

Department of Computing

Software Projects (55-407815-AF-20201)

**Name: Reece Wareham**

**Student ID: c0024660**

**Degree Route: BSc CS**

**Table of Contents**

1. [Stage 1 4](#_bookmark0)
   1. [Software Project Artefacts 4](#_bookmark1)
      1. [Project Description and Users 4](#_bookmark2)
      2. [User Stories and Acceptance Tests 5](#_bookmark3)
   2. [Software and its Presentation 6](#_bookmark5)
      1. [The Software Prototype 6](#_bookmark6)
      2. [Video Presentation 6](#_bookmark7)
   3. [Incorporation of Formative Feedback 6](#_bookmark8)
2. [Stage 2 7](#_bookmark9)
   1. [Software Project Artefacts 7](#_bookmark10)
      1. [Users 7](#_bookmark11)
      2. [User Stories and Acceptance Tests 9](#_bookmark12)
   2. [Software and its Presentation 11](#_bookmark14)
      1. [The Production-Quality Software 11](#_bookmark15)
      2. [Video Presentation 11](#_bookmark16)
   3. [Use Case Diagram 11](#_bookmark17)
   4. [Evidence of Collaborative Work 12](#_bookmark18)
   5. [High-Level Design (Architectural Design) 13](#_bookmark19)
   6. [Detailed-Design 14](#_bookmark20)
   7. [Design Review 15](#_bookmark21)
   8. [Transitioning a Prototype to Production-Quality Software 16](#_bookmark22)
   9. [Test Specification 17](#_bookmark23)
   10. [Unit and Accepting Testing 18](#_bookmark24)
   11. [Incorporation of formative feedback 19](#_bookmark25)
   12. Peer Assessment Form (Stage 2) ………………………………………………………………………………………….19
3. [Project 3 20](#_bookmark26)
   1. [Client Background 20](#_bookmark27)
   2. [Software Project Artefacts 20](#_bookmark28)
      1. [Users 20](#_bookmark29)
      2. [User Stories and Acceptance Tests 21](#_bookmark30)
   3. [Software and Its Presentation 22](#_bookmark32)
      1. [The Software Prototype 22](#_bookmark33)
      2. [Video Presentation 22](#_bookmark34)
   4. [Evidence of Collaborative Work 22](#_bookmark35)
   5. [Incorporation of Formative Feedback 23](#_bookmark36)
   6. [Peer Assessment Form (Stage 3) 23](#_bookmark37)
4. [Evaluative Report on Legal, Social, Ethical and Professional Issued (1000 words) 24](#_bookmark38)
   1. [Stage 2 24](#_bookmark39)
      1. [Relevant Issues 24](#_bookmark40)
      2. [Discussion 24](#_bookmark41)
   2. [Stage 3 25](#_bookmark42)
      1. [Relevant Issues 25](#_bookmark43)
      2. [Discussion 25](#_bookmark44)
5. [References 26](#_bookmark45)

[Appendices 27](#_bookmark46)

[Appendix 1: Software Projects- Peer Marking Form (Stage 2) 27](#_bookmark47)

[Appendix 2: Software Projects- Peer Marking Form (Stage 3) 28](#_bookmark48)

# Stage 1

## Software Project Artefacts

* + 1. Project Description and Users

The project I am working on is an inventory management system. This kind of software is generally used within retail stores and allows for easier organisation.

**Personas**

|  |  |  |
| --- | --- | --- |
| Name | Role | Persona |
| James | Inventory System Manager | James is 32 and has been working in this industry for 15 years. He oversees the entire system and configures it whenever it needs changing such as adding new products, removing them or editing them. |
| Mary | Customer | Mary is 49 and is a mother of 3. She isn’t very tech savvy and struggles to use and understand computers. Mary wants to buy products for her family. |
| Dave | Stock Manager | Dave is 29 and his role as a stock manager means that he must make sure that all products are in stock if possible. This means that he is always in contact with external companies, ordering new stock. |

* + 1. User Stories and Acceptance Tests

|  |  |  |
| --- | --- | --- |
| User Story | Notes | Acceptance Tests |
| James can look up all the available stock in the system. |  | Verify that the system allows the system manager to view all the available stock in the database. |
| James can remove products from the system. |  | Verify that the system allows for the system manager to remove specific products from the system. |
| James can edit the details of a product. |  | Verify that the system allows for the system manager to edit the details of a product and save it. |
| James can add new products to the system. |  | Verify that the system allows for the system manager to add new products to the system database. |
| James can search for specific products in the system. |  | Verify that the system allows for searching up a specific product and displaying its details. |
| Mary can look up all the available products. |  | Verify that the system allows for customers to look up all the available products. |
| Mary can purchase products from the system. |  | Verify that the system allows the customer to purchase products. |
| Mary can view the orders she has placed. |  | Verify that the system shows the customer all of their orders that they have placed. |
| Mary can view all of her details in the systems. |  | Verify that the system allows the customer to see all of their details such as their name, age, address, contact details. |
| Dave can view all the products that are out of stock or low on stock. |  | Verify that the system allows the stock manager to see all the out of stock products/ low on stock products. |
| Dave can view all the products that are in stock. |  | Verify that the system allows the user to view the products that are in stock in a list. |
| Dave can create stock orders. |  | Verify that the system allows the user to create buy orders for stock. |

## Software and its Presentation

* + 1. The Software Prototype

Prototype has been uploaded at the submission point.

* + 1. Video Presentation

[Link to YouTube Video](https://youtu.be/4O21MgBSFYk)

## Incorporation of Formative Feedback

During the development of my prototype in stage 1, I was given lots of feedback from Nnamdi about my prototype. Sometimes on Tuesday and Thursday I would have a short meeting with Nnamdi to discuss the prototype and what changes could be made to make it better. During this he suggested ways of making each page of the prototype interact with each other which was useful as I was still new to Adobe XD and Protopie so my knowledge on these mechanics were limited. Because of this feedback, it allowed me to add functions into my prototype that made it look more professional.

