

**Department of Computing**

**Software Projects**

**(55-407815-AF-20201)**

**Name: Josh Mallison**

**Student ID: 30014437**

**Degree Route: BEng SE**

**Table of Contents**

[1. Stage 1 4](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313381)

[1.1 Software Project Artefacts 4](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313382)

[1.1.1 Project Description and Users 4](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313383)

[1.1.2 User Stories and Acceptance Tests 4](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313384)

[1.2 Software and its Presentation 5](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313385)

[1.2.1 The Software Prototype 5](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313386)

[1.2.2 Video Presentation 5](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313387)

[1.3 Incorporation of Formative Feedback 5](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313388)

[2. Stage 2 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313389)

[2.1 Software Project Artefacts 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313390)

[2.1.1 Users 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313391)

[2.1.2 User Stories and Acceptance Tests 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313392)

[2.2 Software and its Presentation 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313393)

[2.2.1 The Production-Quality Software 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313394)

[2.2.2 Video Presentation 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313395)

[2.3 Use Case Diagram 6](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313396)

[2.4 Evidence of Collaborative Work 7](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313397)

[2.5 High-Level Design (Architectural Design) 7](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313398)

[2.6 Detailed-Design 8](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313399)

[2.7 Design Review 9](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313400)

[2.8 Transitioning a Prototype to Production-Quality Software 9](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313401)

[2.9 Test Specification 10](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313402)

[2.10 Unit and Accepting Testing 10](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313403)

[2.11 Incorporation of formative feedback 11](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313404)

[3. Stage 3 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313405)

[3.1 Client Background 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313406)

[3.2 Software Project Artefacts 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313407)

[3.2.1 Users 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313408)

[3.2.2 User Stories and Acceptance Tests 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313409)

[3.3 Software and Its Presentation 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313410)

[3.3.1 The Software Prototype 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313411)

[3.3.2 Video Presentation 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313412)

[3.4 Evidence of Collaborative Work 12](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313413)

[3.5 Incorporation of Formative Feedback 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313414)

[3.6 Peer Assessment Form (Stage 3) 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313415)

[4. Evaluative Report on Legal, Social, Ethical and Professional Issues (up to 1000 words) 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313416)

[4.1 Stage 2 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313417)

[4.1.1 Relevant Issues 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313418)

[4.1.2 Discussion 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313419)

[4.2 Stage 3 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313420)

[4.2.1 Relevant Issues 13](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313421)

[4.2.2 Discussion 14](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313422)

[5. References 14](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313423)

[Appendices 15](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313424)

[Appendix 1: Software Projects- Peer Marking Form (Stage 2) 15](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313425)

[Appendix 2: Software Projects- Peer Marking Form (Stage 3) 16](file:///C:\Users\USER\Desktop\IT%20projects\Portfolio\Portfolio%20(Template)%20FINAL(2).docx#_Toc65313426)

# 1. Stage 1

## 1.1 Software Project Artefacts

### 1.1.1 Project Description and Users

Inventory Management System - In short, an inventory management system centres around; tracking purchases, this may consist of sub-categories in monitoring sale trends and customer purchasing behaviour. The system must display inventory availability, this can be across a range of contributing sectors relating to a company’s inventory i.e the liquidation of products for wholesale, distributors, ecommerce retail and manufacturing companies. All of these features are vital the functionality of an inventory management system.

|  |  |
| --- | --- |
| **Name** | **Description** |
| Shop Floor Assistant | Checks stock for customers and for re-stocking |
| Store Manager | Updates stock levels to match contents of the store, notifies area manager when levels are low |
| Area Manager | Updates and monitors trend data as well as changing order contents and capacity in response to the data. Has additional access compared with other employees |
| New Employee (Shop Floor Assistant) | Has to create new account and may use the help options available to become familiar with the system |

Tom is a store employee who wants to check the stock availability for a customer in one of the company’s stores.

Ellie is a store manager who wants to access sale trend records to make the correct order for next weeks stock delivery to one of the company’s branches.

Mark is an area manager who wants to block access of rubbers from the inventory as the sale of this product lacks profitability and is losing the company money.

### 1.1.2 User Stories and Acceptance Tests

|  |  |  |
| --- | --- | --- |
| **User Story** | **Notes** | **Acceptance Test(s)** |
| James (New Shop floor assistant) Creates new account and logs in, in order to check stock levels. |  | * Verify that the system allows James to click create new account and enter a username, password, and date of birth. * Verify that the system notifies James if that username has been used previously and if so, request that he chooses another username. * If James is unsure on the login process, verify that the system offers a help link which will display a forum on which he can ask questions and shows previously asked questions with answers. * If James successfully creates the account, verify that the system notifies him of this and redirects him to the login page where he will be able to access the stock checker. |
| Ellie (store manager) requests sale trend records for an item for order to her branch. | Different items will have differing accuracy based on how long the product has been on sale. | * If sufficient trend data for an item is available display current and previous sale records and suggest to Ellie how much to order to her branch for the next delivery. * If there is insufficient sale trend data to give an appropriate suggestion to Ellie display current/previous trends on similar products to make an estimate for how much of the product to order to her branch. * If there is no sale trend data relating to the product to display and thus no product order quantity suggestion, verify they are notified insufficient data is available. |
| Mark (area manager) requests blockage of access to inventory for a product in his branches due to lack of profitability. |  | * If there are no current orders for the product from the inventory to any of his branches display that access to the product in the inventory has been blocked to all employee’s and no current order cancellations are necessary * Any current order’s ready for delivery will be displayed to mark so mark can make arrangements for the cancellation of their delivery. * If the products are stocked within branches notify to mark that store managers may still have access to the stock within their own branch and products still have the potential to be sold. |
| Mark (area manager) Logs in with greater permissions (higher authentication) so he can change stock levels. |  | * Verify that the system allows option to update stock. * Verify the system updates stock across all user levels ie the whole database * System verifies that the correct username and password is entered for the greater permissions. |
| Ellie (store manager) Wants to access her account but has forgotten her password, she requests that she make a new one in order to access her account. | Ellie would need to at least remember her password to regain access to her account in this way. | * Verify that the system gives Ellie the option to select ‘Forgot Password?’ * If Ellie clicks the link verify that the system asks her for her username and date of birth, if the details are correct verify the system allows her to create a new password and notify Ellie that her password has been changed. * If Ellie does not remember her username, verify that the system notifes her that she will have to request her account details from an administrator. |
| James (New Shop floor assistant) requests stock availability for an item at his branch. | Some items may need more than 1 item in stock for availability because they are for display. | * If the item is available for sale at Tom’s branch specify amount of that item in stock for that day. * If the item is unavailable but is available for pre-order on the next delivery. Verify that system notifies Tom to offer a pre-order alternative to the customer. * If the item is unavailable at Tom’s current branch but is available at a nearby branch, verify that the system notifies Tom to suggest to the customer to go to a nearby branch for that item. * If there is no available stock for that item. Verify that the system suggests to Tom similar alternative products with availability in his branch. |
| James (New shop floor assistant) wants to logout of the store computer so other colleagues can access their account |  | * Verify that the system allows him to be able to navigate back to the login page from any given page. * Verify that the system allows another user to log in from the page he has returned once he has logged out. |
| Ellie (store manager) has forgotten her username but needs to access her account to make next week’s delivery order. | She will have to navigate via the forgotten password link to access this | * Verify that the system allows her to navigate to the forgotten username page. * Verify that the system provides her with the appropriate information for her to regain access to her account. * Verify that the system notifies her that an administrator will need to be contacted if she has forgotten her username. |
| Mark (area manager) wants to update the stock level for items at his branch in order to keep the correct amount of stock up to date |  | * Verify that the system allows him to select the ‘update stock’ button. * Once selected verify that the system allows him to select an item of stock, he would like to increase the quantity for. * Once he has selected an item now verify that the system allows him to select the quantity of stock, he would like to increase it by. * Now he has selected the item and amount verify that the system notifies mark that the stock has been updated and the quantity is changed, visible in the inventory table. |

## 

## 1.2 Software and its Presentation

### 1.2.1 The Software Prototype



Extract This zip file to access the prototype, press F5 to run.

Note: you may require ‘Mockplus’ in order to run the program, also uploaded at submission point

### 1.2.2 Video Presentation

<https://youtu.be/o9ubBSWyDqo>

Press ctrl + left click to open.

## 1.3 Incorporation of Formative Feedback

In a zoom meeting during the initial stages of the prototype’s development, Jamie Hufford suggested that there be different levels of access for different users based on their importance within the company. This was to increase the integrity of the company such that a regular shop employee does not have the power to change inventory or block inventory the same way a company’s CEO would. In response to this feedback I added in a manager authorisation login, which would allow only those with higher authentication passwords to access the features of the system that allow users to perform those kinds of actions.

# 2. Stage 2

## 2.1 Software Project Artefacts

### 2.1.1 Users

The project aims to design and produce a production quality system for an inventory management system. From a business perspective the system will aim to provide a desktop application system which will meet all basic functionality for management of inventory, in this case for a stationary company.

|  |  |
| --- | --- |
| **Name** | **Description** |
| Retail Assistant | logs onto application to check stock for customers and collect relevant data in order to serve customers. |
| General Manager | Checks weekly, monthly, and annual trend data to make correct orders for each week and advise their team on improvements that can be made. |
| Stock Officer | Keeps track of stock and checks stock levels to make sure they are up to date and correct. |

Jack is a retail assistant who wants to create a new account to access the stock data in order to serve a customer who is requesting pencils.

Jonathan is a general manager who needs access to sale trend data within the company in order to make decisions for how much product to buy and sell in order to keep the company as profitable as possible.

Joel is a stock officer who wants to update the stock level after the company restocked its inventory for all of its product, he wishes to gain access to the inventory stock levels in order to keep the system accurate to the true level of stock held by the company.

