## **Introduction**

A website application system called asset management covers the asset 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 an asset management system that will display all of the organization's asset 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 system features an admin component. The system includes a general organization profile as well as information on the asset, purchases, and remaining asset as it is displayed in the organization. Along with the specifics of the transaction balance, this application also provides the asset's remaining balance. Each new asset 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 asset 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 asset 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 asset
* To create a program that addresses the requirements of Victoria University on a daily basis.

This application is used to provide information about old and new items as well as the amount of asset 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.

## **Requirements and Modelling**

An asset 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.

### **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.

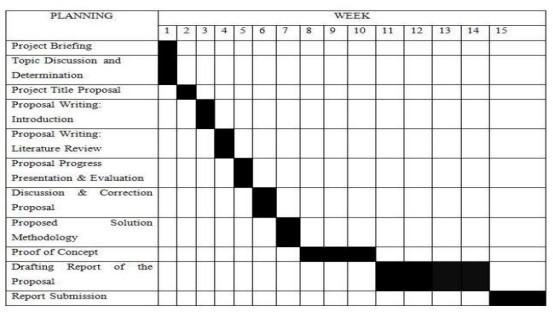
### **Software Requirements**

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

## **Implementation**

### **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.

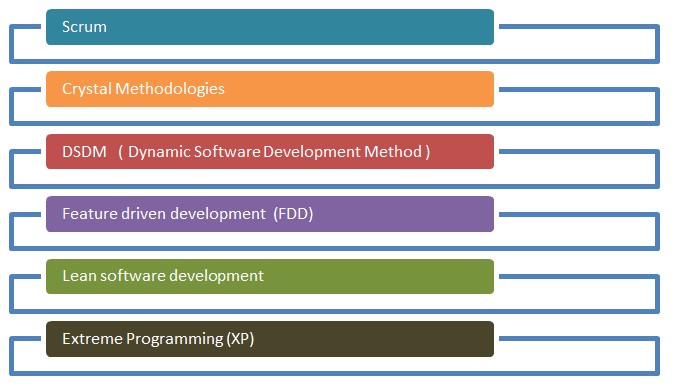


### **Methodology**

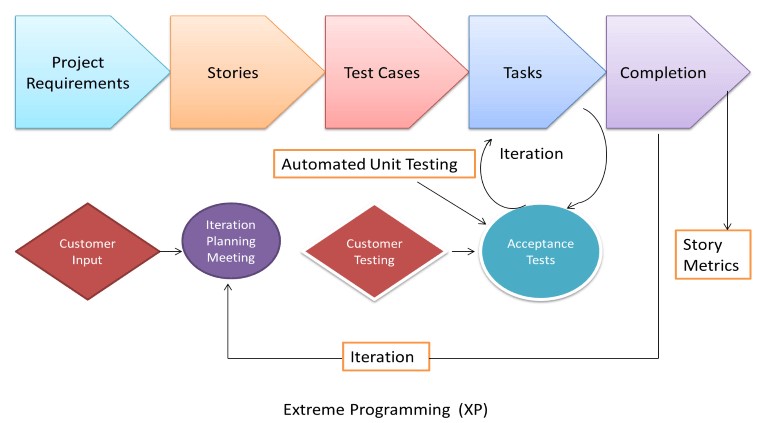
The fundamental guidelines for creating, planning, and also controlling the process of establishing an information system in software engineering are called system development methodologies. The waterfall model, prototyping, spiral development, extreme programming, and other different sorts of techniques were frequently utilized. . Agile software development model technique served as the direction for this system's development. Agile software development is all about moving quickly, releasing frequently, and responding to the actual demands of your users as opposed to traditional software development methods like the Waterfall technique, where we may spend months or years on a project without revealing it to the user.



Agile methodology recognizes our fast changing reality and focuses on people and results in software development. Short delivery times, self-organization, and adaptable planning are at its core. It strives for ongoing quality improvements and is quick and flexible. Iterations, also known as brief time boxes that are used to build software and typically span from one week to one month, are used to reduce risk. Continuous development and testing iterations are encouraged throughout the project's software development lifecycle using the agile methodology. Agile testing uses a variety of methodologies, including scrum, extreme programming (XP), FDD, and others.



### **Method in Agile Methodology**

I've chosen to use extreme programming (XP) as my reference for this project's agile methodology. I chose this approach because extreme programming is a highly useful strategy when dealing with customers that have continually shifting needs or specifications or who are unsure of the system's operation. It promotes quick development cycles and frequent "releases" of the product, which naturally boosts system efficiency and establishes a checkpoint where any client requests may be quickly implemented. Software is created by XP with the intended client in mind. The core principles of XP are frequent releases and rapid development. It makes use of code reviews, pair programming, unit testing, and regular customer contact.

**Planning -** Planning is the first step in the Extreme Programming software development process. Before I began to design this project, I had already identified and met my target users and customers, as well as looked over their problems and personal narratives. This application was created for Common Hardware Sdn. Bhd., a business that keeps its inventory and other items in a warehouse. This application's goal is to assist the manager and staff in allocating the necessary supply or item. In order to cover a smaller portion of the functionality or features needed, I have turned user stories into iterations. Each iteration has four basic phases: designing, coding, testing, and listening.

