for each item in the warehouse	High	3 Days
2.7	Vote for investments	The farmer can vote on investment properpostions presented by the consortium	High	4 Days
2.8	News emails	Farmers will be sent emails when a new piece of news such as a new proposal is available	Low	2 Days

Sprint Key
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6

D

The workflow that is being implemented for the project is that of the “Extreme Programming” or “XP” model for short. This model is very iterative and fits well within the bounds of the agile methodology.

With XP you can break down the agile process into eight distinctive processes that can take place across teams in parallel. The first of these is the “Release plan”. This is the highest level of the planning / feedback loop integrated with XP, it outlines the details of the project making sure key deadlines are defined and usually takes months to get from start to finish of this development plan. From there the project is broken down further into an iteration plan. This process involves having a clear understanding of what requirements need to be met within what sprints that the team(s) do. An iteration plan usually takes a few weeks to go from the start to finish of a sprint using it. These plans can be broken down into further Acceptance tests. Here the key requirements needed are consolidated into formal or functional specifications that are to be used by developers to have a firm understanding of the behaviour of the software that they are going to be making. These usually have a lifespan of a few days within a given sprint. Next is that of Stand-up meetings. These are meetings that take place for the purpose of increasing the team’s communication and allowing for clear understanding and better engagement between developers in each iteration. These meetings are usually held daily and at an allotted time. Within these days comes Pair negotiation, this is the process of two developers who are working together, optimally both bringing a differing perspective that they negotiate between to find optimal solutions to a multitude of issues that require problem solving within the actual programming of the system. This negotiating is held usually many times throughout the day as hurdles arrive that the pair need to overcome, possibly hourly. The day can further be broken down to the level of unit testing, this is the process of creating a small fragment of a program or system that has been written and maintained by two pair programmers, usually accounting for one or two specific functions within a much larger system. A unit test can be developed from start to finish within minutes with good efficiency within pair programming. Now the unit tests can be broken into the actual pair programming part of development. This is the process of two developers contributing to a given task. Usually this involves one who will be programming and the other who oversees their code in real time and acts as a second pair of eyes to not only help spot any potential errors in the code but for the purpose of brainstorming for ways to optimize existing code or solve various problems together. These happen on a second-by-second basis as they work through the day contributing to unit tests with new code. Finally, this is brought down to the lowest level of the model, the code itself, the backbone of the model and any system being developed. It is the cornerstone that connects the different pieces back to the release plan and is as low level of design as is found within a given project. The code itself can be used to reflect individually on any of the given processes within this model such that you would iterate again from wherever the code shows reasonable improvements could be made. This process is repeated until all parties are satisfied with the sprint and the iteration they have made. A visual model of this whole process can be seen below:



(Lucid Content Team)

This workflow allows us to effectively prioritize features as and when they may need to be prioritized. This is due to the nature of the described planning / feedback loop itself. The release plan at the very start of the loop acts as the driving force for this, where each release can be given its own specific features proposed allowing for the segregation for the different requirements such that those more urgently needed can be pushed forward into upcoming release plans. This has the added benefit of allowing designers of the system to have features brought either forward or back in the schedule if more times is needed for certain requirements that need to be further fleshed out to get a more functionally solid system in place when it comes to development and whichever release said feature ends up being implemented it.

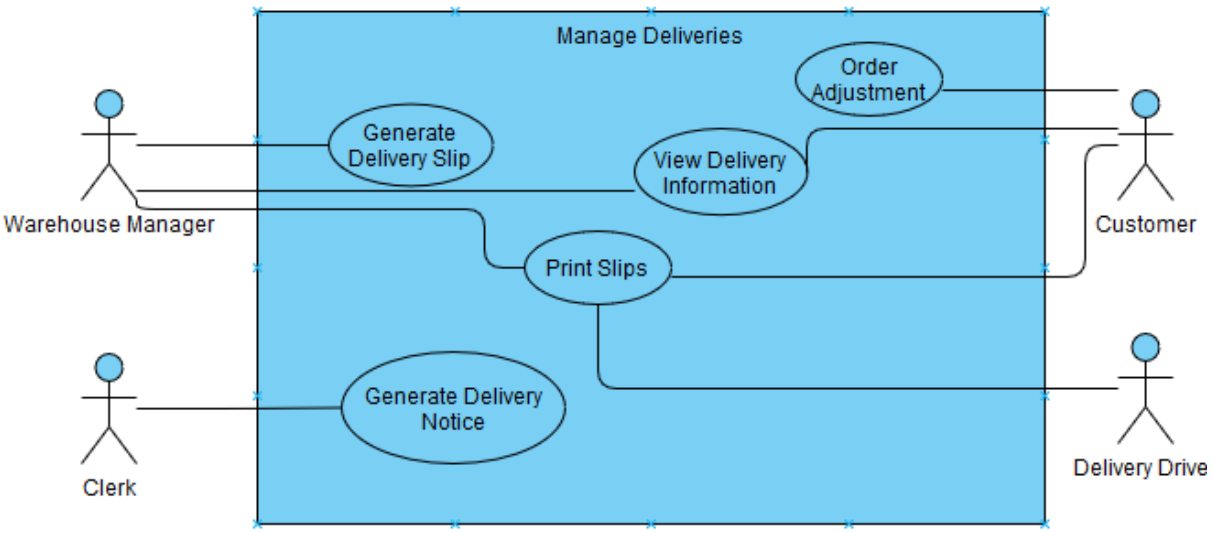
All the way from the release plan, down to even the stand-up meetings are multiple opportunities for the process of feature estimation within the sprint. This means there is ample time to determine if the feature has good business value, is estimable, is small enough to be able to be developed within an iteration and is testable. In the XP methodology these features are called “stories”, this allows for a useful approach which benefits the defining of functionality (digital.ai).

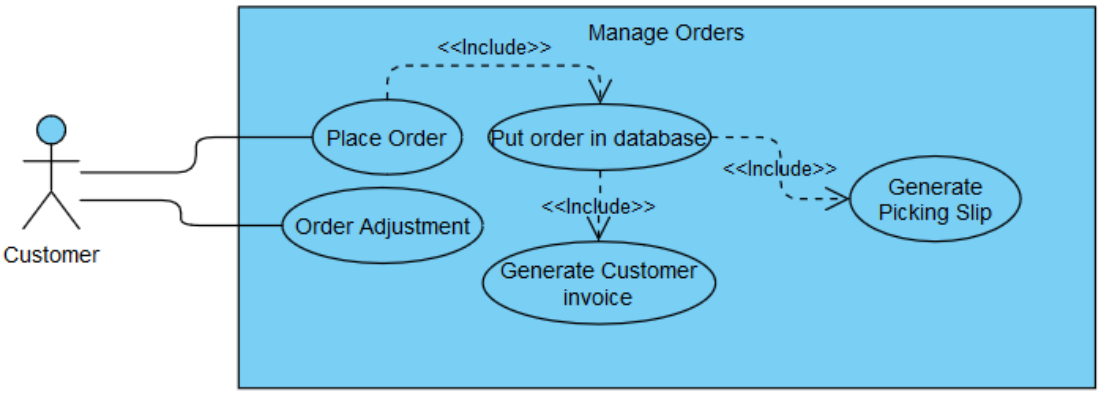
Velocity predictions are utilized to track the “velocity” of the previous sprints and thus estimate the amount of work to be done in future ones, this is calculated by adding the total number of story points completed form each given sprint and dividing it by the total number of sprints or iterations done (Lucid Content Team).

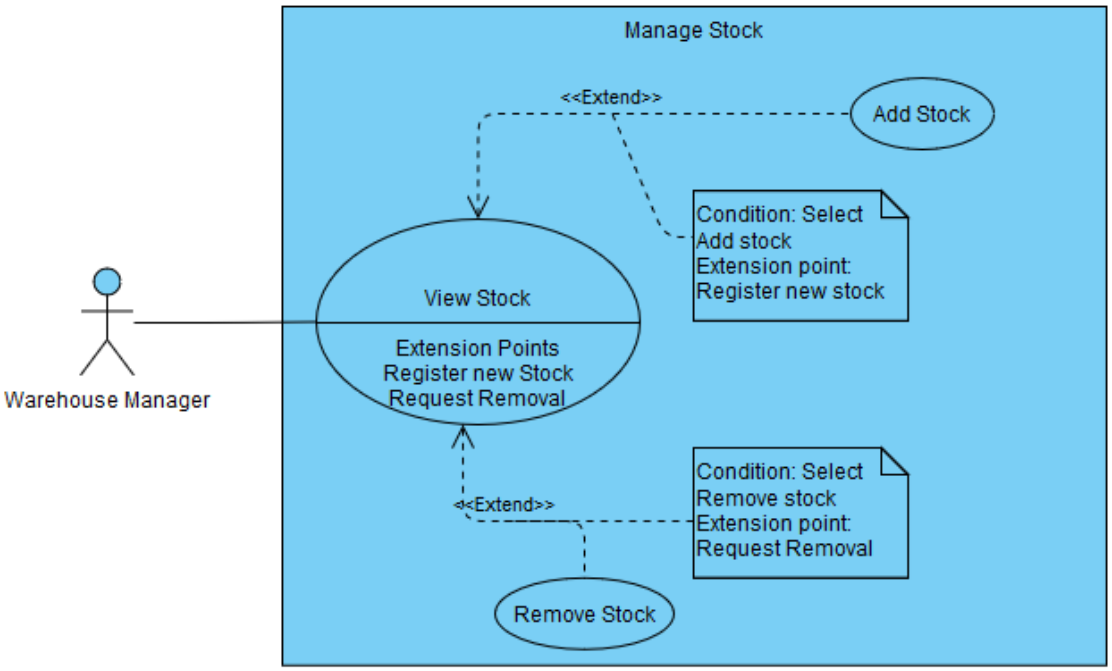
Monitoring the progress of a development team within the agile methodology, specifically that of the XP workflow is shown through the Iteration plan part of the Planning / feedback loop. Here there are a number of methods of analysing the progress of the team, for example backlogs can be used as a way to track features and requirements as they need to be developed as well as their status in regard to it. Burndown charts serve a similar purpose and can be used to track the development of certain software features through iterations in regard to the predicted work to be done, compared to the actual work done (Drumm).

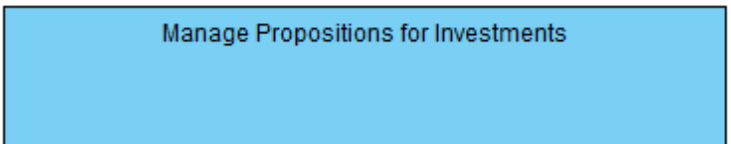
Change management within the agile methodology is very different to that of change management in for example the waterfall methodology. In waterfall we see a project go from start to finish all in one fell swoop, no iterations, no sprints, and thus any change that is needed is done at the end of development after the final finished product is shown to clients and shareholders. Within agile development change is occurring rapidly at multiple levels of development. This means that to control it is key to go into each iteration or sprint know example what needs to be changed and how, thus the shareholders and clients are shown the projects progress and asked for feedback consistently throughout the projects development instead of just at the end such that throughout each of the planning stages of the iterations the designers have a firm understanding of the parts of the system that are to be altered. This has the additional benefit of spreading the work out across the duration of the project meaning there is less need for long overtime after the project is finished like there may be within a waterfall project (Franklin).

