Table of Contents

Introduction

A website application system called stock control management covers the product management for business equipment or other goods that need to be stored. The primary goal of the project is to create a software model for a stock control management system that will display all of the organization's stock information. These days, many businesses use the system to prevent overstock, errors in accounting, and outages. It is a mechanism for better organizing inventory data than was previously used, which is often kept in spreadsheets or manual form books.

To manage records, inventory, and system maintenance for the inventory, this program features an admin component. The program includes a general organization profile as well as information on the stock, purchases, and remaining stock as it is displayed in the organization. Along with the specifics of the transaction balance, this application also provides the stock's remaining balance. Each new stock is formed, entitled, and given a name and entry date. It can also be updated as necessary based on transactions or sales returns, as applicable. In this case, the login page is made to safeguard the organization's stock management in order to shield it from theft and improper usage of the inventory.

The proposal system's goals include:

- A user-friendly system that manages product or item information and calculates it to manage the information system must be designed and developed.
- To assist the user in locating and determining the amount of stored stocks
- To create a program that addresses the requirements of any industrial company on a daily basis.

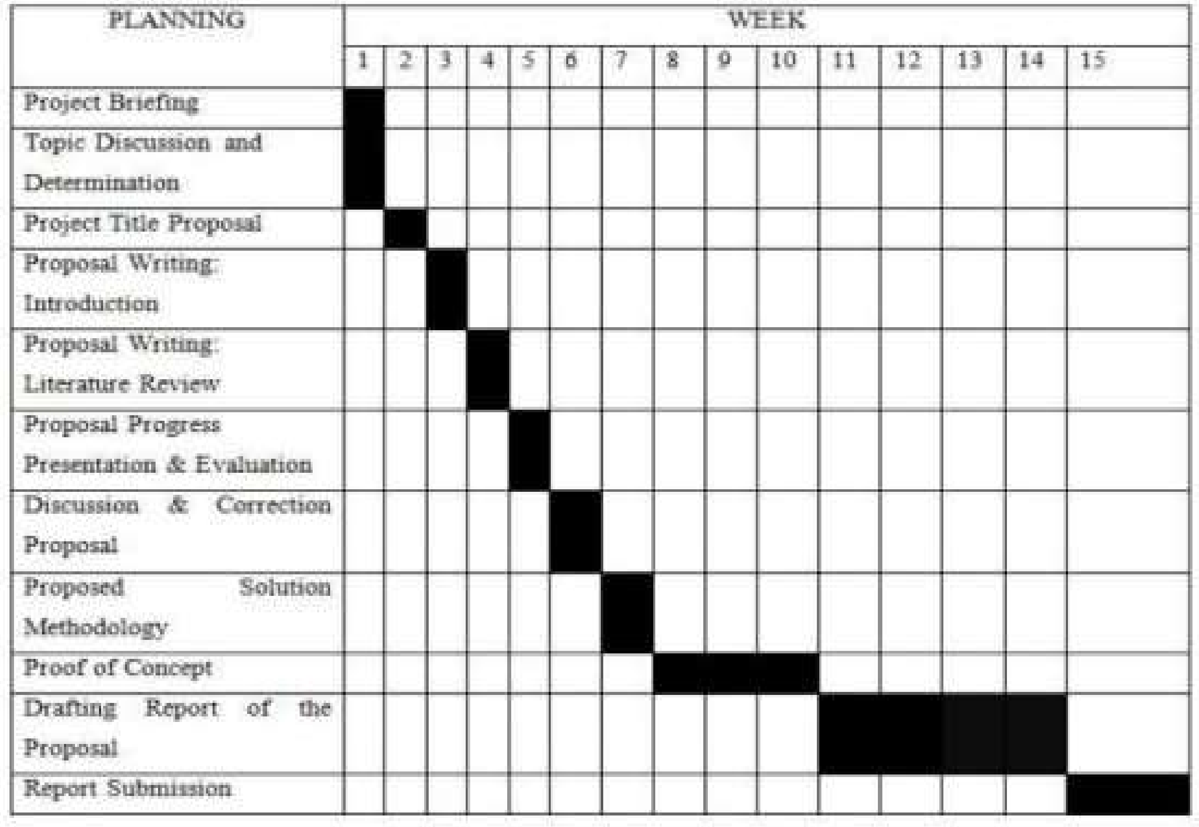
This application is used to provide information about old and new items as well as the amount of stock still available. It provides information on the stock on a daily and weekly basis.

- Login/Sign in page: The login page shows as the application launches. The username and password for the admin login decide who has the power to add, update, and delete stock as necessary for the company.
- Stock details: It displays information on the inventory's remaining supply. Additionally, it displays information about the used stock.
- Purchase details: It displays information on the organization's purchase, including the price and dates.

- Calculation of the number of objects that have been and will be stored using automation.

Gantt Chart

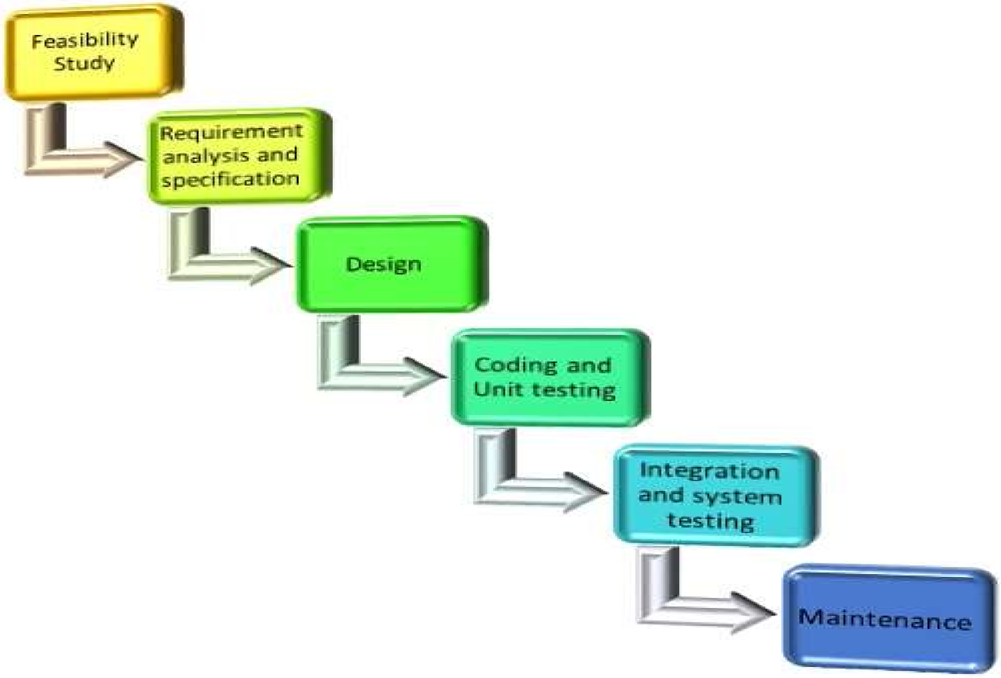
A Gantt chart is a diagram in which a series of horizontal lines compares the quantity of work or production actually completed over time to the amount scheduled for each period.



PLANNING	WEEK														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Project Briefing	■														
Topic Discussion and Determination	■														
Project Title Proposal		■													
Proposal Writing: Introduction			■												
Proposal Writing: Literature Review				■											
Proposal Progress Presentation & Evaluation					■										
Discussion & Correction Proposal						■									
Proposed Solution Methodology							■								
Proof of Concept								■	■	■					
Drafting Report of the Proposal											■	■	■	■	
Report Submission															■

Plan-Driven Development

An application's development process that is more formally precise is called plan-driven software development. The following characteristics are shared by all plan-driven methodologies: repeatability, predictability, defined incremental process, substantial documentation, upfront system architecture, specific plans, process monitoring, controlling, and education, risk management, verification, and validation. The Stock Control System was developed using the Waterfall Model, which is a sequential and linear approach to the Software Development Life Cycle (SDLC). The lifecycle is divided into six distinct stages that don't overlap each other, and each stage builds on the one before it and can begin once the one before it is complete.


Feasibility Study
Requirement analysis and specification
Design
Coding and Unit testing
Integration and system testing
Maintenance

- **Feasibility Study**

  The major objective of this phase is to evaluate the software development's technical and economical viability for the stock control system. Understanding the issue and identifying potential solutions are the first steps in the feasibility study. The advantages and disadvantages of the various solutions are considered, and the best option is selected. The next phases are then carried out in accordance with this solution approach.

- **Requirements analysis and specification**

  The goal of the requirement analysis and specification phase at this point is to accurately identify and record the customer's requirements.

- **Design**

  This phase's objective is to transform the system requirement specifications' collected requirements into a form that can be programmed in a programming language. It encompasses the general software architecture as well as high-level and detailed design. All of this work is documented using a software design document (SDD)

- **Coding and Unit Testing**

  During the coding step, any appropriate programming language is used to convert the software design into source code. Each designed module is therefore coded. The goal of the unit testing step is to determine whether or not each module is functioning correctly.

- **Integration and System Testing**

  Different modules are integrated as soon as they have completed unit testing and coding. Over a number of steps, integration of various modules is done incrementally. The system is tested after each integration phase, which involves adding previously designed components to the partially integrated system. Finally, a fully functional system is obtained and tested when all the modules have been successfully integrated and tested.

- **Maintenance**

  The maintenance stage of a software life cycle is the most crucial. It is done to fix problems that weren't found during the product development stage, improve the system's functionality in response to client requests, and transfer the software to work in a new environment, such as on a new computer platform or with a new operating system.

Functional and Non-functional Requirements

A stock control management system combines technology (hardware and software), processes, and procedures to oversee the monitoring and upkeep of stocked goods, whether they are firm assets, raw materials, supplies, or finished goods that are ready to be dispatched to suppliers or end users. The system will also assist managers in selecting the best products for the recording process, where only recently used and popular products will be ordered. The goal of a stock control management system is to ensure that inventory flows easily and assists in decision-making that will minimize the entire cost of inventory, which is very different from minimizing inventory.

Functional Requirements

Any requirement that specifies what the system should do is referred to as a functional requirement of a system. In other words, a functional requirement will outline a specific action that the system should take when a set of criteria are satisfied.

The functional requirements includes:

- A user's authentication when they attempt to log into the system using some login credentials.
- The system must have features that allow users to sign up for the system by entering various information.
- The system must display both the inventory that is currently on the system and the specifics of the stock.
- The shutdown of the system in the event of a cyberattack.
- The system must include functionality that allows administrators or managers to add and remove stock or product details.
- Data on revenue and transactions can only be viewed by personnel at the managerial level.
- Every time a user registers for the first time on a software system, a verification email is issued to them.
- Users must be able to easily log off of the system using the system.

Non-Functional Requirements

Any need that details how the system executes a specific function is considered a non-functional requirement. In other words, a non-functional requirement will outline the expected behavior of a system and the boundaries of its functionality. For instance, Emails must to be sent with a maximum latency of 12 hours, It should take 10 seconds to process each request, When there are more than 10,000 concurrent visitors, the website should load in three seconds.

Non-functional requirements includes:

- **Speed** - How quickly an application reacts to commands is determined by speed. For instance, the speed of the search engine affects how soon you obtain search results when you type a word into the search bar. Assessing a system's capacity to handle a growing workload when you utilize multiple programs at once is another aspect of speed.

- **Security** - You might think about creating security features that aren't functional to protect sensitive data. Secure databases are used, for instance, to store data in stock control systems. To prevent unwanted access, their databases may be secured with user authentication.
- **Portability** - The term "portability" refers to how well a system works in one environment as opposed to another. The application is highly portable if it performs as well in the new environment as it did in the previous one. As a developer, you can create programs that run well on a variety of platforms to increase portability.
- **Compatibility** - When additional apps are running on a device, highly compatible systems often work well. Additionally, compatibility enables users of various operating systems to utilize the same programs.
- **Reliability** - Highly dependable technology continues to perform well even after extensive use.
- **Environment** - The environment consists of outside elements that affect how well your system works. The environment of an application may also include the time frame during which it operates, such as constantly or only when the user activates it.

Hardware Requirements

- Processor: x86 or x64
- RAM : 512 MB (minimum), 1 GB (recommended)
- Hard disk: up to 200 MB of available space may be required. However, 50 MB free space is required in boot drive even if you are installing in other drive.

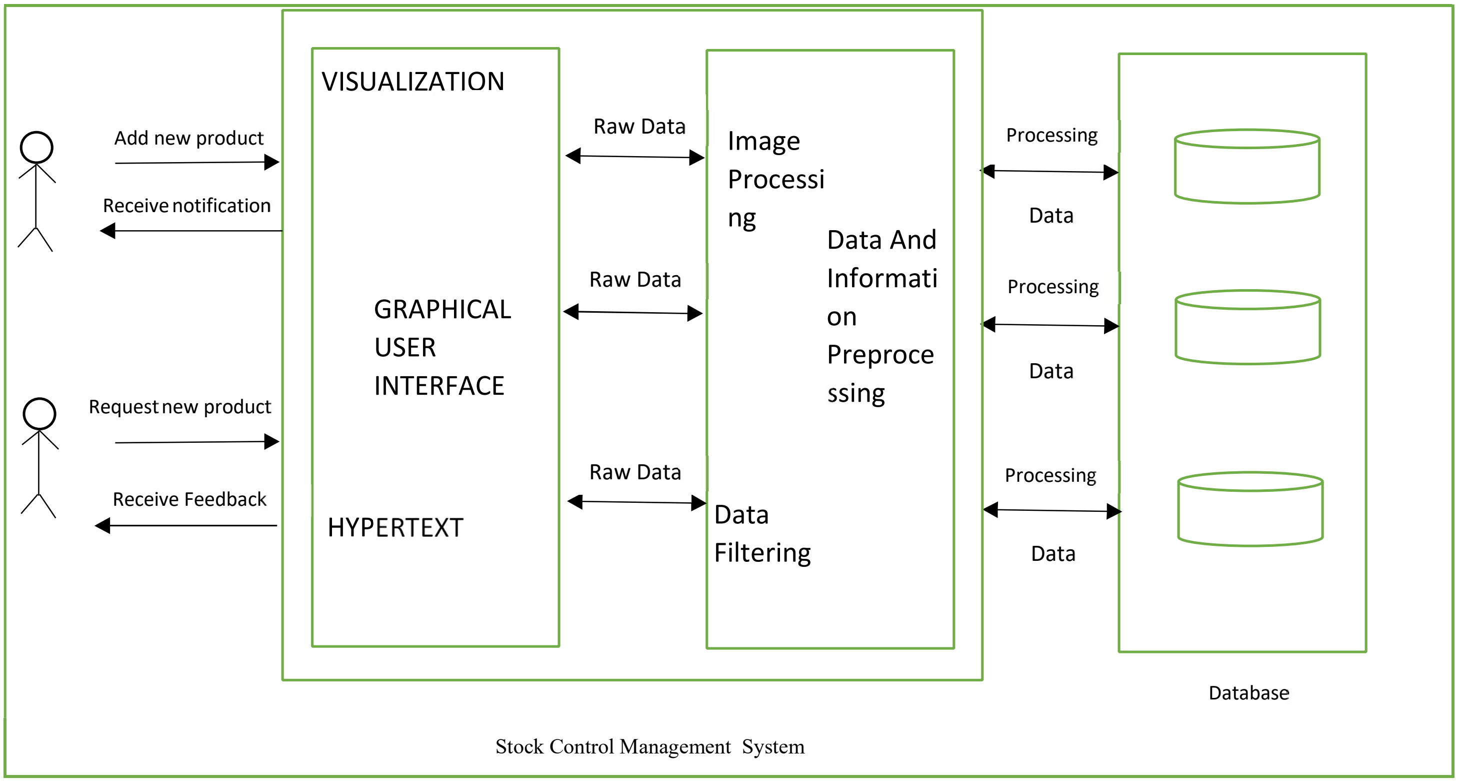
Software Requirements

- PHP 5.3.3+ version.
- Apache Tomcat
- MySQL (Necessary for DataBase related functionalities)

