

East West University
Inventory Management System

By

Name: Deener Farjana Binti

ID: 2011-2-60-029

A project submitted in partial fulfillment for the degree of B.Sc. in
Computer Science and Engineering

In the

Faculty of Science and Engineering



Department of Computer Science and Engineering

August 2016

Declaration

We hereby declare that this submission is our own work and that to the best of our knowledge and belief it contains neither material nor facts previously published or written by another person. Further, it does not contain material or facts which to a substantial extent has been accepted for the award of any degree of a university or any other institution of tertiary education except where an acknowledgement.

(Deener Farjana Binti)

Letter of Acceptance

The project entitled Inventory Management System submitted by Deener Farjana Binti (2011-2-60-029), to department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted by the department in fulfillment of requirements for the Award of the Degree of Bachelor of science in Computer Science and Engineering on August, 2016.

Md.Sarwar Kamal

Supervisor

Senior Lecturer

Department of Computer Science and Engineering,
East West University, Dhaka, Bangladesh.

Dr.Mozammel Huq Azad Khan

Chairperson

Professor and Chairperson,

Department of Computer Science and Engineering,
East West University, Dhaka, Bangladesh.

Acknowledgement

First of all thanks to my supervisor **Md. Sarwar Kamal** for providing me this opportunity to test my skill in the possible manner. He enlightened, encouraged and provided me with integrity to transform my vision into reality.

This project would not be possible without the help of my project supervisor, **Md. Sarwar Kamal**, senior Lecturer, Department of Computer Science and Engineering, East West University. He help me to understand all the matters easily which make me create this project. I would like to express my sincere and deep regards to him.

TABLE OF CONTENTS

Contents	Page No
Abstract	
1 Introduction	1
1.1 Introduction and Motivation	2
1.2 Objectives	2
2 Background	
2.1 Inventory Management System	3
2.2 History of Inventory Management System	3
3 System Representation	
3.1 Technical Feature	5
3.2 Overall Steps of Add a new product	5
3.3 Dashboard/Home Page	7
3.4 Item Page	8
3.5 Implementation of each component	8
4 User Manual	
4.1 Process Overview	10
4.2 Work Flow	10
4.3 Configure your workspace	10
4.4 Add New Product	11
4.5 Check-In Item	11
4.6 Check-Out Item	12
4.7 Logs	12
5 Conclusion and Future Work	
5.1 Strength of this System	13
5.2 Drawback of the System	13
5.3 Future work	14
Appendix	15

Abstract

The concept of an Inventory Management System is to promise about approaching for increasing planned reusability in industry. Inventory Management System is enable for planning and strategic decisions of product. Inventory Management System contributes to efficiency and structure of company activities. Due to the absence of an inventory management specific performance measurement evaluation tool, this system is conducted to acquire that knowledge. Still some interesting improvements can be made in this field. ERP claims to provide several improvements at this point and a good performance measurement tool would help to evaluate those statements. Financial and technical product information must be available through the Inventory System, as needed to support the functional responsibilities of personnel within the finance and contracts management departments.

Chapter 1

Introduction

1.1 Introduction and Motivation

Inventories constitute the principal item in the working capital of the majority of trading and industrial companies. In inventory, we include raw materials, finished goods, work in progress, supplies and other accessories. To maintain the continuity in the operations of business enterprise, a minimum stock of inventory required.

However, the physical control of inventory is the operating responsibility of stores superintendent and financial personnel have nothing to do about it but the financial control of these inventories in all lines of activity in which they comprise a substantial part of the current assets is a frequent problem in the management of working capital. Management of inventory is designed to regulate the volume of investment in goods on hand, the types of goods carried in stock to meet the needs of production and sales while at the same time, the investment in them is to kept at a reasonable level.

1.2 Objective

The main objective of our inventory management is to maintain inventory at appropriate level to avoid excessive or shortage of inventory because both the cases are undesirable for business. Inventory Management is one of the basic problems in almost every company. Before computer age and integration, paper tables and paperwork solutions were being used as inventory management tools. These were very far from being a solution, took so much time, even needed employees just for this section of organization. There was no efficient solution available in the many companies during these days. Every process was based on paperwork, human fault rate was high, the process and the tracing the inventory losses were not possible, and there was no efficient logging systems. After the computer age, every process is started to be integrated into electronic environment. And now we have qualified technology to implement new solutions to these problems. Software based systems bring the advantages of having the most efficient control with less effort and employees. These developments provide new solutions for also inventory management systems in this context. In this paper, a new solution for Inventory Management System (IMS) is designed and implemented.