# Stage 2

## Software Project Artefacts

* + 1. Users

The project I chose to create was the inventory management system. The program that is created from this project would likely be used in many different retail businesses as it allows employees to easily see the amount of stock they have, the types of products they sell and if there are any complaints from customers who may not have had a good experience. The users that would likely use this program are as follows:

|  |  |  |
| --- | --- | --- |
| Name | Role | Persona |
| Jack | Management Department Employer | Jack is a 25 year old Computing Graduate from the University of Sheffield. He likes communicating with all his staff and is very good at time management. |
| Susan | Purchasing Department Employee | Susan is a 37 year old finance graduate. Susan is very good with numbers and spreadsheets. |
| Mehmet | Production Department Employee | Mehmet is a 31 year old university graduate. He is very hands on and likes to stay organised. |
| Kieran | Sales Department Employee | Daniel is a 20 year old employee who has only recently joined the business. He is confident and eager to help. |
| Ethan | Software Department Employee | Ethan is a 28 year old software engineer who has a master’s degree in Software Engineering. |
| Daniel | Customer | Kieran is a 22 year old factory worker who likes shopping. |

**Roles**

**Management Department Employer –** Themanagement department dictates the timeframe for which it wants products to be ready for sale, the quantity and quality of products and management of all other departments. They need full access to all information regarding product location, cost of goods, quantity of goods, the dates when they reach each step in the process, serial numbers and lot numbers. They also need constant communication with the software team so that any new features or bugs stopping the company from functioning can be implemented quickly.

**Purchasing Department Employee** – Part of the first stage in the Inventory Management process, the purchasing department needs to order goods (raw material) in order to manufacture products. A purchasing employee needs access to the purchasing database and item location after the purchase has been completed in order to pass on information to the production department.

**Production Department Employee** – Oversee the manufacturing process where raw materials are processed into products. The production department needs clear, precise communication with the purchasing department so they can source the correct materials for the required products set by the management department. It is also important that the production department knows when materials arrive and leave the factory so they can pass on how much stock is available to the sales department.

**Sales Department Employee** – The sales department has direct contact with the customer and manages the final stage in the inventory management system. They need to use the application to communicate with customers, handle any inquiries, track the location and state of the products and take customer payment and details.

**Software Department Employee** – Part of the maintenance and development team. Can view/ manage all sales, product/material locations and purchases. They are in close contact with all other departments, especially management as they are needed in order to fix any software instabilities/bugs and train/help all employees learn how to use the software.

**Customer** – Need a clear user interface where they can see all available products, search by category or item and purchase. If products are not currently available, they can see an estimated restock date or they can communicate with the sales department, where they can pass on any feedback for the inventory management system.

* + 1. User Stories and Acceptance Tests

|  |  |  |
| --- | --- | --- |
| User Stories | Notes | Acceptance Tests |
| Jack can delete, edit, add and view the order database. |  | Verify that Jack can delete, edit, add and view the order database. |
| Jack can add a new warehouse to the warehouse database. |  | Verify that Jack is able to add a new warehouse. |
| Jack can delete, edit, add and view products. |  | Verify that Jack can view, edit, add and delete the product database. |
| Susan can view product information. Such as price, product ID and volume. |  | Verify that the system allows Susan to search for specific products, which displays the information about the selected product. |
| Susan can view warehouse information. Such as location and inventory. |  | Verify that Susan is able to select the warehouse she requires and the information. |
| Susan can directly contact the product department, regarding specific products and other inquiries. | Include product ID or order number so the correct information can be given. | Verify that the selected information is sent to the Production Department.  Verify that the information has been received by the Production Department. |
| Mehmet can view the purchases database. Such as order number and comments. |  | Verify that Mehmet is able to view the purchases database. |
| Mehmet can edit a warehouse. Such as changing the available space and used inventory space. |  | Verify that Mehmet is able to edit the warehouse database. |
| Mehmet can check the information on every product |  | Verify that Mehmet is able to search for a product and the information about the selected product is displayed. |
| Daniel can search for pre-order items. Using the preorder database. |  | Verify that the system can allow Daniel to view the pre-order database. |
| Daniel can search for available items. Using the product database. |  | Verify that the system can allow Daniel to view the available items database. |
| Daniel can create or login to a customer account. |  | Verify that the system can create an account or login a customer. |
| Ethan can view all the databases. |  | Verify that Ethan can view the warehouse, product, order and user databases. |
| Ethan can edit delete and add to all the databases. |  | Verify that Ethan can delete, edit and add to the warehouse, product, order and user databases. |
| Ethan can login to anyone’s account to test software issues. |  | Verify that Ethan can login to other employees’ accounts. |
| Kieran can view the product database for information, such as product ID and volume. |  | Verify that the system allows Kieran to view the product database. |
| Kieran can view the purchases database for information. Such as order number. |  | Verify that the system allows Kieran to view the purchases database. |
| Kieran can view inquiries from customers and reply. |  | Verify that the system allows Kieran to view customer inquiries.  Verify that the system allows Kieran to send a reply.  Verify that the customer receives a reply. |
| Any user can log out and be taken back to the login or create an account page. |  | Verify that the system allows all users to log out. |

