

Chapter 1: Topic Overview

1.1 Introduction

Inventory management system is a computer-based system for tracking inventory levels, orders, sales and deliveries. It can also be used in the manufacturing industry to create a work order, bill of materials (BOM) and other production-related documents. Companies use inventory management software to avoid product overstock and outages. It is a tool for organizing inventory data that before was generally stored in hard-copy form or in spreadsheets. It is often associated with and is similar to distribution software, as distributors that can compete with less cash tied up in inventories have a distinct advantage over their competitors.

1.2 Why it is used?

Companies often use inventory management software to reduce their carrying costs. The software is used to track products and parts as they are transported from a vendor to a warehouse, between warehouses, and finally to a retail location or directly to a customer. Inventory management software is used for a variety of purposes, including:

- Maintaining a balance between too much and too little inventory.
- Tracking inventory as it is transported between locations.
- Receiving items into a warehouse or other location.
- Picking, packing and shipping items from a warehouse.
- Keeping track of product sales and inventory levels.
- Cutting down on product obsolescence and spoilage.
- Avoiding missing out on sales due to out-of-stock situations.

1.3 Advantages

There are several advantages to using inventory management system. They are:

- **Cost savings:** In many cases, a company's inventory represents one of its largest investments, along with its workforce and locations. Inventory management software helps companies cut expenses by minimizing the amount of unnecessary parts and products in storage. It also helps companies keep lost sales to a minimum by having enough stock on hand to meet demand.
- **Increased efficiency:** Inventory management system often allows for automation of many inventory-related tasks. For example, it can automatically collect data, conduct calculations, and create records. This not only results in time savings, cost savings, but also increases business efficiency.
- **Warehouse organization:** Inventory management system can help distributors, wholesalers, manufacturers and retailers optimize their warehouses. If certain products are often sold together or are more popular than others, those products can be grouped together or placed near the delivery area to speed up the process of picking. With the automated processes, cycle counts will be performed more often and with less effort, increasing inventory visibility, and leading to more accurate fulfillment, fewer out of stock situations and fewer lost sales. More confidence in inventory accuracy will lead to a new focus on optimizing mix, expanding selection and accelerating inventory turns.
- **Data security:** With the aid of restricted user rights, company managers can allow many employees to assist in inventory management. They can grant employees enough information access to receive products, make orders, transfer products and do other tasks without compromising company security. This can speed up the inventory management process and save managers' time.
- **Insight into trends:** Tracking where products are stocked, which suppliers they come from, and the length of time they are stored is made possible with inventory

management system. By analyzing such data, companies can control inventory levels and maximize the use of warehouse space.

Chapter 2: Problem Analysis

“Inventory” when we hear this word we feel like we know a lot about it. It is place where a lot of items of different types are kept. The items may include anything such as electronic items, cosmetic items, grocery items, stationary items and the list continues. Those items need frequent management and maintenance.

2.1 Profile of the Problem

Inventory management can be a very tiresome work for the administrators and managers as there occur a lot of problems such as calculation error, item count error and in some cases, loss of items. The major problem includes slow management, calculation and counting errors. Other problems include no information about items getting low or out of stock. The main focus of this project is to provide reliable and faster management of inventory items and also to remove those problems faced by end-users (administrator).

As mentioned already, inventory is a place that requires a lot of maintenance and management. There is always a chance of error when an inventory manager is managing it manually. Thus, an automated management system can be implemented to remove some load off the inventory managers.

2.2 Product Definition

The product name is “Inventory management system”. This is a software that can assist administrator and managers to manage their inventory. As we already know, an inventory has a lot of items and manually managing everything is very hard. So to make things a lot easier, inventory management system can be used. In this system, there are two types of

users: **Administrator** and **Users**. An administrator can be called “manager”, in this system, who looks over everything from maintaining stock to buying and selling items. As for the users, they are only allowed to sell/buy items and create invoice of the sold or bought product.

2.3 Project Plan

- **Scope of project**

Since this project is related to inventory, it can be used in any type of organization that has an inventory. Initially, this project was targeted for supermarkets and stores. But later during development, the product was made compatible for any type of inventory which increased the product's scope.

- **Requirements**

- a. Invoice Management
- b. Maintain stock
- c. Detailed records
- d. Security
- e. Data Integrity

- **Methodology**

Since this project is minor project, methodology used is waterfall model which involves following steps:

- a. **Feasibility study:** It is done to know and identify whether or not the system is feasible to develop (or worth developing it).

- b. Requirement analysis:** It is done to analyze the needs and requirements gathered in the initial stage so that they can be used in the system to actually meet what we desire.
- c. Design:** It is done to find out the functional requirements that the system needs to have. It consists of logical and physical design process.
- d. Verification:** This phase consists of testing. During this phase, a test environment is created where developed system is tested.
- e. Implementation:** In this phase, the system is deployed in production environment (User's end). And smoke test of the product is also conducted to see if the product is working like it should.
- f. Maintenance:** In this phase, any bugs or errors encountered during the working time of the system is fixed. This fixing process is also called patching.

- **Time schedule**

Following diagram is time schedule for the project.

S.N	Tasks	Date						
		MAY		JUN			JUL	AUG
		2	17	1	12	26	24	8
1	Synopsis							
2	Problem Finding							
3	Requirements							
4	Designing							
5	Coding							
6	Testing							
7	Reporting							

Fig: Gantt chart showing Project schedule

2.4 Feasibility Analysis

- **Economical Feasibility:** This system is highly feasible for small to medium sized organization that have small amount of users and inventory. This system is built using JAVA which is an open source platform. Thus, development cost is significantly low. It uses MySQL as its primary (only) database which is also an open source software that is available free of cost.
- **Operational Feasibility:** The system has graphical user interface .So, users will have no problem operating this software to its full capacity.
- **Technical Feasibility:** This software is based on inventory management. So, any individual who as the concept of inventory, accounting and of course, computer can use this software with ease.

2.5 Tools used for development

Following is the list of tools which is used to develop this project.

- **Net Beans IDE 7.2.1:** Net Beans IDE is an open source integrated development environment. Net Beans IDE supports development of all Java application types (Java SE (including Java FX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactoring and version control (supporting CVS, Subversion, Mercurial and Clear case).
- **Maven Tool:** Maven is a Yiddish word that means **accumulator of knowledge**. It is a standard way to build the projects. It gives a clear definition of what the project consisted of. It is an easy way to publish project information and a way to share JARs across several projects. It is a tool that can be used for building and managing any Java-based project. It aims to make the day-to-day work of Java

developers easier and generally help with the comprehension of any Java-based project. It's main objectives are:

- Making the build process easy
 - Providing a uniform build system
 - Providing quality project information
 - Providing guidelines for best practices development
 - Allowing transparent migration to new features
-
- **Oracle JDK 1.7:** The **Java Development Kit (JDK)** is an Oracle Corporation product aimed at Java developers. Since the introduction of Java, it has been by far the most widely used Java Software Development Kit (SDK). On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software.
 - **Fire Bug (for Firefox web Browser):** It is used to design the user interface for the system and to debug the bug regarding the design of the system .So that the system can be open in any browser.
 - **MySQL:** It is the world's second most widely used open-source relational database management system (RDBMS). MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (like wamp, xampp).

Chapter 3: Software Requirement Analysis

3.1 Introduction

Inventory Management System is an automated system for controlling and managing the inventory on a particular domain/area. This product is going to discourage a business house (supermarket and stores) to manage inventory by traditional method of ledger keeping or storing data on excel sheets. This system will do its best on encouraging inventory manager on using this system. This system is web-based that means it will be hosted on local area network of the organization. Thus, a person who is in-charge of the inventory is not bound to a single computer to use this system. This system tends to decrease the time taken to plan and order inventory items to be stocked. It helps in generating daily reports of the sales and order. Such, information is useful to find profit and loss that the organization has to bear so far. Another thing is that, it will ensure the data integrity. That means, it will have feature of getting and storing only valid data that meet the certain criteria. Other than that, this system consists of access control which provides security. Thus, user can only access those things that the administrator provides him/her access to. This system contains a database that will store data that is relevant to the inventory management system. The data may include user login information, user details (like name, address, contact, email), items information like stocked items information and sold items information. This system helps in minimizing the chance of data redundancy that may occur when two different users try to insert same information on the database. It also has printing feature. That means when an item is sold, the system will help to create an invoice that clearly states the price of the item, its quantity and the total amount. Not only that, it also helps in keeping the information of the sold items in the database.

3.2 General Description

The main objective of this system is to speed up the sales activities, maintain stock, generate invoice and provide security. It will also maintain history of transaction, minimize the error rate of staff and preserve data integrity.

3.3 Specific Requirements

- **Reduce cost and improve performance:** This system will automate entire working procedure. Thus, business activity will be faster and the number of employee will be reduced.
- **Generate Invoice:** The main objective of this system is to generate invoice for its customer and printing other related information.
- **Maintain stock:** The next important objective of this system is to maintain its stock so that items can be ordered at right time. This system will also keep track the under-stocked items.
- **Detailed records:** This system will also keep detailed record of product, supplier, and customer in the database so that data can be accessed easily when needed.
- **Data Integrity:** Data integrity makes sure that any data entered into system is correct. This system will have both field level and form level validation so that data will be stored correctly and properly.

3.4 Non-Functional Requirements

- **Security:** The system will have user login so that only authorized person will have access to the system.
- **Performance:** The system will have good execution and response time. Users don't have to wait too long for a transaction to complete.
- **Easy to use interface:** The system will be rich in graphical user interface.

Chapter 4: Design

4.1 Introduction

It is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. It consists of two parts. They are defined below:

- **Logical design** is done to list out the components of the software and their relationship to each other as they appear to the user. It explains inputs and outputs, processing function to be performed, business procedures, data models and controls.
- **Physical design** is done by translating the logical design into the specific technical design for the new system. It produces the actual specification for hardware, software, physical database, input/output media, manual procedures and specific controls.

4.2 System Design

Following figures are the logical designs for this system.

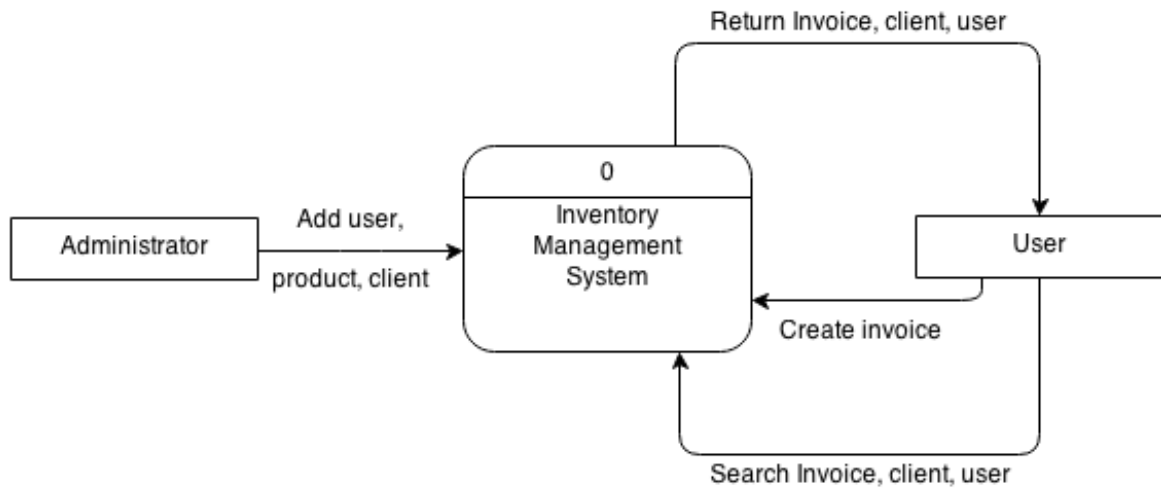


Fig: Context Level Data Flow Diagram

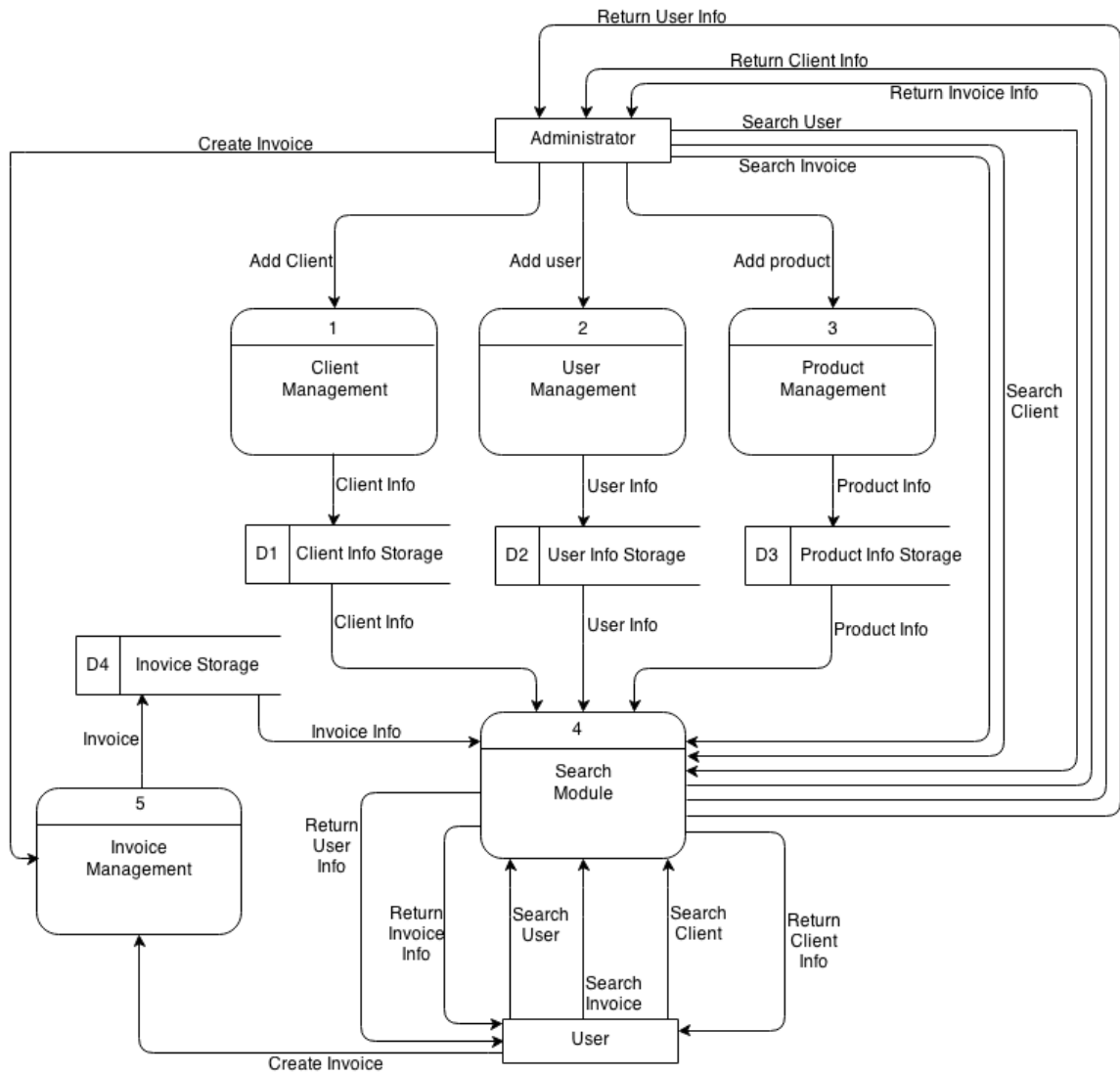


Fig: Level 1 Data Flow Diagram

According to above figures, in this system, there will be two users. They are: *Administrator* and *User*. *Administrator* will have access of adding new users, new products and new clients. Not only that, administrator will also have access to create invoice (sell/buy products) and search for information related to client, user and product. As for *users*, they are responsible for creating invoice (sell/buy products), search for invoice (reviewing sales history), client (to see the available client) and users (for knowing a user's general information). The system itself will return the details requested by the user and administrator.