Chapter 2

Background

2.1 Inventory Management System

Inventory Management Systems is a key instrument for businesses when tracking their inventory. Typically, Inventory Management Systems are used by firms that either sell a product or manufacture a product for purposes of accounting for all the tangible goods that allow for a sale of a finished product, or parts for making a product. The size and volume of a firm help dictate whether or not a firm is in need of such a system as they can be quite extensive and costly. Large firms that have thousands of components must have a system in place for the primary objective of tracking their assets. There are three main reasons why an Inventory Management System is needed such as timing/lead time, forecasting, and utilizing economies of scale.

2.2 History of Inventory Management System

I suppose inventory management was first invented by Adam when he named all the animals or by Noah when he counted the clean and unclean beasts for the Ark. But for the sake of brevity, we'll jump ahead to modern times.

Before the Industrial Revolution, merchants basically had to write down all of the products they sold every day. Then they had to order more products based on their hand-

written notes and their gut feelings. This was an incredibly inefficient and inaccurate way of doing business.

Merchants couldn't really account for stolen goods unless they did time-consuming physical counts on a regular basis. They also had trouble making sure they got the right number of products when orders came in because of sparse record keeping. But it was the best they could do.

In the early 1980s, personal computers began to be popular. This further pushed down the cost of barcodes and readers. It also allowed the first versions of inventory management software to be put into place. One of the biggest hurdles in selling readers and barcodes to retailers was the fact that they didn't have a place to store the information they scanned. As computers became more common and affordable, this hurdle was overcome. Once barcodes and inventory management programs started spreading through grocery stores, inventory management by hand became less practical. Writing inventory data by hand on paper was replaced by scanning products and inputting information into a computer by hand.

