

**ONLINE REGISTRATION SYSTEM FOR
TEXTILE INDUSTRIES AND GSP APPLICATION: A CASE STUDY**

by

Muhammad Hafizur Rahaman

POST GRADUTE DIPLOMA IN INFORMATION AND COMMUNICATION
TECHNOLOGY

Institute of Information and Communication Technology(IICT)
BANGLADESH UNIVERSITY OF ENGINEERING TECHNOLOGY(BUET)

2015

This project titled “ONLINE REGISTRATION SYSTEM FOR TEXTILE INDUSTRIES AND GSP APPLICATION: A CASE STUDY” submitted by Muhammad Hafizur Rahaman, Roll No:1008311007, Session: October/2008, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Post Graduate Diploma in Information and Communication Technology 23 May, 2015.

Board of Examiners

1.

Dr. Md. Saiful Islam
Professor
Institute of Information and Communication Technology
BUET, Dhaka-1000, Bangladesh. Chairman

2.

Dr. Md. Rubaiyat Hossain Mondal
Assistant Professor
Institute of Information and Communication Technology
BUET, Dhaka-1000, Bangladesh. Member

3.

Ms. Shamim Ara Showkat
Assistant Professor
Institute of Information and Communication Technology
BUET, Dhaka-1000, Bangladesh. Member

Candidate's Declaration

It is hereby declared that this report or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Muhammad Hafizur Rahaman

Dedicated
To
My Parents and Family Members

TABLE OF CONTENT

Title	Page No.
Board of Examiners	ii
Candidate Declaration	iii
Dedication	iv
Table of Contents	v
List of Figures	viii
Abbreviation & Key Terms	ix
Acknowledgement	x
Abstract	xi

CHAPTER-I: INTRODUCTION

1.1	Introduction	01
1.2	Objectives with specific aims and possible outcome	02
1.3	Organization of the Project Report	02

CHAPTER-II: SYSTEM STUDY & IMPLEMENTATION TOOLS

2.1	Situation analysis	03
2.2	Strategy	03
2.3	Functions of Online Registration and GSP application	03
2.4	Search features	03
2.5	Software as Document Management System	03
2.6	Process Analysis	04
2.7	Process Flow	05
2.8	Process-Based Time Estimation	05
2.9	Introduce tools and technology	07
2.10	Implementation Technologies	08
2.11	PHP	12
2.12	Chapter Summery	13

CHAPTER-III: SYSTEM DESIGN AND DEVELOPMENT

3.1	Database	14
3.2	Definition of Database	
3.3	Types of Database	

TABLE OF CONTENT

Title	Page No.
3.4 Database System versus File system	14
3.5 Procedures of Database	15
3.6 Information Modeling	15
3.7 Concept of Information Modeling	15
3.8 Procedures for Information Modeling	16
3.9 Data Modeling	17
3.10 Types of data Modeling	17
3.11 Entity–relationship model	18
3.12 Methodology	19
3.13 Selection of the Project	19
3.14 Collection of Information for Developing	20
3.15 Work Flow Diagram of online reg. System for textile industry and GSP Application	20
3.16 Entity Relationship Diagram	22
3.17 Relationship	23
3.18 Database Development	23
3.19 Description of the Database Development	23
3.20 Testing	26
3.21 Steps Performed for Testing	26
3.22 Testing Classification	26
3.23 Sample Data used for the Development & Testing	27
3.24 Chapter Summery	28

CHAPTER-IV: IMPLEMENTATION

4.1 What is System Design	29
4.2 Basic Design of the PC Software	29
4.3 Architecture Overview	32
4.4 Collection and update Process of Basic Information	37
4.5 GSP Service	39
4.6 Basic View or Administrator Interface	41
4.7 Chapter Summery	46

CHAPTER-V: CONCLUSION

5.1 Conclusion	47
5.2 Recommendation	47
5.3 Future Works	48

TABLE OF CONTENT

Title	Page No.
<i>References</i>	49
<i>Appendix</i>	50

LIST OF FIGURES

Figure No	Figure Caption	Page No
Fig-2.1	The process flow of Registration System for Textile Industries and GSP application	05
Fig-2.2	Three-tier architecture	07
Fig -2.3	Communication between client and web-server	09
Fig -2.4	Steps for creating a client side Dynamic Web Page	11
Fig-2.5	The steps involved in creating a server side dynamic web page	12
Fig-3.1	One-to-One Relation	16
Fig-3.2	One-to-Many Relation	16
Fig-3.3	Many-to-One Relation	16
Fig-3.4	Many-to-Many Relation	16
Fig-3.5	Work Flow Diagram of online GSP application process	21
Fig-3.6	Work Flow Diagram of online Registration process	22
Fig-3.6	E-R Diagram of Online Registration System for Textile Industries and GSP Application system	23
Fig-3.7	The relationship among the table.	24
Fig-4.1	Architecture of web application	33
Fig-4.2	Registration Form (ERF)	38
Fig-4.3	Administrator Login Page.	41
Fig-4.4	Software Admin Home Page.	41
Fig-4.5	Data Insert Process (Apply for Reg.).	42
Fig-4.6	Data Insert Process(Apply for GSP.).	43
Fig-4.7	Data View and Update Process.	44
Fig-4.8	User Login Page.	45
Fig-4.9	User Interface Home Page.	45

Abbreviation & Key Terms

EPB	Export Promotion Bureau
GSP	Generalized System of Preferences
EU	European Union
PHP	Hypertext Preprocessor
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
REX	Register Exporter
UIR	User Interface Re-engineering
DR	Database Re-engineering
DAI	Database Administrative Interface
EBF	Existing Bug Fixing
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
GPL	General Public License
DBMS	Database Management System
ER	Entity–relationship
RRA	reshaped relational algebra
BIN	Business Identification Number
HS code	harmonized system code
PC	Personal Computer
TCP/IP	Transmission Control Protocol/ Internet Protocol
WWW	World Wide Web
ERC	Exporter Registration Certificate
ERF	Exporter Registration Form
LDC	Least Developed Countries
RMG	Ready–Made Garments
GDP	Gross domestic product

Acknowledgement

First of all I would like to express my gratitude to Almighty Allah for giving me the opportunity to conduct this project. I would like to thank Dr. Md Saiful Islam, Professor, Institute of Information and Communication Technology, Bangladesh University of Engineering Technology, Dhaka, Bangladesh. He has assigned me an interesting and useful topic, which has a wide range of application in real world. He has provided all sorts of support regarding the project work. Without his proper guidance, advice, continual encouragement and active participation in this process of this work, it would have not been possible.

I am grateful to all the teachers, officers and staffs of Information and Communication Technology for giving me their kind support and information during the study.

I am also indebted to my parents whose continuous support all over my life has brought me this far in my career.

I finally offer special thanks to the Almighty that I have been successful in my effort to accomplish the study.

Abstract

The economy of Bangladesh is largely dependent on agriculture. However, the Ready Made Garments (RMG) sector has emerged as the biggest earner of foreign currency. The sector contributes significantly to the Gross Domestic Product (GDP). It also provides employment to around 4.2 million Bangladeshi people. In this context the government authority should provide better services to enhance our total export. The present Generalised System of Preferences(GSP) process has, over the years, become heavy on paper and light on verification which must be tackled now to avoid having to deal with more bottlenecks and fraudulent transactions in the future and to comply with the European Union(EU) regulations on verification. This case study has conducted a detailed work for developing and integrated and interconnected “GSP online” system based on a detailed review of steps and processes for the existing Form-A filing and a substantial leap forward toward efficiency and process improvement to meet the requirements. As a consequence, on-line registration for textile Garments and GSP application submission system is a database system that makes a bridge between the Ready Made Garments (RMG) Exporter and the service providing organization. So to manage a large scale of customer (RMG exporter), it requires a complete database. In this case study, the database is designed efficiently. A web based application that can be used for both Exporter and the service providing organization to view and update information by using user interface and administrator interface. The Exporter can easily use this software from anywhere for submit his/her application for registration and also for tracking the record. They can also submit their GSP application and keep all previous GSP certificate records. On the other hand, the administrators of the organization can easily serve their client through this database software. The main objective of the service providing organization is to store REX(Registered Exporter) and manage the GSP issuing process to solve their daily problems which is perfectly organized in this database project.

1.1 Introduction

The state-owned Export Promotion Bureau (EPB) has taken initiatives to prevent forgery by using Generalized System of Preferences (GSP). The initiatives include verification of documents just like customs and shipping line, port authority, etc. EPB is struggling to cope with massive volumes of paperwork and total reliance on the manual data entry [1]. There is no use of risk management or post clearance audits. No verification visits are there and all application are submitted in equal risk. The EPB is entirely reactive to European Union (EU) investigations. No meaningful proactive ex-officio action is taken. GSP fraudsters are using customs, trade associations, Bangladesh Bank, European Union etc. Ineffectual checks involving dozens of signatures and stamps are conducted by the EPB are duplicated by customs and other agencies. Verification requested from EU and EU member states are often poorly handled and the responses frequently out-of-time and unreliable. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity [2-6]. A web application is an application that is accessed via web or mobile over a network [7]. A computer based software system is essential for managing the total database of textile industries to solve the said problem of EPB. [8]. In this project, a web application has been developed to solve the above problems of the textile industries of Bangladesh.

This is a web application that runs in a web browser. It is created in a browser-supported programming language (such as the combination of JavaScript, PHP, HTML and CSS) and relies on a web browser to render the application. Nowadays, web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility.

1.2 Objectives with specific aims and possible outcome

With regular software, we are limited to work on a computer that has the complete program and latest data. My place of work is dictated by the location and availability of this computer. Such restriction is not present when I use web-based software. It does not impose any limit to where I can work. It can be the head office, an international branch, my seat while I am traveling, or any place where I can access the Internet. In this context, the objective of the whole system has five elements such as:

- i. A web based software system has developed for textile industries to handle GSP issuing process.
- ii. This system has provided appropriate interface to the administrator and different level of application user.
- iii. The system has stored and updated the detail information about client and it must ensure transparency and accountability of the persons involved.
- iv. The system has been enabled to meet the future requirement.
- v. The system does not impose any limit to where you can work. It can any place where you can access the Internet from.

1.3 Organization of the Project Report

- Chapter 1 describes the overview and objectives of the project.
- Chapter 2 describes about the System study & Implementation tools
- Chapter 3 describes the Tools and Technology about the software
- Chapter 4 describes step by step Implementation of the system.
- Chapter 5 Finally describes the conclusion and recommendations for future works of the system.

2.1 Situation analysis:

The state-owned Export Promotion Bureau (EPB) has taken initiatives to prevent forgery by using generalized system of privilege (GSP). The initiatives include, verification of documents just like customs and shipping line , port authority etc. EPB is struggling to cope with massive volumes of paperwork and total reliance on the manual data entry. There is no use of risk management or post clearance audits. No verification visits are there and all application are submitted in equal risk. The EPB is entirely reactive to European Union investigations. No meaningful proactive ex-officio action is taken. GSP fraudsters are using customs, trade associations, Bangladesh Bank, European Union etc. Ineffectual checks involving dozens of signatures and stamps are conducted by the EPB are duplicated by customs and other agencies. Verification request from EU and EU member states are often poorly handled and the responses frequently out-of-time and unreliable.

2.2 Strategy:

A well conceived strategy pushes the decision making down to the lowest possible level, closest to where the problems are encountered, while at the same time provides the oversight necessary to build a coherent system.

The system has five elements to it:

- Establish valid Exporter Database;
- Information Management System for GSP Certificate;
- Protect Fraudulent activity;
- GSP Certificate issuing process would be transparent and Faster;
- Reduce Cost and Time;

2.3. Functions of Online Registration and GSP application:

The Registration System for Textile Industries and GSP application software has seven functions which are described below:

1. Exporters submit online application along with the hardcopies of all required papers;
2. Admin registers of exporters;
3. Exporters submit Form-A online;
4. Exporters type paper Form-A and submits to EPB with payment confirmation;
5. EPB Admin verifies Form-A information;
6. EPB Admin Form-A, delivers, and closes the file
7. EPB Admin can cancel application for valid requests by exporters.

2.4. Search features:

The software has a common search box runs across all modules and tabs for easy reference of GSP files by using a) EPB Reg. No, Form-A No, Date, Contact, Country etc.

2.5. Software as Document Management System:

The software itself can be used to filing inventory and document management system by all users. It will keep all information for upwards of two years and can be searched.

2.6. Process Analysis

Online Registration System for Textile Industries and GSP application software addresses registration of the exporters, GSP application filing and submit on-line in web-enabled environment. The delivered certificate can be tracked, verified, and retrieved in the future by the users of the software. The record can be crosschecked with export receipts(Bangladesh Bank) and customs data(Bangladesh Customs) for accuracy and for assurance that the goods have indeed originated from Bangladesh. The software will help in managing and strengthening EPB's GSP operations from the exporter registration to the operations of the back office and fulfilling EU compliance in accordance with the Commission Regulations(EU) 1063/2010.

