**Computing Studies & Information Systems**

"Software Engineering"

Winter 2022: CSIS 3275 – 002

**Final Project Report: “Clothing Stock Management System”**

Graphical user interface

Description automatically generated**Team Members:**

Name \_ *Pamela G. Lemus Villagrana*\_\_\_\_ Student # \_\_*300344812*\_\_\_\_

Name \_\_\_*Karamjeet Singh Jandu*\_\_\_\_\_\_\_\_ Student # \_\_*300305341*\_\_\_\_

Name \_\_\_*Minh Bui*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student # \_\_*300345000*\_\_\_\_\_

Name \_\_\_ *Jasmine Kaur* \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student # \_\_*300345933*\_\_\_\_\_

Name \_\_\_ *Jaspal Singh*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student # \_\_ *300345938*\_\_\_\_\_

**Statement of Work - Project Writeup**

**Introduction – Project Description:**

Our system will be designed to manage stock in a warehouse, focus on clothing storage. Development will be used by managers and operators to record every new item in and out of the storage. The system to manage the inventory particularly focus on the clothes storage. The system will be used by store managers and employees to record each new item in and out of the storage. Managers and employees will be differentiated by the employee ID and they will have different responsibilities. Users can update the quantity of an item and add new category.

Technologies we have decided to develop our system are MS SQL server to save detail of products and employee information, for structure of the system we are using HTML and Bootstrap framework. For frontend development we are using ReactJS and for backend we are using Dot Net to generate APIs.

The user interface shows the customer journey, and it includes the following:

* **Login:** the user login with their employee id and pin.
* **Home page:** the menu options for transactions, stock in, stock out, current stock and user profile.
* **Transaction:** the stock records which will record each transaction, with the type either stock out or stock in.
* **Stock In:** the user input form with fields for product id, product name, quantity, and category for adding item to the stock.
* **Stock out:** the user input form with fields for product ID, supplier and quantity to get stock out.
* **Current stock:** the record of stock available currently in the storage. In case of a manager or an operator found mismatch of stock quantity in the current stock and data from system, they can keep track of each item and the user that confirm every movement. Manager or operator can search product by identification number, category, supplier or customer, in order to check the current stock quantity and control the inventory. Manager or operator can then adjust the correct amount of stock and leave a comment.

The system will be connected to a central database that includes information of product identification, assign a unique number to each item; a category such as Pants, Shirts, Sweaters, Coats, Jackets and Others; supplier, identifying the origin of the item; customer that requests the item; stock record, to maintain track of each item; and user’s list, including username and password or pin in order to allow access to the system.

**Project Plan – Statement of Work:**

As a team, we have defined all required tasks in order to complete the software development. We have identified the duration of each task, start and end date planned, as well as dependencies between all. We developed a Gantt chart, and a Network Diagram on Microsoft Project:

Graphical user interface, table, Excel

Description automatically generated

Chart

Description automatically generated

Graphical user interface

Description automatically generated

Box and whisker chart

Description automatically generated with low confidence

Graphical user interface, application

Description automatically generated

**Use Case Diagram:** Scenarios of system usage described from primary actors perspective.Diagram

Description automatically generated

**User Interface Design**

**Login Screen:** the user login with their employee ID and pin.

Graphical user interface

Description automatically generated

**Stock Records Screen:** once authentication is successful, user could access to all the records on the system. A Menu to go to other screens or search for specific item will be display at the top of the screen.

Table

Description automatically generated

**Stock In Screen:** user must enter all required fields in order to introduce a new item on the system. More product details will be display, according with entered Product ID (first field). If the information is correct, the system will generate a Transaction ID, showing the corresponding message to the user.

Graphical user interface, text, application, email

Description automatically generated

**Stock Out Screen:** user introduce required fields, and take out a specific item from the system.Graphical user interface, text, application, email

Description automatically generated

**Current Stock Screen:** this screen displays the master table of the information stored through the system, allowing user to search any item.

Table

Description automatically generated

**Suppliers and Customers Screens:** this tables are directories that allow users to look for the contact information of all Suppliers or Customers stored in the system.