### 

### 2.1.2 User Stories and Acceptance Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **User Stories** | **Background and reasons for wanting a new inventory management system** | **Notes** | **Acceptance test(s)** |
| Johnathan (General Manager) | Johnathan is a manager of a wholesaler. He has an old, outdated inventory management system and is looking for a new one, which will be friendly and easy to use for his employees, functionality will be a main priority in this as it needs to be friendly for staff members who have disabilities. | Manager is to get full access to all parts of the system. | * Verify that the system is fully accessible to the general manager only. * For the general manager and the admin team to set users depending on employee status. |
| Jonathan (General Manager) | Jonathan would like to be able to access trend data for stock that has been ordered each week. This should be available to view weekly, monthly, and annually. | * The manager would like to be able to access the data for the stock ordered and link it to sales to be able to determine which month had more sales and to be able to prepare for the periods where its busy. * The manager will also be able to look up the bestselling products and compare them to when they were discounted, | * Only the manager will have access to this data, if an employee tries to access this it will come up with an error. * The manager will be able to search for the data on the best-selling items in the store and online. * The manager is also able to view the frequency of returned products, this will help in finding out how many shoppers were happy with their purchase. |
| Jonathan (General Manager) | Jonathan’s old system did not allow him to set one-time passwords for his employees if they need to access different parts of the system. however, he would like this functionality on the new system. | * The manager will be able to set access rights to parts of the system without the help of admin. | * This functionality has been added for the managers peace of mind and total control over the system. this is to reduce discrepancies and exploitation of the system. |
| Jack (Retail Assistant) | Jack (retail assistant) wants to login into existing or new account to access stock data for a customer |  | * Verify that the system gives Jack the option to create a new account from the login page. * The system must verify that Jack enters; a username he would like to use, his date of birth, choose a password and then to re-enter the password. * If Jack selects a username that is already in use, verify that the system notifies him of this and ask him to choose another. * If Jack is unsure on the login process, verify that the system offers a help link which will provide support in the account making process. * Verify that the system notifies Jack when the account has been made successfully and then redirect him to the login page. |
| Jack (Retail Assistant) | Jack would like to access the main menu from all available screens on the system |  | * Jack will be able to access the main menu from the different screens. * Verify that the system closes the previous window when the main menu is re-opened |
| Jack (Retail Assistant) | Jack requests in-store availability of an item at his store. | Some items may require more than one item in stock to be available as they may be for display purposes. | * Verify that the system gives Jack the option to ‘check stock’. * Verify that the system requests him to select his current store and then request which item he would like to check for. * If stock is available verify that the system notifies him of how many are in stock/are available as well as the date and quantity of the last/next order * If no stock is available and there are no current orders for the item verify that the system give him the option to ‘suggest order’, this will notify a senior employee that an item has been requested with no stock and no current order. |
| Joel (Stock Officer) | The general manager did not have sufficient security on the old system meaning that the employees could have logged in via username. | The stock officer will not be able to log in with a wrong password as an error message will pop up if he puts in the wrong password. | * Error message will pop up if wrong password has been entered. * Verify that the system allows him to re-enter his username and password to once again try logging in. |
| Joel (Stock Officer) | The stock officer would like to add a new product to the system | In order to access reports on a new product the program must be restarted. | * Verify that the system allows him to enter the details for the new product. * Verify that the system allows him to save the new product to the database. * Verify that the system adds a reports page for the newly added product. |
| Joel (Stock Officer) | Joel would like to increase the quantity levels of his existing stock inventory |  | * Verify that the system allows him to enter how much quantity they would like to add. * Verify that the system allows him to save the added quantity, on an existing product, to the database |

## 2.2 Software and its Presentation

### 2.2.1 The Production-Quality Software

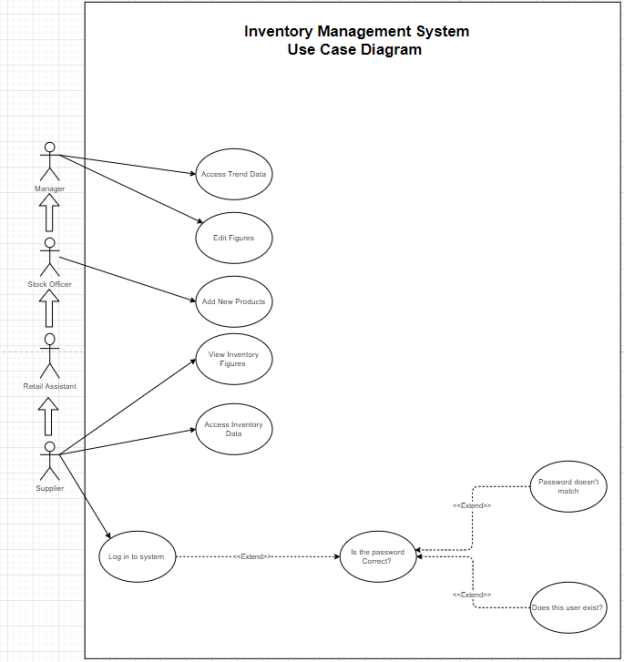
The Inventory Management System program was uploaded to blackboard at the relevant submission point by one of our team members.

### 2.2.2 Video Presentation

https://youtu.be/[rDFOacH2Xpc](https://youtu.be/rDFOacH2Xpc)

Press ctrl + left click to follow the link.

## 2.3 Use Case Diagram



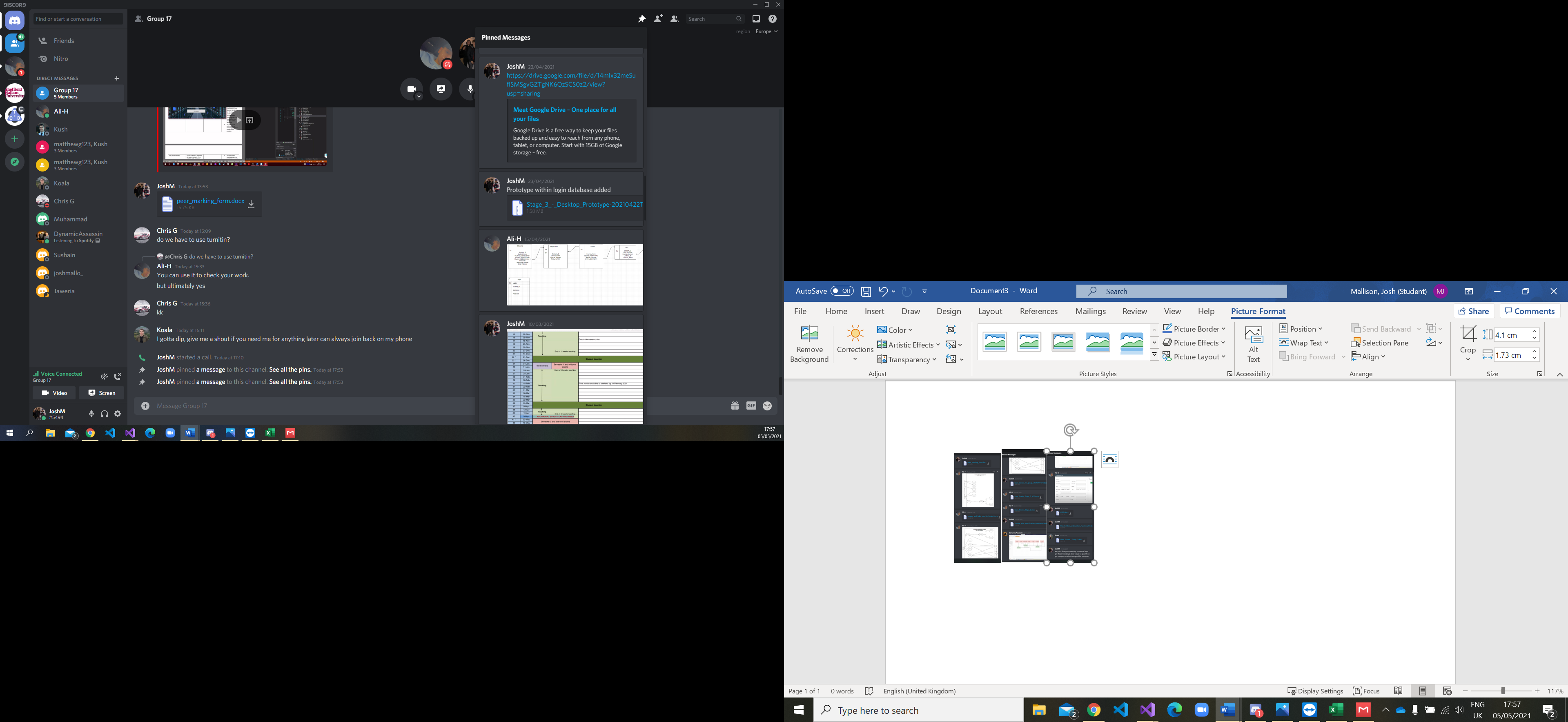
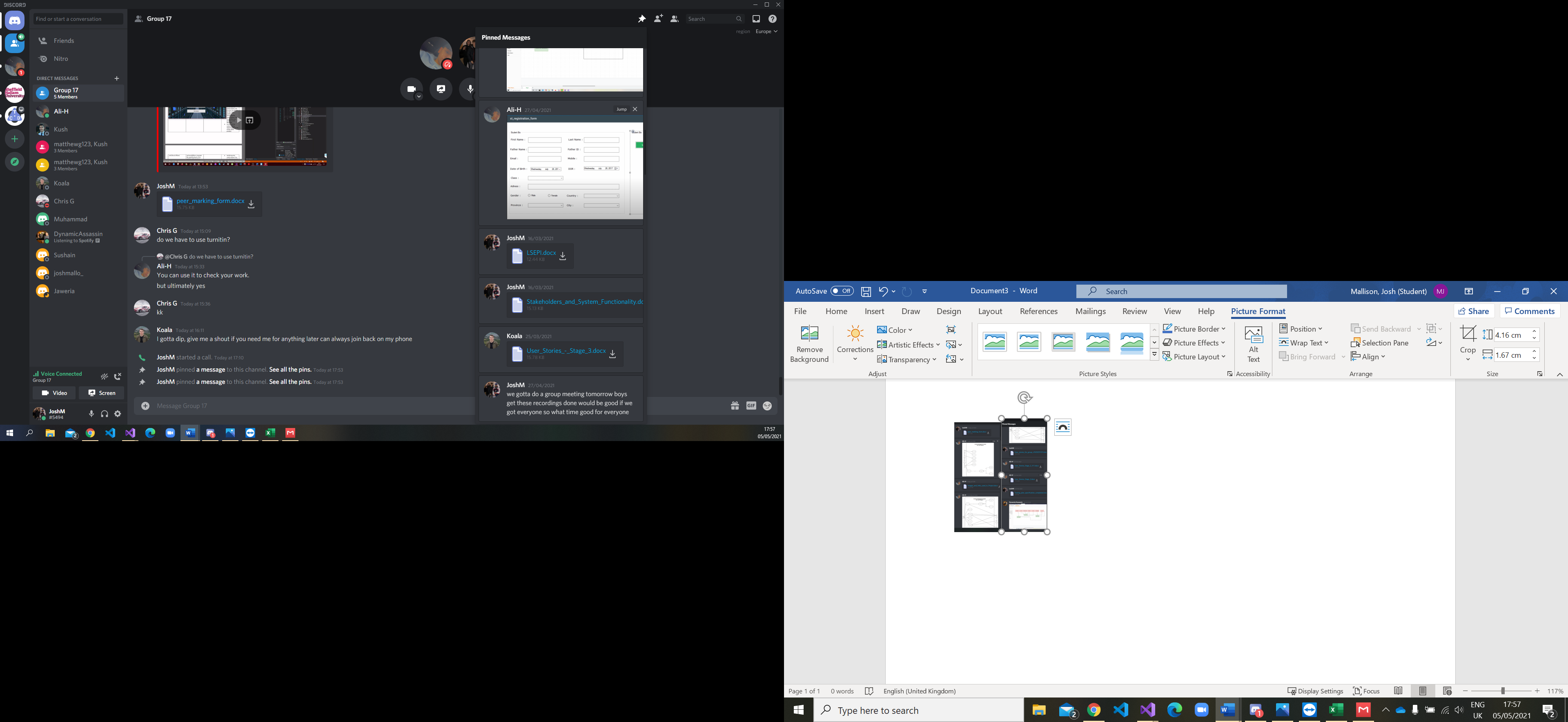
The use case diagram shown above determines four actors shown as stick figures these have been identified as; a supplier, a retail assistant (regular employee), a stock officer and a manager, these all inherit information from the previous actor represented as the bubble arrows between each of the actors. The solid black arrows between the actors and he use case represents a communication between and actor and the system across the system boundary shown. As seen above a supplier would not need to communicate with a case where trend data s accessed or a case where figures are added. This is repeated in the diagram represented as solid black arrows between actors and use cases across a system boundary. The bottom part of the diagram shows extended communication between use cases this is because some cases require further notation to accurately predict the outcome of a certain use case. In this case, to log into a system another case would have to be identified and communicated across to determine whether that outcome will happen, above this is whether the password is correct and is the username correct.

