

System Implementation

Stock Management system

Final Report – Report 3

Wajih, Zaid

IUPUI – INFO -C451

SHAWN DAI

Contents

Section 1: Customer Problem Statement	2
Section 2: System Requirements	4
Section 3: Functional Requirements Specification	5 ((Revised))
Section 4: User Interface Specification	9 ((Revised))
Section 5: Domain Analysis	10
Section 6: Plan of work	13 ((Revised))
Section 7: Interaction Diagrams	14 ((Revised))
Section 8: Class diagram	23 ((Revised))
Section 9: Typical Use Cases	24 ((Revised))
Section 10: Data Types	27 ((Revised))
Section 11: System Architecture	29
Section 12: Project Management	31 ((Revised))
Section 13: User Interface Design	32
Section 14: Design of Test	32 ((Revised))
Section 15: Project Management and Plan of Work	32 ((Revised))
Section 16: Final Report 3	34
References	38

Section 1: Customer Problem Statement

A case study at "Zake's Warehouse" (warehouse that keeps all sorts of products such as home supplies and office supplies) they cited issues regarding keeping track of all their warehouse inventory products information, Imports, and Exports; Imports as all new products that have been ordered and export all products that all Zake's clients have purchased.

The user currently manually keeps track of each warehouse inventory using folders and folder cabinets to archive the data in case of a need to get back to it, such as any record or audits purposes, calculating and analyzing the data to enhance the performance in the future. This process usually requires human resources and a significant amount of time to complete it; it's also not free from human errors. Users keep track of; Sold product information, inventory of their stock in each warehouse, reports, supplier information, payment details information, all client information. Sometimes they miss sending orders because they lose track of sales details and place orders to the vendor by doing it manually.

I propose a solution for Zake's Warehouse accountant or any user that wants to perform one/all and more tasks from mentioned above using software linked with a database. Each warehouse has a separate DB that holds its record, but for now, they want to pilot the software in their main warehouse and see if it's worth investing in the rest of the warehouses to use this software.

The software will keep track of inventory in the warehouse and revise it according to the daily sales and received inventory. All individual products are linked to a unique product number that helps identify the product in different tables in the database. To help keep all data up to date in the DB through utilizing the inventory. The software is dynamic if the company decides to expand their inventory, item and new product can easily be added to the DB.

The software provides an efficient interface to the users in the warehouse to pull any data they need promptly to help improve the productivity of the employee day. The software helps with pulling reports after analyzing all the data of total sales/ordered products. Users can add and delete any products listed in the system. Users can view/edit all supplier information. As I mentioned, the software is dynamic if decided to add it in all other warehouses; the software then allows the user to coordinate all inventory across all warehouses locations and sales to help better keep the logic flow and enhance it and to ensure that all stock inventory managed effectively and efficiently for the whole company.

Section 2: System Requirements

In this table, I will list all system functionality from the problem statement, sort them by their priority weight for the business, and describe each one.

NO.	Priority Weight	Description
REQ-1 (log in)	Critical	This page allows the users to enter their credentials to log in
REQ-2 (MainStock_page)	Critical	This is the main page the user will land on after the login process
REQ-3 (product_page)	Critical	On this page, the user can view the details of the products and modify them as needed
REQ-4 (vendor_page)	Critical	On this page, the user(manager, Admin) can view and modify the vendor details
REQ-5(employee_page)	Critical	On this page, the user(manager, Admin) can view and modify the employee details as well as add a new employee
REQ-6 (manger_page)	Critical	Admin can add and modify manager details as well as add a new manager
REQ-7(position_page)	High	On the page, the user(Admin, manager) can add and modify position details

Nonfunctional requirements-

- System to have an easy interface
- System to be reliable
- System to be able to hold a decent amount of data
- System to be able to pull, add, delete and update data in a timely matter

Section 3: Functional Requirement Specification

Stakeholders

In this case, the stakeholder is Zake's warehouse owner and the warehouse managers, and employees.

The owner of Zake's warehouse benefits because it enhances the productivity of the warehouse; it will free more time for the managers to do more data and inventory analysis to improve the flow in the warehouse.

The employee benefits from the software because it automates many daily tasks, and the software will provide it more efficiently.

Use Cases

1. User to have a different user account log-in

Users are directed to the log-in page; users need to enter their credentials.

2. User to be able to add a vendor (company)

Users can add a new vendor to the vendor list in the DB.

3. User to be able to check the inventory

This page displays the overall inventory for the warehouse. Users can search by product name

4. Add new product

Users can add a new product to the DB.

5. Delete products

Users can delete products if they stop supplying them.

6. Remove vendors (suppliers)

Users can remove vendors in case they stop dealing with them.

7. Modify suppliers information

Users can update the vendor information in case of change.

8. View order details

Users can check all order states; canceled orders, placed orders, and returned orders.

9. View payment details

Users can check all payments details and payments for the vendors

10. View client

Users can check all customer's details as well as modify them in case of any changes.

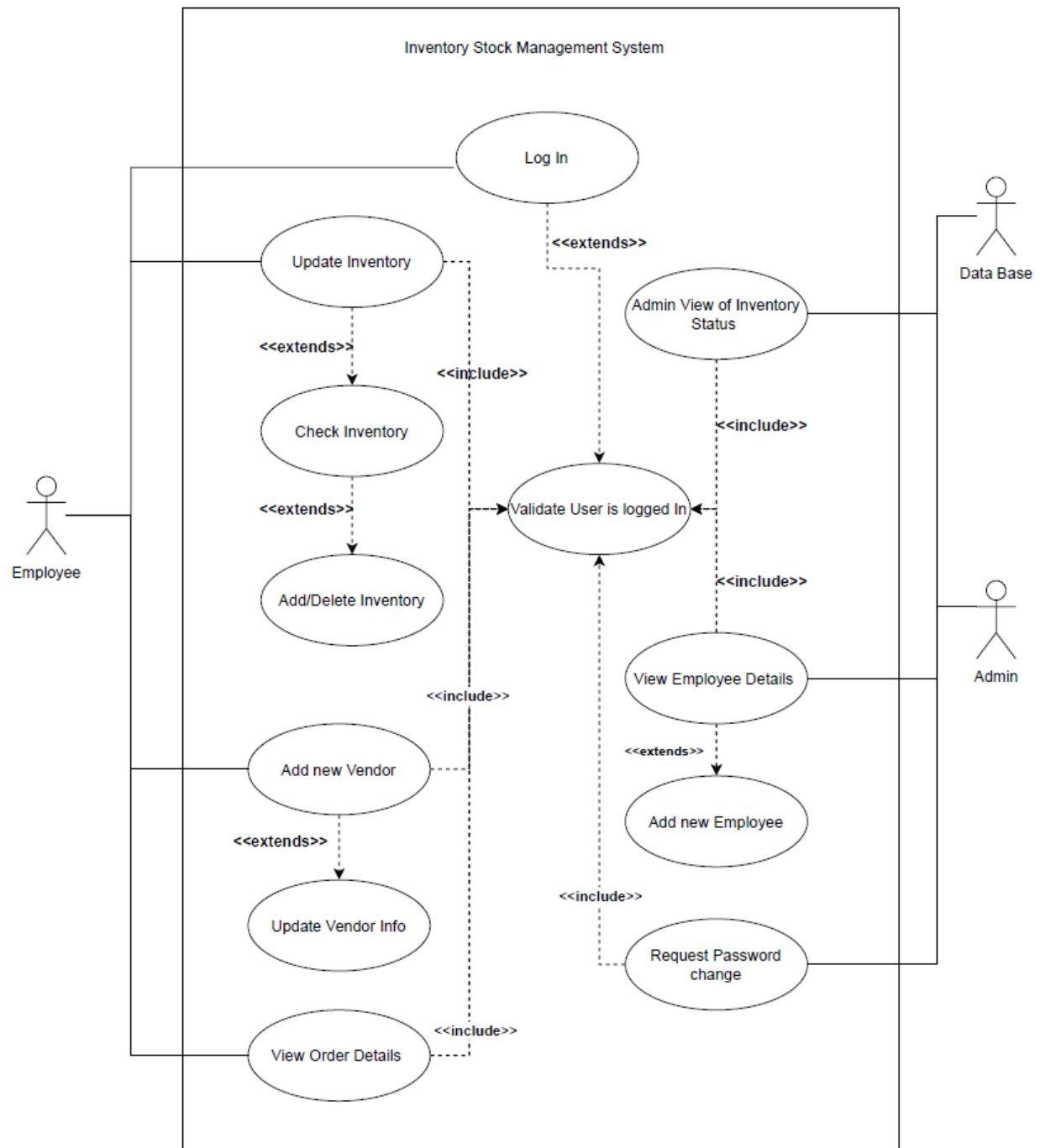
11. Add Users

This is for the Admin user for the system; Admin user has access to add and remove user accounts in the software and controls their access.