E

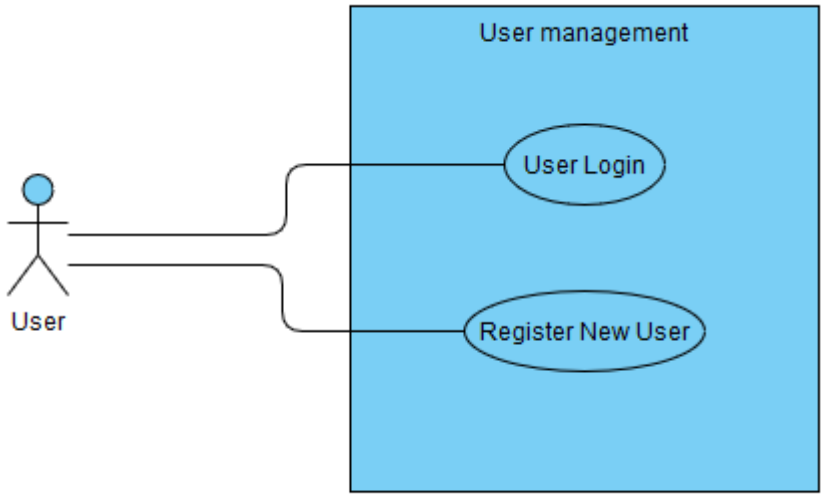

Manage Deliveries
Order Adjustment
Generate Delivery Slip
View Delivery Information
Warehouse Manager
Customer
Print Slips
Generate Delivery Notice
Clerk
Delivery Drive


Manage Orders
<<Include>>
Place Order
Put order in database
<<Include>>
Generate Picking Slip
Customer
Order Adjustment
<<Include>>
Generate Customer invoice


Manage Stock
<<Extend>>
Add Stock
Condition: Select Add stock Extension point: Register new stock
View Stock
Extension Points
Register new Stock
Request Removal
Warehouse Manager
Condition: Select Remove stock Extension point: Request Removal
<<Extend>>
Remove Stock


Manage Propositions for Investments

Use-Case Diagrams



High Level Use-Cases

Use case	High Level Description
Register New User	New users will be prompted to create an account in order to store their details.
User login	Users will be prompted to provide their login details when they attempt to use any part of the system that requires their details such as requesting a picking slip.
Place Order	Customers can place an order for the amount of product they require.
Put Order in database	Orders placed will be placed in the database which can be viewed by the relevant users.
Generate Customer Invoice	Invoices will be automatically generated by the system, then sent to the customer via email
Send Payment to Farmer	Every Thursday the system calculates the payment owed to farmers. This is done by checking the relevant deliveries made and logged on the system for each farmer.
Order Adjustment	Customers can make modifications to their pending orders.
View Delivery Information	Relevant users can request to view the information for all deliveries they have access to .
Generate Picking Slip	When an order is made by a customer, the system generates a picking slip.
Generate Delivery Slip	Once an order is marked as picked, the system generates a delivery slip.
Print Slip	Users can request for the system to print out the above slips.

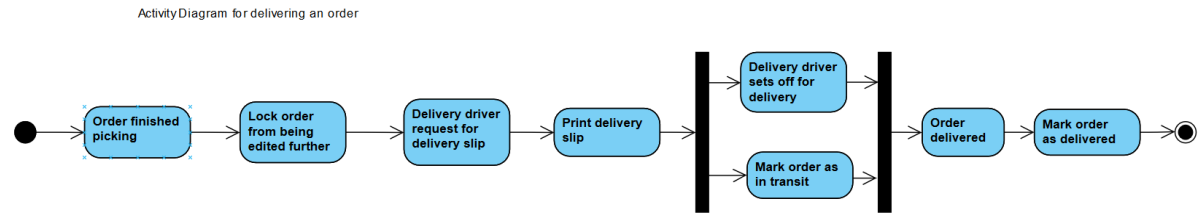
Generate Delivery Notice	A Clerk can input the delivery details to request the system to generate a delivery note.
Add Stock	Warehouse managers can register new stock into the system.
Remove Stock	Warehouse managers can remove stock from the system.
View Stock	Warehouse managers can request to view the current stock values for each item they have registered in their warehouse.
Submit Proposition	A Consortium Member can submit propositions to the system for voting.
Vote for Investments	Farmers can vote on investment propositions submitted to the system by the consortium.
Send News Emails	The system will automatically email all registered farmers when new propositions are added to the system.

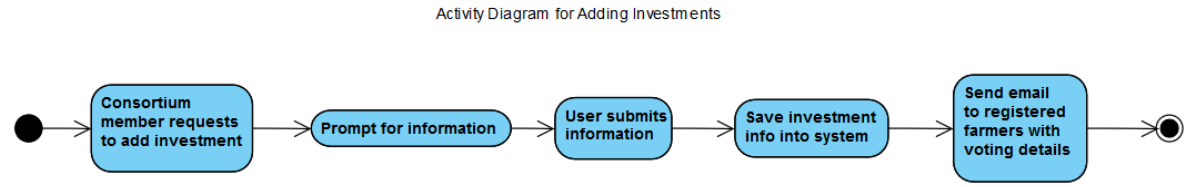
Actors

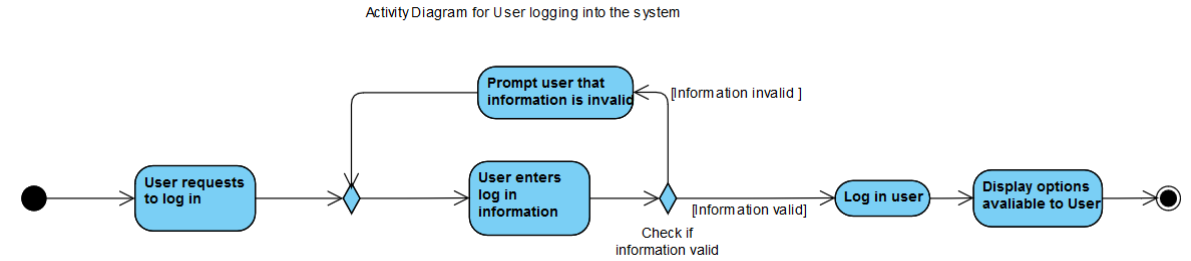
Actors	Description
Customer	General customer, any user that needs to order goods.
Farmer	Farmers are specialised Customers that can also vote on investment propositions.
Consortium Member	Users who belong to the consortium and have access to administrative features.
Warehouse Manager	Users who have the ability to adjust stock numbers for their managed warehouses.
Clerk	User with access to the ability to generate a delivery notice
Delivery Driver	Actor who is responsible for delivering the goods