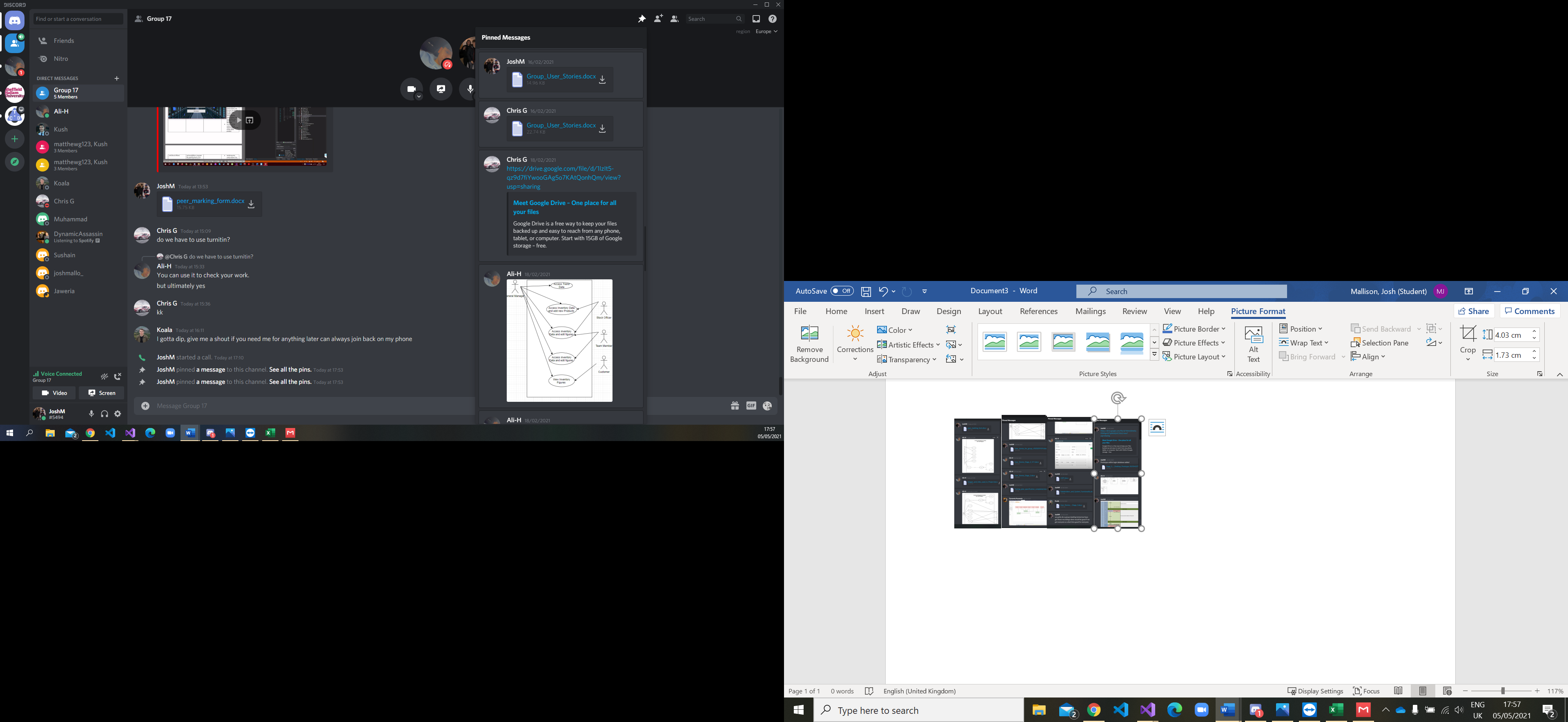
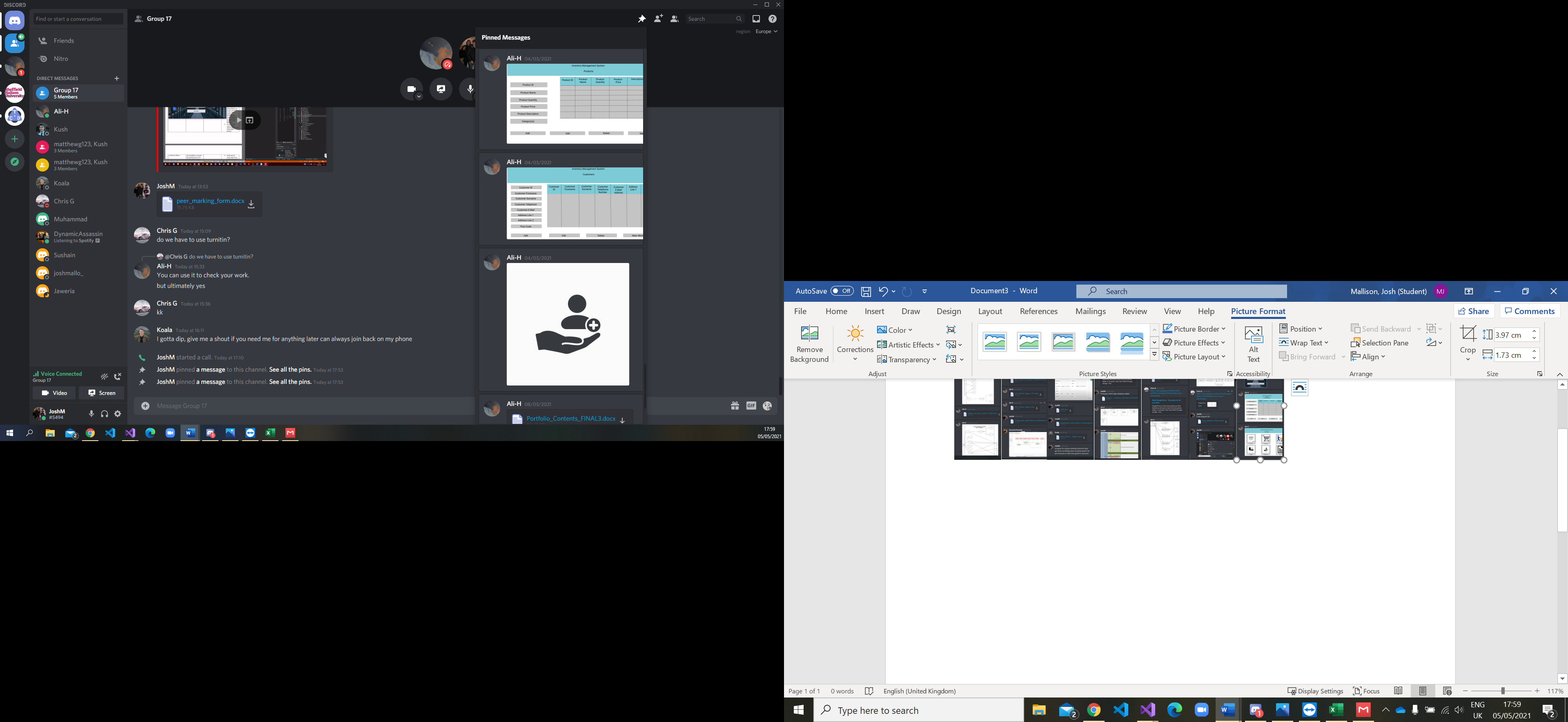
## 2.4 Evidence of Collaborative Work

## 

Note: this is evidence of collaborative work only part of the way through the process as the completed file is on another device due to the local database being stored on another device.