2.7. Process step:

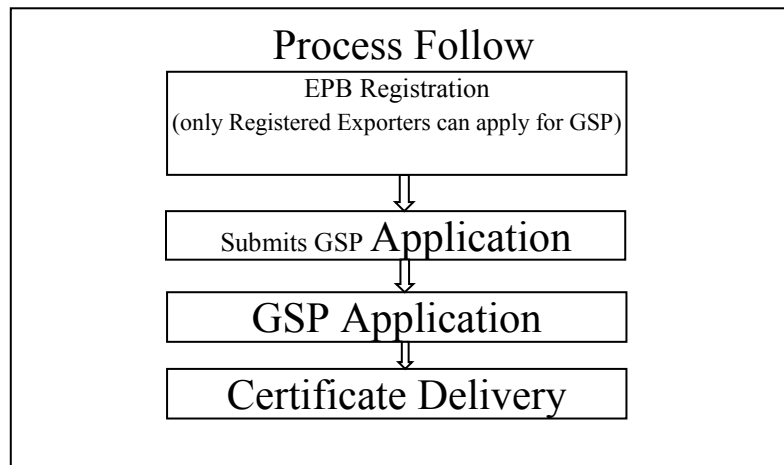


Fig 2.1: Figure shows the process steps of Registration System for Textile Industries and GSP application

2.8. Process-Based Time Estimation: For process-based time estimation, the process is decomposed into a relatively small set of activities or tasks. Then, the effort required to accomplish each task is estimated. Based on project scope, the following software functions are defined:

- **User Interface Re-engineering (UIR) :** Reengineering the user interface of a legacy system may involve redesigning the underlying computation to increase interaction Data, that is internal to a computation, must be exposed so that it can be presented in a graphical interface, and the computation must be restructured to compute this data on demand.
- **Database Re-engineering (DR):** Database engineering is a sub domain of software engineering that addresses such processes as analysis, design, implementation, reengineering, migration, integration of databases and of their applications.
- **Database Administrative Interface (DAI):** Displays all of the Databases, Groups, Forests, Security objects, and Hosts configured for your system.
- **Existing Bug Fixing (EBF):** Small bugs include types in code or specification, algorithm improvements for correctness or performance.

Calculation:

Time Distribution: Time allocation is followed the time distribution ratio.

Time Distribution Ratio:
Let X = 4; Customer evaluation = Customer Comments = Planning = 2X;
Risk analysis = 3X;
Engineering (Analysis + Design) = 9X;
Constructive Release = (coding +testing) = 27X;

Development period 6 months = 180 Days; so time estimation is as follows:

SL	Activity	Duration
1	Customer Comments	8 days
2	Planning	8 days
3	Risk analysis	12 days
4	Engineering(Analysis + Design)	36 days
5	Constructive Release(Code + test)	108 days
6	Customer evaluation	8 days
Total		180 days = 6 months

Time Estimation Chart:

Activity	Cust. Comm. (days)	Planning (days)	Risk Analysis (days)	Engineering		Constructive Release		Cust. Eval. (days)	Totals (days)
				Analysis	Design	Code	Test		
Task →									
Function									
UIR	2	2	3	4	5	25	2	2	45
DR	2	2	3	4	5	25	2	2	45
DAI	2	2	3	4	5	25	2	2	45
EBF	2	2	3	4	5	25	2	2	45
Sub Total	8	8	12	16	20	100	8	8	180
% effort	4.44%	4.44%	6.67%	8.89%	11.11%	55.56%	4.44%	4.44%	100%

Estimated time 6 Months

2.9 Introduce tools and technology :

To implement a web application client-server architecture is required. The most popular client-server architectures are the two-tier and the three-tier architecture. The choice of architecture affects the development time and the future flexibility and maintenance of the application. While selecting the architecture which would be the most suitable for an application, many factors including the complexity of the application, the number of users and their geographical dispersion are considered. This system is designed based on a traditional three-tier architecture used by many web applications. Three-tier architecture includes a presentation layer, business rules/ logic layer, and the data layer. The three-tier architecture is shown in fig-2.2.

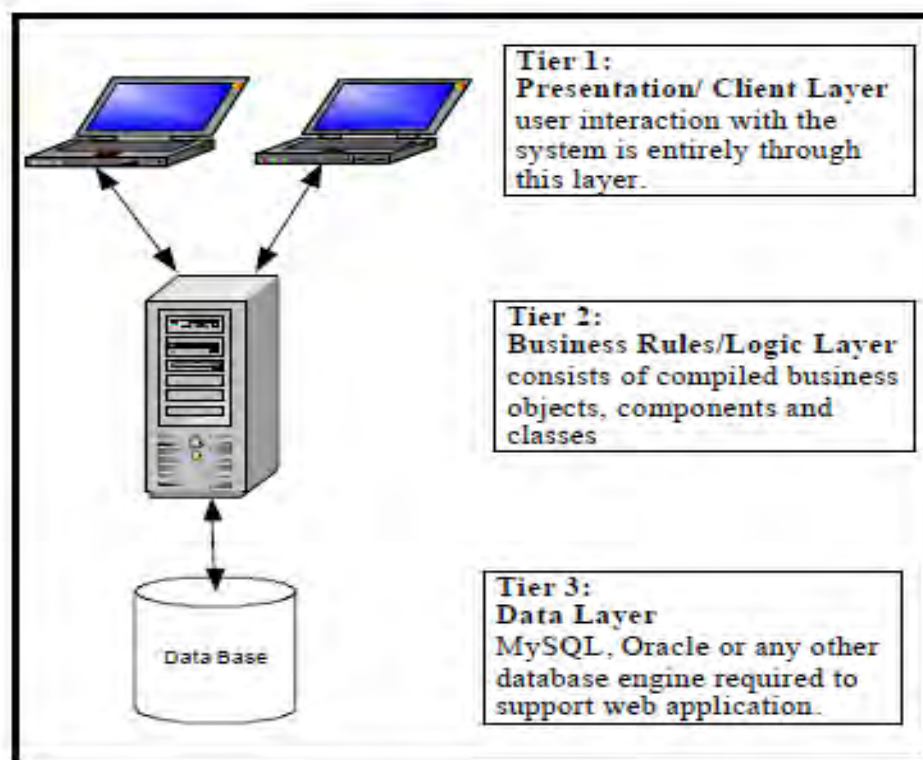


Figure 2.2: Three-tier architecture

The three-tier architecture is generally used when an effective distributed client/server design is needed that provides

- increased performance
- flexibility
- maintainability

- reusability and
- scalability

This model hides the complexity of distributed processing from the user. These features have made the three-tier architecture a popular choice over the two-tier architecture for Internet applications. The three layers are discussed below.

The Data layer is responsible for data storage. Primarily this tier (layer) consists of one or more relational databases and/or file systems.

The Business Rules/Logic layer is the middleman between the presentation layer and the data layer. This middle tier was introduced to overcome the deployment limitation (whenever the application logic changed the application had to be redistributed to each and every client) of the two-tier architecture. The middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users.

The Presentation Layer, also called the Client tier, is responsible for the presentation of data, receiving user events, and controlling the user interface. The user interaction with the system is entirely through this layer.

2.10 Implementation Technologies:

To implement any web-based application a web server is required. A web server is a piece of software that manages web pages and makes them available to the 'client' browser – via a local network or over the Internet. The web server can be accessed remotely or locally. There are many web servers available such as Apache, Internet Information Services IIS, Netscape Web Server and so on. By typing a URL (Uniform Resource Locator) into the address box of the browser the communication between a browser and a web server is started. Each conversation consists of two pieces:

- a request for information from the browser software and
- a response from the server addressed by the URL.

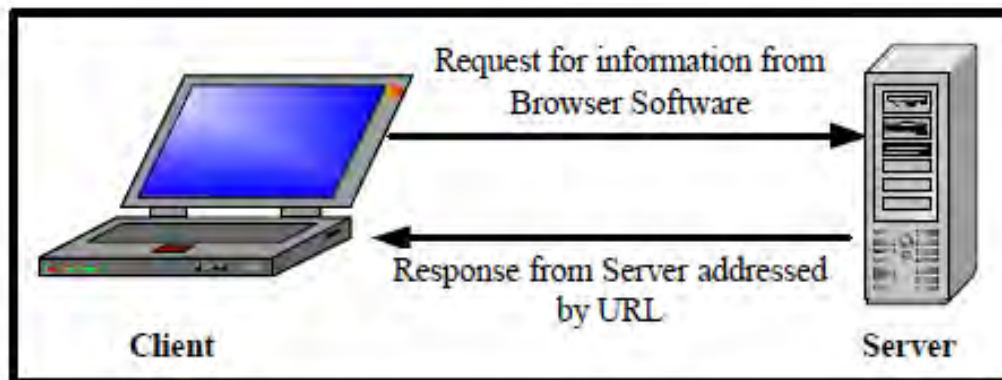


Figure 2.3: Communication between client and web-server

The principle of communication between a client and a server is composed of successions of requests and responses. This communication is shown in Figure 3.2. for the implementation of this application, IIS version 5.1 is used as web server.

2.10.1 Apache

The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to text content.

A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary storage, but this is not necessarily the case and depends on how the web server is implemented.

While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files.

Many generic web servers also support server-side scripting using PHP, or other scripting languages. This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to create HTML documents dynamically ("on-the-fly") as opposed to

returning static documents. The former is primarily used for retrieving and/or modifying information from databases. The latter is typically much faster and more easily cached.

2.10.2. Dynamic Web Pages

In a dynamic web page contents (text, images, fields, etc.) on the web page can change, in response to different contexts or conditions. There are two ways to create this kind of web pages:

1. Using client-side scripting to change interface behaviors within a specific web page
2. Using server-side scripting to change the sequence of the web pages or web content supplied to the browser. These are determined by conditions such as data in a posted HTML form, parameters of the URL, the type of browser is being used and so on.

2.10.2.1. Client-Side Dynamic Web Page

In the client-side model, modules (or plug-ins) attached to the browser do all the work of creating dynamic pages. The HTML code is sent to the browser along with a separate file containing a set of instructions. These instructions are referenced from and within the HTML page. It is also quite common to find these instructions intermingled with the HTML codes. The modules within the browser then use the instructions to generate pure HTML for the page, generating the page dynamically on request, which is sent back to the browser. This model hence involves six steps:

1. An author writes a set of instructions for creating HTML, and saves it within an .htm file. The instructions might be contained within the .htm file, or within a separate file.
2. Sometime later, a client (user) requests a page by typing it into their browser, and the request is passed from the browser to the web server.
3. The web server locates the .htm page, and any other file that contains the instructions.

4. The web server sends both the newly created HTML stream and instructions back across the network to the browser.
5. A module within the browser processes the instructions and returns it as HTML
6. The HTML is then processed by the browser which displays the page

Figure 2.3: shows the steps involved in creating a client side dynamic web page

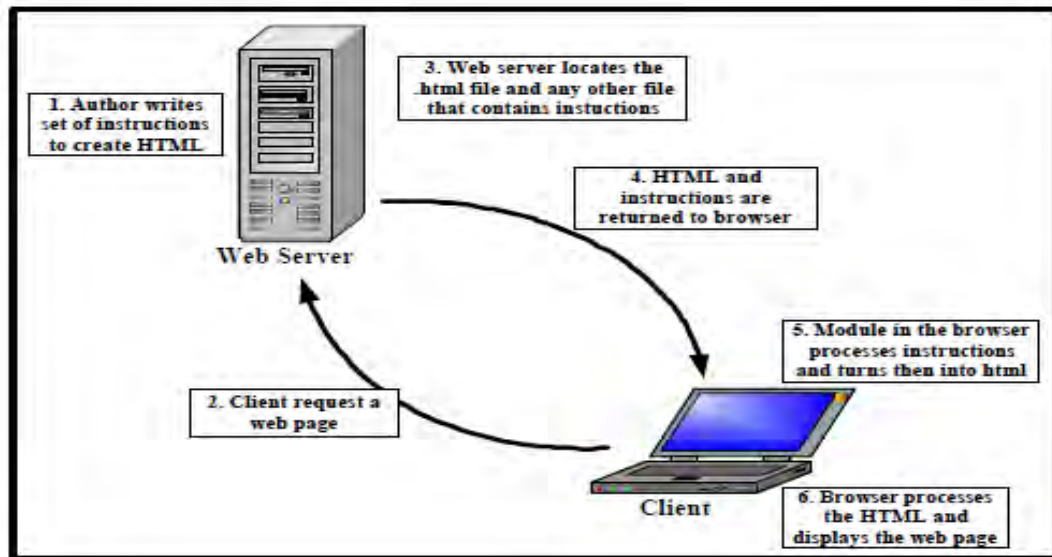


Figure 2.4: Steps for creating a client side Dynamic Web Page

Client-side technologies have fallen out of favor in recent times. The main reason is that it takes a long time to download, especially when there is a separate second file for instructions. Another drawback is that each browser interprets these instructions in a different way and there is no guarantee that all the browsers will understand them. It is also difficult to write client-side code that uses server-side resources like database, as it is interpreted at the client-side. There are client-side technologies like Ajax (shorthand for Asynchronous Java and XML) for providing dynamic contents. The client side technologies are a mixture of scripting languages, controls and full fledged programming languages including JavaScript.