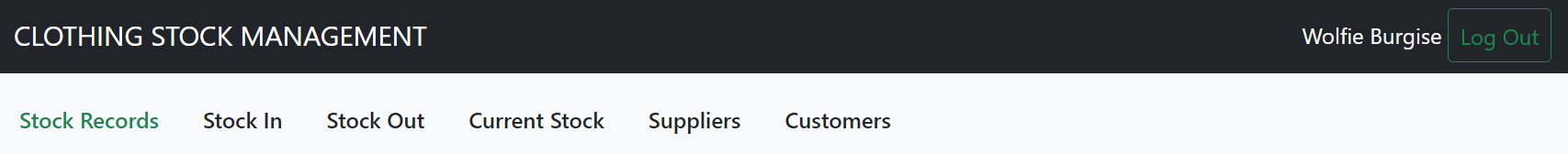
Table

Description automatically generated

Table

Description automatically generated

At the top of the screens, we have the name of user that has an active session, which allows user to Logout at any time:



The original user interface design was developed in Adobe, and you could review the flow on the next link:

<https://xd.adobe.com/view/00188da1-acbf-4744-af18-e8d35f734e84-554b/?fullscreen>

**Data Design**

In order to structure the database needed for the software, we have decided to use EER (Enhanced Entity-Relationship) Modelling, and Entity Diagram. Definition of entities, relationships, tables, fields, and data type are described below.

* **Database Relational Schema:**
* System\_User (EmployeeID, FName, LName, Email, Role, PIN, Phone)
* Product (ProductID, ProductName, Description, Size, Category, Quantity)
* Supplier (SupplierID, SupplierName, Email, Phone)
* Customer (CustomerID, CustomerName, Location, Email, Phone)
* User\_StockIN/OUT\_Product (TransactionNumber, TransactionId, **ProductID**, Quantity, Date, **EmployeeID**, Comments, Type, **SupplierID, CustomerID**)
* **EER Modeling Diagram:**

**System User**

**Customer**

.

**1**

**1**

**Product**

Have

Stock IN/OUT

**M**

**1**

**Supplier**

.

**M**

**1**

Have

* **Entity Diagram:**

Diagram

Description automatically generated

* **Data Types and Constraints for creating all tables, script made on MS SQL Server:**

/\*Once Database is created please comment the next line of code.\*/

CREATE DATABASE ClothingStock\_DB;

USE ClothingStock\_DB;

CREATE TABLE "System\_User" (

EmployeeID INT PRIMARY KEY,

FName VARCHAR (20) NOT NULL,

LName VARCHAR (20),

Email VARCHAR (50) UNIQUE NOT NULL,

Role VARCHAR (10) CHECK (Role='Manager' OR Role='Operator'),

PIN INT UNIQUE NOT NULL,

Phone VARCHAR (50) NOT NULL );

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

ProductName VARCHAR (100) NOT NULL,

Description VARCHAR (1000) NOT NULL,

Size VARCHAR (20) CHECK (Size='XS' OR Size ='S' OR Size ='M' OR Size ='L' OR Size ='XL') NOT NULL,

Category VARCHAR (20) CHECK (Category ='Pants' OR Category ='Shirts' OR Category ='Sweaters' OR Category ='Coats' OR Category ='Jackets' OR Category ='Others') NOT NULL,

Quantity INT NOT NULL );

CREATE TABLE Supplier (

SupplierID INT PRIMARY KEY,

SupplierName VARCHAR (50) NOT NULL ,

Email VARCHAR (50) UNIQUE NOT NULL,

Phone VARCHAR (50) NOT NULL );

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY ,

CustomerName VARCHAR (50) NOT NULL ,

Location VARCHAR (200) NOT NULL,

Email VARCHAR (50) UNIQUE NOT NULL,

Phone VARCHAR (50) NOT NULL );

CREATE TABLE User\_StockIN\_OUT\_Product (

TransactionNumber INT PRIMARY KEY IDENTITY(1,1),

TransactionId VARCHAR(MAX) NOT NULL,

ProductID INT NOT NULL,

Quantity INT NOT NULL,

Date DATETIME NOT NULL,

EmployeeID INT NOT NULL,

Comments VARCHAR (50),

Type VARCHAR (20) CHECK (Type ='IN' OR Type ='OUT' OR Type ='ADJUST') NOT NULL,

SupplierID INT,

CustomerID INT,

CONSTRAINT Stock\_Employee FOREIGN KEY (EmployeeID) REFERENCES "System\_User" (EmployeeID) ON DELETE CASCADE,

CONSTRAINT Stock\_Product FOREIGN KEY (ProductID) REFERENCES Product (ProductID) ON DELETE CASCADE,

