

**Department of Computing**

**Software Projects**

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

**Name: Annie Bradley**

**Student ID:** **29039131**

**Degree Route: BSc CS**

**Table of Contents**

[1. Stage 1 4](#_Toc64642712)

[1.1 Software Project Artefacts 4](#_Toc64642713)

[1.1.1 Project Description and Users 4](#_Toc64642714)

[1.1.2 User Stories and Acceptance Tests 4](#_Toc64642715)

[1.2 Software and its Presentation 5](#_Toc64642716)

[1.2.1 The Software Prototype 5](#_Toc64642717)

[1.2.2 Video Presentation 5](#_Toc64642718)

[1.3 Incorporation of Formative Feedback 5](#_Toc64642719)

[2. Stage 2 6](#_Toc64642720)

[2.1 Software Project Artefacts 6](#_Toc64642721)

[2.1.1 Users 6](#_Toc64642722)

[2.1.2 User Stories and Acceptance Tests 6](#_Toc64642723)

[2.2 Software and its Presentation 6](#_Toc64642724)

[2.2.1 The Production-Quality Software 6](#_Toc64642725)

[2.2.2 Video Presentation 6](#_Toc64642726)

[2.3 Use Case Diagram 6](#_Toc64642727)

[2.4 Evidence of Collaborative Work 7](#_Toc64642728)

[2.5 High-Level Design (Architectural Design) 7](#_Toc64642729)

[2.6 Detailed-Design 8](#_Toc64642730)

[2.7 Design Review 9](#_Toc64642731)

[2.8 Transitioning a Prototype to Production-Quality Software 9](#_Toc64642732)

[2.9 Test Plan 10](#_Toc64642733)

[2.10 Unit and Accepting Testing 10](#_Toc64642734)

[2.11 Incorporation of formative feedback 11](#_Toc64642735)

[3. Stage 3 12](#_Toc64642736)

[3.1 Client Background 12](#_Toc64642737)

[3.2 Software Project Artefacts 12](#_Toc64642738)

[3.2.1 Users 12](#_Toc64642739)

[3.2.2 User Stories and Acceptance Tests 12](#_Toc64642740)

[3.3 Software and Its Presentation 12](#_Toc64642741)

[3.3.1 The Software Prototype 12](#_Toc64642742)

[3.3.2 Video Presentation 12](#_Toc64642743)

[3.4 Evidence of Collaborative Work 12](#_Toc64642744)

[3.5 Incorporation of Formative Feedback 13](#_Toc64642745)

[3.6 Peer Assessment Form (Stage 3) 13](#_Toc64642746)

[4. Evaluative Report on Legal, Social, Ethical and Professional Issues (up to 1000 words) 13](#_Toc64642747)

[4.1 Stage 2 13](#_Toc64642748)

[4.1.1 Relevant Issues 13](#_Toc64642749)

[4.1.2 Discussion 13](#_Toc64642750)

[4.2 Stage 3 13](#_Toc64642751)

[4.2.1 Relevant Issues 13](#_Toc64642752)

[4.2.2 Discussion 14](#_Toc64642753)

[5. References 14](#_Toc64642754)

[Appendices 15](#_Toc64642755)

[Appendix 1: Software Projects- Peer Marking Form (Stage 2) 15](#_Toc64642756)

[Appendix 2: Software Projects- Peer Marking Form (Stage 3) 16](#_Toc64642757)

# 1. Stage 1

## 1.1 Software Project Artefacts

### 1.1.1 Project Description and Users

|  |  |  |
| --- | --- | --- |
| **Name** | **Persona** | **Role** |
| Business Owner | Katie | Owner of the products within the inventory system. Takes on a more admin role (e.g. tracking sales figures, editing/removing/ adding/restocking products, viewing orders). |
| Warehouse Employee | Blake | In charge of moving products/stock around the warehouse and completing orders. |

### 1.1.2 User Stories and Acceptance Tests

**Business Owner User Stories and Acceptance Tests:**

|  |  |  |
| --- | --- | --- |
| **User Story** | **Notes** | **Acceptance Test(s)** |
| Katie (Business Owner) wants to create an account for her business within the inventory management system | Add a new user to the database | * Verify Katie is added to the database as a new business * Verify Katie is taken to the correct admin business homepage |
| Katie (Business Owner) receives a new order | The business owner receives an alert for the new order. The owner can then progress the order to the warehouse employee so the order can start to be put together | * Verify that there is a new order * Verify the business owner receives an alert for the new order * Verify the stock of the products deceases accordingly |
| Katie (Business Owner) receives an alert that a product is out of stock/low in stock | That product that is now out of stock, will alert the business owner | * Verify that the product stock is less than 5 * Verify the business owner receives an alert for the new order. * Verify the business owner receives an alert that the product is out of stock if product stock is 0 * Verify the business owner receives an alert that the product is low in stock if the product stock is less than 5 but more than 0 |
| Katie (Business Owner) wants to restock an item in her business | The quantity of stock for that product will update to the new quantity | * Verify that the stock quantity is updated |
| Katie (Business Owner) wants to view all the products in her business | The database containing all her products is displayed | * Verify that all the products within the product database are displayed |
| Katie (Business Owner) wants to sort her products by popularity | The products are sorted by the number of items which have been purchased within a given amount of time | * Verify that the system sorts the products in the order requested |
| Katie (Business Owner) wants to change the details of an already existing product. | Changes the details of certain columns within the product database | * Verify the product details have been updated * Verify the business owner receives an alert that *insert product name* has been edited |
| Katie (Business Owner) no longer wants to stock a certain item in her business | Removes a product from the database | * Verify the product has been removed from the database * Verify the business owner receives an alert that *insert product name* has been deleted * It no longer appears on the businesses product page |
| Katie (Business Owner) wants to add a new product to the system. | Add a new product to the database. | * Verify the new product is added to the database * Verify the business owner receives an alert that a new product has been added to the system. |
| Katie (Business Owner) wants to search for a product | The business owner can use a search bar to search for any relevant product | * Verify that products relevant to the search are displayed |