12. Change Password

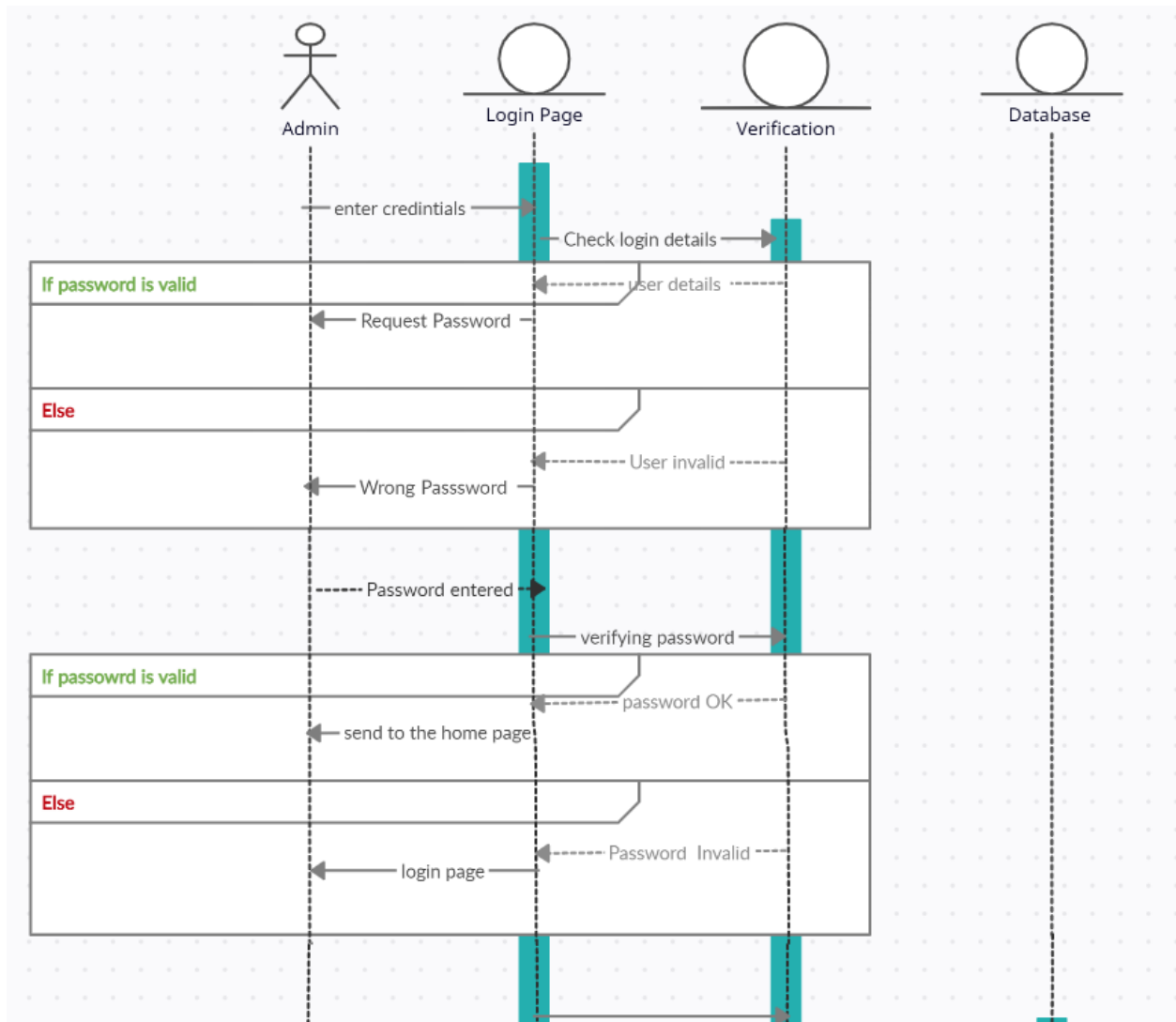
This allows admin users to reset or change other users' account passwords in case of a lost or locked account.

Use Case Diagrams



System Sequence Diagram

Login process



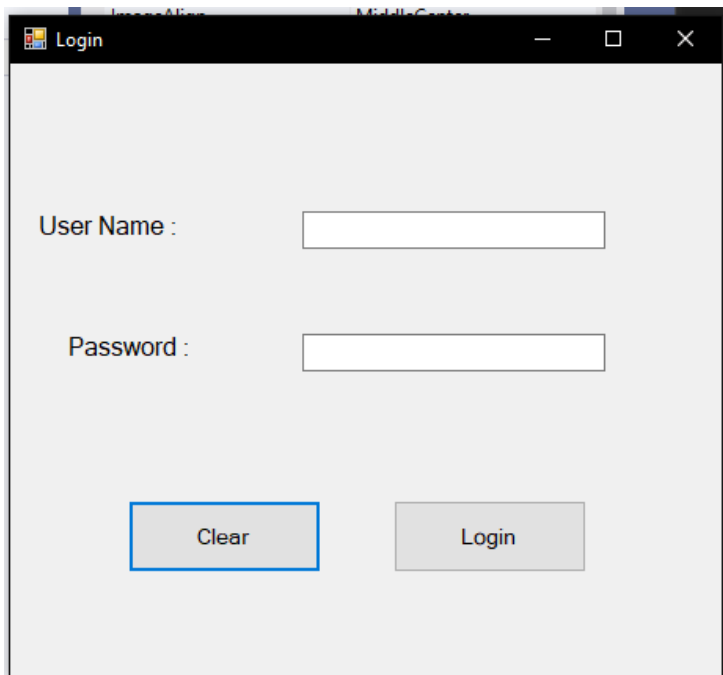
Section 4: User Interface Specification

Preliminary Design / User Effort Estimation

This will be the login popup window once started the software.

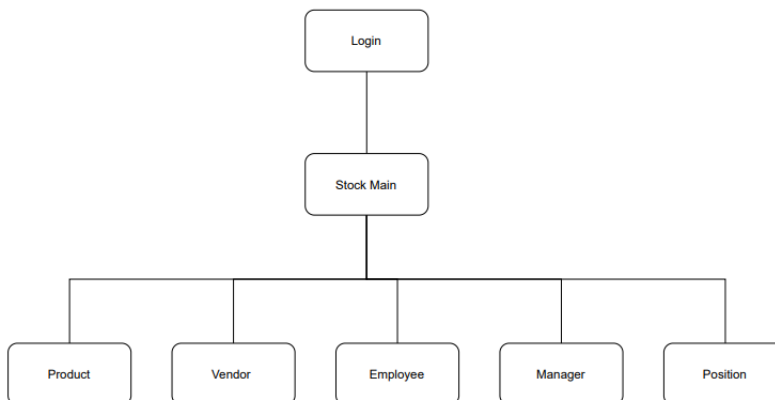
Users need to fill in their credentials for the username; usually, users enter approx ~8-12 letters and approx. ~ 8-24 characters for the password and then one mouse click to log in.

Users must enter their username and then click on forget the password to start the forgotten password process.



The screenshot shows a login window with a title bar containing the text 'Login' and standard window control buttons (minimize, maximize, close). The main area of the window is light gray and contains two input fields. The first field is labeled 'User Name :' and the second is labeled 'Password :'. Below these fields are two buttons: 'Clear' and 'Login'. The 'Clear' button is highlighted with a blue border.

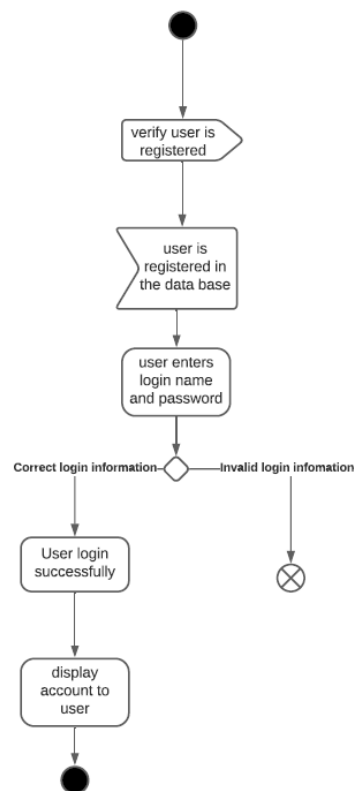
This is for the login process flow



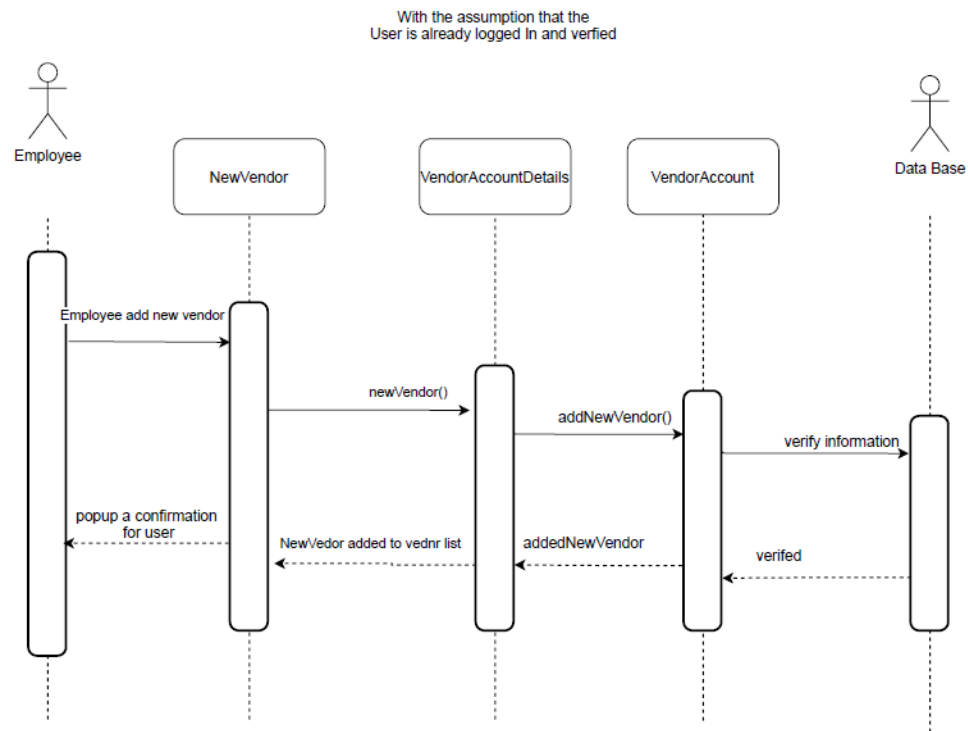
Section 5: Domain Model

This software aims to help manage and control the import and export products in the warehouse to help minimize waste costs, damage..etc. As mentioned in the customer problem statement, this software will only be piloted in the HQ warehouse to test and see if there is a domain for deploying it in the rest of Zake's warehouses. The software is dynamic and can easily be deployed for the rest of the warehouses by adding the needed DB. We aim to have complete visibility over the company warehouse to help with comprehensive, accurate data and numbers of reports.