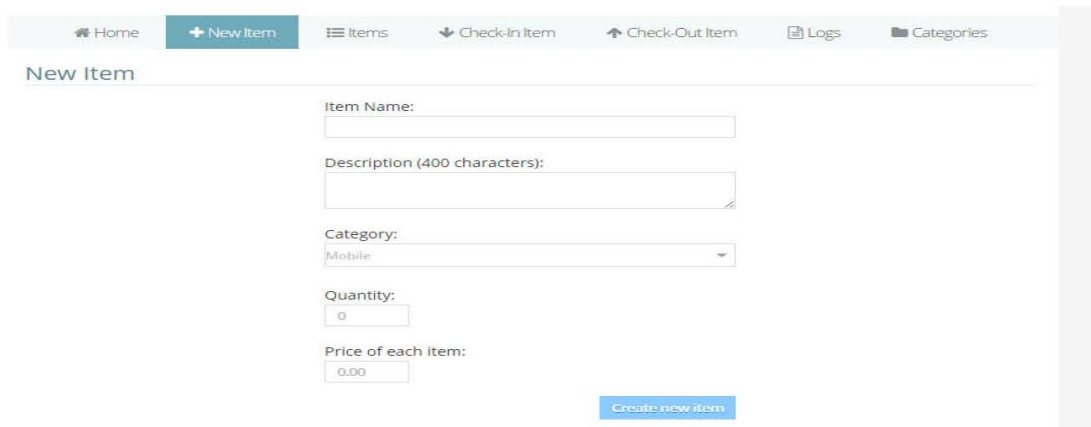
Chapter 3

System Representation

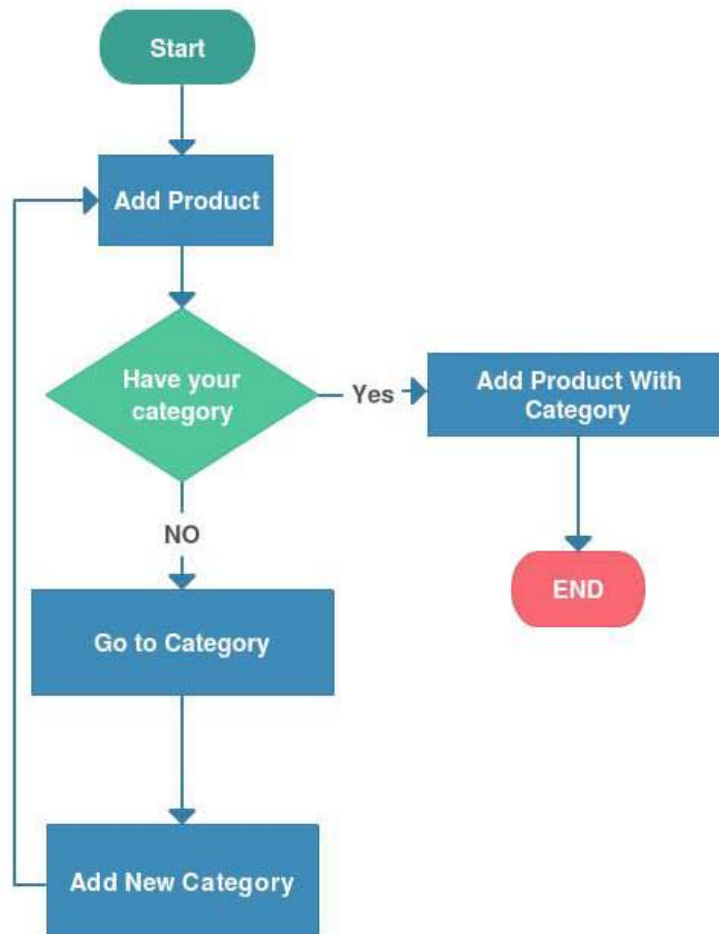
3.1 Technical Feature

1. Programming Language: PHP
2. Database: MySQL
3. Design: HTML and CSS
4. Responsive: Bootstrap

3.2 Overall Steps of Add a new product



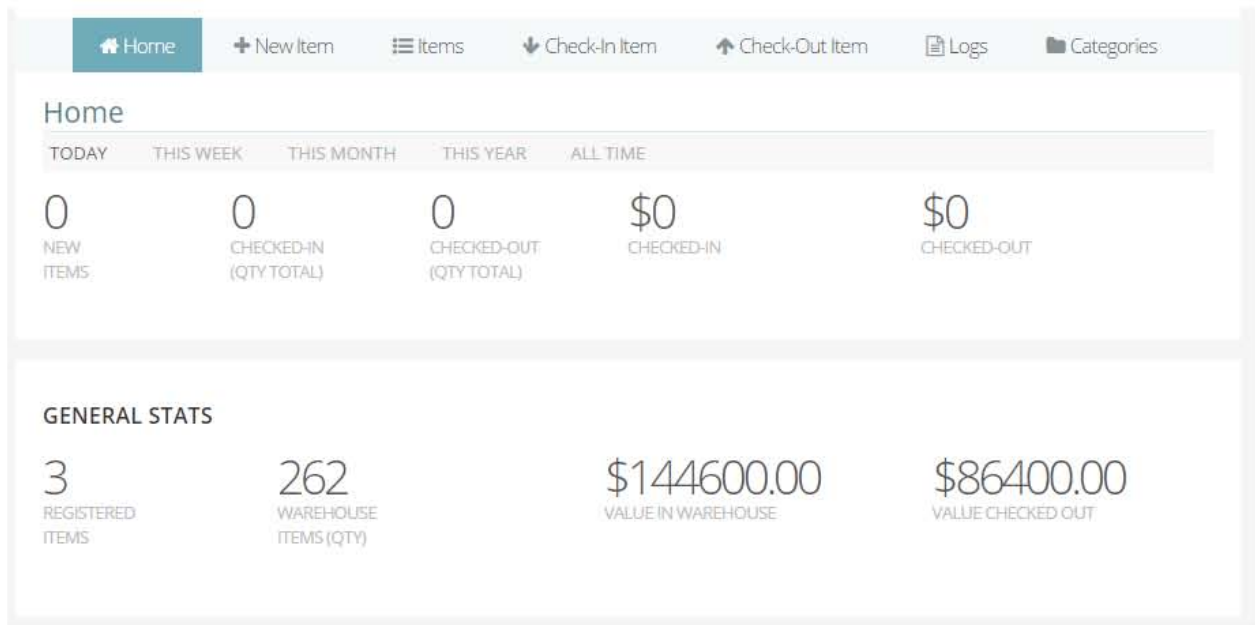
The screenshot shows a web application interface with a navigation bar at the top. The navigation bar contains the following items: Home (with a house icon), New Item (with a plus icon and highlighted in blue), Items (with a list icon), Check-In Item (with a downward arrow icon), Check-Out Item (with an upward arrow icon), Logs (with a document icon), and Categories (with a folder icon). Below the navigation bar, the page title is "New Item". The main content area contains a form with the following fields: "Item Name:" (text input), "Description (400 characters):" (text area), "Category:" (dropdown menu with "Mobile" selected), "Quantity:" (text input with "0" entered), and "Price of each item:" (text input with "0.00" entered). A blue button labeled "Create new item" is located at the bottom right of the form.



