**MOBILE STORE MANAGEMENT SYSTEM**

**ABSTRACT**

The Mobile Store Management System is developed for desktop systems to facilitate mobile shop owners’ management of customer details and inventory data, which will include mobile phones and accessories. It can be used efficiently for physically separated shops in different locations. This software will provide in a simple and easy to operate user interface, which can be managed by any user without having prior in-depth knowledge of the computer system. One can use this software to get a sales report. Administrators can pull data, from any location from the server.

This software is a complete package for small organizations which will allow them to keep track of their sales and inventory, and provide a computerized billing system. There are various applications with more complex implementation and features available in the market, but they are generally very expensive. Therefore, creating an application with the basic requirement of low cost is essential for small organizations. This application will allow stores to manage customer details, keep inventory of all products and purchase information, in a very simple way, using a state-of the-art software application. It will automatically generate invoices and update inventory.

Mobile shop Management System project ( C#.NET)with billing inventory system is a software application designed to manage sundry types of works included in showrooms like billing, sales, report generation, updating stock details, purchase records for monthly, weekly and yearly.In present trend most of the showrooms like cloth stores, mobile stores, and music stores are utilizing software applications for managing daily activities to reduce manual work and amend standards for data management. In this project we designed software application for mobile show rooms for handling customers purchase details, bill generation for every purchase, maxima transaction details, updating latest stock details to database, and analyzing monthly reports. This software can be utilized in any music, mobile stores, predicated on show room requisites users can integrate more modules.

**1. INTRODUCTION**

The main goal of this project is an managing the mobile store, which can provide the stock details, purchase and sales details which may very helpful to using the clients. In this software reducing man power timing and working. We can finding the stock details systematically. It’s more powerful to use to this software, we can using in this software as user friendly, main purpose of this project is an user friendly application. The project is aimed to develop by **JAVA** as Front end and **MS SQL SERVER** as Back end. The back end is used to store the information in this system.

**1.1 SYSTEM SPECIFICATION**

**1.1.1 HARDWARE SPECFICATION:**

* Processor : P 4 700 GHz.
* RAM : 4GB RAM
* Hard Disk Drive : 40 GB HDD

**1.1.2 SOFTWARE SPECIFICATION:**

* + Operating System : Windows XP/7/8/10
  + Front End : JAVA
  + Back End : MY SQL

1. **SYSTEM STUDY**

**2.1 EXISTING SYSTEM:**

As mentioned above most of details are maintained manually. Due to this the data retrieved is time consuming. Due to human calculation errors occur.

Even when the data is maintained on spreadsheet inconsistency occurs as an order might be missed or wrongly entered or twice.

Data are stride an excel sheet which takes lot of time and data may be corrupted.

**2.1.1 DRAWBACKS:**

The existing system has the following drawbacks.

* As storage and exchange of data is achieved only by use of excel sheets which lack validation capabilities, there is always risk of invalid, inaccurate or incomplete data being fed in computer
* Difficulty in managing multiple forms.
* Lack of security

**2.2 PROPOSED SYSTEM:**

The proposed system is computerized and has been developed using advance language therefore it gives more facilities than present system. It provides quick access to any data. In this system user have to enter the data only once and then it get linked with all files. This reduces the workload of user and it is also a time saving process.

The information about any Subscriber can be easily retrieved. The system maintains all records easy.

**2.2.1 FEATURES:**

* All the information about sale, purchase will be maintain properly in this system.
* All manual calculation of sale or all the money management will be performed by the computer automatically.
* This system will provide timely report information.
* It will produce report for sale, bill information.

1. **SYSTEM DESIGN AND DEVELOPMENT**

**3.1 FILE DESIGN**

The selection of the file system design approach is done according to the needs of the developers what are the needed requirements and specifications for the new design. It allowed us to identify where our proposal fitted in with relation to current and past file system development. Our experience with file system development is limited so the research served to identify the different techniques that can be used. The variety of file systems encountered show what an active area of research file system development is. The file systems may be from one of the two fundamental categories. In one category, the file system is developed in user space and runs as a user process. Another file system may be developed in the kernel space and runs as a privileged process. Another one is the mixed approach in which we can take the advantages of both aforesaid approaches. Each development option has its own pros and cons. In this article, these design approaches are discussed.