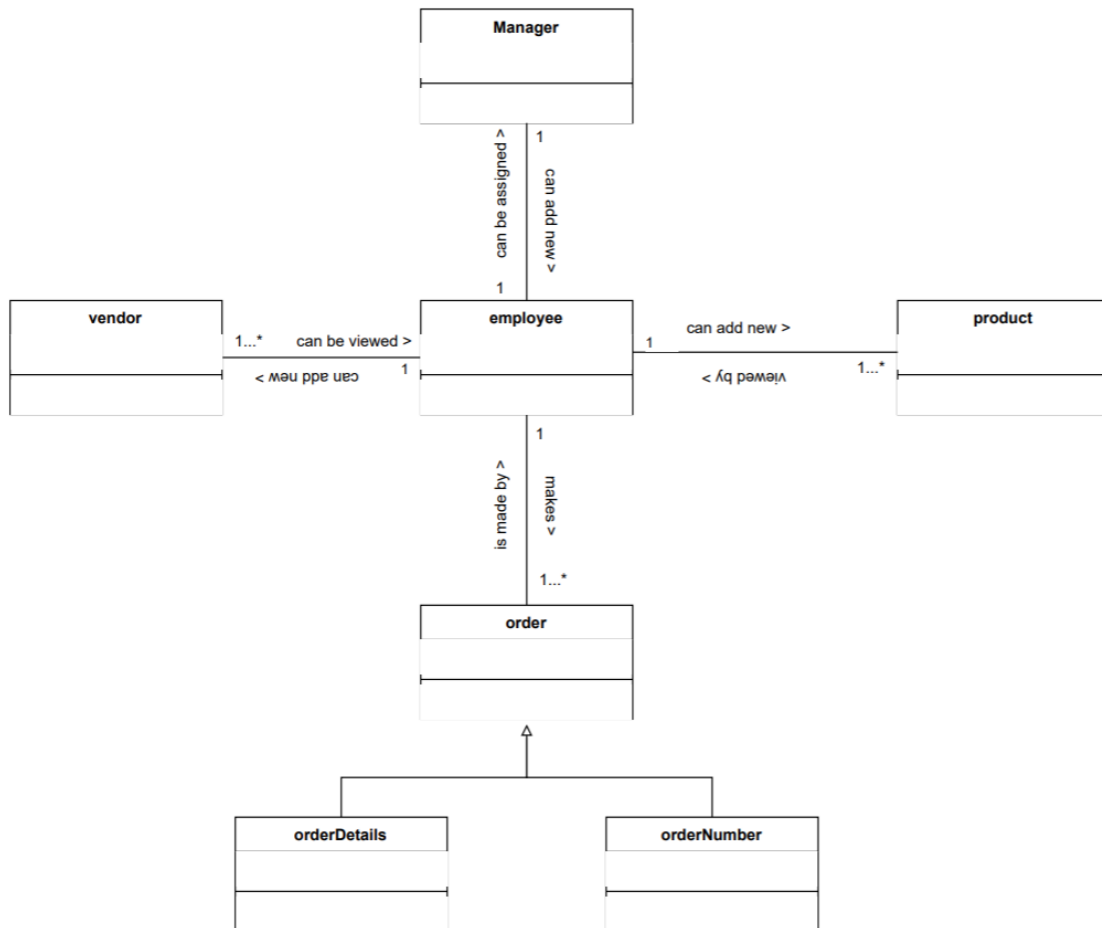
Activity Diagram for User login



Sequence Diagram for adding a new vendor to the vendor list in the database



Class Diagram for the employee function

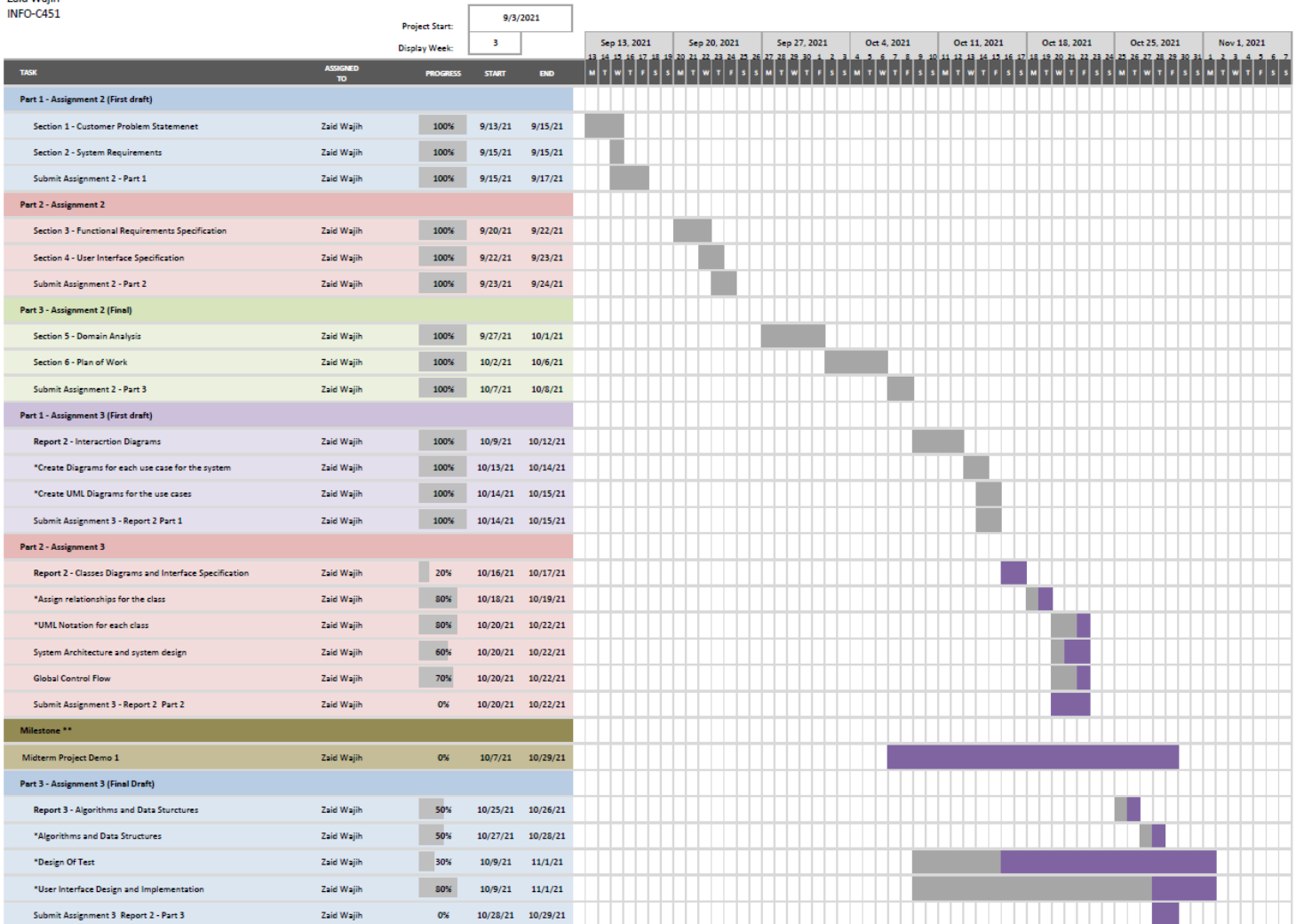


Section 6: Plan of Work

In this project, a system must be designed to support connectivity with a database management system that will store and keep all records for the system. The software application to be made consists of at least main business functions and meet all business requirements. The application will involve basic e-commerce activities, such as employee, manager login, browsing between products, and buy/sell activities. The entire system must be developed in (C#) to be easy to maintain and extend. Below is the project plan for each implementation section for the project.