Flow Chart Diagram For Add a New Product

The user can create new item for the system through this page. If an user fill up all the text such as item name, description, category, price of each item and click the submit button create new item ,the new item will be created for the system.

3.3 Dashboard/Home Page

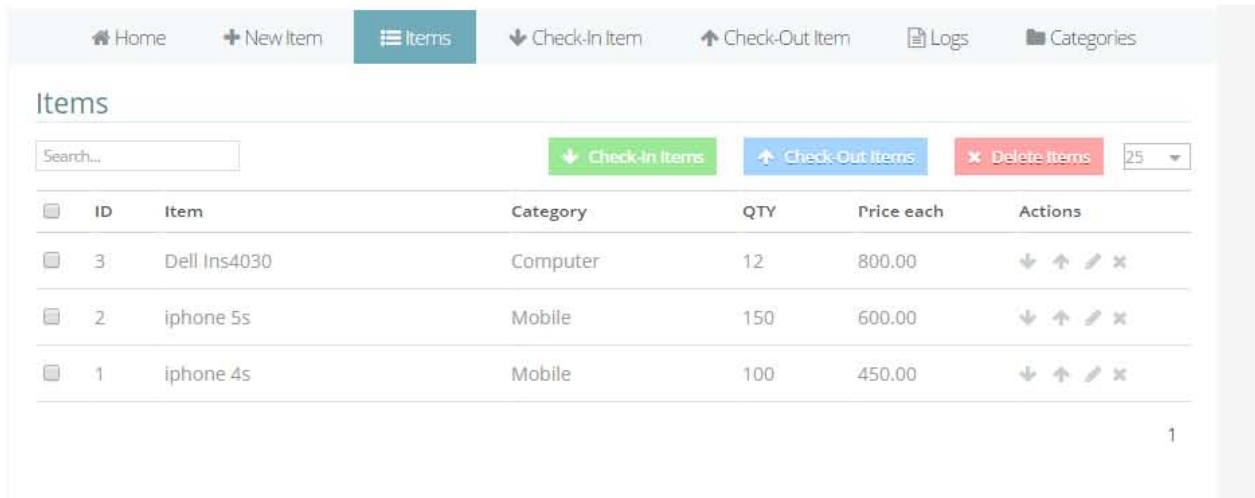


Dashboard of Inventory Management System

The dashboard page state that, today there are no new items entry in the system. There is no item checked in the system. As a result the checked- in item (QTY TOTAL) is 0. Similarly, there is no item checked out in the system. As a result the checked –out (QTY TOTAL) represents 0. So the checked- in value represent \$0 and the checked- out value also represent \$0.

The general stats represent already 3 registered items exists in the system. The general stats also represent total quantity of the warehouse items is 262. Similarly the general stats represent the total value in warehouse is \$144600.00 and value checked out is \$86400.00.

3.4 Item Page



Item page for inventory management system

Items menu represents the items id, item name, category of the items, total quantity of the items, price of each items and actions of each item. The user can check what type of items exists in the system, what type of category items exists in the system, total quantity of the each items and price of each items. The user can also check in items, check out items, edit items and delete the items.

3.5 Implementation of each component

	#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	id	int(11)			No	None
<input type="checkbox"/>	2	name	varchar(200)			No	None
<input type="checkbox"/>	3	place	varchar(200)			No	None
<input type="checkbox"/>	4	descrip	varchar(400)			No	None
<input type="checkbox"/>	5	date_added	date			Yes	0000-00-00

Category Table

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(200)			No	None	
<input type="checkbox"/>	3 descr	varchar(400)			No	None	
<input type="checkbox"/>	4 category	int(11)			No	None	
<input type="checkbox"/>	5 qty	int(11)			No	None	
<input type="checkbox"/>	6 price	decimal(15,2)			No	None	
<input type="checkbox"/>	7 date_added	date			Yes	0000-00-00	

Check all With selected: Browse Change Drop Primary Unique

Item Table

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2 username	varchar(100)			No	None	
<input type="checkbox"/>	3 password	char(32)			No	None	
<input type="checkbox"/>	4 name	varchar(300)			No	None	
<input type="checkbox"/>	5 email	varchar(255)			No	None	
<input type="checkbox"/>	6 role	int(1)			No	None	
<input type="checkbox"/>	7 date_added	date			Yes	0000-00-00	

Check all With selected: Browse Change Drop Primary Unique

User Table

.Chapter-4

User Manual

4.1 Process Overview

To use this system we need a PHP5 supported hosting services. XAMPP is one of the best localhost services. For online use we can maintain any online hosting service.