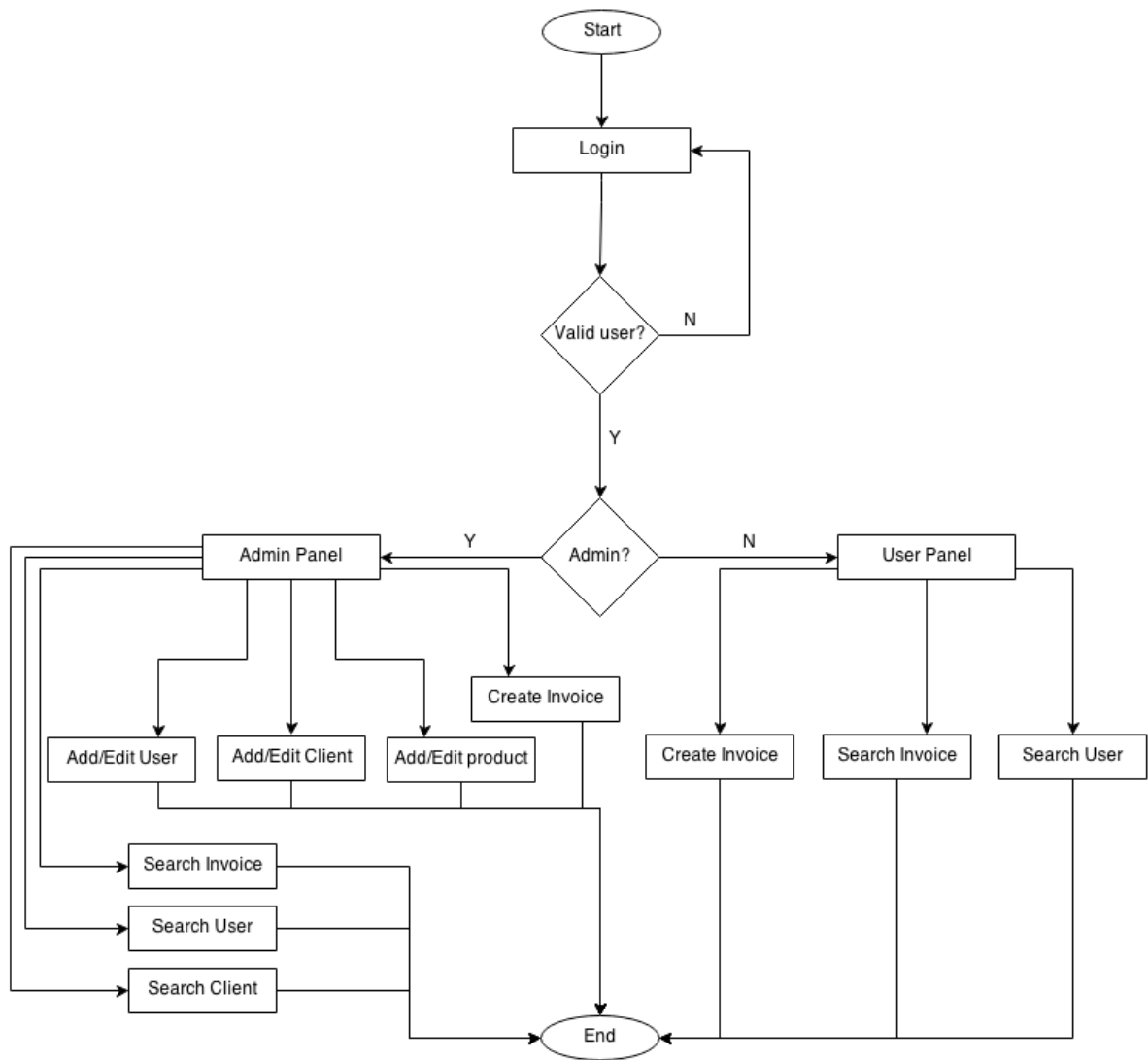


Fig: Flow chart diagram

Chapter 5: Testing

5.1 Introduction

Testing is one of the crucial stages in Software Development Life Cycle. Testing is a mechanism of checking the functioning and usability of a system. Test plans are made as soon as the coding work was started.

5.2 Levels of testing

Following levels of testing were done in this project.

- **Function-level testing**

Under this testing level, each of the functions used in the development of the system were tested as and when implemented. The mistakes were corrected and modifications were made as required.

- **Module-level testing**

With the collection and integration of variety of functions, modules were developed which again had to be tested in order to recognize the mistakes underlying when being integrated. This was also done during the coding phase after which the working of each module was presented to the supervisors for their comments so that further modifications could be made or could be thought of. The supervisor's

advice was taken into consideration and further correction and modification was made.

- **System-level testing**

It is done when the system is almost ready. As mentioned before, the system was created by collecting and integrating required modules. After the system was developed, it was tested and the errors were identified and rectified as and when required. Further modifications were made as per the errors and the bugs that were discovered during the testing phase.

- **Regression testing**

It is done when any changes that are made to an existing system/software. This type of testing is done to determine whether the newly implemented fix or feature has created new problem/bug or not. This type of test has been performed after every fixes.

5.3 Structural testing

Structural testing, also known as glass box testing or white box testing is an approach where the tests are derived from the knowledge of the software's structure or internal implementation. In structural testing an internal perspective of the system, as well as programming skills, are used to design test cases. Such types of test cases were also developed to test this project.

5.4 Testing the project

While testing the project, several bugs were encountered. Most of them were fixed with ease. Some gave a lot of problem. Supervisors allocated for this project helped in fixing them. Following list of test was performed during development phase whereas above mentioned test were performed during and after development:

- **Unit Testing:** It is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for

use. This test was performed at every step during development phase. All of the developed modules and features passed this test.

- **Integration Testing:** It is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. After adding previously developed and tested modules, this testing was performed to see if that module is compatible or not. And final result came positive at every step.
- **Validation Testing:** Validation testing checks that the product which is designed satisfies or fits the intended use. That means, it is done to check whether the software meets the requirements or not. This test was also done and output was positive.

Chapter 6: Implementation

6.1 Implementation of the project

After development and testing, this is the final stage of this project. Since this project is **Inventory Management System**, the scope of its implementation is wide. For actual implementation, following tools and technology must be used.

6.2 Hardware Requirement

- **Processor:** Pentium IV or greater.
- **Operating System:** any OS (32 bit or 64 bit)
- **RAM:** at least 512 MB or greater
- **HDD:** at least 200 MB for project deployment only

6.3 Software Requirement

- **Database Server:** MySQL Ver. 5.1.36 or greater (for data storage)
- **Application Server:** Glassfish 3.1.2.2 or greater (for application deployment)
- **Browser:** any modern browser (user's choice)

6.4 System Installation:

Following steps must be followed strictly for this project to work properly.

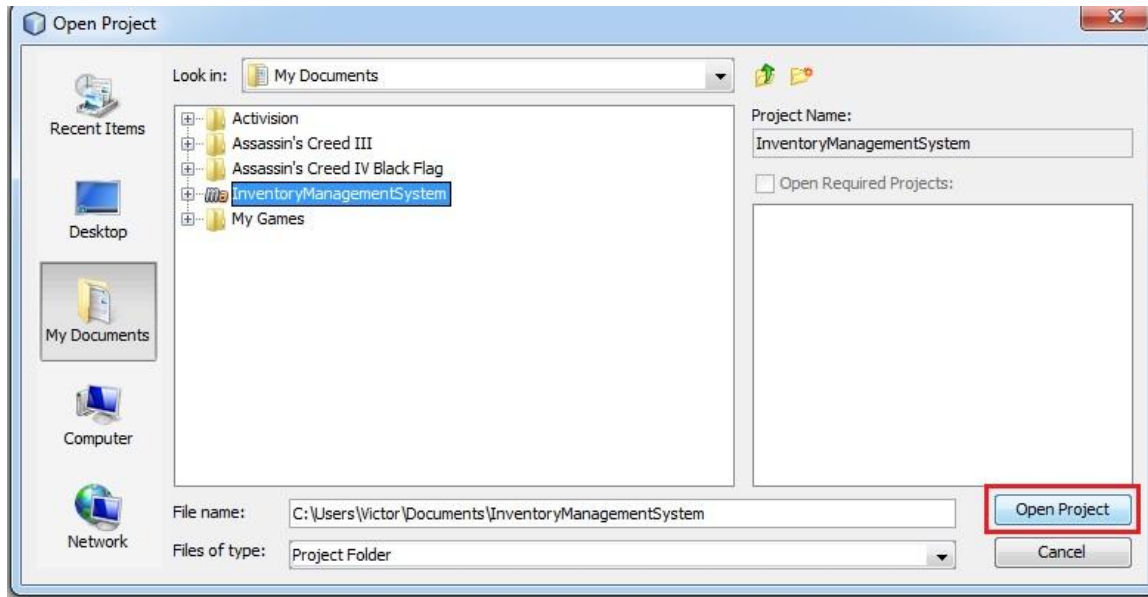
1. Download and install above mentioned version (or higher) of database server and application server on the host machine.
2. Install NetBeans IDE 7.2.1 that support java projects for building the project.
3. Also install glassfish server included inside this Netbeans IDE .
4. **Building with Netbeans 7.2.1.**

Open Netbeans IDE 7.2.1

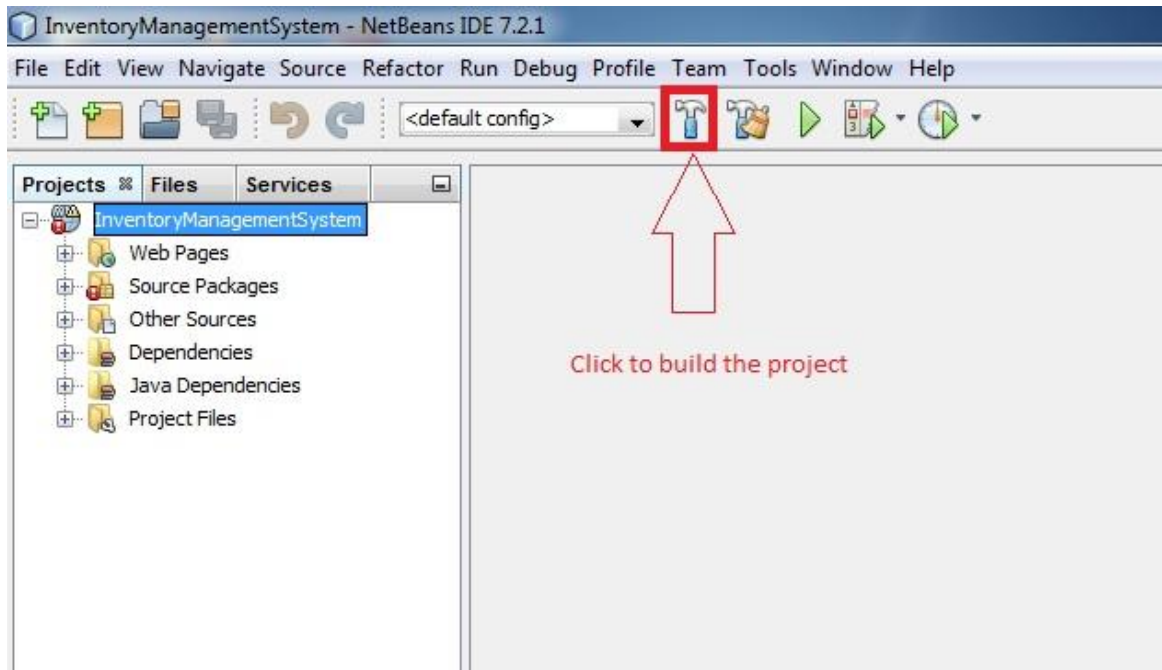


Then press **ctrl+shift+o** to open project.

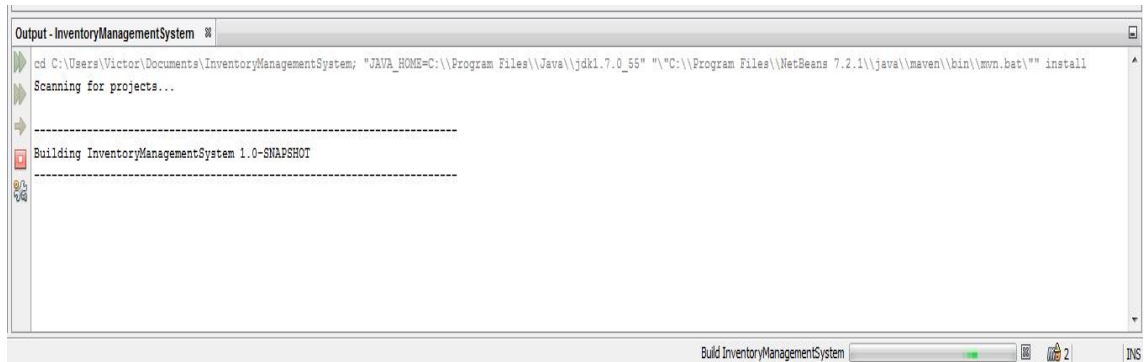
Browse to the location where the project files are kept and click open project.



After that select project--right click and click on build. Also clicking on icon next to build icon will **clean and rebuild** the project.



Then project will start to build by downloading dependency for the first time.