## Software and its Presentation

* + 1. The Production-Quality Software

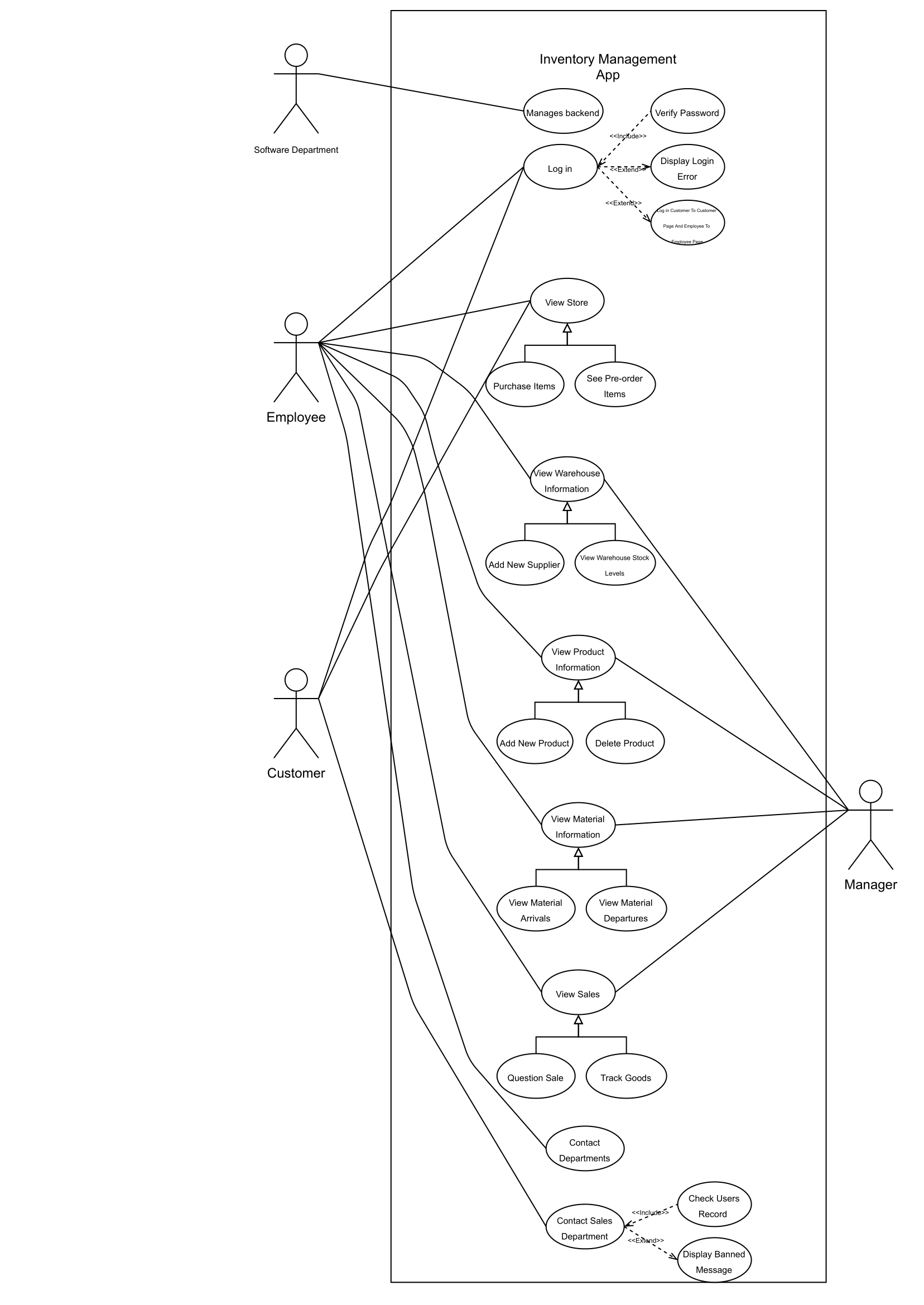
Uploaded at the submission point.

https://github.com/SheffieldHallamSoftwareProjectsGroup50/Stage2Project

* + 1. Video Presentation

[Link to YouTube Video](https://www.youtube.com/watch?v=bGIpmxkJ_gU)

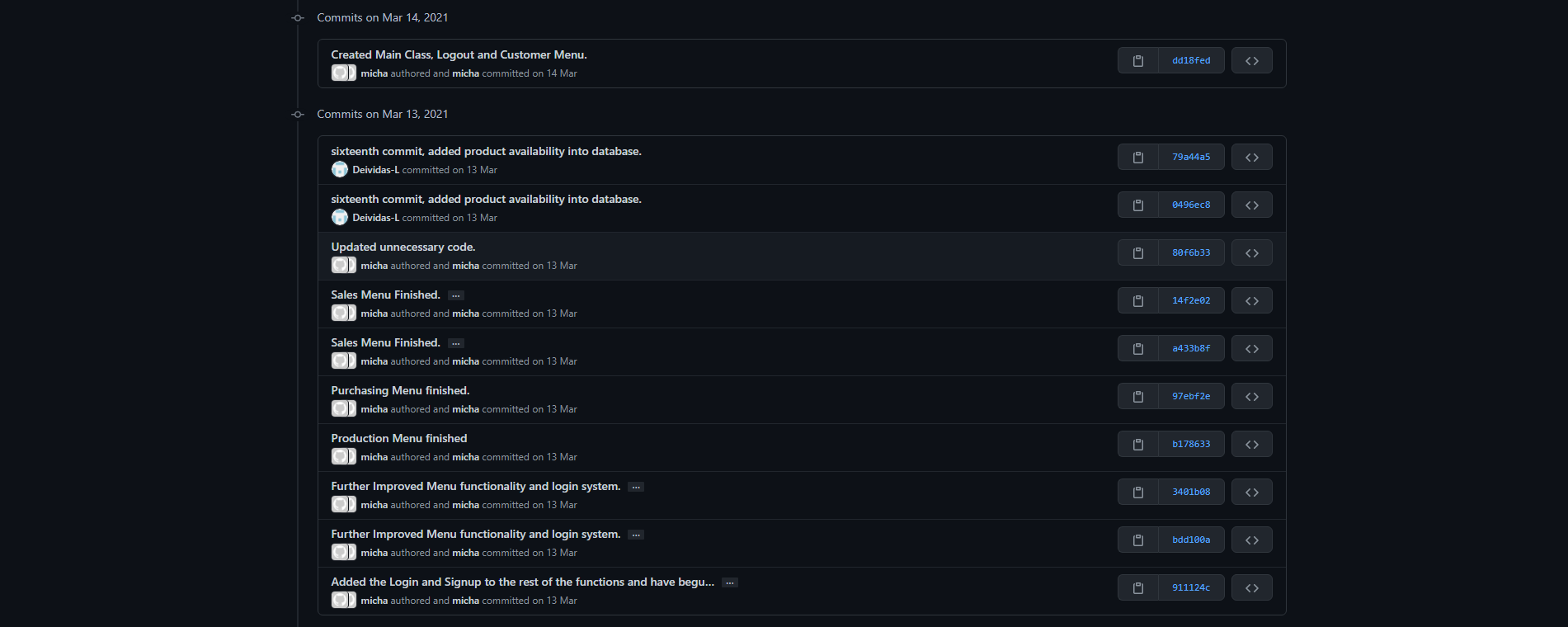
## Use Case Diagram

****

Four actors are identified. Customer (Standard and special) represent bank customers. Customers have access to different functionality depending on their account type. Bank employees have access to back-end functions of the system.

In the use case diagram, there is 4 actors. The customer represents the shop customers. They have access to viewing and buying products and contacting the departments. The software department has access to the backend of the program and can configure everything that is not visible to the user. The employees of the company can have access to view sales, product information, warehouse information, the store and have access to contact each department. Lastly, the manager has access to everything the regular employees have but they can add, remove and edit in each of the sections.