System Modelling

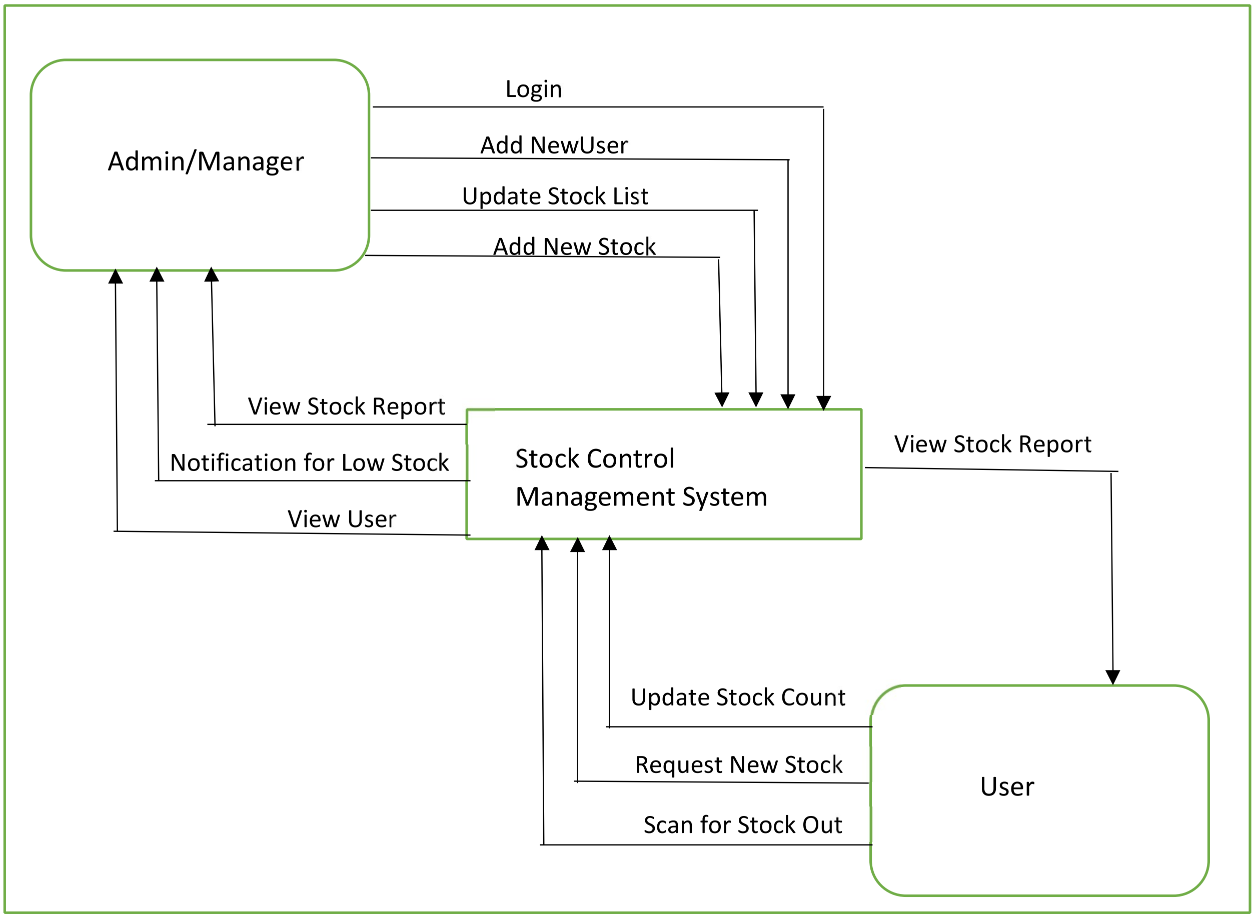
System modeling is the process of creating abstract representations of a system, with each model offering a distinctive viewpoint. It has to do with graphical notation, which is now nearly invariably based on notations in the Unified Modeling Language, to depict a system (UML).

Framework Design

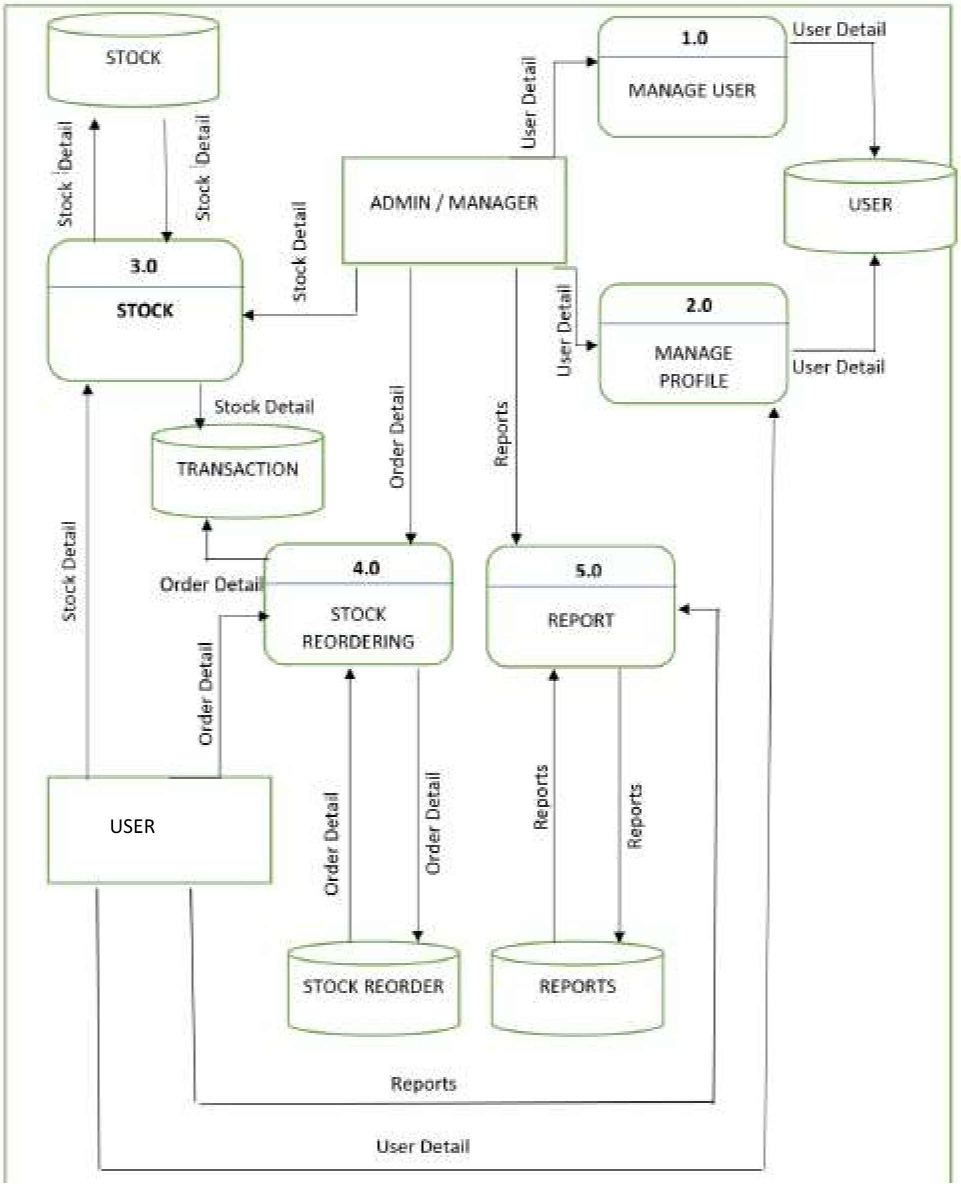


Framework Design

Context Diagram



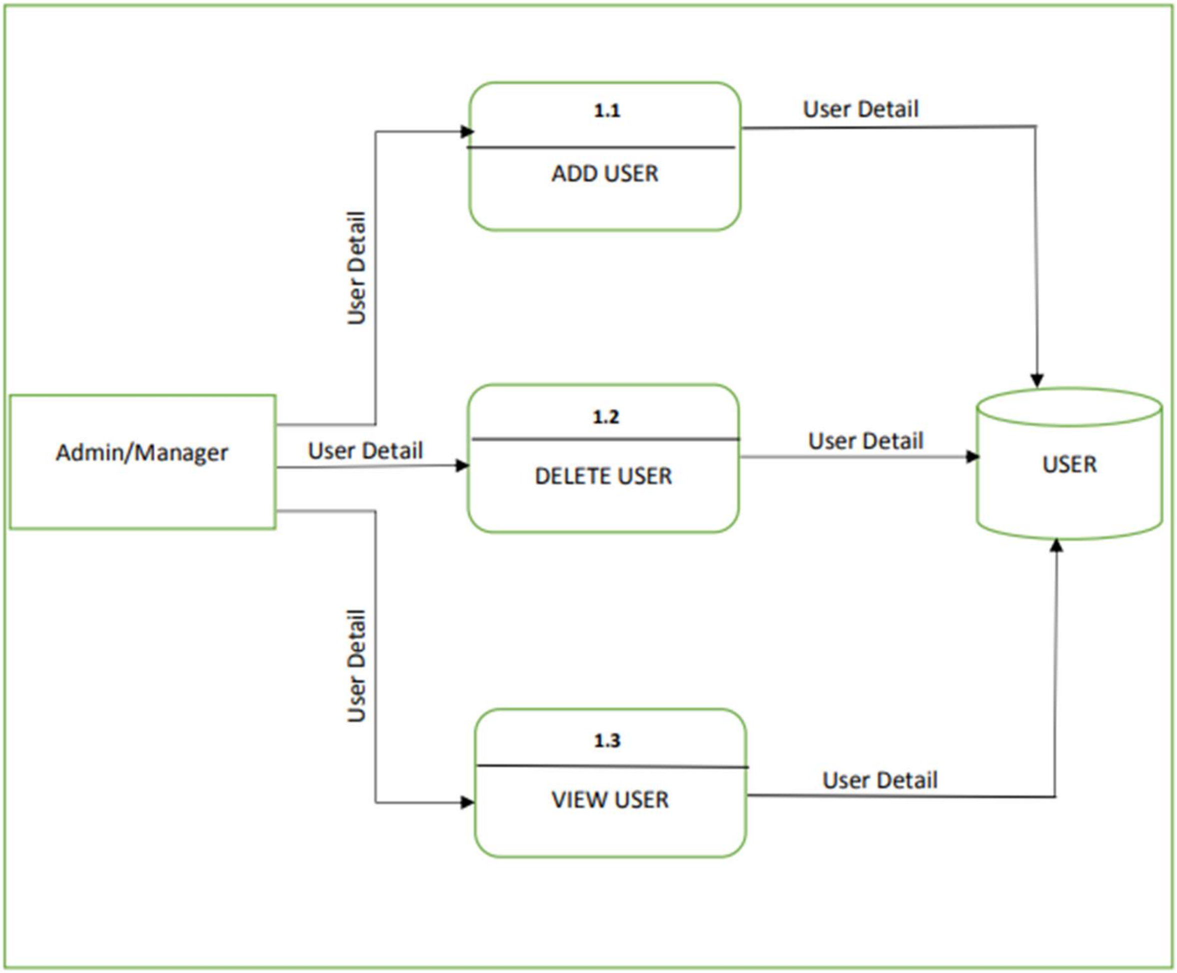
Data Flow Diagram Level 0



The ability to manage other system users belongs exclusively to the Admin/Manager role. The Admin/Manager will be in charge of adding and removing users. The system's administrator can also log in and manage new stocks, reports, and the stock reordering process. The Transaction database will save each and every stock transaction that occurs.

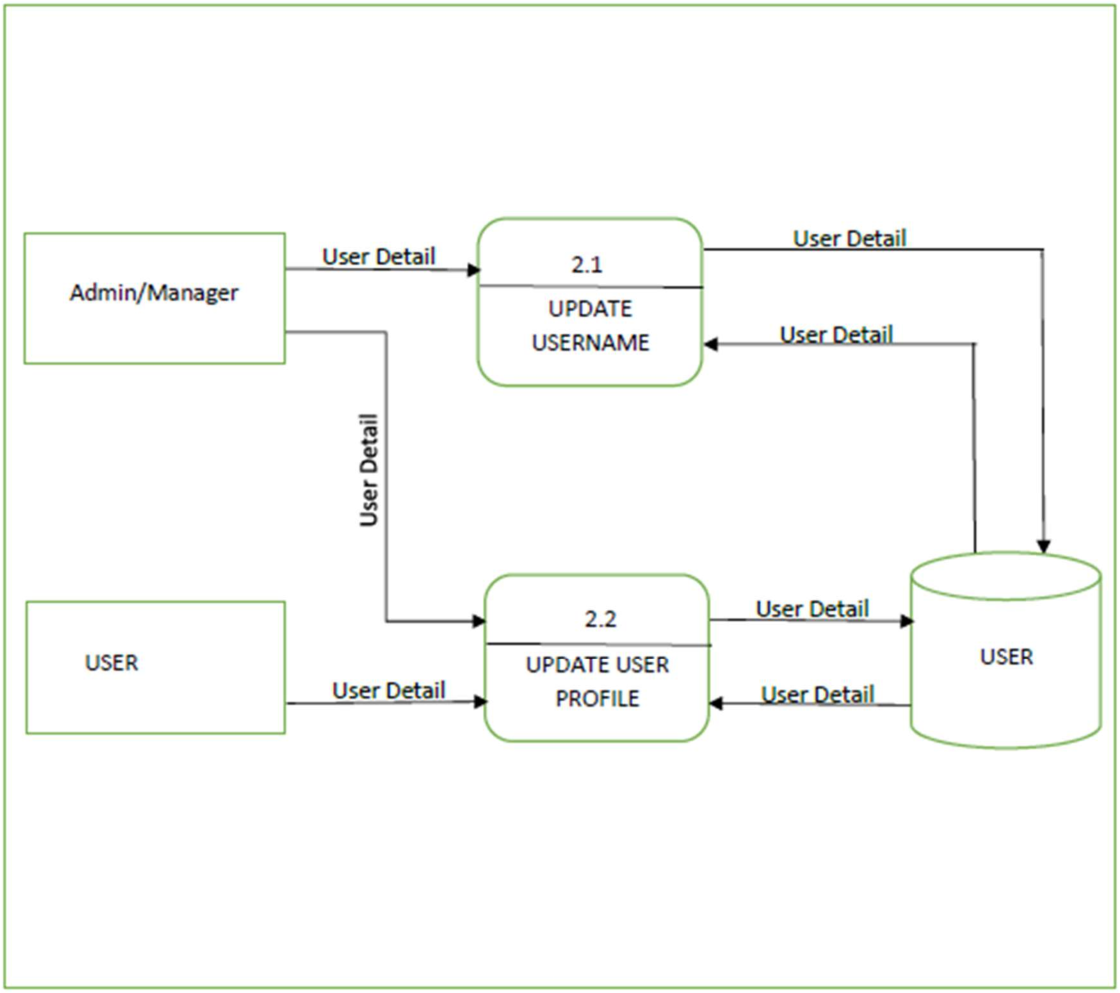
Data Flow Diagram Level 1 (Process 1.0)

