

# **Textile Workorder Management System**

Submitted by

Farhad Mahmud Khan

**ID:** 2011-2-96-005

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Computer Science and Engineering

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**EAST WEST UNIVERSITY BANGLADESH**

**May 2015**

## **ABSTRACT**

In this project we developed Textile Workorder Management System that briefly describes about the requirements, design and all special features of this system. Through this system our traditional textile work process will turn into an automated textile work process, where every order quantity, order list, buyer list, buyer information, deadline schedule, fabrics receive, cutting information, printing information, embroidery information, production information, finishing information will be viewed. Every department user has a User ID and Password. So they can login to their panel and they will be able to update their work process, see their present status and print report. This software will provide the Admin a clear view of product current information for delivery the product in due time to buyer.

## DECLARATION

I hereby, declare that all the work presented in this project is the outcome of the investigation and research performed by me under the supervision of Md. Shamsujjoha, Lecturer, Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh. I also declare that neither it nor part of it has been submitted for the requirement of any degree or diploma or for any other purposes except for publications.

Countersigned

Signature

.....

.....

(Md. Shamsujjoha

Farhad Mahmud Khan

Lecturer

Department of CSE

East West University

Dhaka, Bangladesh)

## **LETTER OF ACCEPTANCE**

The project entitled “Textile Workorder Management System” is submitted by Farhad Mahmud Khan, Id: 2011–2–96–005 to the department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh is accepted by the Department for the partial fulfillment of the requirements for the degree of MS in Computer Science and Engineering.

Dr. Shamim H. Ripon

Associate Professor

and Chairperson

Department of Computer

Science & Engineering

East West University

Dhaka, Bangladesh

Md. Shamsujjoha

Lecturer

Department of Computer

Science & Engineering

East West University

Dhaka, Bangladesh

## **ACKNOWLEDGEMENT**

I am truly grateful to Almighty Allah, whose blessings have always been enormous and who gave me the ability and strength to complete this project. I would like to dedicate my project to my parents, who enlightened me the value of education and always keep me on the right track. It is a great honor and pleasure for me to record this deep sense of gratitude and insightful indebtedness to my respected supervisor, Md. Shamsujjoha for his valuable contribution, constant guidance, intuitive advice, helpful criticism, valuable suggestions, commendable support, and also endless patience for the completion of this project work. I am very much grateful to him and feel proud to have worked with him because it was not possible for me to complete this work without his inspiring enthusiasm and encouragement.

## **Table of Content**

|                            |     |
|----------------------------|-----|
| Abstract .....             | II  |
| Declaration .....          | III |
| Letter of Acceptance ..... | IV  |
| Acknowledgement .....      | V   |
| List of figures .....      | IX  |
| List of tables .....       | XI  |

## **Chapter 1: Introduction**

|                                       |   |
|---------------------------------------|---|
| 1.1 Introduction .....                | 1 |
| 1.2 Motivation .....                  | 1 |
| 1.3 Objectives .....                  | 2 |
| 1.4 Contributions .....               | 3 |
| 1.5 Organization of the Project ..... | 3 |
| 1.6 Summary .....                     | 4 |

## **Chapter 2: Review of the Existing System, Requirements for the new System and System Design**

|                                               |    |
|-----------------------------------------------|----|
| 2.1 Existing System .....                     | 5  |
| 2.2 Requirement Analysis .....                | 5  |
| 2.3 System Module .....                       | 5  |
| 2.4 Requirement List .....                    | 6  |
| 2.5 Actors .....                              | 7  |
| 2.6 Use Case Diagram .....                    | 9  |
| 2.6.1 Use Case Diagram for Normal Users ..... | 10 |

|                                                |    |
|------------------------------------------------|----|
| 2.6.2 Use Case Diagram for System Manager----- | 11 |
| 2.6.3 Use Case Diagram for System Admin-----   | 12 |
| 2.7 Entity-Relationship Diagram (ERD) -----    | 13 |
| 2.8 Summary -----                              | 15 |

### **Chapter 3: Implementation, Database Design and Testing**

|                                  |    |
|----------------------------------|----|
| 3.1 Software Implementation----- | 16 |
| 3.1.1 HTML-----                  | 16 |
| 3.1.2 PHP-----                   | 16 |
| 3.1.3 MySQL-----                 | 16 |
| 3.1.4 XAMPP -----                | 17 |
| 3.2 Hardware Implementation----- | 17 |
| 3.3 Database Design-----         | 17 |

### **Chapter 4: User interface and Manual**

|                                              |    |
|----------------------------------------------|----|
| 4.1 Software Requirement for the system----- | 23 |
| 4.2 Server Requirement-----                  | 23 |
| 4.3 Work station requirement-----            | 24 |
| 4.4 Installation-----                        | 25 |
| 4.4.1 Installation of XAMPP Server-----      | 25 |
| 4.5.1 User login Page-----                   | 26 |
| 4.5.2 Home page -----                        | 26 |
| 4.5.3 Add new buyer page -----               | 27 |
| 4.5.4 Add new cutting page-----              | 27 |
| 4.5.5 Add new embroidery page -----          | 28 |

|                                      |    |
|--------------------------------------|----|
| 4.5.6 Add new fabric page -----      | 28 |
| 4.5.7 Add new finishing page -----   | 29 |
| 4.5.8 Add new production page -----  | 29 |
| 4.5.9 Add new user page -----        | 30 |
| 4.5.10 Add new workorder page -----  | 30 |
| 4.5.11 View buyer info page -----    | 31 |
| 4.5.12 View buyer ratings page ----- | 31 |
| 4.5.13 View company info page -----  | 32 |
| 4.5.14 View user setting -----       | 32 |
| 4.5.15 View Cutting page -----       | 33 |
| 4.5.16 View embroidery page -----    | 33 |
| 4.5.17 View fabrics page -----       | 34 |
| 4.5.18 View groups page -----        | 34 |
| 4.5.19 View workorder page -----     | 35 |
| 4.5.20 View printing page -----      | 35 |
| 4.5.21 View Production page -----    | 36 |

## **Chapter 5: Conclusion and Future Works**

|                                 |    |
|---------------------------------|----|
| Conclusion and Future Work----- | 37 |
|---------------------------------|----|

## **References**

|                 |    |
|-----------------|----|
| Reference ----- | 38 |
|-----------------|----|

## **Annexure**

|                  |    |
|------------------|----|
| Sample Code----- | 39 |
|------------------|----|



## List of Figures

| <b>Figures</b>                                          | <b>Page</b> |
|---------------------------------------------------------|-------------|
| Figure 2.1: Use Case Diagram for Normal users -----     | 10          |
| Figure 2.2: Use Case Diagram for System Manager -----   | 11          |
| Figure 2.3: Use Case Diagram for System Admin -----     | 12          |
| Figure 2.4: Entity relationship diagram(ERD) -----      | 14          |
| Figure 3.1: User information table -----                | 18          |
| Figure 3.2: Buyer Information Table -----               | 18          |
| Figure 3.3: Company Information Table -----             | 19          |
| Figure 3.4: Cutting process information Table -----     | 19          |
| Figure 3.5: Embroidery process information Table -----  | 20          |
| Figure 3.6: Fabrics information Table -----             | 20          |
| Figure 3.7: Finishing information Table -----           | 21          |
| Figure 3.8: Workorder information Table -----           | 21          |
| Figure 3.9: Printing information Table -----            | 22          |
| Figure 3.10: Production information Table -----         | 22          |
| Figure 4.1: login page -----                            | 26          |
| Figure 4.2: Home page -----                             | 26          |
| Figure 4.3: Add new buyer page is shown here -----      | 27          |
| Figure 4.4: Add new cutting page is shown here -----    | 27          |
| Figure 4.5: Add new embroidery page is shown here ----- | 28          |
| Figure 4.6: Add new fabric page is shown here -----     | 28          |
| Figure 4.7: Add new finishing page is shown here -----  | 29          |
| Figure 4.8: Add new production page is shown here ----- | 29          |

|                                                          |    |
|----------------------------------------------------------|----|
| Figure 4.9: Add new user page is shown here -----        | 30 |
| Figure 4.10: Add new workorder page is shown here -----  | 30 |
| Figure 4.11: View buyer info page is shown here -----    | 31 |
| Figure 4.12: View buyer ratings page is shown here ----- | 31 |
| Figure 4.13: View company info page is shown here -----  | 32 |
| Figure 4.14: Add new buyer page is shown here -----      | 32 |
| Figure 4.15: View cutting page is shown here -----       | 33 |
| Figure 4.16: View embroidery page is shown here -----    | 33 |
| Figure 4.17: View fabrics page is shown here -----       | 34 |
| Figure 4.18: View groups page is shown here -----        | 34 |
| Figure 4.19: View workorder page is shown here -----     | 35 |
| Figure 4.20: View printing page is shown here -----      | 35 |
| Figure 4.21: View production page is shown here -----    | 36 |

## List of Tables

| <b>Tables</b>                   | <b>Page</b> |
|---------------------------------|-------------|
| Table 1: Requirement List ----- | 6           |
| Table 2: Actors -----           | 7           |

# Chapter: 1

## Introduction

### 1.1 Introduction

The textile industry has played an important role in Bangladesh. Bangladesh's textile industry is comprised of a mix of small to large scale privately and publicly owned companies. In our county, most of the textile workorders are paper oriented. Though there are some organizations which are using automated system, but the number is very small. When business automation planning indicates the need for additional labor, organizations have a number of choices to make. This may be the first step in a full-scale operation process, but sometimes hiring additional employees is not the best method to obtain additional labor. It may be appropriate for an organization to consider alternatives for full scale automation, such as developing and implementing module wise, instate of developing a full system. A careful automation planning must consider the overall growth prospects of the organization and accurate forecasting of future needs.

### 1.2 Motivation

Having been exposed and have had a good and long experienced hand in IT expertise in my professional and career line up, I have been motivated towards the automation of Textile Workorder Management System to eliminate the drawbacks and ineffectiveness in the traditional system adopted by Bangladesh in this regard.

The purpose of this project is to develop an application that is totally feasible, secured and obtainable. This project is needed not only for convenience; cost saving and improved efficiency in the related process, but most especially it will completely replace the manual system.

The scopes of the developed system are very dynamic. Dynamic in the scene of, the user of this system can use it dynamically. With this system, Admin can manage the workorder information as well as the information of product shipment.

### 1.3 Objectives

To make a successful and effective textile workorder system, I started thinking of removing the drawbacks and loopholes of the traditional Textile Workorder Management System which is mainly manual and paper based document. In this Workorder Management System, I found some ineffectiveness which makes the program in vain. So, my objectives of this program is to removes the defects of this traditional Textile Workorder Management System into a automated, software and most of all, web-based tool where Admin can restore all relevant data until the completion the workorder system with the help of IT Technology to follow up, to a successful completion.

The broad objective of this project is to develop an automatic system to automate the process of the textile workorder management system. Where the system should be user friendly, scalable, secured. The system has to have some technology to mitigate any sort of disaster with proper backup plan. To replace the traditional system, the new system has to have some more features, the cost over the system should be very efficient in the respect to time and money and human hour.

This system will minimize the following disadvantages of our traditional textile workorder management system

- i. **Paper intensive:** From the initial stage of recording of workorder is fully based on paper forms.
- ii. **Time Consuming:** It takes significant amount of time to search any information regarding the activities of workorder such as how many product do not take schedule shipment for a specific period of time

- iii. **Prone to missing reports:** Reports are prone to be lost not only during the manual processing but also during the manual storage. There is an unacceptably high chance of misplacing the reports or even data while it is in the processing phase.
- iv. **No opportunity for backup of data:** As the whole process is totally paper based, so, all data regarding workorder are preserved only once in the register. If the register is lost or damaged, there is no way to recover the data except collect from the user which is most difficult task.

## 1.4 Contribution

The primary philosophical theme of this project is the concept of Automation of textile workorder management system. The work explores the concepts of automated systems, processes and phenomena from the workorder perspective. The project attempts to represent a true secure automated online monitoring system. Since manual monitoring systems are naturally un-optimized and unauthentic, in terms of their interaction and process, this model inherent design, structure and dynamics of such systems, with abilities of better processes. The basic objective of this project was to forward ideas and concepts on automation towards new world's automated process. In this system the contributions;

- i. A scalable, automatic computational program.
- ii. Targeted to solve specific problems.
- iii. Has been designed and developed for large-scale multidimensional integration management system.
- iv. Developing a cost efficient System.

## 1.5 Organization of the Project

The remainder of this project paper is structured as follows:

In Chapter 2, describe the background study of this project. Also provides a broad overview of the Use Case Diagram and ER Diagram.

In Chapter 3, discuss the user interface and manual for both a user and an Admin. In broad, how the Admin will use this software, and which button is doing what, is clearly mentioned in this chapter.

In Chapter 4, describes the complete database design, implementation and testing.

Finally, Chapter 5 provides some conclusions derived from the project and explores some future directions for this work.

## **1.6 Summary**

The broad objective of this project is to develop an automatic system to automate the process of the textile workorder management system. Where the system should be user friendly, scalable, secured. The system has to have some technology to mitigate any sort of disaster with proper backup plan. To replace the traditional system, the new system has to have some more features, the cost over the system should be very efficient with respect to time and money and human hour.

# Chapter: 2

## Background Study

### 2.1 Existing System

The most common method used for monitoring textile work process maintains a register to record work order and product information. The buyers informs to the clients of the workorder about the schedule of shipment of their buyer. At the scheduled date, clients send their product for shipment. This total process works manually, and the data sent through paper. For any important announcement to the clients is always some manual way. It is the responsibility of the clients of the product to remember the shipment date. If they forget to remember or they miss to take schedule for shipment of their product, the client has nothing to do as the client has no automated system to find product for shipment.

### 2.2 Requirement Analysis

The requirement analysis includes a column to show which uses cases to provide the functionality of each requirement. The Textile Workorder Management System software requires the following requirements. There will be only four actors. Those are:

- Normal user
- Data entry operator
- System Manager
- System Admin.

### 2.3 System Module

The Textile Workorder Management System software requires the following Module

- ⑤ Product information management



- ⑤ Sipment management
- ⑤ Authentication and security service management
- ⑤ Documentation management

## 2.4 Requirement List

The requirement list includes a column to show which use cases provide the functionality of each requirement.

**Table 1. Requirement List**

| No. Requirement                                                                  | Use Case(s)              |
|----------------------------------------------------------------------------------|--------------------------|
| 1 To record buyer name, work order no, style no color, item, quantity, deadline  | Add new order            |
| 2 To view order information                                                      | View order               |
| 3 To update buyer name, work order no, style no, color, item, quantity, deadline | Edit order               |
| 4 To delete all information of a order                                           | Delete order             |
| 5 To view company information                                                    | View company information |
| 6 To edit company information                                                    | Edit company information |
| 7 To add buyer name, state, city, country phone no, email                        | Add new buyer            |
| 8 To edit buyer name, state, city, country phone no, email                       | Edit new buyer           |
| 9 To delete buyer name, state, city, country phone no, email                     | Delete new buyer         |
| 10 To view buyer ratings                                                         | View buyer ratings       |
| 11 To add new user                                                               | Add new user             |
| 12 To edit user                                                                  | Edit user                |
| 12 To reset password                                                             | Password reset           |

|                                                                                        |                             |
|----------------------------------------------------------------------------------------|-----------------------------|
| 14 To delete user                                                                      | Delete user                 |
| 15 To view group                                                                       | View group                  |
| 16 To edit group                                                                       | Edit group                  |
| 17 To add fabric unit, total receive, total return,<br>Net receive, total issue        | Add fabrics<br>Add cuttings |
| 18 To add cutting line no, quantity                                                    | Add embroidery              |
| 19 To add embroidery total receive, total<br>Embroidery, total output                  | Add printing                |
| 20 To add printing total printing, finish<br>Printing, hand quantity                   | Add Production              |
| 21 To add production line no, input quantity,<br>Output quantity, input balance        | Add Finishing               |
| 22 To add finishing shipped quantity, cartoon<br>Quantity, short, excess, mode of ship |                             |

## 2.5 Actors

Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications.