**Designing -** Designing is the first step in an XP programming iteration. To ensure compatibility while creating the system, use systems metaphors or standards for names, class names, and methods, as well as agreement on standardized styles and formats. During this stage, the Class Responsibilities and Collaboration (CRC) Cards software was utilized. With the help of this program, object-oriented technology is made possible and a break from the conventional procedural approach is achievable. These cards enable every project user and developer to offer ideas, and to combine the finest ideas into the design..

**Framework Design**

|  |  |  |
| --- | --- | --- |
| VISUALIZATION    HYPERTEXT    Raw Data    GRAPHICAL USER INTERFACE    Raw Data    Raw Data    Processing    Processing    Processing    Data    Data    Data    Data And Information Preprocessing    Image Processing    Data Filtering    Add new product      Receive  notification      Request    new product    Receive Feedback    Database | | |
|  | Stock Control Management System |  |

**Framework Design**

**Context Diagram**

Admin/Manager

User

Stock Control Management System

Login

Add New

User

Update Stock List

Add New Stock

View Stock Report

View User

Notification for Low Stock

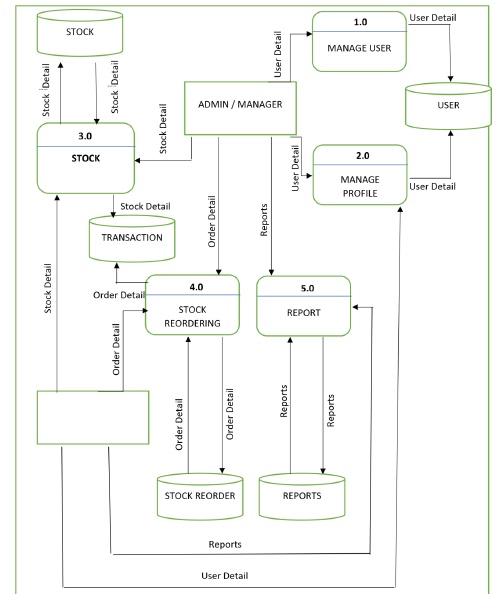
View Stock Report

Scan for Stock Out

Request New Stock

Update Stock Count

**Data Flow Diagram Level 0**

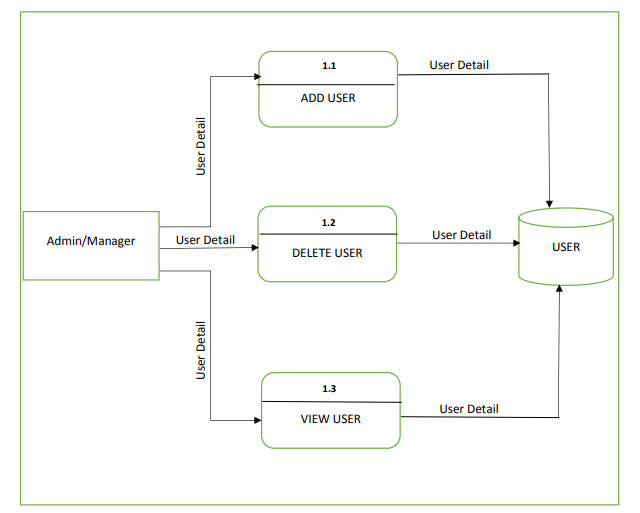


USER

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.

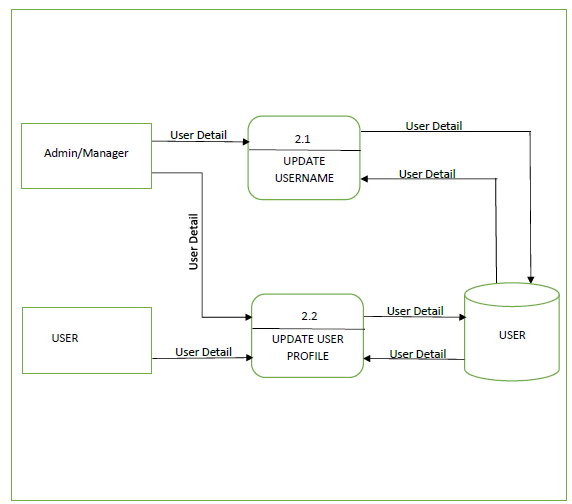
**Data Flow Diagram Level 1 (Process 1.0)**

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



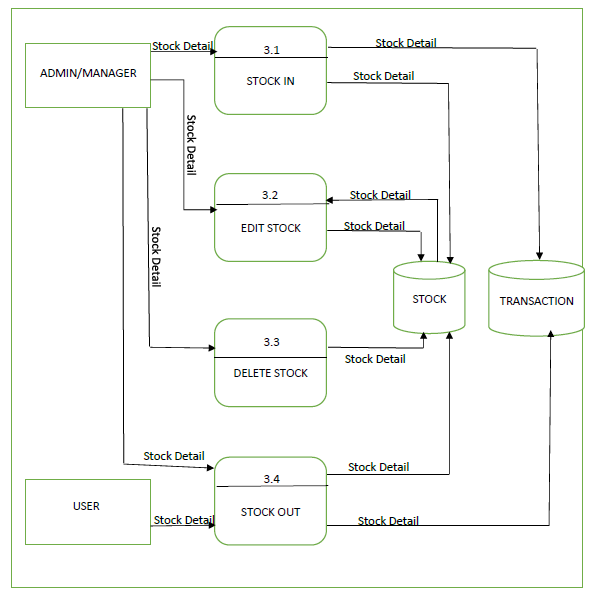
**Data Flow Diagram Level 1 (Process 2.0)**