Stock Management System Implementation

Zaid Wajih
INFO-C451

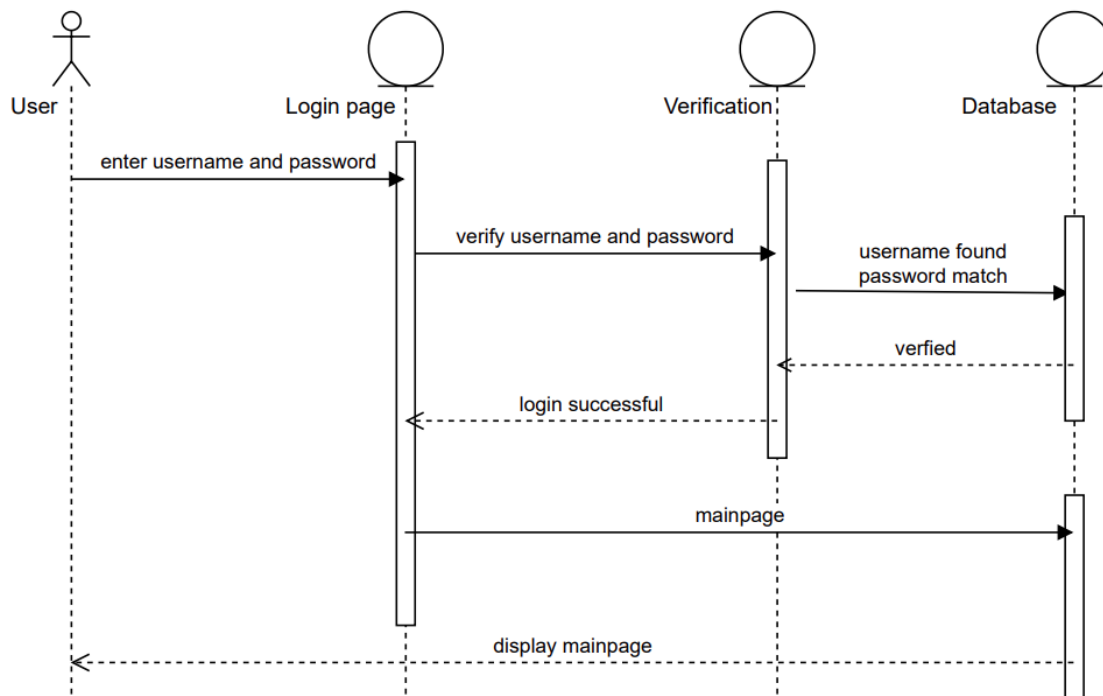


Section 7: Interaction Diagrams

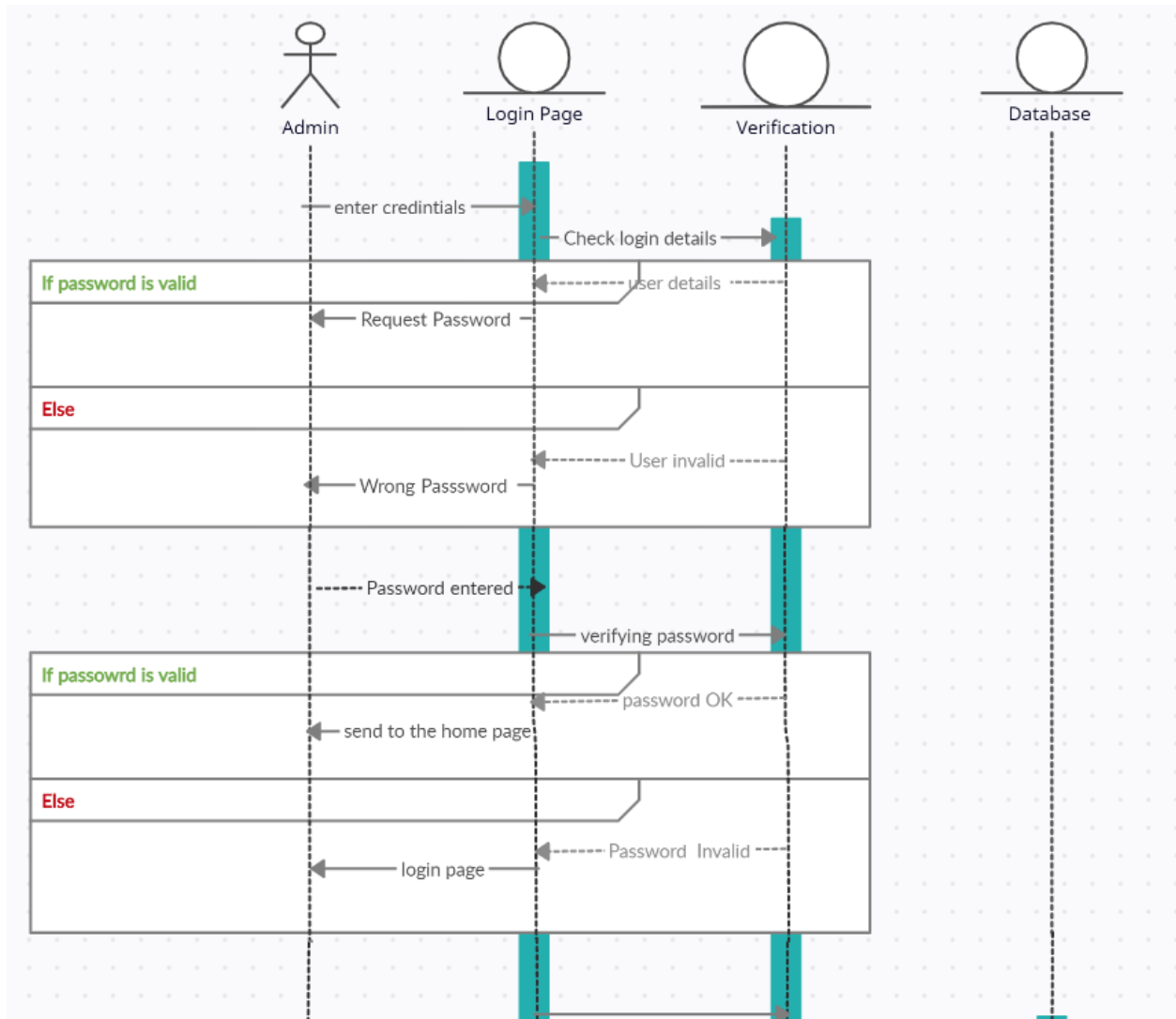
I will provide all the sequence diagrams for the software I will implement in this session.

Login –

Once the user launches the app, a pop-up window will display and let the user enter the username and password, is the user clicks to log in without entering the user name and password, a pop-up window will display “Enter user name and password.” When the user successfully enters the username and password that matches the one in the DB, the main page for the application, the user can choose which page he wants to be on.

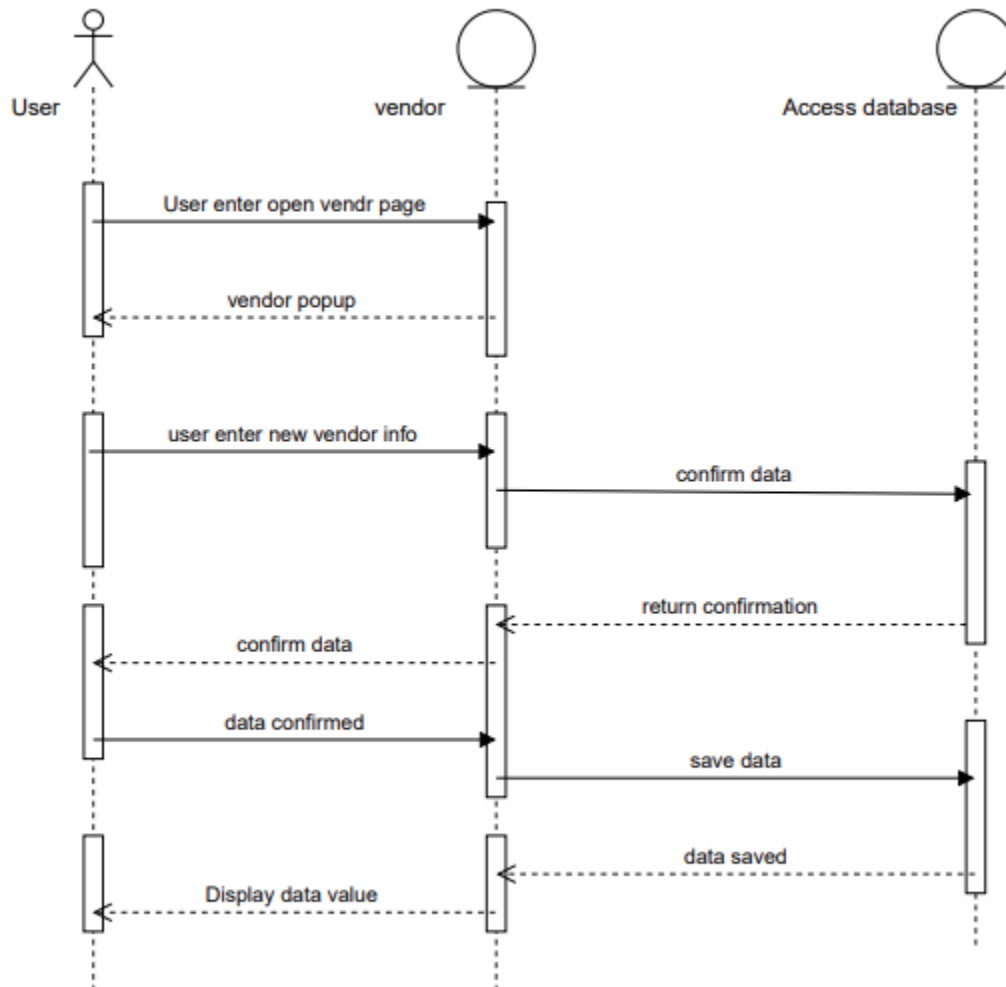


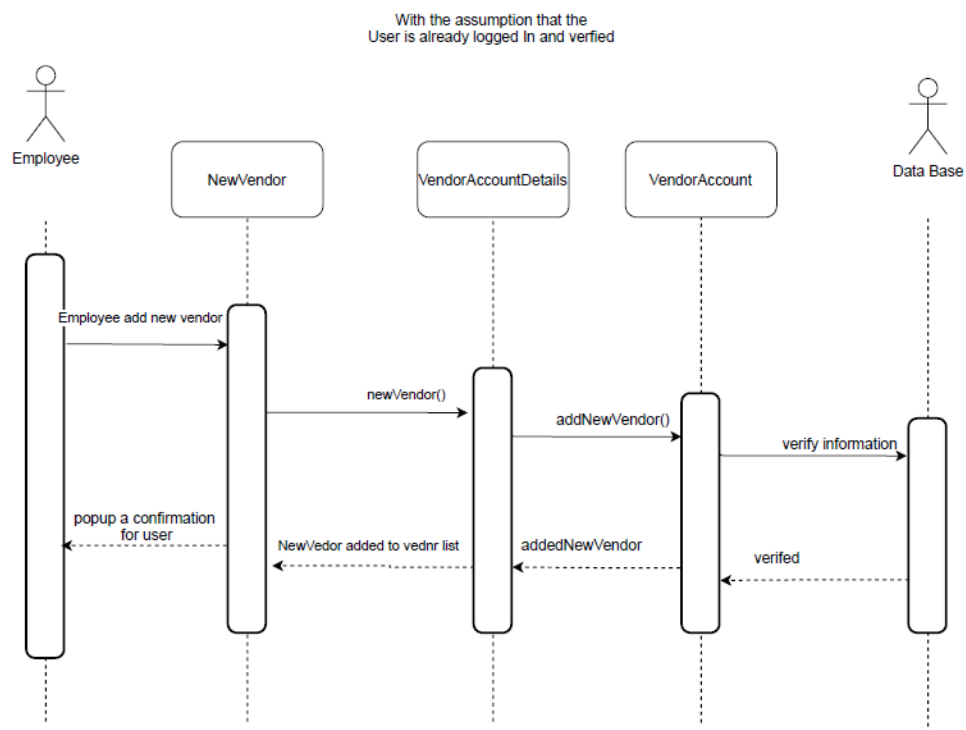
With comments -



Add New Vendor to the vendor list –

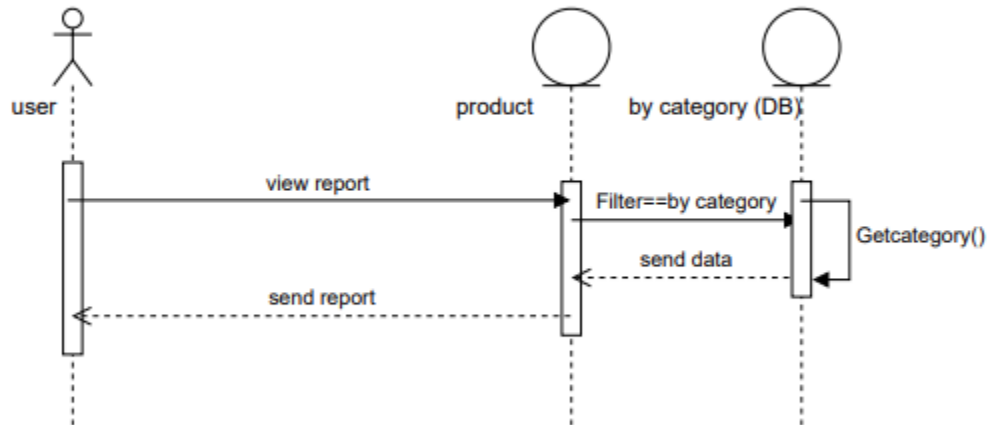
Once the user selects the vendor page from the main page and added the new vendor information into the text field, the user can click the “Add” button. Once click the system to override the info in the database.