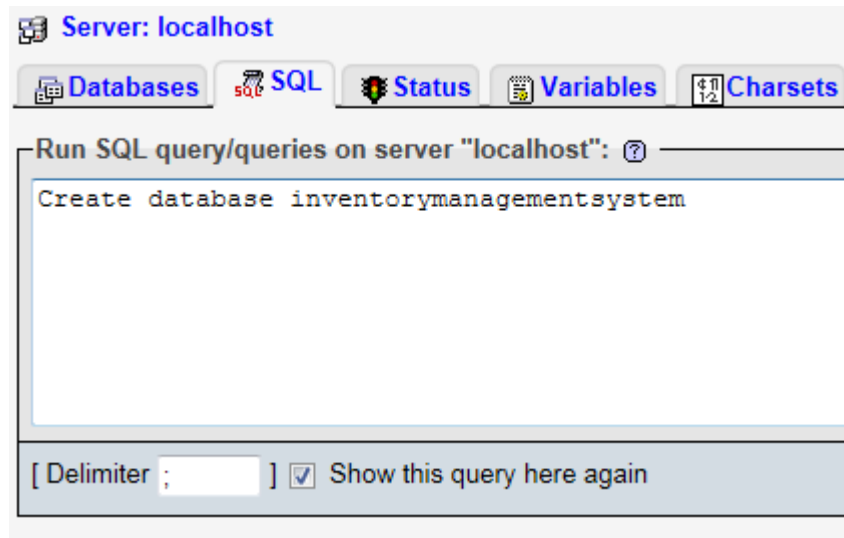
Following picture is build success notification.

```
-----  
BUILD SUCCESS  
-----
```

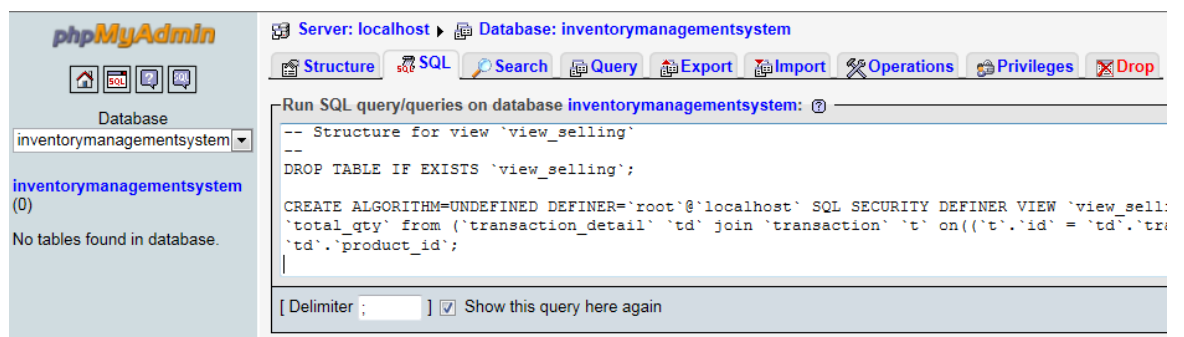
```
Total time: 23:39.940s  
Finished at: Sun Nov 02 17:45:14 NPT 2014  
Final Memory: 25M/90M  
-----
```

After the build process is complete the .war file will be created inside the project directory under **target** folder.

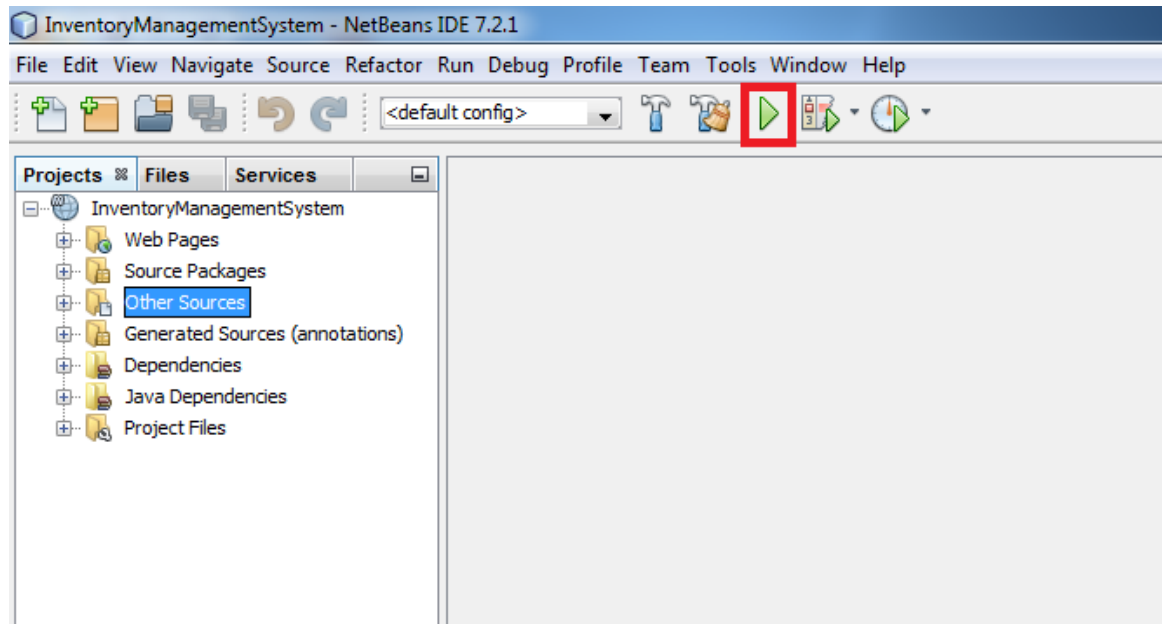
5. After that create a database called **inventorymanagementsystem**, in MySQL server.



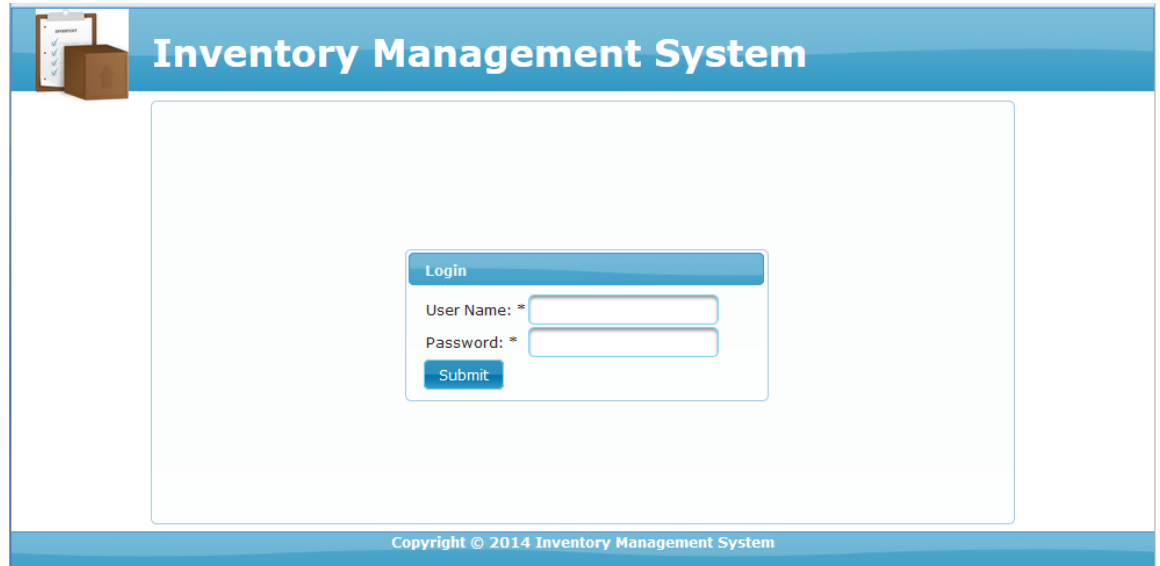
- Then copy paste that queries from **database.txt** file into the newly created database .



- After that in **Netbeans IDE** click on play button to deploy the **war** file.



8. After that login page will open in system's default browser.



9. Default credentials are: **username:** admin & **password:** admin for administrator and **username:** user & **password:** user for ordinary users.

For complete guide of the system, please refer to **Chapter 8: User manual**

Chapter 7: Project Legacy

7.1 Current Status of the Project

As we already know, inventory management system is very much important almost all type of business. Due to that there are a lot of such systems that supports various platforms. This is

creating a kind of competition among such software vendors. Top inventory management software has various features like bar code reader support, E-mail Alerts, Tools to manage customer inquiries/supplier quotes and so on. When we compare this project to with such software then this project does not even cover a fraction of it.

7.2 Remaining Areas of Concern

This project has lot of things missing when it is compared to commercial level software. Features like bar code reader support, alerting features, accounting support etc are still needed before making this project available to users. For doing that, knowledge of the technology is necessary. Not only that, fund for development is also necessary and time needed for development is also an important factor.

7.3 Technical and Managerial Lessons Learnt

During this project development, I have learned a lot of technical lessons. They include: JAVA programming language concept, JSF and Prime Faces framework knowledge and also whole lot of things about software development. While this doing project, I learned the importance of time. Not only that, this project also taught me that nothing is impossible to learn when we put our mind to it. I also learned how hard it is to develop software alone and also I learned about the problems that are faced by the developer during software development. I learned few things management as well. They include time management for studying and development of the project.

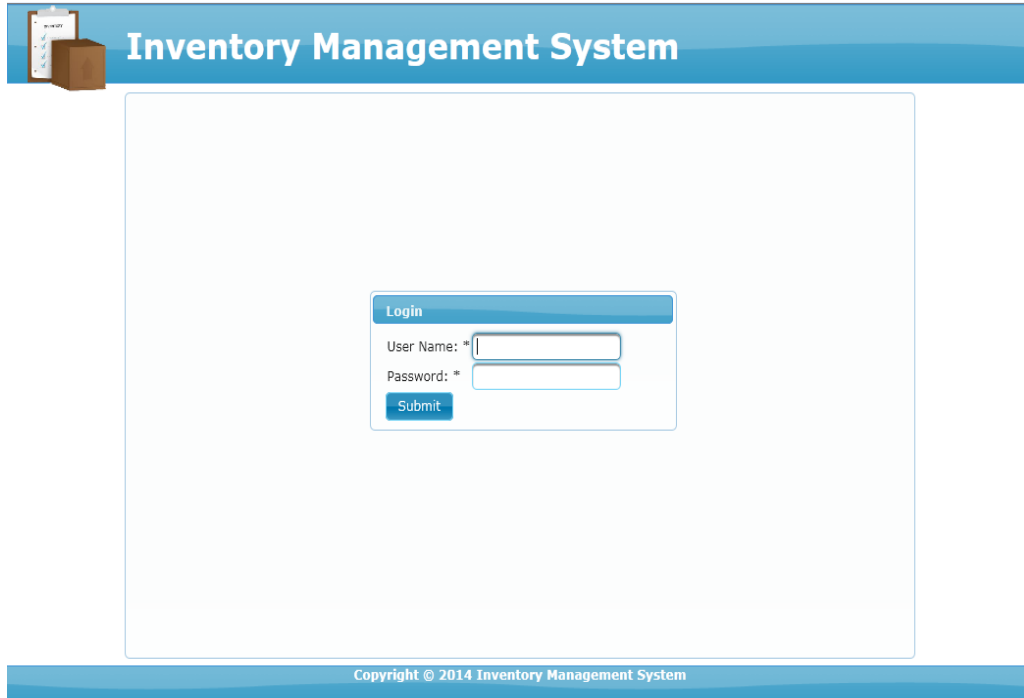
Chapter 8: User manual

For using the software, please go through it step by step.

- Open your browser and type the following URL.

<http://localhost:8080/InventoryManagementSystem/>

- Then login screen appears.



Inventory Management System

Login

User Name: *

Password: *

Copyright © 2014 Inventory Management System

Fig: Login Screen

- After that for accessing admin panel, type **admin** in username and again **admin** in password.
- For accessing user panel, type **user** in username and again **user** in password

- Following screenshot shows the admin panel and user panel.