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



**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.



**Coding** - The most crucial stage of the Extreme Programming life cycle is coding. In order to ensure that the customer receives something of significant value at the end of the day, XP programming prioritizes the actual coding over all other responsibilities, such as documentation. I've adopted a principle of shared code ownership and developed the code using the accepted metaphors and standards. strict observance of 40-hour workweeks without exception. This makes sure the developers are at the top of their physical and mental game. Code integration into the dedicated repository should be done frequently, with only one pair integrating at a time to avoid conflicts, and optimization should be done at the end.

## **Testing**

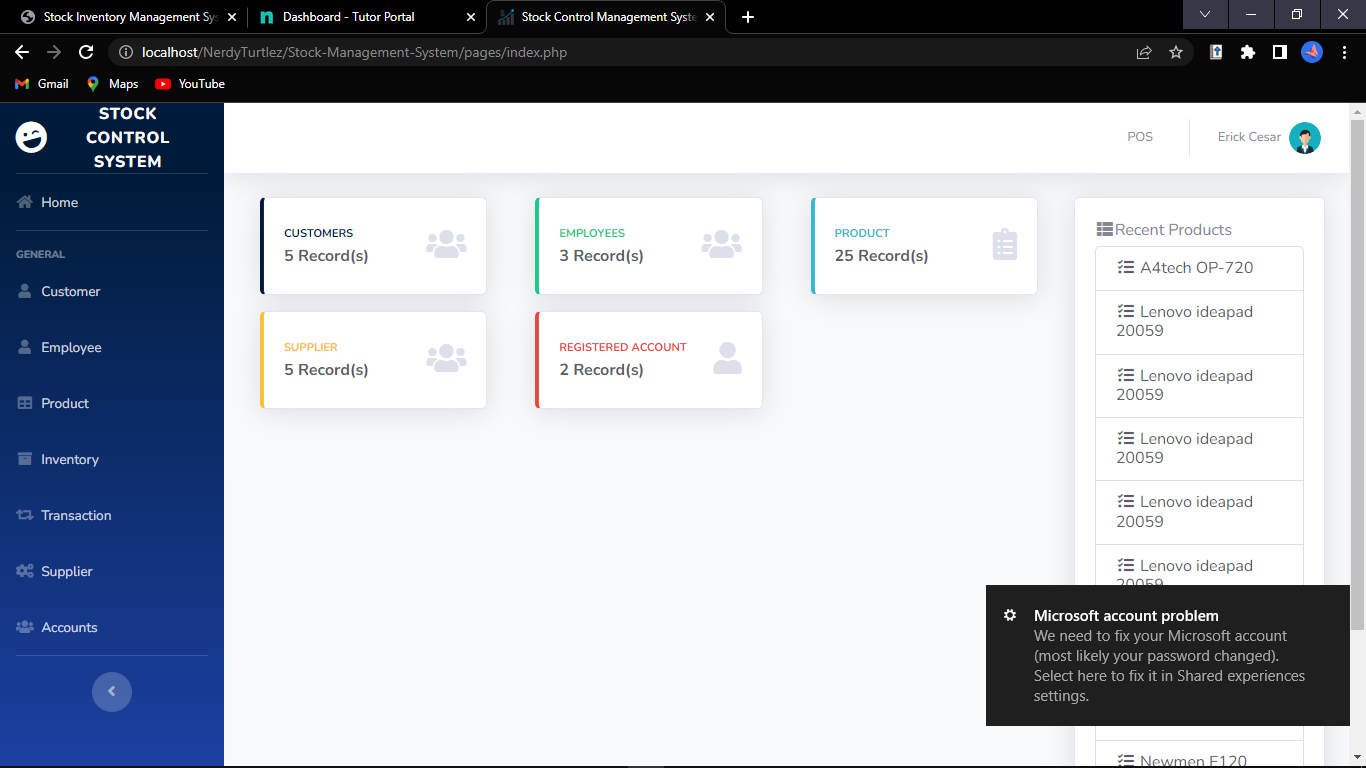
### Testing is integrated into the development phase of an extreme program rather than being done after it. All programs have unit tests to find and fix flaws, and the program must pass all of these tests before it can be released. Client acceptance tests, which are based on customer specifications, 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.