**Warehouse Employee User Stories and Acceptance Tests:**

|  |  |  |
| --- | --- | --- |
| **User Story** | **Notes** | **Acceptance Test(s)** |
| Blake (Warehouse Employee) receives a notification for a new order. | Customer has made a new order, which needs to be collected ready for shipment in the warehouse. | * Verify that the order is completed * Verify the tracking system is used to locate the products and put them together in order to be shipped |
| Blake (Warehouse Employee) wants to locate a specific product | Warehouse employee can use a unique product id to search for the location of a product within the warehouse. | * Verify the location of the product * Verify the tracking system to locates where in the warehouse the product is |
| Blake (Warehouse Employee) wants to let the Business Owner know that an order has been completed | Change the status of the order to *ready to be shipped.* | * Verify that the order is ready for shipment * Verify Blake can check off the order on the system which changes the status of the order for the business owner |

## 1.2 Software and its Presentation

### 1.2.1 The Software Prototype



### 1.2.2 Video Presentation

# <https://youtu.be/VZdeKgRTGuA>

## 1.3 Incorporation of Formative Feedback

The formative feedback I was given in the early stages of stage one helped me develop my user stories so that my prototype reflected an inventory management system more accurately. Originally I had 3 users: a business owner, a warehouse employee and a customer but upon consultation, with my tutor, I was able to recognise that the customer was not a necessary user and instead their user stories could be merged with the warehouse employee and the business owner for a more accurate set of user stories. This meant that when I was developing my prototype I had a very clear idea as to what I was expected to produce in terms of acceptance tests allowing me to easily understand what I believed the requirements for an inventory management system would be and create a wireframe based off of those. It also allowed me time to customise my wireframes by inserting images and a colour scheme in all the scenes which would allow stakeholders to get a better understanding of what the product I would be making them not only functions like but also looks like.

# 2. Stage 2

In this section of the portfolio, the group project development shall be discussed and presented in a detailed and informative manner. The initial discussion regarding how we would tackle this project effectively as a group, but also as individuals, had us firstly discuss our initial prototype systems and how we could utilize them and the skills we had developed in creating them.

Three of the group members had developed a low-fidelity prototype and one had a functional high-fidelity prototype, based on this we decided to utilise the already existing prototype as a base framework.

As we had a solid idea of the functionality we required, Jacob was tasked with further developing the prototype, still as a console application that utilised CSVs, to get an idea for how the application should flow and work in synchronization with itself.

From there we decided upon at least three versions of the system. Version 1 would be the fully functional CSV based console application, Version 2 would still be console-based but will have switched out the CSVs for a more optimal SQL database system. The third and final version will utilise either JavaFX or Swing to create a GUI in order to have a final product that looks good as well as being functional.

## 2.1 Software Project Artefacts

### 2.1.1 Users

|  |  |  |
| --- | --- | --- |
| **Name** | **Persona** | **Role** |
| Admin | Carol | Editing, removing, adding, restocking products, viewing orders |
| Employee | Joe | Search, sort, report errors and view details within the inventory management system |

### 2.1.2 User Stories and Acceptance Tests

|  |  |  |
| --- | --- | --- |
| **User Story** | **Notes** | **Acceptance** |
| Carol (Admin) wants to add new items | Adds a new item to the database | * Verify the item has been added to the database * Admin receives an alert that the item was successfully added |
| Carol (Admin) wants to edit the details of items | Edits selected column in the database | * Verify the details of the item have been updated * Admin receives an alert that the action was a success |
| Carol (Admin) wants to remove an item from the system | Removes an item from the database | * Verify the item has been removed from the database * Admin receives an alert that the item was successfully removed |
| Carol (Admin) wants to restock an item | Increases the stock quantity for a select item within the stock column | * Verify that the stock quantity is updated |
| Carol (Admin) wants to add a new user | Adds a new admin or employee to the database | * Verify that the new user is added to the database * Admin receives an alert that the user was successfully added |
| Carol (Admin) wants to edit a user’s details | Edits selected columns within the database | * Verify that the user details have been updated * Admin receives an alert that the action was a success |
| Carol (Admin) wants to remove a user from the system | Remove admin or employee from database | * Verify that the user has been removed from the database * Admin receives an alert that the user was successfully removed |
| Joe (Employee) and/or Carol (Admin) want to search for a product | Searches through the database based off the user input | * Verify the result of the search is accurate |
| Joe (Employee) and/or Carol (Admin) want to sort products in the system by price (high - low) | Sort products within the database from highest priced to lowest price | * Verify that the products have been sorted in the order requested |
| Joe (Employee) and/or Carol (Admin) want to view the products within the inventory system | Display the products within the database | * Verify that all the products in the database are displayed |
| Joe (Employee) and/or Carol (Admin) want to change select details regarding themselves. | Employees and Admins can change select columns within the database (e.g. Name, Password) | * Verify that the user details have been updated * User receives an alert that the details have been successfully updated |