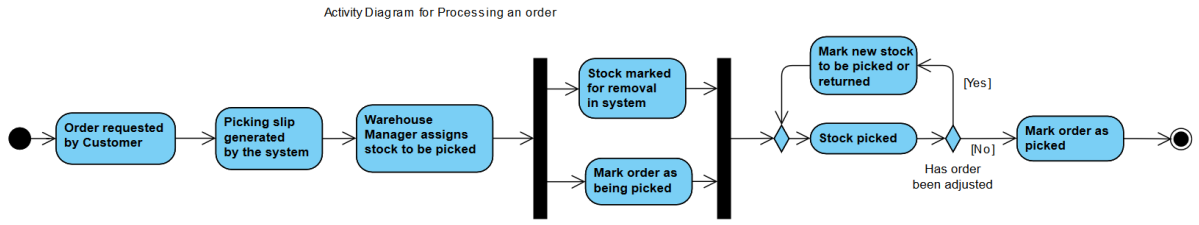
UML Diagrams

Activity Diagrams

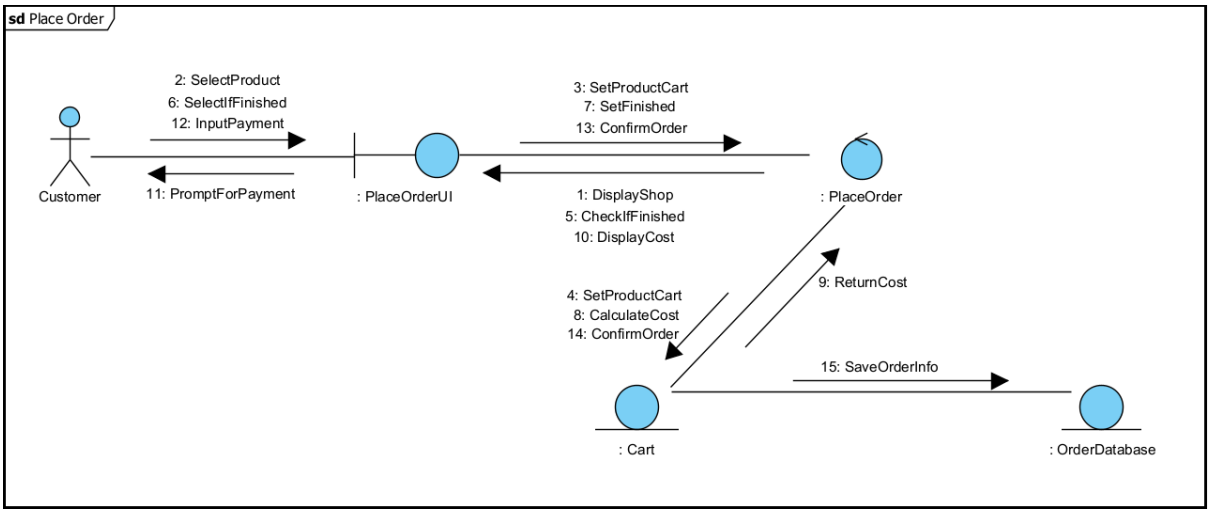


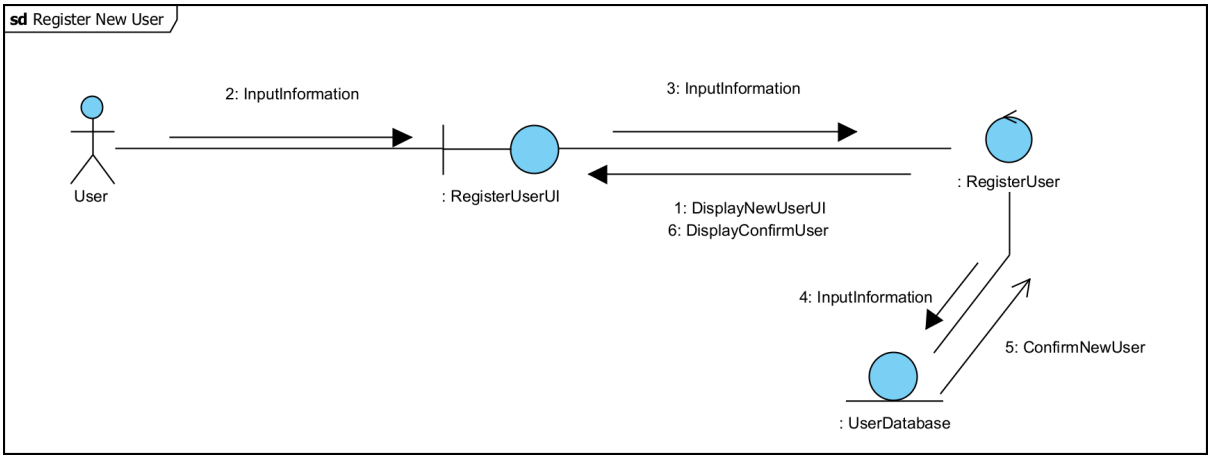


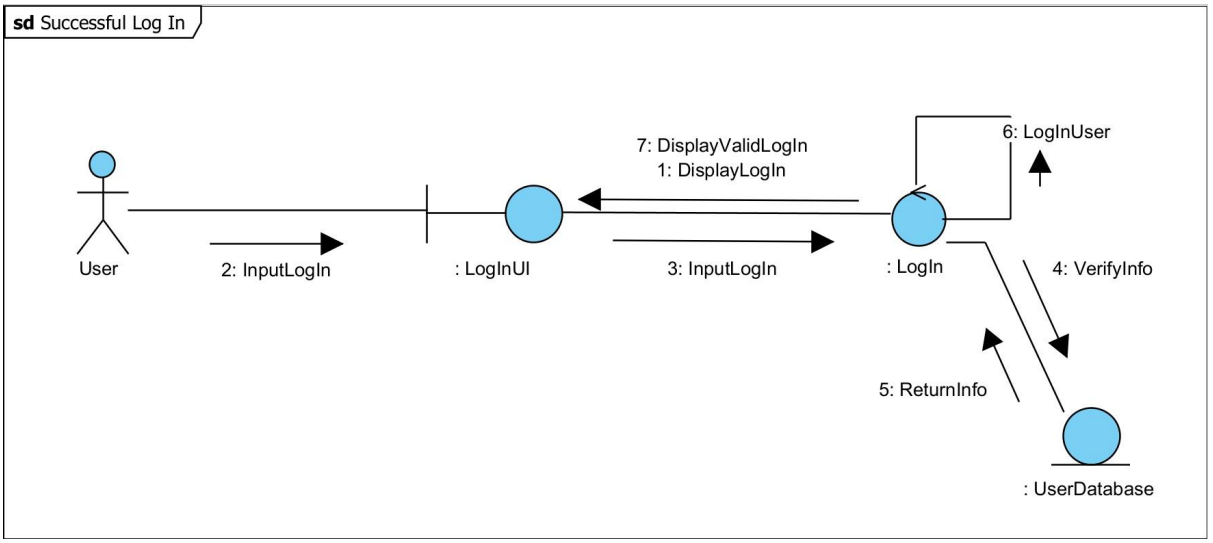


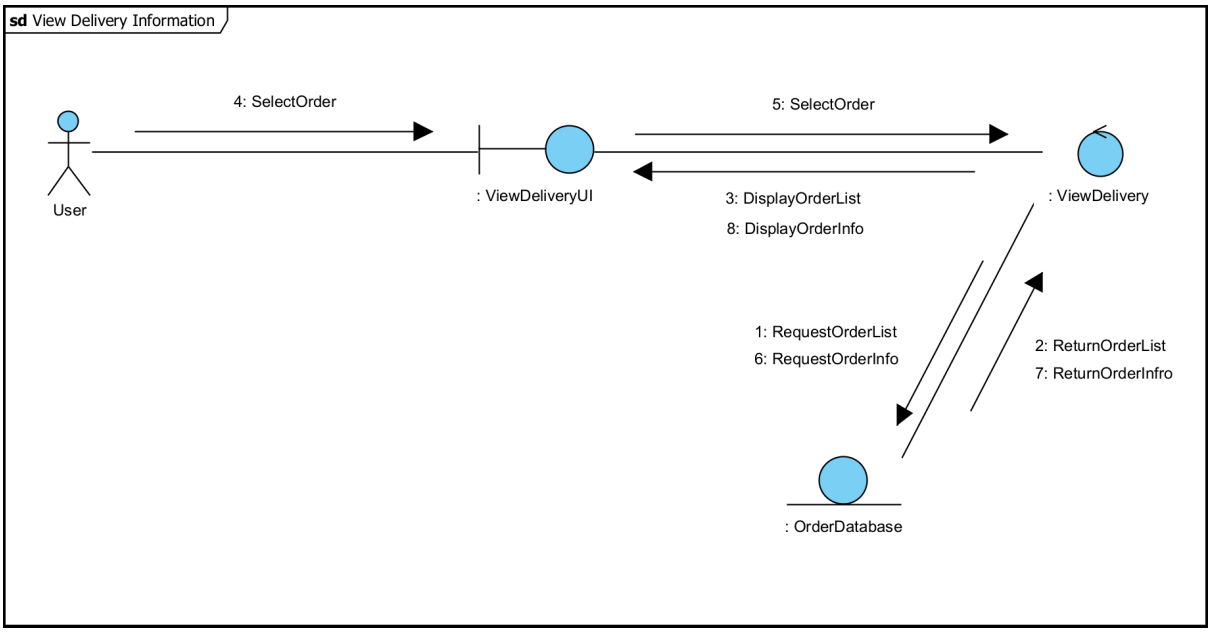


Communication Diagrams

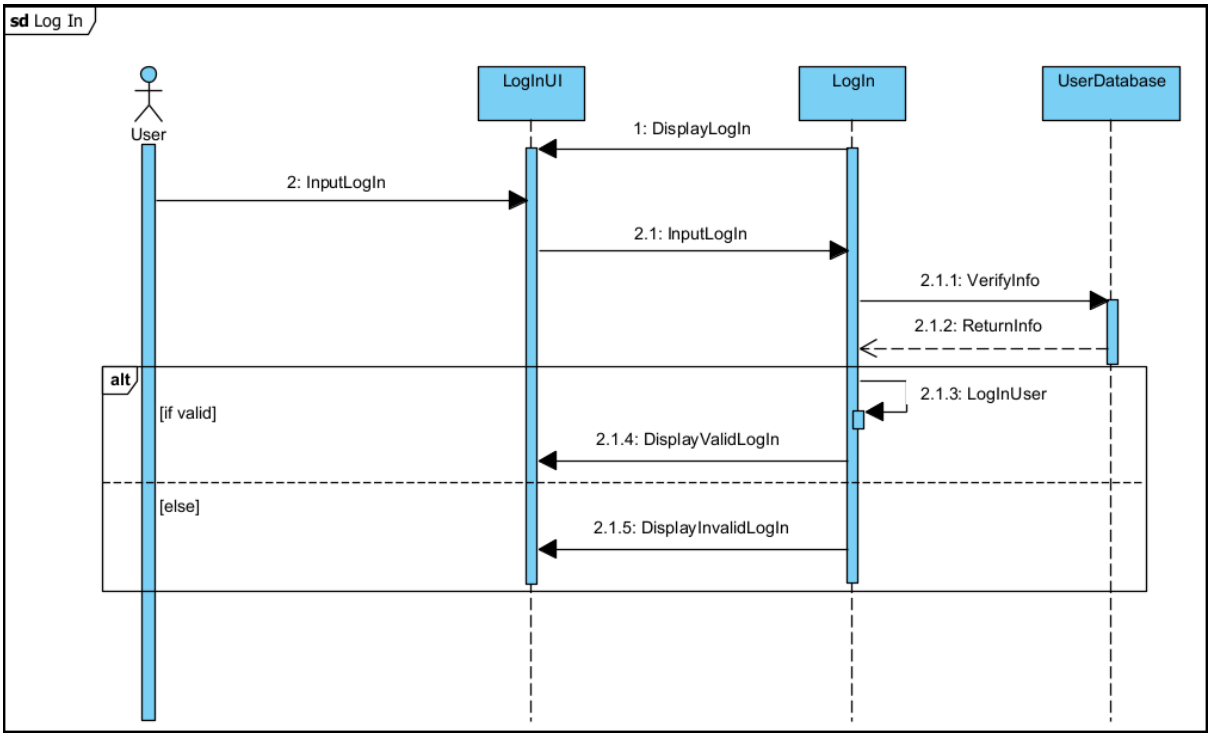


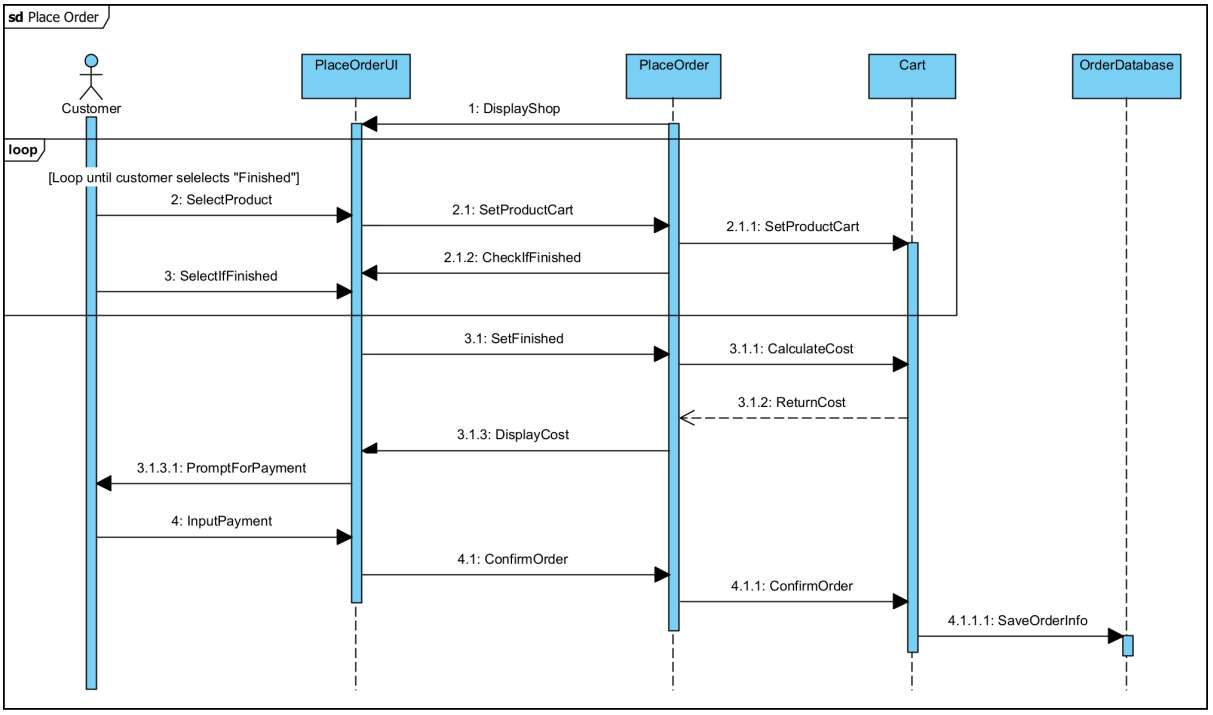


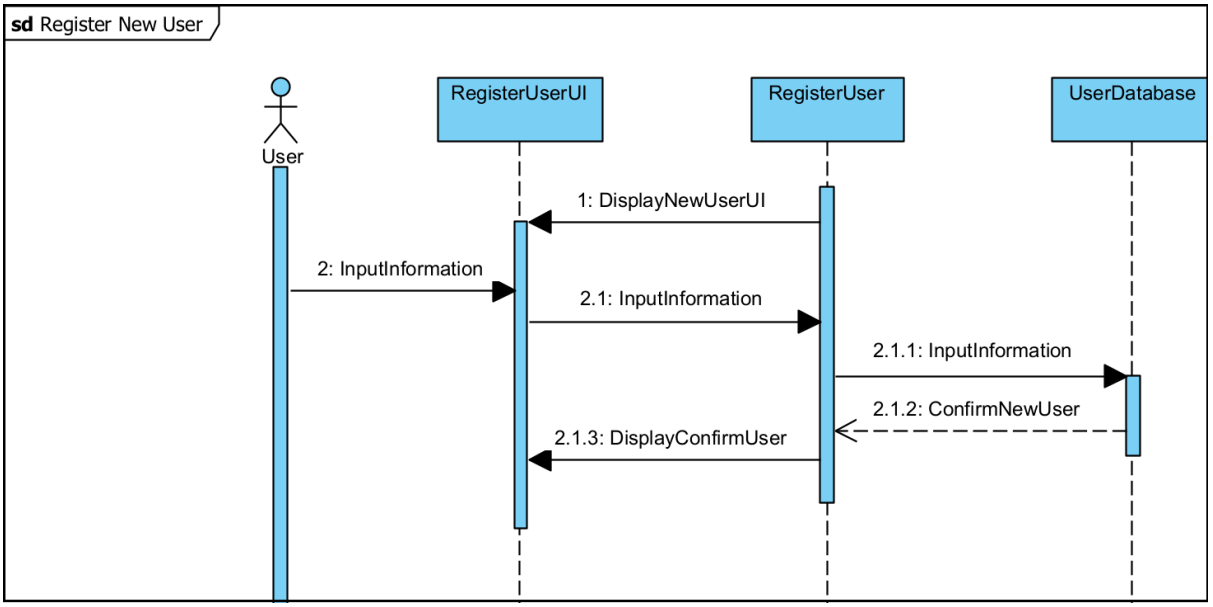


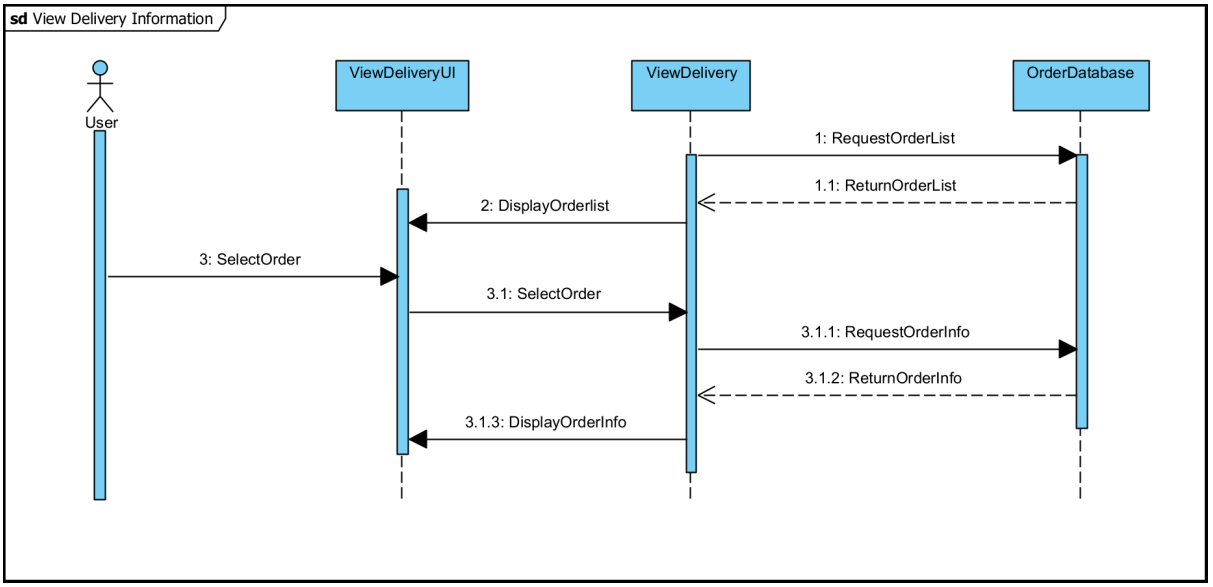


Sequence Diagrams

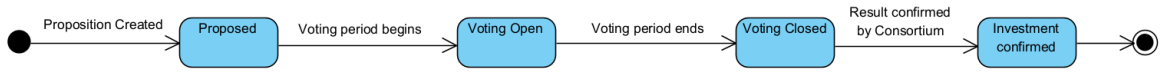

sd Log In
User
LogInUI
LogIn
UserDatabase
1: DisplayLogIn
2: InputLogIn
2.1: InputLogIn
2.1.1: VerifyInfo
2.1.2: ReturnInfo
alt
[if valid]
2.1.3: LogInUser
2.1.4: DisplayValidLogIn
[else]
2.1.5: DisplayInvalidLogIn

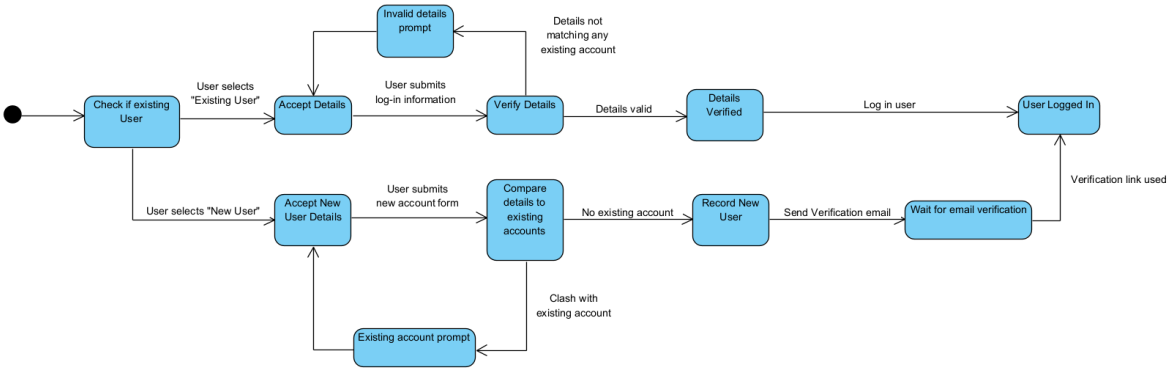

sd Place Order
Customer
PlaceOrderUI
PlaceOrder
Cart
OrderDatabase