### **Test Cases**

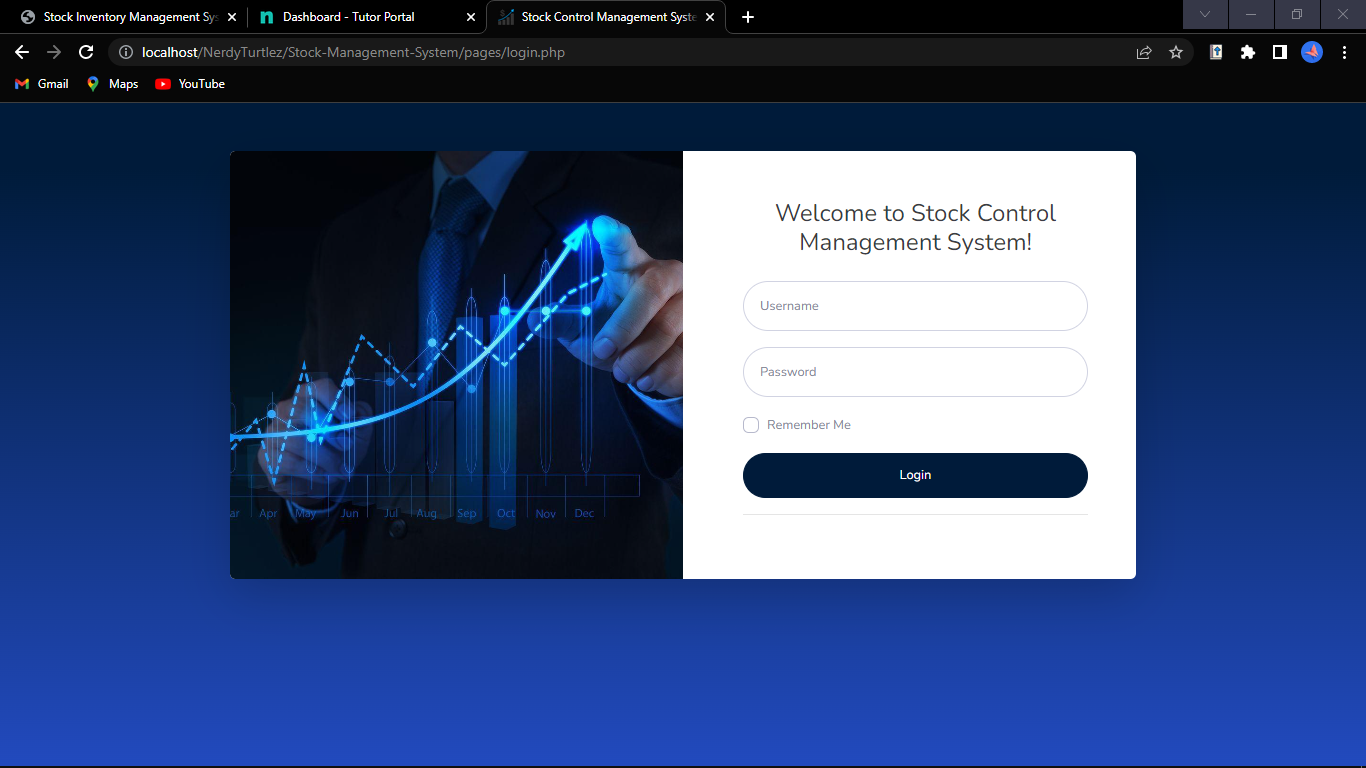
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case ID** | **Input for the test** | **Expected result** | **Actual result** | **Pass/Fail** |
| **Test case 1** | Login= Incorrect Password= something | A window “Incorrect credentials” must be shown to the user and access must not be granted. | A window “Incorrect credentials” is shown to the user and access is not granted. | **Pass** |
| **Test case 2** | Login= admin  (Correct name) | A window “Incorrect credentials” must be shown to the | A window “Incorrect credentials” is shown to the user | **Pass** |
|  | Password=  Incorrect | user and access must not be granted. | and access is not granted. |  |
| **Test case 3** | Login= (blank)  Password=  (blank) | A window “Incorrect credentials” must be shown to the user and access must not be granted. | A window “Incorrect credentials” is shown to the user and access is not granted. | **Pass** |
| **Test case 4** | Login= admin    Password= admin (correct password) | Access must be granted to the user to the main interface. | Access is granted to the user to the main interface. | **Pass** |