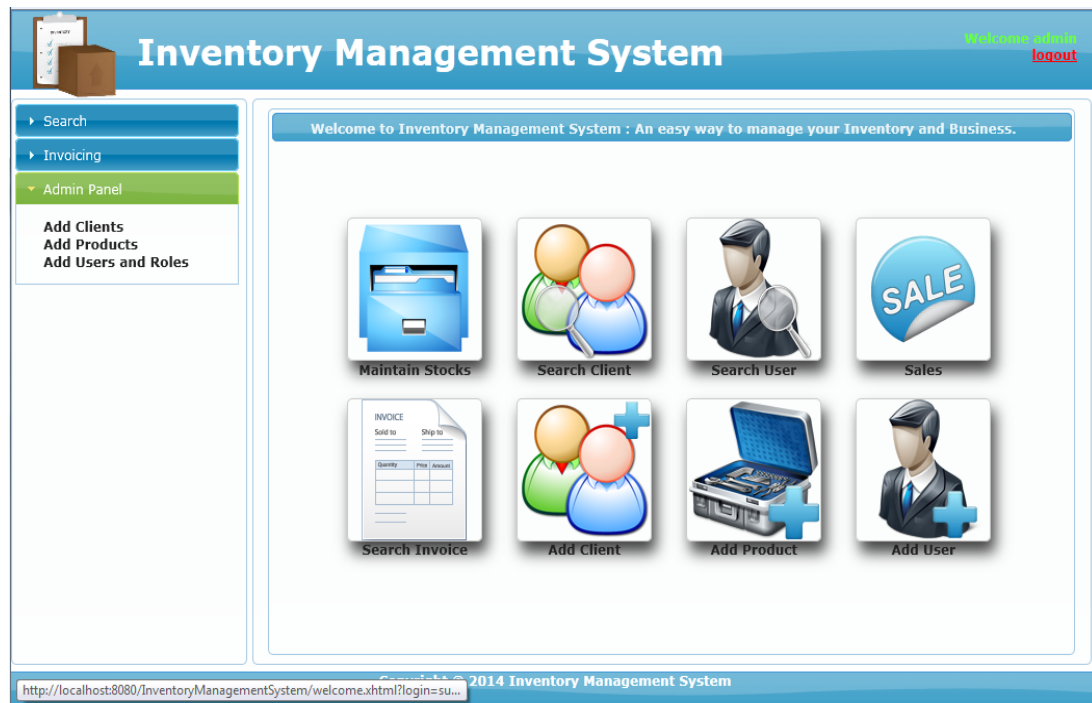


Fig: Admin panel

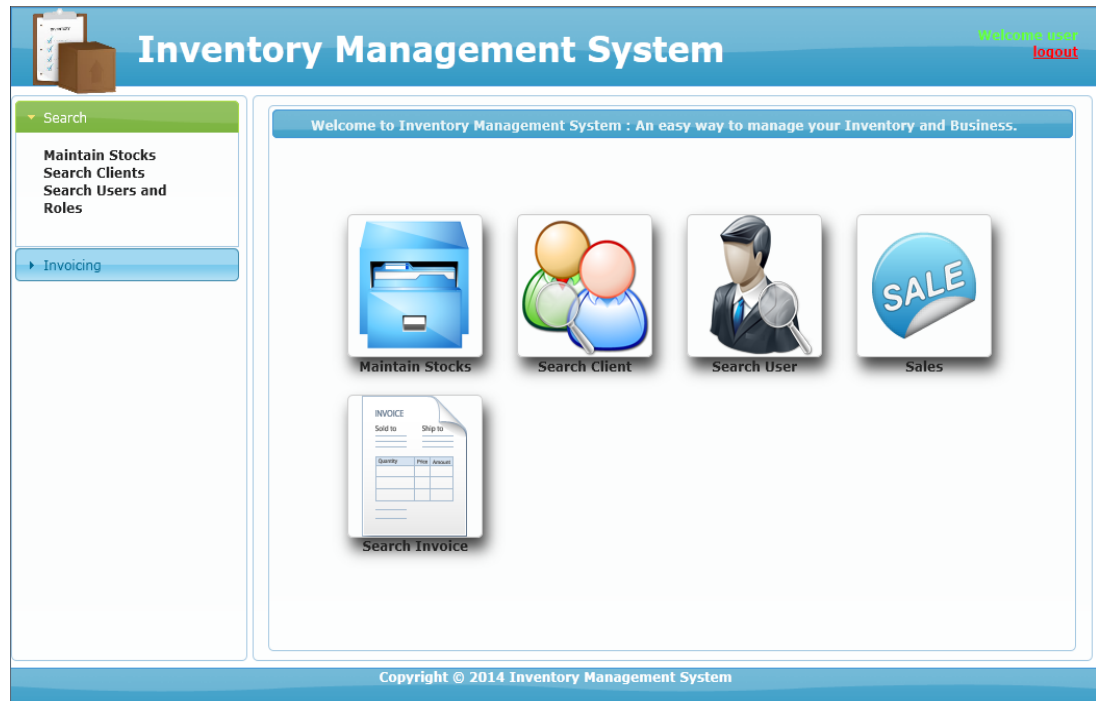



Fig: User Panel

- For adding product user must be logged in as admin.
- After that click on add client first. It is so because without adding client, we cannot order or sell product.



- Then in following page give all the details about the client like name, address and so on click on submit.



Inventory Management System

Welcome admin
[logout](#)

Search

Invoicing

Admin Panel

Add Clients

Add Products

Add Users and Roles

Add Clients

Name: *


Address: *

Email:

Phone: *

Mobile:

Copyright © 2014 Inventory Management System



Inventory Management System

Welcome admin
[logout](#)

Search

Invoicing

Admin Panel

Add Clients

Add Products

Add Users and Roles

All Clients

(1 of 1)

1


10

Name	Address	Email	Phone	Mobile	Delete	Edit
ABC-Corp	Kathmandu	ABC@gmail.com	4856231	984952687	x	edit

(1 of 1)

1


10



Copyright © 2014 Inventory Management System

<http://localhost:8080/InventoryManagementSystem/listClients.xhtml#>

- After that add another client following same process.



Inventory Management System

Welcome admin
[logout](#)


- Search
- Invoicing
- Admin Panel
 - Add Clients
 - Add Products
 - Add Users and Roles

All Clients

(1 of 1)
1
10


Name	Address	Email	Phone	Mobile	Delete	Edit
ABC-Corp	Kathmandu	ABC@gmail.com	4856231	984952687	x	
XYZ-Corp	Pokhara	XYZ@gmail.com	49815678	987812345	x	

(1 of 1)
1
10



Copyright © 2014 Inventory Management System

- For returning to dashboard click on the icon at top.



Inventory Management System

Welcome admin
[logout](#)

Search

Invoicing

Admin Panel

Add Clients

Add Products

Add Users and Roles

All Clients

(1 of 1) 1 10

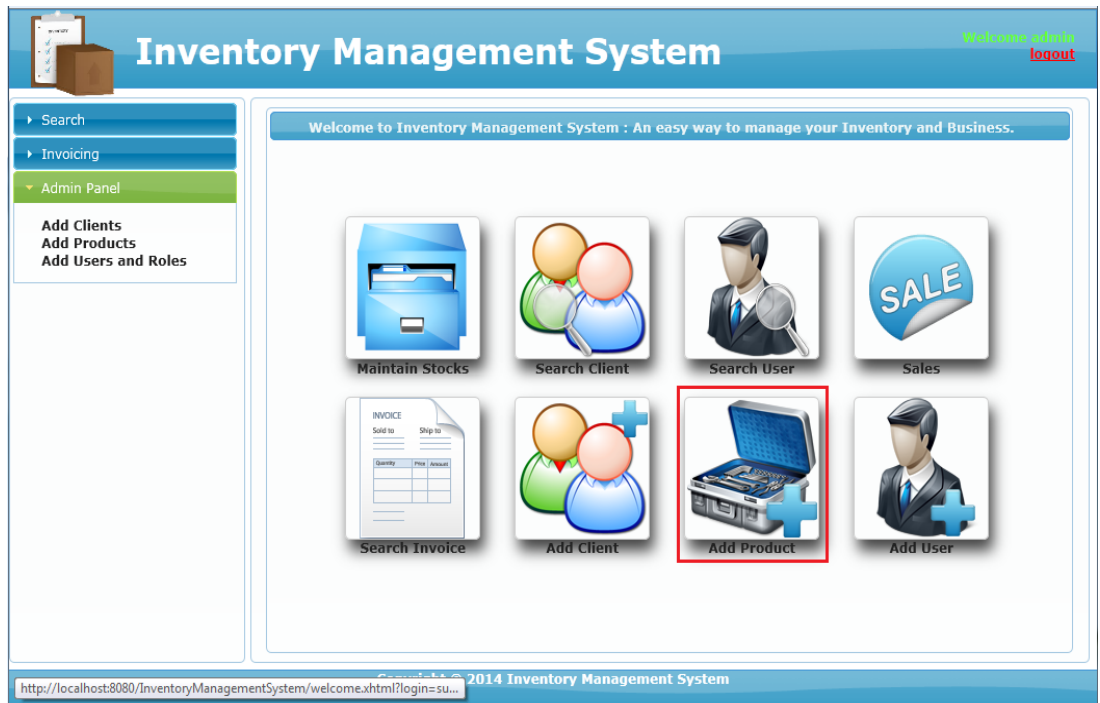
Name	Address	Email	Phone	Mobile	Delete	Edit
ABC-Corp	Kathmandu	ABC@gmail.com	4856231	984952687		
XYZ-Corp	Pokhara	XYZ@gmail.com	49815678	987812345		

(1 of 1) 1 10


Copyright © 2014 Inventory Management System

http://localhost:8080/InventoryManagementSystem/listClients.xhtml#

- Now click on add product icon to add product.



- On following page, add a category and press **add**.



Inventory Management System

Welcome admin
[logout](#)

Search

Invoicing

Admin Panel

Add Clients

Add Products

Add Users and Roles

Add Category

Category Name: *

Add Prdouct

Product Name: *


Description: *

Category: *

Purchase Price/unit: *

Selling Price/unit: *

Copyright © 2014 Inventory Management System



Inventory Management System

Search

Invoicing

Admin Panel

Add Clients

Add Products

Add Users and Roles

Category was successfully created.

Add Category

Category Name: *

AddClear

Add Prdouct

Product Name: *

Description: *

Category: *

Select one

Purchase Price/unit: *


Selling Price/unit: *

AddClear

Copyright © 2014 Inventory Management System

<http://localhost:8080/InventoryManagementSystem/addProduct.html>

- Now give all the details about that product in **add product** box and press add.



Inventory Management System

Welcome, admin
[logout](#)

Search

Invoicing

Admin Panel

Add Clients

Add Products

Add Users and Roles

Add Category

Category Name: *

AddClear

Add Prdouct

Product Name: *

Description: *

Category: *


Purchase Price/unit: *

Selling Price/unit: *

AddClear

Copyright © 2014 Inventory Management System

- After that user will be redirected to **maintainStock.xhtml** page.



Inventory Management System

Welcome admin
[logout](#)

Search

Invoicing

Admin Panel

Add Clients
Add Products
Add Users and Roles

Available Products

Available Categories


(1 of 1)

1

Product Name ↕	Category ↕	Product Description ↕	Purchase Price ↕	Selling Price ↕	Total Stock ↕	Delete	Edit
Heater	Electronics	This is a heater	Rs.2000.00	Rs.3000.00	0	✕	✎


(1 of 1)

1



Copyright © 2014 Inventory Management System

- Again do same process to add another product. Also, note that a product can be added without adding new category.



Inventory Management System

Welcome admin [logout](#)

Search

Invoicing

Admin Panel

Add Clients
Add Products
Add Users and Roles

Available Products

Available Categories

(1 of 1)

1


5

Product Name ↕	Category ↕	Product Description ↕	Purchase Price ↕	Selling Price ↕	Total Stock ↕	Delete	Edit
Heater	Electronics	This is a heater	Rs.2000.00	Rs.3000.00	0	✕	✎
Rice Cooker	Electronics	This is an electronic cooker	Rs.1000.00	Rs.1500.00	0	✕	✎

(1 of 1)

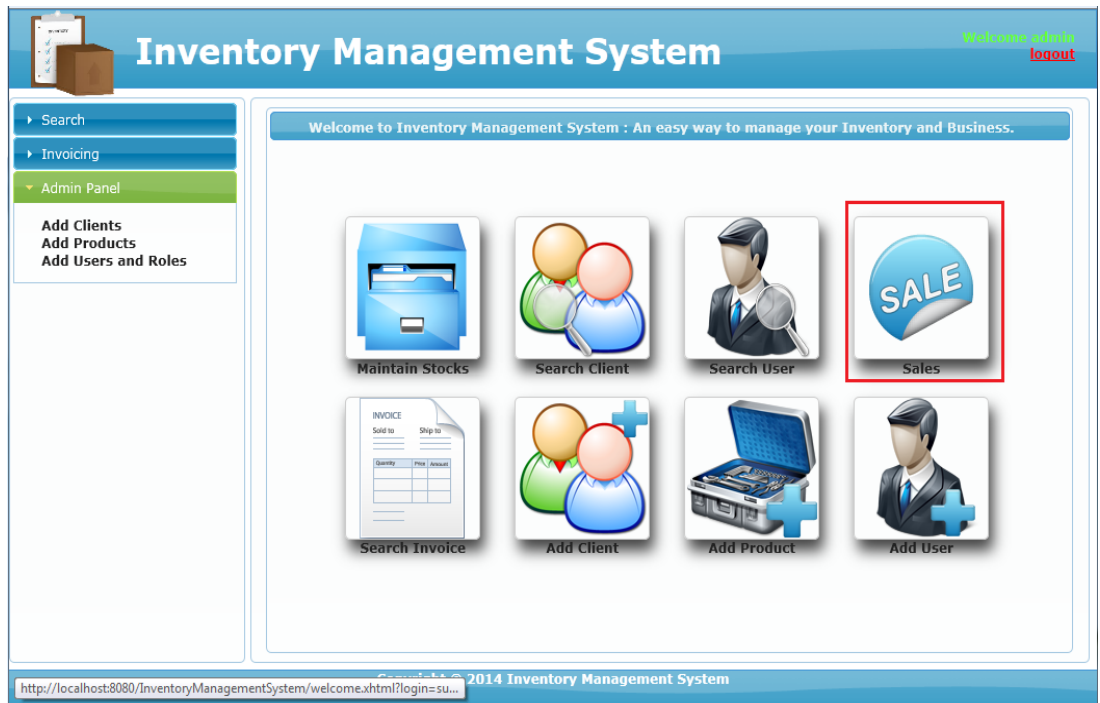
1

5




Copyright © 2014 Inventory Management System

- After adding products, now there quantities must be more than zero. So, go ahead and let get back to dash board and click on **Sales** icon.



- Actually, inside **sales** we can not only sell product but also buy it.
- So, let's go ahead and create an invoice for buying quantities for the product we added previously.



Inventory Management System

Welcome admin
[logout](#)

- Search
- Invoicing
- Admin Panel
 - Add Clients
 - Add Products
 - Add Users and Roles

Create Invoice


Date Created:
Transaction With: * Select one ▾
Transaction Status: * Select one ▾
Create

Transaction Details With				
Transaction Status:		Transaction Date:		
Product	Price/piece	Details	Quantity	Total Price
No records found.				
Total Amount:				Rs..00

Submit
Clear

Copyright © 2014 Inventory Management System

- Fill the first box and click on **create**. After that fill second box and click on **submit** button next to it.



Inventory Management System

Welcome admin
[logout](#)


- Search
- Invoicing
 - Create Invoice
 - Search Invoice
- Admin Panel

Create Invoice

Date Created: *
Transaction With: *
Transaction Status: *


Products: *
Quantity: *
Price: 2000
Amount: *

Transaction Details With				
Transaction Status:			Transaction Date:	
Product	Price/piece	Details	Quantity	Total Price
No records found.				
Total Amount:				Rs..00



Copyright © 2014 Inventory Management System

- Then after that finally click on submit button at the bottom to save the record in database.