**3.2 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
* It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
* will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**3.3 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**3.4 DATABASE DESIGN**

Today's businesses depend on their databases to provide information essential for day-to-day operations, especially in case of electronic commerce businesses who has a definite advantage with up-to-date database access. Good design forms the foundation of any database, and experienced hands are required in the automation process to design for optimum and stable performance.

Software Solutions have been constantly working on these platforms and have attained a level of expertise. We apply proven methodologies to design, develop, integrate and implement database systems to attain its optimum level of performance and maximize security to meet the client's business model.

### Business needs addressed:

* Determine the basic objects about which the information is stored
* Determine the relationships between these groups of information and the objects
* Effectively manage data and create intelligent information
* Remote database administration or on site administrative support
* Database creation, management, and maintenance
* Information retrieval efficiency, remove data redundancy and ensure data security

**3.5 SYSTEM DEVELOPMENT**

**3.5.1 DESCRIPTION OF MODULES**

1. Mobile and accessories Registration
2. Purchase Mobiles and accessories Product
3. Sales Mobiles and accessories Product
4. Billing Details

**Mobile and accessories Registration**

Here owner register mobile models and accessories id which is very helpful to searing the product.

**Purchase Mobiles and accessories Product**

In this module owner will added purchased mobile phones and accessories. User should register before selling that product.

**Sales Product**

This module has used to register the product details before selling. So we can identify the stock of product and billing details.

**Billing Details**

We can calculate billing details based on sales details. This module will generate billing details based on its sales.

1. **SYSTEM TESTING AND IMPLEMENTATION**

**SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

**The objective of testing is as follows:**

* + Testing is the process of executing a program with the intent of finding an error.
  + A successful test is that one of the cover of undiscovered error.

### TESTING ISSUES

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**TESTING METHODOLOGIES**

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfills this objective a series of test step unit, integration. validation and system tests were planned and executed the test steps are:

**System Testing**

Testing is an important phase in project development. System testing makes a logical assumption that if all parts of the system are correct, and the goal will be achieved successfully. The software must meet the user specification and it must satisfy according to the needs of the users.

Testing is the process of executing a project within the intend of finding errors. A good test case is one that has a high probability of finding an undiscovered error.

**Unit Testing**

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

**In Project**, Each module such customer registration module, request module, employee details module, stock module, vehicle module and area detail modules are tested individually for example, Customer details module can contain the more forms to maintain the information so all forms could be tested like entered information store appropriately in database access page or not. If correctly accessed means the testing of registration module successfully completed. Likewise all modules are tested successfully.

**Integration Testing**

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

**In Project,** Integration of two modules can be tested together such as customer registration details and customer login module for verification purposes providing proper accessibility to users. The communication of Registration and Login module can test and executed successfully.

**Validation Testing**

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

**In this project,** Admin login details form Enter without username and password in textbox enter the submit button then Login failed message otherwise checks the both textbox value that is true means valid page displayed. Enter Password Displaying password character \*.if it displays the characters security is not availed so testing of software is failed.

**Output Testing**

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

**In project** All the forms are tested as it gives the necessary output to the user’s search such as view response details.

1. **CONCLUSION**

This project main objective is that we can find or track the stock details in the mobile shop. Admin login and add product details after the product added then only admin can use purchase and sales details. Based on purchase and sales the stock report will be generated.We can also track our daily billing details. Which is very useful to find the every product details.

**FUTURE ENHANCEMENT**

Currently The Mobile store management is fully focused on just maintain stock details, in our future enhancement is an to give an offer to an regular customer in programaticaly, who are purchasing more we will send a gift on every year. This is main concept of our future enhancement.