### **Screenshots**

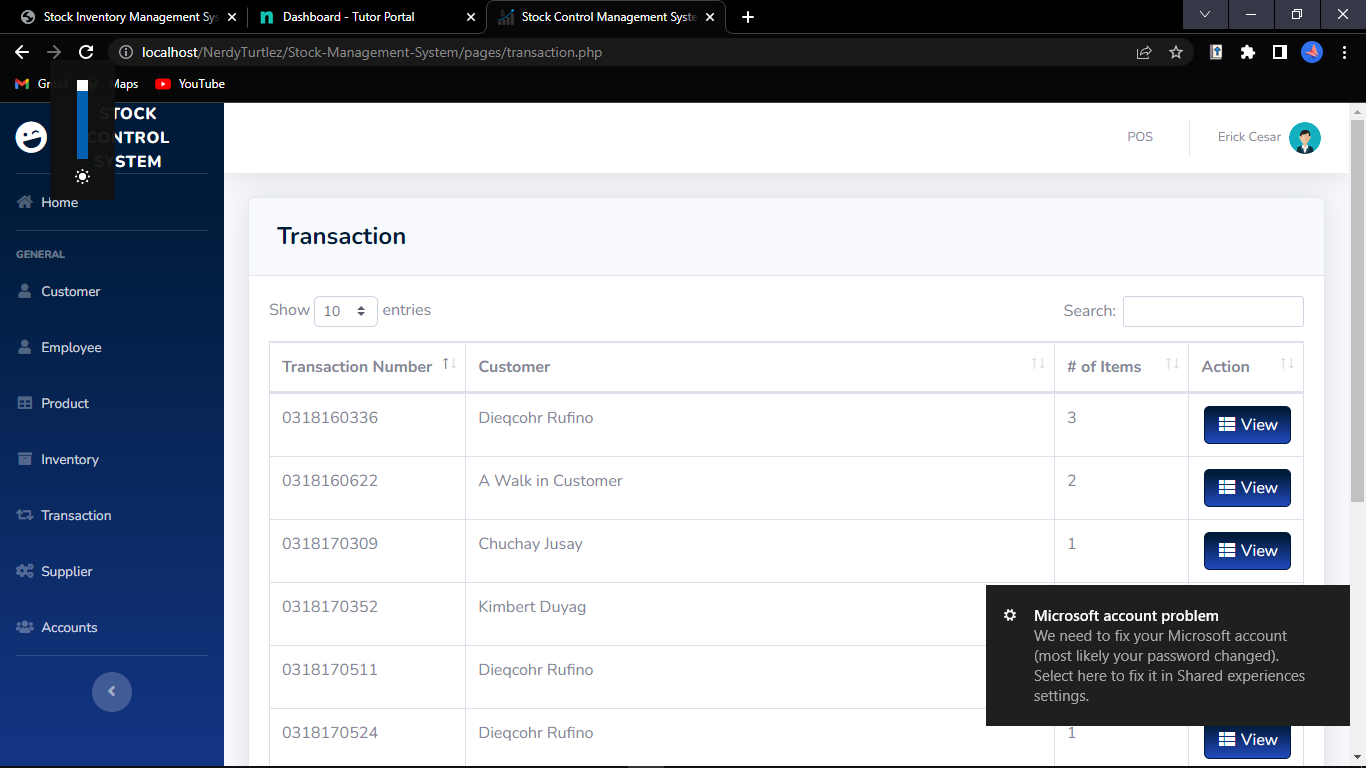
**Dashboard/ Home Page**



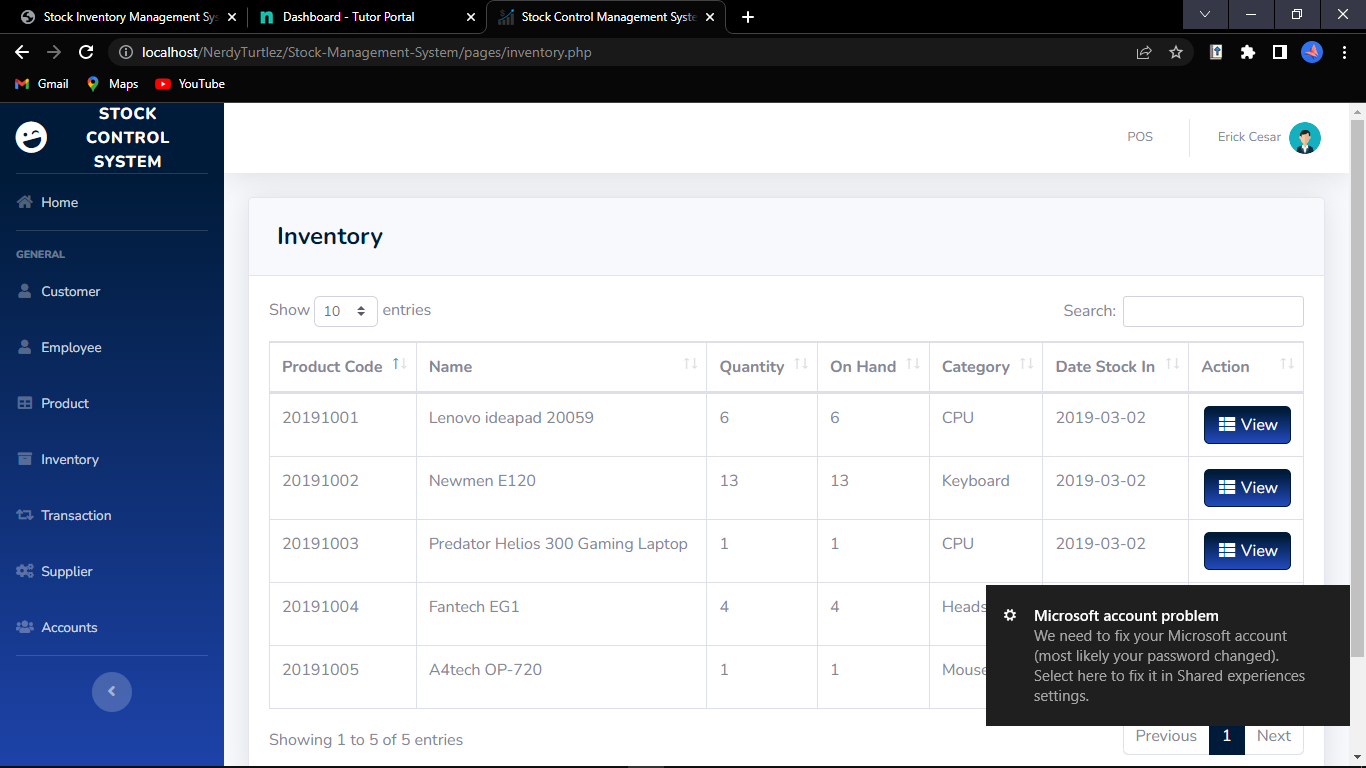
**Login page**



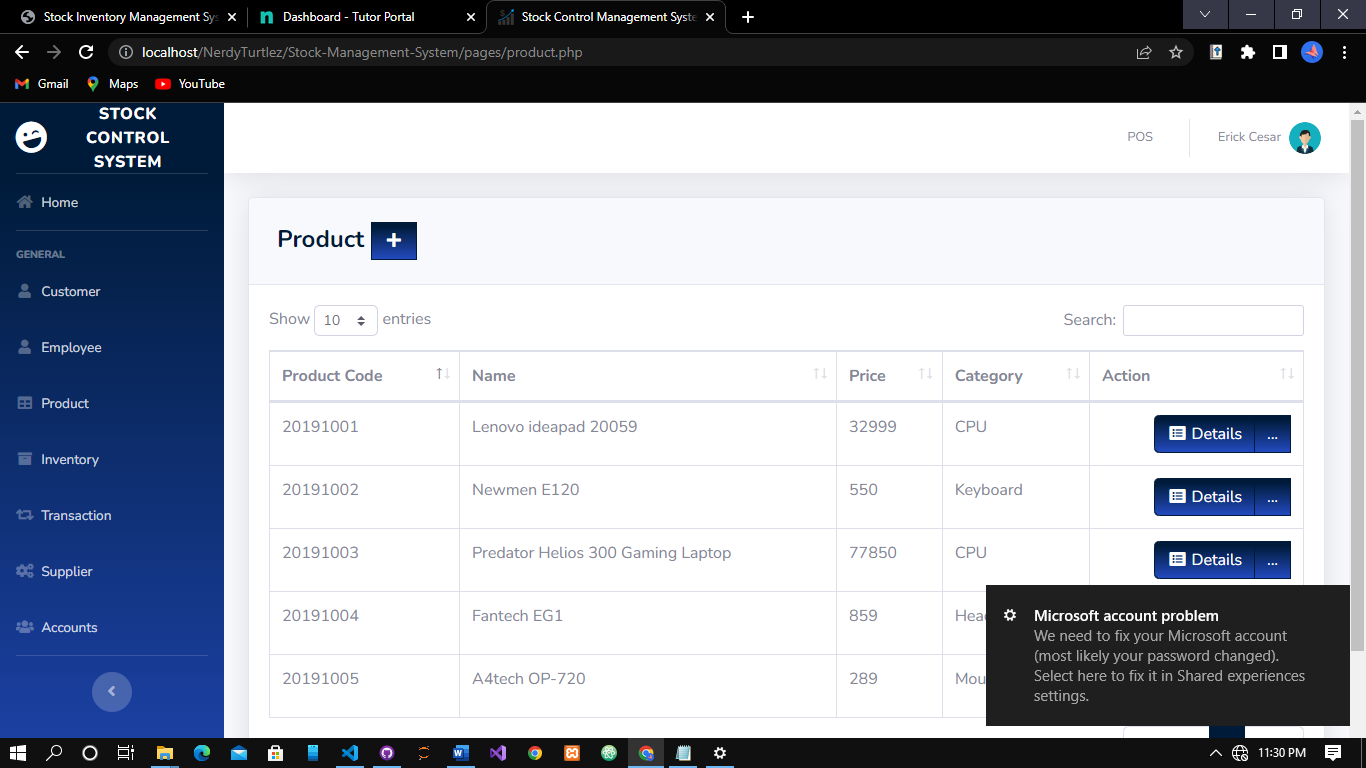
**Transaction Page**



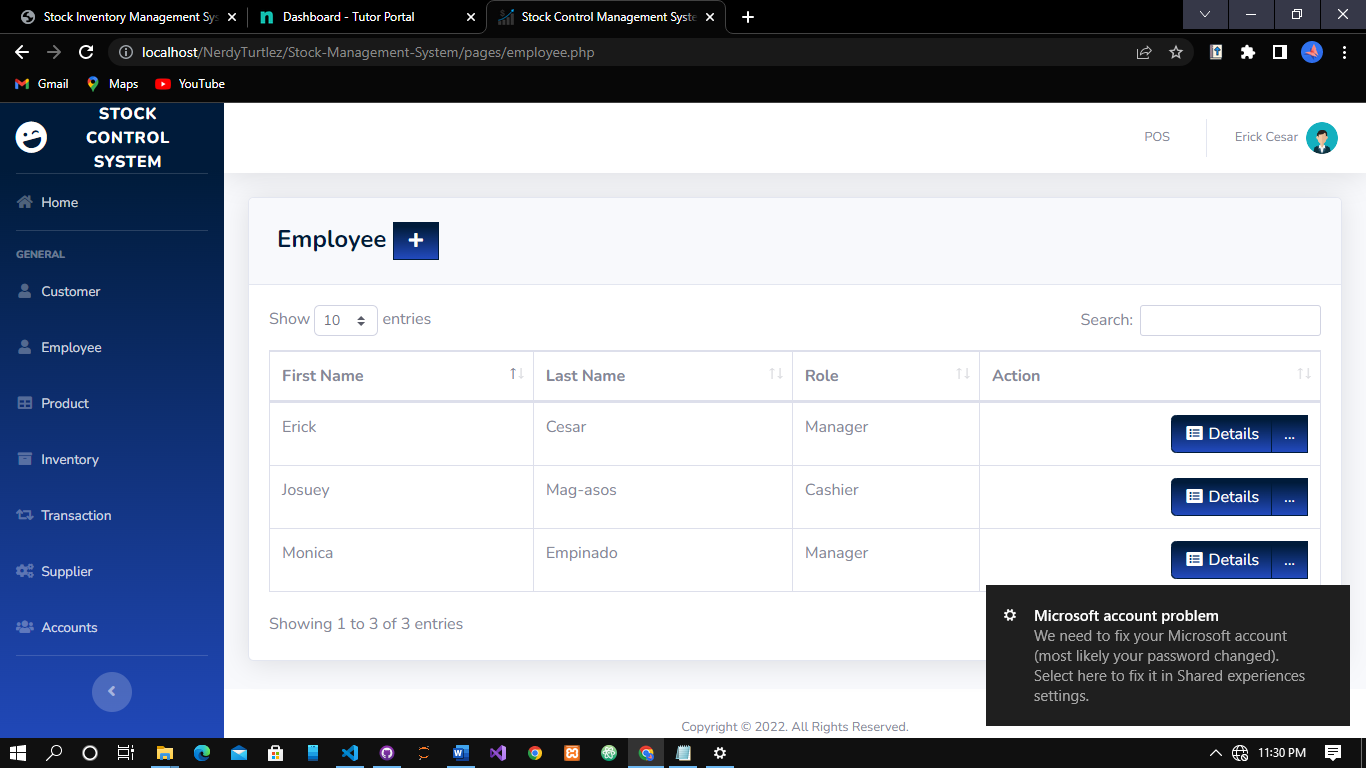