CONSTRAINT Stock\_Supplier FOREIGN KEY (SupplierID) REFERENCES Supplier (SupplierID) ON DELETE CASCADE,

CONSTRAINT Stock\_Customer FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID) ON DELETE CASCADE );

* **MS SQL Server Script for populating all tables with massive information:**

USE ClothingStock\_DB;

declare @DataPath varchar(1000)

--Set the path to the data folder in your system. Ex: 'C:\Data'

set @DataPath = 'C:\Data'

--Insert product data

declare @sql\_product varchar(1000)

set @sql\_product = 'Bulk insert Product

from ''' + @DataPath + 'Product.psv ' +

'''with

(

fieldterminator = ''|'',

rowterminator = ''\n''

)'

exec(@sql\_product)

--Insert System User data

declare @sql\_system\_user varchar(1000)

set @sql\_system\_user = 'Bulk insert "System\_User"

from ''' + @DataPath + 'System\_User.psv ' +

'''with

(

fieldterminator = ''|'',

rowterminator = ''\n''

)'

exec(@sql\_system\_user)

--Insert Supplier data

declare @sql\_supplier varchar(1000)

set @sql\_supplier = 'Bulk insert "Supplier"

from ''' + @DataPath + 'Supplier.psv ' +

'''with

(

fieldterminator = ''|'',

rowterminator = ''\n''

)'

exec(@sql\_supplier)

--Insert Customer data

declare @sql\_customer varchar(1000)

set @sql\_customer = 'Bulk insert "Customer"

from ''' + @DataPath + 'Customer.psv ' +

'''with

(

fieldterminator = ''|'',

rowterminator = ''\n''

)'

exec(@sql\_customer)

**Functionalities of Software**

**Actual running of Software System**

* **Flowchart:** description of every available functionality of the software.

Diagram

Description automatically generated

* **Basis Path Testing - Flow Graph**: diagram that shows the number of independent paths in the software, describing every set of processing statements.