2.10.2.2. Server-Side Dynamic Web Page

In the server-side model, when a user types a page request such as an ASP, PHP or ASP.NET page, the web server locates the page and invokes the appropriate servicing program. The servicing program is not a part of the Web server but it is an

independent executable program running on the Web server. The servicing program, processes any user input, determines the action that must be taken, interacts with any external sources and finally produces an HTML document and terminates. The Web server then sends the HTML document back to the user's browser where it is displayed. The page is thus generated dynamically upon request. The six steps involved in developing a server side dynamic web page are

1. A web author writes a set of instructions for creating HTML, and saves these instructions within a file such as a .php file
2. Sometime later, a user types a page request into their browser, and the request is passed from the browser to the web server
3. The web server locates the file of instructions and invokes the appropriate servicing program
4. The servicing program follows the instructions in order to create a stream of HTML
5. The web server sends the newly created HTML stream back across the network to the browser
1. The browser processes the HTML and displays the page

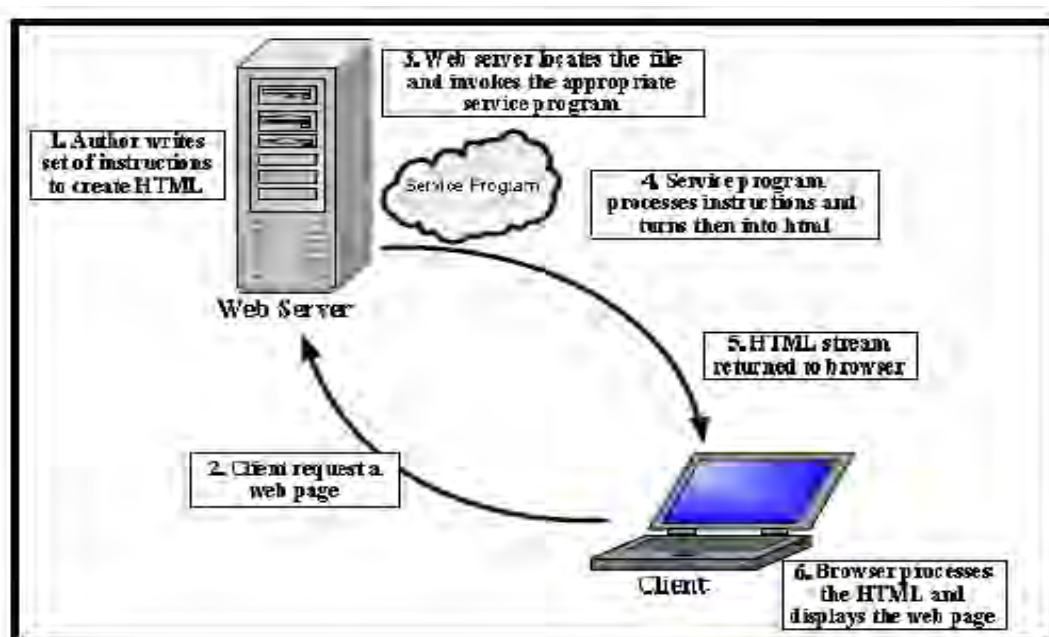


Figure 2.5: shows the steps involved in creating a server side dynamic web page

2.11. PHP

PHP is a server-side scripting language designed for web development and also used as a general-purpose programming language. PHP is now installed in more than 244 million websites and 2.1 million web servers. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to the restrictions on the usage of the term PHP. PHP can be deployed on most web servers and also as a standalone shell in almost every operating system and platform, free of charge.

2.12. Chapter Summery

This chapter has described the situation analysis, strategy, software functions and features and time based estimation . This chapter has also focused on the platform of software and the tools and technique which are used for developing this system.

3.1. Database

Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes attempts to unauthorized access. If data are to be shared among several users, the system must avoid possible anomalies result.

3.2. Definition of Database

A database management system (DBMS) is a collection of interrelated data and a set of program to access those data. The collection of data usually refer to the database, containing information relevant to an enterprise. The primary goal of a database is to provide a way to store and retrieve database information that is both convenient and efficient.

3.3. Types of Database Application

There are two types of application:

1. Desktop Based
 - Standalone, - Network Support
2. Web Based

3.4 Database System versus File system

The typical file system is supported by a conventional operating system. The system stores permanent record in various files and it needs different application programs to extract records from it and add records to the application file. Before database management system came along, organization usually stored information/data in such system.

Keeping data/information in file processing system has number of major disadvantages:

- Data redundancy and inconsistency
- Difficult in data access

- Data isolation
- Integrity problem
- Atomicity problem
- Concurrent access anomalies
- Security problem

3.5 Procedures of Database

There are various procedures for database Design. Some of the database procedures are shown below:

- Create Database Tables
- Normalization
- Relationship
- Implement with Oracle/MySQL
- Coding
- Debugging and Testing
- Database level Modification
- Finalization
- Implementation

3.6 Information Modeling

Whenever understanding of a real world phenomenon or interaction is desired the first step is to build a simple model representing the phenomenon and perform study and evaluation of the model rather than actual interaction. In building model the ambiguity and complexity in the real world is avoided totally or compensated with valid approximation. In fact a systematic study of any phenomenon whether economic, industrial or scientific is done through modeling. Information modeling pertains to the development of model in information generation, storage, destruction, evaluation, manipulation, synthesis and utilizations. These models help in systematization of information generation, flow, interpretation synthesis of more information.

3.7 Concept of Information Modeling

The first step of information modeling is to precisely choose that part of reality, which is of interest. This is known as Entity. Entity defines other parameter. Entities

are atomic that they are invisible. The parameters of entity are known as attributes. Attributes are known as quantum of information, which describes the entity entirely. The second step in information modeling is relationship among entities. It is the most important in making sense of the entity and inter-entity relationship. The relationship is the information, which links two entities. The relationship can be of four categories:

- One-to-One (1:1) – One instance of the first entity can correspond to only one instance of the second entity. It is known as binary relationship.



Fig-3.1: One-to-One Relation

- One-to-Many(1:M) – One instance of the first entity can correspond to more than one of the second entity.

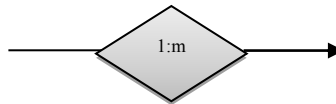


Fig-3.2: One-to-Many Relation

- Many-to-One (M:1) – More than one instance of the first entity can corresponds to the same one instance of the second entity

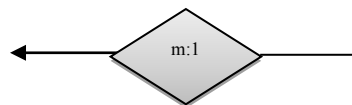


Fig-3.3: Many-to-One Relation

- Many-to-Many (M:M) – More than one instance of the first entity can correspond to more than one instance of the second entity.

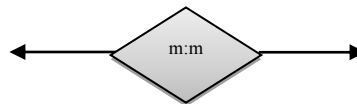


Fig-3.4: Many-to-Many Relation

3.8 Procedures for Information Modeling

The steps for Information Modeling and rules of data normalization are described below:

- ❖ The universe of discourse or the part of real world should be selected properly with minimum redundancy but completely in terms of entities.
- ❖ Entities are to be classified according to their attributes contents and merged, which has One-to-One (1:1) Relationship into composite entities.
- ❖ Their relationship among entities should be defined and named.
- ❖ Normalization should be done to reduce redundancy and increase integrity.

3.9 Data Modeling

A data model is a model describing the data in an organization. It provides frame work for abstracting the essential qualities or characteristics of data. Data modeling is the process of abstracting and documentation using a data model.

Data modeling creates hierarchies of abstraction along two dimensions: aggregation and generalization. Aggregation identifies data item as arts of higher-level, more aggregate descriptor. Generalization creates categories into which a data item may be classified.

There are two major classes of data models – logical data models and physical data models. These two classes reflects the fact that efficient physical storage and retrieval of data must be designed around the physical characteristics of storage media and devices, but user of data should be able to describe, think about and use data without being concerned of its physical storage.

3.10 Types of data Modeling

The data modeling consists of three interrelated pieces of information, the data object, the attributes that describes the data object and the relationship that connects data objects to one another. A data object is a representation of almost any composite information that must be understood by software. Composite information means something that has a number of different properties or attributes.

Six different types of data models are given below:

- Entity-Relationship Model (E-R Model)
- Relational Model
- Object-Oriented Data Model

- Object-Relational data Model
- Hierarchical Data Model
- Network Data Model

3.11. Entity–relationship model

An entity–relationship model is a systematic way of describing and defining a business process. The process is modeled as components (entities) which are linked with each other by relationships that expresses the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) which characterizes them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams.

An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers represent the relationships.

Entity–relationship modeling

An entity may be defined as a thing capable of an independent existence that can be uniquely identified. An entity is an abstraction from the complexities of a domain. When we speak of an entity, we normally speak of some aspects of the real world that can be distinguished from other aspects of the real world.

An entity is a thing that exists either physically or logically. An entity may be a physical object such as a house or a car (they exist physically), an event such as a house sale or a car service, or a concept such as a customer transaction or an order (they exist logically—as a concept). Although the term entity is the one most commonly used, following chain we should really distinguish between an entity and an entity-type. An entity-type is a category. An entity, strictly speaking, is an instance of a given entity-type. There are usually many instances of an entity-type. Because the term entity-type is somewhat cumbersome, most people tend to use the term entity as a synonym for this term.

Entities can be thought of as nouns. Examples: a computer, an employee, a song, a mathematical theorem.

A relationship captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. Examples: an owners relationship between a company and a computer, a supervises relationship between an employee and a department, a performs relationship between an artist and a song, a proved relationship between a mathematician and a theorem.

The model's linguistic aspect described above is utilized in the declarative database query language ERROL, which mimics natural language constructs. ERROL's semantics and implementation are based on reshaped relational algebra (RRA), a relational algebra that is adapted to the entity–relationship model and captures its linguistic aspect.

Entities and relationships can both have attributes. the proved relationship may have a date attribute.

Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key.

Entity–relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets(all entities of the same entity type) and relationship sets(all relationships of the same relationship type). Example: a particular song is an entity. The collection of all songs in a database is an entity set. The eaten relationship between a child and her lunch is a single relationship. The set of all such child-lunch relationships in a database is a relationship set. In other words, a relationship set corresponds to a relation in mathematics, while a relationship corresponds to a member of the relation.

3.12. Methodology

Methodology of study portraits the way how the study is being completed. An organized methodology is the guidelines for successful completion of study. Following methodology is being used to complete this project.

3.13. Selection of the Project

In recent years, Export Promotion Bureau is struggling to cope with massive volumes of paperwork. A lot of time is being consumed at the both end(exporter and the

Service provider). On the other hand Export Promotion Bureau have identified the need of quick and transparent service. As a consequence, Online Registration System for Textile Industries and GSP Application System has risen to the agenda of strategies.

3.14. Collection of Information for Developing Database

Exporter is the main resource in Online Registration System for Textile Industries and GSP Application System. Information such as exporter related information(factory name, address, owner name, BIN etc.), GSP certificate related information(Issue Number, Form-A No, Importer name, category, hs code etc.). that should be implemented in the database system. Every raw data has been captured and stored in a file for further analysis.

The work flow diagram of Online Registration System for Textile Industries and GSP Application System is given below:

3.15 Work Flow Diagram of online reg. System for textile industry and GSP Application
GSP Issuing Process

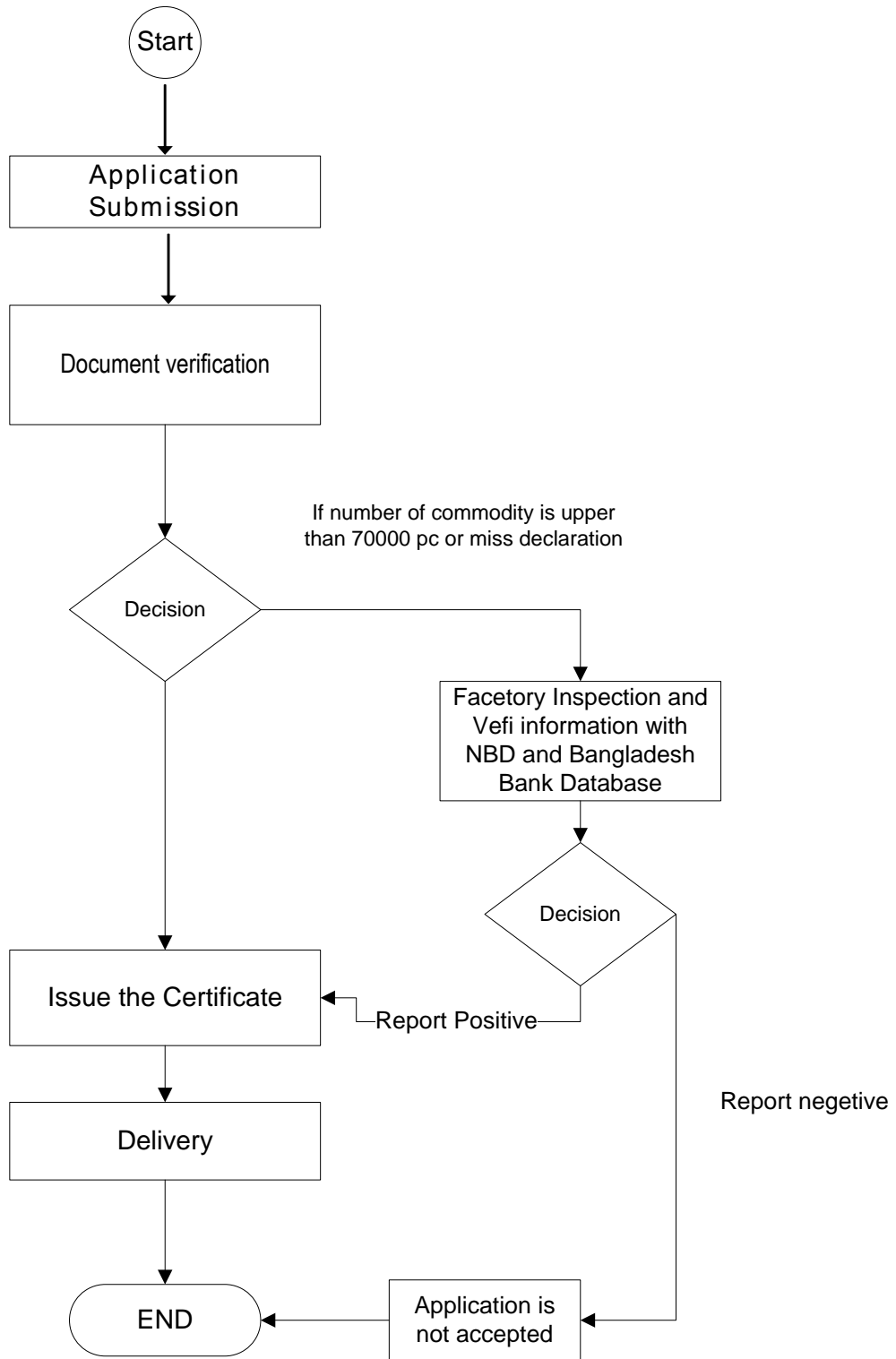


Fig-3.5: Work Flow Diagram of online GSP Application process

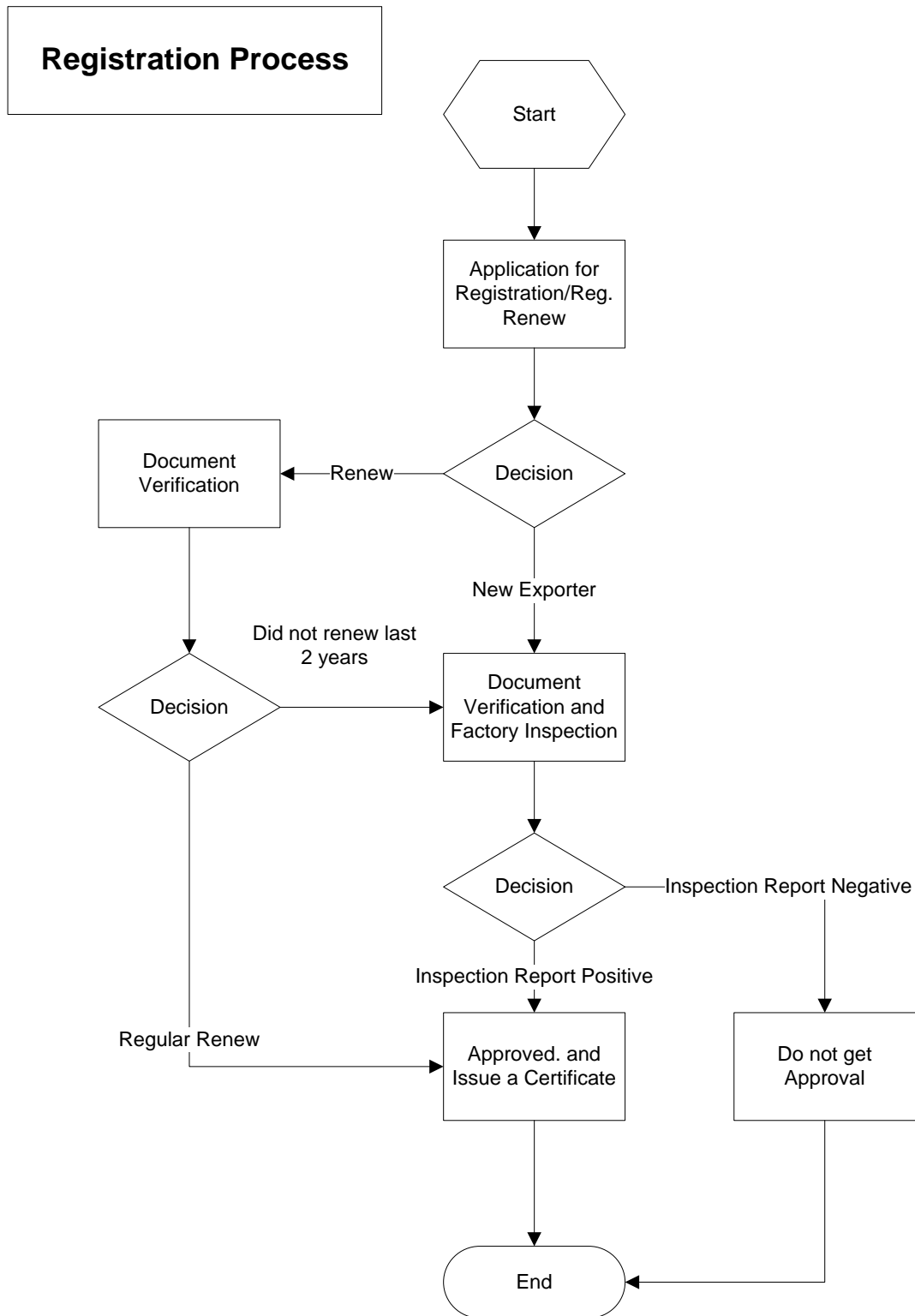


Fig-3.6: Work Flow Diagram of online Registration process

3.16 Entity Relationship Diagram

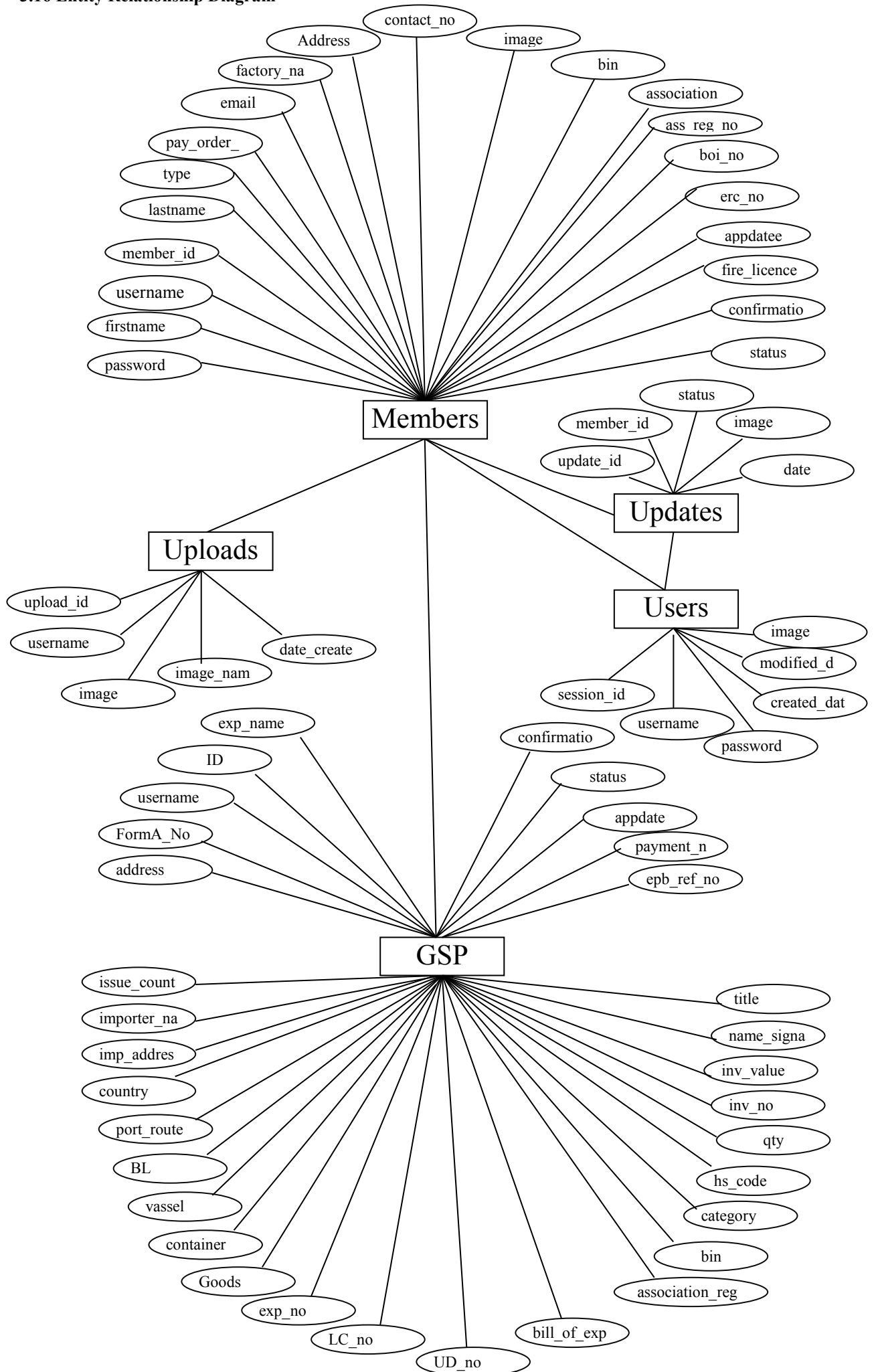


Fig: 3.6 E-R Diagram of Online Registration System for Textile Industries and GSP Application system

3.17 Relationship

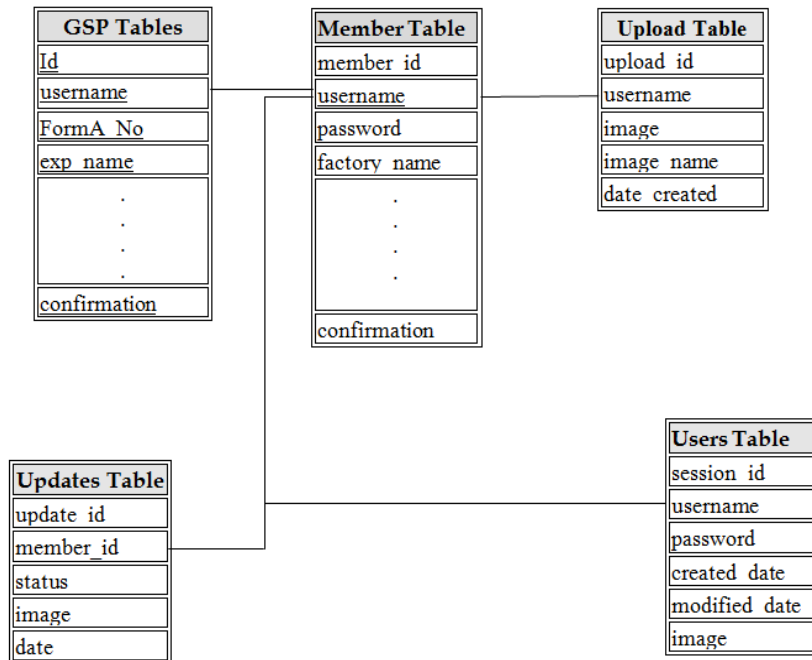


Figure-3.7: The relationship among the table.

3.18 Database Development

Relational database management system was used to manage the database. The database was developed in MySQL. The total data have been normalized and then tables are obtained.

3.19 Description of the Database Development

For simplicity, five tables have been designed namely members, GSP, updates, uploads and user. Brief description of the tables is given below:

- **Members Table:** This table contain twenty four columns (member_id, username, password, factory_name, firstname, lastname, address , contact_no, email, pay_order_no, association, ass_reg_no, bin, imagemedium, typevar, boi_no, fire_licence, erc_no, appday, appmonth, appyear, appdate , status and confirmation).

Sl	Field	Type	Sl	Field	Type
1	member_id	int(11)	13	Bin	varchar(20)
2	Username	varchar(20)	14	Image	mediumtext
3	Password	varchar(20)	15	Type	varchar(20)

4	factory_name	varchar(30)	16	boi_no	varchar(20)
5	Firstname	varchar(30)	17	fire_licence	varchar(20)
6	Lastname	varchar(30)	18	erc_no	varchar(20)
7	Address	varchar(200)	19	Appday	varchar(20)
8	contact_no	varchar(30)	20	Appmonth	varchar(20)
9	Email	varchar(30)	21	Appyear	varchar(20)
10	pay_order_no	varchar(20)	22	Appdate	varchar(20)
11	association	varchar(20)	23	Status	varchar(2)
12	ass_reg_no	varchar(20)	24	confirmation	varchar(2)

- **GSP Table:** It has thirty three columns (Id, username, FormA_No , exp_name, address, issue_country, importer_name , imp_address, country, port_route, BL, vassel, container, Goods , exp_no, LC_no, UD_no, bill_of_exp, association_reg_no, bin, category, hs_code, qty, inv_no, inv_value, name_signatory, title, epb_ref_no, payment_no, appdate, status, confirmation).