## 2.2 Software and its Presentation

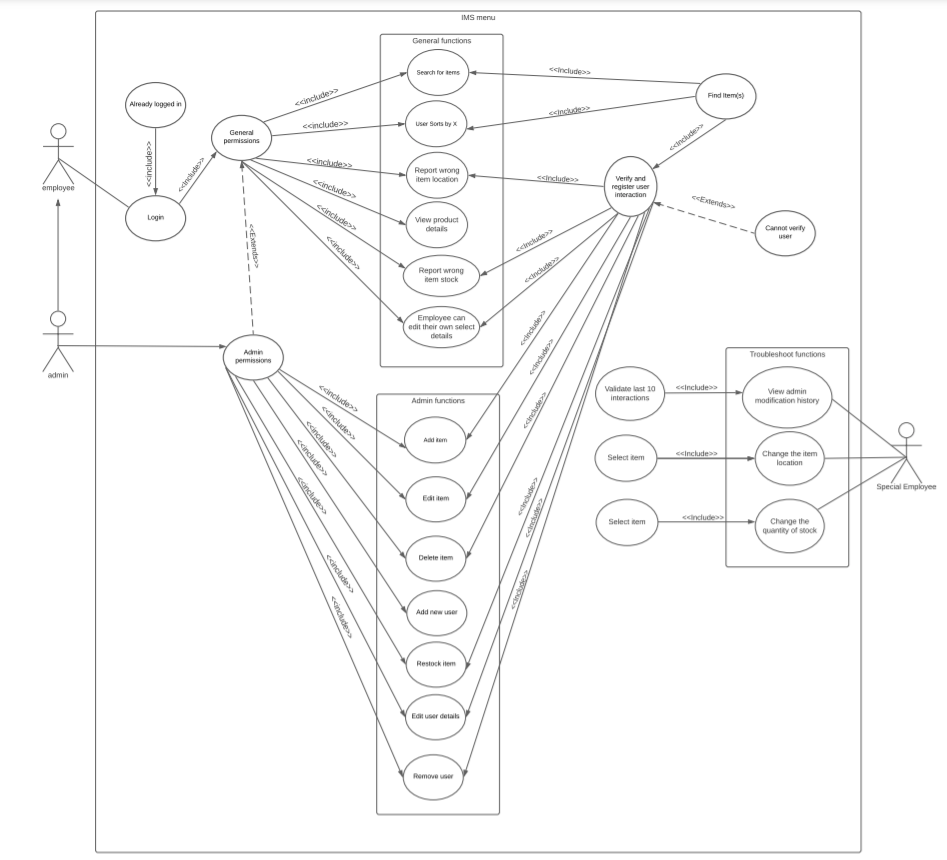
### 2.2.1 The Production-Quality Software

You are expected to submit the project, including all of its components (e.g., codebase), compressed in a zip file (or 7z). The file should be named “Project 2 (Your name)” and must be uploaded to Blackboard as directed in the relevant submission point.

### 2.2.2 Video Presentation

<https://youtu.be/FxizJL4iu-I>

## 2.3 Use Case Diagram

Three actors are identified. Employee (Standard and special) represent warehouse employees, and Admin represents the business owner/managers. Depending on the users role (admin or employee) they have access to different functionality. Admins have access to back-end functions of the system such as edit products, remove products, remove users, etc. While employees (unless they are special employees) can only view, sort and search for products and update their own details.

We decided whilst making the software that we would not include the role of special employee. This is due to the fact that their role overlapped heavily with the admin and it would be difficult to implement the idea of employees manually checking stock and location when the warehouse is fabricated. Alternatively we chose to include multiple admins in our system.