1: DisplayShop
loop
[Loop until customer selelects "Finished"]
2: SelectProduct
2.1: SetProductCart
2.1.1: SetProductCart
2.1.2: CheckIfFinished
3: SelectIfFinished
3.1: SetFinished
3.1.1: CalculateCost
3.1.2: ReturnCost
3.1.3: DisplayCost
3.1.3.1: PromptForPayment
4: InputPayment
4.1: ConfirmOrder
4.1.1: ConfirmOrder
4.1.1.1: SaveOrderInfo

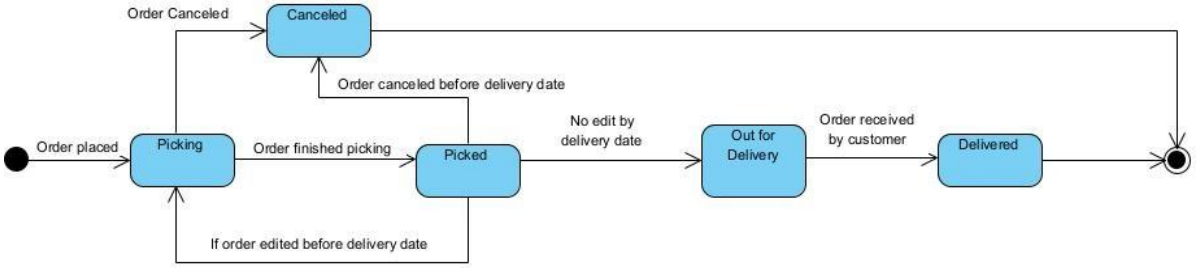

sd Register New User
User
RegisterUserUI
RegisterUser
UserDatabase
1: DisplayNewUserUI
2: InputInformation
2.1: InputInformation
2.1.1: InputInformation
2.1.2: ConfirmNewUser
2.1.3: DisplayConfirmUser


sd View Delivery Information
User
ViewDeliveryUI
ViewDelivery
OrderDatabase
1: RequestOrderList
1.1: ReturnOrderList
2: DisplayOrderlist
3: SelectOrder
3.1: SelectOrder
3.1.1: RequestOrderInfo
3.1.2: ReturnOrderInfo
3.1.3: DisplayOrderInfo