Diagram

Description automatically generated

* **Independent paths / Test Cases:** in order to demonstrate every functionality of the software, we have described 23 possible paths, including screen shots showing corresponding results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Path** | **Steps** | **Test Case Description** | **Result** | **Comments** | **Pass/Fail** |
| **1** | 1-3 | User do not enter credentials | As Expected | Error message display | Pass |
| **2** | 1-2-3 | User enter wrong credentials | As Expected | Error message display | Pass |
| **3** | 1-2-4-27 | User enter correct credentials, have access to functionalities and log out (no changes) | As Expected | User do not made any modifications on stock | Pass |
| **4** | 1-2-4-5-12-13-14-12 | Correct credentials, access to functionalities, STOCK IN transaction, **user do not enter product ID or it is no valid** | As Expected | If user enter an ID non existing in database corresponding message is displayed, if do not enter any number, software requires corresponding fields. | Pass |
| **5** | 1-2-4-5-12-15-16-13-14-12 | Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, **user do not enter quantity** | As Expected | Error message display requiring corresponding fields | Pass |
| **6** | 1-2-4-5-12-15-16-17-18-19-20-21-23-24-26-4 | Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, **user do not select supplier,** "Stock Records" updated with **ADJUST** type, "Current Stock" updated (quantity), **user remain in software** | As Expected | Updated quantity is shown in STOCK IN screen, and successfully updated in CURRENT STOCK | Pass |
| **7** | 1-2-4-5-12-15-16-17-18-19-20-21-23-24-26-27 | Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, **user do not select supplier**, "Stock Records" updated with **ADJUST** type, "Current Stock" updated (quantity), **user log out** | As Expected | Pass |
| **8** | 1-2-4-5-12-15-16-17-18-19-20-22-23-24-26-4 | Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, **user select supplier**, "Stock Records" updated with **STOCK IN** type, "Current Stock" updated (quantity), **user remain in software** | As Expected | Pass |
| **9** | 1-2-4-5-12-15-16-17-18-19-20-22-23-24-26-27 | Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, **user select supplier**, "Stock Records" updated with **STOCK IN** type, "Current Stock" updated (quantity), **user log out** | As Expected | Pass |
| **10** | 1-2-4-6-12-13-14-12 | Correct credentials, access to functionalities, STOCK OUT transaction, **user do not enter product ID or it is no valid** | As Expected | If user enter an ID non existing in database corresponding message is displayed, if do not enter any number, software requires corresponding fields. | Pass |
| **11** | 1-2-4-6-12-15-16-13-14-12 | Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, **user do not enter quantity** | As Expected | Error message display requiring corresponding fields | Pass |
| **12** | 1-2-4-6-12-15-16-17-18-19-20-21-23-25-26-4 | Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, **user do not select supplier,** "Stock Records" updated with **ADJUST** type, "Current Stock" updated (quantity), **user remain in software** | As Expected | Updated quantity is shown in STOCK OUT screen, and successfully updated in CURRENT STOCK, quantity in "Stock Records" is shown with minus symbol | Pass |
| **13** | 1-2-4-6-12-15-16-17-18-19-20-21-23-25-26-27 | Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, **user do not select supplier**, "Stock Records" updated with **ADJUST** type, "Current Stock" updated (quantity), **user log out** | As Expected | Pass |
| **14** | 1-2-4-6-12-15-16-17-18-19-20-22-23-25-26-4 | Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, **user select supplier**, "Stock Records" updated with **STOCK OUT** type, "Current Stock" updated (quantity), **user remain in software** | As Expected | Pass |
| **15** | 1-2-4-6-12-15-16-17-18-19-20-22-23-25-26-27 | Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, **user select supplier**, "Stock Records" updated with **STOCK OUT** type, "Current Stock" updated (quantity), **user log out** | As Expected | Pass |
| **16** | 1-2-4-7-11-26-4 | Correct credentials, access to functionalities, **STOCK RECORDS** transaction, search option, user remain in software | As Expected | User type product ID, description or a key word for any column of the corresponding transaction, all possible matches starting with the keyword are shown successfully | Pass |
| **17** | 1-2-4-7-11-26-27 | Correct credentials, access to functionalities, **STOCK RECORDS** transaction, search option, user log out | As Expected | Pass |
| **18** | 1-2-4-8-11-26-4 | Correct credentials, access to functionalities, **CURRENT STOCK** transaction, search option, user remain in software | As Expected | Pass |
| **19** | 1-2-4-8-11-26-27 | Correct credentials, access to functionalities, **CURRENT STOCK** transaction, search option, user log out | As Expected | Pass |
| **20** | 1-2-4-9-11-26-4 | Correct credentials, access to functionalities, **SUPPLIERS** transaction, search option, user remain in software | As Expected | Pass |
| **21** | 1-2-4-9-11-26-27 | Correct credentials, access to functionalities, **SUPPLIERS** transaction, search option, user log out | As Expected | Pass |
| **22** | 1-2-4-10-11-26-4 | Correct credentials, access to functionalities, **CUSTOMERS** transaction, search option, user remain in software | As Expected | Pass |
| **23** | 1-2-4-10-11-26-27 | Correct credentials, access to functionalities, **CUSTOMERS** transaction, search option, user log out | As Expected | Pass |

**LOG IN Functionalities:**

**Path 1:** 1-3 “User do not enter credentials”.

Graphical user interface, application

Description automatically generated

**Path 2:** 1-2-3 “User enter wrong credentials”.

Graphical user interface, application

Description automatically generated

**Path 3:** 1-2-4-27 “User enter correct credentials, have access to functionalities and log out”.

Table

Description automatically generated

**STOCK IN Functionalities:**

**Path 4:** 1-2-4-5-12-13-14-12 “Correct credentials, access to functionalities, STOCK IN transaction, user do not enter product ID or it is no valid”.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Path 5:** 1-2-4-5-12-15-16-13-14-12 “Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, user do not enter quantity”.

Graphical user interface, text, application

Description automatically generated

**Path 6:** 1-2-4-5-12-15-16-17-18-19-20-21-23-24-26-4 “Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, user do not select supplier, "Stock Records" updated with ADJUST type, "Current Stock" updated (quantity), user remain in software”.

Graphical user interface, text, email

Description automatically generated

Text

Description automatically generated with medium confidence

**Path 7:** 1-2-4-5-12-15-16-17-18-19-20-21-23-24-26-27 “Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, user do not select supplier, "Stock Records" updated with ADJUST type, "Current Stock" updated (quantity), user log out”.

**Path 8:** 1-2-4-5-12-15-16-17-18-19-20-22-23-24-26-4 “Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, user select supplier, "Stock Records" updated with STOCK IN type, "Current Stock" updated (quantity), user remain in software”.