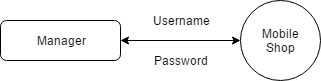
**BIBLIOGRAPHY**

* **Chan98**  
  Patrick Chan, The Java Developers Almanac, Addison-Wesley, 1998.
* **Flanagan96**  
  David Flanagan, Java in a Nutshell, second ed., O'Reilly, 1996.
* **Flanagan99**  
  David Flanagan, Java Foundation Classes in a Nutshell, O'Reilly, 1999.
* **ELW98**  
  Robert Eckstein and Marc Loy and Dave Wood, Java Swing, O'Reilly, 1998.
* **Englander97**  
  Robert Englander, Developing Java Beans, O'Reilly, 1997.
* **Flanagan96**  
  David Flanagan, Java in a Nutshell, second ed., O'Reilly, 1996.

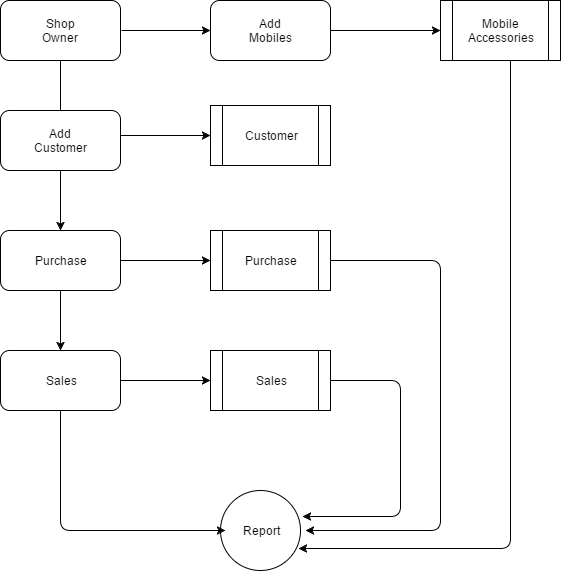
**APPENDICES**

1. **DATA FLOW DIAGRAM**

LEVEL 0:

****

LEVEL 1:

****

1. **TABLE STRUCTURE**

**TABLE NAME : ADMIN**

**PRIMARY\_KEY : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int |  | Admin id |
| Username | Varchar | 30 | Admin username |
| password | Varchar | 30 | Admin password |

**TABLE NAME : CUSTOMER**

**PRIMARY KEY : cid**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Cid | Int | 10 | Customer id |
| Name | Varchar | 30 | Customer name |
| Address | Varchar | 30 | Address |
| Contactno | Varchar | 10 | Contact number |
| Gender | Varchar | 10 | Gender |
| Email | Varchar | 10 | eamil |

**TABLE NAME : PRODUCT**

**PRIMARY KEY : product\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Product\_id | Int | 10 | Product id |
| Company | Varchar | 10 | Company name |
| Model | Varchar | 30 | Model name |
| Price | Int | 10 | price |

**TABLE NAME : PURCHASE**

**PRIMARY KEY : PID**

**FOREIGN KEY : PROID,MID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Pid | Int | 10 | Product id |
| Mid | Int | 10 | Model id |
| Mname | Varchar | 10 | Model name |
| Quantity | Int | 10 | quantity |

**TABLE NAME : SALES**

**PRIMARY KEY : SID**

**FOREIGN KEY : PROID,MID,UID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Pid | Int | 10 | Product id |
| Uid | Int | 10 | User id |
| Mid | Int | 10 | Model id |
| Mname | Varchar | 10 | Model name |
| Quantity | Int | 10 | quantity |

1. **SAMPLE CODING**

<!doctypehtml><html>

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<!-- jQuery library -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css">

<!-- <link rel="stylesheet" href="path/to/font-awesome/css/font-awesome.min.css"> -->

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

</head>

<body>

<div class="container">

<br />

<br />

<br />

<br />

<center> <b id="login-name">Mobile Shop</b> </center>

>

<div class="row">

<div class="col-md-6 col-md-offset-3" id="login">

<form>

<div class="form-group">

<label class="user"> UserName </label>

<div class="input-group">

<span class="input-group-addon" id="iconn"> <i class="glyphicon glyphicon-user"></i></span>

<input type="text" class="form-control" id="text1" name="tl" placeholder="username">