Inventory Management System

Temp TransactionDetailCreated

- Search
- Invoicing
- Admin Panel
 - Add Clients
 - Add Products
 - Add Users and Roles


Create Invoice

Date Created: * 2014-11-25
 Transaction With: * ABC-CORP
 Transaction Status: * Buying
 Create

Products: * Select one
 Quantity: *
 Price: 0
 Amount: *
 Submit

Transaction Details With ABC-Corp				
Transaction Status: Buying		Transaction Date: Tue Nov 25 00:00:00 NPT 2014		
Product	Price/piece	Details	Quantity	Total Price
Heater	Rs.2000.00	This is a heater	5	Rs.10000.00
Total Amount:				Rs.10000.00

Submit




Clear

<http://localhost:8080/InventoryManagementSystem/addInvoice.xhtml#>

Copyright © 2014 Inventory Management System

- After that we will be redirected to **listInvoice.xhtml**



Inventory Management System

Welcome **admin**
logout


- Search
- Invoicing
- Admin Panel
 - Add Clients
 - Add Products
 - Add Users and Roles

Search Invoice

(1 of 1)
1
10

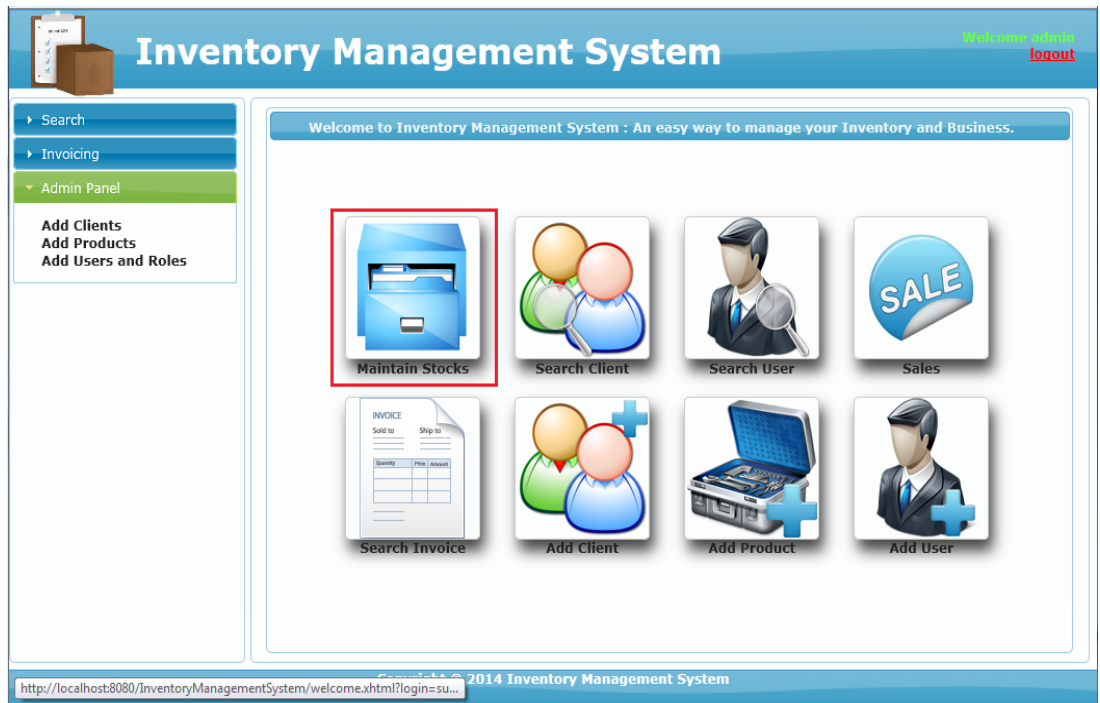
Transaction Date	Transaction Client	Transaction Status	Product	Quantity	Total Price
Tue Nov 25 00:00:00 NPT 2014	ABC-Corp	buying	Heater	5	10000

(1 of 1)
1
10




Copyright © 2014 Inventory Management System

- Follow same steps to add more quantities to the inventory for different products.
- After that let's go back to admin panel again.
- This time, let's click on **maintain stocks** icon.



- Now we have more items in our stock.



Inventory Management System

Welcome admin
[logout](#)

- Search
- Invoicing
- Admin Panel
 - Add Clients
 - Add Products
 - Add Users and Roles

Available Products Available Categories


(1 of 1)

1
5

Product Name	Category	Product Description	Purchase Price	Selling Price	Total Stock	Delete	Edit
Rice Cooker	Electronics	This is an electronic cooker	Rs.1000.00	Rs.1500.00	0	✕	✎
Heater	Electronics	This is a heater	Rs.2000.00	Rs.3000.00	5	✕	✎

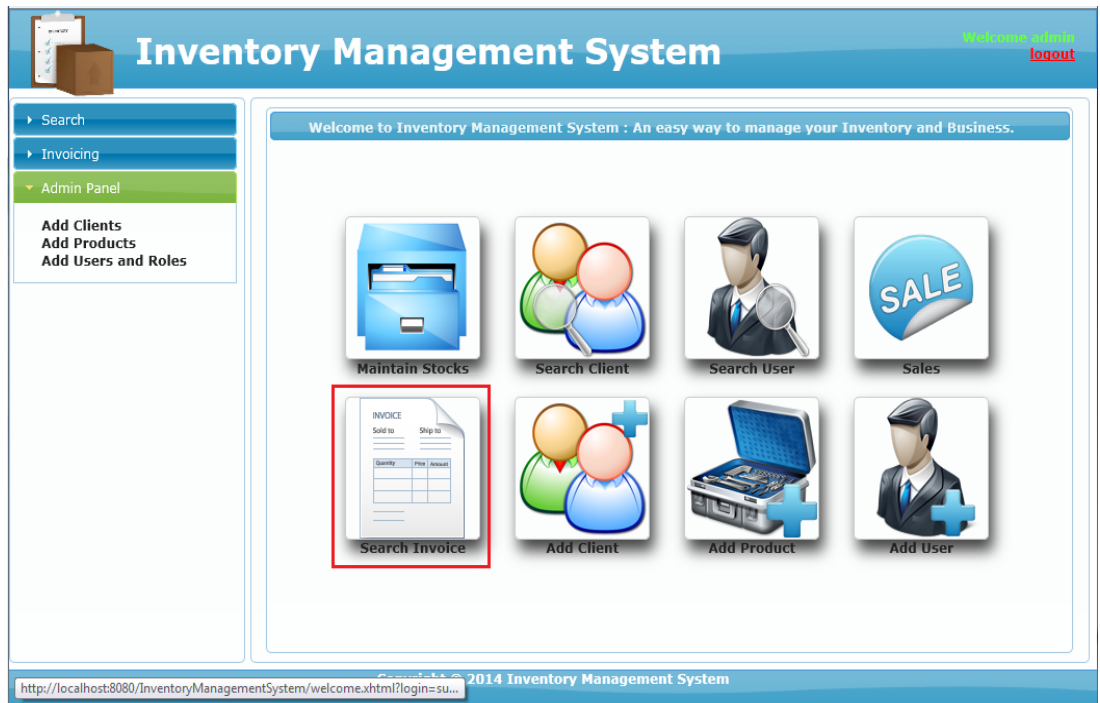
(1 of 1)

1
5




Copyright © 2014 Inventory Management System

- The one in red text indicates the product is getting low or is out of stock.
- Now, let's go ahead and get to **dashboard** and into **search invoice**.



- Here, the latest transactions are shown that is related to buying and selling.



Inventory Management System

Welcome admin
[logout](#)


- Search
- Invoicing
- Admin Panel
 - Add Clients
 - Add Products
 - Add Users and Roles

Search Invoice

(1 of 1) 1 10


Transaction Date	Transaction Client	Transaction Status	Product	Quantity	Total Price
Tue Nov 25 00:00:00 NPT 2014	ABC-Corp	buying	Heater	5	10000

(1 of 1) 1 10



Copyright © 2014 Inventory Management System

- Now let's get back to **sales**. For easy navigation we can also click on invoicing and **create invoice**.



Inventory Management System

Welcome admin
[logout](#)

Search

Invoicing

Create Invoice

Search Invoice


Admin Panel

Search Invoice

(1 of 1) 1 10


Transaction Date	Transaction Client	Transaction Status	Product	Quantity	Total Price
Tue Nov 25 00:00:00 NPT 2014	ABC-Corp	buying	Heater	5	10000

(1 of 1) 1 10



Copyright © 2014 Inventory Management System

- This time we are going to sell our item to another client.



Inventory Management System

Welcome admin [logout](#)

- Search
- Invoicing
 - Create Invoice
 - Search Invoice
- Admin Panel

Create Invoice

Date Created: *
Transaction With: *
Transaction Status: *


Transaction Details With				
Transaction Status:		Transaction Date:		
Product	Price/piece	Details	Quantity	Total Price
No records found.				
Total Amount:				Rs.10000.00

Copyright © 2014 Inventory Management System

- Rest of the process is same as buying. The only difference, we select **selling** in **transaction status**.
- Also please note that we cannot sell more than the available quantity in our stock. Not only that, we also cannot sell items that is less than or equal to 5.
- After that, let's get back to dashboard and click on search user.



- Following page looks like this:



Inventory Management System

Welcome admin
[logout](#)


- Search
- Invoicing
- Admin Panel
 - Add Clients
 - Add Products
 - Add Users and Roles

All Users

(1 of 1)

First Name	Last Name	Gender	Phone	Address	Emp ID	Email	User Role	Delete	Edit
Admin	admin_last	male	1234567890	kathmandu	IMS_1	admin@gmail	admin		
User	user_last	male	12345678	Pokhara	IMS_2	user@gmail.c	user		

(1 of 1)



Copyright © 2014 Inventory Management System

- As we can see, the available users of the system are only two. They are admin and user.
- That means, admin has the access to edit or delete user information.
- Again let's get back to dashboard and click on add user icon.



- Add user page looks like this:

The screenshot shows the 'Inventory Management System' Admin Panel. The left sidebar contains a 'Search' button and a list of menu items: 'Invoicing', 'Admin Panel' (highlighted), 'Add Clients', 'Add Products', and 'Add Users and Roles'. The main content area is titled 'Add/Edit Users' and contains a form with the following fields:

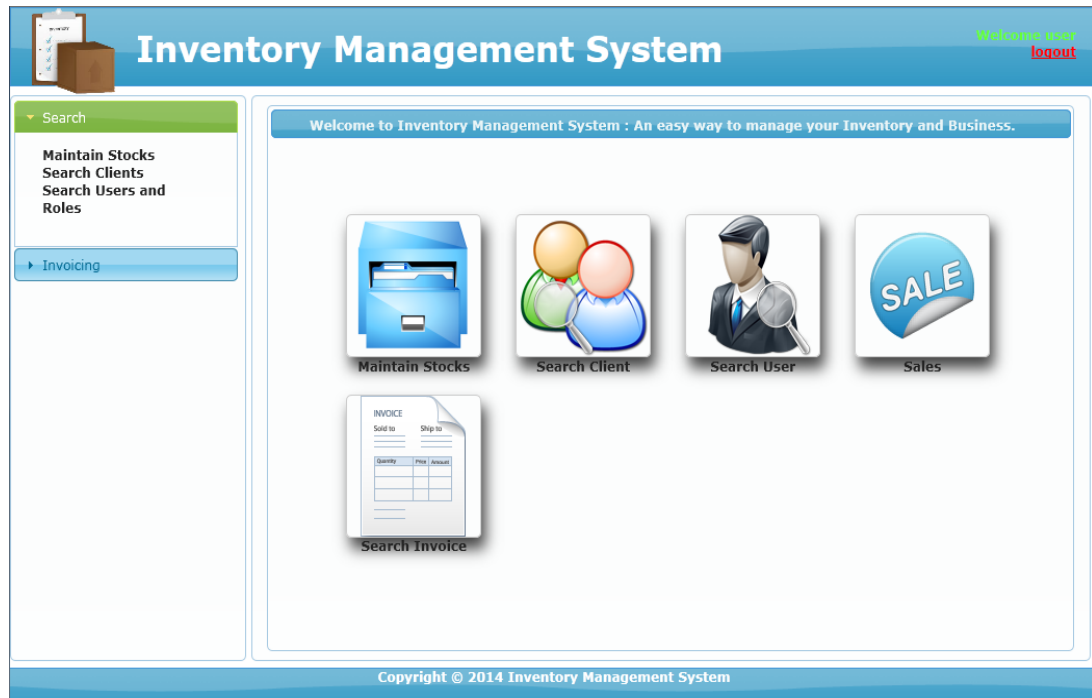
- Employee ID: * (text input with value 'IMS_3')
- Firstname: * (text input)
- Lastname: * (text input)
- DOB: (text input)
- Gender: * (radio buttons for Male, Female, Third; Male is selected)
- Email: * (text input)
- Phone: (text input)
- Address: (text input)
- Username: * (text input)
- Password: * (text input)
- Confirm Password: * (text input)
- User Role: * (dropdown menu with 'Admin' selected)

At the bottom of the form are 'Add' and 'Clear' buttons. The footer of the page reads 'Copyright © 2014 Inventory Management System'.


- Through here, only admin can add users and assign roles to the new user.
- Now, only thing that is left is the search client icon. So, let's get to dashboard and click on **search client** icon.



- The following page contains the list of the clients we have added previously.
- Now let's logout off the system and login as the user and take a look around.
- As we may have noticed, the dashboard looks like this.



- This is because an ordinary user does not have access to add anything. But he/she can order or sell any product he/she wants. That way, user can maintain stocks in absence of administrator.
- Here, inside search user, user can only see and edit his/her details only. The software was designed this way increase security.



Inventory Management System

Welcome user
[logout](#)

Search


Maintain Stocks
Search Clients
Search Users and Roles

Invoicing

All Users


(1 of 1)

1
10

First Name	Last Name	Gender	Phone	Address	Emp ID	Email	User Role	Edit
User	user_last	male	12345678	Pokhara	IMS_2	user@gmail.co	user	

(1 of 1)

1
10



http://localhost:8080/InventoryManagementSystem/listUser.xhtml#

Copyright © 2014 Inventory Management System

- Also the following icon means details present on that page can be generated as PDF document. This applies for both admin and user panel.



- Other than that everything, else is same as admin panel.

