**Computing Studies & Information Systems**

"Software Engineering"

Winter 2022: CSIS 3275 – 002

**Project Proposal: “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*\_\_\_\_\_

**Introduction:**

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 Summary:**

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 actor and secondary actors(supplier/customer).

Diagram

Description automatically generated

**User Interface Design:**

**Login Screen:** the user login with their employee ID and pin.Graphical user interface, application

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 all screens.Graphical user interface, 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, application

Description automatically generated

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

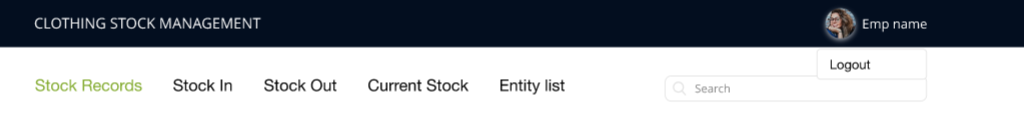
Description automatically generated

**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

**Entity List Screen:** this table is the directory that allow users to look for the contact information of all Suppliers or Customers stored in the system.Graphical user interface, table

Description automatically generated

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

The complete 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, the 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, **ProductID**)
* Customer (CustomerID, CustomerName, Location, Email, Phone, **ProductID**)
* User\_StockIN/OUT\_Product (**EmployeeID, ProductID,** Quantity, Date, TransactionNumber, Comments, Type)

Table

Description automatically generated

* EER Modeling Diagram:

**System User**

**Customer**

.

**1**

**1**

**Product**

Have

Stock IN/OUT

**M**

**1**

**Supplier**

.

**M**

**1**

Have

* Data Types and Constraints, scripts 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 (20) NOT NULL,

Description VARCHAR (20) 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 (20) NOT NULL ,

Email VARCHAR (50) UNIQUE NOT NULL,

Phone VARCHAR (50) NOT NULL,

ProductID INT NOT NULL,

CONSTRAINT SupplierFK FOREIGN KEY (ProductID) REFERENCES Product (ProductID) ON DELETE CASCADE );

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY ,

CustomerName VARCHAR (20) NOT NULL ,

Location VARCHAR (50) NOT NULL,

Email VARCHAR (50) UNIQUE NOT NULL,

Phone VARCHAR (50) NOT NULL,

ProductID INT NOT NULL,

CONSTRAINT CustomerFK FOREIGN KEY (ProductID) REFERENCES Product (ProductID) ON DELETE CASCADE

);

CREATE TABLE User\_StockIN\_OUT\_Product (

EmployeeID INT PRIMARY KEY,

ProductID INT ,

Quantity INT NOT NULL,

Date DATETIME NOT NULL,

TransactionNumber INT UNIQUE NOT NULL,

Comments VARCHAR (50),

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

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

);