Sl	Field	Type	Sl	Field	Type
1	Id	int(10) unsigned	17	bill_of_exp	varchar(25)
2	username	varchar(15)	18	association_reg_no	varchar(25)
3	FormA_No	varchar(16)	19	bin	varchar(25)
4	exp_name	varchar(100)	20	category	varchar(25)
5	address	varchar(200)	21	hs_code	varchar(25)
6	issue_country	varchar(50)	22	qty	varchar(25)
7	importer_name	varchar(50)	23	inv_no	varchar(25)
8	country	varchar(200)	24	inv_value	varchar(25)
9	port_route	varchar(20)	25	name_signatory	varchar(25)
10	BL	varchar(100)	26	title	varchar(25)
11	vassel	varchar(50)	27	epb_ref_no	varchar(25)
12	container	varchar(50)	28	payment_no	varchar(25)
13	Goods	varchar(150)	29	payment_no	varchar(25)
14	exp_no	varchar(25)	30	appdate	date
15	LC_no	varchar(25)	31	status	varchar(25)
16	UD_no	varchar(25)	32	confirmation	varchar(2)

- **Upload Table:** Upload table has five columns (upload_id, username, image, image_name, date_created).

SI	Field	Type	SI	Field	Type
1	upload_id	int(11)	4	image_name	varchar(20)
2	username	varchar(20)	5	date_created	varchar(20)
3	image	mediumtext			

➤ **Updates Table:** It has five columns(update_id, member_id, status, image, date).

SI	Field	Type	SI	Field	Type
1	update_id	int(11)	4	image	mediumtext
2	Member_id	int(11)	5	date	varchar(20)
3	Status	text			

➤ **Users Table:** It has six columns (session_id, username, password, created_date, modified_date, image).

SI	Field	Type	SI	Field	Type
1	session_id	int(11)	4	created_date	timestamp
2	username	varchar(255)	5	modified_date	timestamp
3	password	varchar(255)		image	varchar(255)

3.20 Testing

Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user. It tells about the performance of the system. It is also very important feature of any data entry system. Testing is done to find the errors or any kind of invalid input. So, it also involves evaluating the result. This stage is carried out to find out if the new system is working according to the user requirements and any kinds of improvements are necessary or not.

3.21 Steps Performed for Testing

The testing has been performed by the following steps:

- Application's higher-risk aspects have been identified, priorities are set, and scope and limitations of tests are determined.
- Test approaches and methods - unit, integration, functional, system, load, usability tests, etc. Have been determined

- Input test data domain has been identified.
- Test plan document is defined and needed reviews/approvals are achieved.
- Specific Test cases are written.
- Reviews/inspections/approvals of test cases are achieved.
- Tests are conducted according to specification.
- Evaluation is done and reports are submitted.

3.22 Testing Classification

Testing of dynamic web application can be broken down into three types of activities. These are Unit testing, System testing and Acceptance testing. In this project, Unit testing and System testing were done.

3. 22.1. Unit Tests

Unit testing is the most 'micro' scale of testing; to test particular functions or code modules, typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. In unit testing each program is separately tested. It focuses on the modules to locate errors. If any error found, the module is revised and possible cause of error is defined and corrected. The thing that is considered to do the unit testing are narrated below-

- The forms are tested to ensure that information flows into and out properly.
- Data are examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution.
- Boundary conditions are tested to ensure that the forms operate properly at boundaries established to limit or restrict processing.
- Finally, all error handling paths are tested.
- Testing was done with wrong password; mandatory fields are filled up
- Or not and also invalid inputs.

3. 22.2 Acceptance testing

Acceptance testing means the final certification that the system is ready to be used in production setting. It is not done yet because this software is only for one part of the total business process. After completing and integrating all the modules, the entire software will be installed in all the machines and then the users will test to find out the lacking of the computerized system. In testing part tester must input data as an end user. Also the software must give some kind of warnings or error messages to warn or instruct the end users.

3. 22.3 Integration Testing

These tests are usually carried out to check whether a number of modules are working together appropriately. There are a number of forms attached to the Resource Allocation Process. It contains several forms together and each form is working parallelly with one another. In other processes, the scenario is the same case like before.

3.23. Sample Data used for the Development & Testing

Member Table:

Member_id	Username	Password	Factory_name	firstname	status
162	BD44234	*****	J&L Ltd	Jahangir	1
160	BD23232	*****	AJI Apparels	Uzzal	0

GSP Table:

Id	username	formA_No	exp_name	address	status
162	BD44234	A332493	J&L Ltd	Sharifpur, Gazaipur	1
160	BD23232	A332428	AJI Apparels	Asulia, Savar	0

Update Table:

Update_id	Member_id	Status	image	Date
36	86	changed profile picture.	uploads/Photo0065.jpg	05/15/2015
37	90	changed profile picture.	uploads/uzzal_4.jpg	04/11/2015

Uploads Load:

Upload_id	Username	image	Image_name	Date_created
32	BD44234	uploaded_photos/Photo0065.jpg	BGMEA Certificate	05/15/2015
37	BD23232	uploaded_photos/Photo0066.jpg	BGMEA Certificate	04/11/2015

Users Tables:

Session_id	Username	Password	Created_date	Modified_date	Image
163	BD44234	*****	05/15/2015	05/15/2015	uploaded_photos/uzzal.jpg
177	BD23232	*****	04/11/2015	05/15/2015	uploaded_photos/image001.jpg

3.24 Chapter Summery

This chapter described the whole process of the system. I have tried to make this process understandable. Here I have elaborated step by step analysis and designing technique of this system. This chapter has also focused on the software testing process and the relationship between the table and draw ER diagram with data flow diagram of this software.

4.1 What is System Design

System design known as the logical design of the software. Software design is a process through which requirements are translated into a “blueprint” for developing software. Initially the blueprint depicts a holistic view of software. That is, the design which is represented at a high level of abstraction. The level that can be directly traced to the specific system objective, detailed data, functional and behavioral requirements.

4.2 Basic Design of Software

The Automation Application administrator can manipulate data with the help of this software. It assists administrator to keep information of the Exporter and GSP related record with more comfortably, safely and securely. There are two interfaces of the Stake holder Relationship Management System which are described below:

- **Administrator Interface:** The software is connected with company server database, thus no more connection with other systems is needed. No system interface is needed during the development of this project.
- **User Interface:** The software must be designed as a web based that has a main user interface. Format of main screen must be standard and flexible. The system must be user friendly. Pages must be connected to each other in a consistent way. Operations to be done with the system being repeatable.

4.2.1 Hardware Requirement

Hosting Server: Midrange server with Windows or Linux Operating System.

Workstation : A general configured personal computer (PC) which is able to use internet browsing.

4.2.2 Programming Language and Software Platform

Programming language is very much important because it helps to Design interface and runs the application smoothly. In addition, A programming language provides a

structured mechanism for defining pieces of data, and the operations or transformations that may be carried out automatically on that data:

Software Interfaces:

Name & Version	Source	Purpose	Definition of the Interface
Firefox(18.0 or Later), Chrome 37.0.2062.120 and Microsoft Internet Explorer(IE) 8 or Later	Mozilla, Google and Microsoft Corporation	The web browser specified above is required in order to execute the user side of the software.	Firefox is one of the world's most popular browsers. It includes a variety of features that are designed to give you a fast and secure browsing experience. The Microsoft Internet Explorer provides easier, faster, safer, flexible and reliable browsing experience with enhanced web privacy features for all users. Google Chrome is a freeware web browser developed by Google. It used the WebKit layout engine until version 27 and, with the exception of its iOS releases, from version 28 and beyond uses the WebKit fork Blink.
Apache HTTP Server-2.2.14	The Apache Software Foundation	The web server specified above is required as the provider of the client software at the server site.	The Apache HTTP server project is a combined software development effort aimed at creating a strong, commercial-grade, full-feature, and freely-available source code implementable with operating systems including UNIX and Windows NT.
PHP-5.3.1	PHP Group.	PHP is used for server-	PHP is a mostly-used general-

		side web development PHP generally runs on a web server which work with MySQL database and Apache server.	purpose scripting language that has improved object-oriented capabilities especially suited for web development and can be embedded into HTML
MySQL-5.1.41	MySQL.	Required as relational database server.	MySQL is the world's most popular and powerful open source relational database software, with over 100 million copies of its software downloaded or distributed throughout its history. With flexibility, superior speed, reliability, and ease of use, MySQL has become the preferred choice of corporate IT managers because it eliminates the major problems associated with downtime, maintenance, administration and support.
Macromedia Dreamweaver MX-8	Macromedia Inc.	The web development tool specified above is required for designing and coding of the software.	Macromedia Dreamweaver is professional HTML editor tool, enabling users to efficiently design, coding, develop and maintain standards websites and applications.

4.2.3. Communication Interfaces

The default communication protocol for data transmission between server and the client is Transmission Control Protocol/ Internet Protocol (TCP/IP). At the upper level

hypertext transfer protocol (HTTP) will be used for communication between the web server and client.

Memory Constrain

There is not a specific memory constraint for this software.

4.2.4 Software System Attributes

a) Reliability

Reliability engineering is an engineering that emphasizes dependability in the lifecycle management of a product. Dependability, or reliability, describes the ability of a system or component to function under stated conditions for a specific period of time. Reliability engineering represents a sub-discipline within systems engineering. Reliability is theoretically defined as the probability of success ($\text{Reliability} = 1 - \text{Probability of Failure}$), as the frequency of failures, or in terms of availability, as a probability derived from reliability and maintainability. Maintainability and maintenance are often defined as a part of "reliability engineering" in Reliability Programs. Reliability plays a key role in the cost-effectiveness of systems. The software must operate 100% of the time.

b) Security

The authorization mechanism of the system will block the unwanted attempts to the server and also let the system decide which privileges the user. The system has different types of users so there are different levels of authorization. There will be also a firewall installed on the server so the incoming transactions can be filtered. Data integrity for critical variables will also be checked.

c) Maintainability

The requirements, modules that are explained in this document are enough to satisfy the project goal. So, the maintainability shall be easily done by the admin of the system.

d) Portability

This Software is an online service. So, anyone can use the service. Only the server of the system must have the required software including MySQL, Apache. The interface is designed with the help of PHP as it provides easy abstract window toolkit and used widely. PHP is an extremely rich programming language and it contains the basic components of developing user's interface. It is an integrated development

environment in which we can develop, run, test and debug our application. The programmer can easily use the buildup tool kits for any application software. It has evolved into a major development environment that covers every aspects of programming, from education applications to database and from financial applications to Internet components. In future, the platform independent Language Java can be used.

4.3 Architecture Overview

The basic functionality of the software involves the On-Line registration system for textile Industries and GSP application over the Internet at a minimal cost. The fundamental requirement of the project is a web application built in Apache, MySQL and PHP. The basic architecture of web application is described below:

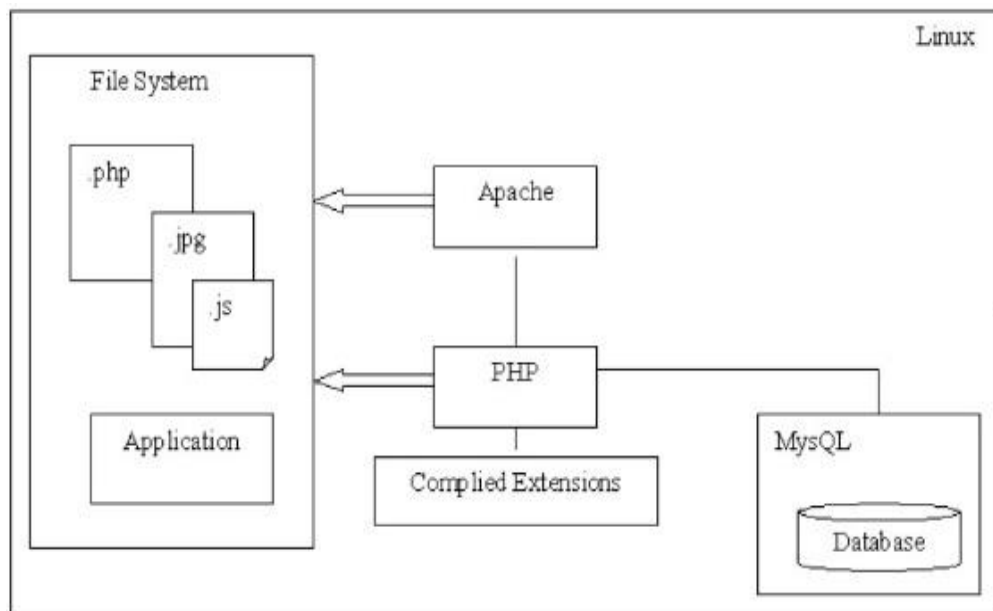


Fig. 4.1: Architecture of web application

4.3.1 The Client Side

A client, i.e. the computer, laptop, mobile etc which requests the resources, through the internet with a user interface (typically a web browser) for presentation purposes.

4.3.2. The Admin Side

Admin is also a client, i.e. the computer, laptop, mobile etc. which requests the resources, through the internet with a user interface (typically a web browser) for creating, updating and deleting information.