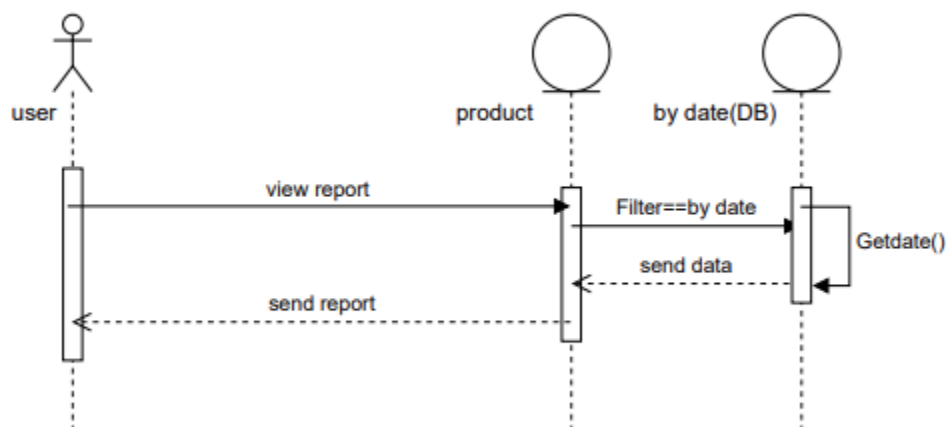
For the employee side –

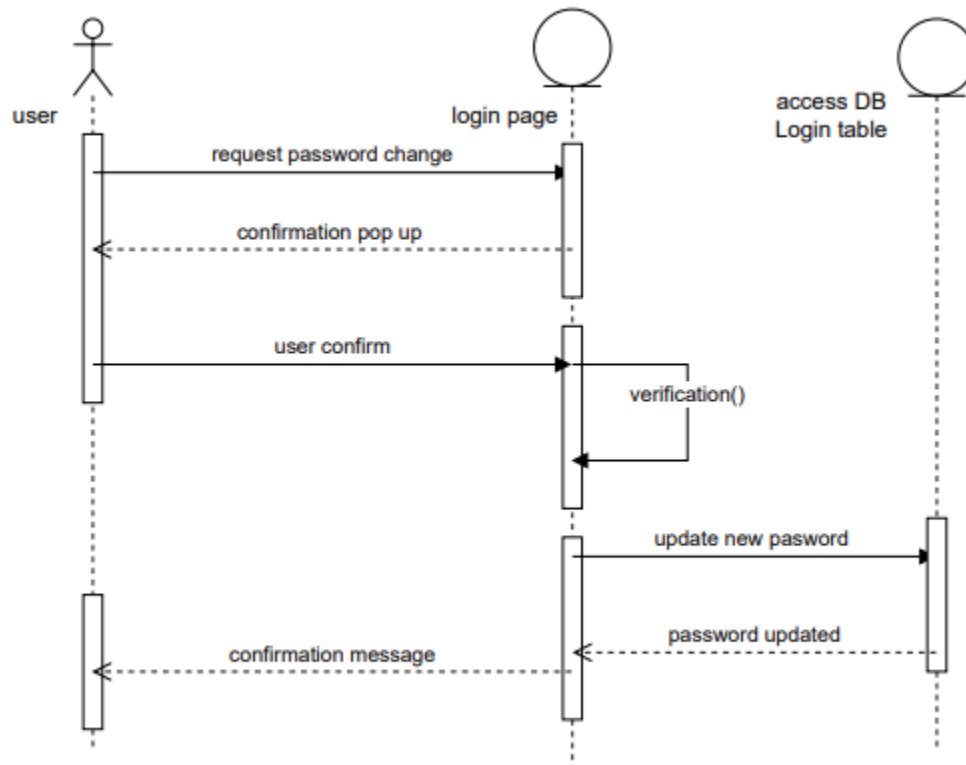
View Inventory (products) –

For this request, the user will have to be in the report page, and select the category product, then the user is required to click enter to initiate the process.



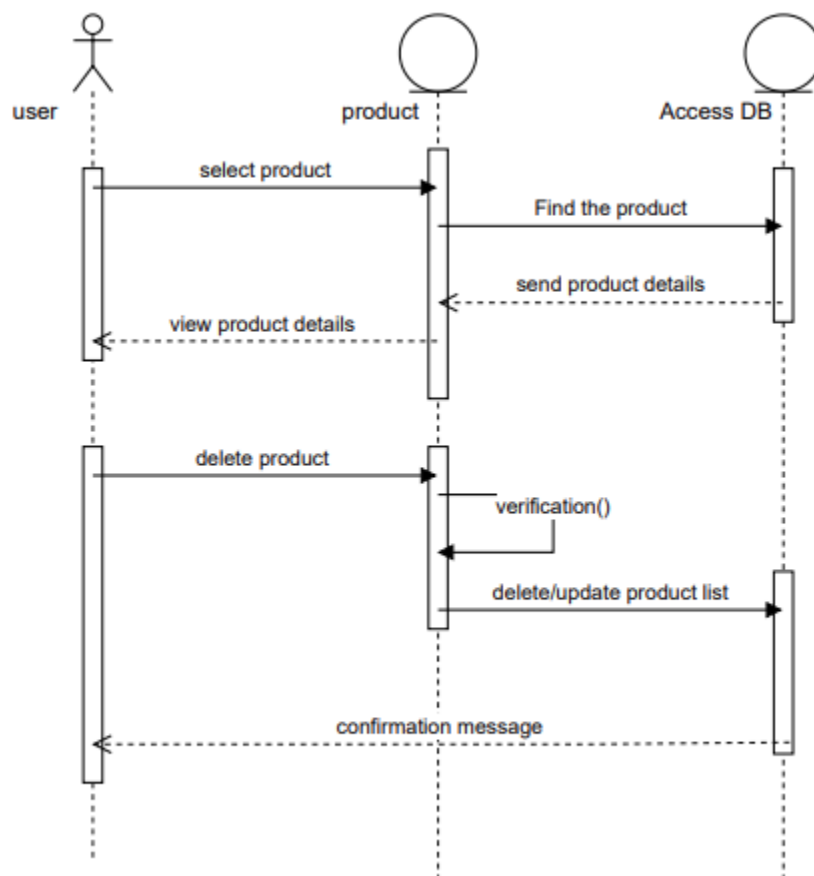
View product by order date –



User login password update -

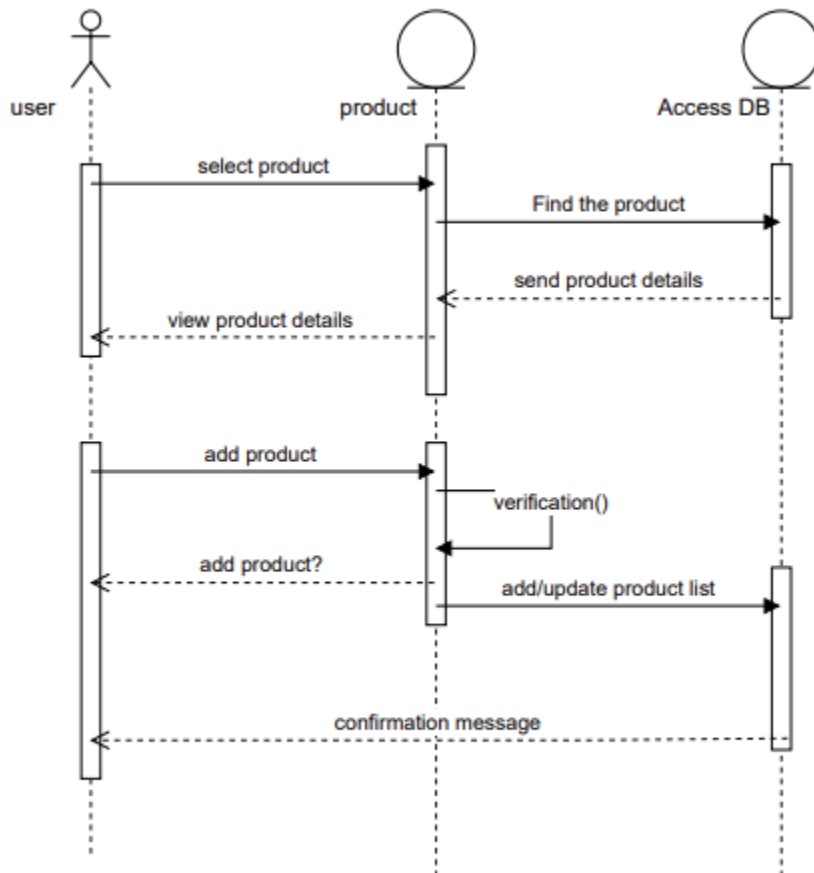
User to be able to Delete product. (With the assumption that the user is already logged in and validated and on the product page.

Once the user selects the product that wants to delete, the user can click on the “Delete” button to delete the product from the database.