**Table 2. Actors**

| <b>Actor</b>   | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Manager | System manager can insert, edit, delete and view, the entire running system's statue, status of deliverables, activities of the textile workorder system. He/she will provide authentication / security service to the users so that the system will be secured from any type of threat. He/she can delete all information of a user if required. He/she can delete buyer name and schedule of the shipment if required. He/she can delete documentation if required. |

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Normal Users | <p>Normal users can access into the system without authentication.</p> <p>Normal users will be able to see buyer order and product deadline schedule, departmental report and can communicate with Workorder Management System.</p>                                                                                                                                                                                                                                                   |
| System Admin | <p>System Admin can insert, edit, delete and view, the entire running system's statue, status of deliverables, activities of the textile workorder system. He/she will provide authentication / security service to the users so that the system will be secured from any type of threat. He/she can not add or delete all information of a user if required. He/she can delete buyer name and schedule of the shipment if required. He/she can delete documentation if required.</p> |

## 2.6 Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

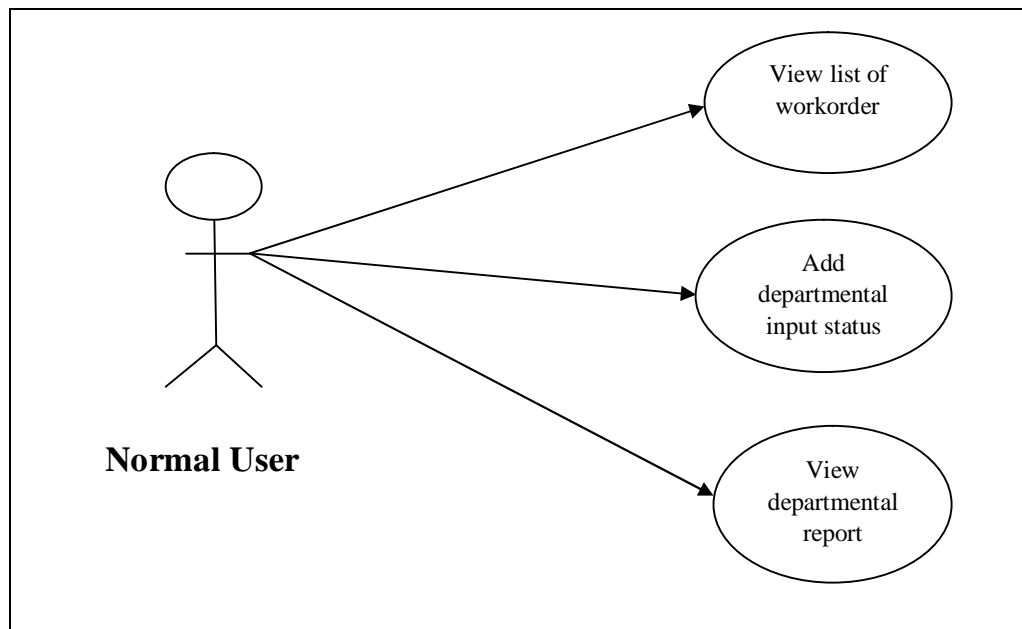
- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

A use case diagram looks something like a flowchart. Intuitive symbols represent the system elements.

The Unified Modeling Language's (UML) graphical notations were employed in this project as a technique to model out the user's requirements. UML is a standardized notation and provides a diagramming technique that can be used to document the client's requirements (Bennett, 2002). UML is a language used to specify, visualize and document the objects of the system. (Whitten, 2001) defined UML, is a set of modeling conventions that used to specify or describe a software system in terms of objects. The UML was chosen because it is one of the techniques, which is easy to use and understand. Microsoft Office Visio 2007 has been used to draw the CASE diagram. The UML diagram that was constructed in this project is use case diagram. The following section defines the different Use Case Diagram which is used to determine the requirements of the system.

### 2.6.1 Use Case Diagram for Normal Users

The use case diagram of a Normal user, who do not required to logon to access into the system to see all report. User can view list of workorder, add departmental input status and view departmental report. The details case diagram is shown in the figure below



*Figure 2.1: Use Case Diagram for Normal users*

## 2.6.2 Use Case Diagram for System Manager

The use case diagram a registered user, who do not required to logon to access into the system to see all report, shipment schedule, and important documentation. User can change buyer setting, user setting, group setting and workorder input.

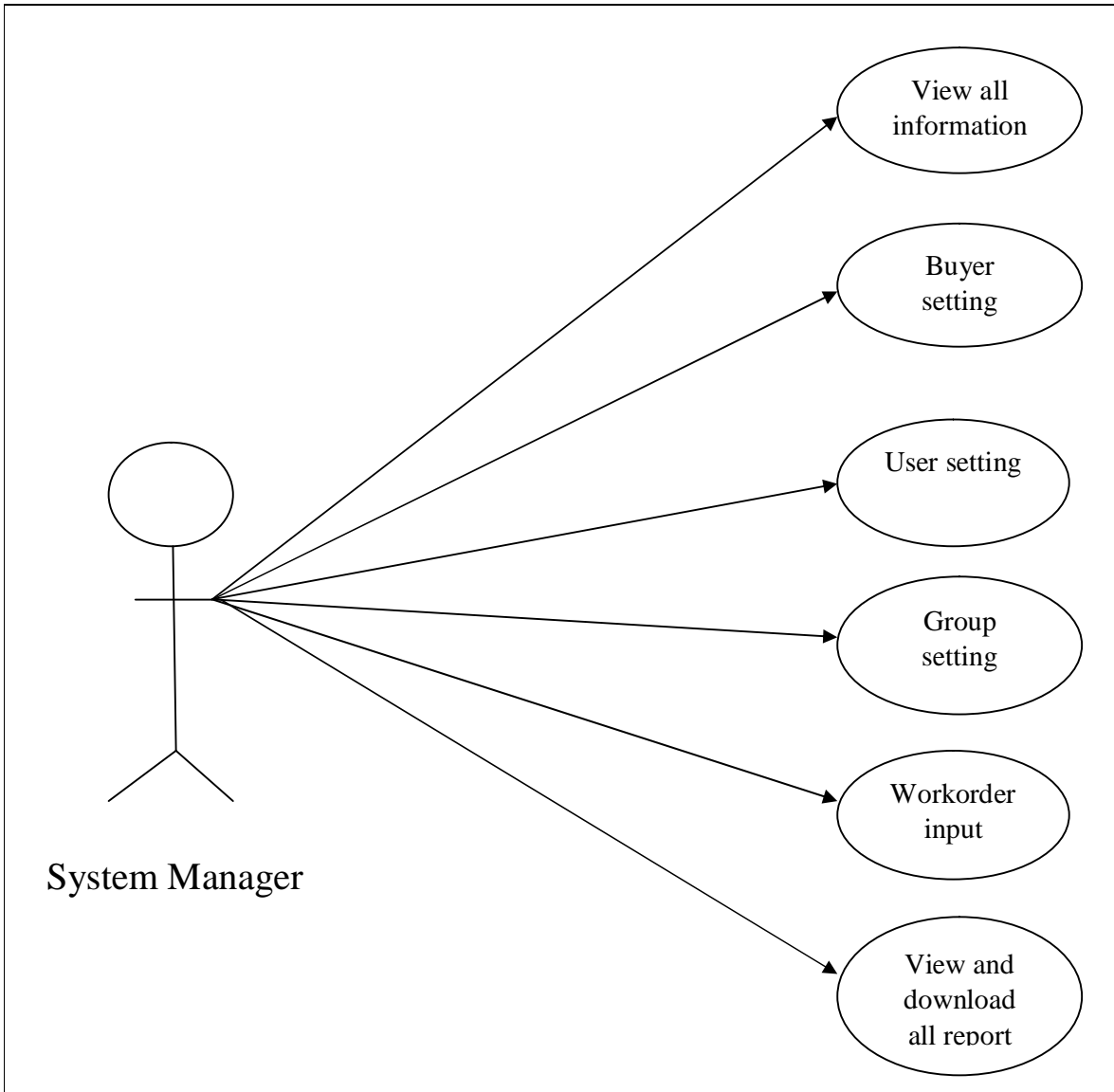


Figure 2.2: Use Case Diagram for System Manager

### 2.6.3 Use Case Diagram for System Admin

The use case diagram a registered user, who do not required to logon to access into the system to see all report, shipment schedule, and important documentation. User can change buyer setting and workorder input.

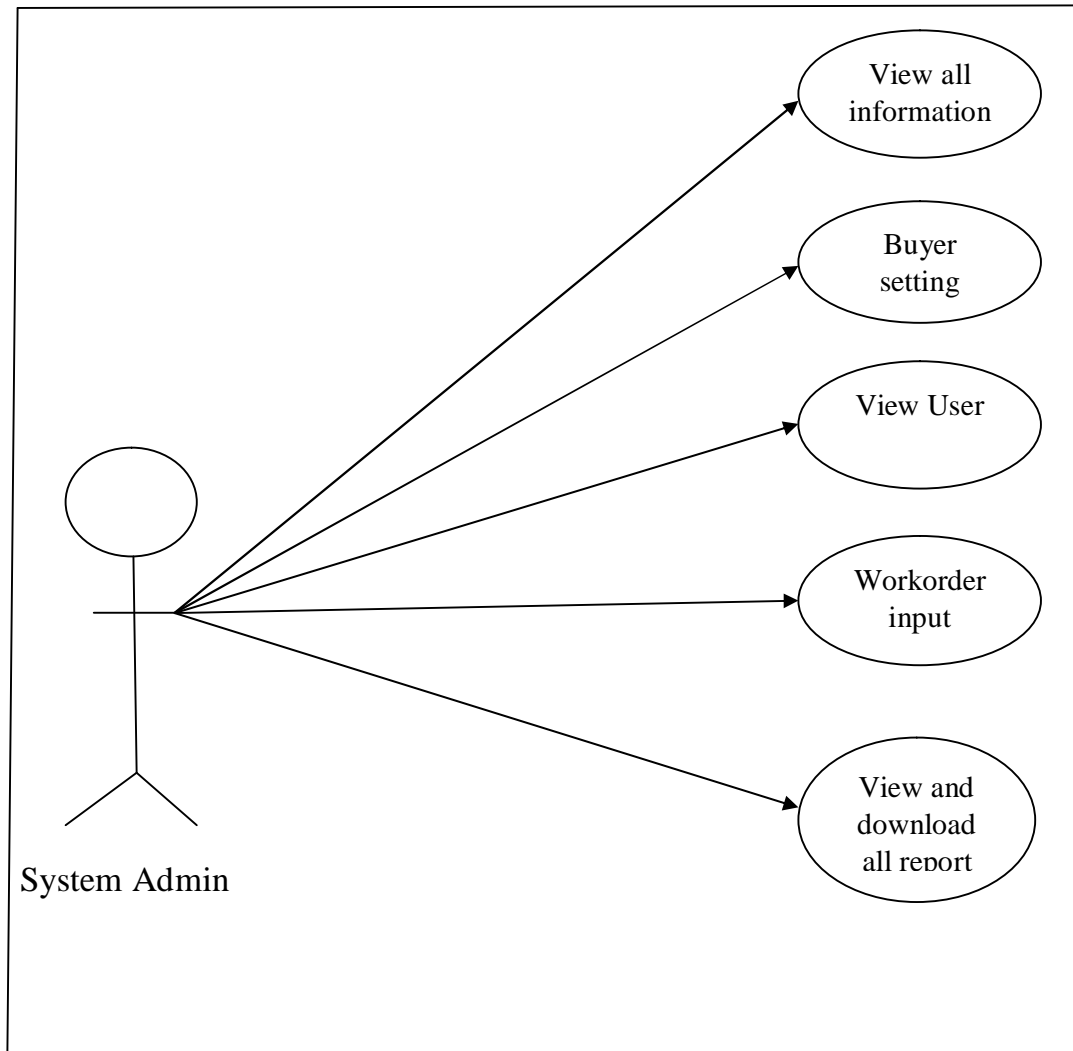


Figure 2.3: Use Case Diagram for System admin

## **2.7 Entity-Relationship Diagram (ERD)**

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions. An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. The elements of an ERD are: Entities, Relationships and Attributes.



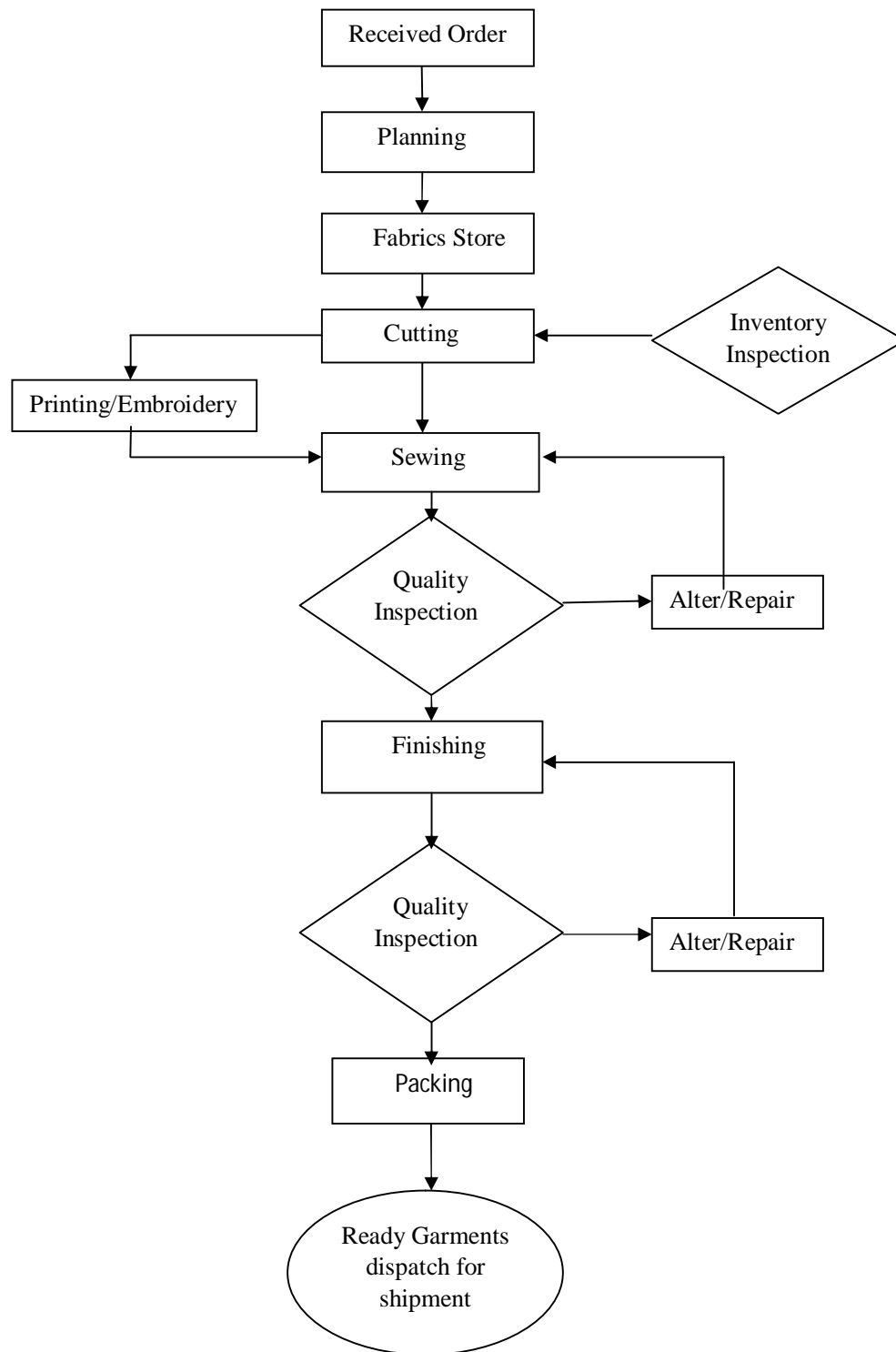


Figure 2.4: Entity Relationship Diagram (ERD)

In the diagram, the elements inside rectangles are called entities while the items inside diamonds denote the relationships between entities. While able to describe just about any system, ER diagrams are most often associated with complex databases that are used in software engineering and IT networks. In particular, ER diagrams are frequently used during the design stage of a development process in order to identify different system elements and their relationships with each other.

In the above section the E-R Diagram of the Textile Workorder Management System is given.

This is the Entity Relationship Diagram of Textile Workorder Management System. In this diagram it shows that, between Workorder and Shipment Schedule entity there is a one to one relation. Because, one workorder has one shipment schedule.