Chapter 9: Source Code

Actual source of this project is very big and spans among variuos files. Therefore, following codes are just a part of the actual source code. Also for complete source code please refer to the **CD** that is provided with this report.

welcome.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:ui="http://java.sun.com/jsf/facelets"

    xmlns:h="http://java.sun.com/jsf/html"

    xmlns:p="http://primefaces.org/ui"

    xmlns:f="http://java.sun.com/jsf/core">

    <body>

        <ui:composition template="./WEB-INF/template/mainTemplate.xhtml">
```

```
<ui:define name="content">
```

```
<p:panel header="Welcome to Inventory Management System : An easy way to  
manage your Inventory and Business." style="margin:0 auto; text-align: center; height:  
100%; position: relative">
```

```
<div class="dashboard">
```

```
<h:form>
```

```
< h : c o m m a n d L i n k  
action="#{viewBuyingController.setNullAndReturnPage(viewSellingController)}"  
styleClass="dash_link">
```

```

```

```
<h:outputText value="Maintain Stocks" styleClass="center_text"/>
```

```
</h:commandLink>
```

```
<h:commandLink action="listClients.xhtml?faces-redirect=true"  
styleClass="dash_link">
```

```

```

```
<h:outputText value="Search Client" styleClass="center_text"/>
```

```
</h:commandLink>
```

```
<h:commandLink action="listUser.xhtml?faces-redirect=true"  
styleClass="dash_link">
```

```



<h:outputText value="Search User" styleClass="center_text"/>

</h:commandLink>

<h:commandLink action="addInvoice.xhtml?faces-redirect=true"
styleClass="dash_link">



<h:outputText value="Sales" styleClass="center_text"/>

</h:commandLink>

<h:commandLink action="listInvoice.xhtml?faces-redirect=true"
styleClass="dash_link">



<h:outputText value="Search Invoice" styleClass="center_text"/>

</h:commandLink>

<h:commandLink action="addClients.xhtml?faces-redirect=true"
rendered="{userController.adminPanelrender}" styleClass="dash_link">



<h:outputText value="Add Client" styleClass="center_text"/>

</h:commandLink>

```

```

        <h:commandLink          action="addProduct.xhtml?faces-redirect=true"
rendered="#{userController.adminPanelrender}" styleClass="dash_link">

        <h:outputText value="Add Product" styleClass="center_text"/>

    </h:commandLink>

    <h:commandLink          action="#{userController.prepareCreate()}"
rendered="#{userController.adminPanelrender}" styleClass="dash_link">

        <h:outputText value="Add User" styleClass="center_text"/>

    </h:commandLink>

</h:form>

<div style="clear:both"></div>

</div>

</p.panel>

</ui.define>

</ui.composition>

</body>

</html>

```

maintainStock.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:ui="http://java.sun.com/jsf/facelets"

    xmlns:h="http://java.sun.com/jsf/html"

    xmlns:p="http://primefaces.org/ui"

    xmlns:pe="http://primefaces.org/ui/extensions"

    xmlns:f="http://java.sun.com/jsf/core">

    <body>

        <ui:composition template="./WEB-INF/template/mainTemplate.xhtml">

            <ui:define name="content">

                <h:form>

                    <p:tabView activeIndex="#{productController.tabPaneIndex}">

                        <p:tab title="Available Products">

                            <p:growl id="growl"/>

                            <p:dataTable value="#{productController.allList}" var="product"
paginator="true" rows="5" editable="true">
```

```

paginatorTemplate="{CurrentPageReport} {FirstPageLink}
{PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

```

```

rowsPerPageTemplate="5,10" id="product1"
rendered="#{userController.adminPanelrender}"

```

```

sortBy="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(pro
duct.id)}" sortOrder="ASCENDING"

```

```

rowStyleClass="#{(viewBuyingController.getStock(product.id)-viewSellingController.getSt
ock(product.id)) >= 5 ? null:'old'}">

```

```

<p:column sortBy="#{product.name}" headerText="Product Name"
filterBy="#{product.name}">

```

```

<p:cellEditor>

```

```

<f:facet name="output">

```

```

<h:outputLabel value="#{product.name}" />

```

```

</f:facet>

```

```

<f:facet name="input">

```

```

<p:inputText required="true" value="#{product.name}"
size="15"/>

```

```

</f:facet>

```

```

</p:cellEditor>

```



```

        </p:column>

        <
            p
            :
            c
            o
            l
            u
            m
            n
            sortBy="#{categoryController.cateName(product.categoryId)}"    headerText="Category"
            filterBy="#{categoryController.cateName(product.categoryId)}">

            <p:cellEditor>

                <f:facet name="output">

                    <
                        h
                        :
                        o
                        u
                        t
                        p
                        u
                        t
                        L
                        a
                        b
                        e
                        l
                        value="#{categoryController.cateName(product.categoryId)}"/>

                </f:facet>

                <f:facet name="input">

                    <p:selectOneMenu                value="#{product.categoryId}"
                    style="width: 84px;">

                        <f:selectItem itemLabel="Select One" itemValue=""/>

                        <f:selectItems                value="#{categoryController.allList}"
                        var="category" itemLabel="#{category.name}" itemValue="#{category.id}"/>

                    </p:selectOneMenu>

                </f:facet>

            </p:cellEditor>

        </p:column>

        <p:column    sortBy="#{product.description}"    headerText="Product
        Description">

            <p:cellEditor>

```

```

        <f:facet name="output">

            <h:outputLabel value="#{product.description}"/>

        </f:facet>

        <f:facet name="input">

            <p:inputText required="true" value="#{product.description}"
size="15"/>

        </f:facet>

    </p:cellEditor>

</p:column>

<p:column                sortBy="#{product.purchasePrice}"
headerText="Purchase Price" filterBy="#{product.purchasePrice}">

    <p:cellEditor>

        <f:facet name="output">

            <h:outputLabel        value="Rs.#{product.purchasePrice}.00"
style="float: right"/>

        </f:facet>

        <f:facet name="input">

            <p:inputText                required="true"
value="#{product.purchasePrice}" size="15"/>

        </f:facet>

    </p:cellEditor>

```

```

</p:column>

<p:column sortBy="#{product.sellingPrice}" headerText="Selling
Price" filterBy="#{product.sellingPrice}">

    <p:cellEditor>

        <ffacet name="output">

            <h:outputLabel value="Rs.#{product.sellingPrice}.00"
style="float: right"/>

        </ffacet>

        <ffacet name="input">

            <p:inputText required="true" value="#{product.sellingPrice}"
size="15"/>

        </ffacet>

    </p:cellEditor>

</p:column>

<
    p
    :
    c
    o
    l
    u
    m
    n
sortBy="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(pro
duct.id)}"
headerText="Total
Stock"
filterBy="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(pro
duct.id)}">

    <
        h
        :
        o
        u
        t
        p
        u
        t
        L
        a
        b
        e
        l
value="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(prod
uct.id)}" style="float: right"/>

```

```

</p:column>

<p:column headerText="Delete">

    <h:commandLink styleClass="ui-icon ui-icon-close" style="margin:
0 auto" action="#{productController.destroy(product)}" />

</p:column>

<p:column headerText="Edit">

    <h:commandLink styleClass="ui-icon ui-icon-pencil" style="margin:
0 auto" action="#{productController.prepareEdit(product)}" />

</p:column>

</p:dataTable>

<p:dataTable value="#{productController.allList}" var="product"
paginator="true" rows="5"

                paginatorTemplate="{CurrentPageReport} {FirstPageLink}
{PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

                rowsPerPageTemplate="5,10" id="product2"
rendered="#{userController.userPanelrender}"

sortBy="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(pro
duct.id)}" sortOrder="ASCENDING"

rowStyleClass="#{(viewBuyingController.getStock(product.id)-viewSellingController.getSt
ock(product.id)) >= 5 ? null:'old'}">

```

```
<p:column sortBy="#{product.name}" headerText="Product Name"
filterBy="#{product.name}">
```

```
<h:outputLabel value="#{product.name}" />
```

```
</p:column>
```

```
< p : c o l u m n
sortBy="#{categoryController.cateName(product.categoryId)}" headerText="Category"
filterBy="#{categoryController.cateName(product.categoryId)}">
```

```
< h : o u t p u t L a b e l
value="#{categoryController.cateName(product.categoryId)}" />
```

```
</p:column>
```

```
<p:column sortBy="#{product.description}" headerText="Product
Description">
```

```
<h:outputLabel value="#{product.description}" />
```

```
</p:column>
```

```
<p:column sortBy="#{product.purchasePrice}" headerText="Selling
Price" filterBy="#{product.purchasePrice}">
```

```
<h:outputLabel value="Rs.#{product.purchasePrice}.00"
style="float: right"/>
```

```
</p:column>
```

```
<p:column sortBy="#{product.sellingPrice}" headerText="Selling
Price" filterBy="#{product.sellingPrice}">
```

```
<h:outputLabel value="Rs.#{product.sellingPrice}.00" style="float: right"/>
```

```
</p:column>
```

```
< p : c o l u m n  
sortBy="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(pro  
duct.id)}" headerText="Total Stock"  
filterBy="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(pro  
duct.id)} ">
```

```
< h : o u t p u t L a b e l  
value="#{viewBuyingController.getStock(product.id)-viewSellingController.getStock(prod  
uct.id)}" style="float: right"/>
```

```
</p:column>
```

```
</p:dataTable>
```

```
<br/>
```

```
<p:separator/>
```

```
<br/>
```

```
<p:commandLink ajax="false" style="float: right ; display: inline"  
rendered="#{userController.adminPanelrender}">
```

```
<p:graphicImage value="/resources/images/export.jpeg" height="40"  
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:  
7px;border: 3px solid #006600"/>
```

```
<pe:exporter type="pdf" target="product1 "  
fileName="ProductDetails" />
```

</p:commandLink>

<p:commandLink ajax="false" style="float: right ; display: inline"
rendered="#{userController.userPanelrender}">

<p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>

<pe:exporter type="pdf" target="product2"
fileName="ProductDetails" />

</p:commandLink>

<div class="clear"></div>

</p.tab>

<p.tab title="Available Categories">

<p:growl id="growl1"/>

<p:dataTable value="#{categoryController.allList}" var="category"
paginator="true" rows="5"

paginatorTemplate="{CurrentPageReport} {FirstPageLink}
{PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

rowsPerPageTemplate="5,10" id="category1"
rendered="#{userController.userPanelrender}">

<p:column sortBy="#{category.name}" headerText="Category
Name" filterBy="#{category.name}">

```

        <h:outputLabel value="#{category.name}"/>

    </p:column>

</p:dataTable>

<p:dataTable value="#{categoryController.allList}" var="category"
paginator="true" rows="5" editable="true"

                paginatorTemplate="{CurrentPageReport} {FirstPageLink}
{PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

                rowsPerPageTemplate="5,10" id="category2"
rendered="#{userController.adminPanelrender}">

    <p:column sortBy="#{category.name}" headerText="Category
Name" filterBy="#{category.name}">

        <p:cellEditor>

            <ffacet name="output">

                <h:outputLabel value="#{category.name}"/>

            </ffacet>

            <ffacet name="input">

                <p:inputText required="true" value="#{category.name}"
size="15"/>

            </ffacet>

        </p:cellEditor>

    </p:column>

```



```

        <p:column headerText="Delete">

            <h:commandLink styleClass="ui-icon ui-icon-close" style="margin:
0 auto" action="#{categoryController.destroy(category)}" />

        </p:column>

        <p:column headerText="Edit">

            <h:commandLink styleClass="ui-icon ui-icon-pencil" style="margin:
0 auto" action="#{categoryController.prepareEdit(category)}" />

        </p:column>

    </p:dataTable>

    <br/>

    <p:separator/>

    <br/>

    <p:commandLink ajax="false" style="float: right ; display: inline"
rendered="#{userController.adminPanelrender}">

        <p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>

        <pe:exporter type="pdf" target="category1" fileName="AllCategory"
/>

    </p:commandLink>

    <p:commandLink ajax="false" style="float: right ; display: inline"
rendered="#{userController.userPanelrender}">

```

```

        <p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>

        <pe:exporter type="pdf" target="category2" fileName="AllCategory"
/>

        </p:commandLink>

        <div class="clear"></div>

    </p.tab>

    </p.tabView>

    </h:form>

    </ui:define>

</ui:composition>

</body>

</html>

```

listUser.xhtml

```

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:ui="http://java.sun.com/jsf/facelets"

```

```

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:f="http://java.sun.com/jsf/core"

xmlns:pe="http://primefaces.org/ui/extensions">

<body>

<ui:composition template="./WEB-INF/template/mainTemplate.xhtml">

    <ui:define name="content">

        <h:form>

            <p:growl id="growl"/>

            <p:panel header="All Users">

                <p:dataTable          value="#{userController.allList}"          var="users"
paginator="true" rows="10"

                    id="users1"

                    paginatorTemplate="{CurrentPageReport}          {FirstPageLink}
{PreviousPageLink}          {PageLinks}          {NextPageLink}          {LastPageLink}
{RowsPerPageDropdown}"

                    rowsPerPageTemplate="5,10"          editable="true"
rendered="#{userController.adminPanelrender}">

                    <p:column    sortBy="#{users.FName}"    headerText="First    Name"
filterBy="#{users.FName}">

```

```

        <p:cellEditor>

            <f:facet name="output">

                <h:outputLabel value="#{users.FName}"/>

            </f:facet>

            <f:facet name="input">

                <p:inputText      required="true"      value="#{users.FName}"
size="8"/>

            </f:facet>

        </p:cellEditor>

    </p:column>

    <p:column sortBy="#{users.LName}" headerText="Last Name"
filterBy="#{users.LName}">

        <p:cellEditor>

            <f:facet name="output">

                <h:outputLabel value="#{users.LName}"/>

            </f:facet>

            <f:facet name="input">

                <p:inputText      required="true"      value="#{users.LName}"
size="8"/>

            </f:facet>

        </p:cellEditor>

```

```

        </p:column>

        <p:column sortBy="#{users.gender}" headerText="Gender"
filterBy="#{users.gender}" >

            <p:cellEditor>

                <ffacet name="output">

                    <h:outputLabel value="#{users.gender}" />

                </ffacet>

                <ffacet name="input" >

                    <p:selectOneMenu value="#{users.gender}" style="width:
84px;">

                        <fselectItem itemLabel="Male" itemValue="male"/>

                        <fselectItem itemLabel="Female" itemValue="female"/>

                        <fselectItem itemLabel="Third" itemValue="third"/>

                    </p:selectOneMenu>

                </ffacet>

            </p:cellEditor>

        </p:column>

        <p:column sortBy="#{users.phone}" headerText="Phone">

            <p:cellEditor>

                <ffacet name="output">

```

```

        <h:outputLabel value="#{users.phone}"/>

    </ffacet>

    <ffacet name="input">

        <p:inputText required="true" value="#{users.phone}" size="8"/>

    </ffacet>

</p:cellEditor>

</p:column>

<p:column sortBy="#{users.address}" headerText="Address"
filterBy="#{users.address}">

    <p:cellEditor>

        <ffacet name="output">

            <h:outputLabel value="#{users.address}"/>

        </ffacet>

        <ffacet name="input">

            <p:inputText required="true" value="#{users.address}"
size="9"/>

        </ffacet>

    </p:cellEditor>

</p:column>

<p:column sortBy="#{users.empId}" headerText="Emp ID">

```

```

        <h:outputLabel value="#{users.empId}"/>

    </p:column>

    <p:column sortBy="#{users.email}" headerText="Email"
filterBy="#{users.email}" >

```

```

        <p:cellEditor>

            <f:facet name="output">

                <h:outputLabel value="#{users.email}"/>

            </f:facet>

            <f:facet name="input">

                <p:inputText required="true" value="#{users.email}" size="15"/>

            </f:facet>

        </p:cellEditor>

    </p:column>

```

```

    <p:column sortBy="#{users.role}" headerText="User Role">

        <p:cellEditor>

            <f:facet name="output">

                <h:outputLabel value="#{users.role}"/>

            </f:facet>

            <f:facet name="input">