</div>

</div>

<div class="form-group">

<label class="user"> Password </label>

<div class="input-group">

<span class="input-group-addon" id="iconn1"> <i class="glyphicon glyphicon-lock"></i></span>

<input type="text" class="form-control" id="text2" name="tl" placeholder=" Enter Password">

</div>

</div>

<div class="form-group">

<input id="login\_btn" type="button" class="btn btn-success" value="login" style="border-radius:0px;">

</div>

<br /><br /><br />

</form>

</div>

</div>

</div>

<script>

$("#login\_btn").on('click',function(){

if(($("#text1").val()=="admin") && ($("#text2").val()=="admin")){

window.location ="/home";

}else{

alert('Invalid username and password')

}

})

</script>

</body></html>

<html>

<head>

<link href="/bootstrap/bootstrap.css" rel="stylesheet" />

<link href="/bootstrap/datatables.css" rel="stylesheet" />

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

</head>

<body>

<nav class="navbar navbar-findcond navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<a class="navbar-brand" href="#">Mobile Store</a>

</div>

<div class="collapse navbar-collapse" id="navbar">

<ul class="nav navbar-nav navbar-right">

<li><a href="#htab0" data-toggle="tab">Add Mobiles</a></li>

<li><a href="#htab1" data-toggle="tab">Add Customer</a></li>

<li><a href="#htab2" data-toggle="tab">View Customers</a></li>

<li><a href="#htab3" data-toggle="tab">Purchase</a></li>

<li><a href="#htab4" data-toggle="tab">Sales</a></li>

<li><a href="#htab6" data-toggle="tab">Billing</a></li>

<li><a href="#htab5" data-toggle="tab">Stock Details</a></li>

<li><a href="/" >Logout</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="row">

<div class="col-sm-12">

<br>

<div class="tab-content" style="margin: 30px">

<div role="tabpanel" class="tab-pane fade in active" id="htab0">

<h1>Add Mobiles and Accesories</h1>

<div class="col-sm-4">

<div class="form-group">

<label>Company Name:</label>

<input id="add\_company" type="text" class="form-control">

</div>

<div class="form-group">

<label>Model:</label>

<input id="add\_model" type="text" class="form-control">

</div>

<div class="form-group">

<label>Price:</label>

<input id="add\_price" type="text" class="form-control">

</div>

<button id="add\_login" type="button" class="btn btn-success">Add</button>

</div>

</div>

<div role="tabpanel" class="tab-pane fade in " id="htab1">

<h1>Add Customer</h1>

<div class="col-sm-4">

<div class="form-group">

<label>Name:</label>

<input id="cus\_name" type="text" class="form-control">

</div>

<div class="form-group">

<label>Mobile Number:</label>

<input id="cus\_mobile" type="text" class="form-control">

</div>

<div class="form-group">

<label>Address:</label>

<input id="cus\_address" type="text" class="form-control">

</div>

<div class="form-group">

<label>Gender:</label>

<select id="cus\_gender" class="form-control" id="sel1">

<option>Male</option>

<option>Female</option>

<option>Transgender</option>

</select>

</div>

<div class="form-group">

<label>Email:</label>

<input id="cus\_email" type="text" class="form-control">

</div>

<button id="add\_customer" type="button" class="btn btn-success">Add Customer</button>

</div>

</div>

<div role="tabpanel" class="tab-pane fade" id="htab2">

<h1>Customer Details</h1>

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Name</th>

<th scope="col">Mobile Number</th>

<th scope="col">Address</th>

<th scope="col">Gender</th>

<th scope="col">Email</th>

</tr>

</thead>

<tbody id="customer\_body">

</tbody>

</table>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab3">

<h1>Purchase</h1>

<div class="col-sm-4">

<div class="form-group">

<label>Select Model:</label>

<select id="model\_datas" class="form-control" id="sel1">

</select>

</div>

<div class="form-group">

<label>Quantity:</label>

<input id="pur\_quantity" type="text" class="form-control">

</div>

<button id="add\_purchase" type="button" class="btn btn-success">Add Purchase</button>