## **2.8 Summary**

With the continuing advance of technology, one of the more recent trends of the 21st century has been the so-called paperless office. A paperless office is not actually completely paperless, but rather uses a minimum of paper and converts as much documentation as possible to digital form. A manual reporting system is a noticeable obstacle to make the workorder process paperless and smart. Moreover manual reporting is done manually.

The main goal of this project was to forward ideas and concepts on automation towards new world's automated process. Where, the system should be user friendly, scalable, secured. The system has to have some technology to mitigate any sort of disaster with proper backup plan. To replace the traditional system, the new system have to have some more features, the cost over the system should be very efficient in the respect to time and money and human hour.

# Chapter: 3

## Proposed Implementation of the Textile Workorder Management System

### 3.1 Software Implementation

To implement this software, the tools were used are fully open source, so that there are no costing involves to develop this software. For designing this project, HTML is used which is open source, Apache web server is used as web server, PHP is used for database coding, MySQL is used as database server.

#### 3.1.1 HTML

In this project, the website is designed using HTML (Hyper Text Markup Language) because; the HTML is very easy to use. It supports on almost every browser in the client end. It used widely. Very easy to learn, and the most importantly, HTML is free.

#### 3.1.2 PHP

PHP stands for Hypertext Preprocessor, PHP is chosen because, it is a Server Side Language, and execute on server side. It supports various types of database, like ORACLE, MySQL. It is easy to use and also a open source software [13].

#### 3.1.3 MySQL

In this project as a database server, MySQL is a small database server; MySQL used in this project because of, it is very efficient for small or medium size application. Supports standard Structured Query Language (SQL). It complies with number of platforms like; Windows, Linux, SUN, UNIX etc. MySQL is also a free tool [14].

### **3.1.4 XAMPP**

XAMPP is a Windows web development environment. XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl. XAMPP is really very easy to install and to use - just download, extract and start. It allows creating web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows managing easily database on the project [15].

## **3.2 Hardware Implementation**

For implementing this project we will required some extra hardware like, server (for running the software, keeping data, and database).

## **3.3 Database Design**

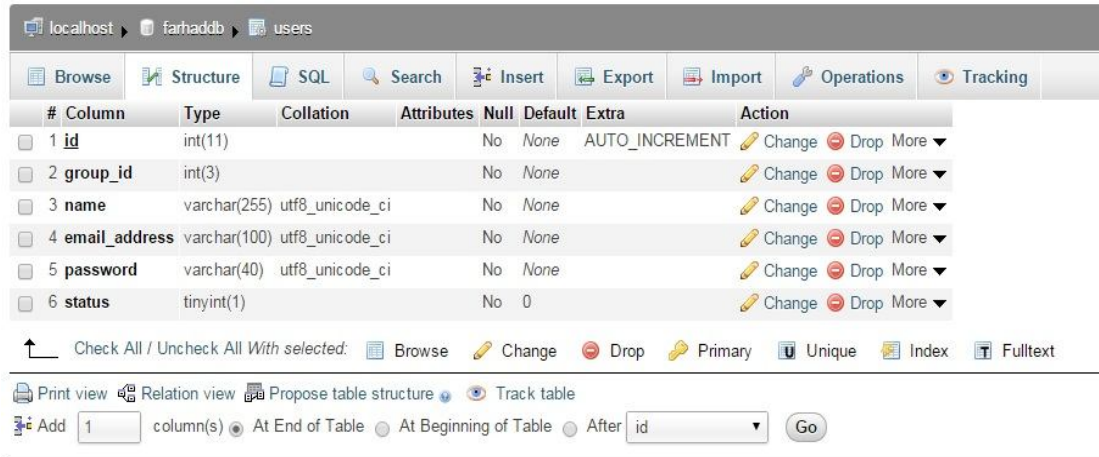
Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system. The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- Determine the relationships between the different data elements.
- Superimpose a logical structure upon the data on the basis of these relationships.

In the following section a detailed database structure of the Textile Workorder Management System is elaborated with all the details.

### Table Number 1: user Purpose: User Information

**Description:** This table will contain the user information. Users are who is using this software for administration. This table will be using for authentication of users.



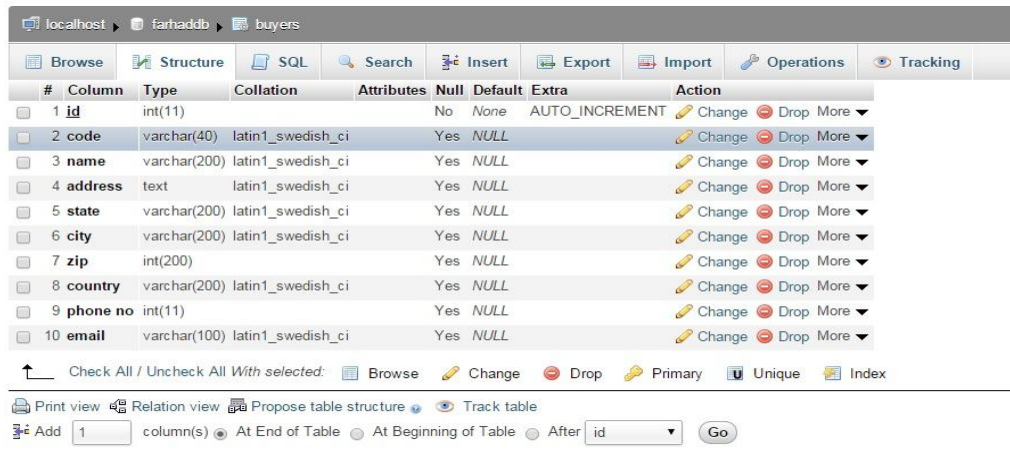
| #                        | Column          | Type         | Collation       | Attributes | Null | Default | Extra          | Action             |
|--------------------------|-----------------|--------------|-----------------|------------|------|---------|----------------|--------------------|
| <input type="checkbox"/> | 1 id            | int(11)      |                 |            | No   | None    | AUTO_INCREMENT | Change  Drop  More |
| <input type="checkbox"/> | 2 group_id      | int(3)       |                 |            | No   | None    |                | Change  Drop  More |
| <input type="checkbox"/> | 3 name          | varchar(255) | utf8_unicode_ci |            | No   | None    |                | Change  Drop  More |
| <input type="checkbox"/> | 4 email_address | varchar(100) | utf8_unicode_ci |            | No   | None    |                | Change  Drop  More |
| <input type="checkbox"/> | 5 password      | varchar(40)  | utf8_unicode_ci |            | No   | None    |                | Change  Drop  More |
| <input type="checkbox"/> | 6 status        | tinyint(1)   |                 |            | No   | 0       |                | Change  Drop  More |

Figure 3.1: User Information Table

### Table Number 2: Buyer\_detail

**Purpose:** Buyer Information

**Description:** This table will contain the buyer details information. This information will help to create new buyer and delete old buyer.



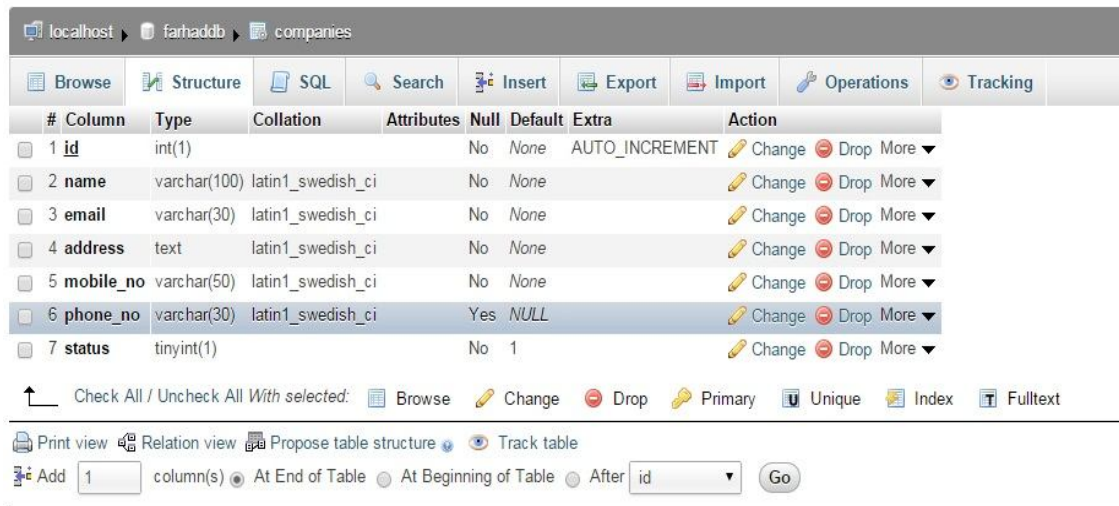
| #                        | Column     | Type         | Collation         | Attributes | Null | Default | Extra          | Action             |
|--------------------------|------------|--------------|-------------------|------------|------|---------|----------------|--------------------|
| <input type="checkbox"/> | 1 id       | int(11)      |                   |            | No   | None    | AUTO_INCREMENT | Change  Drop  More |
| <input type="checkbox"/> | 2 code     | varchar(40)  | latin1_swedish_ci | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 3 name     | varchar(200) | latin1_swedish_ci | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 4 address  | text         | latin1_swedish_ci | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 5 state    | varchar(200) | latin1_swedish_ci | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 6 city     | varchar(200) | latin1_swedish_ci | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 7 zip      | int(200)     |                   | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 8 country  | varchar(200) | latin1_swedish_ci | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 9 phone no | int(11)      |                   | Yes        | NULL |         |                | Change  Drop  More |
| <input type="checkbox"/> | 10 email   | varchar(100) | latin1_swedish_ci | Yes        | NULL |         |                | Change  Drop  More |

Figure 3.2: Buyer Information Table

### Table Number 3: company

**Purpose:** Company Information

**Description:** This table will contain company information.



The screenshot displays the MySQL Enterprise Workbench interface for the 'company' table. The table structure is as follows:

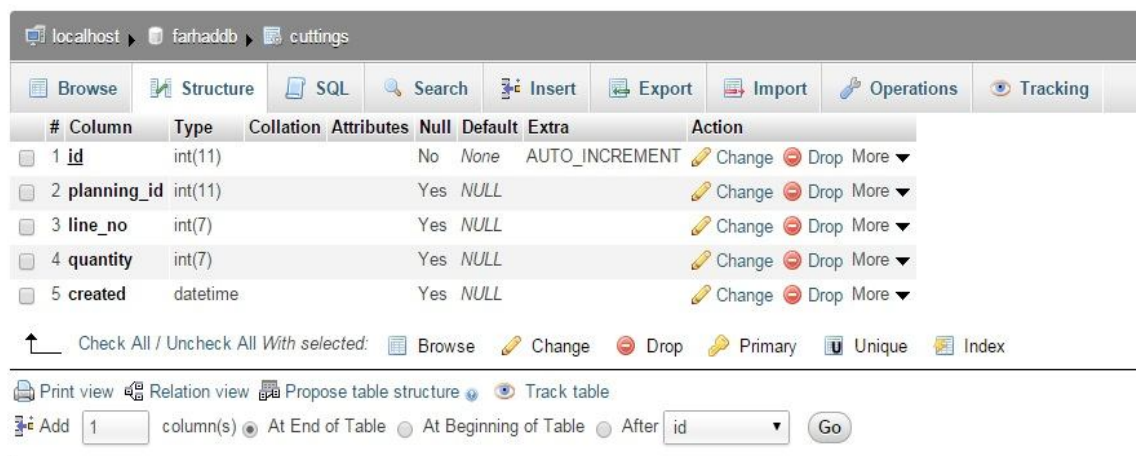
| # | Column    | Type         | Collation         | Attributes | Null | Default | Extra          | Action           |
|---|-----------|--------------|-------------------|------------|------|---------|----------------|------------------|
| 1 | id        | int(1)       |                   |            | No   | None    | AUTO_INCREMENT | Change Drop More |
| 2 | name      | varchar(100) | latin1_swedish_ci |            | No   | None    |                | Change Drop More |
| 3 | email     | varchar(30)  | latin1_swedish_ci |            | No   | None    |                | Change Drop More |
| 4 | address   | text         | latin1_swedish_ci |            | No   | None    |                | Change Drop More |
| 5 | mobile_no | varchar(50)  | latin1_swedish_ci |            | No   | None    |                | Change Drop More |
| 6 | phone_no  | varchar(30)  | latin1_swedish_ci |            | Yes  | NULL    |                | Change Drop More |
| 7 | status    | tinyint(1)   |                   |            | No   | 1       |                | Change Drop More |

Figure 3.3: Company Information Table

### Table Number 4: cuttings

**Purpose:** Cutting process information

**Description:** This table will contain cutting information of product status. This information will help to manage and monitor cutting process.



The screenshot displays the MySQL Enterprise Workbench interface for the 'cuttings' table. The table structure is as follows:

| # | Column      | Type     | Collation | Attributes | Null | Default | Extra          | Action           |
|---|-------------|----------|-----------|------------|------|---------|----------------|------------------|
| 1 | id          | int(11)  |           |            | No   | None    | AUTO_INCREMENT | Change Drop More |
| 2 | planning_id | int(11)  |           |            | Yes  | NULL    |                | Change Drop More |
| 3 | line_no     | int(7)   |           |            | Yes  | NULL    |                | Change Drop More |
| 4 | quantity    | int(7)   |           |            | Yes  | NULL    |                | Change Drop More |
| 5 | created     | datetime |           |            | Yes  | NULL    |                | Change Drop More |

Figure 3.4: Cutting process information Table

### Table Number 5: embroideries

**Purpose:** embroidery process information

**Description:** This table will contain embroidery information.

| #                        | Column           | Type     | Collation | Attributes | Null | Default | Extra          | Action               |
|--------------------------|------------------|----------|-----------|------------|------|---------|----------------|----------------------|
| <input type="checkbox"/> | 1 id             | int(11)  |           |            | No   | None    | AUTO_INCREMENT | Change  Drop  More ▼ |
| <input type="checkbox"/> | 2 planning_id    | int(11)  |           |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 3 total_received | int(7)   |           |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 4 total_emb      | int(6)   |           |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 5 total_output   | int(8)   |           |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 6 created        | datetime |           |            | Yes  | NULL    |                | Change  Drop  More ▼ |

Figure 3.5: Embroidery process information Table

### Table Number 6: fabrics

**Purpose:** Information of fabrics

**Description:** This table will contain fabrics information

| #                        | Column          | Type        | Collation         | Attributes | Null | Default | Extra          | Action               |
|--------------------------|-----------------|-------------|-------------------|------------|------|---------|----------------|----------------------|
| <input type="checkbox"/> | 1 id            | int(11)     |                   |            | No   | None    | AUTO_INCREMENT | Change  Drop  More ▼ |
| <input type="checkbox"/> | 2 planning_id   | int(11)     |                   |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 3 unit          | varchar(50) | latin1_swedish_ci |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 4 total_receive | int(7)      |                   |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 5 total_return  | int(7)      |                   |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 6 net_receive   | int(7)      |                   |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 7 total_issue   | int(7)      |                   |            | Yes  | NULL    |                | Change  Drop  More ▼ |
| <input type="checkbox"/> | 8 created       | datetime    |                   |            | Yes  | NULL    |                | Change  Drop  More ▼ |

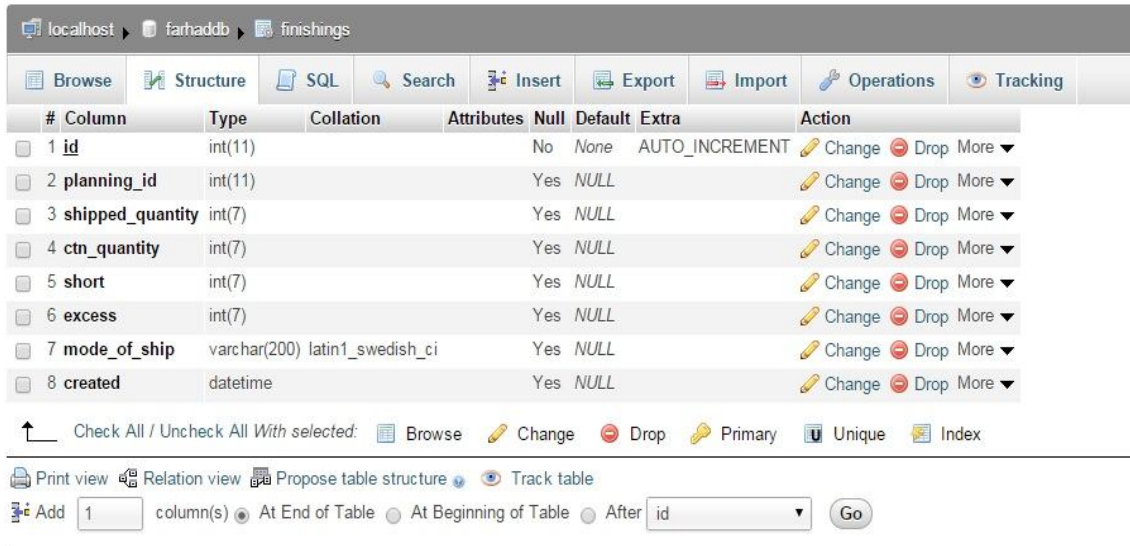
Figure 3.6: Fabrics information Table



### Table Number 7: finishings

**Purpose:** Information of finishing department

**Description:** This table will contain finishing information.



The screenshot shows the MySQL table structure for the 'finishings' table. The table has 8 columns: id, planning\_id, shipped\_quantity, ctn\_quantity, short, excess, mode\_of\_ship, and created. The 'id' column is the primary key and is auto-incrementing. All other columns are nullable.