Sequence for using the software to manage functions:

1. Configure your workspace
2. Login in system
3. Add categories
4. Add Products
5. Check-In/Check-Out Products

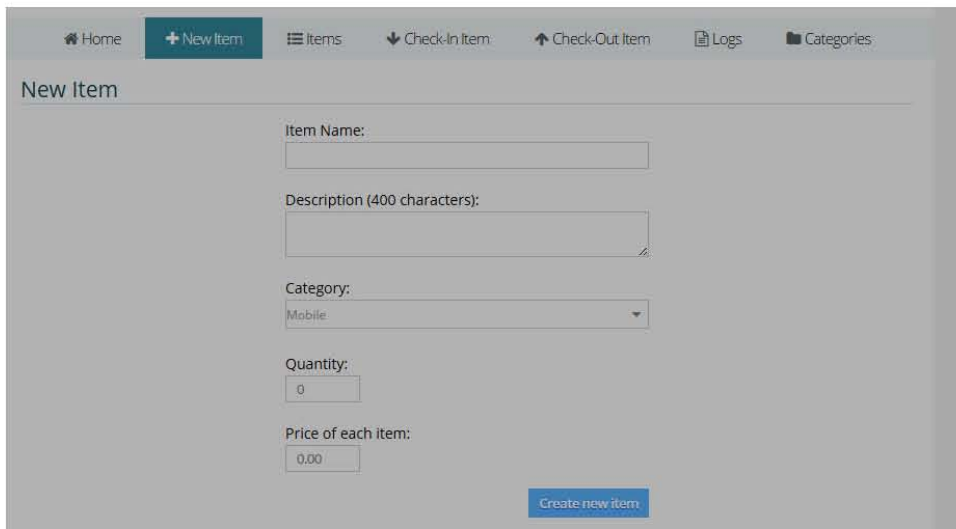
4.2 WorkFlow

Here we describe step by step the procedure to use the tools. How to create the product category. How to add a product and how to check-in and check out.

4.3 Configure your workspace

- Install PHP5 in your web host server.
- Put all source files on root folder.
- Create a database in phpmyadmin mysql database.
- Add the database on phpmyadmin mysql database.
- Configure database username and password with system.

4.4 Add New Product



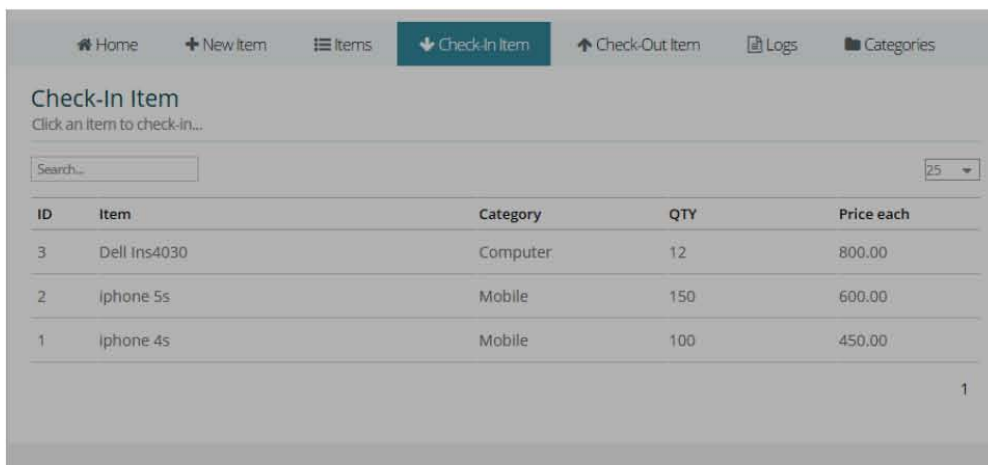
The screenshot shows a web application interface for adding a new item. The navigation bar includes 'Home', '+ New Item', 'Items', 'Check-In Item', 'Check-Out Item', 'Logs', and 'Categories'. The 'New Item' form contains the following fields:

- Item Name:
- Description (400 characters):
- Category:
- Quantity:
- Price of each item:

A blue button labeled 'Create new item' is located at the bottom right of the form.

By clicking New Item user can add a new item with its details.