State Diagrams





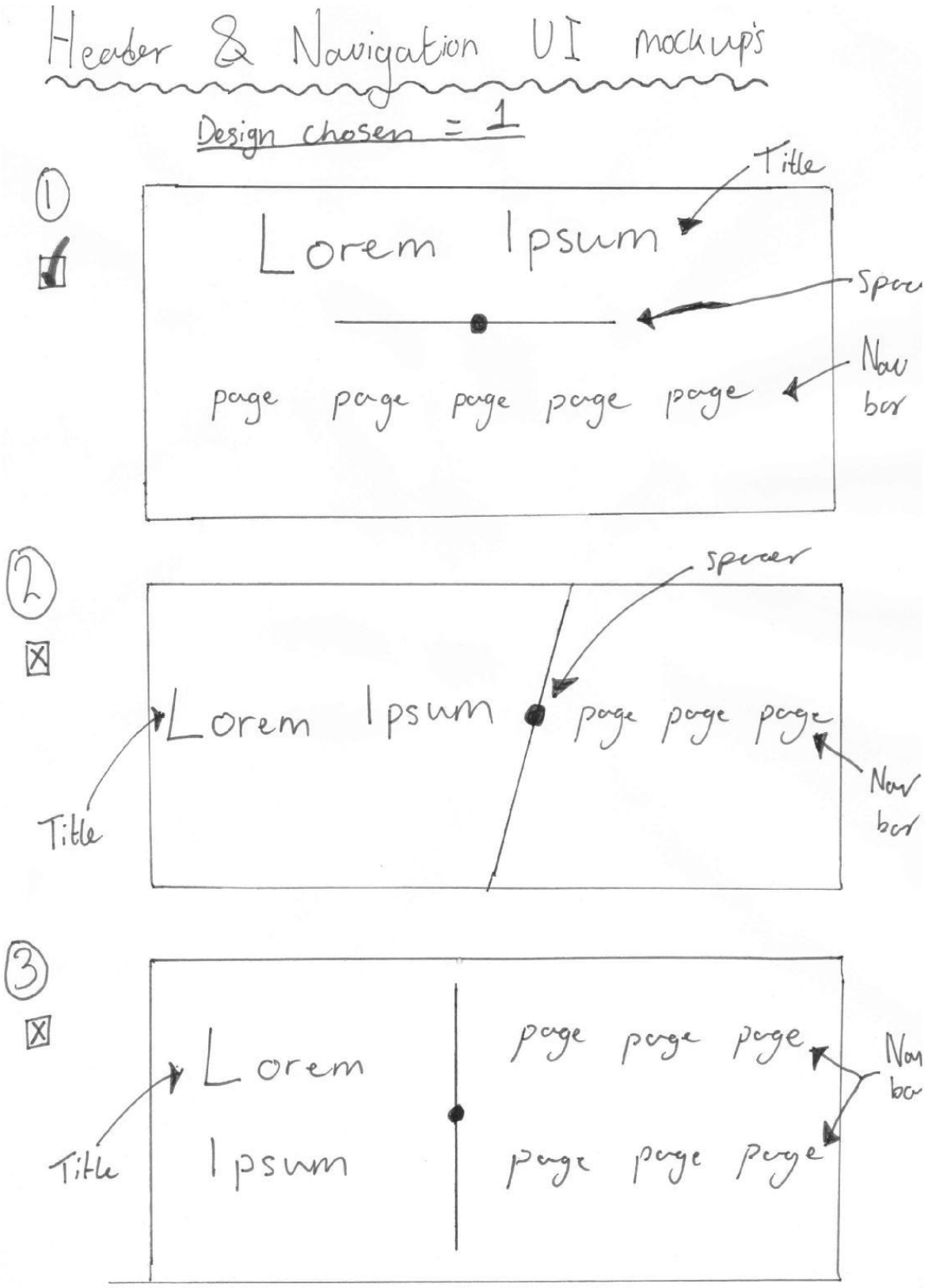


Non-Functions

Non-Function	Details	Context
Reliability	The system must always behave the same way. This is both for having the transactions be reliable in amount, as well as the timing of payments and when it is allowed for things to be edited or voted on.	Due to current worries about fraud, the system has to be completely reliable to ensure any fraud is caught. It also must be reliable in regards to delivering payments and backing up data.
Scalability	The system must be expandable, with the ability to easily integrate new users, warehouses and features.	Croft Farm Foods is currently only a group of 28 farmers with two warehouses, however the system needs to be capable of adding more should they need so in the future.

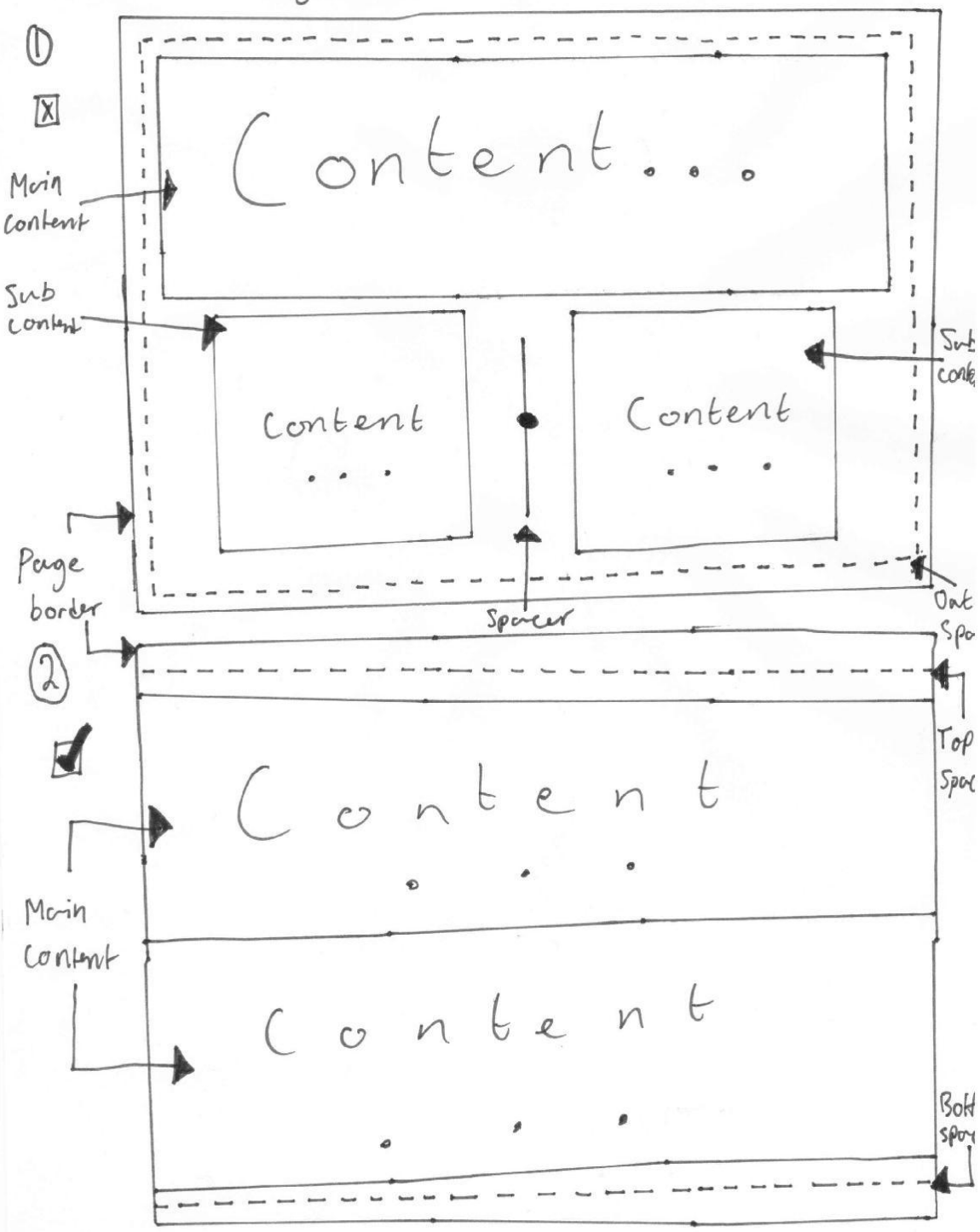
Capacity	The system by default must match the required capacity of the client.	The default system must be set up with the capacity for the current state of Croft Farm Foods, in this case being the 28 farmers, 2 warehouses, etc.
Accuracy and Precision	The system must be able to calculate the costs to a high accuracy and precision. This means that the margin of error must be negligible and result to be to the correct precision (For example, prices must be to 2 decimal points as the relevant currency is used as £x.xx).	As the system is handling orders and payments, the results must be accurate, especially as current fraud concerns are with transactions.
Security	The system must be secure against people attempting to use the system to exploit the owners/customers.	One of the major concerns for the client is security due to the potential fraud occurring.
Usability	The system must be simple to use while being sufficient to perform the required tasks.	Usability is a very important factor to this system as one of the Lancaster branch dislikes using computers, so the system needs to be useable by people with less computer knowledge

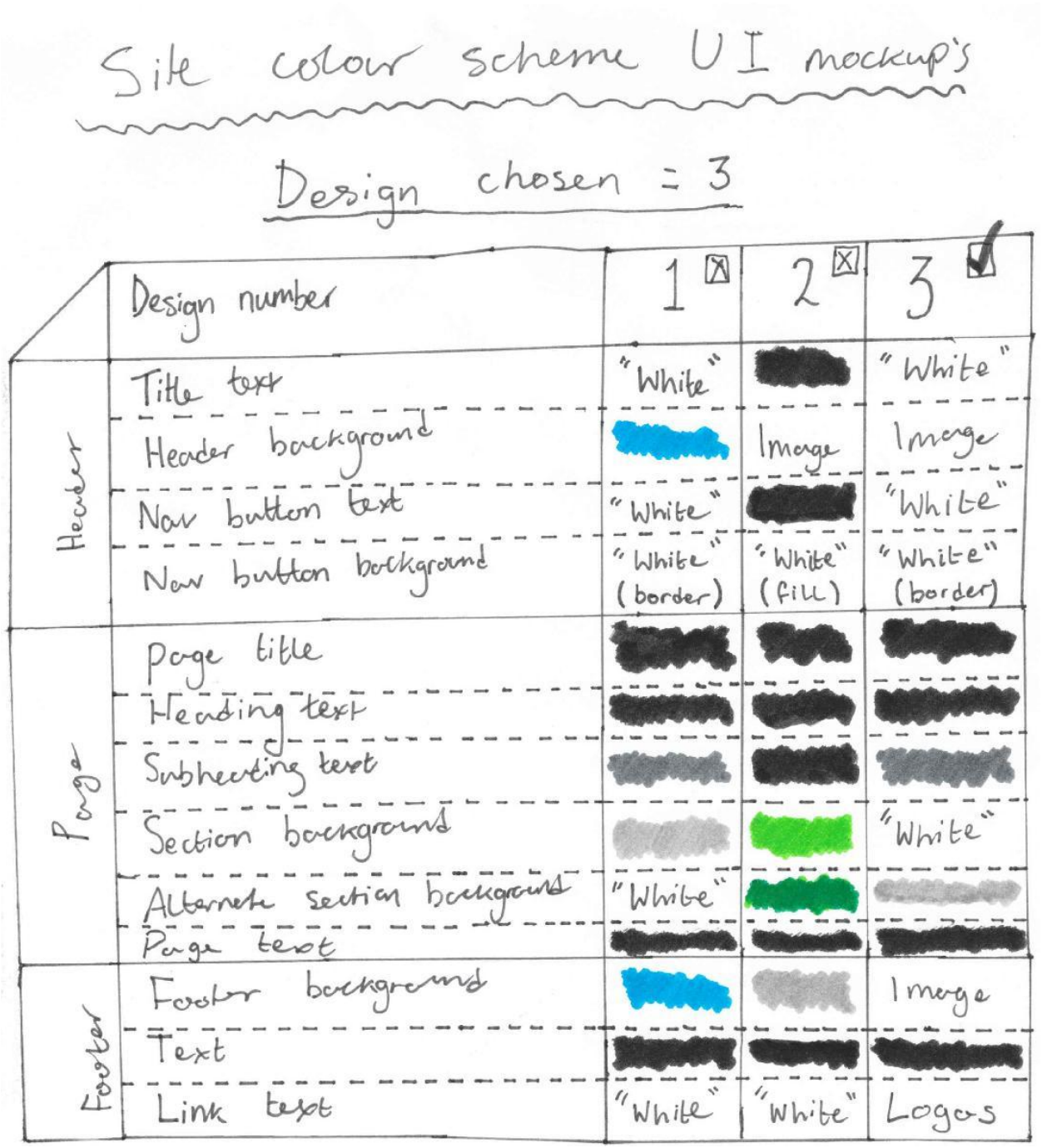
F


Header & Navigation UI mockups
Design chosen = 1
1
Title
Lorem Ipsum
Spacer
page page page page page
Nav bar
2
spacer
Lorem Ipsum
page page page
Title
Nav bar
3
Lorem
Ipsum
page page page
page page page
Title
Nav bar