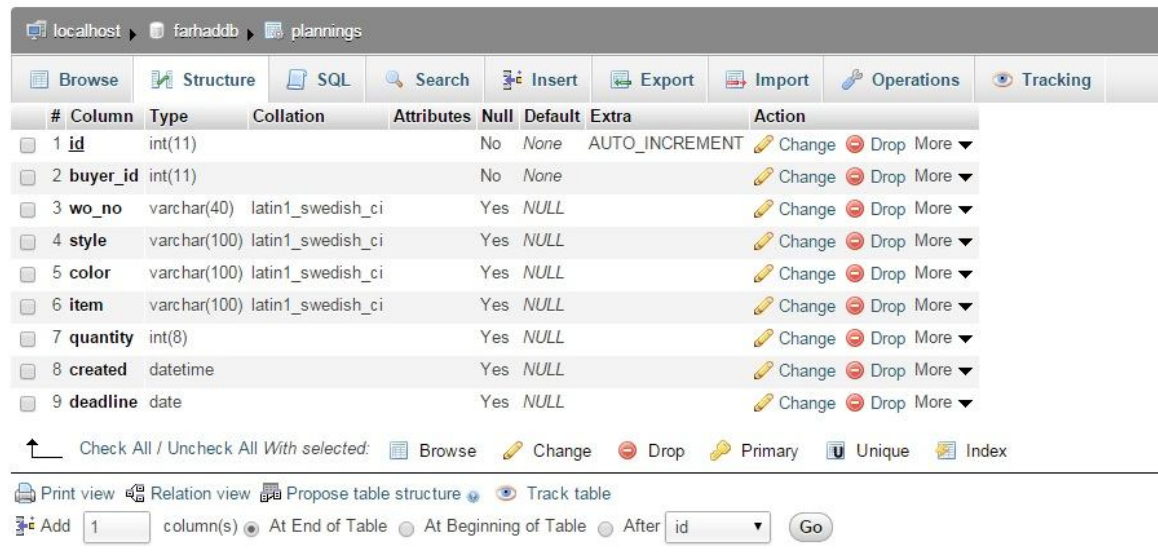
| # | Column           | Type         | Collation         | Attributes | Null | Default | Extra          | Action           |
|---|------------------|--------------|-------------------|------------|------|---------|----------------|------------------|
| 1 | id               | int(11)      |                   |            | No   | None    | AUTO_INCREMENT | Change Drop More |
| 2 | planning_id      | int(11)      |                   |            | Yes  | NULL    |                | Change Drop More |
| 3 | shipped_quantity | int(7)       |                   |            | Yes  | NULL    |                | Change Drop More |
| 4 | ctn_quantity     | int(7)       |                   |            | Yes  | NULL    |                | Change Drop More |
| 5 | short            | int(7)       |                   |            | Yes  | NULL    |                | Change Drop More |
| 6 | excess           | int(7)       |                   |            | Yes  | NULL    |                | Change Drop More |
| 7 | mode_of_ship     | varchar(200) | latin1_swedish_ci |            | Yes  | NULL    |                | Change Drop More |
| 8 | created          | datetime     |                   |            | Yes  | NULL    |                | Change Drop More |

Figure 3.7: *Finishing information Table*

### Table Number 8: plannings

**Purpose:** Information of planning department work

**Description:** This table will contain workorder information.



The screenshot shows the MySQL table structure for the 'plannings' table. The table has 9 columns: id, buyer\_id, wo\_no, style, color, item, quantity, created, and deadline. The 'id' column is the primary key. 'wo\_no', 'style', 'color', and 'item' are nullable and have a collation of latin1\_swedish\_ci.

| # | Column   | Type         | Collation         | Attributes | Null | Default | Extra          | Action           |
|---|----------|--------------|-------------------|------------|------|---------|----------------|------------------|
| 1 | id       | int(11)      |                   |            | No   | None    | AUTO_INCREMENT | Change Drop More |
| 2 | buyer_id | int(11)      |                   |            | No   | None    |                | Change Drop More |
| 3 | wo_no    | varchar(40)  | latin1_swedish_ci |            | Yes  | NULL    |                | Change Drop More |
| 4 | style    | varchar(100) | latin1_swedish_ci |            | Yes  | NULL    |                | Change Drop More |
| 5 | color    | varchar(100) | latin1_swedish_ci |            | Yes  | NULL    |                | Change Drop More |
| 6 | item     | varchar(100) | latin1_swedish_ci |            | Yes  | NULL    |                | Change Drop More |
| 7 | quantity | int(8)       |                   |            | Yes  | NULL    |                | Change Drop More |
| 8 | created  | datetime     |                   |            | Yes  | NULL    |                | Change Drop More |
| 9 | deadline | date         |                   |            | Yes  | NULL    |                | Change Drop More |

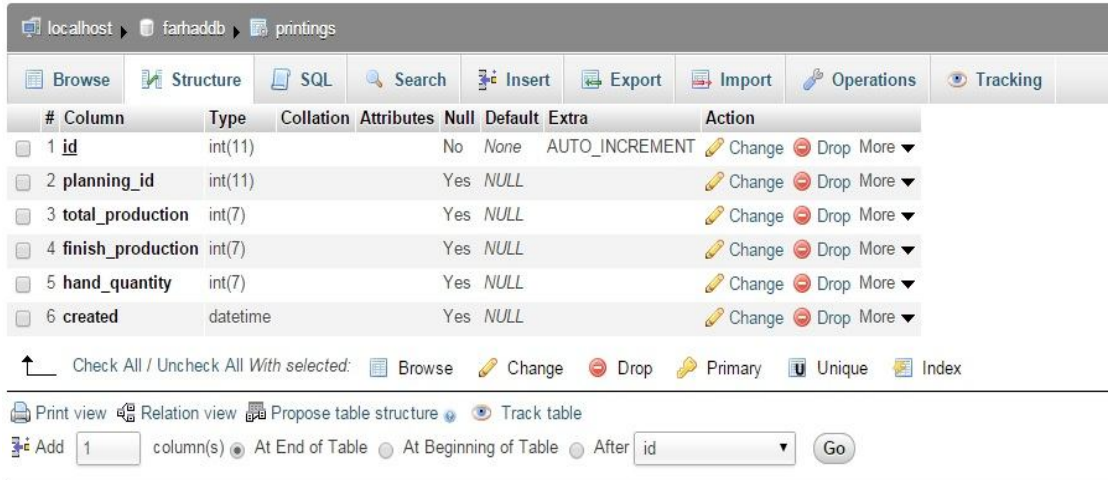
Figure 3.8: *Workorder information Table*



### Table Number 9: printings

**Purpose:** Information of printing department

**Description:** This table will contain printing information.



The screenshot shows the MySQL table structure for the 'printings' table in the 'farhaddb' database. The table has six columns: 'id', 'planning\_id', 'total\_production', 'finish\_production', 'hand\_quantity', and 'created'. The 'id' column is the primary key and is auto-incrementing. All other columns are integers or datetimes, and all are nullable.

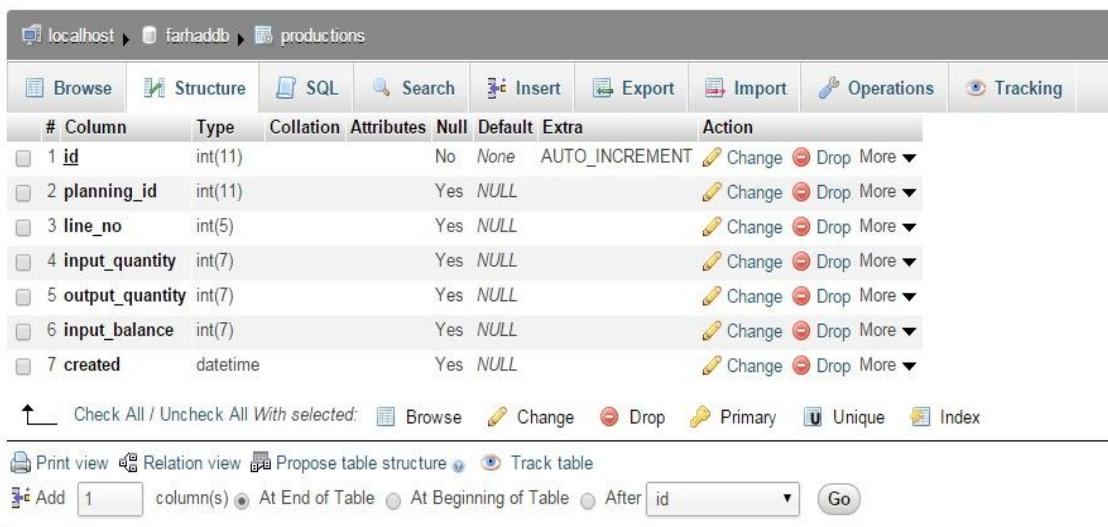
| #                        | Column              | Type     | Collation | Attributes | Null | Default | Extra          | Action             |
|--------------------------|---------------------|----------|-----------|------------|------|---------|----------------|--------------------|
| <input type="checkbox"/> | 1 id                | int(11)  |           |            | No   | None    | AUTO_INCREMENT | Change  Drop  More |
| <input type="checkbox"/> | 2 planning_id       | int(11)  |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 3 total_production  | int(7)   |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 4 finish_production | int(7)   |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 5 hand_quantity     | int(7)   |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 6 created           | datetime |           |            | Yes  | NULL    |                | Change  Drop  More |

Figure 3.9: Printing information Table

### Table Number 10: productions

**Purpose:** Information of production department

**Description:** This table will contain production information.



The screenshot shows the MySQL table structure for the 'productions' table in the 'farhaddb' database. The table has seven columns: 'id', 'planning\_id', 'line\_no', 'input\_quantity', 'output\_quantity', 'input\_balance', and 'created'. The 'id' column is the primary key. All other columns are integers or datetimes, and all are nullable.

| #                        | Column            | Type     | Collation | Attributes | Null | Default | Extra          | Action             |
|--------------------------|-------------------|----------|-----------|------------|------|---------|----------------|--------------------|
| <input type="checkbox"/> | 1 id              | int(11)  |           |            | No   | None    | AUTO_INCREMENT | Change  Drop  More |
| <input type="checkbox"/> | 2 planning_id     | int(11)  |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 3 line_no         | int(5)   |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 4 input_quantity  | int(7)   |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 5 output_quantity | int(7)   |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 6 input_balance   | int(7)   |           |            | Yes  | NULL    |                | Change  Drop  More |
| <input type="checkbox"/> | 7 created         | datetime |           |            | Yes  | NULL    |                | Change  Drop  More |

Figure 3.10: Production information Table

# Chapter: 4

## User Interface and Manual

In this chapter, we will briefly discuss about the software and hardware requirement for this system, also discuss the installation and use of the system. This system is built mostly with open source system, so that the cost requirement for this system is very minimal. The major part of the costing is on the hardware and partial software (for sending SMS).

### 4.1 Software Requirement for the system

For installing the system, we mainly required the following software, installed on the hosting system,

- i. XAMPP (Basis Package) version 1.7.3
- ii. PHP 5.3.1
- iii. MySQL 5.1.41
- iv. Cake php frame work

All of the above softwares are platform independent, so that we can use any operating system on our host computer.

### 4.2 Server Requirement

For running this system, the following hardware specification is required for the server;

- Processor: XEON Quad Core / 3.16 GHz / 1333 MHz
- Cache: L 2 12 MB or more
- RAM: 12 GB PC2 – 5300 CLS ECC DDR2 FBDIMM, should be scslable to 48 GB

- Host Bus Adapter 1 no. – Single port, 4 GBps
- HDD: Minimum 146 GB SAS, 15 K rpm SCSI HDD's
- HDD Bays: Minimum 6 or more
- Optical Drive: 16x DVD ROM or higher
- Adapter slots: Min 4 – P or more
- Ethernet: 2 Ethernet Ports, 10/100/1000 Mbps
- RAID Should be able to perform RAID 1 in current configuration
- Power Supply & Cooling Redundant
- OS: Windows Server / Linux

### **4.3 Work station requirement**

For running this system, any midlevel workstation can handle this system. The system specifications are;

- Processor: Pentium 4 (2.8 GHz), similar or above.
- Cache: L2 2 MB or more
- RAM: 1 GB
- HDD: Minimum 120 GB SATA
- Optical Drive: 16x DVD ROM or higher
- Adapter slots: Min 2
- Ethernet: 1 Ethernet Ports, 10/100/1000 Mbps
- Power Supply & Cooling Redundant
- OS: Windows XP ( minimum)
- A network connection.
- A printer (if user requires any printing)
- A web browser (Firefox, Chrome, internet explorer)

## 4.4 Installation

For installing this system on the server, it requires a fresh operating system installed server with internet / intranet connection. Then, have to follow the following procedure. [20]