```

```

        <p:selectOneMenu value="#{users.role}" style="width: 84px;">

            <fselectItem itemLabel="Admin" itemValue="admin"/>

            <fselectItem itemLabel="User" itemValue="user"/>

        </p:selectOneMenu>

    </ffacet>

</p:cellEditor>

</p:column>

<p:column headerText="Delete">

    <h:commandLink styleClass="ui-icon ui-icon-close" style="margin: 0
auto" action="#{userController.destroy(users)}" />

</p:column>

<p:column headerText="Edit">

    <h:commandLink styleClass="ui-icon ui-icon-pencil" style="margin: 0
auto" action="#{userController.prepareEdit(users)}" />

</p:column>

</p:dataTable>

<p:dataTable value="#{userController.selected}" var="users"
paginator="true" rows="10" id="users2"

                paginatorTemplate="{CurrentPageReport} {FirstPageLink}
{PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

```



```

        r o w s P e r P a g e T e m p l a t e = " 5 , 1 0 "
rendered="#{userController.userPanelrender}">

        <p:column sortBy="#{users.FName}" headerText="First Name"
filterBy="#{users.FName}">

            <h:outputLabel value="#{users.FName}"/>

        </p:column>

        <p:column sortBy="#{users.LName}" headerText="Last Name"
filterBy="#{users.LName}">

            <h:outputLabel value="#{users.LName}"/>

        </p:column>

        <p:column sortBy="#{users.gender}" headerText="Gender"
filterBy="#{users.gender}">

            <h:outputLabel value="#{users.gender}"/>

        </p:column>

        <p:column sortBy="#{users.phone}" headerText="Phone">

            <h:outputLabel value="#{users.phone}"/>

        </p:column>

        <p:column sortBy="#{users.address}" headerText="Address"
filterBy="#{users.address}">

            <h:outputLabel value="#{users.address}"/>

        </p:column>

```

```

        <p:column sortBy="#{users.empId}" headerText="Emp ID">

            <h:outputLabel value="#{users.empId}" />

        </p:column>

        <p:column          sortBy="#{users.email}"          headerText="Email"
filterBy="#{users.email}">

            <h:outputLabel value="#{users.email}" />

        </p:column>

        <p:column sortBy="#{users.role}" headerText="User Role">

            <h:outputLabel value="#{users.role}" />

        </p:column>

        <p:column headerText="Edit">

            <h:commandLink styleClass="ui-icon ui-icon-pencil" style="margin: 0
auto" action="#{userController.prepareEdit(users)}" />

        </p:column>

    </p:dataTable>

    <br/>

    <p:separator/>

    <br/>

    <p:commandLink ajax="false" style="float: right ; display: inline"
rendered="#{userController.adminPanelrender}">

```

```
<p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>
```

```
<pe:exporter type="pdf" target="users1" fileName="AllUsers" />
```

```
</p:commandLink>
```

```
<p:commandLink ajax="false" style="float: right ; display: inline"
rendered="#{userController.userPanelrender}">
```

```
<p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>
```

```
<pe:exporter type="pdf" target="users2" fileName="AllUsers" />
```

```
</p:commandLink>
```

```
<div class="clear"></div>
```

```
</p:panel>
```

```
</h:form>
```

```
</ui:define>
```

```
</ui:composition>
```

```
</body>
```

```
</html>
```

listInvoice.xhtml

```

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:ui="http://java.sun.com/jsf/facelets"

    xmlns:h="http://java.sun.com/jsf/html"

    xmlns:p="http://primefaces.org/ui"

    xmlns:pe="http://primefaces.org/ui/extensions">

    <body>

        <ui:composition template="./WEB-INF/template/mainTemplate.xhtml">

            <ui:define name="content">

                <h:form>

                    <p:panel header="Search Invoice">

                        <p:dataTable
                                value="#{transactionDetailController.allList}"
                                var="transactionDetail" paginator="true" rows="10"

                                paginatorTemplate="{CurrentPageReport} {FirstPageLink}
{PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

                                rowsPerPageTemplate="5,10" id="thirddt"

                                sortBy="#{transactionDetail.transactionId.dateCreated}"
                                sortOrder="DESCENDING">

```

```

        <p:column headerText="Transaction Date"
filterBy="# {transactionDetail.transactionId.dateCreated.date}"
sortBy="# {transactionDetail.transactionId.dateCreated}">

```

```

        < h : o u t p u t L a b e l
value="# {transactionDetail.transactionId.dateCreated}" for="" />

```

```

    </p:column>

```

```

        <p:column headerText="Transaction Client"
filterBy="# {transactionDetail.transactionId.transactionWith}"
sortBy="# {transactionDetail.transactionId.transactionWith}">

```

```

        < h : o u t p u t L a b e l
value="# {transactionDetail.transactionId.transactionWith}" />

```

```

    </p:column>

```

```

        <p:column headerText="Transaction Status"
filterBy="# {transactionDetail.transactionId.transactionStatus}"
sortBy="# {transactionDetail.transactionId.transactionStatus}">

```

```

        < h : o u t p u t L a b e l
value="# {transactionDetail.transactionId.transactionStatus}" />

```

```

    </p:column>

```

```

    <p:column headerText="Product">

```

```

        <h:outputLabel value="# {transactionDetail.productId.name}" />

```

```

    </p:column>

```

```

    <p:column headerText="Quantity">

```

```

        <h:outputLabel value="#{transactionDetail.qty}" />

    </p:column >

    <p:column headerText="Total Price">

        <h:outputLabel value="#{transactionDetail.totalPrice}" />

    </p:column >

</p:dataTable>

<br/>

<p:separator/>

<br/>

<p:commandLink ajax="false" style="float: right ; display: inline">

    <p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>

    <pe:exporter type="pdf" target="thirdtd" fileName="dataTable" />

</p:commandLink>

<div class="clear"></div>

</p:panel>

</h:form>

</ui:define>

</ui:composition>

```

</body>

</html>

listClients.xhtml

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:p="http://primefaces.org/ui"

xmlns:f="http://java.sun.com/jsf/core"

xmlns:pe="http://primefaces.org/ui/extensions">

<body>

<ui:composition template="./WEB-INF/template/mainTemplate.xhtml">

<ui:define name="content">

<h:form>

<p:growl id="growl"/>

<p:panel header="All Clients" >

```

        <p:dataTable      value="#{clientsController.allList}"      var="clients"
paginator="true" rows="10" id="clients1"

                paginatorTemplate="{CurrentPageReport}      {FirstPageLink}
{PreviousPageLink}      {PageLinks}      {NextPageLink}      {LastPageLink}
{RowsPerPageDropdown}"

                r o w s P e r P a g e T e m p l a t e = " 5 , 1 0 "
rendered="#{userController.userPanelrender}">

        <p:column      sortBy="#{clients.name}"      headerText="Name"
filterBy="#{clients.name}">

                <h:outputLabel value="#{clients.name}"/>

        </p:column>

        <p:column      sortBy="#{clients.address}"      headerText="Address"
filterBy="#{clients.address}">

                <h:outputLabel value="#{clients.address}"/>

        </p:column>

        <p:column      sortBy="#{clients.email}"      headerText="Email"
filterBy="#{clients.email}">

                <h:outputLabel value="#{clients.email}"/>

        </p:column>

        <p:column sortBy="#{clients.phone}" headerText="Phone">

                <h:outputLabel value="#{clients.phone}"/>

        </p:column>

```



```

        <p:column sortBy="#{clients.mobile}" headerText="Mobile">

            <h:outputLabel value="#{clients.mobile}"/>

        </p:column>

    </p:dataTable>

    <p:dataTable          value="#{clientsController.allList}"          var="clients"
paginator="true" rows="10" id="clients2"

                        paginatorTemplate="{CurrentPageReport}"          {FirstPageLink}
{PreviousPageLink}          {PageLinks}          {NextPageLink}          {LastPageLink}
{RowsPerPageDropdown}"

                        rowsPerPageTemplate="5,10"                      editable="true"
rendered="#{userController.adminPanelrender}">

        <p:column          sortBy="#{clients.name}"          headerText="Name"
filterBy="#{clients.name}">

            <p:cellEditor>

                <f:facet name="output">

                    <h:outputLabel value="#{clients.name}"/>

                </f:facet>

                <f:facet name="input">

                    <p:inputText          required="true"          value="#{clients.name}"
size="15"/>

                </f:facet>

            </p:cellEditor>

```

```

        </p:column>

        <p:column        sortBy="#{clients.address}"        headerText="Address"
filterBy="#{clients.address}">

            <p:cellEditor>

                <f:facet name="output">

                    <h:outputLabel value="#{clients.address}" />

                </f:facet>

                <f:facet name="input">

                    <p:inputText        required="true"        value="#{clients.address}"
size="15"/>

                </f:facet>

            </p:cellEditor>

        </p:column>

        <p:column        sortBy="#{clients.email}"        headerText="Email"
filterBy="#{clients.email}">

            <p:cellEditor>

                <f:facet name="output">

                    <h:outputLabel value="#{clients.email}" />

                </f:facet>

                <f:facet name="input">

```

```

        <p:inputText      required="true"      value="#{clients.email} "
size="15"/>

    </f:facet>

</p:cellEditor>

</p:column>

<p:column sortBy="#{clients.phone} " headerText="Phone">

    <p:cellEditor>

        <f:facet name="output">

            <h:outputLabel value="#{clients.phone}"/>

        </f:facet>

        <f:facet name="input">

            <p:inputText      required="true"      value="#{clients.phone} "
size="15"/>

        </f:facet>

    </p:cellEditor>

</p:column>

<p:column sortBy="#{clients.mobile} " headerText="Mobile">

    <p:cellEditor>

        <f:facet name="output">

            <h:outputLabel value="#{clients.mobile}"/>

```

```

        </ffacet>

        <ffacet name="input">

            <p:inputText      required="true"      value="#{clients.mobile}"
size="15"/>

        </ffacet>

    </p:cellEditor>

</p:column>

<p:column headerText="Delete">

    <h:commandLink styleClass="ui-icon ui-icon-close" style="margin: 0
auto" action="#{clientsController.destroy(clients)}" />

</p:column>

<p:column headerText="Edit">

    <h:commandLink styleClass="ui-icon ui-icon-pencil" style="margin: 0
auto" action="#{clientsController.prepareEdit(clients)}" />

</p:column>

</p:dataTable>

<br/>

<p:separator/>

<br/>

    <p:commandLink ajax="false" style="float: right ; display: inline"
rendered="#{userController.adminPanelrender}" />

```

```
        <p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>
```

```
        <pe:exporter type="pdf" target="clients1" fileName="AllClients" />
```

```
    </p:commandLink>
```

```
        <p:commandLink ajax="false" style="float: right ; display: inline"
rendered="#{userController.userPanelrender}">
```

```
        <p:graphicImage value="/resources/images/export.jpeg" height="40"
width="35" style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>
```

```
        <pe:exporter type="pdf" target="clients2" fileName="AllClients" />
```

```
    </p:commandLink>
```

```
    <div class="clear"></div>
```

```
    </p:panel>
```

```
    </h:form>
```

```
    </ui:define>
```

```
    </ui:composition>
```

```
    </body>
```

```
    </html>
```

index.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

      xmlns:ui="http://java.sun.com/jsf/facelets"

      xmlns:h="http://java.sun.com/jsf/html"

      xmlns:p="http://primefaces.org/ui"

      xmlns:f="http://java.sun.com/jsf/core">

    <body>

        <ui:composition template="./WEB-INF/template/mainTemplate.xhtml">

            <ui:define name="content1">

                <p:ajaxStatus onStart="statusDialog.show();" onSuccess="statusDialog.hide();"/>

                <p:dialog modal="true" widgetVar="statusDialog" header="Loading..."

                        draggable="false" closable="false" width="245" height="20"

resizable="false">

                    <p:graphicImage value="./resources/images/loadingbar.gif" />

                </p:dialog>

            </h:form>

        </ui:define>

    </body>

</html>
```

```
<p:panel id="panel" header="Login" style="position: absolute; width:
306px;top:35%;left: 50%;margin-left: -153px;">
```

```
<p:focus context="panel"/>
```

```
<p:growl sticky="true" autoUpdate="true"/>
```

```
<h:panelGrid columns="2" cellpadding="5">
```

```
<h:outputLabel for="username" value="User Name: *" />
```

```
<p:inputText id="username"
```

```
value="#{userController.username}"
```

```
required="true" label="User Name" style="float:right">
```

```
<f:validateLength minimum="2" />
```

```
</p:inputText>
```

```
<h:outputLabel for="password" value="Password: *" />
```

```
<p:inputText id="password"
```

```
value="#{userController.password}"
```

```
required="true" label="Password" type="password"
```

```
style="float:right"/>
```

```
</h:panelGrid>
```

```
<p:commandButton value="Submit" update="panel"
```

```
action="#{userController.login()}"/>
```

```
</p:panel>
```

```
</h:form>

</ui:define>

</ui:composition>

</body>

</html>
```

addUser.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

      xmlns:ui="http://java.sun.com/jsf/facelets"

      xmlns:h="http://java.sun.com/jsf/html"

      xmlns:p="http://primefaces.org/ui"

      xmlns:f="http://java.sun.com/jsf/core">

  <body>

    <ui:composition template="./WEB-INF/template/mainTemplate.xhtml">

      <ui:define name="content">

        <h:form>
```



```

<p:panel header="Add/Edit Users">

    <p:growl id="growl" sticky="true" autoUpdate="true"/>

    <h:panelGrid id="confirmation" columns="6">

        <h:outputText value="Employee ID: *" />

        <p:inputText value="#{userController.empID}" readonly="true"/>

        <p:inputText rendered="false"/>

        <p:inputText                                     rendered="false"
value="#{userController.selected.setEmpId(userController.empID)}"/>

        <h:outputText value="Firstname: *" />

        <p:inputText required="true" label="Firstname"

            value="#{userController.selected.FName}" />

        <h:outputText value="Lastname: *" />

        <p:inputText required="true" label="Lastname"

            value="#{userController.selected.LName}" />

        <h:outputText value="DOB: " />

        <p:calendar    required="true"    value="#{userController.selected.dob}"
navigator="true"      yearRange="c-100:c+10"      showButtonPanel="true"
pattern="yyyy-mm-dd"/>

        <h:outputText value="Gender : *" />

        <p:selectOneRadio                                     id="options"
value="#{userController.selected.gender}">

```

```

        <f:selectItem itemLabel="Male" itemValue="male" />

        <f:selectItem itemLabel="Female" itemValue="female" />

        <f:selectItem itemLabel="Third" itemValue="third" />

    </p:selectOneRadio>

    <h:outputText value="Email: *" />

    <p:inputText required="true" label="Email"

        value="#{userController.selected.email}" />

    <h:outputText value="Phone: " />

    <p:inputText                                required="true"
value="#{userController.selected.phone}" />

    <h:outputText value="Address: " />

    <p:inputText required="true" value="#{userController.selected.address}"
/>

    <h:outputText value="UserName: *" />

    <p:inputText required="true" label="Email"

        value="#{userController.selected.username}" />

    <h:outputLabel for="pwd1" value="Password: *" />

    <p:password id="pwd1" value="#{userController.selected.password}"
match="pwd2" label="Password" required="true"/>

    <h:outputLabel for="pwd2" value="Confirm Password: *" />