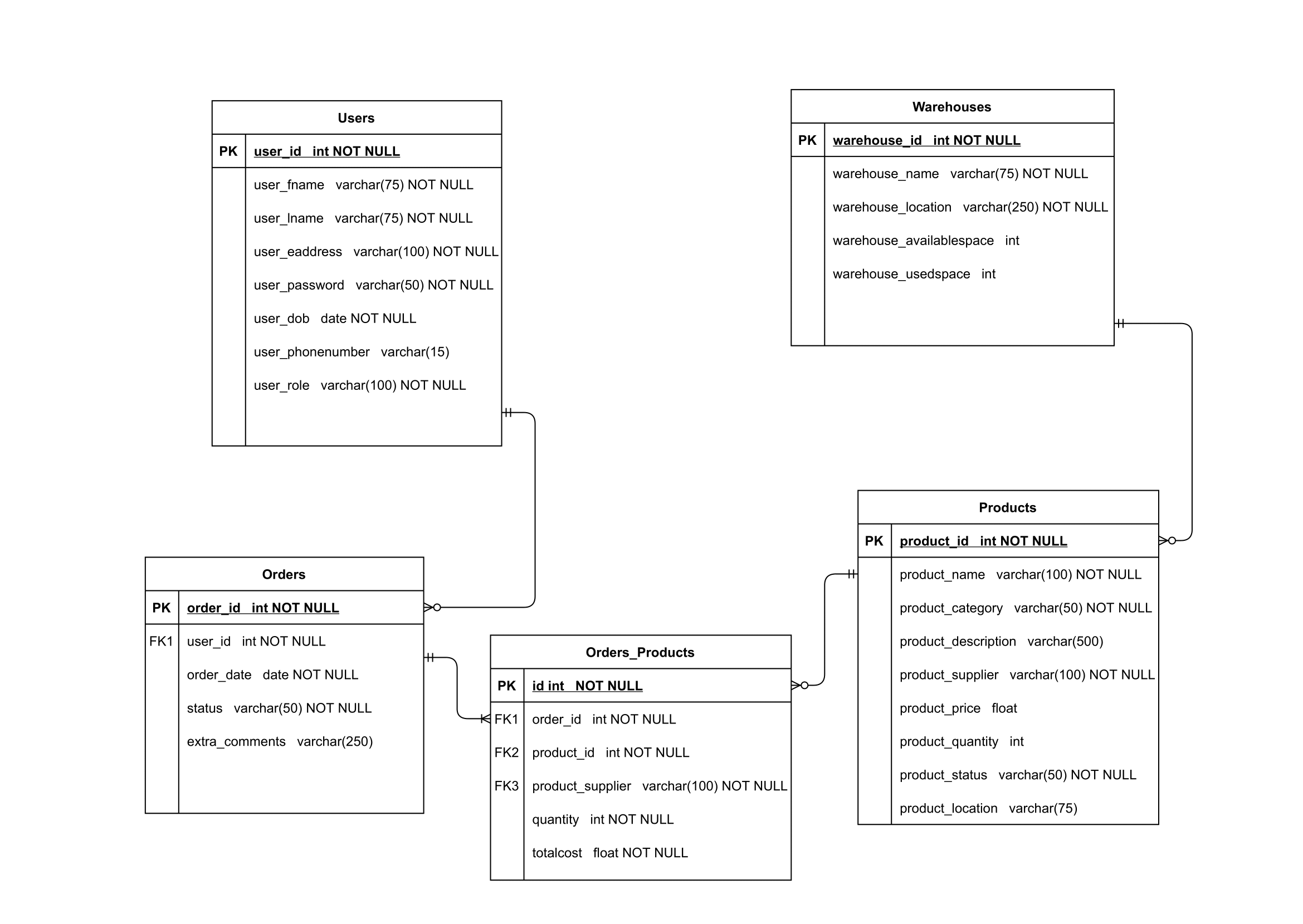
## Evidence of Collaborative Work



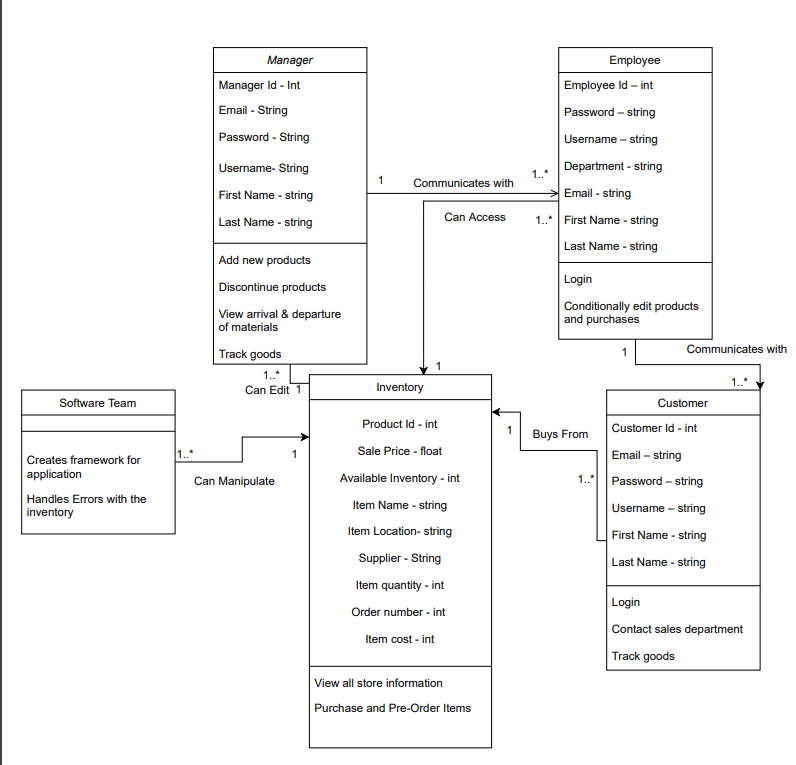
## High-Level Design (Architectural Design)

The diagram above showcases how the system handles the login process. If a customer has an account, it will allow them to login in and browse the product catalogue and if the customer does not have an account, it will create one and then login. It also identifies whether a user is an admin or not which redirects these users to menus with much greater functionality such as editing and removing.