### 4.4.1 Installation of XAMPP Server

**Step 01:** Open the XAMPP website at  
<http://www.4shared.com/get/HdHmCLEu/xampp-win32-173.html> [22]

**Step 02:** Click on the FREE DOWNLOAD link.

**Step 03:** After download, Click on the file



**Step 04:** Select destination folder and Click on the 'Install'

**Step 05:** Type 'y'

**Step 06:** Type 'y'

**Step 07:** Type 'y'

**Step 08:** Type 'y'

**Step 09:** Type 'y'

**Step 10:** Press 'Enter'

**Step 11:** Type '1'

**Step 11:** Click on 'Start' next to Apache and 'Start' next to MySQL

**Step 12:** Type 'x' to exit

### 4.5.1 User login Page

**Description:** This is the login page. From here a user and admin can Log in the system.

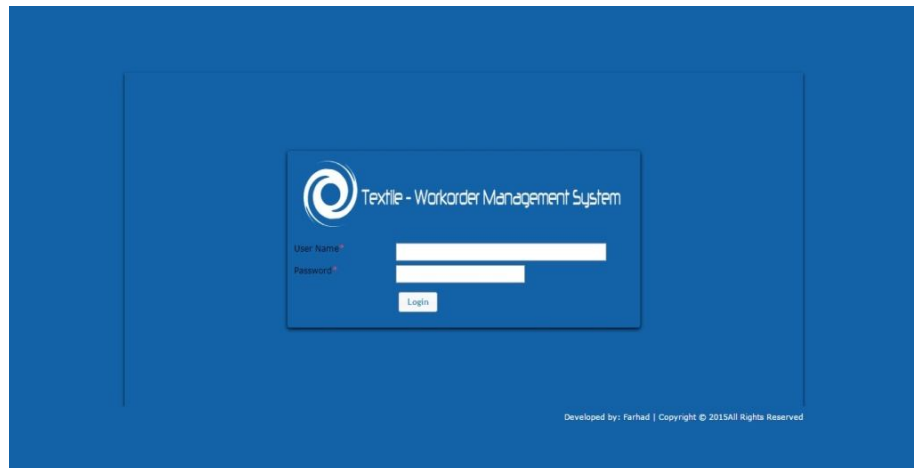


Figure 4.1: *login page*

### 4.5.2 Home page

**Description:** This is home page.

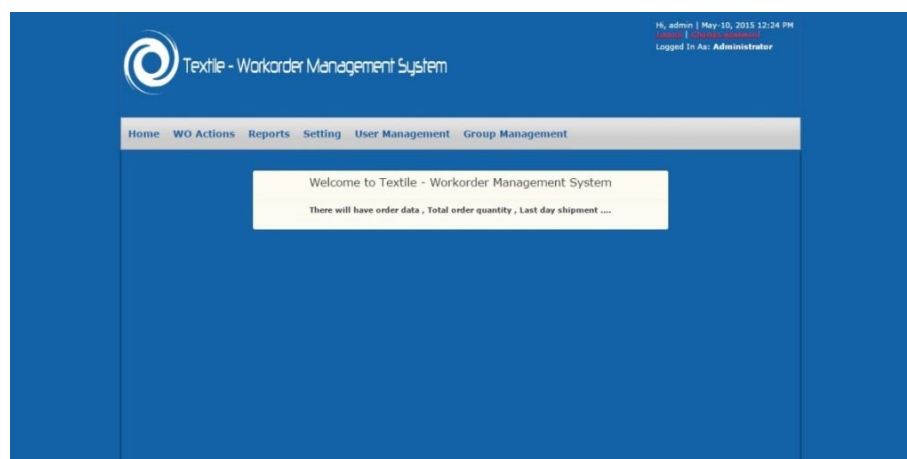


Figure 4.2: *Home page*

### 4.5.3 Add new buyer page

**Description:** Add new buyer page shown here.

Hi, admin | May-10, 2015 3:33 PM  
Logged In As: Administrator

Home WO Actions Reports Setting User Management Group Management

**Actions**  
List Buyers

**Add Buyer**

Name

State

City

Zip

Country

Phone No

Email

Figure 4.3: Add new buyer page is shown here

### 4.5.4 Add new cutting page

**Description:** Add new cutting page shown here.

Hi, | May-10, 2015 3:43 PM  
Logged In As: Cutting User

Home WO Actions Reports

**Actions**  
List Cuttings

**Add Cutting**

| Buyer Info | Wo No  | Style  | Color | Item | Quantity |
|------------|--------|--------|-------|------|----------|
| 12345      | M-2227 | 214332 | White | L    | 15000    |

Work Order Id

Line No

Quantity

Figure 4.4: Add new cutting page is shown here

## 4.5.5 Add new embroidery page

**Description:** Add new embroidery page shown here.

Hi, | May-10, 2015 3:45 PM  
Logged In As: Embroidery User

Home WO Actions Reports

**Actions**  
List Embroideries

**Add Embroidery**

| Work Order Id  | Buyer Info | Wo No  | Style  | Color | Item | Quantity |
|----------------|------------|--------|--------|-------|------|----------|
| 14             | Mexx       | M-2227 | 234332 | White | L    | 15000    |
| Total Received |            |        |        |       |      |          |
| Total Emb      |            |        |        |       |      |          |
| Total Output   |            |        |        |       |      |          |

Submit

Figure 4.5: Add new embroidery page is shown here

## 4.5.6 Add new fabric page

**Description:** Add new fabric page shown here.

Hi, | May-10, 2015 3:40 PM  
Logged In As: Fabric User

Home WO Actions Reports

**Actions**  
List Fabrics

**Add Fabric**

| Work Order Id | Buyer Info | Wo No  | Style  | Color | Item | Quantity |
|---------------|------------|--------|--------|-------|------|----------|
| 14            | Mexx       | M-2227 | 234332 | White | L    | 15000    |
| Unit          |            |        |        |       |      |          |
| Total Receive |            |        |        |       |      |          |
| Total Return  |            |        |        |       |      |          |
| Net Receive   |            |        |        |       |      |          |
| Total Issue   |            |        |        |       |      |          |

Submit

Figure 4.6: Add new fabric page is shown here

## 4.5.7 Add new finishing page

**Description:** Add new finishing page shown here.

The screenshot shows the 'Add Finishing' page in the Textile - Workorder Management System. The page has a blue header with the system logo and name. The main content area is white and contains a form with the following fields:

| Buyer Info | Wo No  | Style  | Color | Item | Quantity |
|------------|--------|--------|-------|------|----------|
| Mexx       | M-2227 | 234332 | White | L    | 15000    |

Below the table, there are five input fields:

- Shipped Quantity
- Ctn Quantity
- Short
- Excess
- Mode Of Ship

A 'Submit' button is located at the bottom of the form. The page also has a navigation bar with 'Home', 'WO Actions', and 'Reports' links. The user is logged in as 'Finishing User'.

Figure 4.7: Add new finishing page is shown here

## 4.5.8 Add new production page

**Description:** Add new production page shown here.

The screenshot shows the 'Add Production' page in the Textile - Workorder Management System. The page has a blue header with the system logo and name. The main content area is white and contains a form with the following fields:

| Buyer Info | Wo No  | Style  | Color | Item | Quantity |
|------------|--------|--------|-------|------|----------|
| Mexx       | M-2227 | 234332 | White | L    | 15000    |

Below the table, there are four input fields:

- Line No
- Input Quantity
- Output Quantity
- Input Balance

A 'Submit' button is located at the bottom of the form. The page also has a navigation bar with 'Home', 'WO Actions', and 'Reports' links. The user is logged in as 'Production User'.