Admin can add, delete and view detail of users in the system



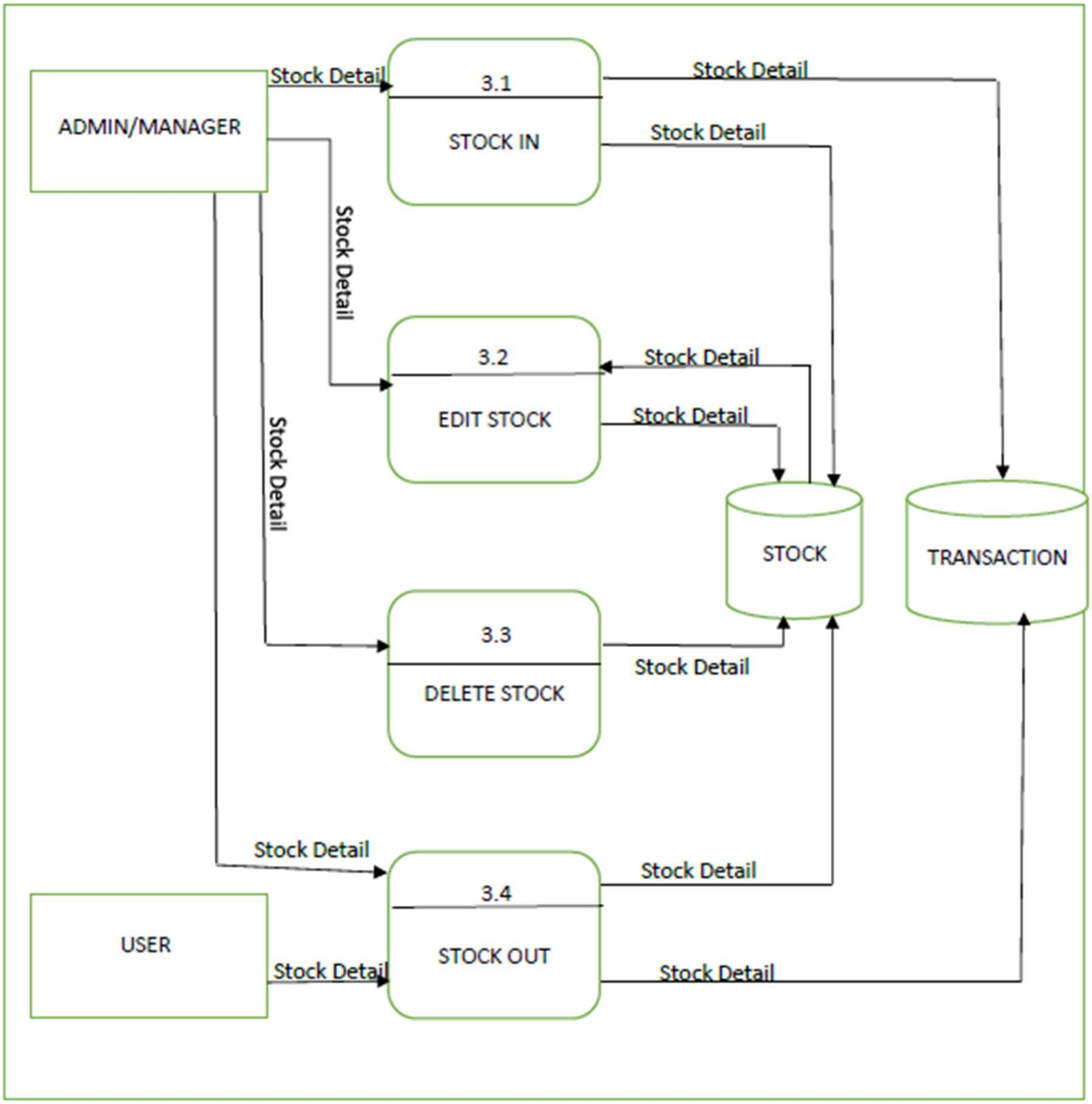
Data Flow Diagram Level 1 (Process 2.0)

Admin can update the username and user profile but User only can update their profile.

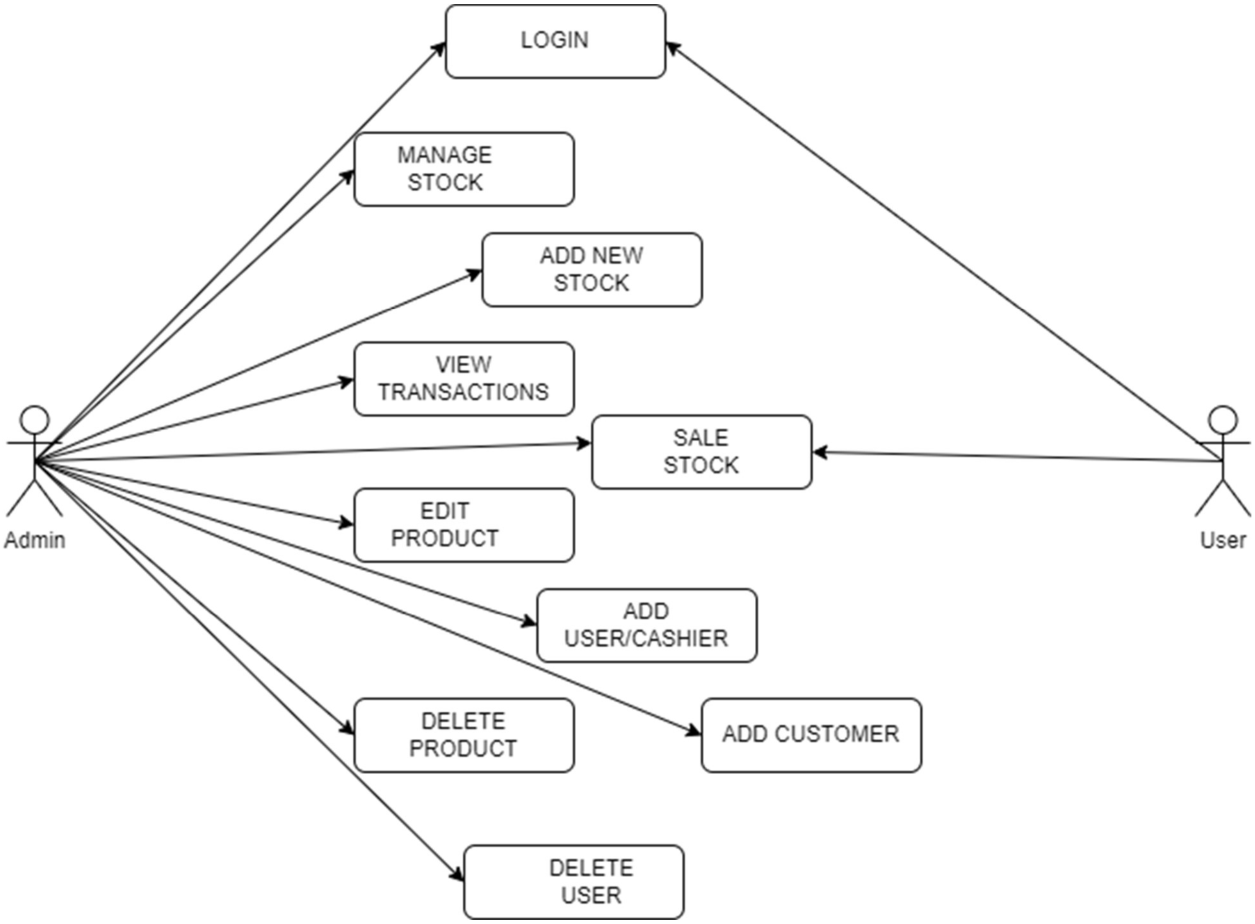

Admin/Manager
User Detail
2.1
UPDATE USERNAME
User Detail
User Detail
User Detail
USER
User Detail
2.2
UPDATE USER PROFILE
User Detail
User Detail
USER

Data Flow Diagram Level 1 (Process 3.0)

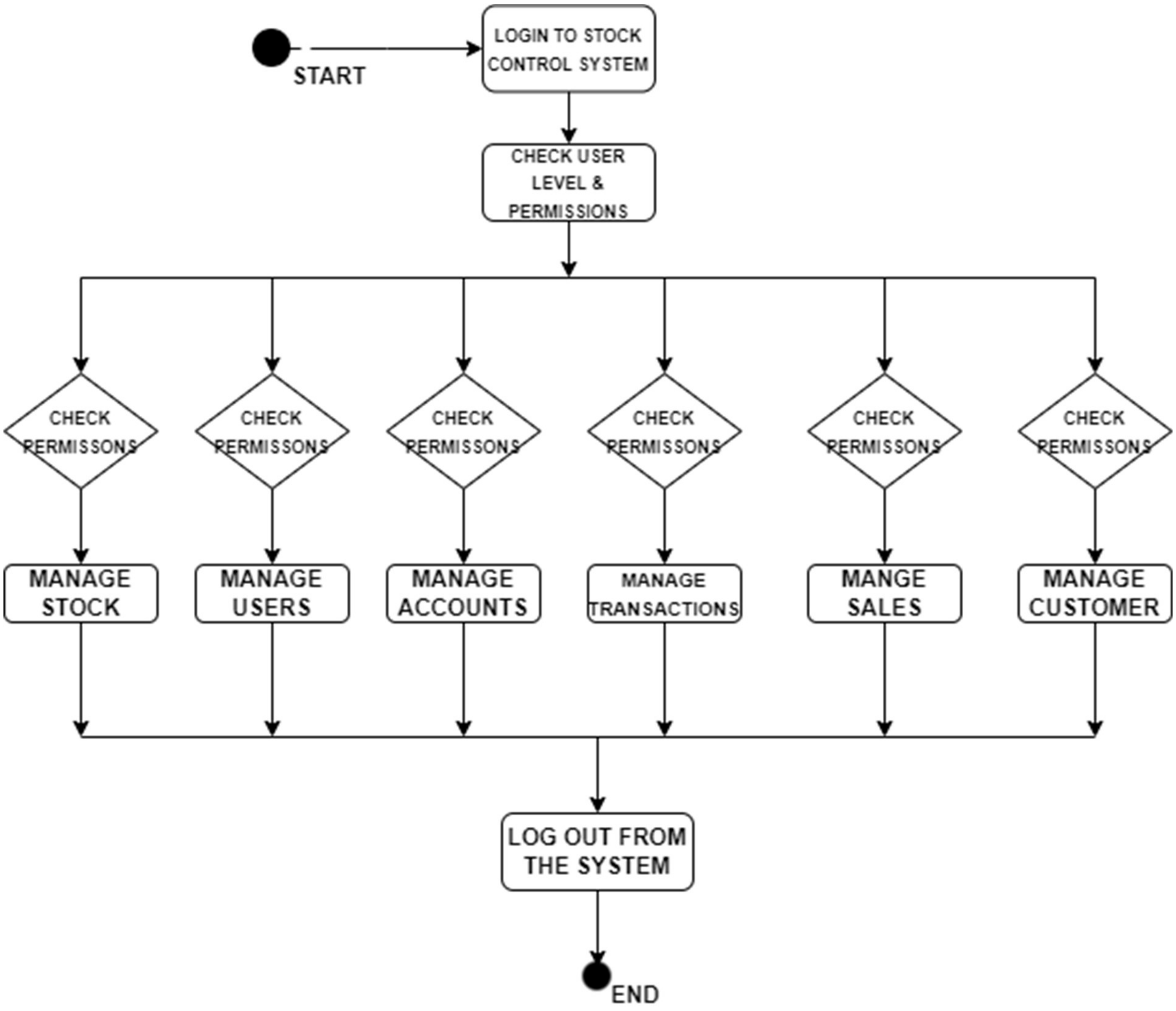
The user can only make a stock out from the system; the administrator can add (Stock in), update, delete, and Stock out the stock from the system. The Transaction Database will save each Stock In and Stock Out transaction.