Site content layout UI mockups

Design chosen = 2





UI prototype:
https://owenperry.weebly.com/

G

Daily stand ups

Sprint 1 – Issue raised, making accounts easy to understand

Callum: I am hoping you have all seen the iterations that are planned out for this project. You should have all been emailed a link to the spreadsheet with all the information present. If you have not, please let me know before we start.

Owen: No issues here.

Jack: Same all good.

Luke: Looking at the spreadsheet we are starting with the account system for the website, correct?

Owen: I also noticed that.

Callum: The base framework of the website has already been made by an external company, so we are here to add the main systems to the website.

Owen: Ah I see, that is fine then.

Jack: I must have overlooked that as I thought we would be making the base framework as well.

Callum: Regarding the account system, we want to make it as streamline as possible with the usability being our main priority.

Luke: That is understandable due to the clientele that we are going to be designing the system for. We can have Jack and Owen work on the front-end development during this sprint as they are more specialised in design. Make sure that the design is usable for any age.

Jack: I am fine with that, Me and Owen can have a sit down later and design some wireframes and discuss the design choices and we can get them back to you no later than tomorrow morning.

Owen: Are there any restrictions or limitations that me and Jack need to take into consideration when designing the wireframes?

Callum: As Luke stated you need to design the system so is easy for the clients to navigate especially if some of the clients are not used to using computers.

Owen: That is fine, we just need to go for a more minimalistic approach to the design and not a lot of flashy colours and designs.

Jack: We also need to make sure the font is readable and large enough for people who might have vision problems.

Luke: Coming back to the development of the backend for this system, me and Callum will set up all the databases to store all the data that has been input during the account creation and can be sourced once they are required by other systems within the project.

Callum: Setting up the database also needs to have a secure system that ensures that no potential threats to the system can compromise the data stored in the system. Also setting up a backup system in case any data is lost for any reason.

Luke: We need to set up a hash system for storing the user's passwords and any sensitive data.

Callum: Setting up the hash system will not take too long and developing a secure system should be our number one priority during this sprint as we need to ensure that no issues arise that could potentially lead us to getting sued or worse our client being sued.

Luke: Looking back at the information we got from our client we also need to take into consideration the availability of the system and ensure that any downtimes for maintenance of the website is not when the largest amount of data will be coming in as we know that our clients must deal with 1,600 customers and any system issues will not only make our client look bad but also will make us look bad.

Callum: That is a very important part that we need to consider when working on the system. Also looking at the document that was provided one thing that sticks out the most is that our client creates a copy of the customers master file and uploads it to a zip drive every Friday evening.

Luke: A system like that is rather unreliable so we should set up a backup system that follows the routine of backing up every Friday evening. Rather than relying on a person to upload the data the system will automatically backup.

Callum: That is understandable, it seems that we have covered everything that we need so if you have any issues, please contact me. We will set up another meeting to discuss the next sprint in the next few days. You should all get an email regarding the time.

Sprint 2 – Issue raised, Failsafe option for orders stored in the database

Callum: Morning everyone.

Jack: Morning.

Luke: Good morning.

Owen: Morning.

Callum: We successfully finished the first sprint and have a working account system set up. Though we did run into a few issues with the simplicity of the layout, that was quickly sorted by Jack and Owen.

Owen: Yeah, just a bit of a snag in the design but we did a few more iterations of the idea we had and finally found the best design for our clients.

Callum: Now onto this sprint, we are now working on the development of the order system. The system explanation was sent to your emails but if you did not receive the email or misplaced it, I will run you all through it again. For this sprint we are working on the order system such as placing orders, having the orders stored in the database and generating a customer invoice for the customer which will get sent to their email through the email that was provided in the account creation. If you have any questions, please raise them now otherwise I will continue.

Luke: No questions here.

Jack: Correct me if I am wrong but me and Owen will be working on the front end again for this as well?

Callum: You and Owen will be working on the front-end development throughout the project.

Owen: That is fine while we are here in this meeting, I will start sketching up how the order page will look.

Jack: I can start sketching up the invoice layout as well if you want me to.

Callum: Luke regarding the database would you want me to set it up or would you prefer to set it up?

Luke: I can set it up while you focus on the functionality of the ordering system.

Callum: Just make sure we can get the information from the database that is required for the invoice without any issues.

Luke: Do not worry I will design the system for the invoice set up as well.

Callum: Then everyone should know what they are doing, and we can end the meeting here unless anyone has any questions?

Jack: No questions here.

Owen: Same.

Luke: None here either.

Callum: Ok we can conclude this meeting here if anything happens, please email me and we can set up a meeting to fix any issues that have popped up.

Issue has popped up and a meeting between Luke and Callum was set up.

Callum: Thank you for emailing me, what exactly is the issue you are having?

Luke: I have successfully set up the database and everything is working accordingly but I wanted to set up the meeting as it would be easier to discuss how we would go about setting up a failsafe for the database in case anything happens we do not want our client and their customers to lose all of the data that was stored in the database as that will cause a lot of issues.

Callum: I see so you wanted to discuss the options we can take to get this sorted.

Luke: yes.

Callum: Okay so what ideas do you have currently?

Luke: I was thinking that we back up the data daily to an external source that is still controlled by our client's company, so they have a fall-back option for their data security. The only issue I think we might have been getting the client to agree to a system like this so if it is possible would you be able to contact the client and discuss this issue before I make any changes to the database and stuff that has already been set up.

Callum: I see that is a good idea for the failsafe option and I can get into contact with the client. What I recommend is to set up a second version of what you are currently doing and start to work on the backing up of a database, so we do not waste time waiting for the client to get back to us on this. Even if they decide that they do not want to go through the method you just suggested we still have the original version.

Luke: Understood, I will start working on this while you are trying to get into contact with the client regarding this issue.

Callum: No worries. Do you have any other issues that need to be brought to my attention?

Luke: Nope that was the only thing I wanted to speak about other than that we are all good to proceed to the next sprint. Jack and Owen have sorted out all the front end and the database has integrated well with the main website and the invoice system.

Callum: Excellent, well let's end this meeting here and I will contact you once I get information from the client.

Sprint 3 – Issues raised, Secure payment method and methods of physical payment

Jack and Owen are not present for the meeting as this is strictly programming based.

Callum: Morning everyone, it seems everything that we worked on in sprint 2 has worked without many issues. Luke I just want to mention that the client has agreed to the failsafe option you proposed in sprint 2 and you can now implement that into the database system.

Luke: Sounds good, I am glad I worked on that like we agreed.

Callum: Now onto the requirements for sprint 3, what we will be doing is working on the payment system for the order system we worked on.

Callum: We want to set up a system that allows for the customers to be able to send their payment through a secure payment system, Order adjustment should also be set up in case the customer wants to change something from their order and all of the systems from sprint 2 such as the generated customer invoice sending an updated version to the customer. Lastly, we need to set up a system that lets the customer and our client view the delivery information.

Luke: Sounds good, I noticed that you started to implement features that will let us just slot in the payment system to the system that we just developed.

Callum: Yes, I wanted to implement the features I knew would be used in future sprints to speed up the process of the development.

Luke: That is fine, that will help in the long run with implementing the new features without having to try and test it constantly.

Callum: So, we want to focus on creating a secure system that will ensure that no payment is lost, or fraud can happen when the customers place their order through the website otherwise the client or we will be held accountable.

Luke: Agreed.

Callum: Ok, so for the readjustment of the orders by the customer we just want to create extra sections in the database that stores the old information and the new information that has been added to the order and that will be then displayed on the invoice. The invoice will show the original order and then show the adjusted order underneath.

Luke: Sounds good. That should not take too long to implement.

Callum: Good we can end the meeting here and start working on the implementation of the features.

Luke sets up a meeting with Callum regarding physical payment.

Luke: Thank you for meeting me.

Callum: No worries, I was wondering why you wanted to set up a meeting.

Luke: I was wondering what we would do regarding physical payments as some farmers might not want to deal with using payments through the internet.

Callum: I see so you want to figure out how we can set up a system regarding physical payments. This might be an issue as we would have to contact the client and discuss it with them but for now do you have a suggestion on how we can go about adding a physical payment method to the system.

Luke: The best suggestion I can think of currently would be collecting payment when delivering the order but that does come with the issues of the customer not having the payment ready and wasting potential space in the delivery van.

Callum: I see, I can suggest this to the client, but I am not sure when it will be accepted.

Luke: That is fine, Other than that issue I have nothing else I needed to mention.

Sprint 4 – Issue raised, Client provided poor requirements and the information that should be on the slip was not present

Callum: Morning everyone, lets just jump right into the meeting as this sprint we will be needing all the time we can get to develop the system we need. So, let us keep this meeting short.

Callum: Regarding the requirements for this sprint, you should all have been sent an email with the spreadsheet explaining everything but if you have not, we are working on the slip system for once the order has been placed to allow for picking and delivery.

Jack: Are you looking for any specific designs to how the layout of the slips will look?

Callum: Not exactly but if you want something to base your designs around, I suggest something that is simple, clean and readable so there is no possibility for mistakes when picking and delivering orders.

Jack: I see that should be fine, Me and Owen can have a meeting later and discuss a few ideas on the design.

Callum: Ok that is fine, Luke since we will be working on the functionality of the system, we should split up the system so we both equally work on it, and it also gives us more time to test the features.