4.3.3. The Web Server

Almost all of the works of web application take place on the server. A specific application, called a web server, is responsible for authentication, authorization and secure communication channel with the browser. A relational-database server stores whatever information the application requires.

4.3.4 The Application Server/Middleware

The application server is also called middleware, whose assignment it is to provide the requested resources, but by calling on another server. PHP belongs to a class of languages known as middleware. These languages work closely with the Web server to interpret the requests made from the World Wide Web (WWW), process these requests, interact with other programs on the server to fulfill the requests, and then pass the web server exactly what to serve to the client's browser.

4.3.5. Functional Requirement Analysis

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Functional recruitment identified as shown below. They are clustered into the following categories:

a) Network specific requirements

SL	Title	Description
01.	Server	System requires a streaming server

b) File and database management requirements

SL	Title	Description
01.	System allows for creation of profiles	User can view their demographic data and edit their personal information. Admin can modify

		customer database and the privilege to user access.
02.	User Authentication	User has the unique user ID and Password to authenticate their Connection.
03.	System file	System contains many file for storing information.
04.	Image file	Files can store image (e.g. .jpeg, .gif, etc) content on the user side and admin can view the image file.
05.	Save data	Stored information is clustered to categories
06.	Data available	Receive content from fixed/wireless public networks.

C) Query and retrieval requirements

SL	Title	Description
01.	User identification	System can select user (Stake Holder, administrator)
02.	View user info	User can view his/her personal info, Previous GSP Record .
03.	Submit GSP Application	User can apply for getting GSP in his own account.
04.	Enable systems	Systems are enabling for browsing and navigate.

D) User Interface requirements

SL	Title	Description
01.	Platform allows	Platform allows users for the admin and the users.
02.	Runs on	Use Interface runs on pc, mobile devices
03.	Provide data	User can provide GSP Certificate data. Admin can edit any data.
04.	Update data	admin can update their provide data.

E) Security requirements

SL	Title	Description
01.	Authentication mechanism	System requires authentication mechanisms for user identification.
02.	User category	Users are classified in three categories (Bonded, Non Bonded and Temporary).
03.	Access information	Users can access to specific content based on their username and password.
04.	Encryption	System uses data encryption

4.3.6. Other Non-Functional Requirement Analysis

The restrictions on the types of solutions that will meet the functional requirements.

This section presents the nonfunctional requirements, which considered during the development of the web based Online Registration System for Textile Industries and GSP application. This summarization is given below:

SL	Title	Description
01.	Performance	This requirement has to do with QoS characteristics, such as high speed internet availability for data intensive transmissions. It also concerns to the time required for performing the operations allowed by the system.
02.	Scalability/ Expandability	The system should be able to scale and expand the dispatch center to be able to handle more traffic. The system's performance attributes should be maintained independent of the number of nodes or documents. A dramatic increase in the number of nodes or documents will have minimal effect on performance and availability.
03.	Availability	Ensures that authorized users have always access to data and associated assets 24/7 with 100% reliability.

04.	Robustness, Fault Detection and Recovery	It should be ensured that content and content delivery services are available at any time even if some hardware or software components fail to function
05.	Maintainability	If a service has been upgraded, then the old version of the service must be available for some time in order for the IT personnel to upgrade the software to use the new version of the service by the IT personnel.
06.	Usability	Easy to use

4.4. Collection and update Process of Basic Information

Working with step by step data manipulation is known as working procedure. There are some criteria to storing and updating of exporter profile, GSP information by Online registration and GSP automation system. The working process is described below:

4.4.1. On-line Registration

As Exporter applies for online registration with all information, he/she has to maintain some official procedure. The exporter has to fill up Exporter information fields(name, address, ERC No, Fire license, Association, BIN Number etc.) during registration. Exporter also includes some other documents such as Photo copy of Trade Licenses, Fire Licenses, Association Membership, Export Registration Certificate, Fire insurance, Group Insurance, Business Identification Number Bank, Pay slip. A sample copy of Exporter Registration Form (ERF) is given later.

After successful completion of Registration form submission, the administrator verify the information (Provide information is correct or wrong). If all information which are provided from the exporter is correct the administration approved his registration. In the meantime the exporter can track his registration process by using limited user access in applicant login page. If the exporter gets approval he will be the permanent member of Export Promotion Bureau.

Sign Up

Please Note down(PIN Number and Password)

PIN Code:

Password:

Factory name:

Factory address:

Exporter Firstname:

Exporter Lastname:

Email:

Cell Number:

Pay Order No:

BoI Number:

Association:

Association Reg No:

ERC No:

Fire licence:

BIN:

Type:

Date:
Month: Day: Year:

Terms of Use

Sign Up

Fig-4.2: Registration Form (ERF)

4.5 GSP Service

GSP means, Generalized System of Preference, which is issued by Export Promotion Bureau. In order to get certificate of origin (GSP), before physical movement of goods, you need to apply to export Promotion Bureau as per their specified request form along with copy of commercial invoice duly signed by the authorized signatory. After verifying properly and collecting necessary charges, export Promotion Bureau certifies the document and issues GSP Certificate of origin along with attested copy of commercial invoices.

EPB provide four types of service about GSP related issue. They are 1. General Issue of GSP certificate 2. Re-Issue GSP Certificate 3. Duplicate Issue GSP certificate and 4. Cancel GSP Certificate. All types of service exporter get from Export Promotion Bureau according to the payment rules of the company.

- **GSP Certificate General Issue:** A valid exporter can apply for GSP. If all required information is available, Export Promotion Bureau issue GSP Certificate for the exporter.
- **Re-Issue GSP Certificate:** A valid exporter submit a application for getting GSP certificate from EPB. But if the exporter does typing error or miss declaration accidentally on Form-A. Those time the exporter has to submit his application again after purifying all export related information.
- **Duplicate Issue GSP certificate:** If the issued GSP certificate is lost or destroyed forever, then EPB issue a duplicate certificate against the valid consignment.
- **Cancel GSP Certificate:** Some time Exporter put wrong information and issued certificate from EPB. Lacks of proper information as a result the certificates cannot fill up buyers requirement. For this reason the certificate has to be cancelled and re-issued as per product real description.

Sample Form-A

1. Goods consigned from (exporter's, business name, address, country)			Reference No. A 6316129 GENERALIZED SYSTEM OF PREFERENCE CERTIFICATION OF ORIGIN (Combined declaration and certificate) Form A Issued in..... (country) Se notes overleaf		
2. Goods consigned to (consignee's name, addresses, country)					
3. Means of transport and route (as far as known)			4. For official use		
5. Item num- ber	6. Marks and numbers of package	7. Number and kind of packages; description of goods	8. Origin criterion (see notes overleaf)	9. Gross weight or others quantity	10. Number and date of invoice
<p>11. Certification</p> <p>It is hereby certified, on the basis of control carried out, that</p> <p style="padding-left: 40px;">the declaration by the exporter is correct</p> <p>.....</p> <p>Place and date, signature and stamp of certifying authority</p>			<p>12. Declaration by the exporter</p> <p>The undersigned hereby declares that the above details and statements are correct; that all the goods were</p> <p style="text-align: right;">(country)</p> <p style="padding-left: 40px;">produced in</p> <p>and that they comply with the origin requirements specified for those goods in the generalized system of preferences for goods exported to</p> <p style="text-align: right;">..... (Importing country)</p> <p>.....</p> <p>Place and date, signature of authorized signatory</p>		

4.6.0. Basic View of Administrator Interface

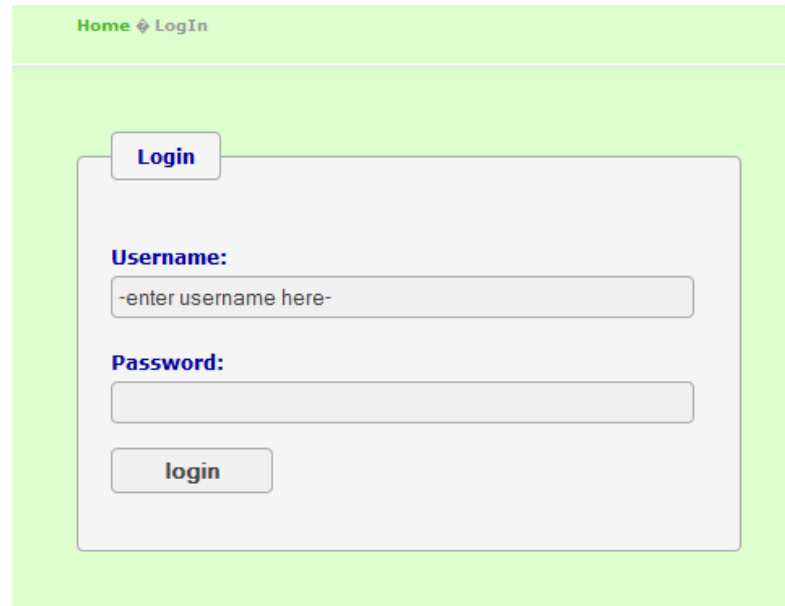


Fig-4.3: Administrator Login Page.

4.6.1. Function of the Administrator Interface

Administrator interface is graphically designed for the administrators to manipulate their stakeholders' information. Graphically the interface consists of four parts. Those are described below:

- * **Top Bar:** Top bar graphically contains the automation software platform banner.
- * **Right Bar:** Right bar contains the software main menu.
- * **Bottom Bar:** Bottom bar contains the software copy right message.

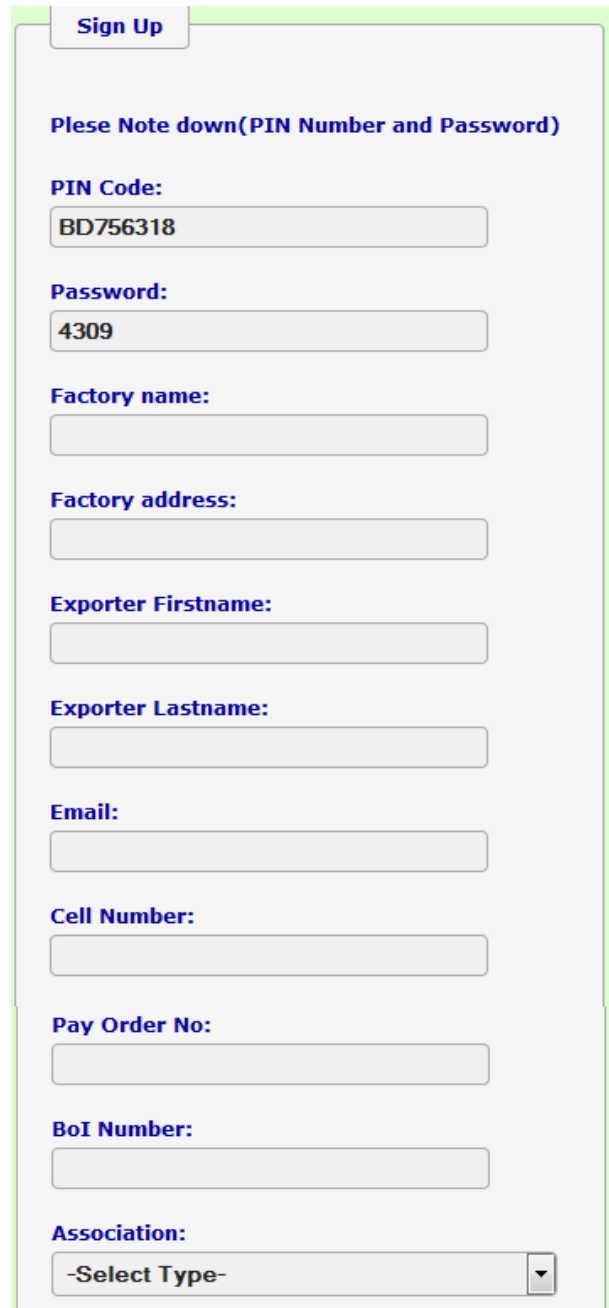


Fig-4.4: Software Admin Home Page.

First of all, administrator needs user name and password to login to the software. After that, he or she may be able to approve users who apply for registration, store and update of their stakeholders' profile and GSP Application form.

4.6.2 Data Insertion Process

Apply for Registration:



The image shows a web form titled "Sign Up" with a light green border. The form contains the following fields and labels:

- Sign Up** (tab)
- Please Note down(PIN Number and Password)** (instruction)
- PIN Code:**
- Password:**
- Factory name:**
- Factory address:**
- Exporter Firstname:**
- Exporter Lastname:**
- Email:**
- Cell Number:**
- Pay Order No:**
- BoI Number:**
- Association:**

Fig-4.5: Data Insert Process (Apply for Reg.).

Apply for GSP Certificate:

Application Submission Form	
User name	BD00090
Form A No	
1. Goods Consigned From	(Exporter's business name, address, country)
Exporter Name	Hamim Pvt Ltd
Exporter Address	Ashulia, Savar Dhaka
Issue Country	Bangladesh
2. Goods Consigned To	(Consignee's name, address, country)
Importer Name	
Importer Address	
Importer Country	-Select Type- ▼
3. Transport and Goods description	
Loading Port& Route	
Bill of Lading(B/L)	
Vassal	
Container	
Goods	
Exp No	
LC No	
UD No	
Bill of Exp	
Assiciation Reg. No	23424
Bin No	23423424242
Category No	
HS Code	
Quantity	
Invoice No	
Invoice Value	
Date:	
<input type="button" value="Submit"/>	

Fig-4.6: Data Insert Process(Apply for GSP).