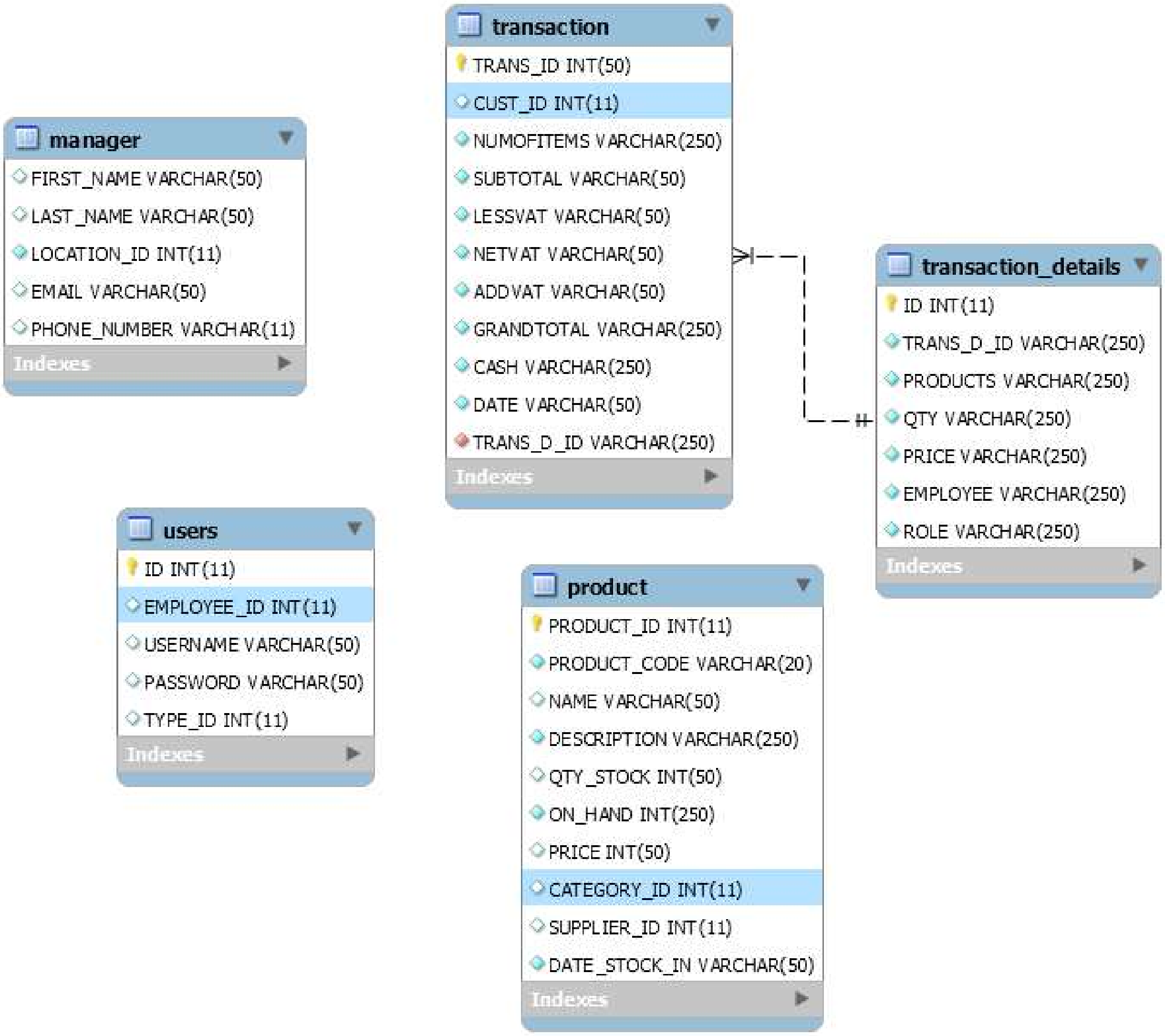
USER CASE DIAGRAM

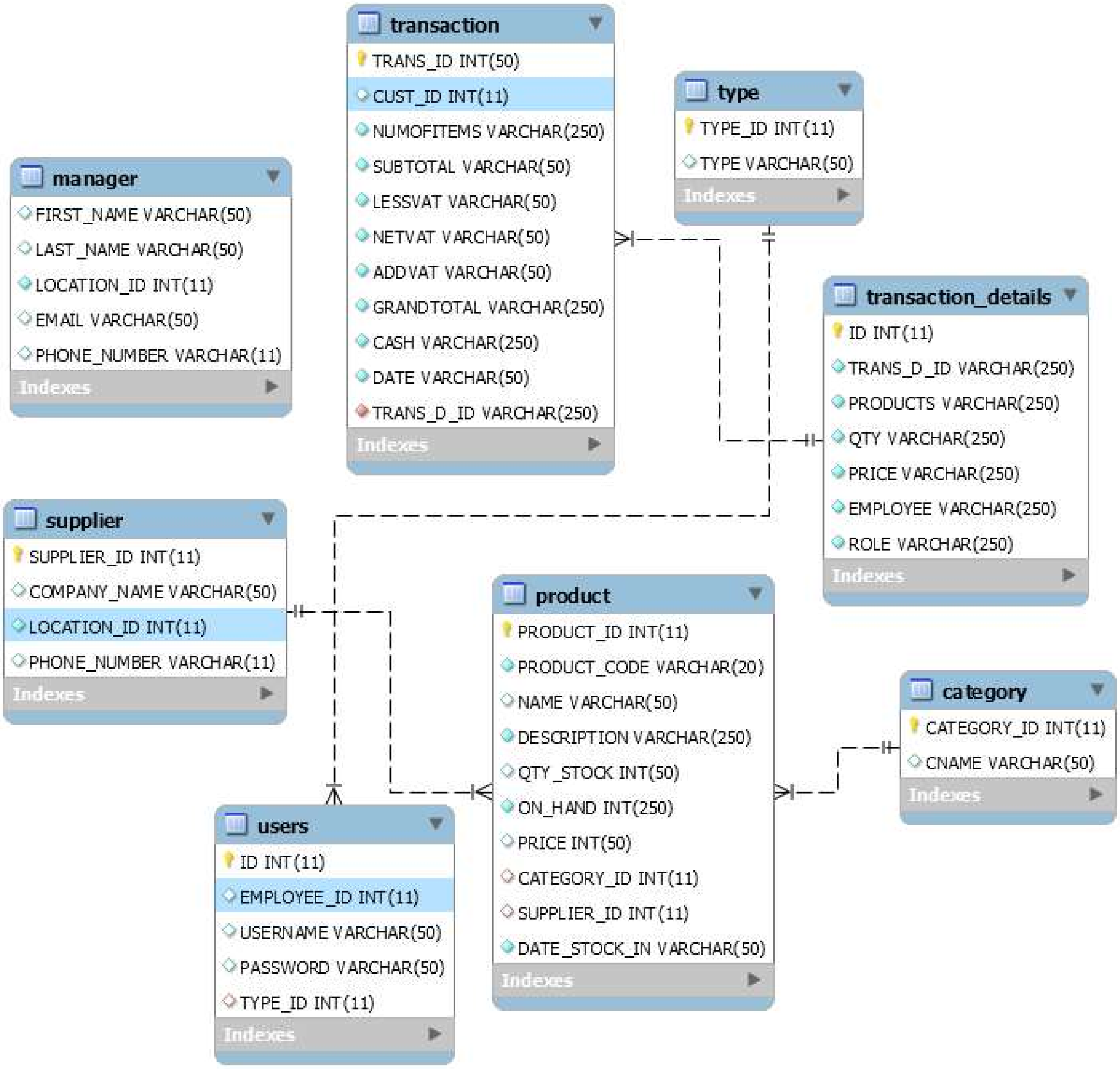


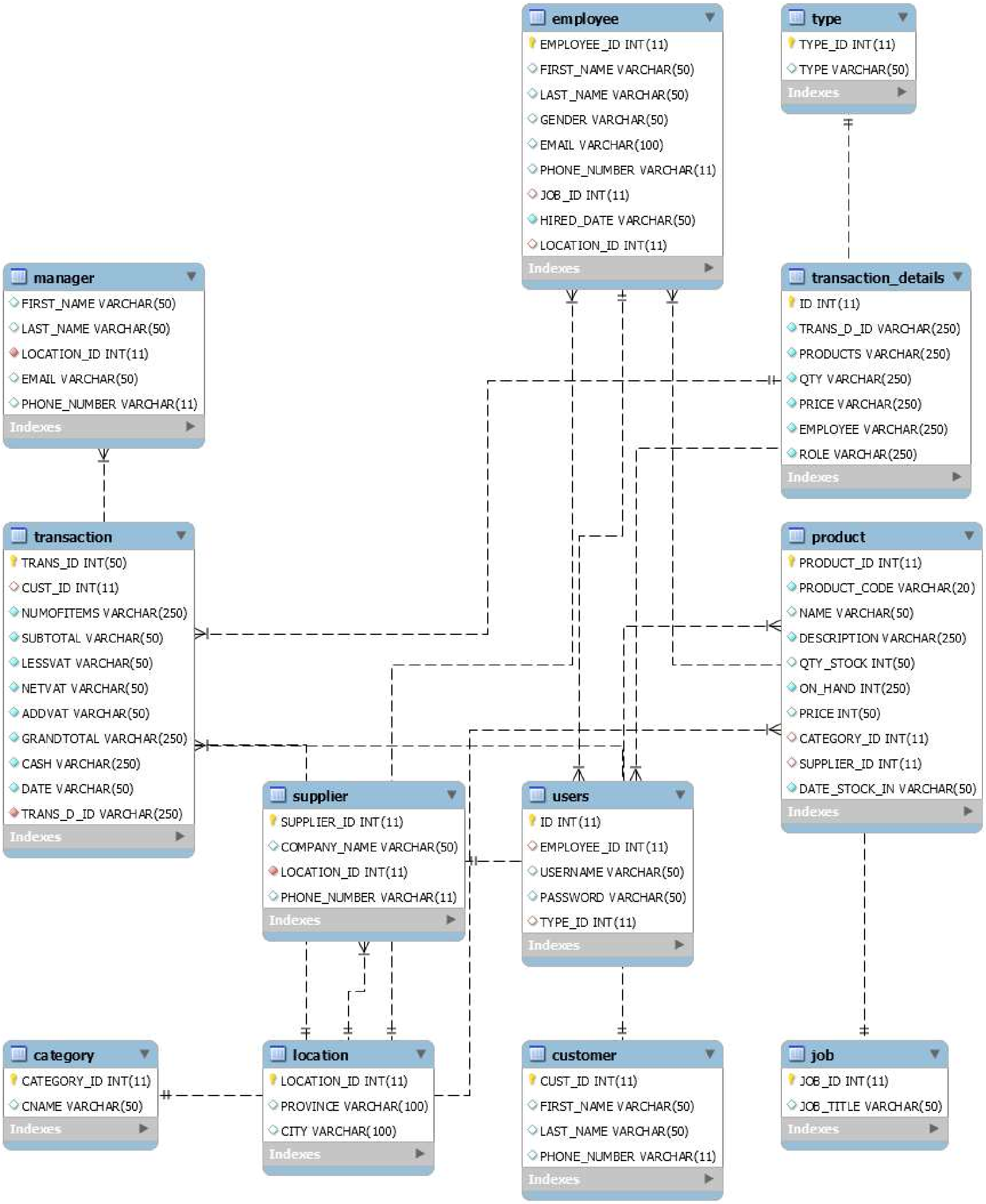
ACTIVITY DIAGRAM



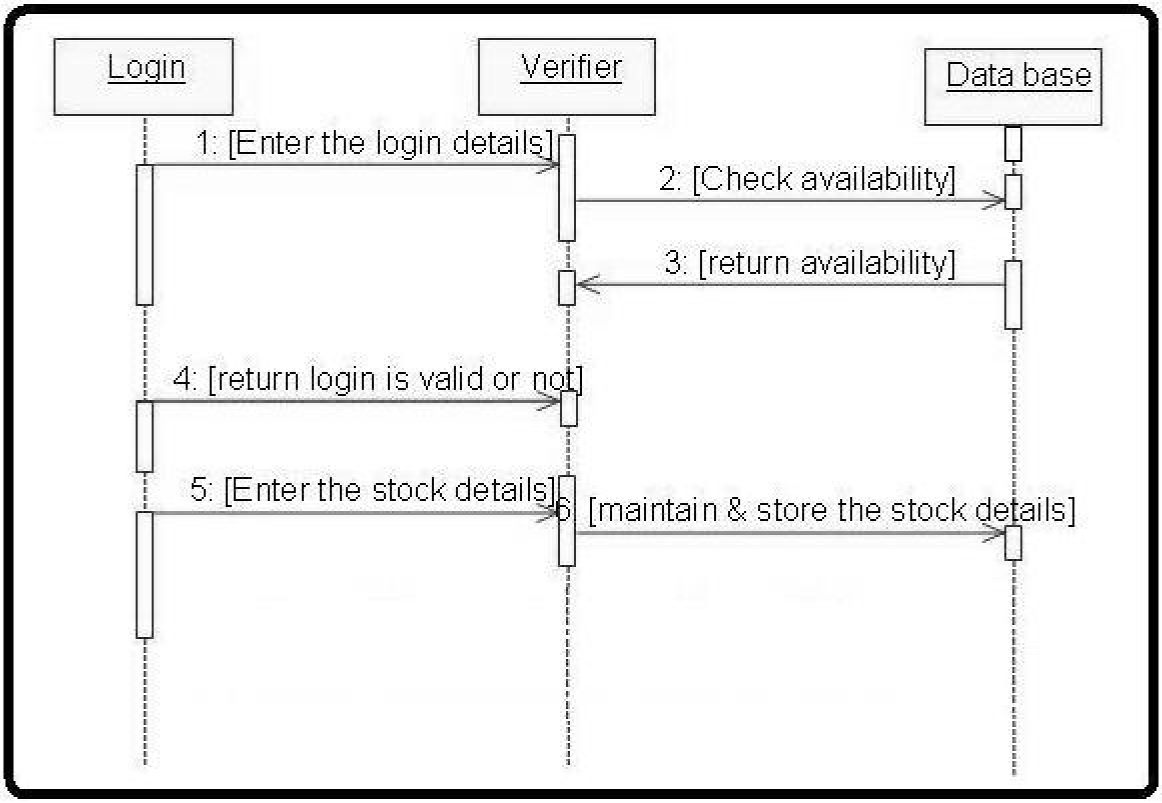
CLASS DIAGRAM

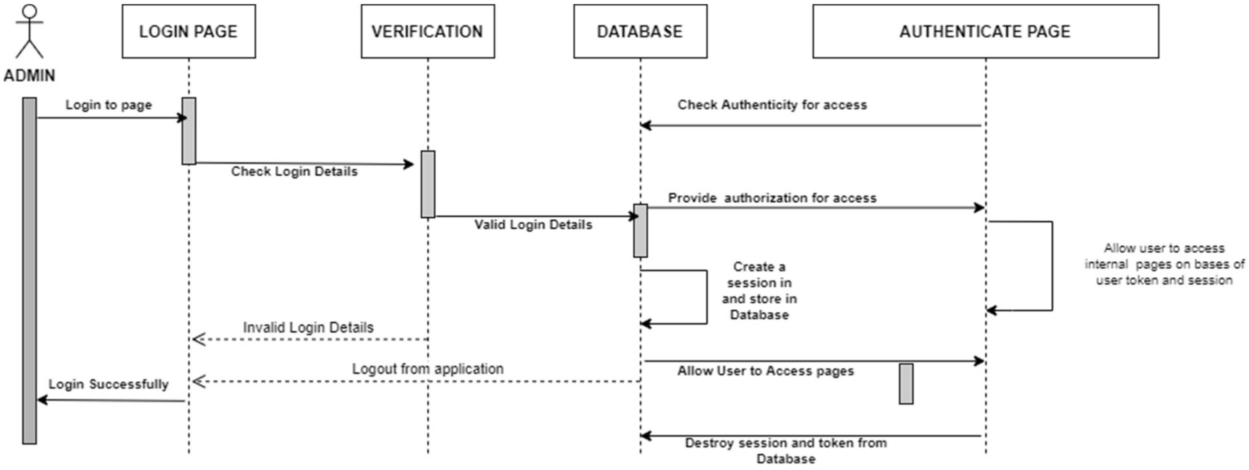


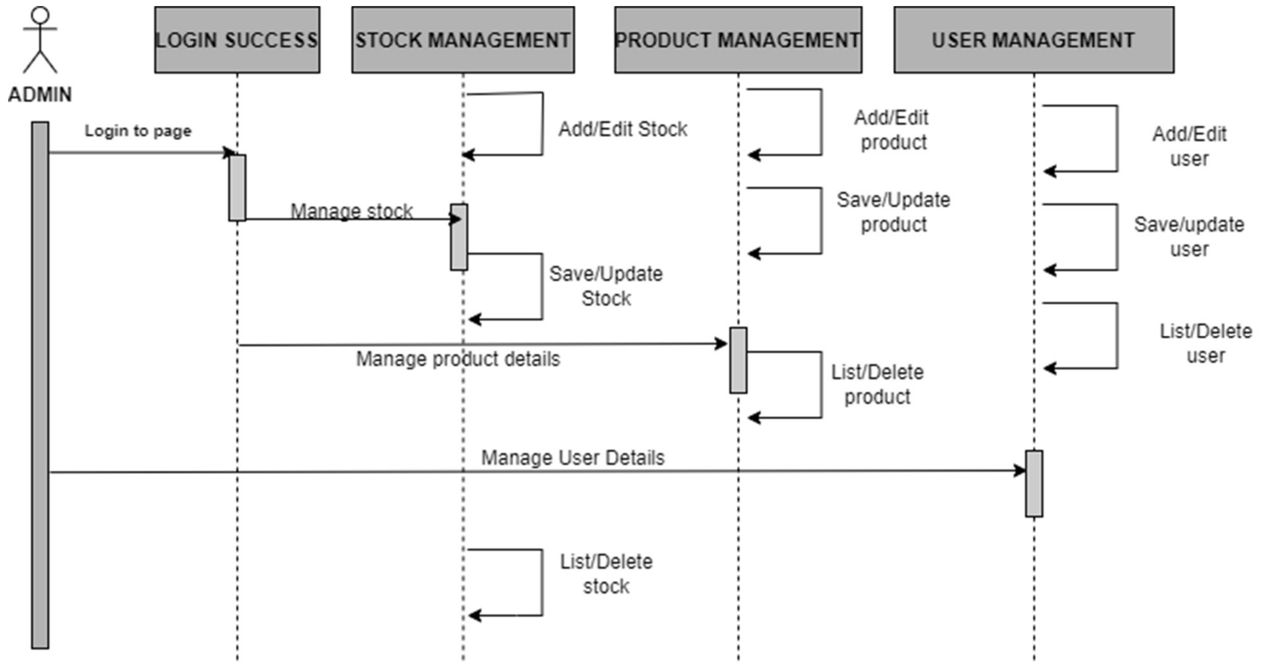

transaction
TRANS_ID INT(50)
CUST_ID INT(11)
NUMOFITEMS VARCHAR(250)
SUBTOTAL VARCHAR(50)
LESSVAT VARCHAR(50)
NETVAT VARCHAR(50)
ADDVAT VARCHAR(50)
GRANDTOTAL VARCHAR(250)
CASH VARCHAR(250)
DATE VARCHAR(50)
TRANS_D_ID VARCHAR(250)
Indexes
type
TYPE_ID INT(11)
TYPE VARCHAR(50)
Indexes
manager
FIRST_NAME VARCHAR(50)
LAST_NAME VARCHAR(50)
LOCATION_ID INT(11)
EMAIL VARCHAR(50)
PHONE_NUMBER VARCHAR(11)
Indexes
transaction_details
ID INT(11)
TRANS_D_ID VARCHAR(250)
PRODUCTS VARCHAR(250)
QTY VARCHAR(250)
PRICE VARCHAR(250)
EMPLOYEE VARCHAR(250)
ROLE VARCHAR(250)
Indexes
supplier
SUPPLIER_ID INT(11)
COMPANY_NAME VARCHAR(50)
LOCATION_ID INT(11)
PHONE_NUMBER VARCHAR(11)
Indexes
product
PRODUCT_ID INT(11)
PRODUCT_CODE VARCHAR(20)
NAME VARCHAR(50)
DESCRIPTION VARCHAR(250)
QTY_STOCK INT(50)
ON_HAND INT(250)
PRICE INT(50)
CATEGORY_ID INT(11)
SUPPLIER_ID INT(11)
DATE_STOCK_IN VARCHAR(50)
Indexes
category
CATEGORY_ID INT(11)
CNAME VARCHAR(50)
Indexes
users
ID INT(11)
EMPLOYEE_ID INT(11)
USERNAME VARCHAR(50)
PASSWORD VARCHAR(50)
TYPE_ID INT(11)
Indexes


employee
EMPLOYEE_ID INT(11)
FIRST_NAME VARCHAR(50)
LAST_NAME VARCHAR(50)
GENDER VARCHAR(50)
EMAIL VARCHAR(100)
PHONE_NUMBER VARCHAR(11)
JOB_ID INT(11)
HIRED_DATE VARCHAR(50)
LOCATION_ID INT(11)
Indexes
type
TYPE_ID INT(11)
TYPE VARCHAR(50)
Indexes
manager
FIRST_NAME VARCHAR(50)
LAST_NAME VARCHAR(50)
LOCATION_ID INT(11)
EMAIL VARCHAR(50)
PHONE_NUMBER VARCHAR(11)
Indexes
transaction_details
ID INT(11)
TRANS_D_ID VARCHAR(250)
PRODUCTS VARCHAR(250)
QTY VARCHAR(250)
PRICE VARCHAR(250)
EMPLOYEE VARCHAR(250)
ROLE VARCHAR(250)
Indexes
transaction
TRANS_ID INT(50)
CUST_ID INT(11)
NUMOFITEMS VARCHAR(250)
SUBTOTAL VARCHAR(50)
LESSVAT VARCHAR(50)
NETVAT VARCHAR(50)
ADDVAT VARCHAR(50)
GRANDTOTAL VARCHAR(250)
CASH VARCHAR(250)
DATE VARCHAR(50)
TRANS_D_ID VARCHAR(250)
Indexes
product
PRODUCT_ID INT(11)
PRODUCT_CODE VARCHAR(20)
NAME VARCHAR(50)
DESCRIPTION VARCHAR(250)
QTY_STOCK INT(50)
ON_HAND INT(250)
PRICE INT(50)
CATEGORY_ID INT(11)
SUPPLIER_ID INT(11)
DATE_STOCK_IN VARCHAR(50)
Indexes
supplier
SUPPLIER_ID INT(11)
COMPANY_NAME VARCHAR(50)
LOCATION_ID INT(11)
PHONE_NUMBER VARCHAR(11)
Indexes
users