## Detailed-Design



The entity relationship diagram above shows how each part of the program will work together. The purpose of the “Users” entity is that it allows the program to store information about different users and allow them to login to the system using the details stored such as the email and password. The “Users” is connected to the “Orders” entity in a one-to-many relationship. This is because the program allows users to create orders. “Orders” is also connected to the “Products” entity using a link table due to it being a many to many relationship. This is because many products can be put into many different orders. This means that the system will allow users to create orders with many different products. Lastly, “Products” is connected to “Warehouses” in a many to one. This means that many different products can be stored in a warehouse.



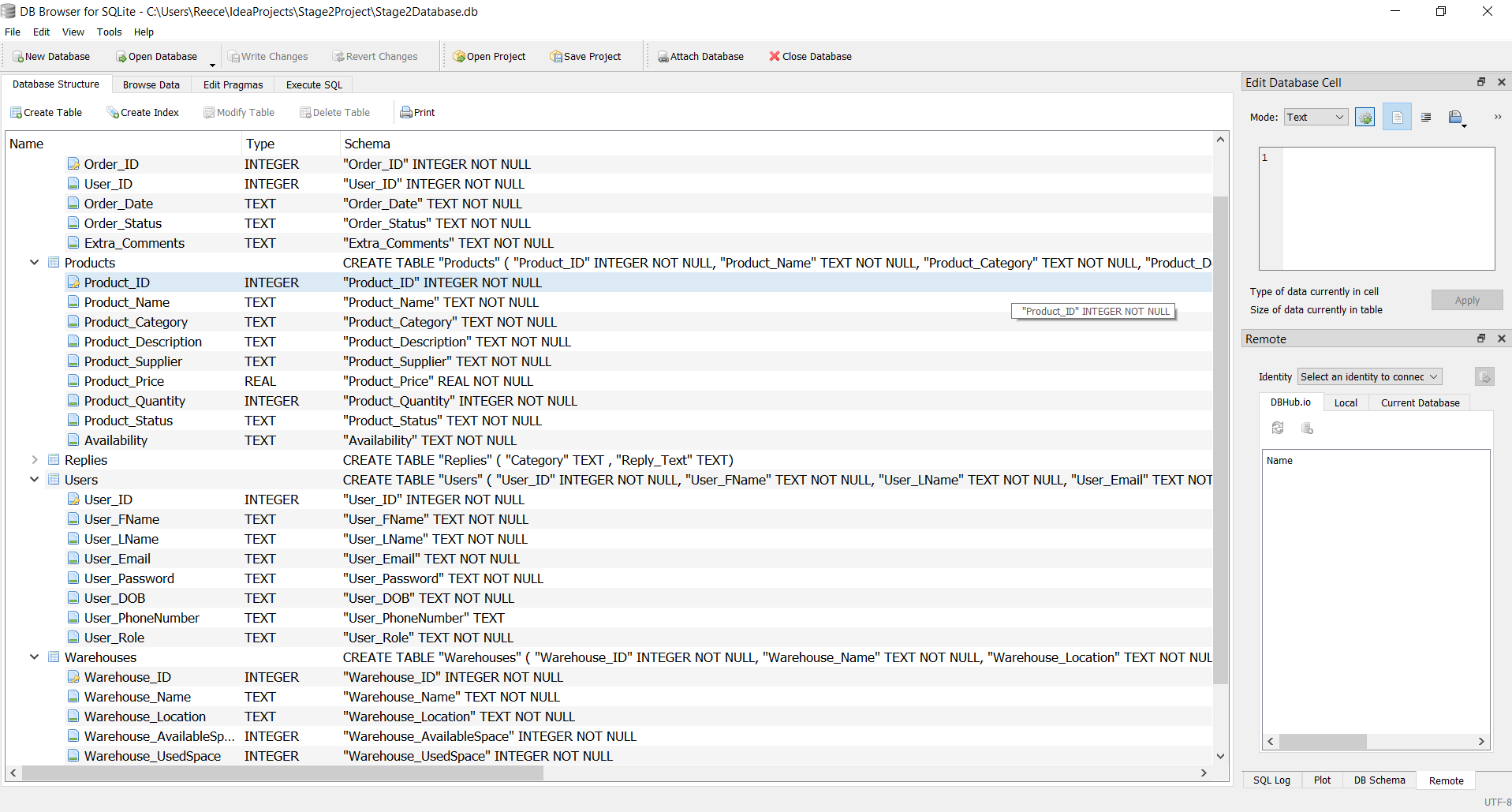
The diagram above shows the attributes of each component of the software and how it interacts with other components. The software team handles all the backend of the software and therefore are the ones who edit and tweak everything. The store interacts with the employees, manager and the customer. It allows the manager to edit and customise the store front such as its products. It allows the employees to access the store database so that they can see what products are available etc and lastly it allows the customer to buy from the store and receive goods. The manager also communicates with the employees by telling them what to do and therefore the employees communicate with the customer by answering questions etc.

## Design Review

[Link to YouTube Video](https://youtu.be/EDJ8YFovU9U)

## Transitioning a Prototype to Production-Quality Software

Transitioning from a prototype to the production-quality software was a simple task due to the diagrams and other documentations we had created. First, transitioning the entity-relationship diagram into a proper database in the software was very simple as the tables and its attributes for the database were already decided so using the program DB Browser, we just simply copied over what we had already created such as the table names including their attributes and data types. We also transitioned the basic design from the wireframe into our software such as the menu layouts and the classes for each portion. This was simple as it only required us to take a quick look at the wireframe and then create the corresponding classes in the main project as the wireframe had all the classes we needed. An example of the transition of the ERD to database can be seen below.