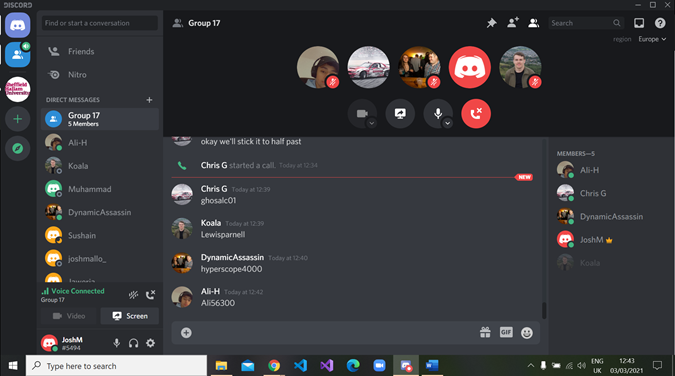
## 





## 

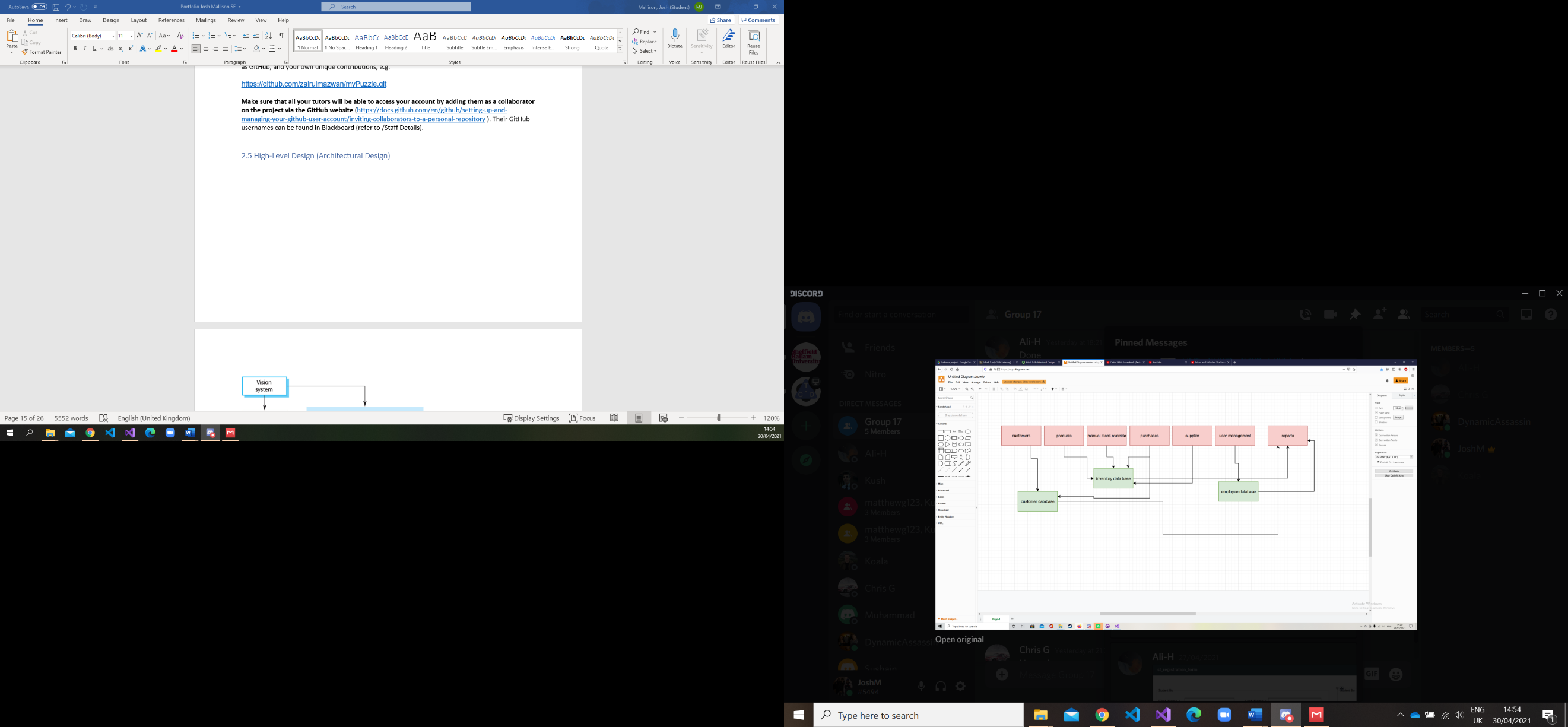
The images above are screenshots of pinned messages in the group discord chat where we have shared design ideas, diagrams, and organised group meetings.



The Screenshot above shows all group members in an active discord call working on designs and setting up Github.

## 2.5 High-Level Design (Architectural Design)

**Architectural Design:**



Our high-level design represented by the architectural design shown above, aims to provide a framework for the development of our Inventory Management System (IMS). This is done through a process of defining collections of software components and their interfaces such that a framework for further evolution is established.

The architectural design is made up of various categories that define the different areas of the system, the relationships between them, and it’s overall function, they can be defined as:

* A set of components that all perform a function required by the system, this can be a database or computational modules for example.
* A form of connection between components that will help in coordination, communication, and cooperation.
* Conditions which present the various components in such a way that they can be integrated to form the system.
* Semantic models that help the designer to understand the overall properties of the system.

The architectural design we have presented exemplifies the aforementioned categories that make up a system. The main parts of the system indicated in the red boxes represent the main system components that make up the inventory management system, they are:

* “Customers”
* “Products”
* “Manual Stock Override”
* “Purchases”
* “Supplier”
* “User Management”
* “Reports”

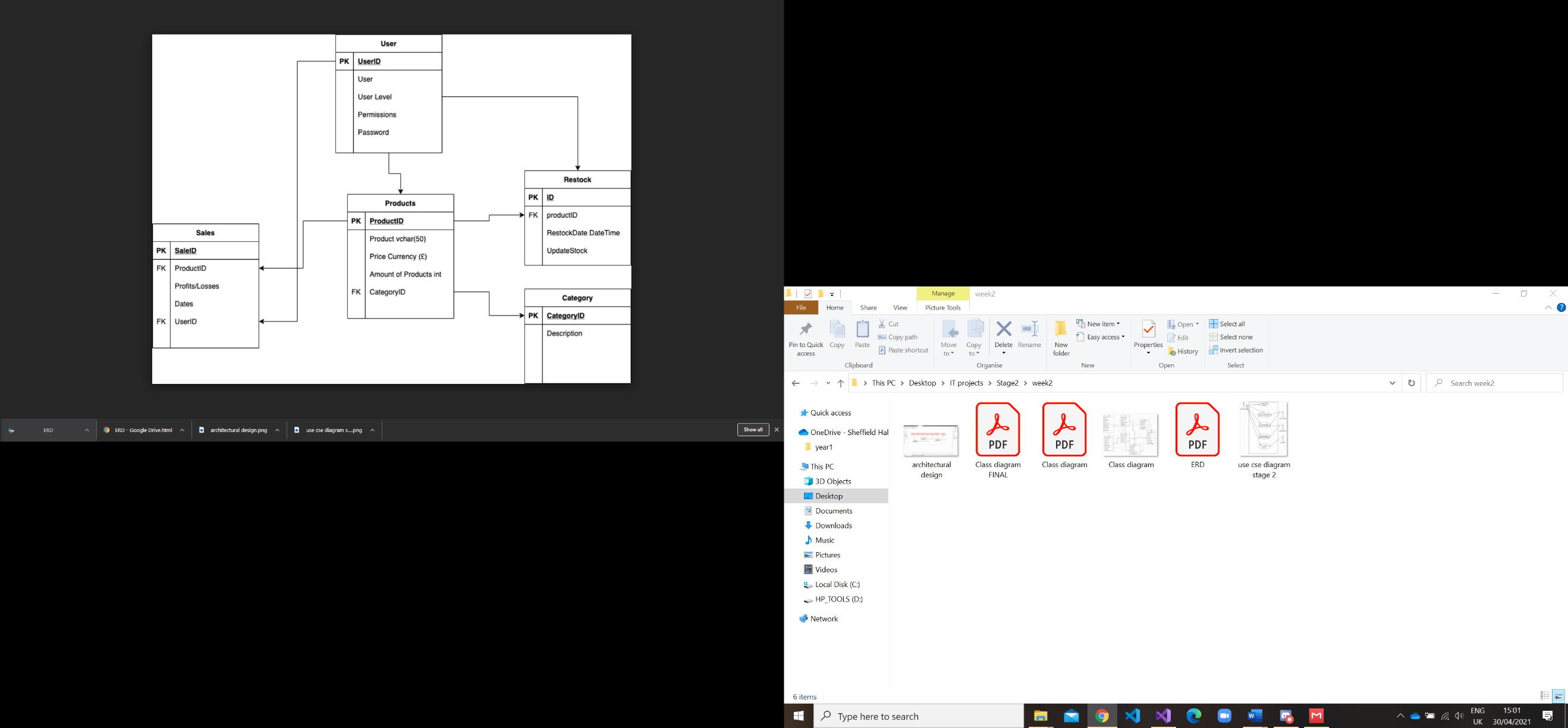
The components shown in the green boxes indicate components in the system represented as databases the other components of the system will be connected to. These three main databases are:

* A customer database
* An inventory database
* An employee database

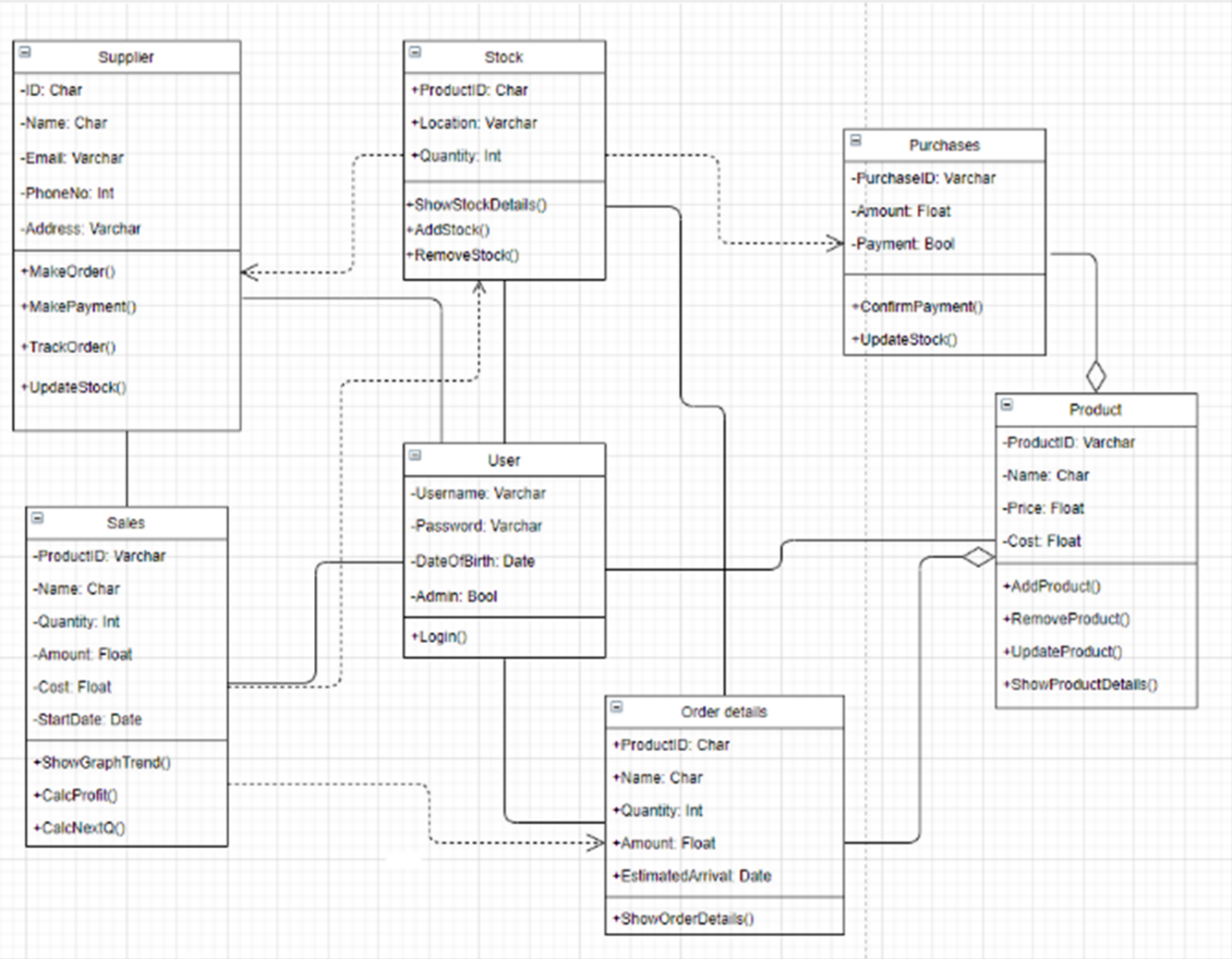
The arrangement of arrows between the various components of the system denotes some form of connection between the two parts. For example, the ”Purchases” component of the database connects to the inventory and customer database, this is because the purchase of an item will influence data relating to customers in that they will have an extra item of purchase matched the account thus purchase connect to and influence the customer database. Purchases also connect to the inventory database as, an item that is purchased will change the stock level in the inventory and thus has an influence over the data stored their as, stock levels for items decrease with each purchase, such that there is a relationship between the two. These kinds of relationships are denoted by the arrows between components across the diagram.

## 2.6 Detailed-Design

**Entity Relationship Diagram:**



The entity relationship diagram (ERD) shown above allows us to see how the different entities in this case – the user, sales, products, categories and restock relate to one another in the application. The entities are denoted as the large rectangles and have different levels in strength from high in the user and products lo lower with categories and re-stock. The relationship are denoted by the arrows or lines connecting the entities ie you need products for restock, in order to have products in inventory you need restock. The attributes are what is held within the entities ie productid is an attribute of sales products and restock.

Class diagram: 

The classes represent a set of objects that have the same structure behaviours and relationships with objects of other classes, thus purchases represent a class as do sales and a user. The attribute of the class shown at the top of the class is the value that defines the properties and behaviour of the object for example a date of birth will be defined as a date and will posses the properties of a date or a price will be represented as a float as money is denoted using decimal places to assign its value. The bottom of the class holds the operations of the class this is a function that can be applied to the objects of the class for example, a function that uses a username and password in the user class would be to login or a function which affect the quantity in stock would be to add or remove stock as shown above. The relationship between the classes is shown as various lines between the classes. The type of relationship the line represents depends on how the data in one class relies or interacts with another for example order details relies on the attributes and operation of the product class on order for order details to define its attributes and perform its operations but not the other way around.