ID INT(11)
EMPLOYEE_ID INT(11)
USERNAME VARCHAR(50)
PASSWORD VARCHAR(50)
TYPE_ID INT(11)
Indexes
category
CATEGORY_ID INT(11)
CNAME VARCHAR(50)
Indexes
location
LOCATION_ID INT(11)
PROVINCE VARCHAR(100)
CITY VARCHAR(100)
Indexes
customer
CUST_ID INT(11)
FIRST_NAME VARCHAR(50)
LAST_NAME VARCHAR(50)
PHONE_NUMBER VARCHAR(11)
Indexes
job
JOB_ID INT(11)
JOB_TITLE VARCHAR(50)
Indexes

SEQUENCE DIAGRAM


Login
Verifier
Data base
1: [Enter the login details]
2: [Check availability]
3: [return availability]
4: [return login is valid or not]
5: [Enter the stock details]
6: [maintain & store the stock details]


ADMIN
LOGIN PAGE
VERIFICATION
DATABASE
AUTHENTICATE PAGE
Login to page
Check Authenticity for access
Check Login Details
Provide authorization for access
Valid Login Details
Allow user to access internal pages on bases of user token and session
Create a session in and store in Database
Invalid Login Details
Logout from application
Allow User to Access pages
Login Successfully
Destroy session and token from Database

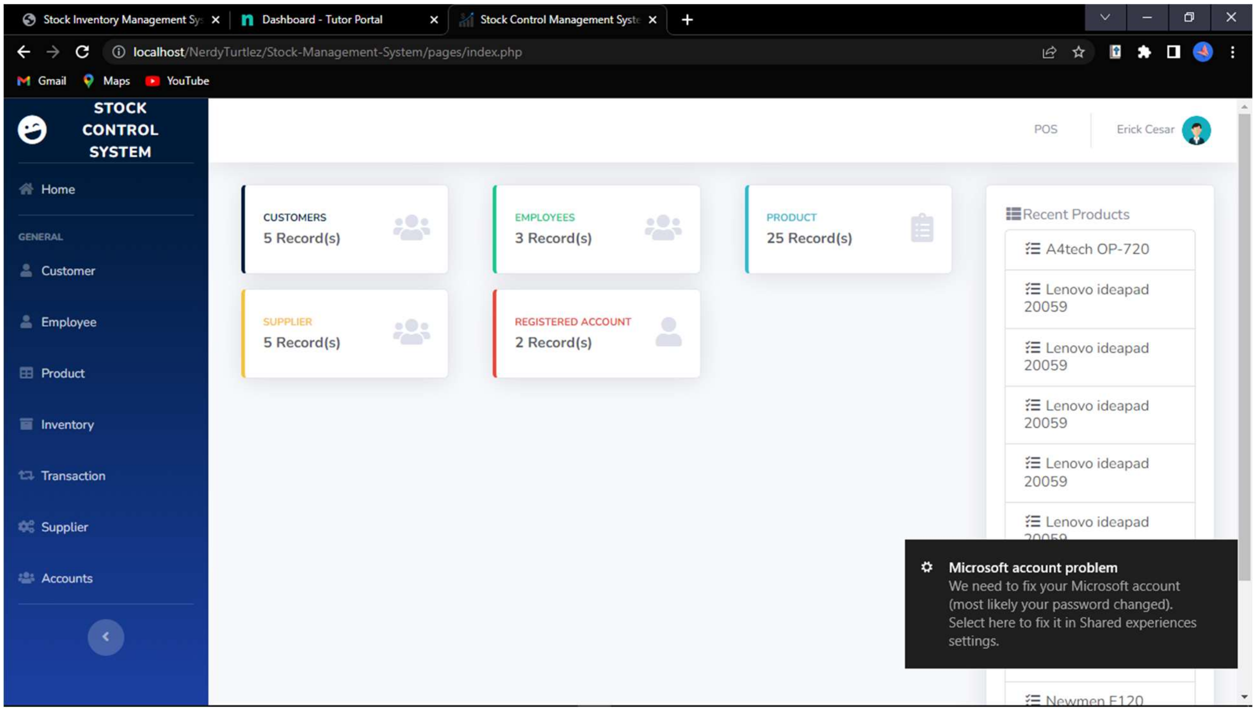


Implementation

The Stock Control System's features were presented as a website in the Implementation section of the research project. The system includes a number of capabilities, including login, registration, inventory, and many others. I used HTML to create and save the web document, then CSS to build the layout's appealing structure. Additionally, Bootstrap is used to make the design mobile-friendly and responsive. Additionally, PHP and JavaScript are used to create the system's web pages. Additionally, MySQL is used to access, administer, and query the database's data.

Dashboard/Home Page

The stock control system's home page and landing page are depicted in the diagram below.



CUSTOMERS 5 Record(s)

EMPLOYEES 3 Record(s)

PRODUCT 25 Record(s)

SUPPLIER 5 Record(s)

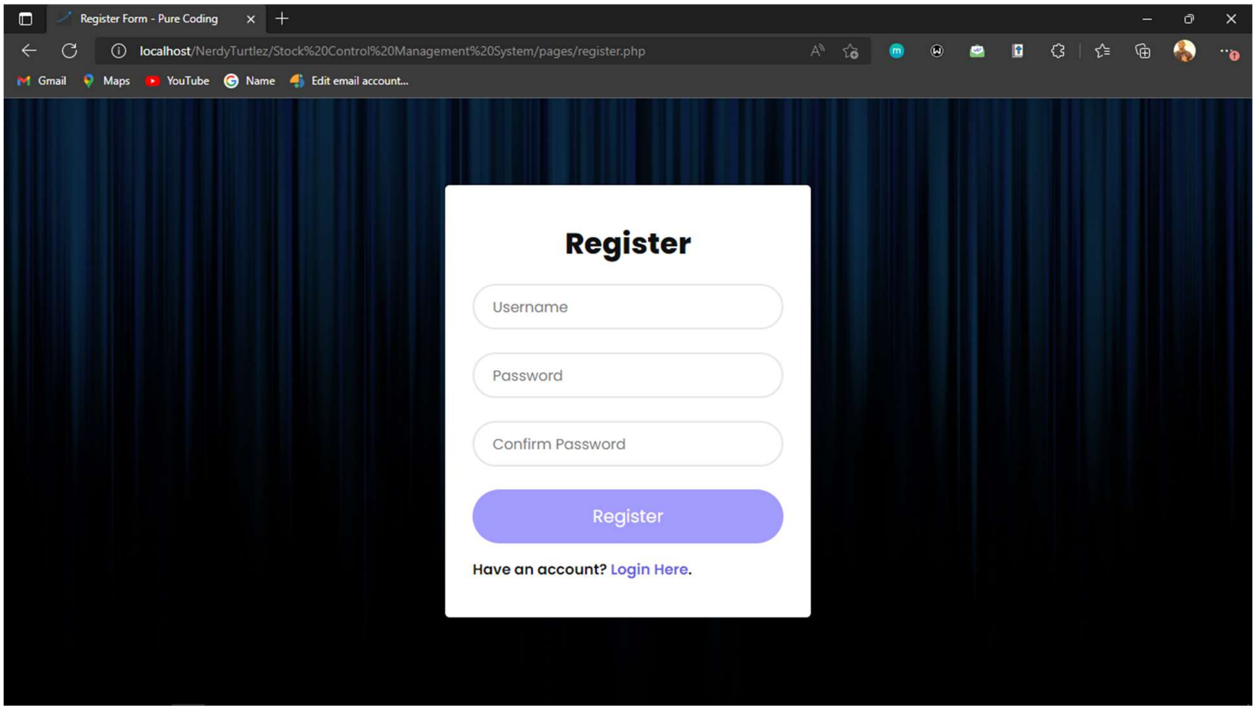
REGISTERED ACCOUNT 2 Record(s)

Recent Products
- A4tech OP-720
- Lenovo ideapad 20059
- Lenovo ideapad 20059
- Lenovo ideapad 20059
- Lenovo ideapad 20059
- Lenovo ideapad 20059
- Newmen E120

Microsoft account problem
We need to fix your Microsoft account (most likely your password changed). Select here to fix it in Shared experiences settings.

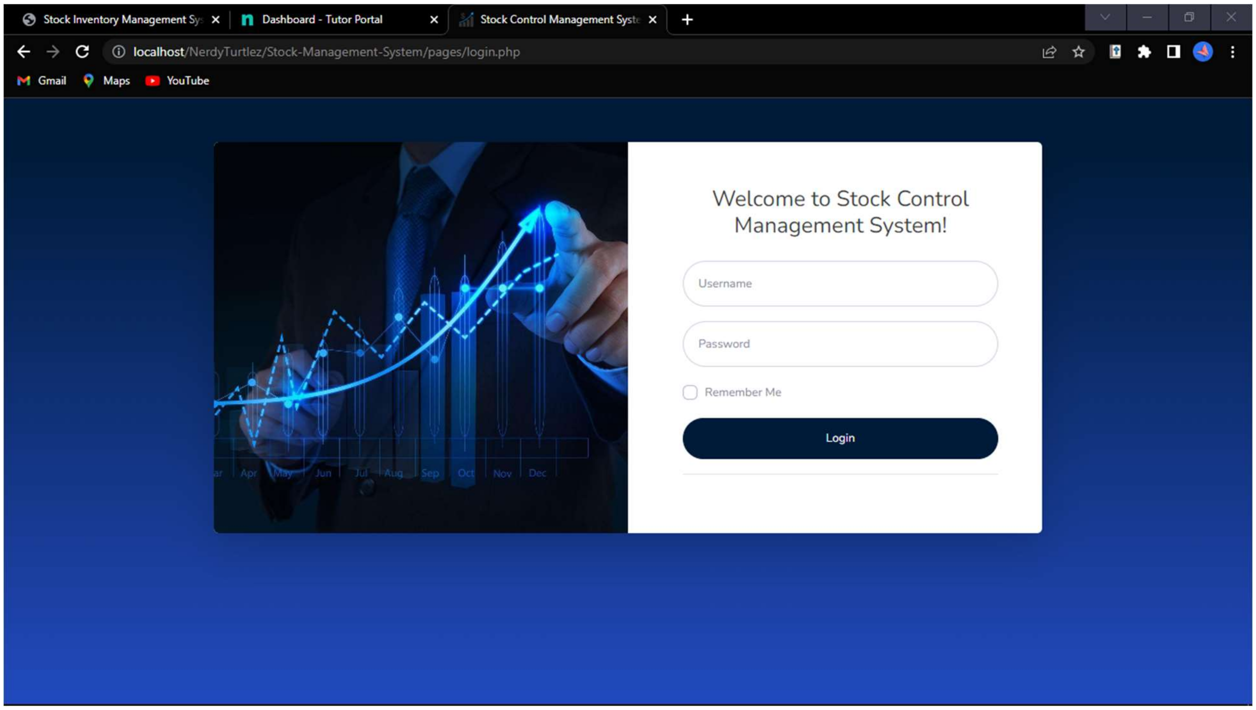
Registration Page

Based on the user registration webpage, the figure below was created. By providing their username and password, users can quickly establish a new account.