**Inventory Page**

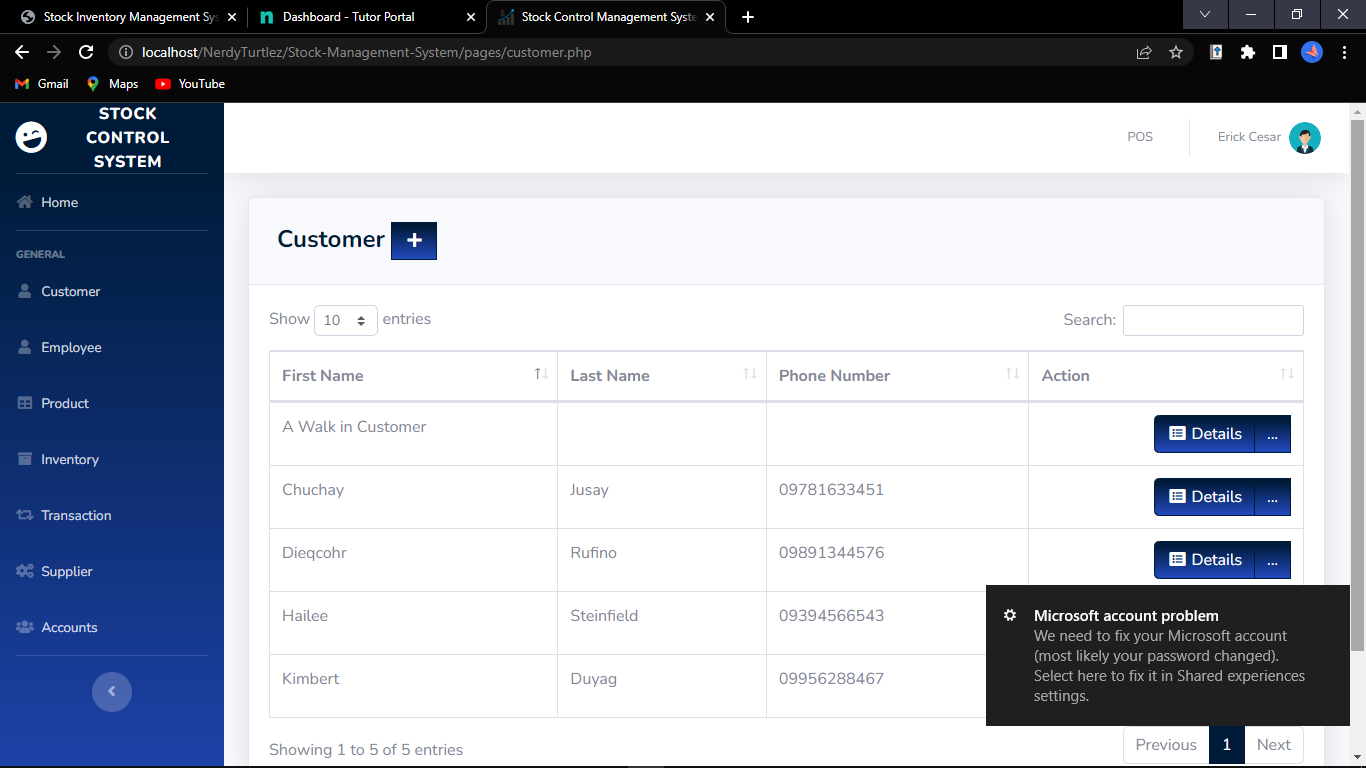


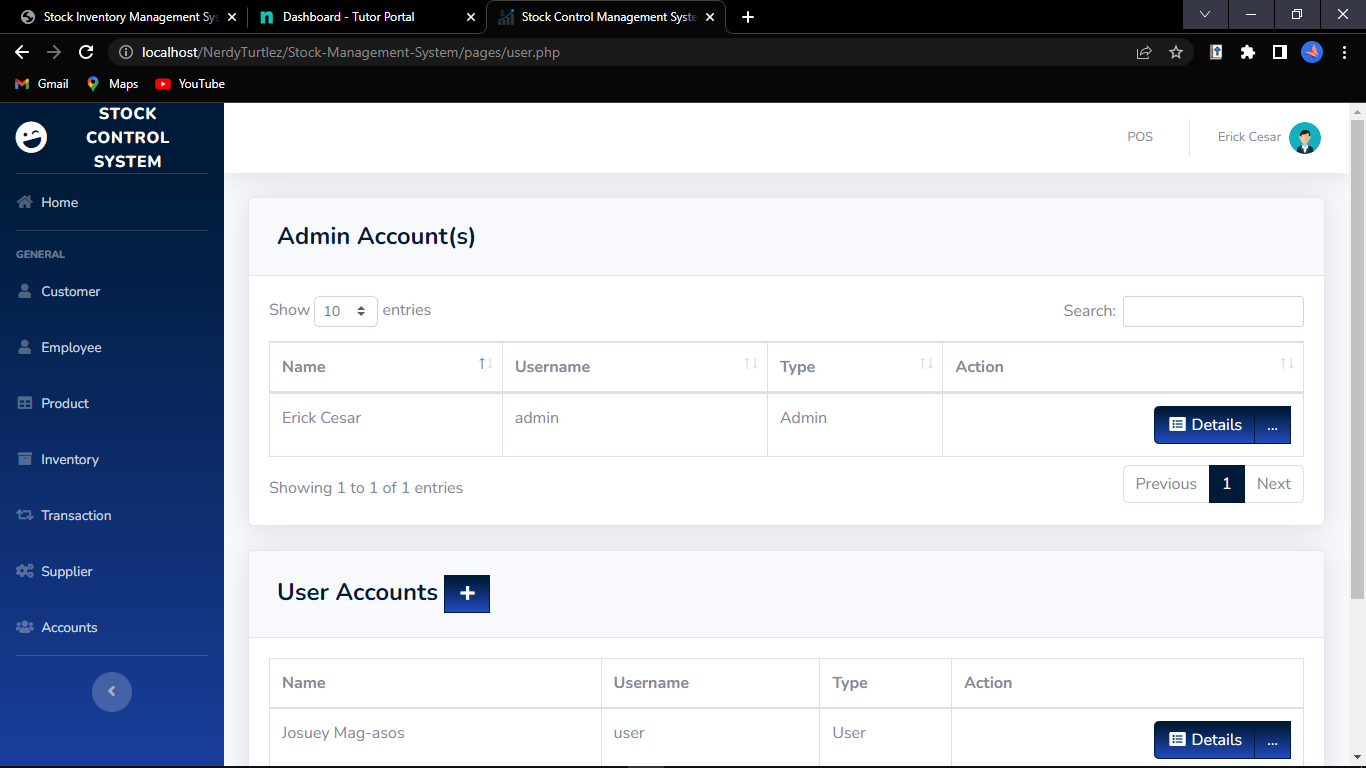
**Products Page**



**Users Page**







## **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, 2023, from Tutorialspoint.com website: https://www.tutorialspoint.com/software\_engineering/index.htm

Wikipedia Contributors. (2022, June 8). Software engineering. Retrieved July 10, 2023, 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 10, 2023, 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 August 7, 2023, from Unleashed Software website: https://www.unleashedsoftware.com/blog/five-basics-stock-control

‌ What is Software Engineering? Definition, Basics, Characteristics. (2020, January 5). Retrieved August 10, 2023, from Guru99 website: https://www.guru99.com/what-is-software-engineering.html

‌