</div>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab4">

<h1>Sales</h1>

<div class="col-sm-4">

<div class="form-group">

<label>Mobile Number:</label>

<input id="sal\_mobile" type="text" class="form-control">

</div>

<div class="form-group">

<label>Name:</label>

<input id="sal\_name" type="text" class="form-control" disabled>

</div>

<div class="form-group">

<label>Address:</label>

<input id="sal\_address" type="text" class="form-control" disabled>

</div>

<div class="form-group">

<label>Gender:</label>

<input id="sal\_gender" type="text" class="form-control" disabled>

</div>

<div class="form-group">

<label>Mobile Model:</label>

<select id="model\_datas\_sales" class="form-control" id="sel1">

</select>

</div>

<div class="form-group">

<label>Price:</label>

<input id="sal\_price" type="text" class="form-control" disabled>

</div>

<div class="form-group">

<label>Quantity:</label>

<input id="sal\_quantity" type="text" class="form-control" >

</div>

<button id="add\_sales" type="button" class="btn btn-success">Add</button>

</div>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab5">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Company</th>

<th scope="col">Model</th>

<th scope="col">Price</th>

<th scope="col">Quantity</th>

</tr>

</thead>

<tbody id="stock\_body">

</tbody>

</table>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab6">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Name</th>

<th scope="col">Company</th>

<th scope="col">Model</th>

<th scope="col">Quantity</th>

<th scope="col">Amount</th>

<th scope="col">Bill Date</th>

</tr>

</thead>

<tbody id="billing\_body">

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</body>

<script src="/bootstrap/jquery.min.js"></script>

<script src="/home/home.js"></script>

<script src="/bootstrap/datatable.min.js"></script>

<script src="/bootstrap/datatable.bootstrap.min.js"></script>

<script src="/bootstrap/bootstrap.min.js"></script></html>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab5">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Company</th>

<th scope="col">Model</th>

<th scope="col">Price</th>

<th scope="col">Quantity</th>

</tr>

</thead>

<tbody id="stock\_body">

</tbody>

</table>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab6">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Name</th>

<th scope="col">Company</th>

<th scope="col">Model</th>

<th scope="col">Quantity</th>

<th scope="col">Amount</th>

<th scope="col">Bill Date</th>

</tr>

</thead>

<tbody id="billing\_body">

</tbody>

</table>

</div>

</div>

</div>

// Material Design example

$(document).ready(function () {

$("#add\_login").on('click', function () {

var company = $("#add\_company").val();

var mobile = $("#add\_model").val();

var price = $("#add\_price").val();

if (company.length != 0 && mobile.length != 0 && price.length != 0) {

$.ajax({

type: 'POST',

url: '/api/add\_mobiles',

data: { company: company, mobile: mobile, price: price }

}).done(function (datas) {

alert(datas);

$("#add\_company").val("")

$("#add\_model").val("")

$("#add\_price").val("")

update();

});

} else {

alert('Should enter all fields');

}

});

$("#model\_datas\_sales").on('change', function () {

var model\_id = $("#model\_datas\_sales").children(":selected").attr("id");

$.ajax({

type: 'POST',

url: '/api/get\_price',

data: { model\_id: model\_id }

}).

done(function (datas) {

$("#sal\_price").val(datas[0].price);

});

});

$("#sal\_mobile").on('change', function () {

var mobile = $(this).val();

$.ajax({

type: 'POST',

url: '/api/update\_customer',

data: { mobile: mobile }

}).

done(function (datas) {

$("#sal\_name").val(datas[0].name)

$("#sal\_address").val(datas[0].address)

$("#sal\_gender").val(datas[0].gender)

});

});

$("#add\_customer").on('click', function () {

var cus\_name = $("#cus\_name").val();

var cus\_mobile = $("#cus\_mobile").val();

var cus\_address = $("#cus\_address").val();

var cus\_gender = $("#cus\_gender option:selected").html();;

var cus\_email = $("#cus\_email").val();

if (cus\_name.length != 0 && cus\_mobile.length != 0 && cus\_address.length != 0 && cus\_gender.length != 0 && cus\_email.length != 0) {