Register Form - Pure Coding
localhost/NerdyTurtlez/Stock%20Control%20Management%20System/pages/register.php
Gmail
Maps
YouTube
Name
Edit email account...
Register
Username
Password
Confirm Password
Register
Have an account? Login Here.

Login page

A user login web page is depicted in the following figure. By entering their username and password on this page, individuals can access their own accounts.

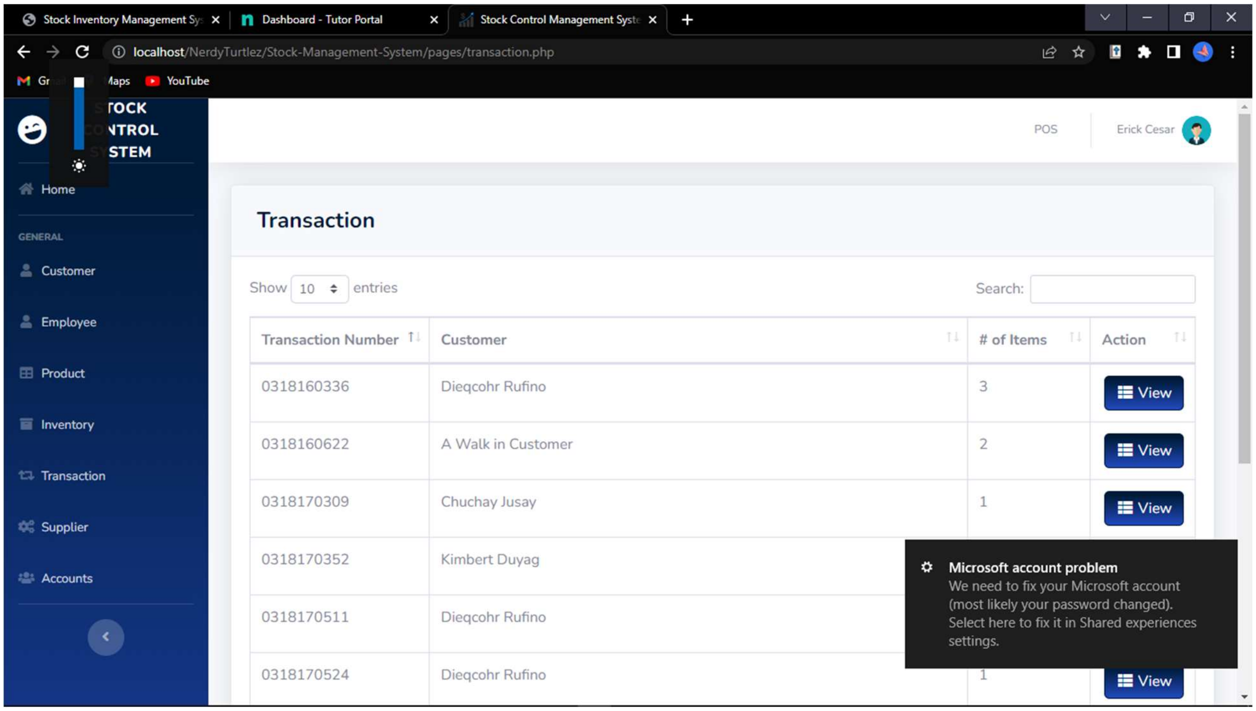


Stock Inventory Management Sy
Dashboard - Tutor Portal
Stock Control Management Syste
localhost/NerdyTurtlez/Stock-Management-System/pages/login.php
Gmail
Maps
YouTube
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Welcome to Stock Control Management System!

Username
Password
□ Remember Me
Login

Transaction Page



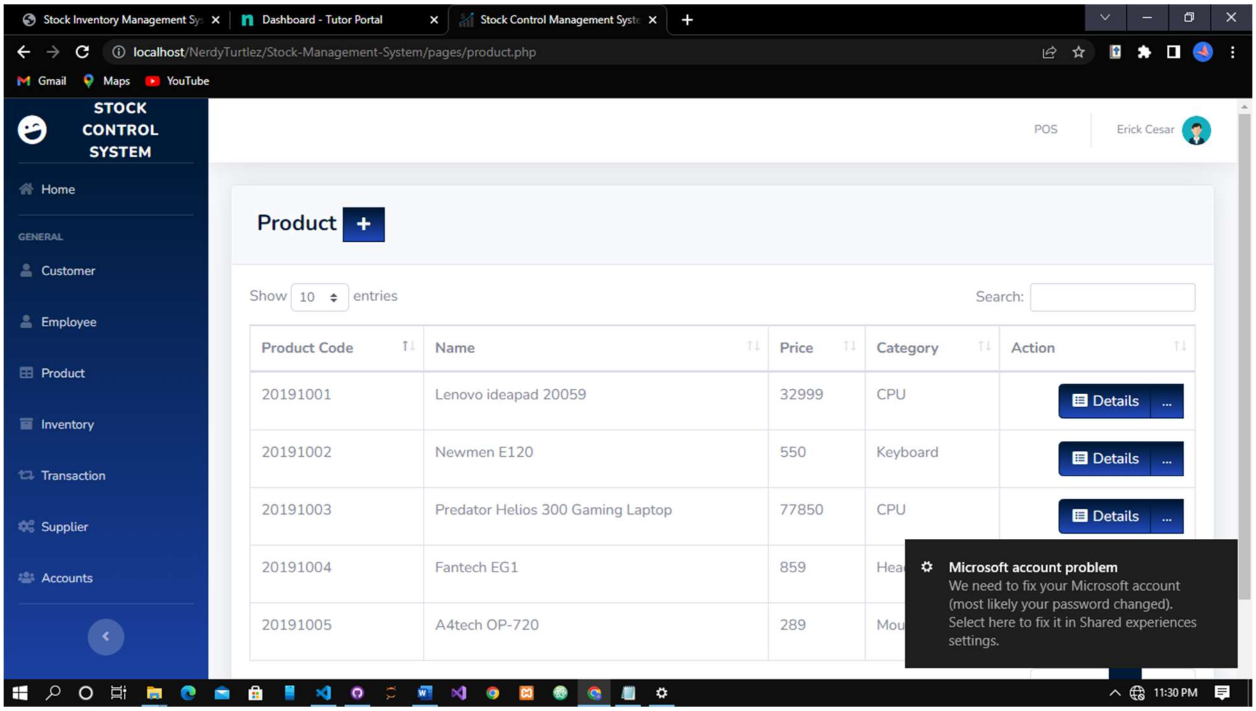
Transaction

Show 10 entries Search:

Transaction Number	Customer	# of Items	Action
0318160336	Dieqcohr Rufino	3	View
0318160622	A Walk in Customer	2	View
0318170309	Chuchay Jusay	1	View
0318170352	Kimbert Duyag		
0318170511	Dieqcohr Rufino		
0318170524	Dieqcohr Rufino	1	View

Microsoft account problem
We need to fix your Microsoft account (most likely your password changed). Select here to fix it in Shared experiences settings.

Products Page



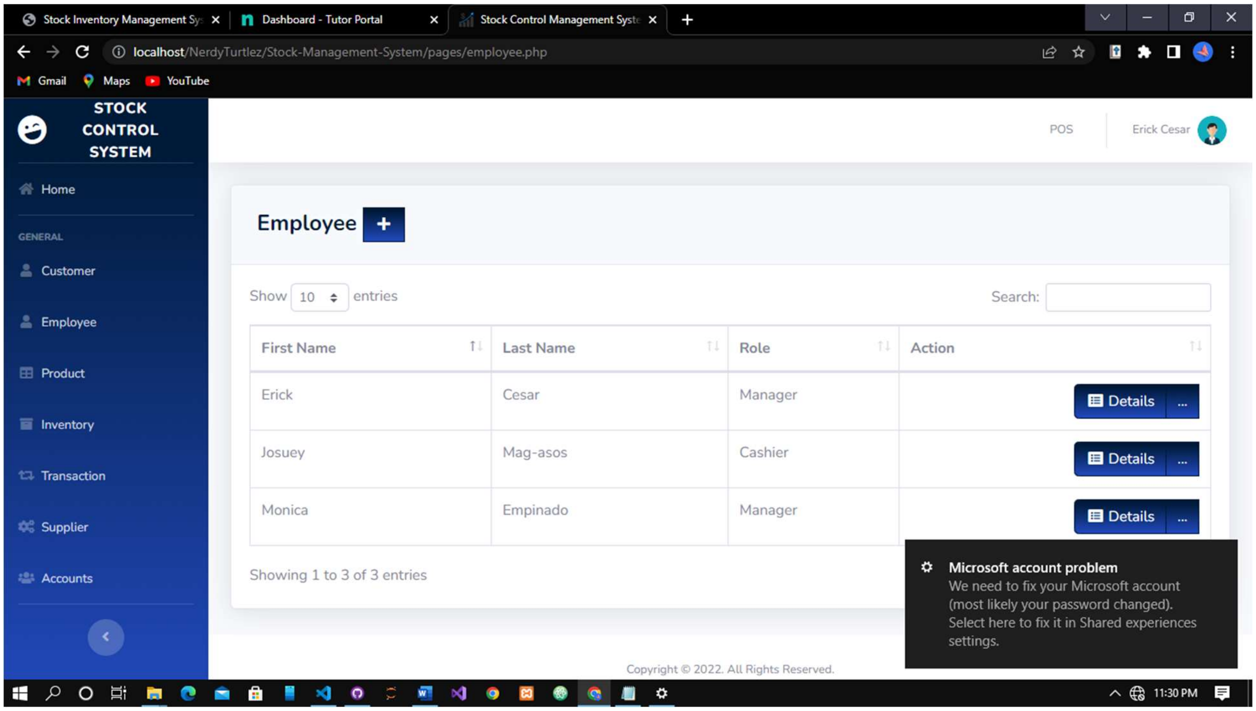
Product +

Show 10 entries Search:

Product Code	Name	Price	Category	Action
20191001	Lenovo ideapad 20059	32999	CPU	Details ...
20191002	Newmen E120	550	Keyboard	Details ...
20191003	Predator Helios 300 Gaming Laptop	77850	CPU	Details ...
20191004	Fantech EG1	859	Hea	
20191005	A4tech OP-720	289	Mou	

Microsoft account problem
We need to fix your Microsoft account (most likely your password changed). Select here to fix it in Shared experiences settings.

Users Page



Employee +

Show 10 entries

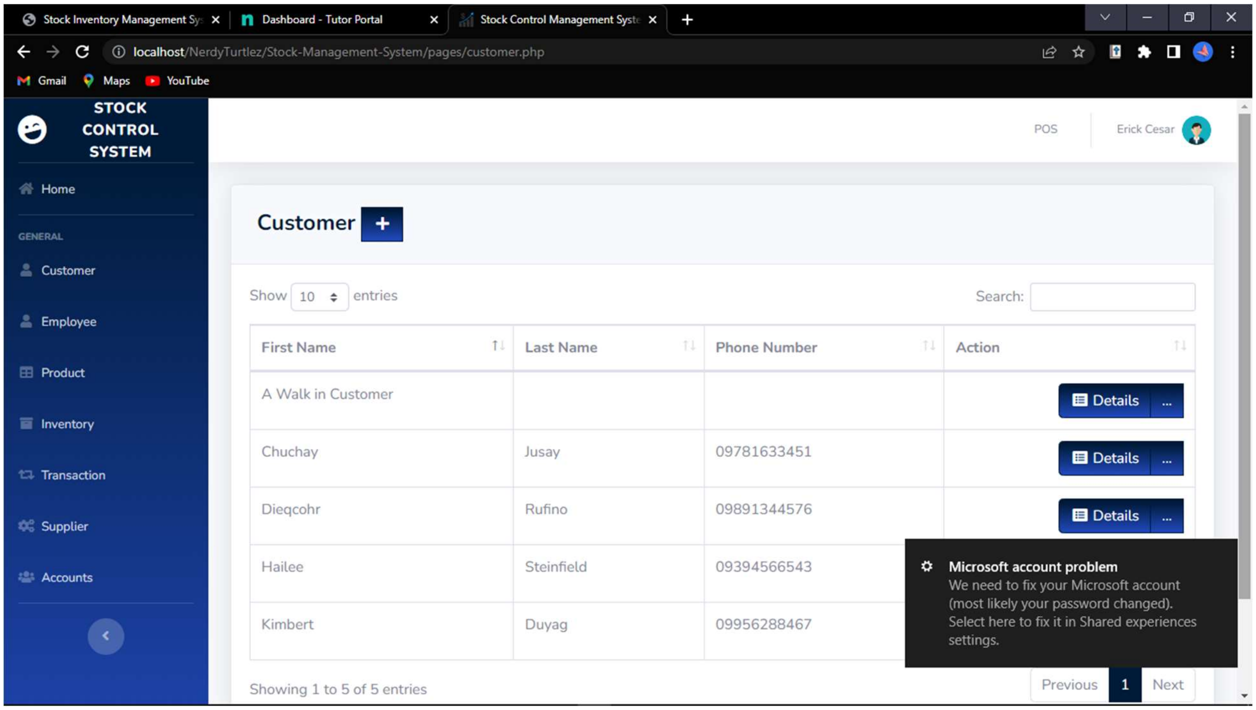
Search:

First Name	Last Name	Role	Action
Erick	Cesar	Manager	Details ...
Josuey	Mag-asos	Cashier	Details ...
Monica	Empinado	Manager	Details ...

Showing 1 to 3 of 3 entries

Microsoft account problem
We need to fix your Microsoft account (most likely your password changed). Select here to fix it in Shared experiences settings.

Copyright © 2022. All Rights Reserved.



Customer +

Show 10 entries

Search:

First Name	Last Name	Phone Number	Action
A Walk in Customer			Details ...
Chuchay	Jusay	09781633451	Details ...
Dieqcohr	Rufino	09891344576	Details ...
Hailee	Steinfield	09394566543	
Kimbert	Duyag	09956288467	

Showing 1 to 5 of 5 entries

Previous 1 Next

Microsoft account problem
We need to fix your Microsoft account (most likely your password changed). Select here to fix it in Shared experiences settings.