4.5 Check-In Item



The screenshot shows the 'Check-In Item' page. The navigation bar includes 'Home', '+ New Item', 'Items', 'Check-In Item', 'Check-Out Item', 'Logs', and 'Categories'. The page title is 'Check-In Item' with the instruction 'Click an item to check-in...'. Below the title is a search bar and a dropdown menu showing '25'. The main content is a table with the following data:

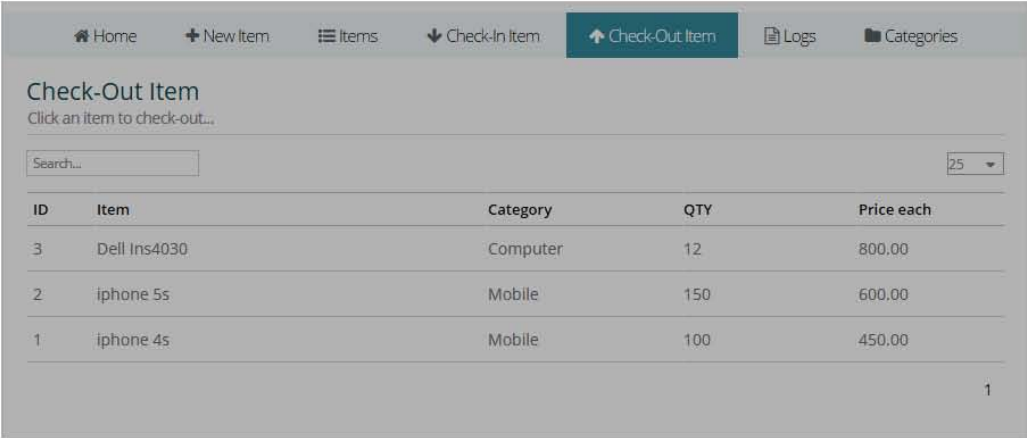
ID	Item	Category	QTY	Price each
3	Dell Ins4030	Computer	12	800.00
2	Iphone 5s	Mobile	150	600.00
1	Iphone 4s	Mobile	100	450.00

A page number '1' is displayed at the bottom right of the table.

Here user can add the quantity of an item.

The check in item menu represent item id, item name, item category, total quantity of each item, price of each item. If a user wants to check in item for the system, click on specific item and number of quantity and click the save button the item will be checked in.

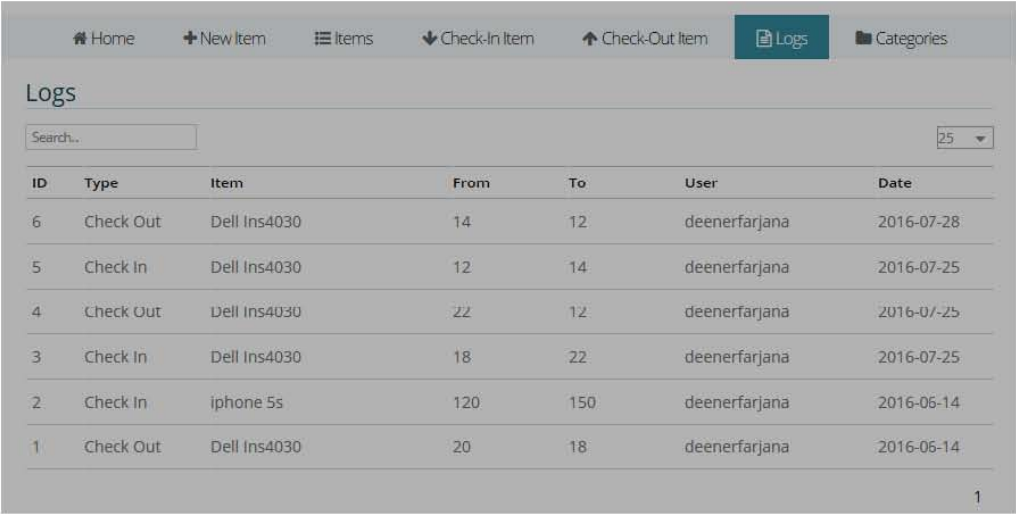
4.6 Check-Out Item



Here user can decrease the quantity of an item.

The check out item menu represent item id, item name, item category, total quantity of each item, price of each item. If an user wants to check out item for the system, click on specific item and number of quantity and click the save button the item will be checked out

4.7 Logs



Here user can get the latest activity of the system and users.

Chapter 5

Conclusion and Future Work

The project “Inventory Management” mainly deals with the calculation of the available and processed resources for an accurate inventory control and process management for a domain specific client who is related to the subject of system. This enables the inventory to be applied at every level.