Luke: That is fine, what part of the system do you want to work on? As I am fine to work on any part of it.

Callum: I can work on the slip printing and generation of delivery notes. So, if you could work on the generating of the picking and delivery slip that would be great.

Luke: Can do.

Luke: Has the client specified any information they want on the slip?

Callum: I am looking at the document provided by our client, and it appears that they have not specified any of that information. I can try and get into contact with them, or you can try and set up a meeting with them to be able to discuss everything that is required for the slip generation.

Luke: That is fine, I will contact them after this meeting and try to set up a meeting to discuss with the client the requirements and details that are required for the slips and to ensure that they have not changed their mind on stuff or if they want us to add any other features to the system.

Callum: Sounds good. From the looks of it everyone knows what they need to do so I think we can call the meeting here.

Sprint 5 – Issue raised, Stock not always updating on the website

Callum: Morning everyone. We completed sprint 4 with minimal amounts of issues, Luke contacted me to set up a meeting with our client as they had left out crucial information when developing the slip system.

Callum: Now for this sprint we are working on the stock database system and ensuring the system is as streamline as possible for our client to be able to view stock on a computer or a tablet that has the appropriate set up for the database. Owen and Jack would you be able to design the GUI that will be required for the viewing of the stock on the tablets and computer. Please make it responsive so we don't have any issues of the stock not showing properly on the tablet.

Owen: That shouldn't be much of an issue when designing the responsiveness of the layout, me and Jack can sketch up some wireframes and we can come back to you with the designs to discuss if there will be any issues with them.

Callum: Sounds good.

Jack: Are we also needing to design the layout for the stock system on the website for when the customers are ordering stock?

Callum: Yes that is also required so please add that onto the list of wireframes you need to make.

Jack: I'll write it down now.

Callum: Luke regarding the database set up we also need to integrate it into the website so the stock will automatically update when any changes are made so we don't have any issues of customers buying stock that does not exist.

Luke: That's fine, I will have to change a few things on the website but that should not take too long.

Callum: That is basically everything we needed to cover in this stand up, so you are all free to leave but if you have any questions or if anything pops up during the development please contact me so we can get it fixed.

Jack: Sounds good.

Luke: I will probably ask you to sit down with me so we can test out some of the features.

Callum: That is fine.

Owen: Me and Jack will start working on the wireframes and we will send them to you both so we can get outsider opinions on it.

Callum: Ok, well that is everything. We will catch up in the final sprint.

Issue with stock not adjusting on the website.

Callum: Hello Luke, Thank you for contacting me. So what is the issue that you are running into?

Luke: When testing the database I noticed that some products on the website are not updating the stock adjustments and I tried to debug everything in the database and nothing

was throwing errors. I was wondering if there are any errors in the website's code that might not be adjusting the stocks?

Callum: I don't believe there are any issues in the website's code but I can have a quick look through it to see if there is anything causing this issue.

Luke: Thank you, I will continue to debug the database while you are doing that.

Callum: Looking at the website's code I noticed that some parts are not connecting to the database properly so that might be the issue.

Luke: Ah that's fine, I'm glad it was nothing serious and nothing to do with the database. Could you debug the stuff there so we can fix this issue.

Callum: Yeah that is fine I will do some testing on it and then we can finish off this sprint since I got all of the wireframes and designs from Jack and Owen.

Luke: Excellent, then we are on the final sprint.

Callum: Let's call the meeting here while I get this sorted for you.

Luke: Sounds good, will catch up with you soon.

Sprint 6 – Issue raised, Sends the same email 3 times to accounts whenever a newsletter or vote is sent out.

Callum: Morning everyone, we are in the final sprint now and it should be a pretty simple one. We are just going to be designing an email system for newsletters and a voting system for investments.

Callum: For this we will need to design how the newsletter will look so Jack and Owen we will need you to design a few of the options and then show them to the client and discuss some more options with them.

Jack: Will do, me and Owen will design a few options but will also do one design with the client if they are available to sit down with us just so we can implement some of the ideas they might have for the newsletter design.

Owen: It shouldn't take so long for us to design some newsletter templates, so we can get started on them once this meeting is over.

Callum: Perfect, Luke what me and you need to do is the sending of the email.

Luke: That's fine, we can set up an option in the database that will say whether or not a person wants to receive the emails.

Callum: Yeah that would be useful as not everyone wants to receive emails like newsletters and votes.

Luke: For the gathering of emails we can just call the emails from the database and push out a newsletter with all of the emails that are opted in for receiving the newsletters.

Callum: Perfect. We can end this meeting here unless anyone has any questions.

Luke: I'm good.

Owen: No questions here.

Jack: Same.

Callum: Ok, If anyone runs into any issues please contact me immediately. We can call this meeting here, thank you for joining the meeting.

Issue with newsletter getting sent multiple times

Callum: Morning Luke, thank you for setting up this meeting. What seems to be the issue?

Luke: When testing the newsletter system I noticed that it sent the email multiple times rather than once. Is there an issue on your end?

Callum: Let me have a quick test as well.

Callum: Ah yes I see what you mean, that might be an issue in my code. Thank you for pointing that out. I can get that sorted now.

Luke: No worries, other than that issue the overall system is finished.

Callum: Excellent.

Callum: Let's end this call and I will get this issue sorted.

H

Evidence of risk management within the project

Software Risks

Risk	Affects	Description
Requirements change	Project and product	During the development of the product the client might request some changes to the requirements.
Delays in specification	Project	The client taking their time to send the specifications and changes in turn delaying the development of the product
Issues contacting client	Project and product	Issues contacting the client links to delays in specification overall slowing down the development of the project
Financial issues	Project	During the development prices might change for

	and product	specific products that are required for the development of the product.
Inexperienced staff	Project	Having staff members that might be inexperienced will delay or produce a low quality product, so it is best that they are trained and guided during the development of the product.
Technology change	Business	Some of the technology used in the development of the project might become outdated fast due to the ever changing market and development of products.
Competition	Business	Other competition might come out before the system is released causing issues for our client.

Risk identification

Risk Type	Possible Risks
Technology	The database might not be able to handle the amount of transactions that might come through. Payment methods might not be secure and could cause our clients to lose money. Data breaches could happen due to computer errors.
People	Having to train all of the current staff to understand how the new system will work. Staff training may not be available right away. Staff that are important to the project might not be available due to circumstances. Having to recruit staff with the skills required in a short space of time.
Organisational	Reconstructing organisation causes teams to be split and changed to different projects. Organisation faces financial problems causing cuts in the budget for multiple projects and causing lay off in departments.
Tools	The software used for the development is outdated.
Requirements	Client requests changes to the requirements during the development of the product. If the client requests major design changes.
Estimation	The estimated time of developing the product.

	Underestimated scope of the project by the client or the project team

Risk analysis

Risk	Probability	Effects
Organisation faces financial problems causing cuts in the budget for multiple projects and causing lay off in departments.	Low	Catastrophic
Having to recruit staff with the skills required in a short space of time.	High	Catastrophic
Data breaches could happen due to computer errors.	Low	Catastrophic
Payment methods might not be secure and could cause our clients to lose money.	Low	Serious
Client requests changes to the requirements during the development of the product.	Moderate	Serious
Reconstructing organisation causes teams to be split and changed to different projects.	Low	Serious
If the client requests major design changes.	Moderate	Tolerable
Staff that are important to the project might not be available due to circumstances.	Low	Tolerable
Staff training may not be available right away.	Low	Tolerable
Having to train all of the current staff to understand how the new system will work.	Low	Tolerable
The database might not be able to handle the amount of transactions that might come through.	Moderate	Tolerable
Underestimated scope of the project by the client or the project team	Low	Tolerable
The estimated time of developing the product.	Low	Tolerable
The software used for the development is outdated.	Low	Insignificant

Risk planning

Risk	Strategy
Staff Illness	Adjust the team to be able to spread the work out across evenly to make up for the lack of the staff that is sick, briefing everyone on the situation will help with keeping everyone up to date and on track.
Component Issues	Issues with components that are used during the development of the product will be replaced with components that are known to be far more reliable and cause less of an issue.
Changes in requirements	Have a briefing to discuss how the overall changes in the requirements will impact the development of the product and maximise the amount of time adjusting the project to the updated changes.
Database performance	Look into improving the performance of the database or looking into how much it would cost to buy a higher performance database.
Underestimated Development Time	Look for external sources that could help with the development of the product or worse case scenario ask the client for extended time on the project.
Financial issues	Have a meeting with the senior management to explain how the project is a key part of the organisation and how changing the budget will impact the contribution to the business.
Organisational reconstructing	Have a meeting with the senior management to explain how the project is a key part of the organisation and how changing the team will impact the contribution to the business.
Staff training	Set up a few days during the down periods of the development of the product to have more experienced staff members teach newer members how to do things that they are required to do during the development of the project.
Data breach	Have a briefing with the team to try and find the source of the data breach and find a solution on how to recover from the breach and how to prevent future breaches by taking better precautions.

Risk monitoring

Risk Type	Potential Indicators
Technology	Constant errors in new systems that we have brought to help aid the development of the project, delays in deliveries of hardware required for the continuation of the project. System crashes and database crashes.

People	Lack of morale within the staff pool, relationships between project team members are causing delays in work, availability of job roles within the team.
Organisational	Gossip about issues within the organisation, senior management not pushing for action when needed.
Tools	Some staff members may be reluctant to use new tools, staff may demand for higher spec workstations to improve their performance.
Requirements	Too many changes in the requirements, client complaints
Estimation	Failure to reach an agreed upon deadline and failure to clear reported defects.

Part 2

Owen

A

Within the project my role was in the front end development team. I was responsible for the designing of the various GUI's that were incorporated into the final system, as well as ensuring consistently positive UX across all aspects of said system. Both of these were completed in accordance with the brief that was provided by the client. Agile project management can be split into three main roles that can be analysed. These are the Product Owner, the Scrum Master, and the Development Team.

The Product Owner is the person or persons who represent the client, those who task the development team with creating the system in question. They will provide a list of features that they require for the functioning system to be capable of and want it to be done as high a quality as possible. That being said they also often are under strict budgets so keeping costs low and thus time to completion too is another important factor to these actors.

The Scrum Master is the person that is responsible for keeping all persons of the development team working within the agile framework effectively. They are the project leads that organize sprints, teams and individuals with them for the sake of efficiency. Efficiency is important as it helps to satisfy the product owner which is important as the scrum master is the one who is tasked also with ensuring that said product wonder is aware of the development process, progress as well as any major setbacks that they may need to be aware of. They are the liaison between the development team and the product owner, enabling them to keep a close eye on the cost, quality, scope and estimated time of the project and its various aspects.