Figure 4.8: Add new production page is shown here



## 4.5.9 Add new user page

**Description:** Add new user page shown here.

Hi, admin | May-10, 2015 3:34 PM  
Logged In As: Administrator

Textile - Workorder Management System

Home WO Actions Reports Setting User Management Group Management

**Actions**  
List Users  
List Groups

**Add User**  
Group\* Administrator  
Name  
Email Address\*  
Password\*  
 Status  
Submit

Figure 4.9: Add new user page is shown here

## 4.5.10 Add new workorder page

**Description:** Add new workorder page shown here.

Hi, admin | May-10, 2015 3:36 PM  
Logged In As: Administrator

Textile - Workorder Management System

Home WO Actions Reports Setting User Management Group Management

**Actions**  
List Plannings  
List Buyers  
New Buyer

**Add Work Order**  
Buyer\* Puma  
Wo No  
Style  
Color  
Item  
Quantity  
Deadline May 10 2015  
Submit

Figure 4.10: Add new workorder page is shown here

#### 4.5.11 View buyer info page

**Description:** View buyer info page shown here.

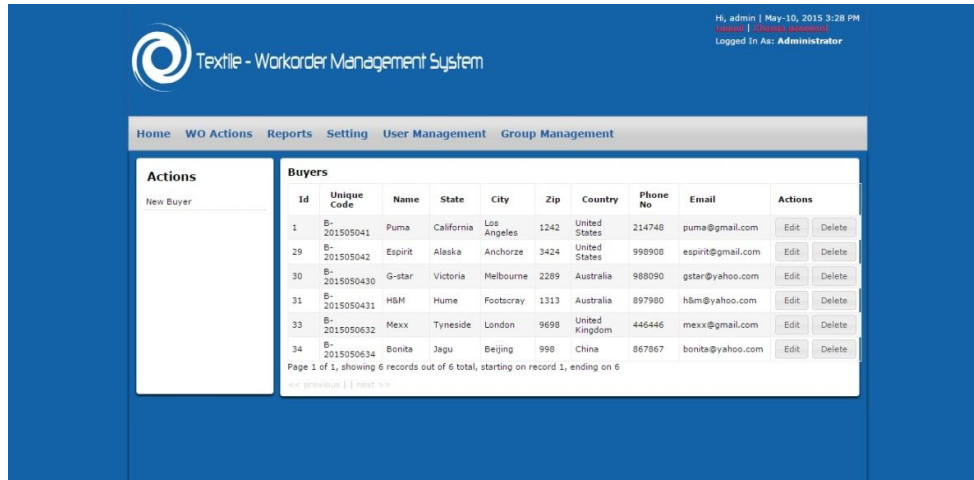


Figure 4.11: View buyer info page is shown here

#### 4.5.12 View buyer ratings page

**Description:** View buyer ratings page shown here.

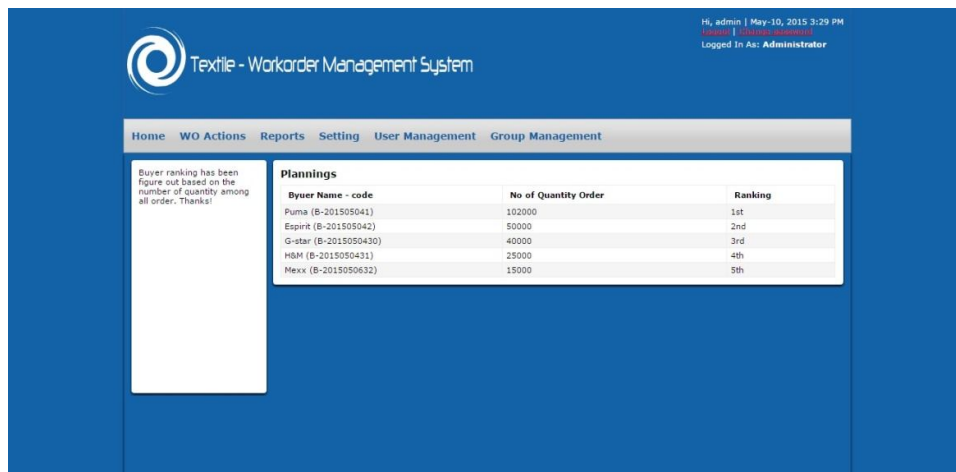


Figure 4.12: View buyer ratings page is shown here

### 4.5.13 View company info page

**Description:** View company info page shown here.

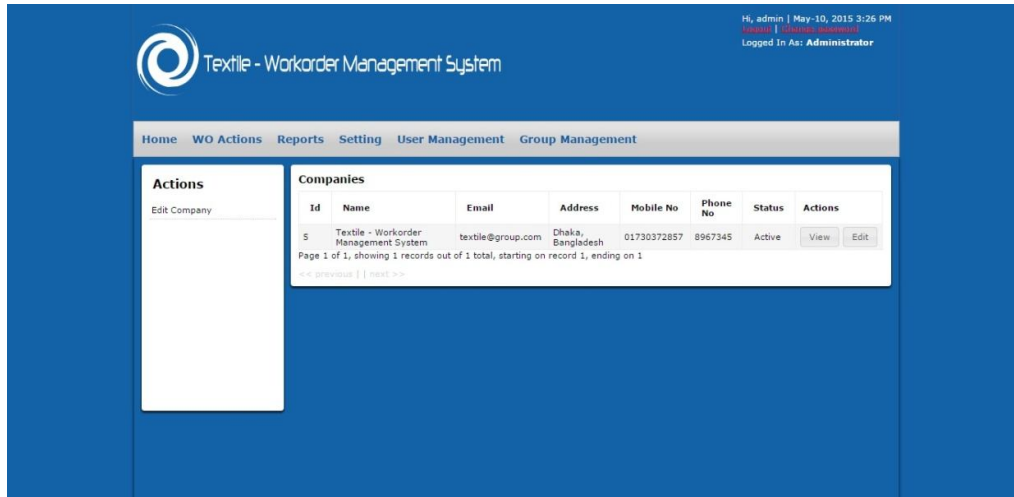


Figure 4.13: View company info page is shown here

### 4.5.14 View user setting

**Description:** View user setting shown here.

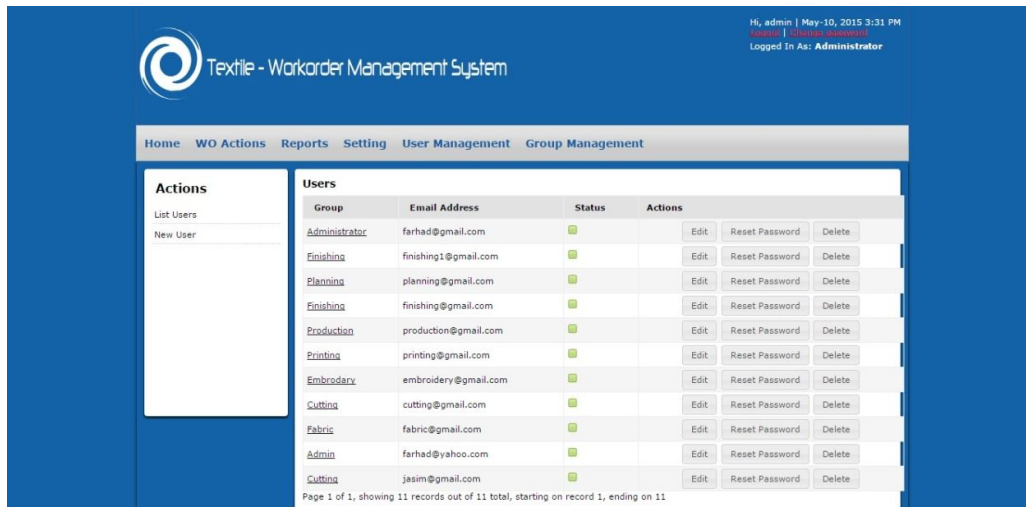


Figure 4.14: Add new buyer page is shown here

#### 4.5.15 View cutting page

**Description:** View Cutting page shown here.

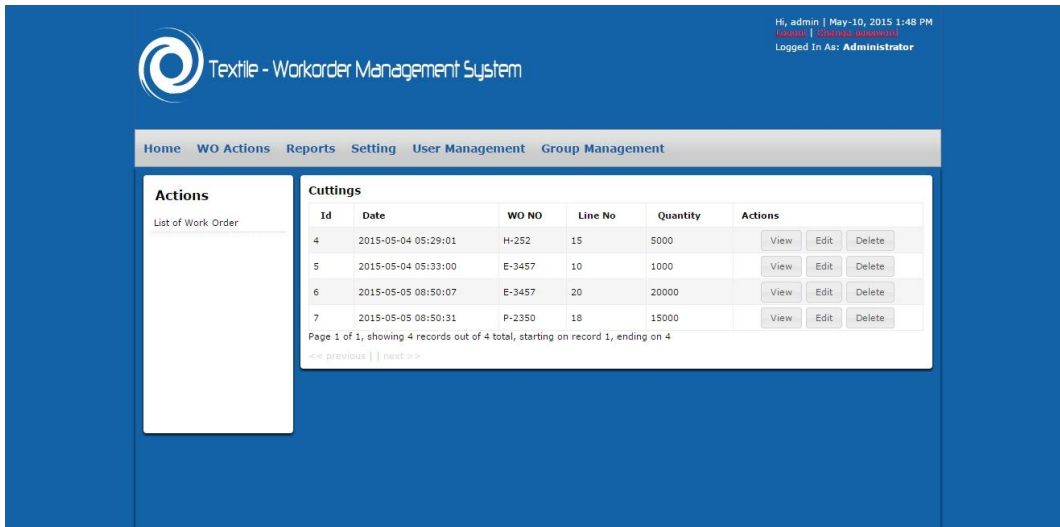


Figure 4.15: View cutting page is shown here

#### 4.5.16 View embroidery page

**Description:** View embroidery page shown here.

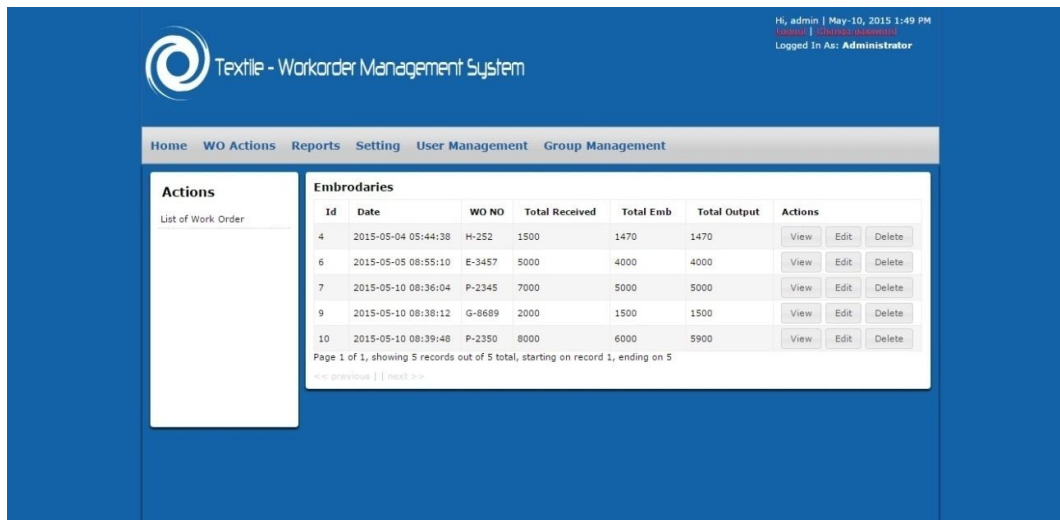


Figure 4.16: View embroidery page is shown here

#### 4.5.17 View fabrics page

**Description:** View fabrics page shown here.

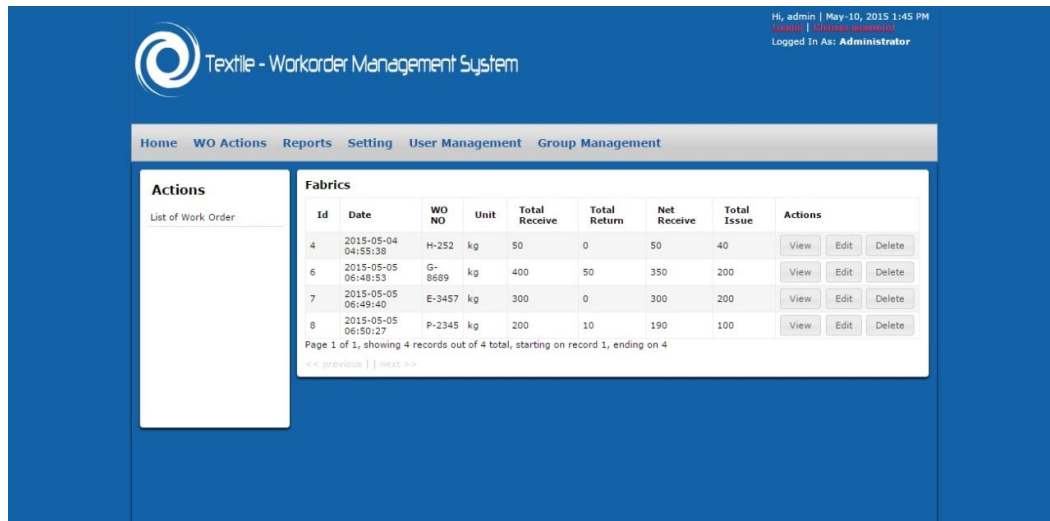


Figure 4.17: View fabrics page is shown here

#### 4.5.18 View Groups page

**Description:** View groups page shown here.

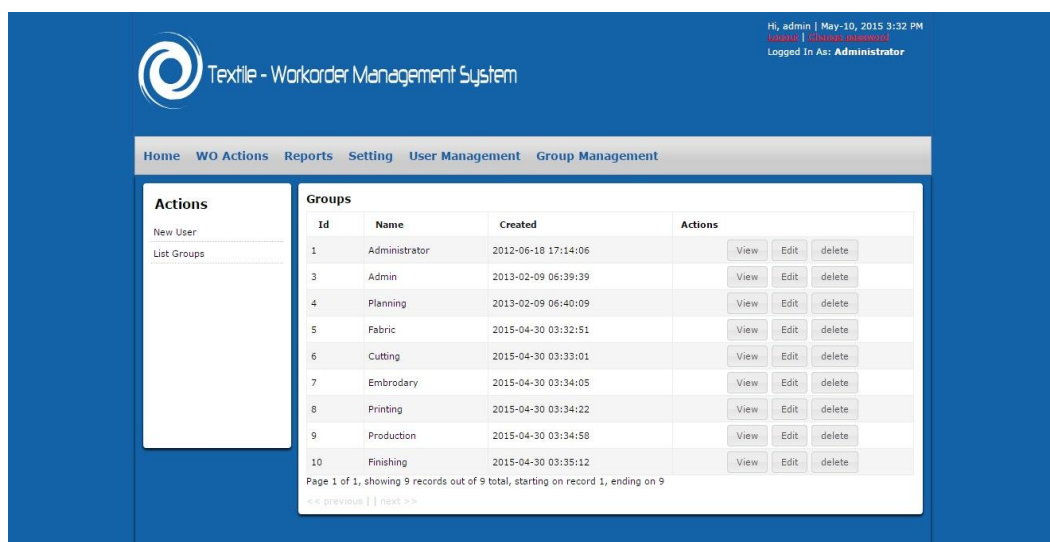
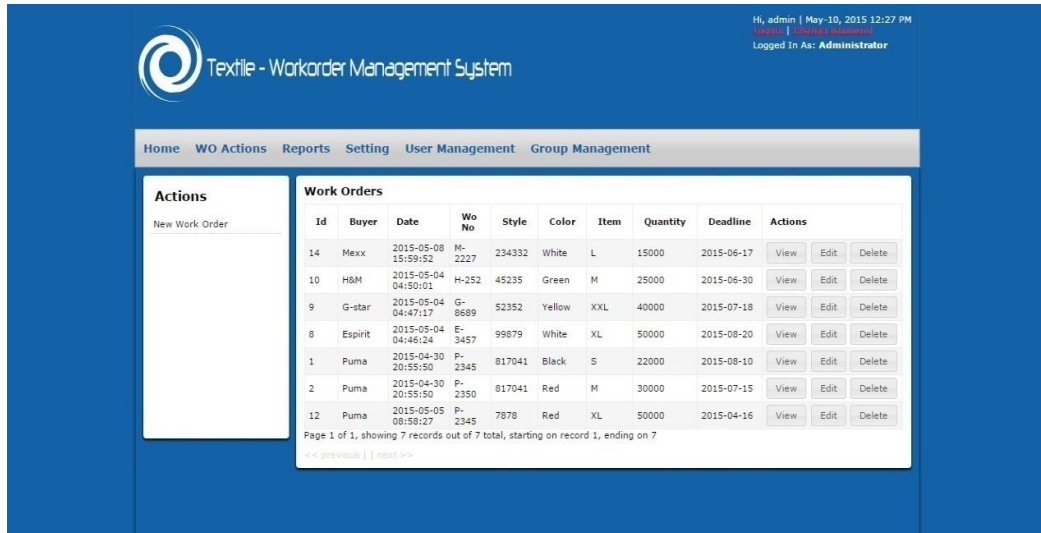


Figure 4.18: View groups page is shown here

## 4.5.19 View Workorder page

**Description:** View workorder page shown here.



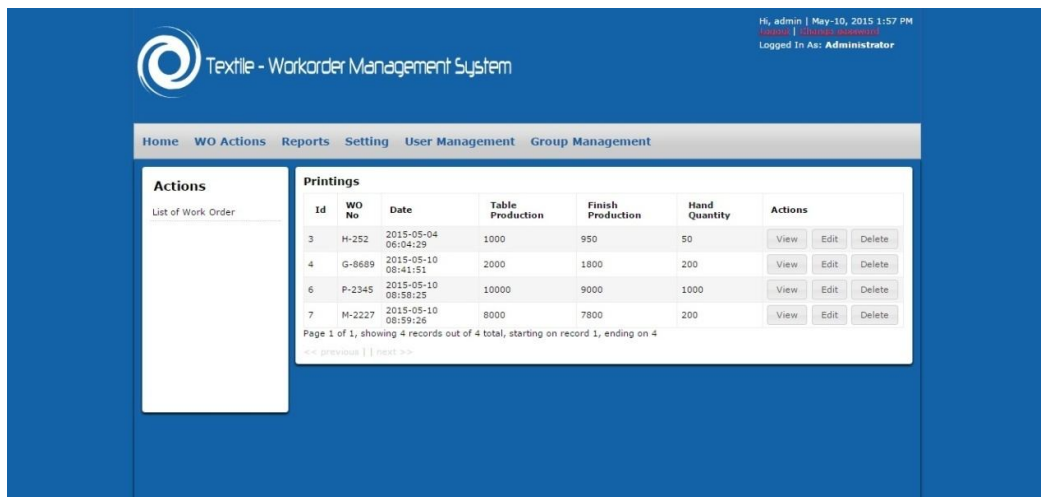
The screenshot displays the 'Textile - Workorder Management System' interface. The top navigation bar includes 'Home', 'WO Actions', 'Reports', 'Setting', 'User Management', and 'Group Management'. The main content area is titled 'Work Orders' and features a table with the following columns: Id, Buyer, Date, Wo No, Style, Color, Item, Quantity, Deadline, and Actions. The table contains 7 records. A sidebar on the left is labeled 'Actions' and contains a 'New Work Order' button. The bottom of the table shows pagination: 'Page 1 of 1, showing 7 records out of 7 total, starting on record 1, ending on 7'.

| Id | Buyer   | Date                | Wo No  | Style  | Color  | Item | Quantity | Deadline   | Actions          |
|----|---------|---------------------|--------|--------|--------|------|----------|------------|------------------|
| 14 | Mexx    | 2015-05-08 15:59:52 | M-2227 | 234332 | White  | L    | 15000    | 2015-06-17 | View Edit Delete |
| 10 | H&M     | 2015-05-04 04:50:01 | H-252  | 45235  | Green  | M    | 25000    | 2015-06-30 | View Edit Delete |
| 9  | G-star  | 2015-05-04 04:47:17 | G-8689 | 52352  | Yellow | XXL  | 40000    | 2015-07-18 | View Edit Delete |
| 8  | Espirit | 2015-05-04 04:46:24 | E-3457 | 99679  | White  | XL   | 50000    | 2015-08-20 | View Edit Delete |
| 1  | Puma    | 2015-04-30 20:55:50 | P-2345 | 817041 | Black  | S    | 22000    | 2015-08-10 | View Edit Delete |
| 2  | Puma    | 2015-04-30 20:55:50 | P-2350 | 817041 | Red    | M    | 30000    | 2015-07-15 | View Edit Delete |
| 12 | Puma    | 2015-05-05 08:59:27 | P-2345 | 7878   | Red    | XL   | 50000    | 2015-04-16 | View Edit Delete |

Figure 4.19: View workorder page is shown here

## 4.5.20 View Printing page

**Description:** View printing page shown here.



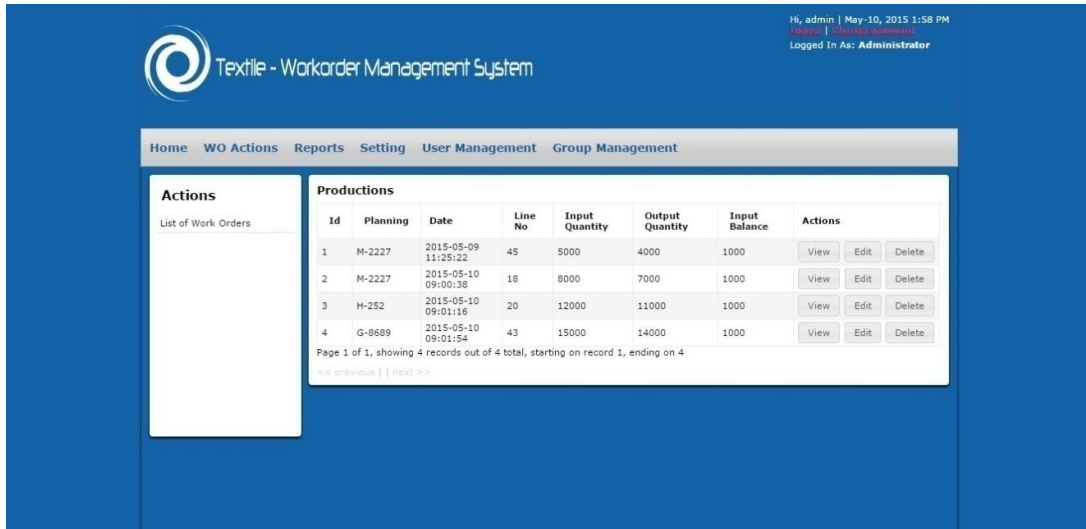
The screenshot displays the 'Textile - Workorder Management System' interface. The top navigation bar includes 'Home', 'WO Actions', 'Reports', 'Setting', 'User Management', and 'Group Management'. The main content area is titled 'Printings' and features a table with the following columns: Id, WO No, Date, Table Production, Finish Production, Hand Quantity, and Actions. The table contains 4 records. A sidebar on the left is labeled 'Actions' and contains a 'List of Work Order' button. The bottom of the table shows pagination: 'Page 1 of 1, showing 4 records out of 4 total, starting on record 1, ending on 4'.

| Id | WO No  | Date                | Table Production | Finish Production | Hand Quantity | Actions          |
|----|--------|---------------------|------------------|-------------------|---------------|------------------|
| 3  | H-252  | 2015-05-04 06:04:29 | 1000             | 950               | 50            | View Edit Delete |
| 4  | G-8689 | 2015-05-10 08:41:51 | 2000             | 1800              | 200           | View Edit Delete |
| 6  | P-2345 | 2015-05-10 08:58:25 | 10000            | 9000              | 1000          | View Edit Delete |
| 7  | M-2227 | 2015-05-10 08:59:26 | 8000             | 7800              | 200           | View Edit Delete |

Figure 4.20: View printing page is shown here

## 4.5.21 View Production page

**Description:** View Production page shown here.



The screenshot shows the 'Textile - Workorder Management System' interface. The top navigation bar includes 'Home', 'WO Actions', 'Reports', 'Setting', 'User Management', and 'Group Management'. The main content area is divided into two sections: 'Actions' (with a sub-section 'List of Work Orders') and 'Productions'. The 'Productions' section contains a table with the following data:

| Id | Planning | Date                | Line No | Input Quantity | Output Quantity | Input Balance | Actions          |
|----|----------|---------------------|---------|----------------|-----------------|---------------|------------------|
| 1  | M-2227   | 2015-05-09 11:25:22 | 45      | 5000           | 4000            | 1000          | View Edit Delete |
| 2  | M-2227   | 2015-05-10 09:00:38 | 18      | 8000           | 7000            | 1000          | View Edit Delete |
| 3  | H-252    | 2015-05-10 09:01:16 | 20      | 12000          | 11000           | 1000          | View Edit Delete |
| 4  | G-8689   | 2015-05-10 09:01:54 | 43      | 15000          | 14000           | 1000          | View Edit Delete |

Page 1 of 1, showing 4 records out of 4 total, starting on record 1, ending on 4  
<< previous | next >>

Figure 4.21: View production page is shown here

# Chapter: 5

## Conclusion and Future Work

Bangladesh is a developing country. Many developing countries are now using information communication technology for different sectors. If we want to change our position in the world we must have to develop our information technology. Through the developed project anyone can visualize the effectiveness and efficiency in the field. It is very helpful for computerization or doing automation of these types of management system. This automated system gives very helpful services for textile factories in our country. In this project, I have developed an automated textile workorder management system. Using this system, Admin can go through the entire system and also generating various types of report. On the other hand, Admin can modify, add and delete the workorder & shipment schedule to and from the system. Also can see the deliverables are on order or not. This system will be one of the steps forward to improve the textile sector of Bangladesh.

The future plan of this project is to improved design, implementation and documentation in such a way that beside workorder, any other work process monitoring service can easily customize and use this project.

I will develop the site more dynamically. In future I will add some modules for better improvement of the project, such as; the system wills automatically giving the recommendation for an event based on some logical criteria for a particular project. Also it will give some recommendation for merging or efficient resource planning. And finally a more secured and user friendly version, for the front end users. I will implement this system in textile industry.



## References

- [1] [http://en.wikipedia.org/wiki/Automated\\_business\\_process](http://en.wikipedia.org/wiki/Automated_business_process) [Last access date: May 10, 2015]
- [2] [http://en.wikipedia.org/wiki/Business\\_process\\_automation](http://en.wikipedia.org/wiki/Business_process_automation) [Last access date: May 10, 2015]
- [3] [http://en.wikipedia.org/wiki/Use\\_case](http://en.wikipedia.org/wiki/Use_case) [Last access date: May 10, 2015]
- [4] <http://whatis.techtarget.com/definition/use-case-diagram> [Last access date: May 10, 2015]
- [5] <http://en.wikipedia.org/wiki/HTML> [Last access date: May 10, 2015]
- [6] <http://en.wikipedia.org/wiki/PHP> [Last access date: May 10, 2015]
- [7] <http://en.wikipedia.org/wiki/MySQL> [Last access date: May 10, 2015]
- [8] [http://en.wikipedia.org/wiki/Database\\_design](http://en.wikipedia.org/wiki/Database_design) [Last access date: May10, 2015]
- [9] <http://en.wikipedia.org/wiki/CakePHP> [Last access date: May 10, 2015]
- [10] <http://www.w3schools.com/php/> [Last access date: May 10, 2015]
- [11] <http://www.w3schools.com/html/> [Last access date: May 10, 2015]
- [12] <http://www.w3schools.com/css/> [Last access date: May 10, 2015]

# Annexure

Annexure contain the sample code of Textile Workorder Management System for this project.