5.1 Strength of this System

- It can work with any number of products.
- This system is responsive for any kind of devices.
- It provides limited features for individual users as the limit given by the admin.
- As much as the family product remains the system can represent it properly.
- The system is fully dynamic.

5.2 Drawback of the System

- Inventory management system can reduce but cannot eliminate business risk.
- Multiple types of relation without mandatory and optional cannot be implemented on child from same mother variant.

5.3 Future Work

In future I want to work with this project. I want to improve this system. There are some features want to add though the system, they are given below

- We will append internal dependency between features of different mother variants.
- In future we will contrivance all type of logical relation between features of same mother tree.

Appendix

Maintain User Access

```
1. <?php
2.
3. class Users {
4.     private $self_file = 'users_core.php';
5.     private $mysqli = false;
6.     private $session = false;
7.
8.     public function __construct($m) { $this->mysqli = $m; }
9.
10.    public function set_session_obj($obj) { $this->session = $obj; }
11.
12.    public function get_users($page, $items_per_page) {
13.        if($page == 0 || $page == 1)
14.            $x = 0;
15.        else
16.            $x = ($items_per_page * ($page-1));
17.        $y = $items_per_page;
18.
19.        // Admin = ALL Users
20.        // General Supervisors = Employees and Supervisors
21.        // Supervisor = Employees
22.        // Employees = None
23.        $role = $this->session->get_user_role();
24.        if($role == 1)
25.            $q = $this->query("SELECT * FROM invento_users ORDER BY id DESC
LIMIT $x,$y", 'get_users()');
26.        elseif($role == 2)
27.            $q = $this->query("SELECT * FROM invento_users WHERE role=3 OR role=4 ORDER BY id
DESC LIMIT $x,$y", 'get_users()');
28.        elseif($role == 3)
```

```

29.         $q = $this->query("SELECT * FROM invento_users WHERE role=4 ORDER BY id DESC
LIMIT $x,$y", 'get_users()');
30.     else
31.         return false;
32.
33.     return $q;
34. }
35. public function delete_user($userid) {
36.     $prepared = $this->prepare("DELETE FROM invento_users WHERE id=?", 'delete_user()');
37.     $this->bind_param($prepared->bind_param('i', $userid), 'delete_user()');
38.     $this->execute($prepared, 'delete_user()');
39.     return true;
40. }
41.
42. public function new_user($name, $username, $password, $email, $role) {
43.     $prepared = $this->prepare("INSERT INTO
invento_users(username,password,name,email,role,date_added) VALUES(?,?,?,?,?)", 'new_user()');
44.     $this->bind_param($prepared->
>bind_param('sssis', $username, $password, $name, $email, $role, $date), 'delete_user()');
45.     $this->execute($prepared, 'new_user()');
46.     return true;
47. }
48.
49. public function user_exists($username) {
50.     $prepared = $this->prepare("SELECT COUNT(*) as c FROM invento_users WHERE
username=?", 'user_exists()');
51.     $this->bind_param($prepared->bind_param('s', $username), 'user_exists()');
52.     $this->execute($prepared, 'user_exists()');
53.
54.     $result = $prepared->get_result();
55.     $row = $result->fetch_object();
56.     if($row->c >= 1)
57.         return true;
58.     return false;
59. }
60.

```

```

61.     public function userid_exists($userid) {
62.         $prepared = $this->prepare("SELECT COUNT(*) as c FROM invento_users WHERE
        id=?", 'userid_exists()');
63.         $this->bind_param($prepared->bind_param('s', $userid), 'userid_exists()');
64.         $this->execute($prepared, 'userid_exists()');
65.
66.         $result = $prepared->get_result();
67.         $row = $result->fetch_object();
68.         if($row->c >= 1)
69.             return true;
70.         return false;
71.     }
72.
73.     public function edit_user($userid, $name, $email, $role) {
74.         $prepared = $this->prepare("UPDATE invento_users SET name=?, email=?, role=? WHERE
        id=?", 'edit_user()');
75.         $this->bind_param($prepared->
        >bind_param('ssii', $name, $email, $role, $userid), 'edit_user()');
76.         $this->execute($prepared, 'edit_user()');
77.         return true;
78.     }
79.
80.     public function update_user($userid, $name, $email) {
81.         if($name == false && $email == true) {
82.             $prepared = $this->prepare("UPDATE invento_users SET email=? WHERE
        id=?", 'update_user()');
83.             $this->bind_param($prepared->bind_param('si', $email, $userid), 'update_user()');
84.         }elseif($name == true && $email == false) {
85.             $prepared = $this->prepare("UPDATE invento_users SET name=? WHERE
        id=?", 'update_user()');
86.             $this->bind_param($prepared->bind_param('si', $name, $userid), 'update_user()');
87.         }elseif($name == true && $email == true) {
88.             $prepared = $this->prepare("UPDATE invento_users SET name=?, email=? WHERE
        id=?", 'update_user()');
89.             $this->bind_param($prepared->
        >bind_param('ssi', $name, $email, $userid), 'update_user()');