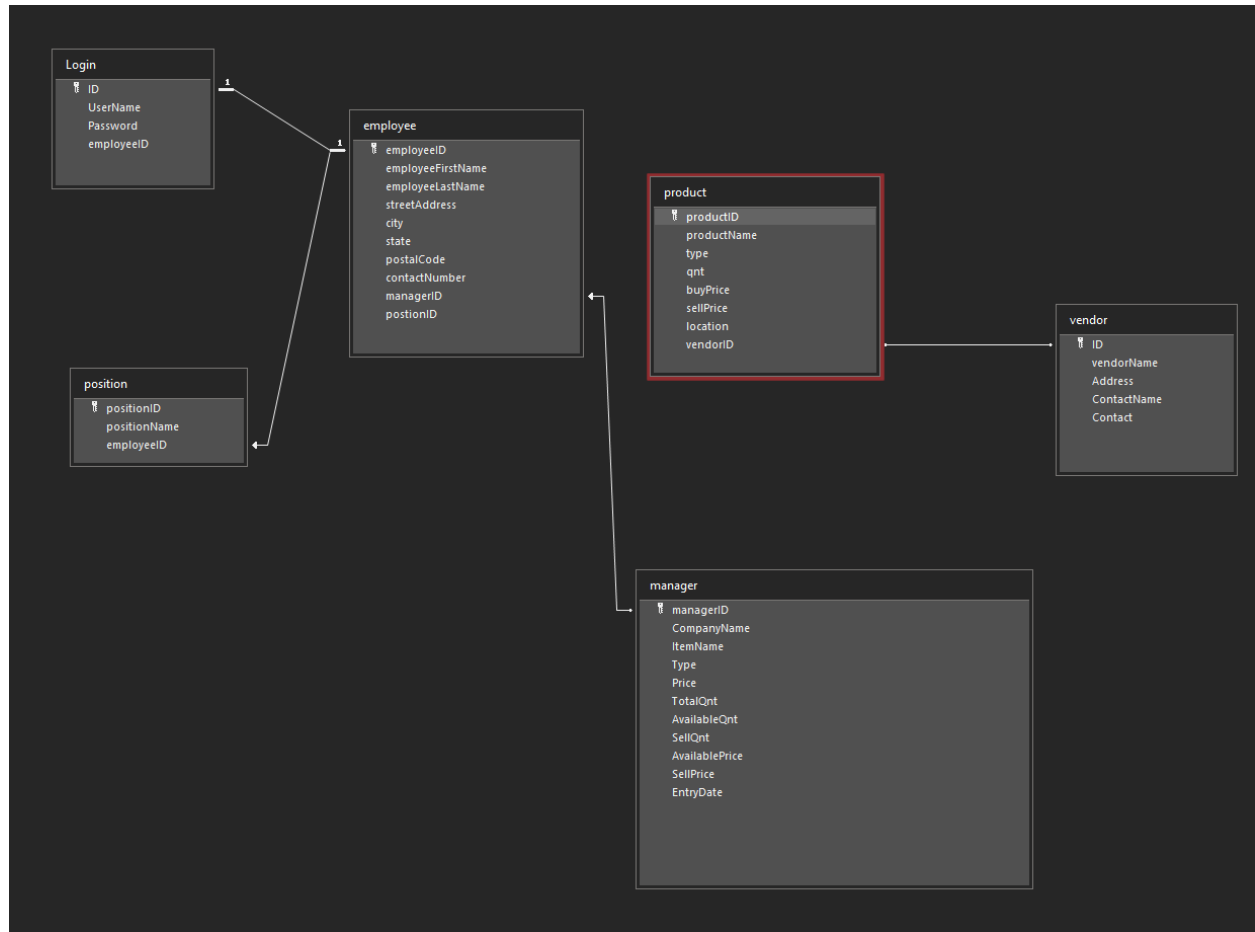


Add new Product-

Once the user is on the product page, the user can enter the new product details into the text files and click “Add” to add the entered data to the database.

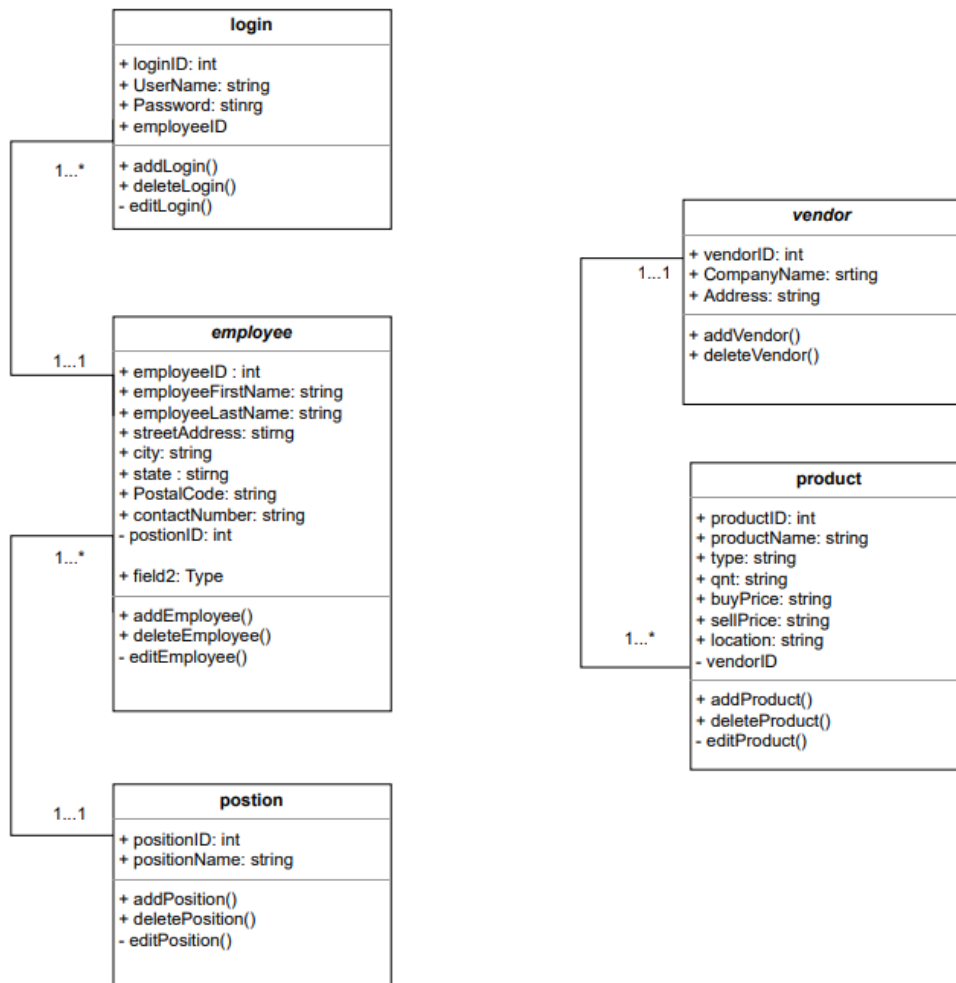


Database Class diagram – (current)



Section 8: Class Diagram

Inventory Management System Class diagram



Section 9: Typical Use Cases

User Login

Use case Name	<i>User Login</i>
Actors	Employee, manager, and Admin
Events	User launch the software User Enter Login info User click Enter
Entry Condition	Server should be running
Exit Condition	If user enter the correct username and password Confirmation popup to confirm logged in User is on the Main Page
Quality Requirements	User must enter correct username and password

View products (Products)

Use case Name	<i>Check Inventory</i>
Actors	Employee, manager, and Admin
Events	User browse to the products page User Enter the products information that wanted to check User click Enter System will search for the product data and then display it for the user
Entry Condition	Server should be running
Exit Condition	If user enter product information that exists in the database User will view the product details
Quality Requirements	User must enter product information that exists in the database

Add Vendor

Use case Name	<i>Add vendor</i>
Actors	manager, and Admin
Events	Manager/Admin need to be on the vendor page Manager/Admin need to enter the new vendor information Manager/Admin must click "Add" to add the new information
Entry Condition	Server should be running
Exit Condition	System will pop up a confirmation window that the new vendor is added
Quality Requirements	Manager must file all text fields to add new vendor

Add Employee

Use case Name	<i>Add employee</i>
----------------------	---------------------

Actors	manager, and Admin
Events	Manager/Admin need to be on the employee page Manager/Admin need to enter the new employee information Manager/Admin must click "Add" to add the new information
Entry Condition	Server should be running
Exit Condition	System will pop up a confirmation window that the new employee is added
Quality Requirements	Manager must file all text fields to add new employee information

Add Manager

Use case Name	<i>Add Manager</i>
Actors	Admin
Events	Admin need to be on the manager page Admin need to enter the new manager information Admin must click "Add" to add the new information
Entry Condition	Server should be running
Exit Condition	System will pop up a confirmation window that the new employee is added
Quality Requirements	Admin must file all text fields to add new manager information

Delete product

Use case Name	<i>Delete Product</i>
Actors	manager, and Admin
Events	Manager/Admin need to be on the product page Manager/Admin to location/select the product info Manager/Admin must click "Delete" to delete product information
Entry Condition	Server should be running
Exit Condition	System will pop up a confirmation window that the new employee is deleted
Quality Requirements	Manager must file all text fields to delete product from the database

Delete Employee

Use case Name	<i>Delete Employee</i>
Actors	manager, and Admin
Events	Manager/Admin need to be on the employee page Manager/Admin to location/select the employee info Manager/Admin must click "Delete" to delete the employee information
Entry Condition	Server should be running
Exit Condition	System will pop up a confirmation window that the new employee is deleted
Quality Requirements	Manager must file all text fields to delete employee from the database

Delete Vendor

Use case Name	<i>Delete Vendor</i>
Actors	manager, and Admin
Events	Manager/Admin need to be on the vendor page Manager/Admin to location/select vendor info Manager/Admin must click "Delete" to delete vendor information
Entry Condition	Server should be running
Exit Condition	System will pop up a confirmation window that vednorXX is deleted
Quality Requirements	Manager must file all text fields to delete vednorXX from the database

Section 10: Data Types

In this section, I took the customer and the order details pages since it is not a critical use case for the pilot program that the warehouse wants to test. It can be added in the future if needed.