Graphical user interface, text, email

Description automatically generated

Table

Description automatically generated with low confidence

**Path 9:** 1-2-4-5-12-15-16-17-18-19-20-22-23-24-26-27 “Correct credentials, access to functionalities, STOCK IN transaction, correct product ID, quantity, user select supplier, "Stock Records" updated with STOCK IN type, "Current Stock" updated (quantity), user log out”.

**STOCK OUT Functionalities:**

**Path 10:** 1-2-4-6-12-13-14-12 “Correct credentials, access to functionalities, STOCK OUT transaction, user do not enter product ID or it is no valid”.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Path 11:** 1-2-4-6-12-15-16-13-14-12 “Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, user do not enter quantity”.

Graphical user interface, text, application

Description automatically generated

**Path 12:** 1-2-4-6-12-15-16-17-18-19-20-21-23-25-26-4 “Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, user do not select supplier, "Stock Records" updated with ADJUST type, "Current Stock" updated (quantity), user remain in software”.

Graphical user interface, text, email, website

Description automatically generated

Text

Description automatically generated with medium confidence

**Path 13:** 1-2-4-6-12-15-16-17-18-19-20-21-23-25-26-27 “Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, user do not select supplier, "Stock Records" updated with ADJUST type, "Current Stock" updated (quantity), user log out”.

**Path 14:** 1-2-4-6-12-15-16-17-18-19-20-22-23-25-26-4 “Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, user select supplier, "Stock Records" updated with STOCK OUT type, "Current Stock" updated (quantity), user remain in software”.

Graphical user interface, text, email

Description automatically generated

A picture containing table

Description automatically generated

**Path 15:** 1-2-4-6-12-15-16-17-18-19-20-22-23-25-26-27 “Correct credentials, access to functionalities, STOCK OUT transaction, correct product ID, quantity, user select supplier, "Stock Records" updated with STOCK OUT type, "Current Stock" updated (quantity), user log out”.

**SEARCH Functionality:**

**Path 16:** 1-2-4-7-11-26-4 “Correct credentials, access to functionalities, **STOCK RECORDS** transaction, search option, user remain in software”.

Table

Description automatically generated

**Path 17:** 1-2-4-7-11-26-27 “Correct credentials, access to functionalities, **STOCK RECORDS** transaction, search option, user log out”.

**Path 18:** 1-2-4-8-11-26-4 “Correct credentials, access to functionalities, **CURRENT STOCK** transaction, search option, user remain in software”.

Graphical user interface, text, email

Description automatically generated

**Path 19:** 1-2-4-8-11-26-27 “Correct credentials, access to functionalities, **CURRENT STOCK** transaction, search option, user log out”.

**Path 20:** 1-2-4-9-11-26-4 “Correct credentials, access to functionalities, **SUPPLIERS** transaction, search option, user remain in software”.

Graphical user interface, text, website

Description automatically generated

**Path 21:** 1-2-4-9-11-26-27 “Correct credentials, access to functionalities, **SUPPLIERS** transaction, search option, user log out”.

**Path 22:** 1-2-4-10-11-26-4 “Correct credentials, access to functionalities, **CUSTOMERS** transaction, search option, user remain in software”.