```

```

90.     }
91.
92.     $this->execute($prepared, 'update_user()');
93.     return true;
94. }
95.
96. public function update_pass($userid, $pass) {
97.     $pass = md5($pass);
98.     $prepared = $this->prepare("UPDATE invento_users SET password=? WHERE
    id=?", 'update_pass()');
99.     $this->bind_param($prepared->bind_param('si', $pass, $userid), 'update_pass()');
100.     $this->execute($prepared, 'update_pass()');
101.     return true;
102. }
103.
104. public function parse_role($role) {
105.     if($role == 1)
106.         return 'Administrator';
107.     elseif($role == 2)
108.         return 'General Supervisor';
109.     elseif($role == 3)
110.         return 'Supervisor';
111.     elseif($role == 4)
112.         return 'Employee';
113.     else
114.         return 'Undefined';
115. }
116.
117. public function parse_date($date) {
118.     return date('m/d/Y', strtotime($date));
119. }
120.
121. $_users = new Users($mysqli);

```


To View Product List

```
1. if($_session->isLoggedIn() == false)
2.     header('Location: index.php');
3. $_items->set_session_obj($_session);
4.
5. $_page = 3;
6.
7. $role = $_session->get_user_role();
8.
9. if(isset($_POST['act'])) {
10.     // Search count
11.     if($_POST['act'] == '1') {
12.         if(!isset($_POST['val']) || $_POST['val'] == '')
13.             die('wrong');
14.         $search_string = $_POST['val'];
15.         if($_items->count_items_search($search_string) == 0)
16.             die('2');
17.             die('3');
18.     }
19.
20.     // Delete item
21.     if($_POST['act'] == '2') {
22.         if(!isset($_POST['id']) || $_POST['id'] == '')
23.             die('wrong');
24.
25.         if($role == 3 || $role == 4)
26.             die('wrong');
27.
28.         if($_items->delete_item($_POST['id']) == true)
29.             die('1');
30.             die('wrong');
31.     }
32.
33.     // Update item quantity (check-in/check-out)
34.     if($_POST['act'] == '3' || $_POST['act'] == '4') {
```

```

35.         if(!isset($_POST['id']) || !isset($_POST['val']) || !isset($_POST['fromval']) || $_POST['id']
== '' || $_POST['val'] == '' || $_POST['fromval'] == '')
36.             die('wrong');
37.         if($_POST['act'] == '3')
38.             $type = 1;
39.         elseif($_POST['act'] == '4') {
40.             $type = 2;
41.             $qty = $_items->get_item($_POST['id']);
42.             $qty = $qty->qty;
43.             if($qty < $_POST['val'])
44.                 die('2');
45.         }
46.
47.         if($_items-
>update_item_qty($type, $_POST['id'], $_POST['fromval'], $_POST['val']) == true)
48.             die('1');
49.         die('wrong');
50.     }
51.
52.     // Delete Items
53.     if($_POST['act'] == '5') {
54.         if(!isset($_POST['data']) || $_POST['data'] == '')
55.             die('wrong');
56.
57.         if($role == 3 || $role == 4)
58.             die('wrong');
59.
60.         $decoded = json_decode($_POST['data']);
61.         $deleted = array();
62.         foreach($decoded as $id) {
63.             if($_items->delete_item($id) == true)
64.                 $deleted[] = $id;
65.         }
66.         $reencoded = json_encode($deleted);
67.         if(count($reencoded) == 0)

```

```
68.         die('wrong');
69.         die($reencoded);
70.     }
71.
72.     die();
73. }
74.
75. if(!isset($_GET['page']) || $_GET['page'] == 0 || !is_numeric($_GET['page']))
76.     $page = 1;
77. else
78.     $page = $_GET['page'];
79.
80.
81. if(!isset($_GET['pp']) || !is_numeric($_GET['pp'])) {
82.     $pp = 25;
83. }else{
84.     $pp = $_GET['pp'];
85.
86.     if($pp != 25 && $pp != 50 && $pp != 100 && $pp != 150 && $pp != 200 && $pp != 300 && $pp != 50
87.     0)
```