Login –

- + loginID: int
- + UserName: string
- + Password: string
- + employeeID

Operation –

- + addLogin()
- + deleteLogin()
- editLogin()

Employee –

- + employeeID : int
- + employeeFirstName: string
- + employeeLastName: string
- + streetAddress: string
- + city: string
- + state : string
- + PostalCode: string
- + contactNumber: string
- positionID: int

Operation –

- + addEmployee()
- + deleteEmployee()
- editEmployee()

Position –

- + positionID: int
- + positionName: string

Operation –

- + addPosition()
- + deletePosition()
- editPosition()

Vendor –

- + vendorID: int
- + CompanyName: string
- + Address: string

Operation –

- + addVendor()
- + deleteVendor()

Product –

- + productID: int
- + productName: string
- + type: string + qnt: string
- + buyPrice: string
- + sellPrice: string
- + location: string
- vendorID

Operation –

- + addProduct()
- + deleteProduct()
- editProduct()

Manager-

- + managerID: int
- + managerFirstName(): sting
- + managerLastName(): string
- + streetAddress(): string
- + city(): string
- + state(): string
- + postalCode(): string
- + contactNumber(): string

Operation –

- + addManager()
- + deleteManager()
- editManager()

Section 11: System Architecture

Architecture Breakdown

Front End: -

Visual Studio 2019 – Front end IDE

- End user GUI design
- Database Modelling
- Control Design

Back End: -

Microsoft Access 356 – for DB

Tables -

- Login
- Employee
- Position
- Vendor
- Product

- Customer
- Order details

Hardware Minimum requirement

Processor – Intel Core i3, 2.0 GHz or higher

RAM – 2GB of RAM

Network Interface card

Hard Drive – 1 GB

Software Requirements

OS – Windows 10

Framework - .NET framework SP 3.0

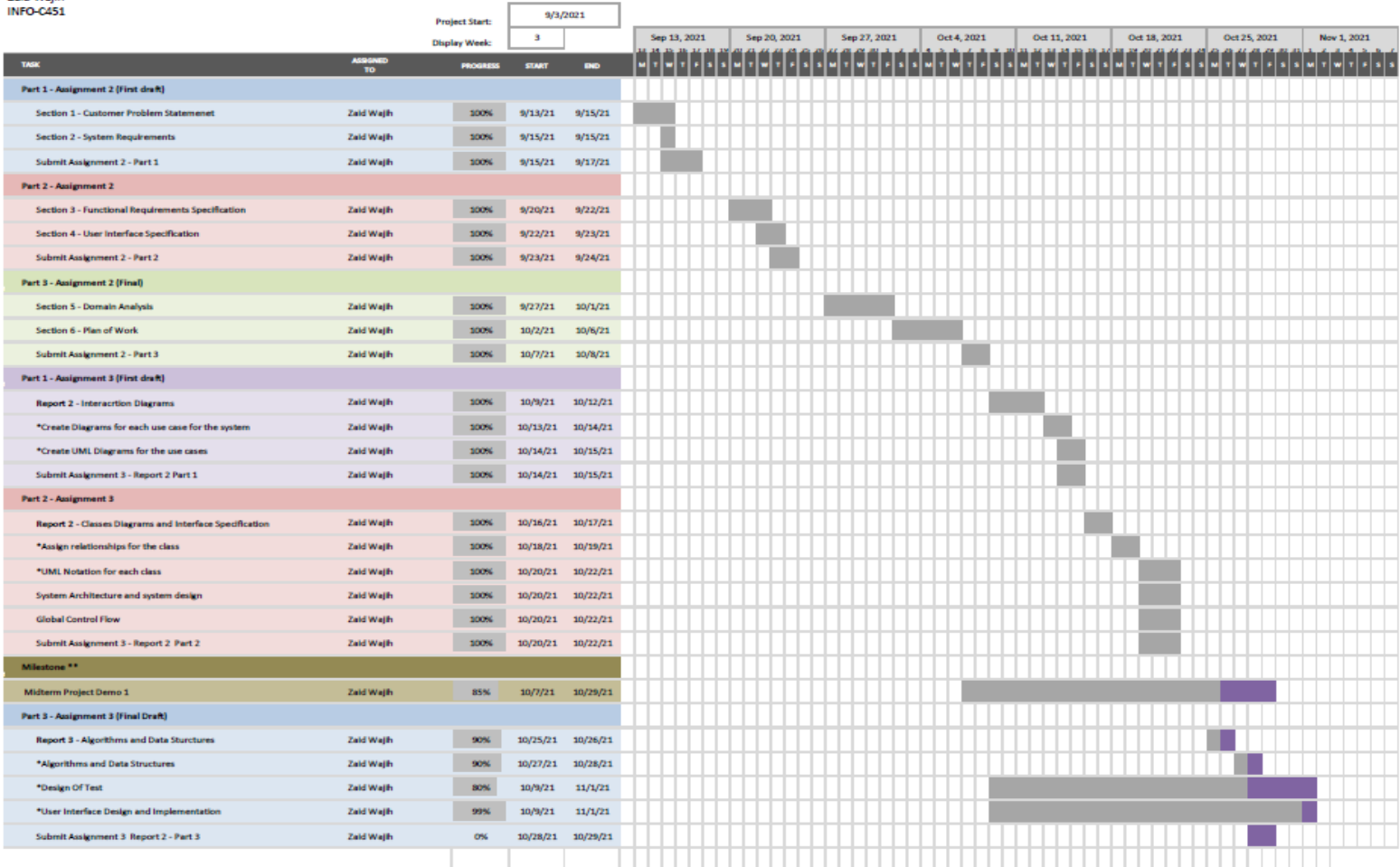
Microsoft Access 365

Section 12: Project Management

In this project, a system must be designed to support connectivity with a database management system that will store and keep all records for the system. The software application to be made consists of at least main business functions and meet all business requirements. The application will involve basic e-commerce activities, such as employee, manager login, browsing between products, and buy/sell activities. The entire system must be developed in (C#) to be easy to maintain and extend. Below is the project plan for each implementation section for the project.

Stock Management System Implementation

Zaid Wajih
INFO-C451



Section 13: User Interface Design

From the test cases that I had put in the code to help me validate any errors or bugs, I was buffing the code as I implemented the rest of the functions.

For example, I was running into window scaling issues on the Main page. Each child page would be different in size depending on the materials it displays, so I revised the main page as I was implementing the rest of the code.

I did not run into a significant change other than when I changed my use case, so I had to change some of the functions in the application. It was time-consuming when I did not have it.

Section 14: Design of Tests

When It comes to testing my software, I use various plugging the test code in my code to help me gather as much data as I can for any errors/ issues to help me prevent them from happening in the rest of the code. I need to pay close attention to the user's need because sometimes going around an issue will let you override an actual use case function. So for my humble use cases, I made sure to meet all requirements without taking or giving up on any functions.

I mainly used the “use case testing”, which I found the easiest way to test the C# language, and it doesn't need me to be savy with the language to troubleshoot issues. I help me test and validate all my pages from within the application.

Here is some of the code I used to test while implementing the functions.

```
catch (Exception ex)
{
    MessageBox.Show("Error " + ex);
    connection.Close();
}
```

```
MessageBox.Show(query);
```

Section 15: Project Management and Plan of Work

Use cases that have been Implemented and what are being tackled:

1. User Login

Implemented, and no issues.

2. Main Page

Implemented and no issues. Adding more sub child to it for other pages.

3. Products Page

Implemented and no issues. I am working on making the edit button work. But the rest of the buttons and views work fine with no issues.

4. Delete product

Implemented and functional.

5. Add products

Implemented and functional.

6. Vendor Page

Implemented with no issues. Working on a couple of functions like clearing the txt

7. Manager Page

Implemented with no issues.

8. View products

Implemented with no issues.