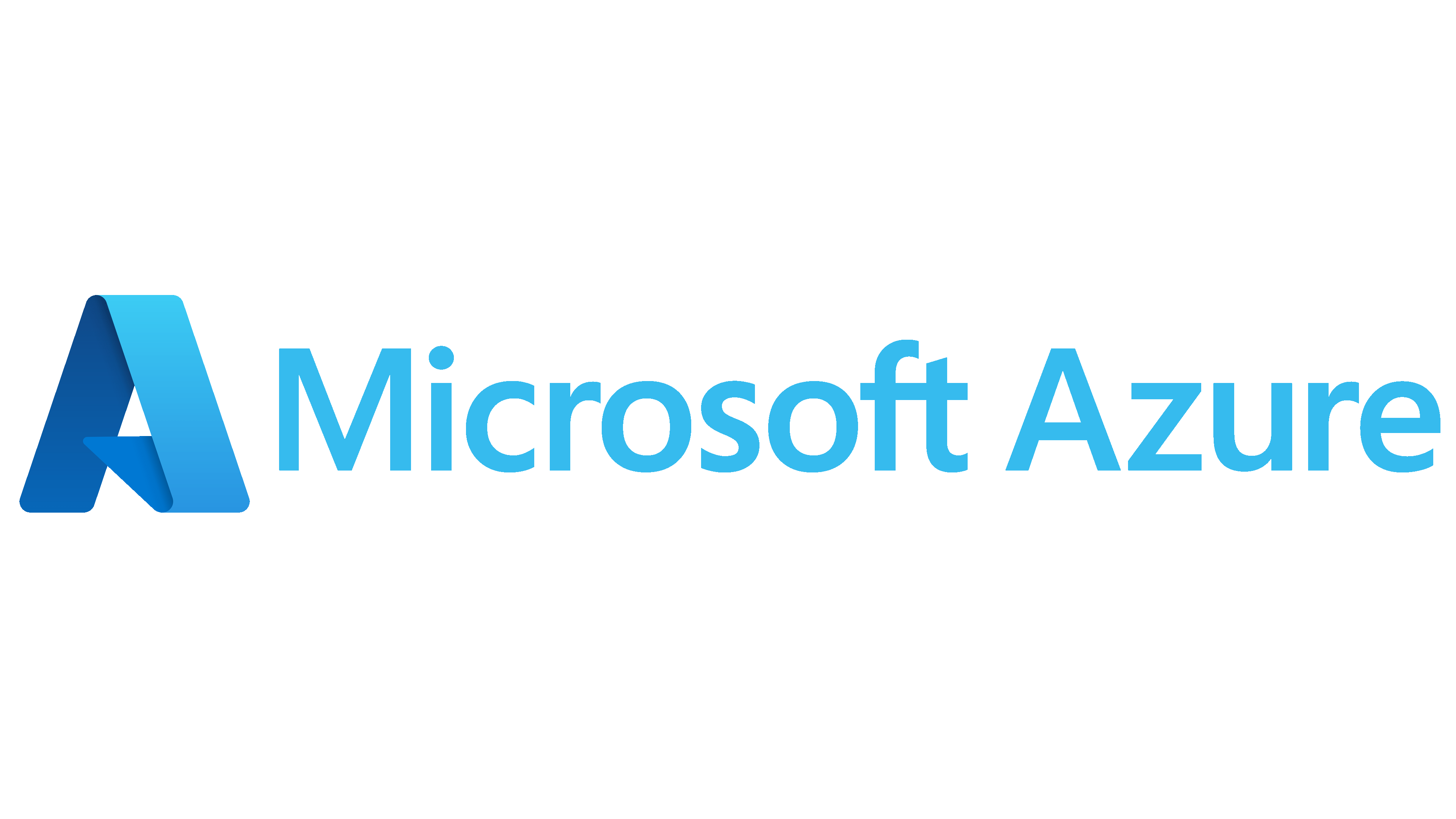
A screenshot of a computer

Description automatically generated

**Path 23:** 1-2-4-10-11-26-27 “Correct credentials, access to functionalities, **CUSTOMERS** transaction, search option, user log out”.

**Suitable use of features of Programming Language – Objects**

The project as a whole is a client-server web application as they don't require any installation on frontend device and only require a web browser. It is suitable for warehouse management as this job requires moving around frequently.



* Logo

  Description automatically generated with low confidence**Backend** is a web service built with ASP.NET Core using Visual Studio as deployment onto Azure is made simple since both tools are developed by Microsoft.



* **Frontend** web client used React as it provides flexibility, good performance and reusable components.

**Reflections and Suggestions**

**Issues/Problems encountered during design phase and development of System:**

* Each team member was only familiar with some of the chosen technologies and had to do their research as they work on the project.
* Team members have different development environments and have minor troubles with setting up the project.
* Backend web service could not be hosted on the lab computers at Douglas College, possibly due to security reason.
* Original database design should be modified a couple of times, due to new definitions about relationship between product and supplier/customer.

**Suggestion on how project can be improved/completed:**

* How product could be handled if come via container: it will have to be opened up and all the products inside will be checked. If the container is not to be opened as it being moved into the warehouse, the whole container will be treated as one product with details of its content put into the product's description.
* In case customers wanted to keep track of their product's location in the warehouse, the project can be extended to have a Storage Area entity. Product will have an attribute to track where they are stored and another attribute for how much space they occupied. Area can use these attributes to show how much space they have left to help warehouse manager with their management.
* New functionalities can be added, such as recover the PIN number to access to the system, implementing all the required security measures. Other extra functionality could be exporting some kind of reports using filters, for example transactions by employee name, date, specific product, among others. Related with look and feel of the system, we can add an image for the user profile at the top of the screen.