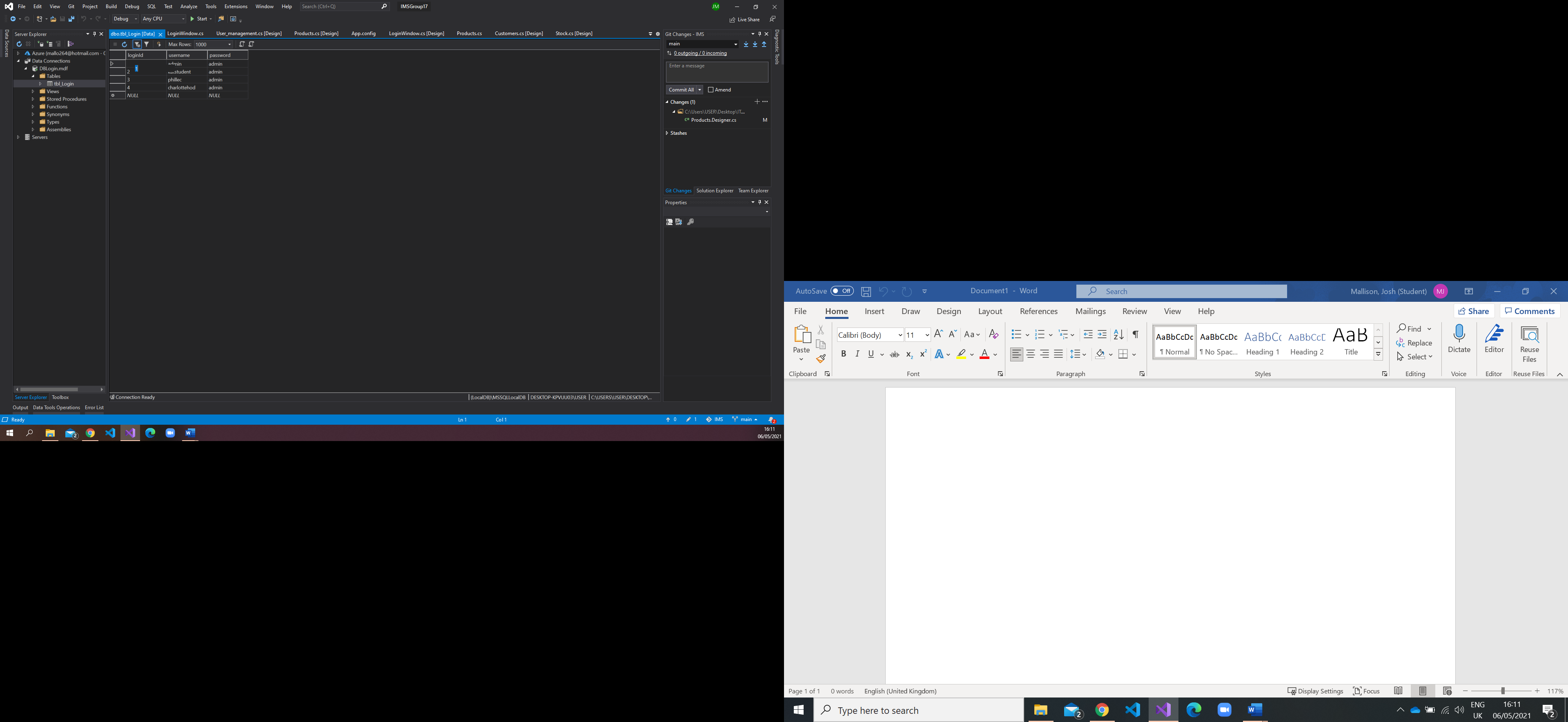
## 2.7 Design Review

<https://youtu.be/sdCydpGpNl4>

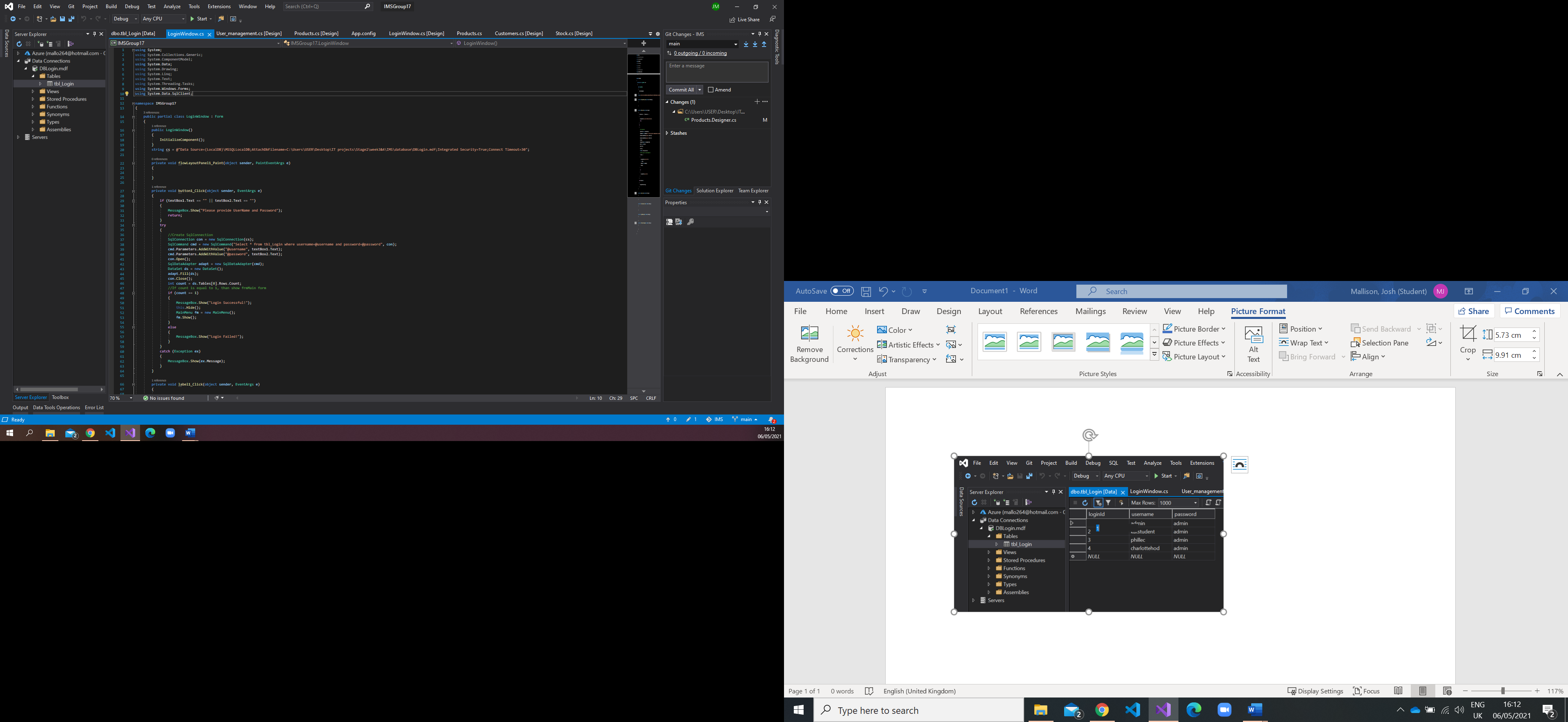
ctrl + left click.

## 

## 2.8 Transitioning a Prototype to Production-Quality Software



The screenshot above shows the login database I produced to store a user’s loginID their username and password. This is an element that was not present within the prototype as the login screen for stage 1 simply showed what the login screen would look like and how it would seem to function rather than actually checking a username or password against a database. This then is evidence to show ow the prototype was built upon towards a more production-quality system.

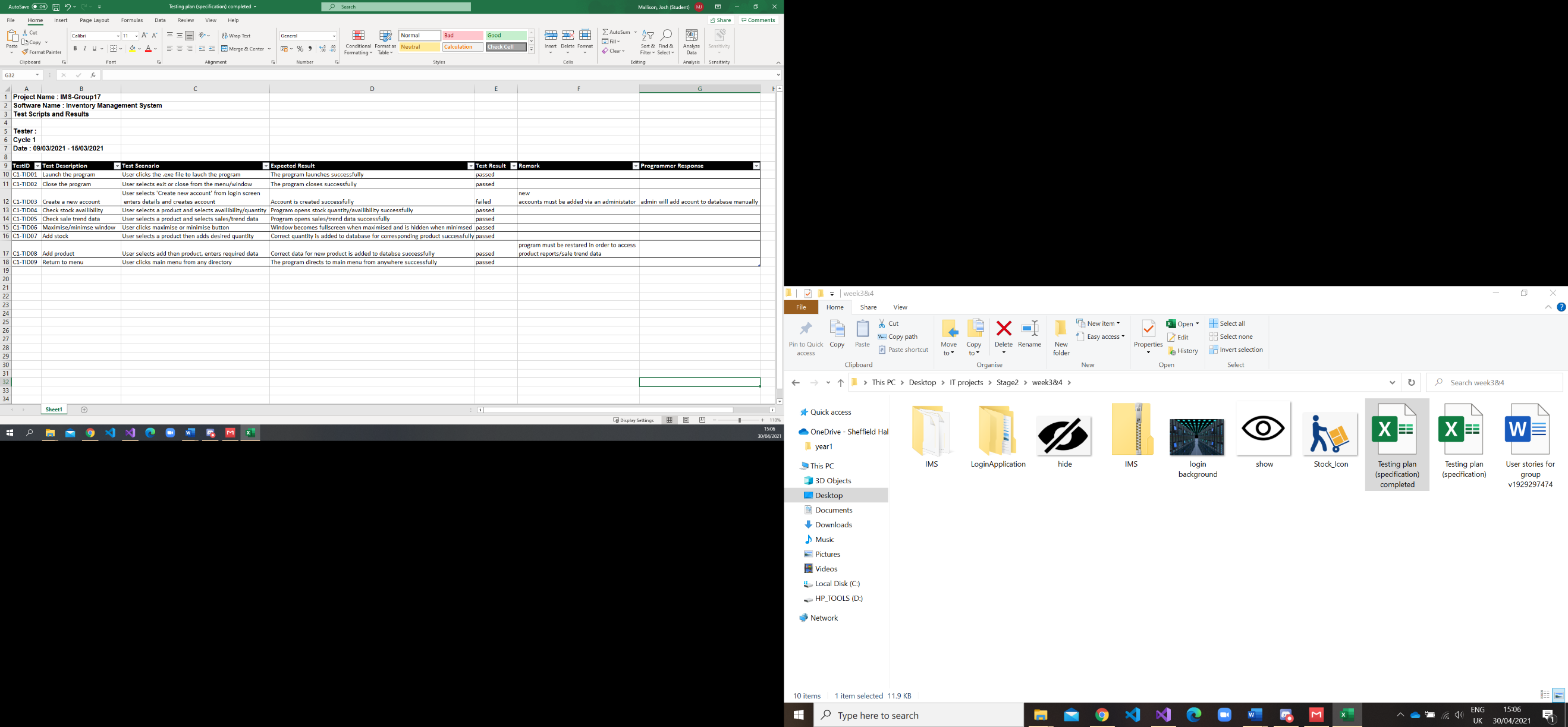


The screenshot above shows how I connected the database to the program and could check a user’s login input against a database.

## 2.9 Test Specification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Test Scenario** | **Expected Result** | **Test Result** | **Remark** | **Programmer Response** |
| C1-TID01 | Launch the program | User clicks the .exe file to launch the program | The program launches successfully |  |  |  |
| C1-TID02 | Close the program | User selects exit or close from the menu/window | The program closes successfully |  |  |  |
| C1-TID03 | Create a new account | User selects 'Create new account' from login screen  enters details and creates account | Account is created successfully |  |  |  |
| C1-TID04 | Check stock availability | User selects a product and selects availability/quantity | Program opens stock quantity/availability successfully |  |  |  |
| C1-TID05 | Check sale trend data | User selects a product and selects sales/trend data | Program opens sales/trend data successfully |  |  |  |
| C1-TID06 | Maximise/minimise window | User clicks maximise or minimise button | Window becomes full screen when maximised and is hidden when minimised |  |  |  |
| C1-TID07 | Add stock | User selects a product then adds desired quantity | Correct quantity is added to database for corresponding product successfully |  |  |  |
| C1-TID08 | Add product | User selects add then product, enters required data | Correct data for new product is added to database successfully |  |  |  |
| C1-TID09 | Return to menu | User clicks main menu from any directory | The program directs to main menu from anywhere successfully |  |  |  |

## 2.10 Unit and Accepting Testing



Evidence of manual unit testing and acceptance testing (based on acceptance tests identified in Sub-section 2.1.3., e.g.

## 2.11 Incorporation of formative feedback

In a zoom meeting during the initial stages of the programs development, Syafiq Zolkeply that the login details for users should be stored on a database rather than hard coded into the software. This was because if a new user wanted to add their account to the system, they would need full back-end access to the code which is not realistic for a production quality system. In response to this I added a login database which stored a user’s ID, username, and password. This means if a new user wishes to add their account in order to gain access to the program, they could do so by adding their relevant details directly to the database or a new account could be created by linking a ‘create new account’ page (which would ask the user for relevant details) to the database, such that a new user would not even have to have direct access to the database itself to add a new account.

# 3. Stage 3

## 3.1 Client Background

The client our team selected for stage 3 was Benedict Connell, a senior assistant registrar for student experience at the British University Egypt (BUE) based in Cairo. With Benedict as a representative of the BUE, the university is accredited with the highest QS ranking as of 2021 and educates over 12,000 students across 47 countries. The curriculum offers flexible modules and 24-hour access to virtual learning facilities whilst providing over 72.45 million in external research funding across Egypt.

