**Abstract**

.

This paper deals with the design of Internet billing system, in which it is possible pay invoices electronically. In the existing system we can give bills to the users by manually going to their home and all these details are maintained in papers. So this is very difficult to maintain large amount of data. In this proposed electronically billing system in which the admin can the bills to the users. By entering the bills details like duration of days and amount. The added bill details will be sending to the particular users. Here users can view the bill details and they can pay amount. Here users can get the reimbursement amount also. The admin can send the reimbursement amount only after paying the bill amount.

**CHAPTER – 1**

**Introduction**

Paper bills are now the primary channel of communication between companies and their customers. However, their potential for personalization is limited, and they are not interactive. If a customer wants to react to something in his paper bill – for example, to make a customer service inquiry or to order a new service – he must make a telephone call. Internet Billing promises far more than a new and inexpensive way to deliver billing information. Industry experts predict that Internet Billing will fundamentally change the way companies interact with their customers. Eventually, the Internet Bill will be an interactive entry to a host of additional services including customer self-care, automated sales one-tone marketing. The Internet Bill will become the gateway through which customers and companies have electronic one to one dialogs. Businesses and consumers are banking on the Internet in more than one sense. Despite the early proliferation of electronic banking applications on private networks through dial-up services, most electronic banking applications have migrated to the Internet.

Consumers will not be tied to one particular bank and its software, nor to a single terminal where the bank's own software must be installed. Banking on the Internet provides the flexibility of banking from any Internet access terminal using the now ubiquitous Web browser. Banking on the Internet can reduce the number of staff banks must maintain without having to make the investment in establishing private networks. The World Wide Web, or the Web, and its user-friendly, graphically rich browsers have made the Internet both friendly and accessible to the common desktop user at home and in the office. The advancement of electronic banking or commonly known as e-banking, began with the use of ATMs and has included telephone banking, Direct bill payment, electronic fund transfer, online banking and other electronic transactions. Banking services offered to consumers over the Internet will allow consumers to generate bank statements, check balances, transfer money between accounts, and authorize fund transfers to deposit money, to pay monthly bills, and to write personal checks. The Internet will provide a very competitive medium for banks to woo consumers. Consumers will be able to quickly and easily scan savings and loan rates and banking fees without having to interact with bank personnel. Beyond home banking, consumers will be able to write electronic checks to online merchants that draw value directly from the consumer's own bank account rather than use a line of credit. The Internet will make banking a much more competitive environment in another critical aspect. Local banks will now be competing with national and international banks whose Internet presence removes barriers of physical distance. In addition, a number of "virtual" banks have now entered the market to compete with traditional banks for clients. The environment created by Internet banking will present the vast array of services currently offered by banks in a form that is very convenient to consumers Commerce.

**1.1System analysis**

**1.1.1 Existing System:**

In the current system we need to keep a number of records related to the users and want to enter the details of the bills manually. In the existing system all activities mostly includes a lot of manual calculations and is mostly paper based. Distribution of bills is also done manually which is a time wastes process.

**Disadvantages:**

1. Time consuming process.

2. difficult to manage records.

**1.1.2 Proposed system:-**

In the proposed electronic billing system in which the user can the bills images as a proof to the admin.here admin can add the bill details to the particular user by viewing the bills uploaded by the users. Here users can view the bills added to him. Then he can pay the amount also. After paying this amount is viewed to the admin then admin can send the reimbursement amount to the user. Here user can view the received reimbursement amount.

**Advantages:**

* Security is high.
* Less time taking process.

**1.2 SYSTEM REQUIREMENTS**

**Hardware Requirements**

# Processor : Pentium –III

* RAM : 256 MB (min)
* Hard Disk : 20 GB
* Key Board : Standard Windows Keyboard
* Mouse : Two or Three Button Mouse
* Monitor : SVGA

**Software Requirements**

* Operating System : Windows family
* Application Server : Tomcat5.0/6.X
* Front End : HTML, CSS
* Scripts : JavaScript.
* Coding : PHP
* Database : MySQL 5.0

**CHAPTER 2**

**LITERATURE SURVEY**

Literature [survey](http://www.blurtit.com/q876299.html) is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the [programmers](http://www.blurtit.com/q876299.html) start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from [book](http://www.blurtit.com/q876299.html) or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

**CHAPTER 3**

**DESIGN AND IMPLIMENTATION**

**3.1 Proposed System**

In the proposed electronic billing system in which the user can the bills images as a proof to the admin.here admin can add the bill details to the particular user by viewing the bills uploaded by the users. Here users can view the bills added to him. Then he can pay the amount also. After paying this amount is viewed to the admin then admin can send the reimbursement amount to the user. Here user can view the received reimbursement amount.

**3.2 System design**

**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**3.2.1 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

**3.2.2 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**3.2.3SEQUENCE DIAGRAM:** A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**2.3.4Collaboration Diagram:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization where as the collaboration diagram shows the object organization.

****

**3.2.5 ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

****

**3.2.6 COMPONENT DIAGRAM**

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries etc. So the purpose of this diagram is different, Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details. Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation.