```

```
<p:password id="pwd2" value="#{userController.selected.password}"
label="Confirm Password" required="true"/>
```

```
<h:outputText value="User Role: *" />
```

```
<p:selectOneMenu                                     required="true"
value="#{userController.selected.role}" style="width: 154px;">
```

```
<f:selectItem itemLabel="Admin " itemValue="admin"/>
```

```
<f:selectItem itemLabel="User " itemValue="user"/>
```

```
</p:selectOneMenu>
```

```
<f:facet name="footer">
```

```
<p:commandButton id="loginButton" update="growl" value="Add"
type="Submit"
```

```
rendered="#{userController.renderBut}" style="float: left ;
display: inline" action="#{userController.create}"/>
```

```
<p:commandButton id="updateButton" update="growl" value="Edit"
type="Submit"
```

```
rendered="#{!userController.renderBut}" style="float: left ;
display: inline" action="#{userController.updateSingle()}/>
```

```
<p:commandButton value="Clear" update="confirmation"
process="@this" immediate="true"
```

```
style="float: right ; display: inline"
rendered="#{userController.renderBut}"/>
```

```
<p:resetInput target="confirmation" />
```

```

        </p:commandButton>

        <div class="clear"></div>

    </f:facet>

</h:panelGrid>

</p:panel>

</h:form>

</ui:define>

</ui:composition>

</body>

</html>

```

addProduct.xhtml

```

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:ui="http://java.sun.com/jsf/facelets"

    xmlns:h="http://java.sun.com/jsf/html"

    xmlns:p="http://primefaces.org/ui"

    xmlns:f="http://java.sun.com/jsf/core">

```

```

<body>

<ui:composition template="/WEB-INF/template/mainTemplate.xhtml">

    <ui:define name="content">

        <h:form id="growl">

            <p:panel header="Add Category">

                <p:growl sticky="true" autoUpdate="true"/>

                <h:panelGrid id="confirmation2" columns="2">

                    <h:outputText value="Category Name: *" />

                    <p:inputText                                required="true"
value="#{categoryController.selected.name}" />

                    <f:facet name="footer">

                        <p:commandButton    id="loginButton"    update="growl;growl1 "
value="Add" type="Submit"

                                rendered="#{categoryController.renderBut}" style="float:
left ; display: inline"

                                action="#{categoryController.create}" />

                        <p:commandButton    id="updateButton"    update="growl;growl1 "
value="Edit" type="Submit"

                                rendered="#{!categoryController.renderBut}" style="float:
left ; display: inline"

```

```

        action="#{categoryController.updateSingle()}" />

        <p:commandButton value="Clear" update="confirmation2"
process="@this"

        rendered="#{categoryController.renderBut}"
immediate="true" style="float: right ; display: inline" >

        <p:resetInput target="confirmation2" />

        </p:commandButton>

        <div class="clear"></div>

        </f:facet>

        </h:panelGrid>

        </p:panel>

    </h:form>

    <h:form id="growl1">

        <p:panel header="Add Prdouct">

            <p:growl sticky="true" autoUpdate="true"/>

            <h:panelGrid id="confirmation" columns="2">

                <h:outputText value="Product Name: *" />

                <p:inputText value="#{productController.selected.name}" />

                <h:outputText value="Description: *" />

                <p:inputText required="true" label="Firstname"

```

```

        value="#{productController.selected.description}" />

<h:outputText value="Category: *" />

<p:selectOneMenu                                required="true"
value="#{productController.selected.categoryId}" style="width: 154px;">

    <f:selectItem itemLabel="Select one" itemValue="" />

    <f:selectItems value="#{categoryController.allList}" var="category"
itemLabel="#{category.name}" itemValue="#{category.id}" />

</p:selectOneMenu>

<h:outputText value="Purchase Price/unit: *" />

<p:inputText required="true" label="Purchase Price/unit"

        value="#{productController.selected.purchasePrice}" />

<h:outputText value="Selling Price/unit: *" />

<p:inputText required="true" label="Selling Price/unit"

        value="#{productController.selected.sellingPrice}" />

<f:facet name="footer">

    <p:commandButton id="loginButton1" update="growl1" value="Add"
type="Submit"

        rendered="#{productController.renderBut}" style="float:
left ; display: inline"

        action="#{productController.create}" />

```

```

        <p:commandButton      id="updateButton1"      update="growl1"
value="Edit" type="Submit"

                                rendered="#{!productController.renderBut}" style="float:
left ; display: inline"

                                action="#{productController.updateSingle()}" />

        <p:commandButton      value="Clear"      update="confirmation"
process="@this" immediate="true"

                                style="float:      right      ;      display:      inline"
rendered="#{productController.renderBut}" />

        <p:resetInput target="confirmation" />

    </p:commandButton>

    <div class="clear"></div>

</f:facet>

</h:panelGrid>

</p:panel>

</h:form>

</ui:define>

</ui:composition>

</body>

</html>

```


addInvoice.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:ui="http://java.sun.com/jsf/facelets"

    xmlns:h="http://java.sun.com/jsf/html"

    xmlns:p="http://primefaces.org/ui"

    xmlns:pe="http://primefaces.org/ui/extensions"

    xmlns:f="http://java.sun.com/jsf/core">

    <body>

        <ui:composition template="./WEB-INF/template/mainTemplate.xhtml">

            <ui:define name="content">

                <h:form>

                    <p:panel header="Create Invoice" id="customersupplier">

                        <p:growl id="growl" sticky="true" autoUpdate="true"/>

                        <p:panelGrid id="firstdt" columns="9" >

                            <h:outputText value="Date Created: *" />

                            <p:calendar
                                value="# { transactionController.selected.dateCreated } "
                                required="true"
```

```
disabled="#{transactionController.viewer}" navigator="true" yearRange="c-100:c+10"
showButtonPanel="true" pattern="yyyy-MM-dd"/>
```

```
<h:outputText value="Transaction With: *" />
```

```
< p : s e l e c t O n e M e n u
value="#{transactionController.selected.transactionWith}"
disabled="#{transactionController.viewer}">
```

```
<f:selectItem itemLabel="Select one" itemValue=""/>
```

```
<f:selectItems value="#{clientsController.allList}" var="clients"
itemLabel="#{clients.name.toUpperCase()}" itemValue="#{clients.name}"/>
```

```
</p:selectOneMenu>
```

```
<h:outputText value="Transaction Status: *" />
```

```
<p:selectOneMenu required="true"
value="#{transactionController.selected.transactionStatus}"
disabled="#{transactionController.viewer}">
```

```
<f:selectItem itemLabel="Select one" itemValue=""/>
```

```
<f:selectItem itemLabel="Buying" itemValue="buying" />
```

```
<f:selectItem itemLabel="Selling" itemValue="selling"/>
```

```
</p:selectOneMenu>
```

```
<p:commandButton value="Create"
actionListener="#{transactionController.create}" update="customersupplier,seconddt,tran"
disabled="#{transactionController.viewer}"/>
```

```
</p:panelGrid><br/>
```

```

        <p:panelGrid                                id="seconddt"                columns="10"
rendered="#{transactionController.viewer}" >

        <h:outputText                                id="tran"
value="#{transactionDetailController.selected.setTransactionId(transactionController.select
ed)}" />

        <h:outputText value="Products: *" />

        < p : s e l e c t O n e M e n u
value="#{transactionDetailController.selected.productId}">

        <f:selectItem itemLabel="Select one" itemValue="" />

        < f : s e l e c t I t e m s
value="#{productController.allListWithGreater5(transactionController.selected,viewBuying
Controller,viewSellingController)}"                                var="product"
itemLabel="#{product.name.toUpperCase()}" itemValue="#{product}" />

        <p:ajax event="change" update="out2,out1,out" />

        </p:selectOneMenu>

        <h:outputText value="Quantity: *" />

        <p:spinner id="out2" value="#{transactionDetailController.selected.qty}"
size="5"                                min="1"
max="#{transactionController.returnMax(transactionDetailController.selected.productId.id,
viewBuyingController,viewSellingController)}" >

        <p:ajax                                event="change"                update="out"
listener="#{transactionDetailController.amtCalculator}" />

```

```

</p.spinner>

<h:outputText value="Price: "/>

<h:outputText id="out1" value="#{transactionDetailController.price}"
style="width: 50px;"/>

<h:outputText value="Amount: *" />

<p:inputText required="true" id="out" readonly="true"
value="#{transactionDetailController.selected.totalPrice}"/>

<p:commandButton value="Submit"
action="#{transactionDetailController.create}" update="customersupplier"/>

</p.panelGrid>

<p:dataTable value="#{transactionDetailController.tempList}"
var="transactionDetail" rows="10" id="thirddt">

<p:columnGroup type="header">

<p:row>

<p:column colspan="5" headerText="Transaction Details With
#{transactionController.selected.transactionWith}"/>

</p:row>

<p:row>

<p:column colspan="2" headerText="Transaction Status:
#{transactionController.selected.transactionStatus.substring(0,
1).toUpperCase().concat(transactionController.selected.transactionStatus.substring(1))}"
/>

```

```

                <p:column colspan="3" headerText="Transaction
Date:#{transactionController.selected.dateCreated}" />

            </p:row>

            <p:row>

                <p:column headerText="Product"/>

                <p:column headerText="Price/piece"/>

                <p:column headerText="Details"/>

                <p:column headerText="Quantity"/>

                <p:column headerText="Total Price"/>

            </p:row>

        </p:columnGroup>

        <p:column>

            < h : o u t p u t L a b e l
value="# {transactionDetail.productId.name.substring(0,
1).toUpperCase().concat(transactionDetail.productId.name.substring(1))}" />

        </p:column>

        <p:column>

            < h : o u t p u t L a b e l
rendered="# {transactionController.selected.transactionStatus=='buying'} "
value="Rs.#{transactionDetail.productId.purchasePrice}.00" style="float: right"/>

```

```

        < h : o u t p u t L a b e l
rendered="# {transactionController.selected.transactionStatus=='selling'} "
value="Rs.#{transactionDetail.productId.sellingPrice}.00" style="float: right"/>

    </p:column >

    <p:column>

        <h:outputLabel value="# {transactionDetail.productId.description}"/>

    </p:column >

    <p:column>

        <h:outputLabel value="# {transactionDetail.qty}" style="margin: 0
auto"/>

    </p:column >

    <p:column>

        <h:outputLabel value="Rs.#{transactionDetail.totalPrice}.00"
style="float: right"/>

    </p:column>

    <p:columnGroup type="footer">

        <p:row>

            <p:column colspan="4" footerText="Total Amount:"
style="text-align:right" />

            < p : c o l u m n
footerText="Rs.#{transactionDetailController.totalAmount}.00" style="text-align:right;"/>

        </p:row>

```

```

        </p:columnGroup>

        </p:dataTable><br/>

        <p:separator/><br/>

        <p:commandLink    ajax="false"    style="margin-left:    400px;    "
rendered="# {transactionController.viewer} ">

        <p:graphicImage    value="/resources/images/export.jpeg"    height="40"
width="35"    style="-moz-border-radius: 7px;border-radius: 7px;-webkit-border-radius:
7px;border: 3px solid #006600"/>

        <pe:exporter type="pdf" target="thirddt" fileName="dataTable" />

        </p:commandLink>

        <p:commandButton            value="Submit"            type="submit"
action="# {transactionController.createInvoice(transactionDetailController)} "    style="float:
left ; display: inline"/>

        <p:commandButton    value="Clear"    style="float: right ; display: inline"
action="# {transactionController.clearAll(transactionDetailController)} "
update="customersupplier">

        <p:resetInput target="thirddt,firstdt,seconddt" />

        </p:commandButton>

        <div class="clear"></div>

        </p:panel>

    </h:form>

```

```

        </ui:define>

    </ui:composition>

</body>

</html>

```

addClients.xhtml

```

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

    xmlns:ui="http://java.sun.com/jsf/facelets"

    xmlns:h="http://java.sun.com/jsf/html"

    xmlns:p="http://primefaces.org/ui"

    xmlns:f="http://java.sun.com/jsf/core">

    <body>

        <ui:composition template="/WEB-INF/template/mainTemplate.xhtml">

            <ui:define name="content">

                <h:form>

                    <p:panel header="Add Clients" id="confirmation">

```



```

<p:growl id="growl" autoUpdate="true"/>

<h:panelGrid columns="2">

    <h:outputText value="Address: *" rendered="false"/>

    <p:inputText required="true" label="sn"

        value="#{clientsController.selected.sn}" rendered="false"/>

    <h:outputText value="Name: *" />

    <p:inputText required="true" label="Name"

        value="#{clientsController.selected.name}" />

    <h:outputText value="Address: *" />

    <p:inputText required="true" label="Address"

        value="#{clientsController.selected.address}" />

    <h:outputText value="Email: " />

    <p:inputText                required="true"                label="Email"
value="#{clientsController.selected.email}" />

    <h:outputText value="Phone: *" />

    <p:inputText required="true" label="Phone"

        value="#{clientsController.selected.phone}" />

    <h:outputText value="Mobile: " />

    <p:inputText                required="true"                label="Moblie"
value="#{clientsController.selected.mobile}" />

```

```

        <p:commandButton id="loginButton" update="growl" value="Submit"
type="Submit"

        rendered="#{clientsController.renderBut}" style="float: left ;
display: inline"

        action="#{clientsController.create}"/>

        <p:commandButton id="updateButton" update="growl" value="Edit"
type="Submit"

        rendered="#{!clientsController.renderBut}" style="float: left ;
display: inline" action="#{clientsController.updateSingle()}" />

        <p:commandButton value="Clear" update="confirmation"
process="@this" immediate="true"

        style="float: right ; display: inline"
rendered="#{clientsController.renderBut}" />

        <p:resetInput target="confirmation" />

    </p:commandButton>

</h:panelGrid>

<br/>

</p:panel>

</h:form>

</ui:define>

</ui:composition>

</body>

```

</html>

Chapter 10: Bibliography

1. <http://www.ijdv1.com/article.asp?issn=0378-6323;year=2008;volume=74;issue=6;spage=687;epage=690;aulast=Betkerur>
2. <http://www.scribd.com/doc/22286019/project-report-on-sales-and-inventory-management-System>
3. <http://www.slideshare.net/ieeexploreprojects/academic-project-inventory-management-system-synopsis-14270983>
4. <http://inventory-management.findthebest.com/compare/14-19/POS-Maid-vs-inFlow-Inventory-Software>
5. <http://www.inflowinventory.com/>
6. <http://alexandriacomputers.com/pos-maid-screenshots-and-features/>