Diagram

Description automatically generated

## Test Specification

For the testing of the project 2 software, we used a strategy that meant that one main person would be responsible for all the testing and giving feedback for each component to each developer so that they can implement any fixes or improvements. When a new feature was added, the person doing the testing would test the functionality and see whether it passed or failed. The outcome of this would then dictate what could be improved. Below is an example of the test plan that we used:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Feature | Test Scenario | Expected Result | Result | Additional Comments | Feedback | Response | Acceptance Test |
| Connection-00 | Created connections from the database to the data. | The system and the database should connect successfully. |  |  |  |  |  |  |
| Input-00 | Created inputs for inserting data. This feature specifically is inserting into the orders table. | The user wishes to insert records into the orders table from the application. |  |  |  |  |  |  |
| Input-01 | Created inputs for inserting new products. | The user wishes to insert records into the products table from the application. |  |  |  |  |  |  |
| Input-02 | Created inputs for inserting new users. | The user wishes to insert records into the user table from the application. |  |  |  |  |  |  |
| Input-03 | Created inputs for inserting new warehouse information. | The user wishes to insert records into the warehouse table from the application. |  |  |  |  |  |  |

## Unit and Accepting Testing

## To easily view the spreadsheet, right click, worksheet object and open.



## Incorporation of formative feedback

## Throughout the development of the stage 2 project, I got lots of feedback from Nnamdi about the creation of our use case diagram, architectural designs and detailed designs. This feedback was given during sessions that took place on Tuesday and Thursday each week. I showed early designs of each diagram which were given suggestions on how they could be made better. The improvements that needed to be made are below:

## **Use Case Diagram**

## Remove types of employees and customers

## Remove the software department relationship

## **Block Diagram**

## User can move inventory after logged in

## Once the user logs in/ registers, check if admin before it can do other admin actions

## **Class Diagram**

## Functions will always have an empty bracket after them

## Do not need the store and the store backend

## Many to many relationship between customers and employees

## These suggestions were then considered, and changes were made to the diagrams to make them better. We also got feedback about our software during its early development from Yomi who liked what we had done so far. He said the functions were very good that we had created and that all we need now is to add a functioning menu to easily link them all together. This was then the part that we worked on next to bring the whole software together.

## Peer Assessment Form (Stage 2)

The peer assessment form can be found in Appendix 1.

# Project 3

## Client Background

Client: Ben Connell from The British University in Egypt

Proposed Project: Attendance Software Prototype

The prototype of this project must meet some key points which are as follows:

* Must enable students of BUE to register their attendance for each lecture using their Student ID
* Must be able to access the application via a computer and mobile device meaning a suitable interface must be used
* Must have some security features to safeguard students and their Student ID numbers. This means that it must guard against ghosting so students cannot spoof their attendance.
* Must have a feature that allows Staff to export and import the attendance into an Excel spreadsheet
* Must be able to send automated notifications when students reach a certain attendance percentage
* Must time stamp all attendance registrations and students should not be allowed to register as attended if the lecture is over unless the lecturer overrides it
* Must be simple and intuitive to use and it must operate systematically, accurately and efficiently.

## Software Project Artefacts

* + 1. Users

The users in this project will be the students and the lecturers.

|  |  |  |
| --- | --- | --- |
| Name | Role | Persona |
| Jack | Student | Jack is a 22 year old Computing Student, and is very good with using software. |
| Susan | Lecturer | Susan is a 57 year old Criminal Psychology Professor. Susan is not very familiar with using automated software. |
| Mehmet | Student | Mehmet is an 18 year old Music Student and has only used a mobile phone and a laptop for a few years. |
| Kieran | Lecturer | Kieran is a 36 year old Chemistry lecturer. Kieran is a perfectionist and enjoys improving things. He also enjoys technology and is very knowledgeable. |

* + 1. User Stories and Acceptance Tests

|  |  |  |
| --- | --- | --- |
| User Stories | Notes | Acceptance Tests |
| Mehmet wants to login to the mobile app using his Student ID. |  | Verify that the system enables Mehmet to login. |
| Mehmet wants to register his attendance for his 9am lecturer. He does so within the first 15 minutes of the lecture starting. |  | Verify that the time period for registering has not expired.  Verify that the system allows Mehmet to register his attendance. |
| Mehmet can easily navigate the menu system of the mobile application. | Can do so by letting people unfamiliar with the application test/use it. | Verify that the mobile application is easy to navigate. |
| A student forgets to register for the lecture even though they were there. Susan is able to edit their attendance for them and mark them as attended. |  | Verify that the system allows Susan to edit the attendance database for her lecture. |
| Susan can view the students that attended the lecture. |  | Verify that the system allows Sarah to see students that attended the lecture. |
| Susan can view the students that did not attend the lecture. |  | Verify that the system allows Sarah to see students that did not attend the lecture. |
| Jack can sign in via his desktop computer using his Student ID. |  | Verify that the system allows Jack to sign in using a desktop computer. |
| Jack cannot register attendance for another student without their Student ID. |  | Verify that the system does not allow Jack to register the attendance of fellow students. |
| Jack can receive a reminder if his attendance is below a certain percentage. |  | Verify that the system can send a reminder to Jack about his attendance. |
| Kieran can export the attendance data for his lectures into a spreadsheet. |  | Verify that the system allows Kieran to export the attendance data onto a spreadsheet. |
| Kieran can generate a QR Code for a specific lecture so students can register for their lecturers. | The QR code will expire within 15 minutes. | Verify that the system allows Kieran to generate a QR Code. |
| Kieran can register a student as late next to their attendance. |  | Verify that the system allows Kieran to register a student as late. |

## Software and Its Presentation

* + 1. The Software Prototype

Uploaded at the submission point.

https://github.com/MikeD1148/Stage3Project

* + 1. Video Presentation

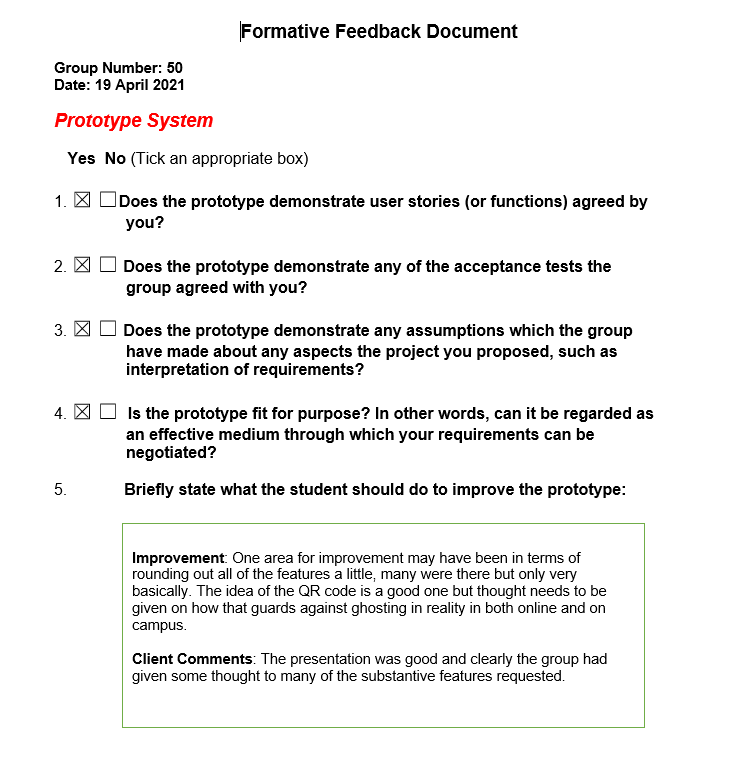
[Link to YouTube Video](https://www.youtube.com/watch?v=CYUTy4a-8os)