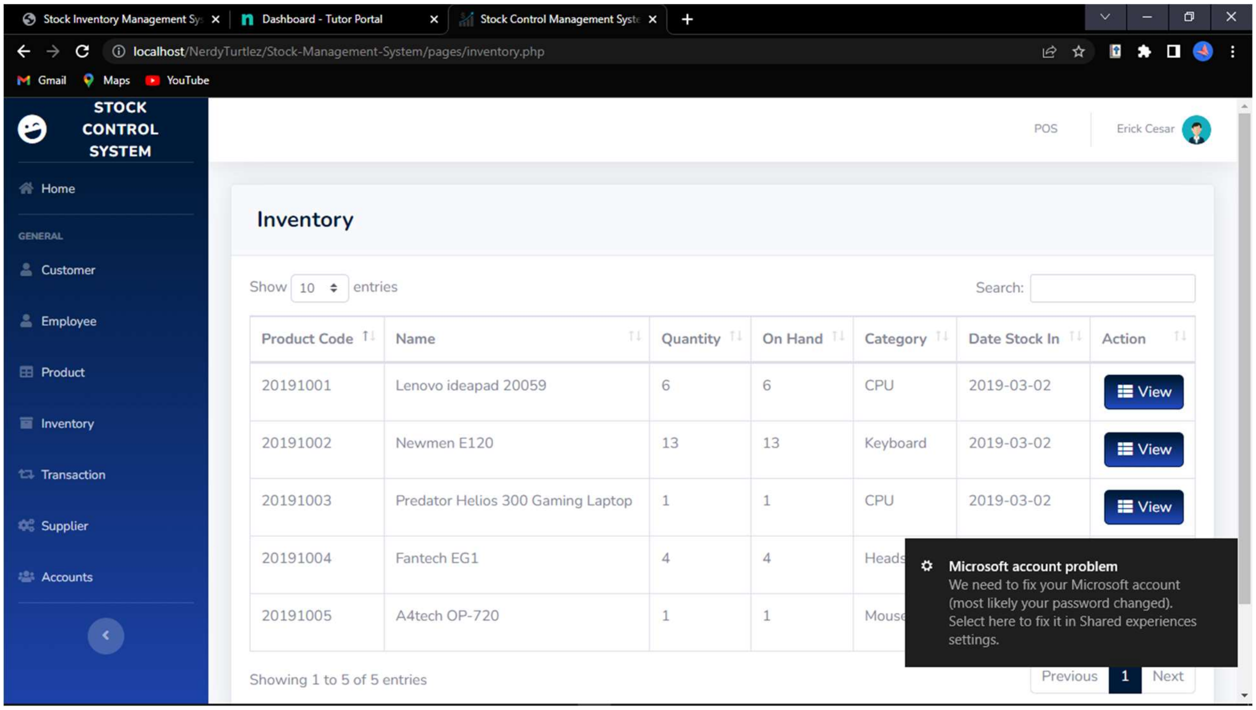
Testing

Testing is integrated into the development phase of an extreme program rather than being done after it. All programs have unit tests to find and fix flaws, and the program must pass all of these tests before it can be released. Client acceptance tests, which are based on customer specifications, are yet another important test. After the coding is finished, acceptance tests are conducted, and the developers show the customer the results as well as demos.

Acceptance Test

User stories, acceptance standards, and use cases are the constant sources of acceptance tests. Only those business tests that need to be verified are represented by these black-box system tests. These should primarily be used to describe product behavior, usage, and flows. The system testing phase in the regression cycles can also take into account the intended acceptance tests in order to boost product confidence before transferring it to the acceptance testing phase.

The web application is hosted on localhost using the Apache Tomcat Server to test the stock and inventory algorithm. The following diagram explains how the stock and inventory algorithm was tested.



Inventory

Product Code	Name	Quantity	On Hand	Category	Date Stock In	Action
20191001	Lenovo ideapad 20059	6	6	CPU	2019-03-02	View
20191002	Newmen E120	13	13	Keyboard	2019-03-02	View
20191003	Predator Helios 300 Gaming Laptop	1	1	CPU	2019-03-02	View
20191004	Fantech EG1	4	4	Heads		
20191005	A4tech OP-720	1	1	Mouse		

Showing 1 to 5 of 5 entries

Microsoft account problem
We need to fix your Microsoft account (most likely your password changed). Select here to fix it in Shared experiences settings.

Conclusion

The technique is what matters most in making sure the system development process goes well. It is wise to employ one of the many various software development approaches available while creating a system. However, picking the appropriate approach is crucial since it will have an impact on how the system development process proceeds. If the proper methodology is applied to create a system, the project can be completed on schedule. Finally, every step of the process is clearly explained to make it easy to understand.

In a system modeling and design, the specifics of the system's data flow and the entities involved are detailed, and every module is completely documented in relation to one another. While the Entity Relationship Diagram (ERD) and database design provide a clear grasp of the system's database and the relationships between each table, the Data Flow Diagram (DFD) design provides insight into the system's flow and straightforward explanations for each flow.

The challenges will be resolved by the stock control management system since all inventory-related information will be recorded in a database, which will be much easier to maintain and update. This system will also help the organization save time and money by automating many inventory-related processes. The project incorporates a database for inventory management, which includes product performance analysis and project-end material cost calculation. The inventory manager no longer needs to spend as much time keeping track of every item in stock because the stock control management system gives them all the information they need and want to grasp inventory-related matters quickly and easily. The inventory manager can use the system to list out all the specifics or to look through all the inventory data with this assistance.

References

Software Engineering Tutorial. (2022). Retrieved July 17, 2022, from Tutorialspoint.com website: https://www.tutorialspoint.com/software_engineering/index.htm

Wikipedia Contributors. (2022, June 8). Software engineering. Retrieved July 17, 2022, from Wikipedia website: https://en.wikipedia.org/wiki/Software_engineering

Bontis, N., Crossan, M. M., & Hulland, J. (2002). Managing An Organizational Learning System By Aligning Stocks and Flows. *Journal of Management Studies*, *39*(4), 437–469. https://doi.org/10.1111/1467-6486.t01-1-00299

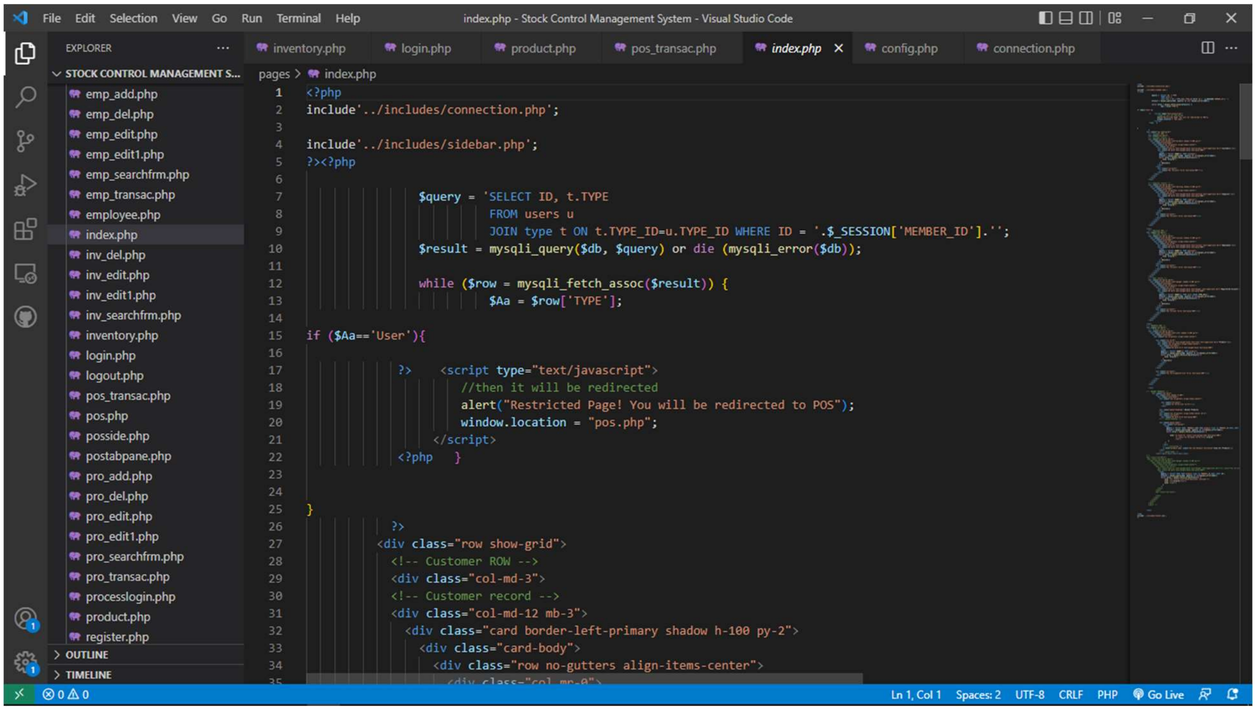
Stock control and inventory. (2022). Retrieved July 17, 2022, from Infoentrepreneurs.org website: https://www.infoentrepreneurs.org/en/guides/stock-control-and-inventory/#:~:text=Stock%20control%2C%20otherwise%20known%20as,raw%20materials%20to%20finished%20goods.

Melanie. (2017, October 17). The 5 Basics of Stock Control & How to Improve it. Retrieved July 17, 2022, from Unleashed Software website: https://www.unleashedsoftware.com/blog/five-basics-stock-control

What is Software Engineering? Definition, Basics, Characteristics. (2020, January 5). Retrieved July 17, 2022, from Guru99 website: https://www.guru99.com/what-is-software-engineering.html

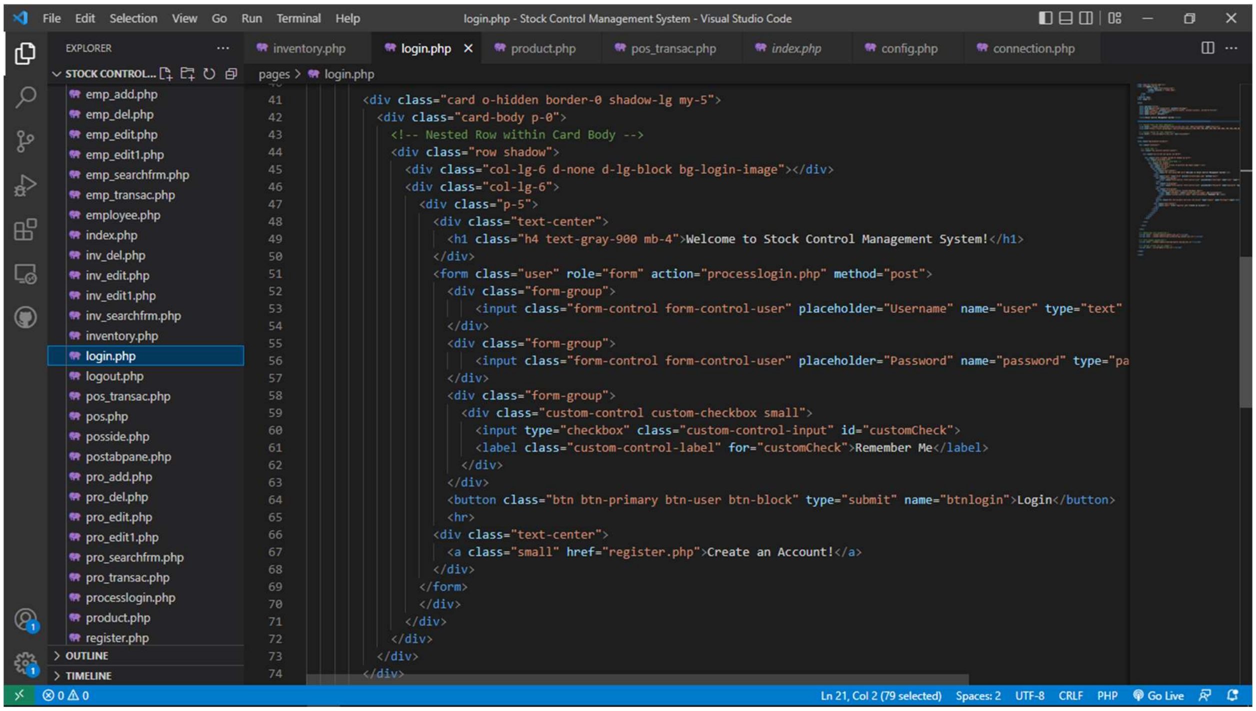
Appendices

Appendix 1: index.php



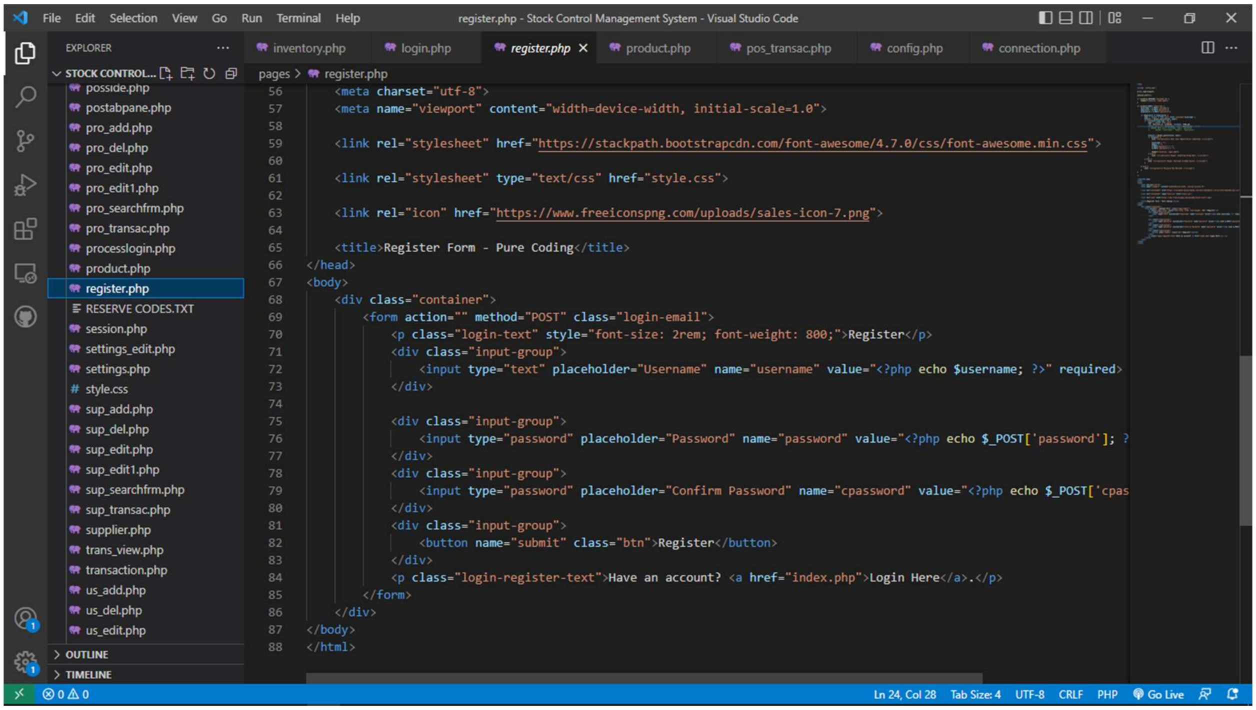
```php
<?php
include'../includes/connection.php';

include'../includes/sidebar.php';
?><?php

                $query = 'SELECT ID, t.TYPE
                          FROM users u
                          JOIN type t ON t.TYPE_ID=u.TYPE_ID WHERE ID = '.$_SESSION['MEMBER_ID'].'';
                $result = mysqli_query($db, $query) or die (mysqli_error($db));

                while ($row = mysqli_fetch_assoc($result)) {
                          $Aa = $row['TYPE'];

if ($Aa=='User'){

              ?>    <script type="text/javascript">
                      //then it will be redirected
                      alert("Restricted Page! You will be redirected to POS");
                      window.location = "pos.php";
                    </script>
              <?php   }


}
            ?>
          <div class="row show-grid">
            <!-- Customer ROW -->
            <div class="col-md-3">
            <!-- Customer record -->
            <div class="col-md-12 mb-3">
              <div class="card border-left-primary shadow h-100 py-2">
                <div class="card-body">
                  <div class="row no-gutters align-items-center">
                    <div class="col mr-0">
```