## 2.4 Evidence of Collaborative Work

**Initial Prototype - CSV & Console Based**

<https://github.com/NashOwens/Stage_2-Version-1.git>

**Version Alpha - SQL & Console Based**

<https://github.com/anniebradders/SQL-Console>

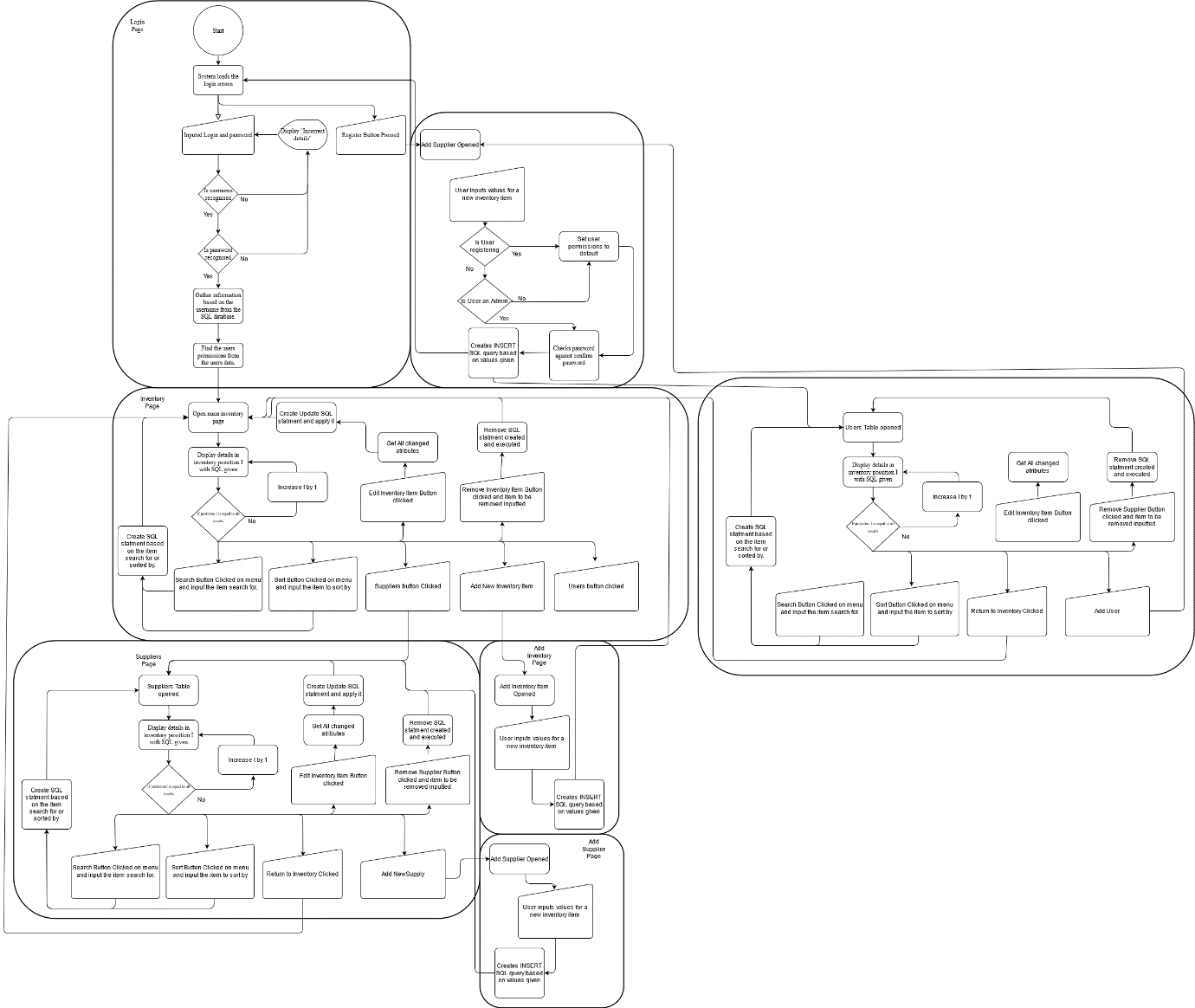
**Swing Version - SQL & GUI Based**

<https://github.com/NashOwens/Stage_2-Version-Swing.git>

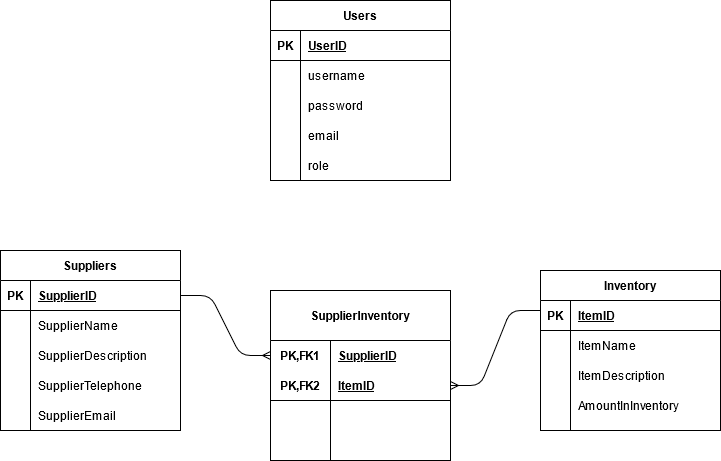
**JavaFX Version - SQL & GUI Based (Final Version)**

<https://github.com/anniebradders/Stage2-Final.git>

## 2.5 High-Level Design (Architectural Design)



## 2.6 Detailed-Design



Originally we planned to have 3 tables. A user table, inventory table and a supplier table. Each item would be linked to a specific supplier so that when the stock of an item is updated the supplier would also receive a notification so the order would be made automatically and it wouldn’t have to be made separately. This would result in less stock quantity errors as the whole system is automated.



Over the course of creating our desktop application however we realised as the supplier would be a third party and we would not be actually working with any companies for this project. This meant that it would be difficult to implement the supplier system as the automated system would rely on the supplier creating our order and shipping it to us. Subsequently we chose to remove the supplier table and only have a user table and inventory table.

## 2.7 Design Review

<https://youtu.be/JEOLOEMvkAA>

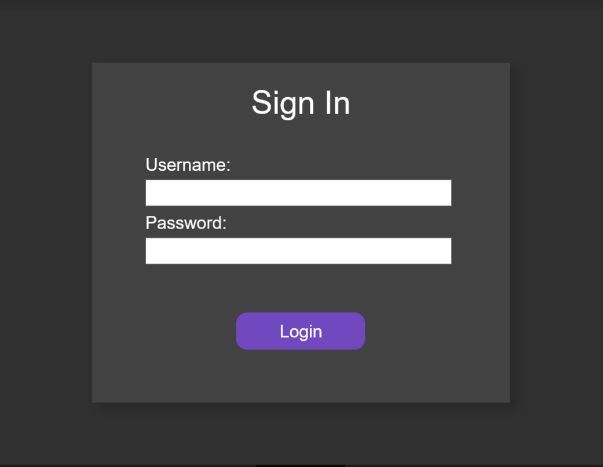
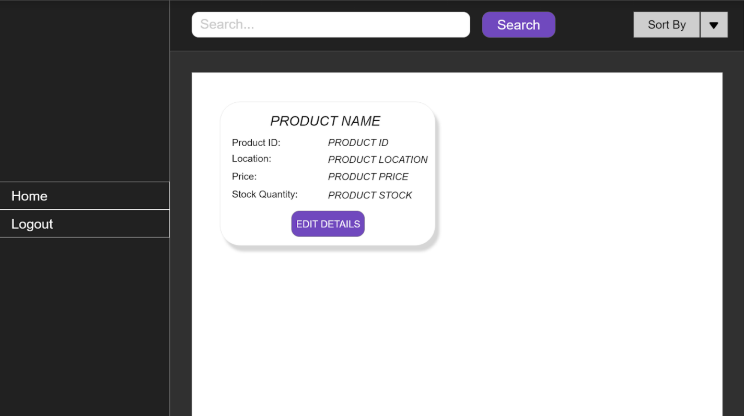
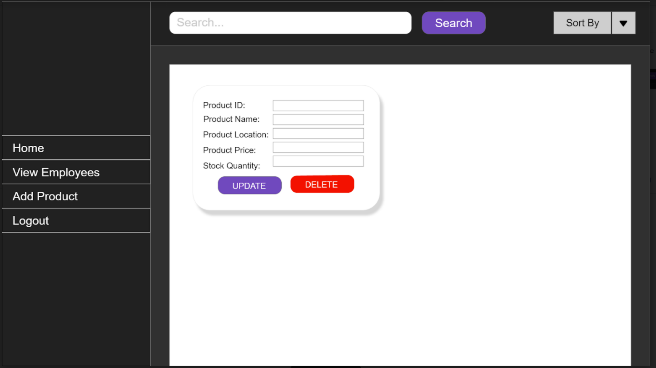
## 2.8 Transitioning a Prototype to Production-Quality Software

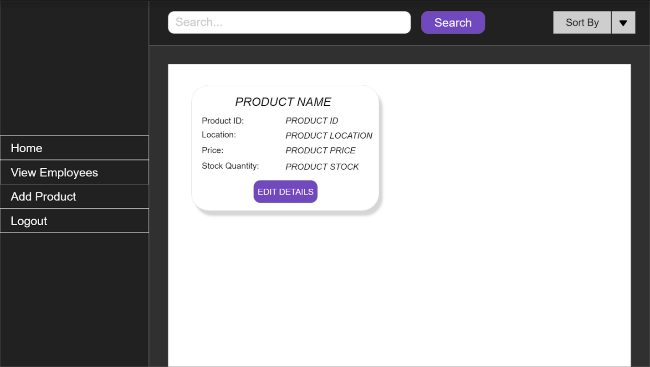
In this section of the portfolio, the group project development shall be discussed and presented in a detailed and informative manner. The initial discussion regarding how we would tackle this project effectively as a group, but also as individuals, had us firstly discuss our initial prototype systems and how we could utilize them and the skills we had developed in creating them.