## Sample Code:

```
<?php
class BuyersController extends ApplicationController {

    var $name = 'Buyers';
    function index() {
        $this->Buyer->recursive = 0;
        $this->set('buyers', $this->paginate());
    }

    function view($id = null) {
        if (!$id) {
            $this->Session->setFlash(__('Invalid department', true));
            $this->redirect(array('action' => 'index'));
        }
        $this->set('buyers', $this->Buyer->read(null, $id));
    }

    function add() {
        $startNumber = 10001;
        if (!empty($this->data)) {

            $lastIntegerId = $this->Buyer->find('first',array('order'=>'id DESC'));
            $lastId = $lastIntegerId['Buyer']['id'] + 1;
            $this->data['Buyer']['code'] = 'B-'.date('Ymd').$lastId;

            $this->Buyer->create();

            if ($this->Buyer->save($this->data)) {
                $this->Session->setFlash(__('The buyers has been saved', true));
                $this->redirect(array('action' => 'index'));
            } else {
                $this->Session->setFlash(__('The buyers could not be saved. Please, try again.', true));
            }
        }
    }

    function edit($id = null) {
        if (!$id && empty($this->data)) {
            $this->Session->setFlash(__('Invalid buyers', true));
            $this->redirect(array('action' => 'index'));
        }
        if (!empty($this->data)) {
```

```

        if ($this->Buyer->save($this->data)) {
            $this->Session->setFlash(__('The buyers has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The buyers could not be saved. Please, try again.', true));
        }
    }
}
if (empty($this->data)) {
    $this->data = $this->Buyer->read(null, $id);
}
}

function delete($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid id for buyers', true));
        $this->redirect(array('action'=>'index'));
    }
    if ($this->Buyer->delete($id)) {
        $this->Session->setFlash(__('Department deleted', true));
        $this->redirect(array('action'=>'index'));
    }
    $this->Session->setFlash(__('Department was not deleted', true));
    $this->redirect(array('action' => 'index'));
}
}
}

```

```

<?php
class EmbrodariesController extends ApplicationController {

    var $name = 'Embrodaries';

    function index() {
        $this->Embrodary->recursive = 0;
        $this->set('embrodaries', $this->paginate());
    }

    function embrodaryreport(){
        $this->Embrodary->recursive = 2;

        if(!empty($this->params['named']['planning_id'])){
            $buyerID = $this->params['named']['planning_id'];
        }

        $conditions = array();

        if(!empty($this->params['named']['planning_id'])){
            $condition = array('Embrodary.planning_id' => $buyerID);
        }
    }
}

```

```

        if(!empty($condition)){
            $this->paginate = array( 'limit'=>20,'conditions'=>$condition,
'order'=>'Embrodary.planning_id ASC');
            $this->set('plannings', $this->paginate());
        }else{
            $this->paginate = array( 'limit'=>20, 'order'=>'Embrodary.planning_id ASC');
            $this->set('plannings', $this->paginate());
        }

        $planning_list = $this->Embrodary->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
        $this->set('planning_list', $planning_list);
    }

function view($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid embrodary', true));
        $this->redirect(array('action' => 'index'));
    }
    $this->set('embrodary', $this->Embrodary->read(null, $id));
}

function add() {
    if (!empty($this->data)) {
        $this->Embrodary->create();
        if ($this->Embrodary->save($this->data)) {
            $this->Session->setFlash(__('The embrodary has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The embrodary could not be saved. Please, try again.', true));
        }
    }
}

    $planning_list = $this->Embrodary->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
//    print_r($planning_list);
    $this->set('planning_list', $planning_list);
    $workOrderInfo = $this->Embrodary->Planning->findById($this->params['named']['id']);
    $this->set('workOrderInfo', $workOrderInfo);
}

function edit($id = null) {
    if (!$id && empty($this->data)) {
        $this->Session->setFlash(__('Invalid embrodary', true));
        $this->redirect(array('action' => 'index'));
    }
    if (!empty($this->data)) {
        if ($this->Embrodary->save($this->data)) {
            $this->Session->setFlash(__('The embrodary has been saved', true));
            $this->redirect(array('action' => 'index'));

```

```

        } else {
            $this->Session->setFlash(__('The embroidery could not be saved. Please, try again.', true));
        }
    }
    if (empty($this->data)) {
        $this->data = $this->Embroidary->read(null, $id);
    }
    $planning_list = $this->Embroidary->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
    $this->set('planning_list', $planning_list);
}

function delete($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid id for embroidery', true));
        $this->redirect(array('action'=>'index'));
    }
    if ($this->Embroidary->delete($id)) {
        $this->Session->setFlash(__('Embroidary deleted', true));
        $this->redirect(array('action'=>'index'));
    }
    $this->Session->setFlash(__('Embroidary was not deleted', true));
    $this->redirect(array('action' => 'index'));
}
}
}

```

```

<?php
class FabricsController extends ApplicationController {

```

```

    var $name = 'Fabrics';

```

```

function index() {
    $this->Fabric->recursive = 1;

    $this->set('fabrics', $this->paginate());
}

```

```

function fabricreport(){

    $this->Fabric->recursive = 2;

    if(!empty($this->params['named']['planning_id'])){
        $buyerID = $this->params['named']['planning_id'];
    }

    $conditions = array();

    if(!empty($this->params['named']['planning_id'])){

```

```

        $condition = array('Fabric.planning_id' => $buyerID);
    }

    if(!empty($condition)){
        $this->paginate = array( 'limit'=>20,'conditions'=>$condition, 'order'=>'Fabric.planning_id
ASC');
        $this->set('plannings', $this->paginate());
    }else{
        $this->paginate = array( 'limit'=>20, 'order'=>'Fabric.planning_id ASC');
        $this->set('plannings', $this->paginate());
    }

    $planning_list = $this->Fabric->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
    $this->set('planning_list', $planning_list);

}

function view($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid fabric', true));
        $this->redirect(array('action' => 'index'));
    }
    $this->set('fabric', $this->Fabric->read(null, $id));
}

function add() {
//     echo $this->params['named']['id'];
    if (!empty($this->data)) {
        $this->Fabric->create();
        if ($this->Fabric->save($this->data)) {
            $this->Session->setFlash(__('The fabric has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The fabric could not be saved. Please, try again.', true));
        }
    }
}

    $planning_list = $this->Fabric->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
//     print_r($planning_list);
    $this->set('planning_list', $planning_list);
    $workOrderInfo = $this->Fabric->Planning->findById($this->params['named']['id']);
    $this->set('workOrderInfo', $workOrderInfo);
}

function edit($id = null) {
    if (!$id && empty($this->data)) {
        $this->Session->setFlash(__('Invalid fabric', true));
        $this->redirect(array('action' => 'index'));
    }
}

```

```

if (!empty($this->data)) {
    if ($this->Fabric->save($this->data)) {
        $this->Session->setFlash(__('The fabric has been saved', true));
        $this->redirect(array('action' => 'index'));
    } else {
        $this->Session->setFlash(__('The fabric could not be saved. Please, try again.', true));
    }
}
if (empty($this->data)) {
    $this->data = $this->Fabric->read(null, $id);
}
$planning_list = $this->Fabric->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
$this->set('planning_list', $planning_list);
}

function delete($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid id for fabric', true));
        $this->redirect(array('action'=>'index'));
    }
    if ($this->Fabric->delete($id)) {
        $this->Session->setFlash(__('Fabric deleted', true));
        $this->redirect(array('action'=>'index'));
    }
    $this->Session->setFlash(__('Fabric was not deleted', true));
    $this->redirect(array('action' => 'index'));
}
}

```

```

<?php
class FinishingsController extends ApplicationController {

    var $name = 'Finishings';
    function index() {
        $this->Finishing->recursive = 0;
        $this->set('finishings', $this->paginate());
    }

    function view($id = null) {
        if (!$id) {
            $this->Session->setFlash(__('Invalid finishing', true));
            $this->redirect(array('action' => 'index'));
        }
        $this->set('finishing', $this->Finishing->read(null, $id));
    }

    function add() {

```

```

if (!empty($this->data)) {
    $this->Finishing->create();
    if ($this->Finishing->save($this->data)) {
        $this->Session->setFlash(__('The finishing has been saved', true));
        $this->redirect(array('action' => 'index'));
    } else {
        $this->Session->setFlash(__('The finishing could not be saved. Please, try again.', true));
    }
}

$planning_list = $this->Finishing->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
$this->set('planning_list', $planning_list);

$workOrderInfo = $this->Finishing->Planning->findById($this->params['named']['id']);
$this->set('workOrderInfo', $workOrderInfo);
}

function edit($id = null) {
    if (!$id && empty($this->data)) {
        $this->Session->setFlash(__('Invalid finishing', true));
        $this->redirect(array('action' => 'index'));
    }
    if (!empty($this->data)) {
        if ($this->Finishing->save($this->data)) {
            $this->Session->setFlash(__('The finishing has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The finishing could not be saved. Please, try again.', true));
        }
    }
    if (empty($this->data)) {
        $this->data = $this->Finishing->read(null, $id);
    }
    $planning_list = $this->Finishing->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
    $this->set('planning_list', $planning_list);
}

function delete($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid id for finishing', true));
        $this->redirect(array('action'=>'index'));
    }
    if ($this->Finishing->delete($id)) {
        $this->Session->setFlash(__('Finishing deleted', true));
        $this->redirect(array('action'=>'index'));
    }
    $this->Session->setFlash(__('Finishing was not deleted', true));
    $this->redirect(array('action' => 'index'));
}

```



```

function finishingreport(){
    $this->Finishing->recursive = 2;

    if(!empty($this->params['named']['planning_id'])){
        $buyerID = $this->params['named']['planning_id'];
    }

    $conditions = array();

    if(!empty($this->params['named']['planning_id'])){
        $condition = array('Finishing.planning_id' => $buyerID);
    }

    if(!empty($condition)){
        $this->paginate = array( 'limit'=>20,'conditions'=>$condition, 'order'=>'Finishing.planning_id
ASC');
    }else{
        $this->paginate = array( 'limit'=>20, 'order'=>'Finishing.planning_id ASC');
        $this->set('plannings', $this->paginate());
    }

    $planning_list = $this->Finishing->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
    $this->set('planning_list', $planning_list);
}
}

```

```

<?php
class PlanningsController extends AppController {

    var $name = 'Plannings';

    function index() {
        $this->Planning->recursive = 0;
        $this->paginate = array( 'limit'=>20,'order'=>'Planning.buyer_id DESC');
        $this->set('plannings', $this->paginate());
    }

    function planningreport(){
        $this->Planning->recursive = 0;

        if(!empty($this->params['named']['buyer_id'])){
            $buyerID = $this->params['named']['buyer_id'];
        }

        $conditions = array();

        if(!empty($this->params['named']['buyer_id'])){
            $condition = array('Planning.buyer_id' => $buyerID);

```

```

    }

    if(!empty($condition)){
        $this->paginate = array( 'limit'=>20,'conditions'=>$condition, 'order'=>'Planning.buyer_id
ASC');
        $this->set('plannings', $this->paginate());
    }else{
        $this->paginate = array( 'limit'=>20, 'order'=>'Planning.buyer_id ASC');
        $this->set('plannings', $this->paginate());
    }

    $buyer_list = $this->Planning->Buyer->find('list', array( 'fields' => array('Buyer.id',
'Buyer.name')));
    $this->set('buyer_list', $buyer_list);
}

function view($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid planning', true));
        $this->redirect(array('action' => 'index'));
    }
    $this->set('planning', $this->Planning->read(null, $id));
}

function add() {
    if (!empty($this->data)) {
        $this->Planning->create();
        if ($this->Planning->save($this->data)) {
            $this->Session->setFlash(__('The planning has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The planning could not be saved. Please, try again.', true));
        }
    }
    $buyer_list = $this->Planning->Buyer->find('list', array( 'fields' => array('Buyer.id',
'Buyer.name')));
    $this->set('buyer_list', $buyer_list);
}

function edit($id = null) {
    if (!$id && empty($this->data)) {
        $this->Session->setFlash(__('Invalid planning', true));
        $this->redirect(array('action' => 'index'));
    }
    if (!empty($this->data)) {
        if ($this->Planning->save($this->data)) {
            $this->Session->setFlash(__('The planning has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The planning could not be saved. Please, try again.', true));
        }
    }
}

```

```

if (empty($this->data)) {
    $this->data = $this->Planning->read(null, $id);
}

$buyer_list = $this->Planning->Buyer->find('list', array( 'fields' => array('Buyer.id',
'Buyer.name')));
$this->set('buyer_list', $buyer_list);

$this->set('buyerNmaeInfo', $this->Planning->read(null, $id));
}

function delete($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid id for planning', true));
        $this->redirect(array('action'=>'index'));
    }
    if ($this->Planning->delete($id)) {
        $this->Session->setFlash(__('Planning deleted', true));
        $this->redirect(array('action'=>'index'));
    }
    $this->Session->setFlash(__('Planning was not deleted', true));
    $this->redirect(array('action' => 'index'));
}

function buyerratting(){
    $this->Planning->recursive = 2;
    $this->paginate = array( 'limit'=>300,
'fields'=>array('Planning.buyer_id','SUM(Planning.quantity) AS countQuantity'),
'group'=>'Planning.buyer_id', 'order'=>'SUM(Planning.quantity) DESC');

    $this->set('plannings', $this->paginate());
}

/* function buyerratting(){
    $this->Planning->recursive = 2;
    $this->paginate = array( 'limit'=>300,
'fields'=>array('Planning.buyer_id','SUM(Planning.quantity) AS countQuantity'),
'group'=>'Planning.buyer_id', 'order'=>'Planning.buyer_id ASC');
    $this->set('plannings', $this->paginate());
}*/
}

```

```

<?php
class PrintingsController extends ApplicationController {

    var $name = 'Printings';
function index() {
    $this->Printing->recursive = 0;
    $this->set('printings', $this->paginate());
}

function printingreport() {
    $this->Printing->recursive = 2;

    if(!empty($this->params['named']['planning_id'])){
        $buyerID = $this->params['named']['planning_id'];
    }

    $conditions = array();

    if(!empty($this->params['named']['planning_id'])){
        $condition = array('Printing.planning_id' => $buyerID);
    }

    if(!empty($condition)){
        $this->paginate = array( 'limit'=>20,'conditions'=>$condition, 'order'=>'Printing.planning_id
ASC');
        $this->set('plannings', $this->paginate());
    }else{
        $this->paginate = array( 'limit'=>20, 'order'=>'Printing.planning_id ASC');
        $this->set('plannings', $this->paginate());
    }

    $planning_list = $this->Printing->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
    $this->set('planning_list', $planning_list);
}

function view($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid printing', true));
        $this->redirect(array('action' => 'index'));
    }
    $this->set('printing', $this->Printing->read(null, $id));
}

function add() {
    if (!empty($this->data)) {
        $this->Printing->create();
        if ($this->Printing->save($this->data)) {
            $this->Session->setFlash(__('The printing has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {

```

```

        $this->Session->setFlash(__('The printing could not be saved. Please, try again.', true));
    }
}

$planning_list = $this->Printing->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
$this->set('planning_list', $planning_list);

$workOrderInfo = $this->Printing->Planning->findById($this->params['named']['id']);
$this->set('workOrderInfo', $workOrderInfo);

}

function edit($id = null) {
    if (!$id && empty($this->data)) {
        $this->Session->setFlash(__('Invalid printing', true));
        $this->redirect(array('action' => 'index'));
    }
    if (!empty($this->data)) {
        if ($this->Printing->save($this->data)) {
            $this->Session->setFlash(__('The printing has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The printing could not be saved. Please, try again.', true));
        }
    }
    if (empty($this->data)) {
        $this->data = $this->Printing->read(null, $id);
    }
    $planning_list = $this->Printing->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
    $this->set('planning_list', $planning_list);
}

function delete($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid id for printing', true));
        $this->redirect(array('action'=>'index'));
    }
    if ($this->Printing->delete($id)) {
        $this->Session->setFlash(__('Printing deleted', true));
        $this->redirect(array('action'=>'index'));
    }
    $this->Session->setFlash(__('Printing was not deleted', true));
    $this->redirect(array('action' => 'index'));
}
}
}