The University boasts a range of faculties across its curriculum including:

* Engineering
* Informatics and Computer Science
* Business Administration, Political Science and Economics
* Nursing
* Pharmacy
* Dentistry
* Communications and Mass Marketing
* Law
* Arts and Humanities
* Arts and Design
* Energy and Environmental Engineering

Benedict Felt the current format for student registration was outdated and in need of an upgrade. Our client outlined a number of issues with the current system for registration:

* Registration is paper based and handwritten - This can often be unreliable if writing is ineligible and high costs due to paper/printing.
* Registers are time consuming – Benedict outlined that registration often involved writing names on a piece of paper that is passed around this distraction for students takes up valuable lesson time again which is costly for the university as a whole.
* Lack of uniform application – different classes have different forms of registration from calling out names to writing names on paper this makes it very difficult to synthesise and track attendance data thus leading to more time wasted and higher costs.
* The registration system is subject to ‘Ghosting’ – because of a lack of precautions in place to protect against attendance fraud students are able to falsely log their attendance when they are leaving class early/joining late or using other students to falsely log their attendance. Thus, the university may miss students who are not performing and need help, hindering student welfare and the overall operation of the university.
* Attendance is difficult to monitor and report on – because of how the registration is conducted and obtained, the whole aim of tracking and reporting on attendance is thus ineffective again negatively impacting students, in being unable to identify and report on those students who need support.
* The current attendance is because of the aforementioned reasons widely dislike by students and staff, respectively.

In response to the shortcomings of the current system, our client requested that the purpose of the attendance prototype would be to capture student attendance systematically and accurately at scheduled teaching activities (teaching activities can take place online or in person).

Benedict then outlined his ideal software requirements for the prototype:

* Needs to be mobile phone compatible.
* Needs to map against students specific ID numbers.
* Needs to guard against misuse or ‘ghosting’.
* Needs to produce reports, or at least import/export, data.
* Needs a system of notification.
* Needs a time stamping function.

## 3.2 Software Project Artefacts

### 3.2.1 Users

|  |  |
| --- | --- |
| **Name** | **Description** |
| Lecturer | Is in charge of the classes that students will be attending, the lecturer will be in part responsible for checking their students attendance and acting upon reported figures. |
| Student | Will be actors within the system who log their attendance (upon arrival and departure) using the QR codes provided, they will also have access to their own attendance record so they can see what is being reported to lecturers. |
| Head of Department | They will have access to all front and back-end parts of the system, they will be responsible for tracking overall attendance across all of the classes, making sure lecturers are correctly reporting and offering support to individuals. The head of department will also want to monitor systemic issues of bad attendance across certain classes/subjects. |

### 3.2.2 User Stories and Acceptance Tests

|  |  |  |
| --- | --- | --- |
| **User Story** | **Notes** | **Acceptance Test(s)** |
| Kat (Student) Scans her classes QR code upon arrival to her lesson and logs her attendance. | If kat arrives more than 15 minutes late her attendance may be flagged as late | * Verify that the QR code provided successfully opens the login portal for kat to enter her username and password. * Verify that the system allows Kat to login to the portal with her details to log her attendance to that class. * Verify that the system notifies Kat that her attendance has been successfully logged. * Verify that the system also shows Kat the time stamp assigned to her login which will subsequently updated to the database. |
| Kat (Student) Logs in on the desktop application to access the main menu in order to navigate the various windows relating to her registration. | As kat is a student she will have restrictions on some areas of the desktop application when compared to university staff. | * Verify that the system opens a log in portal when the application is launched for Kat to enter her details. * Verify that the system notifies her if her log-in details are entered incorrectly and allow her to attempt to re-enter her username and password. * Verify that the system notifies her when her login is successful. * Verify that the system allows her to access the various windows relating to her student registration. * Verify that the system allows her to log-out when she wants to exit the program |
| Kat (Student) Logs into the desktop application in order to fill out her registration form so she can be added to the school’s database. |  | * Verify that system allows her to navigate to the registration button. * Verify that the system then successfully opens the registration form window. * Verify that the system allows her to enter the following details: Forename, Surname, Date of Birth, Email Address, Phone number, address, post code, gender, and her chosen class. * Verify that the system allows her to save her details by pressing save. * Verify that the system notifies her to click update once she has saved in order to update the school’s database. * Verify that the system notifies her that her details have been successfully updated to the database. |
| Phil (Lecturer) Logs into the desktop application with higher permissions on the lecturer account. | Lecturers may not have access to all their students data from other classes | * Verify that the system opens a log in portal when the application is launched for Phil to enter his details. * Verify that the system notifies him if his log-in details are entered incorrectly and allow him to attempt to re-enter her username and password. * Verify that the system notifies him when his login is successful. * Verify that the system gives him access to higher levels of permissions giving him the power to add and remove students from classes as well as checking student attendance data. |
| Phil (Lecturer) Logs into his account and navigates to the student link in order to add a new student to one of his classes. |  | * Verify that the system allows him to navigate to the lecturer button and the subsequent window opens successfully. * Verify that the system allows him to enter a student ID, student name, and class the student is attending. * Verify that the system adds the aforementioned details to the table. * Verify that the system notifies Phil when the student’s data has been added to his class. * Verify that the system does not allow Phil to add students to a class without a student ID, student name or class. |
| Phil (Lecturer) Accesses his lecturer account in order to remove a student from one of his classes as they have dropped out. |  | * Verify that the system allows him to navigate to the lecturer button and the subsequent window opens successfully. * Verify that the system allows Phil to navigate between the various students shown in the table. * Verify that when Phil selects a student from the table they are highlighted in blue. * Verify that the system allows Phil to select remove when one of the student’s details in the table is highlighted. * Verify that the system removes the highlighted person’s details when remove is selected. * Verify that the system does not remove anything from the student table when no-ones details are highlighted. * Verify that the system notifies Phil when a student has been successfully removed from a class. |
| Charlotte (Head of Department) Logs into the desktop application with all permissions for the front and back-end system | The Head of department’s access to some student data may be limited only in cases where it has been requested by a student for external reasons. | * Verify that the system opens a log in portal when the application is launched for Charlotte to enter her details. * Verify that the system notifies her if her log-in details are entered incorrectly and allow her to attempt to re-enter her username and password. * Verify that the system notifies her when her login is successful. * Verify that the system has given her access to all the windows with all powers to removes students from classes and access to all data on students across the department. |
| Charlotte (Head of Department) logs into the desktop application to check the attendance record and reports of students across the department. |  | * Verify that the system allows her to navigate to the reports button to access the required data. * Verify that the system opens the reports window successfully. * Verify that the system displays the following information about each student: Student ID, Student Name, Class, Present, Excused, and Time Stamp. * Verify that the system allows her to select the reports of students from a certain class. * Verify that the system allows her to select a date on which to view attendance/reports on students. |
| Charlotte (Head of Department) logs out of her account to allow another user to access their login (protecting others from accessing the higher permissions on that device) |  | * Verify that the system allows her to navigate back to main menu from any given page on the system. * Verify that the system allows her to select logout from the main menu. * Verify that the system notifies charlotte when she has logged out. * Verify that the system closes the main menu when log out is pressed and that the login portal is re-opened. |

## 3.3 Software and Its Presentation

### 3.3.1 The Software Prototype



The file above is the desktop application prototype for stage 3

The web application and another copy of the desktop application has been uploaded the submission point on blackboard.

### 3.3.2 Video Presentation

<https://youtu.be/3szISOsw88E>

## 3.4 Evidence of Collaborative Work

See 2.4 where evidence for all collaborative work for stage2 & 3 is shown.

## 3.5 Incorporation of Formative Feedback

**Formative Feedback Document**

**Group Number: 17**

**Date: 19 April 2021**

***Prototype System***

**Yes No** (Tick an appropriate box)

1.  **Does the prototype demonstrate user stories (or functions) agreed by you?**

2.   **Does the prototype demonstrate any of the acceptance tests the group agreed with you?**

3.   **Does the prototype demonstrate any assumptions which the group have made about any aspects the project you proposed, such as interpretation of requirements?**

4.   **Is the prototype fit for purpose?** **In other words, can it be regarded as an effective medium through which your requirements can be negotiated?**

5. **Briefly state what the student should do to improve the prototype:**

**Improvement**: The presentation didn’t show much beyond the log in pages, so improving the end user experience and visuals so that the prototype has a much more finished feel to it would be good. Ensuring that the functionality requested gets incorporated and becomes a working reality.

**Client Comments**: Clearly, there were good ideas about how some of the issues posed might be addressed but the prototype seemed to be in the development stage still and so it was hard to get a good sense of what the final functionality might look like from an end user / client perspective.

The program presented to the client was the unfinished web application, most of the system functionality was also based on the desktop application which was not available for showing at the time. Since receiving the client’s feedback, the website has been improved and brought in line with the clients functionality requests. I believe the completed prototype we have submitted is indeed now fit for purpose and would be an effective medium through which requirements can be negotiated. It addresses the issue of ghosting, the database holds time stamps for students, produces reports, and has a unique ID number for students, this is all inline with what the client originally requested.

# 4. Evaluative Report on Legal, Social, Ethical and Professional Issues (up to 1000 words)

## 4.1 Stage 2

## 4.1.1 Relevant Issues

The protection of data will be a huge legal, social, ethical, and professional issue when it comes to our inventory management system as well as all other such systems. In the case of our system, if confidential data about a company’s inventory were not protected or was leaked to the public and thus to a company’s competition, it would be a huge issue professionally. A lack of encryption or a lack of varying permission levels depending on who is using the system, on the part of the developers, would be viewed as poor professionalism in the execution and testing of the program. In such a case this will likely lead to huge loss for the client’s company. Furthermore, some may argue that it violates the Data Protection Act 2018 1 (GOV.UK, 2018)where it suggests that data should be ‘handled in a way that ensures appropriate security, including protection against unlawful or unauthorised processing, access, loss, destruction or damage’. This could be applied to a breach of data if the actions leading to it were deemed irresponsible.

Another LSEP issue relevant to the inventory management system is a company using the program to produce fraudulent data for illicit means. This could be a company or individual publishing inventory documents using the program as a reference to over or underrepresent their true inventory for various means which challenge the LSEP issues. This is also known as inventory fraud.