Data Insert Steps:-

1. Select an item from Bar menu.
2. Click on insert from the top bar and fill up the required field.
3. Finally press the submit button.

4.6.3. Data Viewing and Updating Process

Main Menu

[Issued GSP](#)

SL	User Name.	Exporter Name	Form-A No.	Payment No	Status	View details
24	BD00090	Mr Hamim	A33423234	BB34566	1	Click here
25	BD04590	Mr Hamim	A33473890	BB36796	1	Click here
27	BD00090	Mr Hamim	A33273324	BB25389	1	Click here
29	BD00090	Rony Tex	A42234573	BB23188	1	Click here
30	BD00090	Rony Tex	A42234356	BB44346	1	Click here
38	BD68791	Tomi Knit Fashion	A45387964	BB23451	1	Click here

1 Total Number 6

Fig-4.7: Data View and Update Process.

Data View and update Steps:-

1. Select an item from right menu.
2. Click on view from the left bar.
3. Then click on to the edit button and correct the required field.
4. Finally press the update button.

4.6.4. Basic View of User Interface

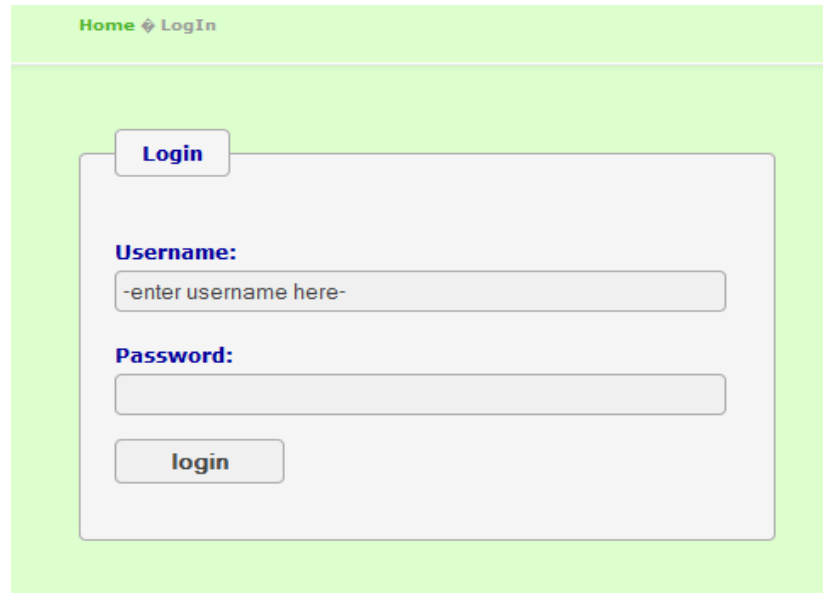


Fig-4.8: User Login Page.

User interface is mainly designed for the exporter's to authenticate their connection to the network. To login exporter needs user name and password which is Provided by the service providing organization. The service providing organization generates user id and password for every individual exporter.

4.6.5. Function of the User Interface



Fig-4.9: User Interface Home Page.

After successful login in a menu appear in the user interface. The menu contains seven topics items which are known as Home, Profile Picture, Info, Certificate, Apply for GSP, View Application, logout. Through which a customer provides their personal information by filling up the registration form, he or she can also view their profile

with the help of this interface, but the users cannot update their information because of some official policy. There are some other functions like update profile picture, upload certificate scanned copy and basic information of the users. Exporter can easily submit the GSP application of their account through this user interface. Changing user's password is also possible with the help of this user interface.

4.7. Chapter Summery

This chapter has described the features and the use of the software. Here, I have tried to explain the basic idea of GSP Form A and information required for Registration.

5.1. Conclusion

The European Union (EU) has been providing GSP facilities to the developing and Least Developed Countries (LDCs) since 1971. GSP policies have been changing according to the political will of the EU countries and the policies are fixed unilaterally depending on the strategies of the concerned countries. The Ready-Made Garments (RMG) sector has emerged as the biggest earner of foreign currency. The sector contributes significantly to the GDP. This case study has conducted a detailed work for developing and integrated and interconnected “GSP online” system based on a detailed review of steps and processes for the existing form A filing and a substantial leap forward toward efficiency and process improvement to meet the requirements. As a consequence on-line registration for textile Garments and GSP application submission system is a database system that makes a bridge between the RMG Exporter and the service providing organization. So to manage a large scale of customer (RMG exporter), it requires a complete database. In this case study, the database is designed powerfully. A web based application that can be used for both Exporter and the service providing organizations to view and update information by using user interface and administrator interface. In this application, We have tried to make sure the GSP issuing process and Exporter Registration process functioning smoothly. The service providing organization is to store REX(Registered Exporter) database and GSP certificate related information to solve their daily problems which is perfectly organized in this database project.

5.2. Recommendation

Only Export Promotion Bureau can use this software with little care. If he/she needs any modification or any other facilities he/she is most welcome to suggest so. Power failure is now very serious problem in our country. It can happen at any time. So a backup power system should be used for this purpose. The other recommendations on this software are:

1. Java programming language can be used instead of PHP programming due to its platform independency.
2. In this software there is no way to reset password at the user end. So Password recovery can be added for better performance.
- 3 This software is only tested for some medium scale. If anyone wants to implement it in a large scale organization, Oracle can be used instead of MySQL for better performance.

5.3. Future Work

Future work related to the present development can be carried out. A few area of the future research is outlined below:

- Security matters certainly constitute a continuing problem for electronic merchants. Many Internet users and even security experts are concerned enough about Internet crime and potential violations of personal privacy. SSL security may be implemented in different modules of the developed software.
- Some organizations are also being selective about how they use the Internet to communicate with their users e.g. making personal contact by phone after a certain stage in the process even as still using their web based Online Registration System for Textile Industries and GSP Application.

References

- [1] Official Journal of the European Union[Commission Regulation[EU] No 1063/2010 of 18 November 2010 amending Regulation(EEC);
- [2] GSP-Handbook on the scheme of the European Union; Publication-2010;
- [3] Tariff Preference: new GSP Rules of Origin; Publication-2011;
- [4] Generalized System of Preferences for LDC countries; Publication-2007 ;
- [5] Rules of origin in GSP schemes and their impact; Publication-2007;
- [6] Al-Fedaghi S., “Developing web applications”, International journal of software engineering and its applications vol. 5 no. 2, April, 2011.
- [7] Farooq A. and Arshad M. J., “A process model for developing semantic web systems”, New York science journal 2010;3(9).
- [8] <http://httpd.apache.org/dev/guidelines.html>, last accessed on 21.02.2013
- [9] <http://www.pjweb.co.uk/blog/2012/10/08/benefits-of-a-web-based-system/>
- [10] <http://www.w3schools.com>, last accessed on 20.02.2013

Appendix

The PHP Source Codes of data insertion are given below:

```
<?php
    session_start();
    function random_string($length){
    $charset = array_merge(range(0, 9));
    shuffle($charset);
    $rand = array_slice($charset, 0, $length);
    return implode ("",$rand);
    }?>
<?php ob_start(); ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<script>
function verifyEmail(){
var emailRegEx = /^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$/i;
    if (document.alokm.email.value.search(emailRegEx) == -1) {
        alert("Please enter a valid email address.");
    }
    return false;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <link rel="stylesheet" type="text/css" href="css/pswd.css" />
    <title>E&G</title>
    <link rel="shortcut icon" HREF="images/aaa.png" />
```

```
<meta content="Metadescription" name="Insert the description of this
page here" />
```

```
<meta content="MetaKeyWords" name="Insert the keywords that describe this
page here" />
```

```
<script type="text/javascript" SRC="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript"
SRC="js/superfish/hoverIntent.js"></script>
```

```
<script type="text/javascript" SRC="js/superfish/superfish.js"></script>
<script type="text/javascript" SRC="js/superfish/supersubs.js"></script>
<script SRC="js/cufon-yui.js" type="text/javascript"></script>
<script SRC="js/Liberation_Sans_font.js"
type="text/javascript"></script>
```

```
<script type="text/javascript">
    Cufon.replace('h1,h2,h3,h4,h5,h6');
    Cufon.replace('.logo', { color: '-linear-gradient(0.5=#FFF, 0.7=#DDD)'
});
</script>
```

```
<SCRIPT type="text/javascript">
pic1 = new Image(16, 16);
pic1.src = "loader.gif";
$(document).ready(function() {
$("#username").change(function() {
var usr = $("#username").val();
if(usr.length >= 4)
{
$("#status").html('&nbsp;Checking
availability...');
$.ajax({
type: "POST",
url: "check.php",
data: "username="+ usr,
success: function(msg){
```

```

$("#status").ajaxComplete(function(event, request, settings){
    if(msg == 'OK')
    {
        $("#username").removeClass('object_error');
        $("#username").addClass("object_ok");
        $(this).html('&nbsp;');
    }
    else
    {
        $("#username").removeClass('object_ok');
        $("#username").addClass("object_error");
        $(this).html(msg);
    }
});
}   }); }
else {
    $("#status").html('<font color="red">The username should have at least
<strong>4</strong> characters.</font>');
    $("#username").removeClass('object_ok');
    $("#username").addClass("object_error");    });});
//-->
</SCRIPT>
<style type="text/css">
<!--
.style1 {font-size: 110%}
-->
</style>
</head>
<body>
<div id="header">
<div class="container_12">
<div class="grid_3">

```

<h6 class="logo style1"> ONLINE
REGISTRATION SYSTEM FOR TEXTILE INDUSTRIES AND GSP

APPLICATION

 </h6>

</div>

<!-- end grid -->

<div class="grid_9"> <ul class="sf-menu"> Home

 LogIn Sign
Up

 Contact

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

<div id="breadcrumb"> <div class="container_12 clearfix"><div class="grid_9">

<p>Home ♦ Sign Up</p>

<p>

<?php

include('session/DBConnection.php');

\$test = \$_POST['username'];

if (isset(\$_POST['register'])) {

if (

!\$_POST['username']|

!\$_POST['password']|

!\$_POST['firstname']|

!\$_POST['lastname']|

!\$_POST['email']|

!\$_POST['address']|

!\$_POST['type']|

!\$_POST['checkbox']){

die('Please complete all the required feilds!'); }

\$_POST['password'] = (\$_POST['password']);


```

        $_POST['appdate'] = ($_POST['appmonth'] . "/" . $_POST['appday'] . "/" .
$_POST['appyear'] ); $_POST['appmonth'] = ($_POST['appmonth']);
    $_POST['appday']     =     ($_POST['appday']);         $_POST['appyear']     =
($_POST['appyear']);
        $_POST['factory_name'] = ($_POST['factory_name']);
        $_POST['address'] = ($_POST['address']);
        $_POST['contact_no'] = ($_POST['contact_no']);
        $_POST['email'] = ($_POST['email']);
        $_POST['pay_order_no'] = ($_POST['pay_order_no']);
        $_POST['association'] = ($_POST['association']);
        $_POST['ass_reg_no'] = ($_POST['ass_reg_no']);
        $_POST['boi_no'] = ($_POST['boi_no']);
        $_POST['erc_no'] = ($_POST['erc_no']);
        $_POST['fire_licence'] = ($_POST['fire_licence']);
        $_POST['bin'] = ($_POST['bin']);
        $_POST['type'] = ($_POST['type']);
        if (!get_magic_quotes_gpc()) {
            $_POST['password'] = addslashes($_POST['password']);
            $_POST[""] = addslashes($_POST['username']);
        }
$insert = "INSERT INTO members SET
username='$_POST[username]',
password='$_POST[password]',
firstname='$_POST[firstname]',
lastname='$_POST[lastname]',
email='$_POST[email]',
contact_no='$_POST[contact_no] ',
address='$_POST[address]',
image='$_POST[image]',
type='$_POST[type]',
factory_name = '$_POST[factory_name]',
pay_order_no = '$_POST[pay_order_no]',
association = '$_POST[association]',