Appendix 2: login.php



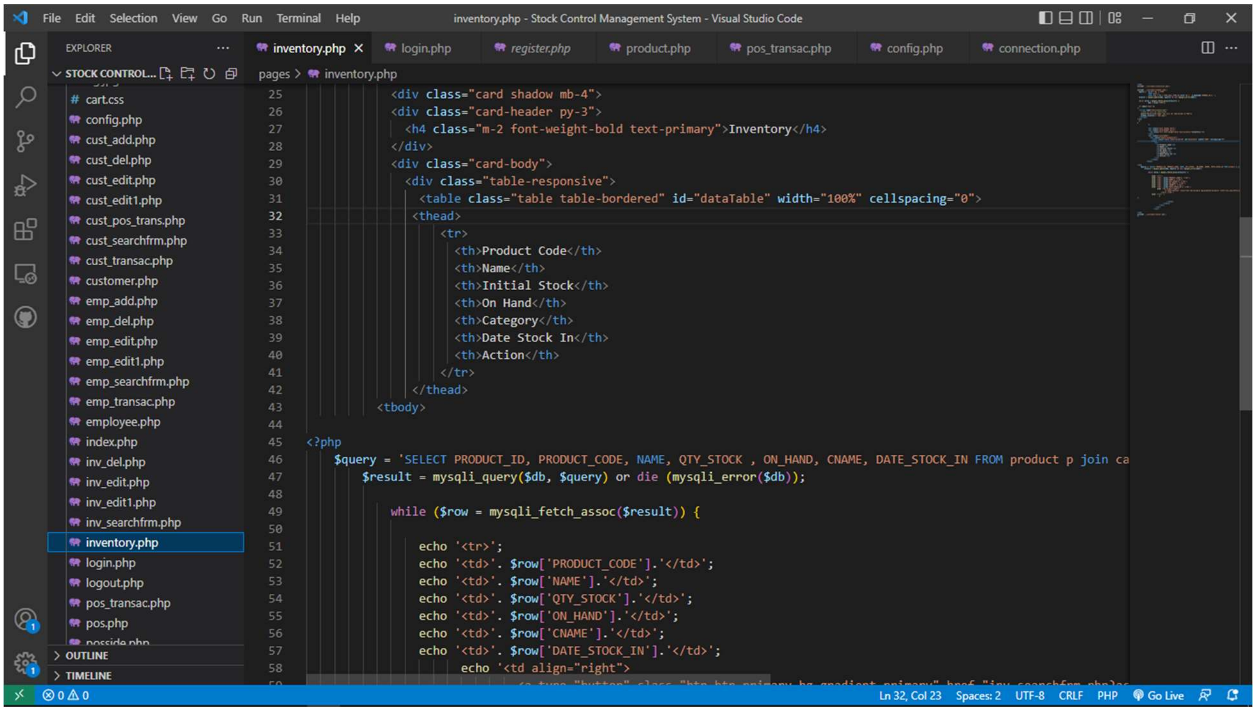
```html
          <div class="card o-hidden border-0 shadow-lg my-5">
            <div class="card-body p-0">
              <!-- Nested Row within Card Body -->
              <div class="row shadow">
                <div class="col-lg-6 d-none d-lg-block bg-login-image"></div>
                <div class="col-lg-6">
                  <div class="p-5">
                    <div class="text-center">
                      <h1 class="h4 text-gray-900 mb-4">Welcome to Stock Control Management System!</h1>
                    </div>
                    <form class="user" role="form" action="processlogin.php" method="post">
                      <div class="form-group">
                          <input class="form-control form-control-user" placeholder="Username" name="user" type="text"
                      </div>
                      <div class="form-group">
                          <input class="form-control form-control-user" placeholder="Password" name="password" type="pa
                      </div>
                      <div class="form-group">
                        <div class="custom-control custom-checkbox small">
                          <input type="checkbox" class="custom-control-input" id="customCheck">
                          <label class="custom-control-label" for="customCheck">Remember Me</label>
                        </div>
                      </div>
                      <button class="btn btn-primary btn-user btn-block" type="submit" name="btnlogin">Login</button>
                      <hr>
                    <div class="text-center">
                      <a class="small" href="register.php">Create an Account!</a>
                    </div>
                  </form>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
```

Appendix 3: register.php



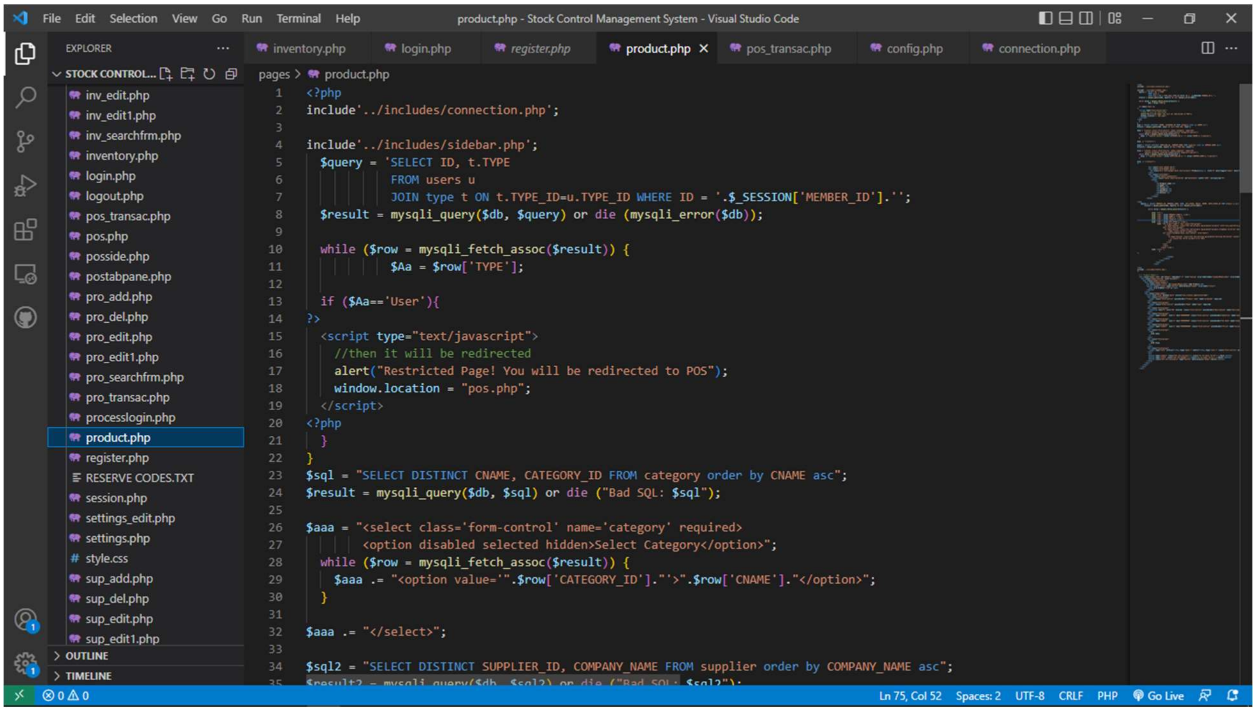
```
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" type="text/css" href="style.css">

    <link rel="icon" href="https://www.freeiconspng.com/uploads/sales-icon-7.png">

    <title>Register Form - Pure Coding</title>
</head>
<body>
    <div class="container">
        <form action="" method="POST" class="login-email">
            <p class="login-text" style="font-size: 2rem; font-weight: 800;">Register</p>
            <div class="input-group">
                <input type="text" placeholder="Username" name="username" value="<?php echo $username; ?>" required>
            </div>

            <div class="input-group">
                <input type="password" placeholder="Password" name="password" value="<?php echo $_POST['password']; ?
            </div>
            <div class="input-group">
                <input type="password" placeholder="Confirm Password" name="cpassword" value="<?php echo $_POST['cpas
            </div>
            <div class="input-group">
                <button name="submit" class="btn">Register</button>
            </div>
            <p class="login-register-text">Have an account? <a href="index.php">Login Here</a>.</p>
        </form>
    </div>
</body>
</html>
```

Appendix 4: inventory.php



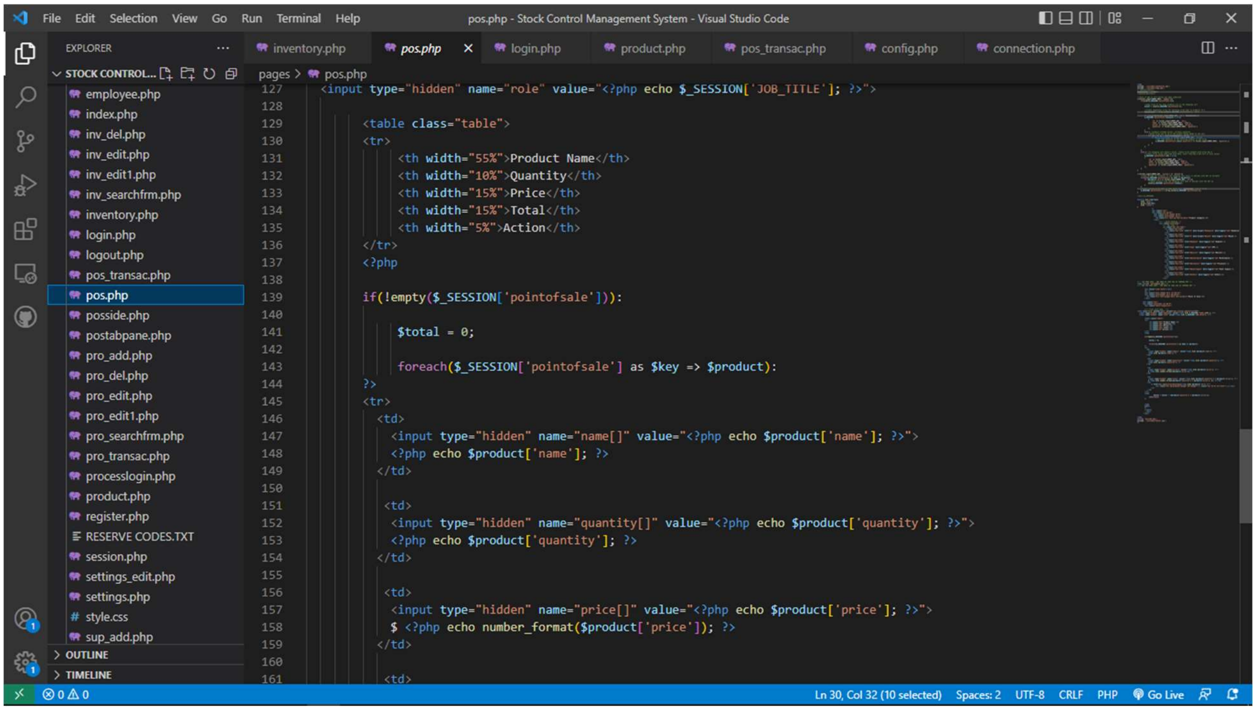
```
            <div class="card shadow mb-4">
            <div class="card-header py-3">
              <h4 class="m-2 font-weight-bold text-primary">Inventory</h4>
            </div>
            <div class="card-body">
              <div class="table-responsive">
                <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
               <thead>
                   <tr>
                     <th>Product Code</th>
                     <th>Name</th>
                     <th>Initial Stock</th>
                     <th>On Hand</th>
                     <th>Category</th>
                     <th>Date Stock In</th>
                     <th>Action</th>
                   </tr>
               </thead>
          <tbody>

<?php
    $query = 'SELECT PRODUCT_ID, PRODUCT_CODE, NAME, QTY_STOCK , ON_HAND, CNAME, DATE_STOCK_IN FROM product p join ca
        $result = mysqli_query($db, $query) or die (mysqli_error($db));

            while ($row = mysqli_fetch_assoc($result)) {

                echo '<tr>';
                echo '<td>'. $row['PRODUCT_CODE'].'</td>';
                echo '<td>'. $row['NAME'].'</td>';
                echo '<td>'. $row['QTY_STOCK'].'</td>';
                echo '<td>'. $row['ON_HAND'].'</td>';
                echo '<td>'. $row['CNAME'].'</td>';
                echo '<td>'. $row['DATE_STOCK_IN'].'</td>';
                      echo '<td align="right">
```

Appendix 5: product.php



```php
<?php
include'../includes/connection.php';

include'../includes/sidebar.php';
  $query = 'SELECT ID, t.TYPE
            FROM users u
            JOIN type t ON t.TYPE_ID=u.TYPE_ID WHERE ID = '.$_SESSION['MEMBER_ID'].'';
  $result = mysqli_query($db, $query) or die (mysqli_error($db));

  while ($row = mysqli_fetch_assoc($result)) {
            $Aa = $row['TYPE'];

  if ($Aa=='User'){
?>
  <script type="text/javascript">
    //then it will be redirected
    alert("Restricted Page! You will be redirected to POS");
    window.location = "pos.php";
  </script>
<?php
  }
}
$sql = "SELECT DISTINCT CNAME, CATEGORY_ID FROM category order by CNAME asc";
$result = mysqli_query($db, $sql) or die ("Bad SQL: $sql");

$aaa = "<select class='form-control' name='category' required>
        <option disabled selected hidden>Select Category</option>";
  while ($row = mysqli_fetch_assoc($result)) {
    $aaa .= "<option value='".$row['CATEGORY_ID']."'>".$row['CNAME']."</option>";
  }

$aaa .= "</select>";

$sql2 = "SELECT DISTINCT SUPPLIER_ID, COMPANY_NAME FROM supplier order by COMPANY_NAME asc";
$result2 = mysqli_query($db, $sql2) or die ("Bad SQL: $sql2");
```

Appendix 6: pos.php



```php
  <input type="hidden" name="role" value="<?php echo $_SESSION['JOB_TITLE']; ?>">

        <table class="table">
        <tr>
            <th width="55%">Product Name</th>
            <th width="10%">Quantity</th>
            <th width="15%">Price</th>
            <th width="15%">Total</th>
            <th width="5%">Action</th>
        </tr>
        <?php

        if(!empty($_SESSION['pointofsale'])):

            $total = 0;

            foreach($_SESSION['pointofsale'] as $key => $product):
        ?>
        <tr>
          <td>
            <input type="hidden" name="name[]" value="<?php echo $product['name']; ?>">
            <?php echo $product['name']; ?>
          </td>

           <td>
            <input type="hidden" name="quantity[]" value="<?php echo $product['quantity']; ?>">
            <?php echo $product['quantity']; ?>
          </td>

           <td>
            <input type="hidden" name="price[]" value="<?php echo $product['price']; ?>">
            $ <?php echo number_format($product['price']); ?>
          </td>

           <td>
```