```

```

<?php
class ProductionsController extends ApplicationController {

    var $name = 'Productions';
function index() {
    $this->Production->recursive = 0;
    $this->set('productions', $this->paginate());
}

function productionreport(){
    $this->Production->recursive = 2;

    if(!empty($this->params['named']['planning_id'])){
        $buyerID = $this->params['named']['planning_id'];
    }

    $conditions = array();

    if(!empty($this->params['named']['planning_id'])){
        $condition = array('Production.planning_id' => $buyerID);
    }

    if(!empty($condition)){
        $this->paginate = array( 'limit'=>20,'conditions'=>$condition,
'order'=>'Production.planning_id ASC');
        $this->set('plannings', $this->paginate());
    }else{
        $this->paginate = array( 'limit'=>20, 'order'=>'Production.planning_id ASC');
        $this->set('plannings', $this->paginate());
    }

    $planning_list = $this->Production->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
    $this->set('planning_list', $planning_list);
}

function view($id = null) {
    if (!$id) {
        $this->Session->setFlash(__('Invalid production', true));
        $this->redirect(array('action' => 'index'));
    }
    $this->set('production', $this->Production->read(null, $id));
}

function add() {
    if (!empty($this->data)) {
        $this->Production->create();
        if ($this->Production->save($this->data)) {
            $this->Session->setFlash(__('The production has been saved', true));
            $this->redirect(array('action' => 'index'));
        } else {
            $this->Session->setFlash(__('The production could not be saved. Please, try again.', true));

```

```

    }
  }
  $planning_list = $this->Production->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
  $this->set('planning_list', $planning_list);

  $workOrderInfo = $this->Production->Planning->findById($this->params['named']['id']);
  $this->set('workOrderInfo', $workOrderInfo);
}

function edit($id = null) {
  if (!$id && empty($this->data)) {
    $this->Session->setFlash(__('Invalid production', true));
    $this->redirect(array('action' => 'index'));
  }
  if (!empty($this->data)) {
    if ($this->Production->save($this->data)) {
      $this->Session->setFlash(__('The production has been saved', true));
      $this->redirect(array('action' => 'index'));
    } else {
      $this->Session->setFlash(__('The production could not be saved. Please, try again.', true));
    }
  }
  if (empty($this->data)) {
    $this->data = $this->Production->read(null, $id);
  }
  $planning_list = $this->Production->Planning->find('list', array( 'fields' => array('Planning.id',
'Planning.wo_no')));
  $this->set('planning_list', $planning_list);
}

function delete($id = null) {
  if (!$id) {
    $this->Session->setFlash(__('Invalid id for production', true));
    $this->redirect(array('action'=>'index'));
  }
  if ($this->Production->delete($id)) {
    $this->Session->setFlash(__('Production deleted', true));
    $this->redirect(array('action'=>'index'));
  }
  $this->Session->setFlash(__('Production was not deleted', true));
  $this->redirect(array('action' => 'index'));
}
}

```

```

<!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">
<head>
    <?php echo $this->Html->charset(); ?>
    <title>login- <?php echo $title_for_layout; ?></title>
    <?php
        echo $this->Html->meta('icon', $this-
>webroot.'favicon.png');

        echo $this->Html->css('style');

        echo $this->Html->css('themes/ui-
lightness/jquery-ui-1.8.10.custom.css');

        echo $this->Html->script('jquery-1.4.4.min');
        echo $this->Html->script('jquery-ui-
1.8.10.custom.min.js');

        echo $scripts_for_layout;
    ?>
<script>
    $(document).ready(function() {

        $(".actions a").button();
        $(".a.btn-delete").button({ icons: {primary:'ui-icon-trash'} });
        $(".a.btn-edit").button({ icons: {primary:'ui-icon-pencil'} });
        $(".a.btn-remove").button({ icons: {primary:'ui-icon-close'} });
        $(".a.btn-restore").button({ icons: {primary:'ui-icon-arrowthick-1-e'} });

        $(".paging a.next").button({ icons: {secondary:'ui-icon-seek-next'} });
        $(".paging a.prev").button({ icons: {primary:'ui-icon-seek-prev'} });
        $(".paging a").button();

        $(".button.btn-save").button({ icons: {primary:'ui-icon-plus'} });

        $("#flashMessage").append('<span style="float: left; margin-right: 0.3em;
margin-top: 6px;" class="ui-icon ui-icon-info"></span>');
        $("#flashMessage").addClass("ui-state-highlight ui-corner-all");
        $("#flashMessage").append('<button id="ok-flashMessage" class="ui-state-
active">Okay</button>');

        $("#authMessage").append('<span style="float: left; margin-right: 0.3em;"
class="ui-icon ui-icon-info"></span>');
        $("#authMessage").addClass("ui-state-error ui-corner-all");

        $(".fieldset").addClass("ui-corner-all");
        $(".error-message").prepend('<span class="ui-icon ui-icon-info" style="float: left;
margin-right: .3em;"></span>');
        $(".error-message").addClass("ui-state-error");
    });

```



```

        $("button").button();
/*      $(function() {
                                           $( "#radio" ).buttonset();
                                           });
*/
    $("#ok-flashMessage").click(function(){
                                           $("#flashMessage").slideUp();
    });

    /*Top Menu*/
    var timeout      = 500;
    var closetimer   = 0;
    var ddmenuitem   = 0;

    function jsddm_open()
    {
        jsddm_canceltimer();
        jsddm_close();
        ddmenuitem =
$(this).find('ul').eq(0).css('visibility', 'visible');}

        function jsddm_close()
        {
            if(ddmenuitem) ddmenuitem.css('visibility',
'hidden');}

        function jsddm_timer()
        {
            closetimer = window.setTimeout(jsddm_close,
timeout);}

        function jsddm_canceltimer()
        {
            if(closetimer)
            {
                window.clearTimeout(closetimer);
                closetimer = null;}}

        $(document).ready(function()
        {
            $("#jsddm > li").bind('mouseover', jsddm_open);
            $("#jsddm > li").bind('mouseout',
jsddm_timer));});
        document.onclick = jsddm_close;

    });
</script>
</head>
<body>
    <div id="container">
        <div id="header">
            <h1>login</h1>
        </div>
        <div id="topmenu" class="clearfix ui-widget-
header">
            <?php echo $this->element('topmenu-
outer'); ?>

```

```

        </div>
        <div id="content-whole">
            <div id="content" class="clearfix">
                <?php echo $this->Session-
>flash(); echo $session->flash('auth'); ?>
                <?php echo $content_for_layout;
?>
            </div>
        </div>
        <div id="footer"><p>Copyright &copy; <?php
echo date('Y'); ?> All Rights Reserved <a href="#" target="_blank"></a></P></div>
        </div>
        <?php echo $this->element('sql_dump'); ?>
</body>
</html>

<!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">
<head>
    <?php echo $this->Html->charset(); ?>
    <title> Work order management system</title>
    <?php
        echo $this->Html->meta('icon', $this-
>webroot.'favicon.png');
        echo $this->Html->css('style');
        echo $this->Html->css('themes/ui-
lightness/jquery-ui-1.8.10.custom.css');
        echo $this->Html->script('jquery-1.4.4.min');
        echo $this->Html->script('jquery-ui-
1.8.10.custom.min.js');
        echo $scripts_for_layout;
    ?>
<script>
    $(document).ready(function() {
        $(".actions a").button();
        $("a.btn-delete" ).button({ icons: {primary:'ui-icon-trash'} });
        $("a.btn-edit" ).button({ icons: {primary:'ui-icon-pencil'} });
        $("a.btn-remove" ).button({ icons: {primary:'ui-icon-close'} });
        $("a.btn-restore" ).button({ icons: {primary:'ui-icon-arrowthick-1-e'} });
        $(".paging a.next" ).button({ icons: {secondary:'ui-icon-seek-next'} });
        $(".paging a.prev" ).button({ icons: {primary:'ui-icon-seek-prev'} });
        $(".paging a" ).button();
    });

```

```

        $( "#button.btn-save" ).button({ icons: {primary:'ui-icon-plus'} });

        $("#flashMessage").append('<span style="float: left; margin-right: 0.3em;
margin-top: 6px;" class="ui-icon ui-icon-info"></span>');
        $("#flashMessage").addClass("ui-state-highlight ui-corner-all");
        $("#flashMessage").append(' <button id="ok-flashMessage" class="ui-state-
active">Okay</button>');

        $("#authMessage").append('<span style="float: left; margin-right: 0.3em; "
class="ui-icon ui-icon-info"></span>');
        $("#authMessage").addClass("ui-state-error ui-corner-all");

        $("#fieldset").addClass("ui-corner-all");
        $(".error-message").prepend('<span class="ui-icon ui-icon-info" style="float: left;
margin-right: .3em;"></span>');
        $(".error-message").addClass("ui-state-error");
        $("button").button();
/*
        $(function() {
                                $( "#radio" ).buttonset();
        });
*/
        $("#ok-flashMessage").click(function(){
                                $("#flashMessage").slideUp();
        });

        /*Top Menu*/
        var timeout      = 500;
        var closetimer   = 0;
        var ddmenuitem  = 0;

        function jsddm_open()
        {
                                jsddm_canceltimer();
                                jsddm_close();
                                ddmenuitem =
$(this).find('ul').eq(0).css('visibility', 'visible');}

        function jsddm_close()
        {
                                if(ddmenuitem) ddmenuitem.css('visibility',
'hidden');}

        function jsddm_timer()
        {
                                closetimer = window.setTimeout(jsddm_close,
timeout);}

        function jsddm_canceltimer()
        {
                                if(closetimer)
                                {
                                        window.clearTimeout(closetimer);
                                        closetimer = null;}}

        $(document).ready(function()

```

```

        {
            $('#jsddm > li').bind('mouseover', jsddm_open);
            $('#jsddm > li').bind('mouseout',
jsddm_timer);});
        document.onclick = jsddm_close;

    });
</script>
</head>
<body>
    <div id="container" >
        <div id="content-whole-front">
            <div id="content" class="clearfix"
style="margin-top:100px;">
                <?php echo $this->Session-
>flash(); echo $session->flash('auth'); ?>
                <?php echo $content_for_layout;
?>
            </div>
        </div>
        <div id="footer"><p>Developed by: Farhad |
Copyright &copy; <?php echo date('Y'); ?>All Rights Reserved <a href="#"
target="_blank"></a></P></div>
    </div>
    <?php echo $this->element('sql_dump'); ?>

</body>
</html>

```

```

<!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">
<head>
    <?php echo $this->Html->charset(); ?>
    <title>Admin Panel-Square Group Workorder Management System<?php echo
$title_for_layout; ?></title>
    <?php
    echo $this->Html->meta('icon', $this->webroot . 'favicon.png');
    echo $this->Html->css('style');
    echo $this->Html->css('themes/smoothness/jquery-ui-1.8.10.custom.css');
    echo $this->Html->script('jquery-1.4.4.min');
    echo $this->Html->script('jquery-ui-1.8.10.custom.min.js');
    echo $this->Html->css('fancybox/jquery.fancybox-1.3.4.css');
    echo $this->Html->script('jquery.fancybox-1.3.4.pack.js');
    echo $this->Html->script('jquery-ui-timepicker-addon.js');
    echo $scripts_for_layout;
    ?>
    <script>
    $(document).ready(function() {

```

```

    {secondary:'ui-icon-seek-next' } });
    {primary:'ui-icon-seek-prev' } });

    $(".paging a.next").button({ icons:
    $(".paging a.prev").button({ icons:
    $(".paging a").button();

    $("#flashMessage").append('<span style="float:
left; margin-right: 0.3em; margin-top: 6px;" class="ui-icon ui-icon-info"></span>');
    $("#flashMessage").addClass("ui-state-highlight
ui-corner-all");
    $("#flashMessage").append('<button id="ok-
flashMessage" class="ui-state-active">Okay</button>');

    $("#authMessage").append('<span style="float:
left; margin-right: 0.3em; " class="ui-icon ui-icon-info"></span>');
    $("#authMessage").addClass("ui-state-error ui-
corner-all");

    $("#fieldset").addClass("ui-corner-all");
    $(".error-message").prepend('<span class="ui-
icon ui-icon-info" style="float: left; margin-right: .3em;"></span>');

    $(".button").button();

    $("#ok-flashMessage").click(function(){
        $("#flashMessage").slideUp();
    });

    /*Top Menu*/
    var timeout      = 500;
    var closetimer   = 0;
    var ddmenuitem  = 0;

    function jsddm_open()
    {
        jsddm_canceltimer();
        jsddm_close();
        ddmenuitem =

$(this).find('ul').eq(0).css('visibility', 'visible');}

    function jsddm_close()
    {
        if(ddmenuitem)

    function jsddm_timer()
    {
        closetimer =

    function jsddm_canceltimer()
    {
        if(closetimer)
        {

window.clearTimeout(closetimer);

```

```

        closetimer = null; }

$( '#jsddm > li' ).bind( 'mouseover', jsddm_open );
$( '#jsddm > li' ).bind( 'mouseout', jsddm_timer );

document.onclick = jsddm_close;

    });
</script>
</head>
<style type="text/css">
.fr a{color:#e80000;}
</style>
<body>
<?php
//      $group_name = 'Admin';
      $group_id = $session->read('Auth.User.group_id');
      if($group_id == '1')
          $group_name = 'Super Administrator';
      elseif($group_id == '2')
          $group_name = 'Content provider';
      elseif($group_id == '3')
          $group_name = 'Operator';
      else
          $group_name = 'Registered User';
?>
      <div id="container">
          <div id="front-header" class="clearfix">
              <div class="fl" style="padding:30px
0"><?php echo $this->Html->image("apps/logo.png", array('width'=>'489sssp',
'height'=>'100px'));?></div>
              <!--<div class="fl" style="line-
height:160px;"><h2 style="font-size:24px">Textile - Workorder Management System</h2></div>--
          >
          <div class="fr" style="color:#fff; margin-right:10px; margin-top:20px; ">
              <?php
                  date_default_timezone_set('Asia/Dhaka');
                  $groups =
array('1'=>'Administrator','2'=>'Content provider','3'=>'Admin','4'=>'Planning User','5'=>'Fabric
User','6'=>'Cutting User','7'=>'Embrodary User','8'=>'Printing User','9'=>'Production
User','10'=>'Finishing User');
                  if ($session-
>check('Auth.User.id')){
                      echo "Hi, " . $session-
>read('Auth.User.name')." | ".date('F-j, Y g:i A');
                      echo '<br \>';
                      echo $html-
>link(__('Logout', true), array('controller' => 'users', 'action' => 'logout', 'operator' => false));
                      echo " | " . $html-

```

```

>link(__('Change password', true), array('controller' => 'users', 'action' => 'changepassword'));
                                                    echo "<div style='margin-
top: 5px'>Logged In As: <strong>".$groups[$session-
>read('Auth.User.group_id')]."</strong></div>";
                                                    }
                                                    else
                                                    echo $html-
>link(__('Login', true), array('controller' => 'users', 'action' => 'login'));
                                                    //echo '<br />';
                                                    //echo $this->Html->link(__('Edit
Profile', true), array('controller' => 'users', 'action' => 'editprofile'));
                                                    ?>
</div>
                                                    </div>
                                                    <div id="topmenu" class="clearfix ui-widget-
header">
                                                    <?php echo $this->element('topmenu');
?>
                                                    </div>
                                                    <div id="content-whole">
                                                    <div id="content" class="clearfix">
                                                    <?php echo $this->Session->flash();
echo $session->flash('auth'); ?>
                                                    <?php echo $content_for_layout; ?>
                                                    </div>
                                                    </div>
                                                    <div id="footer"><p>Copyright &copy; <?php
echo date('Y'); ?> All Rights Reserved</p></div>
                                                    </div>
                                                    <?php echo $this->element('sql_dump'); ?>
</body>
</html>

```