$.ajax({

type: 'POST',

url: '/api/add\_customer',

data: { cus\_name: cus\_name, cus\_mobile: cus\_mobile, cus\_address: cus\_address, cus\_gender: cus\_gender, cus\_email: cus\_email }

}).done(function (datas) {

alert(datas);

$("#cus\_name").val("");

$("#cus\_mobile").val("");

$("#cus\_address").val("");

$("#cus\_gender").val("");

$("#cus\_email").val("");

showCustomerList();

});

} else {

alert('Should enter all fields');

}

update();

});

$("#add\_purchase").on('click', function () {

var model\_id = $("#model\_datas").children(":selected").attr("id");

var pur\_quantity = $("#pur\_quantity").val();

var model\_name = $("#model\_datas option:selected").html();

if (pur\_quantity.length != 0) {

alert("Purchase added");

update();

$.ajax({

type: 'POST',

url: '/api/add\_purchase',

data: { model\_id: model\_id, pur\_quantity: pur\_quantity, model\_name: model\_name },

}).done(function (datas) {

alert("Purchase added");

update();

});

} else {

alert('Should enter all fields');

}

update();

});

$("#add\_sales").on('click', function () {

var model\_id = $("#model\_datas\_sales").children(":selected").attr("id");

var pur\_quantity = $("#sal\_quantity").val();

var model\_name = $("#model\_datas\_sales option:selected").html();

if (pur\_quantity.length != 0) {

alert("Sales added");

update();

$.ajax({

type: 'POST',

url: '/api/add\_sales',

data: { c\_id: "1", model\_id: model\_id, pur\_quantity: pur\_quantity, model\_name: model\_name },

}).done(function (datas) {

alert("Sales Added");

$("#pur\_quantity").val("");

update();

});

} else {

alert('Should enter all fields');

}

update();

});

update();

});

function update() {

showCustomerList();

updateMobile();

showStockList();

showBillings();

}

function showStockList() {

$.ajax({

type: 'GET',

url: '/api/stock\_list',

}).done(function (datas) {

var html = ``;

for (var i in datas) {

var data = datas[i];

html += ` <tr>

<th scope="row">`+ (++i) + `</th>

<td>`+ data.company + `</td>

<td>`+ data.model + `</td>

<td>`+ data.price + `</td>

<td>`+ data.quantity + `</td>

</tr>`;

}

$("#stock\_body").html(html)

});

}

function updateMobile() {

$.ajax({

type: 'GET',

url: '/api/show\_models',

}).done(function (datas) {

var html = ``;

for (var i in datas) {

var data = datas[i];

html += `<option id=` + data.id + `>` + data.company + ` ` + data.model + `</option>`;

}

$("#model\_datas").html(html)

$("#model\_datas\_sales").html(html)

});

}

function showCustomerList() {

$.ajax({

type: 'GET',

url: '/api/show\_customers',

}).done(function (datas) {

var html = ``;

for (var i in datas) {

var data = datas[i];

html += `<tr>

<th scope="row">`+ (++i) + `</th>

<td>`+ data.name + `</td>

<td>`+ data.mobile + `</td>

<td>`+ data.address + `</td>

<td>`+ data.gender + `</td>

<td>`+ data.email + `</td>

</tr>`;

}

$("#customer\_body").html(html);

});

}

function showBillings() {

$.ajax({

type: 'GET',

url: '/api/bill\_list',

}).done(function (datas) {

var html = ``;

for (var i in datas) {

var data = datas[i];

var date = new Date(data.date);

html += ` <tr>

<th scope="row">`+ (++i) + `</th>

<td>`+ data.name + `</td>

<td>`+ data.company + `</td>

<td>`+ data.model + `</td>

<td>`+ data.quantity + `</td>

<td>`+ (data.quantity-0)\*(data.price-0) + `</td>

<td>`+date.getDate()+"-"+(date.getMonth()+1)+"-"+date.getFullYear()+" "+date.getHours()+":"+date.getMinutes() + `</td>

</tr>`;

}

$("#billing\_body").html(html)

});

}