## Evidence of Collaborative Work

A picture containing text, monitor, black, screen

Description automatically generated

## Incorporation of Formative Feedback

During the development of stage 3, we were given lots of feedback from our client Ben who suggested many things that could be added or improved. Ben gave us feedback on the QR code idea that would be used to counteract ghosting. The feedback can be found below in the formative feedback document. Using this feedback however, we gave the QR code some more thought and better implemented it into our prototype. Ben also gave us feedback on most of our other features which have now been improved on.

## Peer Assessment Form (Stage 3)

The peer assessment form can be found in Appendix 2.

# Evaluative Report on Legal, Social, Ethical and Professional Issued (1000 words)

## Stage 2

## Relevant Issues

For the inventory management system, there are a few issues that need to be considered to allow for a good and success program. Two of the main issues that need to be taken into consideration are accessibility and security concerns including the GDPR. These issues are quite major when it comes to most software as having an application that is not accessible to a large amount of people lowers the userbase. This is relevant to the project because it meant that we needed to develop a program that is easy to use and simple so that many users would use it. The other issue is security concerns which is quite relevant due to the fact we are storing user data in our software meaning that this data must be protected.

## Discussion

The impact of accessibility and privacy issues could be quite large if they are not correctly considered. The accessibility of an application is very important as it can dictate whether a software will succeed or fail. Having software that can be accessible to anyone no matter what is important as the userbase of the application will be much larger as it can open up the opportunity for people with disabilities to still be able to use it. For the stage 2 application, the ease of use of the program should be considered therefore the user interface should be very simple so that anyone can understand it and use it effectively. Some other accessibility features that would be useful to add but couldn’t due to time constraints would be other language support and the option for a customisable interface such as changing font and text size.

The other issue is privacy concerns which has always been a concern for any piece of software that collects any type of user information. A software that collects information about a user and stores it should have the proper systems to make sure that none of the information can be leaked to any third party such as hackers. The best way of ensuring the safety of a user’s information is an encryption and decryption routine. This ensures that anyone who does not have the decryption key can not decrypt the information and use it for crimes such as identity theft. This is also another feature that should have been considered to add to our database however due to time constraints it was unable to be complete but if the software was to be used within a business, then it would most definitely be a feature and would comply with the GDPR. An example of an organisation not having the proper security of their user’s information is the Marriott International lawsuit. It involved a major breach in security where 339 million guests had their credit card details, passport numbers and dates of birth stolen. This all happened over the course of 4 years before the company finally discovered that there was unauthorised access since 2014. The total fine was a massive £17.7 million.

## Stage 3

## Relevant Issues

For the attendance system prototype, there are a few issues that should be taken into consideration to allow for a good program. Two of the main issues that need to be considered are accessibility and GDPR. Accessibility is a very important issues as it determines the amount of people who can use a program effectively and with ease. This is relevant to the project because it means that the project must have certain features that would allow anyone to be able to use it such as screen reader support for blind people. The other main issue that needs to be considered is privacy concerns such as the GDPR. This is relevant to this project because that data that is stored in programs such as an attendance system must be securely stored so that no data can be acquired by any third party such as hackers.

## Discussion

The impact of these issues could be quite large if features are not implemented to mitigate their effects. For accessibility, software needs to make sure that it has accessibility features implemented so that it allows everyone to easily use the application. A few features that should be added to software are compatibility with screen readers allowing blind people to be able to easily use the program, proper support for all devices to allow people who don’t have certain devices still be able to use the software for example allow users who only have mobiles to still be able to use the program and lastly support for other languages should be important because it allows anyone no matter if they can speak English or not to have the ability to use the software allowing for a wider userbase. An example of a company not having great accessibility for their customers is the Domino’s lawsuit. A blind man sued the company over the inability for him to order food easily from the company due to their mobile app and website not functioning properly with a screen reader. This lawsuit has been a great example to all companies to make sure that all of their users can have an easy and stress-free experience.

For privacy concerns, software needs to make sure that it contains systems to help keep data secure and private. This means that no other third party without authorisation should have access to any of the data as this would be a breach of security. A way to mitigate against this would be to encrypt all the data we store and not share information with anyone. It is important that we follow the GDPR as not following could lead to major fines if our methods of protecting user data fails. An example of an organisation failing to follow the GDPR is the British Airways situation where the company was fined £20 million due to a security breach on customer’s personal information such as login details, payment information, booking details as well as name and address. This breach is major and shows the importance of keeping information secure as a fine as big as £20 million could cripple a company that is small. A breach like this could also affect the customer relationship with the company as a lot of trust will have been lost due to their information getting stolen.