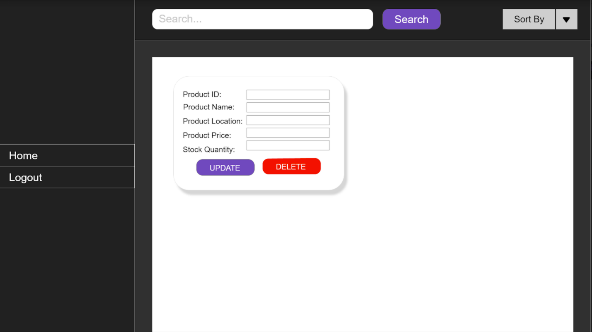
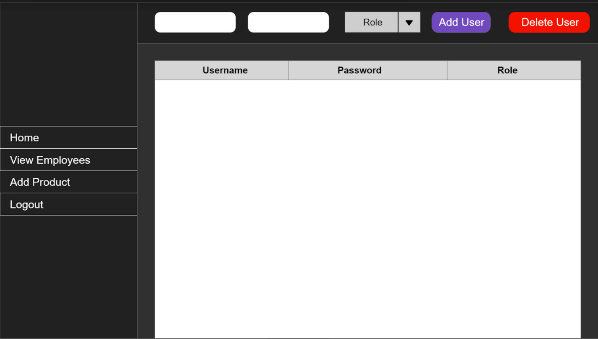
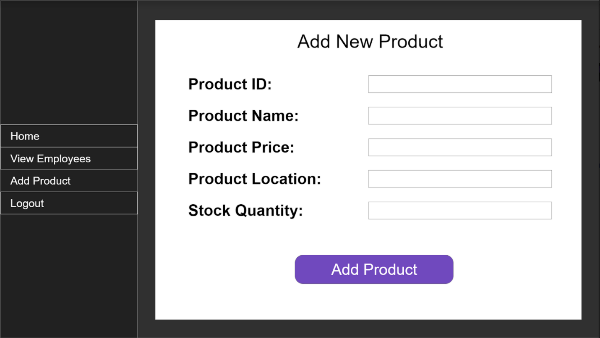
Three of the group members had developed a low-fidelity prototype and one had a functional high-fidelity prototype, based on this we decided to utilise the already existing prototype as a base framework.

As we had a solid idea of the functionality we required, Jacob was tasked with further developing the prototype, still as a console application that utilised CSVs, to get an idea for how the application should flow and work in synchronization with itself.

From there we decided upon at least three versions of the system. Version 1 would be the fully functional CSV based console application, Version 2 would still be console-based but will have switched out the CSVs for a more optimal SQL database system. The third and final version will utilise either JavaFX or Swing to create a GUI in order to have a final product that looks good as well as being functional.

Wireframes:





## 2.9 Test Plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Test Scenario** | **Expected Result** | **Test Result** | **Remark** | **Programmer Response** |
| C1-TID01 | Launch the software | User runs the main | The login.fxml scnee should be displayed |  |  |  |
| C1-TID02 | Exit/Terminate | User selects the x on the top right hand side of the page | The current scene being displayed is closed and program stops running |  |  |  |
| C1-TID03 | Login Button pressed | User enters in their username and password correctly | They are taken to either userMenu.fxml or menu.fxml depending on their assigned role |  |  |  |
| C1-TID04 | Login Button Pressed | Negative Scenario:  User enters the wrong username and/or password | They do not proceed any further |  |  |  |
| C1-TID05 | Login Button Pressed | Negative Scenario:  User enters the wrong username and/or password | They are notified that its either a wrong username and/or password via label and given 2 more attempts to correctly enter them |  |  |  |
| C1-TID06 | Login Button Pressed | Negative Scenario: They enter a wrong username and/or password 3 times | The scene is closed and the program stops running |  |  |  |
| C1-TID07 | View Products | Products.fxml is correctly loaded with the specific product details | The product details are displayed within the product.fxml |  |  |  |
| C1-TID08 | Sort Products | User pressed either the low-high button or the high-low button | The products are sorted in ascending or descending prices depending on the button pressed |  |  |  |
| C1-TID09 | Search Products | User types in the name of a product | Only the product they searched for is displayed |  |  |  |
| C1-TID010 | Search Products | Negative Scenario: user searches for a product that isn’t in the database | A blank page is displayed without any products |  |  |  |
| C1-TID11 | Edit Details of a Product | User enters in the correct data type within the box to update the details | The product details are changed within the database and the new ones are displayed in view products section |  |  |  |
| C1-TID12 | Edit Details of a Product | Negative Scenario:  User enters the wrong data type for a price or stock update | The product details aren’t updated within the database and a label is displayed displaying their error |  |  |  |
| C1-TID13 | Delete Product | User presses the delete button | The product is removed from the database and no longer displayed within the software |  |  |  |
| C1-TID14 | Add new Product | User clicks on the add product button and enters in the correct datatypes for the database | A label is displayed saying the product has been added and when viewing the products it is displayed |  |  |  |
| C1-TID15 | Add new Product | Negative Scenario:  User enters the wrong datatypes when adding the new product | A label is displayed saying that a string has been entered where an integer is and the product isn’t added to the database |  |  |  |
| C1-TID16 | View Employees | User clicks on the view employee button | They are displayed within a table |  |  |  |
| C1-TID17 | Edit employee | User double clicks on the column and row they would like the edit | The database is updated and the new data is displayed |  |  |  |
| C1-TID18 | Add new Employee | User enters into the two textboxes at the top of the scene and selects a role from the drop down list before pressing add user | New user is added to the database and are displayed within the table |  |  |  |
| C1-TID19 | Delete Employee | User clicks on the row (user) they’d like the remove then presses the delete user button n the top right hand corner | User is removed from the database and is no longer displayed in the table |  |  |  |
| C1-TID20 | Logout Button | User clicks on the logout button | The scene is changed to the login scene |  |  |  |

## 2.10 Unit and Accepting Testing



## 2.11 Incorporation of formative feedback

In summary, as a group we feel like we created a very functional and professional system, working together to the best of our abilities whilst all improving our knowledge and growing, not only as academics but also as colleagues.