## 4.1.2 Discussion

In a recent article by the ico (Information Commisioner's office, 2018) a former recruitment consultant was fined for unlawfully taking personal data from his employer when he left his job to set up his own rival company. In the case of our system if a former employee of a client were able to have higher access to inventory data and was able to track sales, they would have unlawful access to a rival company’s data if an employee high up enough did so. This would indeed violate the Data Protection act 1998 as did the man was fined in the referenced article.

Inventory fraud would be a legal and ethical issue concerned with the inventory management system, they often consist of two main types; employee theft and Financial fraud as cited by arxis (Hamilton, n.d.) where they outline that employees who may have access to physical inventory and in our case the IMS they would be able to remove stock for themselves and adjust the inventory levels on the program so when the inventory is checked it matches the amount displayed by the application. Financial fraud also plays it part, arxis says “Inventory overstatement is the most common type of inventory related fraud. Management may be motivated to report high earning s to either satisfy stockholders, achieve compensation targets, or maintain bank lending covenants”. In this case if management of our client’s company were to commit financial fraud it is likely they would use the IMS to sabotage documents and produce fraudulent inventory figures. For obvious reasons this presents may present legal issues relating to our inventory management system.

## 4.2 Stage 3

## 4.2.1 Relevant Issues

One major LSEP issue that relates to our attendance system, if it failed to report to staff when a student was not attending, is the welfare of students in education but also in their wider physical and mental wellbeing. If the program were poorly executed and resulted in many students ghosting or slipping through the system’s net, it may present social issues for a client in not being able to effectively track their students and help them to the best of their ability. If a student were really struggling and because of the poorly made attendance system the student had become mentally/physically ill and was not flagged, it would present a legal issue if the university failed to step in or offer help to that individual in any case where someone was hurt. The failure of the program to perform its function also presents a professional issue on the part of the developers which then leads on into the aforementioned issues.

Another problem our attendance system may face regarding LSEP issues would be responsibly using the data produced by the system. The abuse of this data by a client could lead to social, ethical, and legal issues. This can relate to a range of issues when dealing with student’s personal data. One is ensuring that only staff who have the necessary skills and qualifications are allowed to deal with a student’s personal issues if that relate to a lack of attendance, it may distress a student if a lecturer were to know about these personal issues. This can also apply to other students not having access to this kind of data, furthermore a lack of protection from data leaks or hacks poses further legal issues.

## 4.2.2 Discussion

The huge professional issue in producing an inadequate attendance system for a client, is the severe effect a lack of student attendance has on academic performance in university and education in general. This is backed by a meta-analysis study (Guleker, 2014) which states “if a student does not attend at least 70 % of the classes he/she has a probability of two in three to receive a failing mark and a probability of 4 in 5 to have low grades. The 80% rule stated that if a student does not attend at least 80 % of the courses, the probability to get failing grades is 50 %, whereas the probability not to get high grades is 2 in 3.” This then supports the agreement that a failure to produce an effective program for our client could in turn lead to many students performing below their optimal performance. This would be as a result of the system not finding out and reporting on students who are not attending which means university staff will not be able identify those in need such that students continue to not attend, thus academic performances drop respectively. This then points to a social issue in lack of welfare for students as well as a professional one on the part of the developers.

A failure by the client and the development team to produce a secure system that protects student data from within and externally can pose serious LSEP issues as outlined by a recent case of students data being breached (Merkel, 2020). In this case files from Newcastle university were leaked by a cybercrime group they included student and staff information such as home addresses, phone numbers and personal email addresses. If this were to happen to our client because of a poorly designed or encrypted system, the ramifications would be huge legally and professionally but also socially and ethically when concerning the students who have had their data abused or leaked.

# 5. References

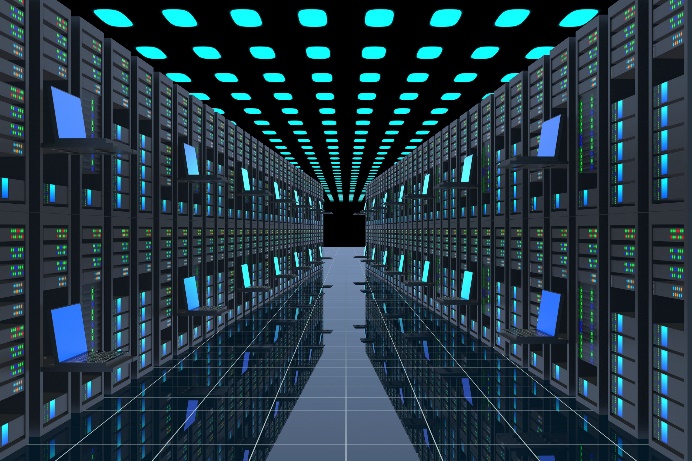
GOV.UK. (2018). *Data Protection*. Retrieved from GOV.UK: https://www.gov.uk/data-protection

Guleker, R. e. (2014, November). The Effect of Attendance on Academic Performance . *Mediterranean Journal of Social Sciences*, 961-966.

Hamilton, C. (n.d.). *Inventory Fraud*. Retrieved from ARXIS FINANCIAL: https://www.arxisfinancial.com/images/pdfs/Inventory\_Fraud.pdf

Information Commisioner's office. (2018). *News and events*. Retrieved from ICO: https://ico.org.uk/about-the-ico/news-and-events/news-and-blogs/2018/05/former-recruitment-consultant-prosecuted-for-stealing-personal-data-from-his-old-employer/

Merkel, T. (2020, December). *Newcastle students’ data including home addresses leaked on dark web after cyber attack*. Retrieved from thetab: https://thetab.com/uk/newcastle/2020/11/02/newcastle-students-data-including-home-addresses-leaked-on-dark-web-after-cyber-attack-52213?itm\_source=parsely-api

reference for Images used in stage 2 and 3

(Login Screen)

<https://www.wired.com/story/its-time-to-think-beyond-cloud-computing/>



(Menu Button)

<https://www.freepik.com/free-icon/online-purchase_815280.htm>



(Menu Button)

<https://www.kindpng.com/imgv/oomxxT_gestion-stock-icon-png-transparent-png/>



<https://www.shutterstock.com/image-vector/forklift-delivery-truck-vector-icons-279338228>



(Menu Button)

<https://depositphotos.com/vector-images/customerretention.html>



(Menu Button)

<https://www.iconsdb.com/black-icons/report-3-icon.html>

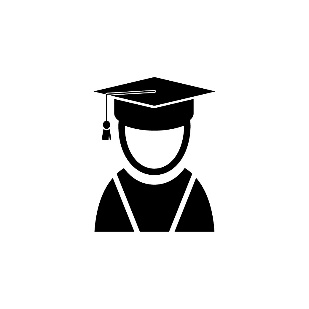
(Main Menu Image)

<https://dailynewsegypt.com/2013/03/02/british-university-students-occupy-auditorium/>



(Login Page Image)

<https://www.linkedin.com/company/thebritishuniversityinegyptbue>

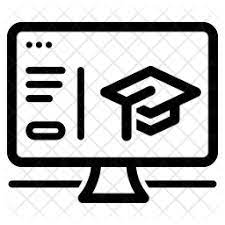


(Menu Button)

<https://thenounproject.com/term/class-discussion/1427532/>

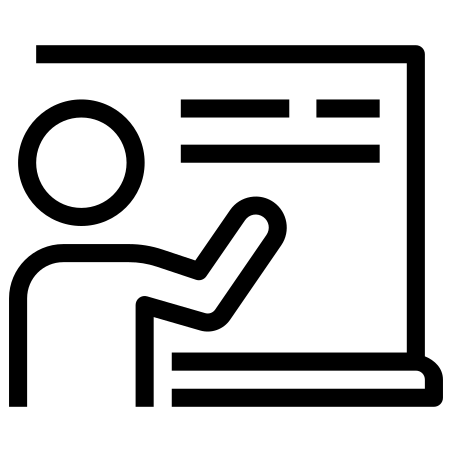
(Menu Button)

<https://www.mcicon.com/product/student-icon-8/>



(Menu Button)

<https://iconscout.com/icon/online-course-8>



(Menu Button)

<https://icons8.com/icon/38HJBFwphJ3I/teacher>



(Menu Button)

<https://www.pinterest.co.uk/pin/595178906990484744/>



<https://brandeps.com/icon/L/Log-out-outline-01>

# Appendices

## Appendix 1: Software Projects- Peer Marking Form (Stage 2)

|  |  |  |
| --- | --- | --- |
| 1 | C G – Program code for IMS, ERD, user stories, video recording. | 10/ 10 |
| 2 | A H – Design/wireframe, user stories, use case diagram, video recording, design review video recording. | 10/ 10 |
| 3 | J M – Majority program code for IMS, class diagram, user stories, video recording, test plan specification, design review video recording. | 10/ 10 |
| 4 | L P – Program code, setup classes, designed windows forms. | 7/ 10 |
| 5 | D R – Architectural design, program code, error fixes, design review video recording. | 8 / 10 |
|  |  |  |



|  |
| --- |
| Add any comments you feel would be useful for the tutor to know about when assessing marks |
| L P : Wasn’t able to contribute as much because of no access to a PC due to personal circumstances. |

## Appendix 2: Software Projects- Peer Marking Form (Stage 3)

|  |  |  |
| --- | --- | --- |
| 1 | C G – Web application, Database, Video recording | 10/ 10 |
| 2 | A H – Windows form design desktop app, some program code, video recording | 10/ 10 |
| 3 | J M – Program code for desktop app, user stories, video recording, system functionality | 10/ 10 |
| 4 | L P – User stories, program code for desktop app | 8.5/ 10 |
| 5 | D R – program code foe desktop app | 6.5 / 10 |

|  |
| --- |
| Add any comments you feel would be useful for the tutor to know about when assessing marks |
| L P : Wasn’t able to contribute as much because of no access to a PC due to personal circumstances. |