# References

BBC (2020, October 16). British Airways fined £20m over data breach. [Article]. Retrieved from <https://www.bbc.co.uk/news/technology-54568784>

Higgins.T (2019, October 7). Supreme Court hands victory to blind man who sued Domino’s over site accessibility. [Article]. Retrieved from <https://www.cnbc.com/2019/10/07/dominos-supreme-court.html>

Partridge.J (2020, August 19). Marriott International faces class action suit over mass data breach. [Article]. Retrieved from https://www.theguardian.com/business/2020/aug/19/marriott-international-faces-class-action-suit-over-mass-data-breach

# Appendices

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

|  |  |  |
| --- | --- | --- |
|  | Team member + work done | Mark out of 10 |
| 1 | **Michael** was instrumental in building the Use Case Diagram, allowing the group to see how Users will interact with critical parts of the application. As development continued Michael had to rewrite the user stories and acceptance tests as we got a clearer understanding of what functionality the finished product should have. This then allowed him to model the Wireframe to include all the functionality of the User Stories and the layout demonstrated by the Use Case Diagram. During the coding of the application Michael was proficient in creating a detailed and user-friendly menu system enabling the user to navigate all the functions available to them. Finally, Michael added the Reply, View Replies, View Inquiries functionality and added vital bug fixes for all edit, login and search functions. | 9/10 |
| 2 | Initially **Oliver** developed the Class Diagram enabling the rest of the group to appreciate the relation ships between different departments, the types of data we will need from users and the functionality each user will have access to. Providing seamless communication and operation for an Inventory Management Company. Throughout app development Oliver was tasked with testing and evaluating every function. He would then provide individual feedback for improvements or to correct mistakes. To finalise the application Oliver made sure to comment the major functions and populate the database tables with useful information. | 8/10 |
| 3 | **Deividas** was steadfast in developing the User Stories, Acceptance Tests and User Descriptions. This allowed the rest of the team to take this information and relate it to their designs, greatly improving and finalising the foreseen functionality. Deividas was then tasked with the research and setup of the application. Deividas lead the entire group by developing a strong foundation with such functions as Search, Add, Delete and View. He also connected the project to a database and informed the rest of the group how to connect external libraries. | 10/10 |
| 4 | **Reece** developed the Entity Relationship Diagram. This was fundamental when it came to the structure of our database. The Primary Keys, tables and parameters were all copied into our database with minimal improvements to enable the database to cover all necessary functionality. Again, during development Reece created and connected both the Login and Registration functions. However, most importantly Reece took on the challenge of the building the edit functions, persevering and connecting multiple different elements of the application to add an improved user experience. | 9/10 |
| 5 | **James** designed the Block Diagram enabling us to see how a computer would process the decisions of the user. We used this information to create a high-fidelity Wireframe capable of displaying all the functionality and the user interface together. Then during production James created the Inquiries function enabling Customers to directly interact with a representative of the Sales department. | 7/10 |

## 

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

|  |  |  |
| --- | --- | --- |
|  | Team member + work done | Mark out of 10 |
| 1 | **Deividas**  Week 1:  Deividas gathered materials relating to GUI’s and the external libraries needed to run them. He created a tutorial for the rest of the team so that we could learn JavaFX and Scenebuilder more efficiently.  Week 2:  Deividas created and finalised the action plan our group used as a guide to better distribute tasks to individual members.  Week 3:  Deividas adapted the GUI wireframe into an intellij file, and build the foundation needed for the rest of the group to start development. This included installing Scenebuilder, JavaFX and SQLite, creating/naming Java classes and FXML files.  Week 4:  Deividas was instrumental in developing functioning code and graphical design in Scenebuilder. He created the buttons we used to navigate the GUI and the general unfinalized layout. He also developed other specific functionality such as the QR generator and finished the attendance edit. | 10/10 |
| 2 | **Oliver**  Week 1&2:  Oliver gathered and formatted information regarding LSEPI relating to our client, project and other challenges we would need to overcome. Oliver was given this task because we believed it to be group work however, with further communication with Mike Bass we learned that this work was individual. With this information Oliver rewrote his personal LSEPI points and instructed the rest of the group on how to think impartially, critically and ethically. This resulted in every member of the group being able to produce separate, individual well thought out LSEPI research points relating to our client and challenges that we would likely face.  Week 3&4:  Oliver was the second person to start working on the Stage 3 prototype and was vital in developing the login system for both students and lecturers. Oliver used techniques and tactics that we had learnt during our development of the Stage 2 Application (Sqlite, Databases, DB connection strings, Login and Registration code) to create a very efficient login system that connects with a database to responsibly store a user’s information. | 8/10 |
| 3 | **Reece**  Week 1-3:  Reece developed the QR code scanner for our prototype. The scanner provided a mock-up of what it would look like if a student wanted to attend a lecture. During research into the scanner, it was decided that it would be too complicated and not a good use of time to develop a function that opened the camera. So instead, Reece displayed a camera interface that would be replaced with a real camera redirect once the project is out of the prototype phase.  Week 4:  Reece then developed the Student Main Menu, which is the gateway into all the functionality the student has. This is where the student can access the QR code scanner and the logout function. | 8/10 |
| 4 | **Michael**  Michael outlined the functionality the client desired, finalised and expressed that in the form of strict bullet points that served as the basis for the User Stories and finished Prototype. Subsequently Michael also outlined the obvious Stakeholders to get a better representation of the Prototype’s Userbase. Michael then wrote the User Stories and Acceptance Tests keeping all the functionality specified by the client.  Week 2&3:  Michael was the first person to start the design and layout of the prototype by creating a high-fidelity wireframe. During the next meeting with our client, Michael demonstrated the clickable wireframe and received feedback on the design and functionality. This feedback enabled the group to begin production on the final prototype with a new design and added features, such as an export as .csv button.  Week 4:  Michael started development on the Edit Attendance feature. This feature serves as a contingency for the Lecturer if the student is unable to register during the lecture. The Lecturer has the option to change the attendance of a student and give a reason why. All edits are time stamped to reduce unofficial edits and for better records. Finally, Michael recorded the video demonstrating the functionality of the prototype and how it relates to the User Stories. | 9/10 |
| 5 | **James**  Week 1-3:  James attended every discord call and added critical insight into the development of the prototype. James also completed all of his individual work including bits left over from Stage 2 and constantly edited his LSEPI to reflect the feedback he received from Mike Bass.  Week 4:  James finalised the design of the graphical interface, he modelled it after The British University’s campus for a warm and radiant look. He matched the colour palette of the buildings and applied it to every page. James was then useful in catching bugs missed throughout development, such as the wrong pages being attached to the wrong buttons or the buttons being hidden behind the background. | 7/10 |

|  |
| --- |
| Add any comments you feel would be useful for the tutor to know about when assessing marks |
| LSEPI and Extra Work:  Each Group member was also allocated a portion of time to complete their individual LSEPI work and anything left over from Stage 2. Independently everyone researched many examples of LSEPI relating to our Project, Client, all the challenges we would likely face, and what kind of impact this would have on development or publication of the software. All of this was done from an impartial perspective with individual feedback from Mike Bass. |