If we had more time to develop this system, we would have loved to implement a log feature that would detail all the changes made to the database, who changed it and what was changed, but we evaluated that it would be a better use of the limited time to make better features than lots of features.

Throughout stage 2 my group an I attended the majority of the Tuesday and Thursday sessions to receive feedback on our work and progress we had made. Upon showing Yomi and Nnamdi our work we were only given minimal feedback as they seemed happy with the work we were producing as well as the pace that we were producing it at. Due to this reason my group and I stuck to our original plan and managed to keep pace with the project, finishing our stage 2 on time.

# 3. Stage 3

In this section of the portfolio, the Stage 3 group project development will be presented and discussed. We selected to work with Kuwait Financial House, in order to develop a support and resolution system for an internal IT department because we already had an idea of where to begin to create a functional and adequate program.

Due to our increased knowledge of group work and Java development from Stage 2, we were more confident in creating this system collaboratively as opposed to splitting the workload into wider tasks. As we discussed how to tackle this problem, we decided that working on one version together would be beneficial as our individual skills, learned from Stage 2, allows us to focus in specific areas of one large system.

Once again, Jacob created a basic, console-based, prototype system in order to demonstrate and get the group on the same page regarding the functionality and overall hierarchy of the system. From there we worked together to create a GUI based system using JavaFX.

## 3.1 Client Background

Kuwait Finance House is a bank which has been operating since 1977 in accordance with the Islamic principles of Shariah. It is one of the leading Islamic banks with business operations in Bahrain, Turkey, Jordan, Saudi Arabia, Malaysia and affiliates in the United Arab Emirates, Oman, Bangladesh.

We were requested to create an IT Service Management System. In order to do this and incorporate our client without using their name or logo we chose to use a similar colour scheme throughout our project. We also decided to create a simpler program for the employee side so that employees would not get confused and could easily make a request and view the status of their requests without having to partake in any form of training.

For the maintenance side we chose to make things a little more complex however still chose to keep it relatively simple in design. This meant that hopefully less human errors would occur as in its simplicity our system was easy to traverse and understand.

## 3.2 Software Project Artefacts

### 3.2.1 Users

|  |  |  |
| --- | --- | --- |
| **Name** | **Persona** | **Role** |
| Maintenance | Laura | Verifies employee requests, changes their status, can view all employee requests |
| Employee | Lena | Can create a new request and search/sort through their own active and inactive requests |

### 3.2.2 User Stories and Acceptance Tests

|  |  |  |
| --- | --- | --- |
| **User story** | **Notes** | **Acceptance Test** |
| Lena (Employee) able to request a ticket for maintenance to be able to view | Employees can add additional notes in case of extra information needed to be given | Employee makes a request, ticket is saved in a database, maintenance staff confirm the ticket is added via ID search |
| Lena (Employee) can view their requests and its status |  | Employee view their request’s and see what their status is and if it’s correct |
| Laura (Maintenance) can view requests |  | Maintenance log in, view all user requests in the database. |
| Laura (Maintenance) can view employee details from requests |  | Maintenance must be logged in to verify actions.  View all requests from the database. |
| Laura (Maintenance) can change the status of requests |  | Once the request has changed in its progress the staff chooses the appropriate status |
| Lena (Employee) can sort their requests by status |  | Displays only the requests for the status the employee has requested |
| Lena (Employee) can search through their requests active and inactive |  | Using the request\_id the employee was given, when they search for a request it displays only the searched request |
| Laura (Maintenance) can sort the requests by status |  | Maintenance can sort all the requests so only the ones with a specific status are displayed |
| Laura (Maintenance) can search through the requests |  | Using the request id, maintenance can search for a request and only that request is displayed within the table. |

## 3.3 Software and Its Presentation

### 3.3.1 The Software Prototype

### 3.3.2 Video Presentation

https://youtu.be/FfbPp41xZtU

## 3.4 Evidence of Collaborative Work

<https://github.com/NashOwens/Stage_3.git>

## 3.5 Incorporation of Formative Feedback

To conclude, we took this opportunity to really build up on the foundations of what we learned from Stage 2 and create an even better product than we made for that stage. In contrast to Stage 2, we started from scratch, making sure to encapsulate all of the secure data correctly and ultimately we believe we made a really great and thought out system.

We did once again come into issues with JavaFX, as was definitely expected with our limited knowledge and experience with the syntax, and this is an area we all are seeking to improve in as we strive to become better software developers.

Throughout stage 3, my group and I attended a few Tuesday and Thursday sessions as well as a couple Monday sessions with the client. We used the Monday sessions to fully understand the clients requirements and used the Tuesday and Thursday sessions to show the lecturers what we had done in order to check that we were on track and doing the right thing. We ran into one bug within our program which we could not fix so attended a Thursday session to ask for help, however the lecturer was unable to find the cause of the bug so we had to take out the Request ID column from the table displayed within our GUI.

## 3.6 Peer Assessment Form (Stage 3)

Appendix 2.

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

## 4.1 Stage 2

## 4.1.1 Relevant Issues

Throughout the creation of our project for stage 2 we became aware of issues within the world of computing we had to take into account if we were to treat our project like a piece of software which would be used within the world. The two main issues we chose to focus on for our inventory management were data breaches within our database and the sale of illegal/counterfeit goods.