```



```
id="status"></span></label>                                </p>
    <p><label>Password:
        <input id="password" style="width: 250px; font-weight: bold; color:
#222;
text-transform: none;" readonly="Yes" type="text" name="password" value = "<?php
echo random_string(4);?>"><br /><span id="status"></span></label></p>
    <p><label>Factory name:
        <input style="width: 250px; font-weight: bold; color: #222;
text-transform: capitalize;" type="text" name="factory_name" /></label>
    </p><p>        <label>Factory address:
        <input style="width: 250px; font-weight: bold; color: #222;
text-transform: capitalize;" type="text" name="address" /></label>
    </p>
    <p>
        <label>Exporter Firstname:
        <input style="width: 250px; font-
weight: bold; color: #222;
text-transform: capitalize;" type="text" name="firstname" /></label>
    </p><p>
        <label>Exporter Lastname:
        <input style="width: 250px; font-
weight: bold; color: #222;
text-transform: capitalize;" type="text" name="lastname" /></label>
    </p>
    <p>
        <label>Email:<br />
        <input type="text" style="width:
250px; font-weight: bold; color: #222; text-transform: none;" onblur="verifyEmail()"
name="email" /></label>    </p><p><label>Cell Number:
        <input style="width: 250px; font-weight: bold; color: #222;
text-transform: capitalize;" type="text" name="contact_no" /></label>
    </p>
    <p>        <label>Pay Order No:<br>
```

```
weight: bold; color: #222; text-transform: capitalize;" type="text"
name="pay_order_no" /></label>
```

```
</p><p><label>BoI Number:<br>
<input style="width: 250px; font-
weight: bold; color: #222;
text-transform: capitalize;" type="text" name="boi_no" /></label>
```

```
</p><p>
<label></label>
</p>
```

```
<label>Association:
<select name="association"
style="width: 300px; font-weight: bold; color: #222;
text-transform: none;">
```

```
<option value="-1" selected="selected">-Select Type-</option>
<option value="BGMEA">BGMEA</option>
<option value="BKMEA">BKMEA</option>
```

```
<option
value="BTMA">BTMA</option>
<option
```

```
value="BTTLMA">BTTLMA</option>
</select></label> </p><p>
```

```
<label>Association Reg No:<br>
<input style="width: 250px; font-
weight: bold; color: #222; text-transform: capitalize;" type="text" name="ass_reg_no"
/></label>
```

```
</p><p><label>ERC No:<br>
<input style="width: 250px; font-
weight: bold; color: #222; text-transform: capitalize;" type="text" name="erc_no"
/></label>
```

```
</p><p><label>Fire licence:<br>
```

```
weight: bold; color: #222; text-transform: capitalize;" type="text"
name="fire_licence" /></label>
```

```
</p><p> <label>BIN:<br>
```

```
<input style="width: 250px; font-
weight: bold; color: #222; text-transform: capitalize;" type="text" name="bin"
/></label>
```

```
</p><label>Type:
```

```
<select name="type" style="width: 300px; font-weight: bold; color: #222;
text-transform: none;"> <option value="-1" selected="selected">-Select
Type-</option>
```

```
<option value="Bonded">Bonded</option>
```

```
<option value="Non Bonded">Non Bonded</option>
```

```
</select></label>
```

```
<p> <p> <label></label> </p>
```

```
<p>
```

```
<label>Date:<br>
```

```
<input name="appdate" type="hidden" size="25" />
```

```
<div id="appmonth"><select class=""
style="width: 100px;font-weight: normal;color: #222;" name="appmonth" ></div>
```

```
<span id="valmonth" style="display:none;">
```

```
<option value="-1">Month:</option>
```

```
<option value="January">Jan</option>
```

```
<option value="February">Feb</option>
```

```
<option value="March">Mar</option>
```

```
<option value="April">Apr</option>
```

```
<option value="May">May</option>
```

```
<option value="June">Jun</option>
```

```
<option value="July">Jul</option>
```

```
<option value="August">Aug</option>
```

```
<option value="September">Sep</option>
```

```
<option value="October">Oct</option>
```

```
<option value="November">Nov</option>
<option value="December">Dec</option>
</select></span>
<span id="appday">
  <select name="appday" style="width: 100px;
font-weight: normal;
color: #222;" autocomplete="on"></span> <span id="valday" style="display:none;">
  <option value="-1">Day:</option>
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
  <option value="6">6</option>
  <option value="7">7</option>
  <option value="8">8</option>
  <option value="9">9</option>
  <option value="10">10</option>
  <option value="11">11</option>
  <option value="12">12</option>
  <option value="13">13</option>
  <option value="14">14</option>
  <option value="15">15</option>
  <option value="16">16</option>
  <option value="17">17</option>
  <option value="18">18</option>
  <option value="19">19</option>
  <option value="20">20</option>
  <option value="21">21</option>
  <option value="22">22</option>
  <option value="23">23</option>
  <option value="24">24</option>
```

```

        <option value="25">25</option>
        <option value="26">26</option>
        <option value="27">27</option>
        <option value="28">28</option>
        <option value="29">29</option>
        <option value="30">30</option>
        <option value="31">31</option>
    </select></span>
    <span id="appyear"> <select name="appyear" style="width: 100px;
font-weight: normal;
color: #222;" autocomplete="on"></span><span id="valyear" style="display:none">
        <option value="-1">Year:</option>
                <option value="2010">2014</option>
                <option value="2011">2015</option>
                <option value="2003">2016</option>
                <option value="2004">2017</option>
        <option value="2005">2018</option>
        <option value="2006">2019</option>
        <option value="2007">2020</option>
    </select></span></p></label> </p>
    <p><input type="radio" name="checkbox" id="check"
checked/>
        <a href="termsfuse.php">Terms of Use</a><br /><br />
        <input class="button" value="Sign Up" type="submit"
name="register" />
    </p></form>
</fieldset>
</div><!-- end grid -->
</div><!-- end container -->
</div><!-- end content -->      <!-- end footer -->
    <div id="sub_footer"> <div class="container_12 clearfix">
<div class="grid_12"> <a class="logo left" href="#">E&G</a>

```

```

        <p><small>&copy;&nbsp; All rights reserved. | <a
href="copyright.php">Copyright</a> | <a href="terms.php">Terms and
Conditions</a> | <a href="privacy.php">Privacy Policy</a> | <a class="current"
href="about.php">About Us</a></small></p>
    </div> </div> </div><script type="text/javascript"> Cufon.now(); </script>
    <script type="text/javascript"> Cufon.now(); </script>
</body> </html><?php ob_flush(); ?>

```

The PHP Source Codes of data view are given below:

```

<table width="100%" border="0" background="products_b.gif">
<tr><th nowrap><h2 align="left"><a href=" ../memberlist.php"
class="style12">Main Menu</a></h2></th> </tr></table>
<?php session_start();$rowsPerPage = 20;
if(isset($_GET['page']))
{$pageNum = $_GET['page'];}
$offset = ($pageNum - 1) * $rowsPerPage;?>
<?php
include ("session/DBConnection.php");
$user = $_SESSION['log']['username'];
$query= "SELECT Id, username, FormA_No, exp_name, payment_no, status FROM
gsp where confirmation = '1' and status = '1' or status = '0'";
$result = mysql_query($query) or die(mysql_error()); ?>
<html>
<head>
<link rel="stylesheet" type="text/css" href="main.css" />
</head>
<td> <div align="center">
    <h1><a href=" ../memberlist.php" class="style12">Issued GSP </a></h1>
</div></td> <table width="100%"> <th colspan = '9'><div></div></th>
    <tr><td class="rowHeader1"><strong>SL</strong></td>
<td class="rowHeader1"><strong>User Name.</strong></td>
<td class="rowHeader1"><strong>Exporter Name</strong></td>

```



```

<td class="rowHeader1"><strong>Form-A No.</strong></td>
<td class="rowHeader1"><strong>Payment No</strong></td>
<td class="rowHeader1"><strong>Status</strong></td>
<td class="rowHeader1"><strong>View details</strong></td>
</tr><?php
while ($row = mysql_fetch_array($result)){?>
<tr><td><?php echo $row['Id'];?></td>
<td><?php echo $row['username'];?></td>
<td><?php echo ucwords(strtolower($row['exp_name']));?></td>
<td><?php echo $row['FormA_No'];?></td>
<td><?php echo $row['payment_no'];?></td>
<td><?php echo $row['status'];?></td>
<?php $exporter = $row['Id'];?>
<td><a href="all_gsp_edit.php?type=all_gsp_info&$exporter=<?php echo
$row['Id']?>">Click here</td></tr><?php
}?> <tr> <th colspan="9">&nbsp;</th>
</tr> </table><tr> <th colspan="9">&nbsp;</th></tr> </table><?php
$query = "select count(FormA_No) as numrows from gsp where confirmation =
'1'";
$result = mysql_query($query) or die('Error, query failed');
$row = mysql_fetch_array($result, MYSQL_ASSOC);
$numrows = $row['numrows'];
// how many pages we have when using paging?
$maxPage = ceil($numrows/$rowsPerPage);
// print the link to access each page
$self = $_SERVER['PHP_SELF'];
$nav = "";
for($page = 1; $page <= $maxPage; $page++)
{ if ($page == $pageNum) {
    $nav .= " $page "; // no need to create a link to current page
} else {
    $nav .= " <a href=\"$self?page=$page\">$page</a> ";
} } // the first and last page

```

```

if ($pageNum > 1) {
    $page = $pageNum - 1;
    $prev = " <a href=\"\$self?page=$page\">[Prev]</a> ";
    $first = " <a href=\"\$self?page=1\">[First Page]</a> ";
} else {    $prev = '&nbsp;'; // we're on page one, don't print previous link
    $first = '&nbsp;'; // nor the first page link }
if ($pageNum < $maxPage) {    $page = $pageNum + 1;
    $next = " <a href=\"\$self?page=$page\">[Next]</a> ";
    $last = " <a href=\"\$self?page=$maxPage\">[Last Page]</a> ";
} else    {
    $next = '&nbsp;'; // we're on the last page, don't print next link
    $last = '&nbsp;'; // nor the last page link
} echo "<div style={float:right}>"; echo "<b>Total Number $numrows</b>"; echo
"</div>";
// print the navigation link
echo $first . $prev . $next . $last . $nav ;
?>

```

The PHP Source Codes of data updates are given below:

```

<?php session_start();?>
<?php
include ("session/DBConnection.php");
$sexpo = $_SESSION['$sl'];
$query_gsp= mysql_query("SELECT * FROM gsp WHERE Id = '$sexpo' and
confirmation ='0'") or die (mysql_error());
        $display_gsp = mysql_fetch_array($query_gsp);
if ($sexporter = $display_gsp['Id']);
{
    $username = $_POST['username'];
    $FormA_No = $_POST['FormA_No'];
    $exp_name = $_POST['exp_name'];
    $address = $_POST['address'];
    $issue_country = $_POST['issue_country'];
    $importer_name = $_POST['importer_name'];

```

```

$imp_address = $_POST['imp_address'];
$country = $_POST['country'];
$port_route = $_POST['port_route'];
$BL = $_POST['BL'];
$vassel = $_POST['vassel'];
$container = $_POST['container'];
$Goods = $_POST['Goods'];
$exp_no = $_POST['exp_no'];
$LC_no = $_POST['LC_no'];
$UD_no = $_POST['UD_no'];
$bill_of_exp = $_POST['bill_of_exp'];
$association_reg_no = $_POST['association_reg_no'];
$bin = $_POST['bin'];
$category = $_POST['category'];
$hs_code = $_POST['hs_code'];
$qty = $_POST['qty'];
$inv_no = $_POST['inv_no']; $inv_value = $_POST['inv_value'];
$name_signatory = $_POST['name_signatory'];
$title = $_POST['title'];
$epb_ref_no = $_POST['epb_ref_no'];
$payment_no = $_POST['payment_no'];           $status           =
$_POST['status'];
$confirmation = $_POST['confirmation'];
$query = "UPDATE gsp set username= '$username' , FormA_No ='$FormA_No',
exp_name='$exp_name',address='$address',issue_country='$issue_country',
importer_name='$importer_name', imp_address='$imp_address', country='$country',
port_route='$port_route', BL='$BL', vassel='$vassel', container='$container',
Goods='$Goods', exp_no='$exp_no', LC_no='$LC_no', UD_no='$UD_no',
bill_of_exp='$bill_of_exp', association_reg_no='$association_reg_no', bin='$bin',
category='$category', hs_code='$hs_code', qty='$qty', inv_no='$inv_no',
inv_value='$inv_value', name_signatory ='$name_signatory', title ='$title', epb_ref_no

```

```

= '$epb_ref_no',    payment_no    ='$payment_no',    status    ='$status',
confirmation='$confirmation' where Id = '$exporter' and confirmation ='0'';
    mysql_query($query) or die(mysql_error());
    $stmt->bindParam(1, $username);$stmt->bindParam(2, $FormA_No);
    $stmt->bindParam(3, $exp_name); $stmt->bindParam(4, $address);
    $stmt->bindParam(5,          $hometown);$stmt->bindParam(6,
$issue_country);
    $stmt->bindParam(7, $importer_name);    $stmt->bindParam(8,
$imp_address);
    $stmt->bindParam(9, $country);    $stmt->bindParam(10,
$port_route);
    $stmt->bindParam(11, $BL);$stmt->bindParam(12, $vassel);
    $stmt->bindParam(13, $container); $stmt->bindParam(14, $Goods);
    $stmt->bindParam(15, $exp_no);$stmt->bindParam(16, $LC_no);
    $stmt->bindParam(17, $UD_no);$stmt->bindParam(18, $bill_of_exp);
    $stmt->bindParam(19,    $association_reg_no);$stmt->bindParam(20,
$bin);
    $stmt->bindParam(21, $category);$stmt->bindParam(22, $hs_code);
    $stmt->bindParam(23, $qty); $stmt->bindParam(24, $inv_no);
    $stmt->bindParam(25, $inv_value);*/}
    echo "Information Updated Successfully";
echo "<br>";echo "<a href = gsp_view_master.php>Go back </a>";
?>

```