9. Add and Delete Vendors

Implemented with no issues

10. Employee page

Implemented with no issues

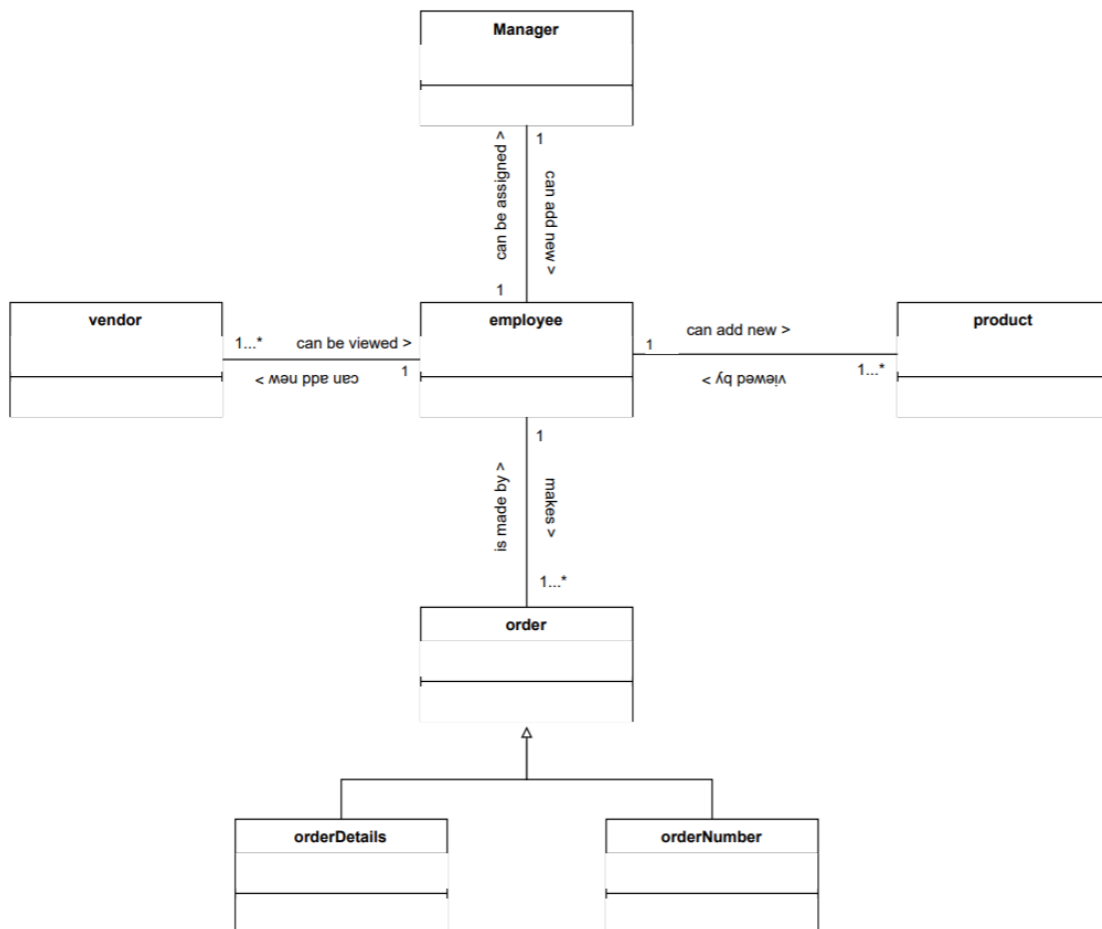
11. Add/Delete and Edit employee

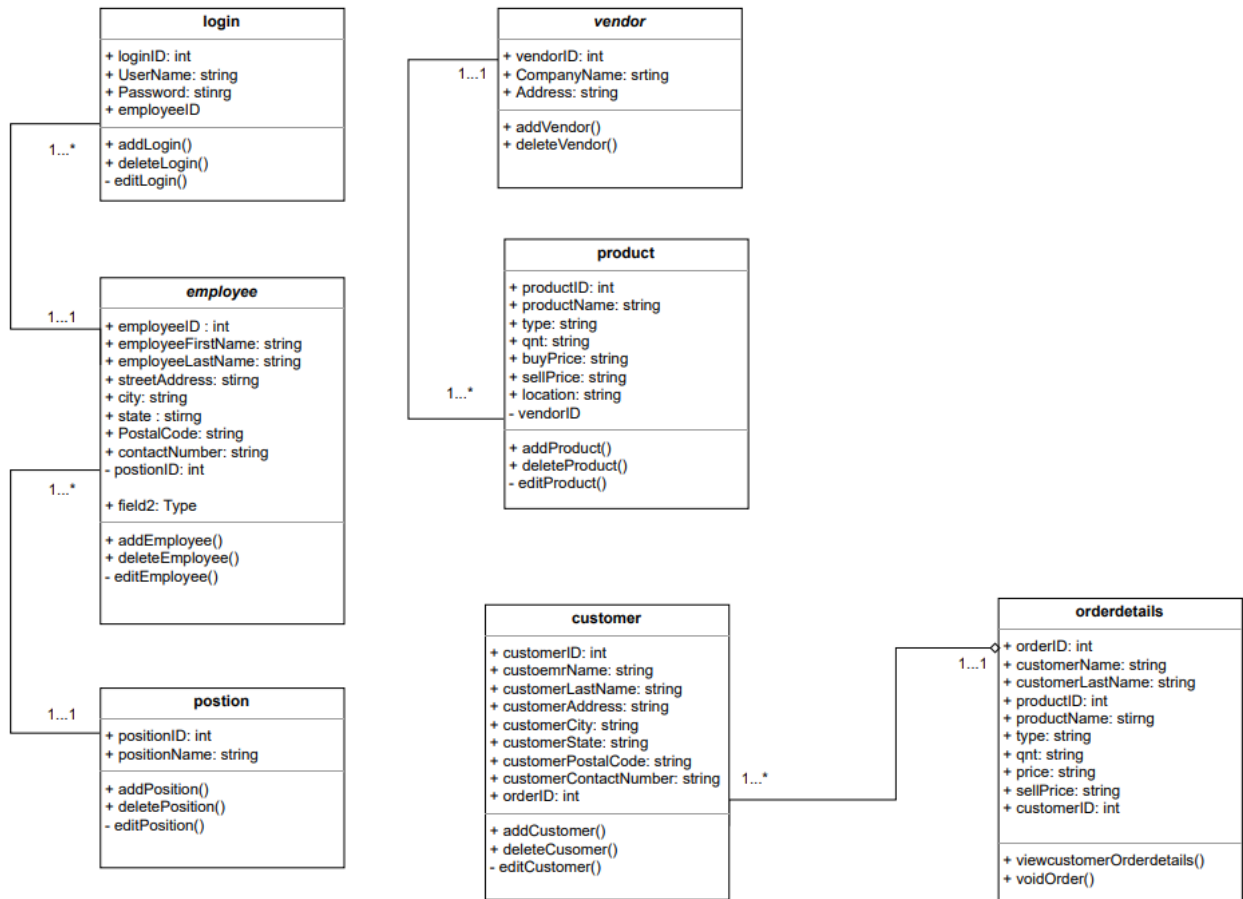
Implemented with no issues

Section 16: Report 3

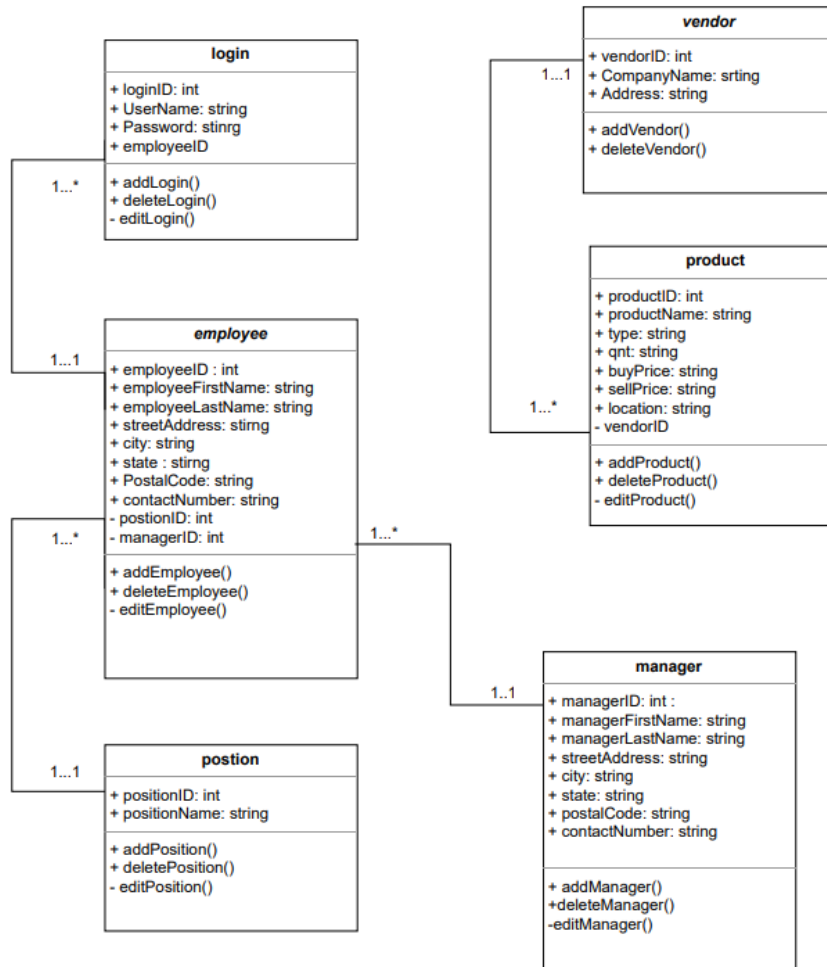
During the project, I have made some changes to the use cases based on the remaining timing for the project submission. This section will display the difference between the original project Domain and the final one that I will be working on to finalize the project.

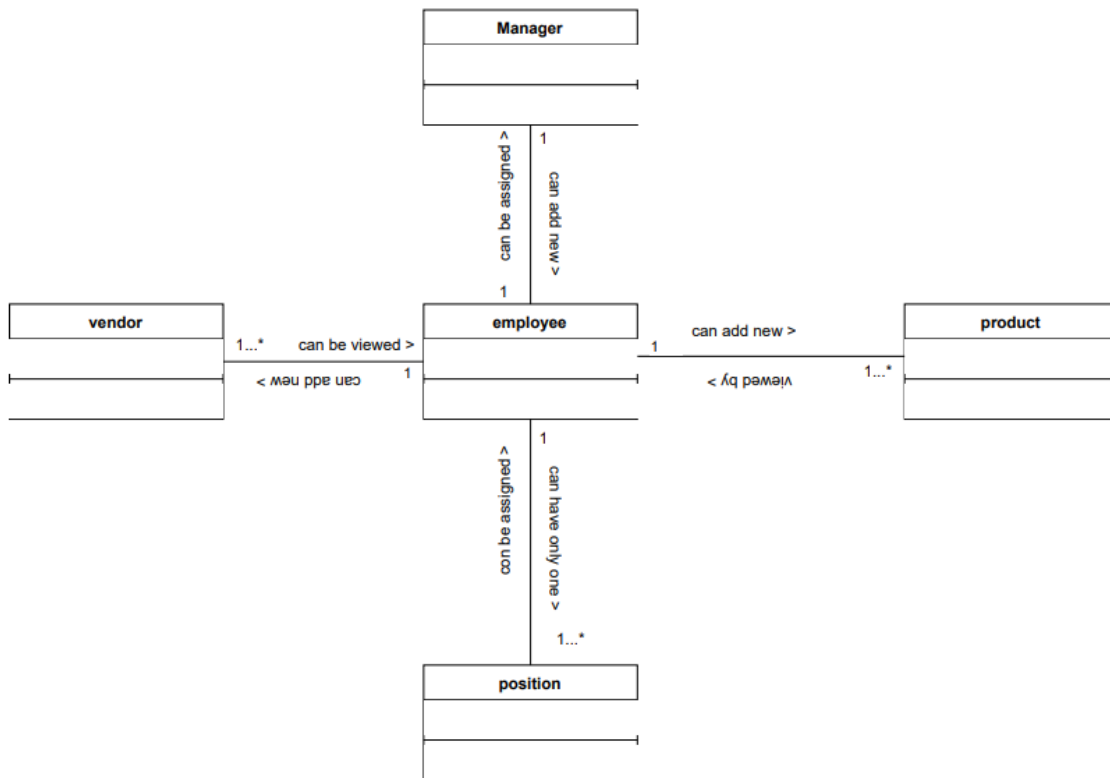
The original domain for the project





The current domain for the project





I intended to have an order page in the software to provide all order details. I had to eliminate this idea due to time limitations, but this feature can be implemented and added to the software in the future. Also, the customer page that was going to be related to the order page gives the customer information and keeps a record of all customers' orders.

Overall, the software allows users to log in and display the products, employees, vendors, managers, and position information. Users can Add, Edit, and Delete from within these pages the required field based on their needs. From now and the final submission to make my software complex, I am adding new functionality that automates some of the functions to save the employees and users some time.

References

I used the below website to get the templet for the project plan.

Smartsheet. (2021). *Powerful Project Planning Software*.

[https://www.smartsheet.com/s/project-](https://www.smartsheet.com/s/project-planning?s=1&c=3&m=461&a=356234702625&k=excel%20project%20plan%20templa)

planning?s=1&c=3&m=461&a=356234702625&k=excel%20project%20plan%20templa
te&mtp=e&adp=&net=g&dev=c&devm=&plc=&ds_rl=1286294&exp=&gclid=Cj0KCQ
iAy4eNBhCaARIsAFDVtI2si_sPHA3rA9Ue4_VrSg0c28IJDyMvefa761nLQFgh_8SOD
umvbJUaAkJFEALw_wcB&gclsrc=aw.ds