## 4.1.2 Discussion

When creating a piece of software the developer has to be aware of what is happening in the computing industry good and bad, any high profile case of misuse as well as those who are leading the way with innovative solutions and building robust systems that meet their clients’ needs.

It is important to be aware of how your software can be used and its vulnerabilities and weaknesses. Throughout our development process, we constantly reflected on our project and debated about instances in which it could be misused and preventative methods we could take. Using these discussions, articles/videos we were provided with and our research we were able to build our software with an awareness of possible legal, ethical and social issues and implement preventative measures.

One of the main issue we became aware of when we began researching was data breaches. Within our software, we were storing passwords and usernames of all the users within a database which is standard practice across the computing world. However, we quickly became aware that having unencrypted passwords stored within the database, could cause a breach in data security. In 2019, Superma, a security company responsible for the web-based Biostar 2 biometrics lock system that allowed centralized control for access to secure facilities like warehouses or office buildings (Taylor, 2019); At the same time two Israeli security researchers Noam Rotem and Ran Locar, were running a program to scan ports looking for familiar IP blocks and using them to find holes in companies that could lead to potential data breaches. They found that the Biostar 2’s; a new security system which Superma implemented into their access control system AEOS; their database was unprotected and mostly unencrypted. They were able to access over 27.8m records and 23 gigabytes worth of data including fingerprint data, unencrypted users and passwords, personal details of staff – names, addresses etc (Taylor, 2019). In order to prevent an issue like this from occurring within our software we created our encryption method so that all the passwords are encrypted within our database including when a new user is created, an admin edits a user’s password or an employee edits their password. This would hopefully mean that if a data breach occurs the passwords would be harder to get as they are encrypted. This is due to the fact that 59% of people use the same password for everything (Truta, 2013) so although gaining access to our inventory management system if only at the employee level our encryption restricts any further access beyond that point, the likelihood of that password being the same for someone’s email or bank is extremely high (Truta 2013).

Another issue that materialised was the selling of illegal or counterfeit goods. As an inventory management system, our software allows people to store various products within our database, we had to be aware that it may not be used as we intended. In 2017 BBC Inside Out carried out an investigation into the availability of illegal goods on craigslist. Although prohibited on the site, they were able to purchase cocaine from a dealer in Derby which they later tested and found contained high purity levels (Gibson, 2017). Another company that poses a risk to consumers is Shopify. It has been reported that nearly 21% of Shopify stores pose a risk to consumers (BBC News, 2020); with 39% described as problematic sellers, 28% as possible scam stores, 17% had negative reviews and 10% had no transaction history. Shopify was quoted saying "To date, we have terminated thousands of stores and routinely implement new measures to address fraud and other activities that violate our policies." and that they routinely monitor the site for fraud and other violations. Upon reading this, our group realised that putting in place guidelines for our software and monitoring what products were added to the database were most likely not enough to deter people from using it to manage the sales of illegal goods. In order to combat this issue, our software would only be given out to businesses which we could verify and review the products. This would put in place preventative efforts to combat the misuse of our software, Our terms and conditions for any business that wants us to provide our software would be transparent and carry full disclosure to any legal inquiries by the appropriate authorities.

In conclusion, there are many ways our software could be misused and the drivers we plan to use is to combat this by using preventative measures and full legal disclosure to the appropriate authorities. By doing this we believe that we will have taken steps that if there is a data breach, the data they gain access to will be meaningless as it cannot be used by hackers to gain access to more sensitive accounts such as bank accounts or email accounts and also carries the additional penalties of supporting any legal actions that may be taken against the individual/(s). Also by reviewing and verifying a company who wants to use our software we have taken steps to prevent the sales of illegal or counterfeit goods on our website as we would have to approve the products they want to initially add before sending them our software to download and use. By implementing these measures we are confident as a group that we have identified and taken the correct steps in order for the chances of our software being misused are greatly minimized and that our users ultimately feel comfortable inputting their own data without fear of it being misused or completely vulnerable to exploitation.

## 4.2 Stage 3

## 4.2.1 Relevant Issues

Throughout the constant reviewing of our software we became ware of three major issues which I will discuss below: unencrypted database, security compromises and accessibility to those with disabilities.

## 4.2.2 Discussion

Throughout stage 3 we were required to constantly review our software system and evaluate how secure it was not only from hackers but also from potential social and ethical issues. As we were working alongside the client The Kuwait Finance House we felt that this was an especially important aspect, as although our system wouldn’t be implemented in real life we needed to take the time for consideration as if it was. This meant looking at our system under a microscope and being flexible with any developments we found along the way which were potential social, legal or ethical issues.

An issue that we chose to investigate more was unencrypted databases. In our case, it would have been our user database. This is due to the fact that it contains personal information for each employee (e.g. phone numbers, emails, etc.). In February of 2020 MGM Grand Resort in Las Vegas experienced a major data breach, where the personal information of roughly 10.6 million guests had been exfiltrated over an unknown length of time (Ikeda, 2020). The data which was taken included customer names, emails, home addresses, phone numbers and dates of birth (Matthews, 2020). Although this data may seem meaningless to some people, it opens the opportunity for fraudsters to create synthetic identities which they can then use to apply for credit cards and open fake accounts to make fraudulent purchases. It also meant that cybercriminals could launch a spear-phishing (email or electronic communications scam targeted towards a specific individual, organization or business) attack in order to gain more sensitive information about individuals. In order to avoid this situation happening if someone gained access to our user's database, we decided to use an encryption algorithm to encrypt our user's details, then throughout our program when their details needed to be displayed we would run a decryption algorithm so the unencrypted data is displayed within the tables correctly.

Another issue we swiftly became aware of whilst building our system was the ability for someone to make an account if they were not an employee or admin and gain access to what issues are occurring within the banks system in detail. According to Verizon “86% of breaches were financially motivated…” (Sobers, 2021), as the client we are working for is a bank this means that the software we are creating is potentially the most at risk.In order to combat the potential risk of this occurring within our system, we decided to remove the feature where a user could create a new employee or admin account and instead keep the database closed. We did this as we believed the company themselves could supply the users (whether maintenance or an employee) with their own login credentials which they use across other various systems. We felt like not only would this make our system safer to use but also employees would be less likely to forget their passwords or usernames as it would be the same as what they use across other areas of their work life.