The development team is the workforce itself that is responsible for developing the front and back end of the systems along with all its UI elements and designs. They work in multiple sprints, each one taking different parts of the system developing features as they go. This is done while prioritizing the development of core systems that are crucial for functionality such that at the every least at any given time the most important things that could be developed has been developed within said time, hence why scope is important to have a firm grasp on within the development team, otherwise it can be easy for feature creep and such to lead to drawn out sprints where core functions and requirements are not being met.

Each of these roles are an integral part of the agile process. Taking any one of these out would lead to an inadequate, failed or non existing system. They each bring key skills and attributes to the table in the case of the scrum master and development team. Even the product owner having some technical skill can be beneficial as they could more clearly describe the things they need the system to do, making it easier for the development team and scrum master to turn it into functional requirements ready for development.

B

As I completed the various tasks within my perceived role I came across multiple hurdles that I had to overcome. I believe that I did this with relative ease and competence. These issues were things like technical limitations between my UI mockups and what the development tool that I used to create the functional mockup was capable of. I completed my tasks in a timely manner, never missing deadlines and did so ensuring that all elements of them were present and easily accessible to the rest of the team for reference if needed. Hence I believe I was a valuable member of the team that made a healthy and meaningful contribution to the project.

Throughout the project I used a number of techniques to ensure that the work that I was producing was of a high quality. For example when making the mockup UI i made detailed designs that incorporated a multitude of web design techniques, aesthetics and methods of getting a positive user experience. This meant that with multiple angles, designs and techniques I was more likely to capture the best possible work that would be used for the final designs of the UI. It is this process of iteration that allowed me to create the best work I could make while still making sure I keep within time frames by pre-planning how many iterations I would do on my own for each piece so as to not let it get too out of control.

C

Despite that described above, others within the project will likely not see my project the same way I do. It is possible that the product owner would see my multitude of designs, each as detailed as the last as a waste of time and thus money as I could have easily just picked one and tried to refine the single design to a finite point as much as I possibly could within the time frame. However the product owner could also find themself in a position of satisfaction with the increased quality of the final design that comes from having iterated on it multiple times. These all depend on the time and cost constraints that said actor is giving us at any given time. A similar thing could be said of the scrum master who could see the iterations as a waste of time, however I believe that this could be resolved by ensuring that they are kept in the loop with the way that I was planning my iterations to ensure that they would not eat into future sprints. Overall the other actors within the project i believe may possibly see my work as being over engineered to some extent however but still be pleased by the end result for the reasons described above.

Luke

A

For this Agile based project, my role was being part of the development team, focusing on the aspects of designing and creating the requested system based on the supplied case study. Agile Project Management roles are broken easily into three main roles: the Product Owner, the Scrum Master, and the Development Team.

The Product Owner is the role of the people who represent the business asking the Development Team to work on the system. They primarily want the system to be as high quality as possible, within the full scope of their requested features (and sometimes more if they request extra features during development) while being done with as low a cost as possible and within as short a timeframe as possible.

The Development Team is typically more focused on also keeping quality as high as they can, but they also must work with the budget and time allocated. As such they will estimate the scope, they can manage over set periods called sprints, during which they develop a portion of the scope to a working, high quality state so if they are unable to finish the whole system with the given resources, they still have a deliverable system for the Product Owner.

The Scrum Master is the role of the people whose job it is to keep the Development Team following the agile framework. They advise both the Development Team as an effective project lead for the sprints, and the Product Owner to keep them in the loop with what the team can accomplish by making estimates and helping them organise the aspects of the requested system they value most so the Development Team knows what to prioritise. They act to keep the aspects of cost, quality, scope, and time balanced.

B

As part of the development team, I helped with the requirements table by breaking down the case study, organised the requirements into the high-level use-case descriptions which I then took to develop UML diagrams to demonstrate the ways the users interact with the system via use case diagrams, and how the system's more complicated features work when broken down into Communication, Sequence, State and Activity diagrams. I also wrote up the six most important non-functions to the system with both a brief description and context as to why they were so important regarding what the case study laid out.

I focused on making sure the diagrams were as clear as possible so that there would be no confusion as to how the system would work. The reason I focused so much on clarity is that the clearer the design of a part of the system is, the easier it is for the rest of the team to later take and develop it. This reduces the amount of time it would take, which in turn reduces the cost while maintaining quality. If the diagrams or descriptions were unclear then it could delay the aspect of the system, which then could potentially compromise its quality, in the long term resulting in reduced scope from sprints not making estimated finish dates.

C

Other roles would likely see my work in a different light. The Product Owner would likely not want so much time spent on each diagram as the increased time to produce each diagram would lead to increase in cost in the long term. The Scrum Master might also not like this if I made too many diagrams with this mindset, as spending more time on the diagrams leaves less time for the rest of the things we need to accomplish for that sprint. However, they would likely also appreciate a good range of diagrams for later use during scrum meetings to streamline the explaining portion of the meetings and allow for more discussion. It also gives them a clear visual representation of the system to show to the Product Owner, who might not like the extra time spent at the time, but will also likely prefer a clear visual example as they likely do not know as much about the more technical side of things.

Jack

A

During the development of the project, I was a part of the front-end development team with Owen working on the design and implementation of the front end of the system. During the development of the GUI for the website me and Owen had to ensure that the overall user experience was up to standard through all sides of the system. The brief provided by the client was completed once we had tested all the front end and got approval from the client themselves. Within Agile projects management can be split into three roles which are the following: Product Owner, Scrum Master, and the Development team. Which I will be going into detail explaining what they are doing within the project.

The Product owner is a person who represents the client that is requesting a service. They will provide a detailed list of features that they want implemented into the system that they are requesting; they will be looking over the quality of the product during the development as they would want to have the highest possible quality system for the shortest development time and at a low price. That is due to the budget they might be set from their higher ups, so they must keep in mind the overall cost of the development of the system.

Development team is mostly focused on producing the highest quality system that is requested by the product owner, in doing so they must understand how long the development time is allocated and the overall budget that has been given to the project. Using the sprint system, they can keep track of the development of the system while also allocating the most suitable amount of time to each individual sprint to be able to optimise the development of the product while also allowing us to leave enough time for testing.

The Scrum master's job is to ensure that the people within the development team are keeping at standard of quality during the development of the system and keeping within the time allocated for each sprint. The sprints are set up by the scrum masters, also having the team and the individuals constantly checked to see how the development of the system is going. The Scrum master is the source for information between the product owner and the development team as they are the ones who discuss with the product owner the possible features and what is possible to add to the system and what might be unrealistic.

B

Within the project my role was a part of the development team and taking the responsibilities of working on the front-end development, during the development I would focus on the work that was set for each sprint, attending the daily stand-up meetings to discuss the progress and issues that we might have faced during the development. During the meetings I was also tasked to transcript the meetings I was attending so the Scrum master had a reference to each meeting and the topics that were discussed during them if they are required to talk with the product owner about how the development is going. I feel that during the project I performed my role as best to my ability while also facing some hiccups along the way due to inexperience with certain fields. That was quickly sorted by sitting down with people that are more experienced in the specific fields and then teaching me the best methods of

development. Any issues that were apparent were dealt with quickly as setting up a meeting was usually set up straight away.

C

From the perspective of the Scrum Master there were a lot of issues during the development of the project mostly due to the circumstances of having to work from home and from the lack of experience with some of members in the team. Due to the way we had to deal with the communication from back-and-forth emails and setting up meetings online, though the circumstances we were in having the daily stand ups helped with the development as it was the best method of discussing topics and issues, we had without having to send individual emails to the people in question. Though we spent a lot of time preparing the sprints and making sure that everything was perfect for the development we still only produced a small amount of the expected prototype due to the circumstances we face.

The perspective of the product owner, the project was a success even though some of the features that we wanted implemented were not present. Other than the lack of some features the development of the product was solid and the designs are what the product owner requested. The overall project was well managed.

Callum

A

Every role within the project influences the cost, quality and scope of the project. If one of these roles fails or does not perform to its fullest potential then it will affect the outcome of the project.

The product manager is responsible for prioritizing and delegating work to the development team. Failure to prioritize can change the scope of the project, cause financial implications and delays which subsequently impacts on the quality of the product. The product manager is also responsible for managing the scrum backlog which if managed ineffectively can mislead the development team and result in the wrong features being worked upon which can overall create a poor quality product. The product manager is also responsible for managing the stakeholders who are pivotal to understanding the true value and quality of the product. Effective communication is essential in keeping the stakeholders informed and involved throughout the whole process.

The scrum manager is involved in assisting the product owner in sprint planning and sprint reviews to offer reassurance that the progress of the project is continuously monitored and ensure value for money for the customer. Should the scrum manager fail to meet any of these criteria it could result in obstacles (blockers), delays, added expense and a poor quality product.

B

My role within the project was being a member of the development team. My responsibilities as a developer included delivering work through the sprints, attending daily scrum meetings, planning sprints and managing the sprint backlog. I felt that I performed well in my role and tried to maintain the highest level of quality throughout the project. For example, through maintaining effective communication. If I experienced a problem I was able to contact a colleague with more expertise in that field.

C

From the scrum master’s perspective I felt that the project did not go as smoothly as it could have. The communication should have been more effective throughout the project, as I felt it hindered the project at certain stages. However, I felt the scrum meetings we held were effective, consistent and were of value to the team. We spent a lot of the time planning the project yet we were only able to produce one prototype which was disappointing as I would have preferred more. I was able to help remove most of the blockers that came up during the project enabling the project to be kept on track.

From the product owner’s perspective the project delivered reasonable value to the client. Although we did not deliver many prototypes we did have a solid design foundation for prototypes to be developed on and implemented into the final product. The backlog of the project was well prioritised and managed.

References

References

digital.ai. “Agile feature estimation.” *digital.ai*, N/A, https://digital.ai/glossary/agile-feature-estimation. Accessed 23 May 2021.

Drumm, Shane. “Extreme Programming for Beginners made Easy – Roles & Practices.” *PM-TRAINING*, N/A, https://pm-training.net/extreme-programming-beginners/. Accessed 23 May 2021.

Franklin, Melanie. “Introduction to Agile Change Management.” *Agile Change Management Limited*, N/A, https://agilechangemanagement.co.uk/wp-content/uploads/2018/10/Introduction-to-Agile-Change-Management-v1.0-1.pdf. Accessed 23 May 2021.

Lucid Content Team. *How to estimate sprint velocity for better Agile planning*, N/A, https://www.lucidchart.com/blog/how-to-estimate-sprint-velocity#:~:text=Simply%20add%20up%20the%20total,by%20the%20number%20of%20sprints.&text=So%2C%20your%20average%20sprint%20velocity,average%20of%2032%20story%20points. Accessed 23 May 2021.

Lucid Content Team. “What Is Extreme Programming? An Overview of XP Rules and Values.” *Lucidchart*, N/A, https://www.lucidchart.com/blog/what-is-extreme-programming. Accessed 23 May 2021.