A final issue we became aware of was the accessibility of our system. According to the Disability Discrimination Act 1995 legally “[23] “(2) Where a physical feature makes it impossible or unreasonably difficult for disabled persons to make use of such a service, it is the duty of the provider of that service to take such steps as it is reasonable, in all the circumstances of the case, for him to have to take in order to- (a) remove the feature; (b) alter it so that it no longer has that effect; (c) provide a reasonable means of avoiding the feature; or (d) provide a reasonable alternative method of making the service in question available to disabled persons. (4) Where an auxiliary aid or service (for example, the provision of information on audio tape or of a sign language interpreter) would- (a) enable disabled persons to make use of a service which a provider of services provides, or is prepared to provide, to members of the public, or (b) facilitate the use by disabled persons of such a service,” (Participant, 1995). In order to keep within these legalities we chose to use none clashing/contrasting colours to make it access to people with colorblindness.

In conclusion, we felt that the project we were building was relatively safe however there were steps which we could and did take as preventative measures to ensure that our system was not as vulnerable to attack. We believe that encrypting the employees details so even if access was gained to our user database the information would be useless, keeping our employees safe from phishing attacks which have risen roughly 70% in the past year according to an annual report called the Digital Defense Report which Microsoft released (Microsoft, 2020). Alongside that by keeping the database closed we feel like that would also limit chances for vulnerabilities within other software systems The Kuwait Finance House use would be limited, it would become more difficult for anyone to infiltrate the system and see whether they're having a critical incident and exploit that situation to their favor. Alongside this we also implemented a few user friendly options by having basic colours and a simple layout so that people with and without disabilities could easily traverse our software.

# 5. References

*Taylor, J. (2019, August 15). Major breach found in biometrics system used by banks, UK police and defence firms. The Guardian. https://www.theguardian.com/technology/2019/aug/14/major-breach-found-in-biometrics-system-used-by-banks-uk-police-and-defence-firms*

*Gibson, B. J. (2017, October 23). Websites allowing illegal activity ‘complicit in crime’. BBC News. https://www.bbc.co.uk/news/uk-england-41601075*

*Truta, F. (2018, May 3). 59% of people use the same password everywhere, poll finds. Security Boulevard. https://securityboulevard.com/2018/05/59-of-people-use-the-same-password-everywhere-poll-finds/*

*BBC News. (2020, December 23). Shopify stores riddled with fakes and fraudsters.* [*https://www.bbc.co.uk/news/business-55420445*](https://www.bbc.co.uk/news/business-55420445)

*Ikeda, S. (2020, July 27). New Details Indicate That Scope of the 2019 MGM Data Breach Is Much Bigger Than Expected. CPO Magazine.* [*https://www.cpomagazine.com/cyber-security/new-details-indicate-that-scope-of-the-2019-mgm-data-breach-is-much-bigger-than-expected/*](https://www.cpomagazine.com/cyber-security/new-details-indicate-that-scope-of-the-2019-mgm-data-breach-is-much-bigger-than-expected/)

*Mathews, L. (2020, July 14). For Sale: Hacked Data On 142 Million MGM Hotel Guests. Forbes.* [*https://www.forbes.com/sites/leemathews/2020/07/14/mgm-142-million-guests-hacked/?sh=82761e352945*](https://www.forbes.com/sites/leemathews/2020/07/14/mgm-142-million-guests-hacked/?sh=82761e352945)

*Microsoft. (2020, October 2). Microsoft report shows increasing sophistication of cyber threats. Microsoft Malaysia News Center.* [*https://news.microsoft.com/en-my/2020/10/02/microsoft-digital-defense-report-cyber-threats/*](https://news.microsoft.com/en-my/2020/10/02/microsoft-digital-defense-report-cyber-threats/)

*Sobers, R. (2021, March 17). 134 Cybersecurity Statistics and Trends for 2021 | Varonis. Inside Out Security. https://www.varonis.com/blog/cybersecurity-statistics/*

# Appendices

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

|  |  |  |
| --- | --- | --- |
|  | Team member + work done | Mark out of 10 |
| 1 | Annie Bradley – javafx project, test plan, unit testing, 33% user stories + acceptance tests, wireframes, 50% SQL | 10/ 10 |
| 2 | Jacob Jarvis – debugging javafx project, 100% encryption routine (used within javafx application), 100% console prototype, 33% user stories acceptance tests, | 10/ 10 |
| 3 | Nash Owens-Kerr – 100% class diagram, 100% swing application, 33% user stories acceptance tests, debugging javafx project, | 10/ 10 |
| 4 | Jude Barnett – 100% ERD, 100% Architecture Design | 4/ 10 |



|  |
| --- |
| Add any comments you feel would be useful for the tutor to know about when assessing marks |
|  |

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

This form must be filled in as a group by honestly evaluating your contribution to the work. Each member’s contribution to the project must be clearly stated. Finally, each member must be rated out of 10 (10 being the highest contribution and 0 being no contribution at all). The highest mark must always be 10, e.g.

|  |  |  |
| --- | --- | --- |
|  | Team member + work done | Mark out of 10 |

|  |  |  |
| --- | --- | --- |
| **1** | Annie Bradley - Cooperation on development of software, user stories, acceptance tests, client background, LSEPI. | 10/ 10 |
| **2** | Jacob Jarvis - Cooperation on development of software, user stories, acceptance tests, organisation of structure, initial prototype. | 10/ 10 |
| **3** | Nash Owens-Kerr - Cooperation on development of software, user stories, acceptance tests, encapsulation of program. | 10/ 10 |
| **4** | Jude Barnett - Cooperation on development of software. | 6/ 10 |

|  |
| --- |
| Add any comments you feel would be useful for the tutor to know about when assessing marks |
|  |