**3.2.7DEPLOYMENT DIAGRAM**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardwares used to deploy the application.

****

**3.2.8 DFD Diagram**

admin

user

login() upload bill images()

register() view bill details()

view users() pay amount()

accept users() view received amount()

add bills() logout()

view paid bills()

send reimbursment amount()

update unpaid bills()

view users()

Data Base

Data Base

user

user

**3.3 Modules Description:**

There are 2 modules

1. Admin

2. users

**Admin:-**

Here admin can login with their credentials. After login admin can view the user’s details and accept them. Here admin can view the users and admin can add the different types of bills to the users like current bill, water bill, phone bill and paper bill. Here admin can view the bills paid by the users and then send the reimbursement amount to them. Here admin can view the unpaid bills and update the details also.

**Users:-**

Here users can login with their credentials. New users can register also. Here users can login only after authenticated by the admin.after login user can view the bills details added to him and then pay the amount by entering the account details. Here user can get the reimbursement amount for the paid bills.

3.4 INPUT DESIGN AND OUTPUT DESIGN

**INPUT DESIGN**

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

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

**OBJECTIVES**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**OUTPUT DESIGN**

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

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

2. Select methods for presenting information.

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

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

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

### 3.5 SYSTEM TESTING

3.5.1 DEFINITION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**3.5.2 TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**3.6 Java Software Environment**

WHY CHOOSE “**PHP**":

Internet has joined the people living around the globe. It has, no doubt, gotten quite tough to sustain your identity in the cyber world as the competition has gone beyond the limits. To make it easier for you to compete and excel in the world of internet, PHP is among the best tools that can be used. PHP is abbreviation of “personal homepage” and sometimes is also known as Hypertext Preprocessor. The latter name is particularly used in the cyber circle. It is, in general, a HTML embedded scripting language being used widely for the web application development. The use of the language has increased in recent times due to the ease it offers to the developer. There are various benefits of using the language over the others developed for the same purpose. Some of the major pros pertinent to the language are discussed as under:

**Double end web development**

Some of the languages used for web development have limitation of purposes. PHP is one of its kinds because it may be used on both front-end and back-end web development. Due to this feature, a programmer may easily alter the present conditions of the website merely by changing a single code. Unlike PHP, other languages need to be uncoded to understand the correlation among the back-end and front-end languages making programming time taking and laborious.

**Why to pay when it’s free?**

Another reason why a programmer must prefer PHP over other languages is its legal free of cost availability. Some of the organizations having its similar programming languages in the market charge programmers against the language they offer. However, PHP may be downloaded and installed using any open source language house easily accessible from one’s computer. Therefore, the basic goal of earning more can be changed into reality by just taking a right step of using PHP as web application development.

**Simplicity and user friendliness**

It goes without saying that everyone wants ease out of the programming language. This is what PHP offers to the users. Unlike C++ and other similar languages, PHP is quite easy to be understood by the users. There is no need of any formal training prior to use the language for the required purpose. PHP programmed web applications are easy to scale and highly secure as compare to applications built in other language.

**Compatible to all the operating software**

Versatility in the available operating software, no doubt, has provided variety to the users but on the other hand has also caused some serious complexities for the programmers. However, you may make it easier for yourself by switching over to PHP as it is compatible to all the famous operating systems. Apple’s famous MAC and Microsoft’s Windows is among the top operating systems that are supported by PHP. Linux is also not out of the line of the compatible systems.

**ABOUT PHP:**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

## Common uses of PHP

* PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
* PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
* You add, delete, modify elements within your database thru PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.
* It can encrypt data.

**PHP: Built-in Database Access**

• PHP provides built-in database connectivity for a wide range of databases

– MySQL, PostgreSQL, Oracle, Berkeley DB, Informix, mSQL, Lotus Notes, and more

– Starting support for a specific database may involve PHP configuration steps

• Another advantage of using a programming language that has been designed for the creation of web apps.

• Support for each database is described in the PHP manual at:

– http://www.php.net/manual/en/ MySQL and PHP

• To connect to a database, need to create a connection

– At lowest level, this is a network connection

– Involves a login sequence (username/password

• Since this is a relatively expensive step, web application environments:

– Share connections

– Have multiple connections

• Whether, and how many, are typical configuration items. In MySQL:

– Allow\_persistent: whether to allow persistent connections

– Max\_persistent: the maximum number of persistent connections

– Max\_links: max number of connections, persistent and not

– Connection\_timeout: how long the persistent connection is left open

• Can also use SSL to encrypt connection

**High-Level Process of Using MySQL from PHP**

• Create a database connection

• Select database you wish to use

• Perform a SQL query

• Do some processing on query results

• Close database connection

**Creating Database Connection**

• Use either mysql\_connect or mysql\_pconnect to create database connection

– mysql\_connect: connection is closed at end of script (end of page)

– mysql\_pconnect: creates persistent connection

• connection remains even after end of the page

• Parameters

– Server – hostname of server

– Username – username on the database

– Password – password on the database

– New Link (mysql\_connect only) – reuse database connection created by previous call to mysql\_connect

– Client Flags

• MYSQL\_CLIENT\_SSL :: Use SSL

• MYSQL\_CLIENT\_COMPRESS :: Compress data sent to MySQL

**Security Note**

• Username and password fields imply that database password is sitting there in the source code – If someone gains access to source code, can compromise the database

– Servers are sometimes configured to view PHP source code when a resource is requested with “.phps” instead of “.php”

– One approach to avoid this: put this information in Web server config. File

• Then ensure the Web server config. file is not externally accessible

**Selecting a Database**

• mysql\_select\_db()

– Pass it the database name

• Related:

– mysql\_list\_dbs()

• List databases available

– Mysql\_list\_tables()

• List database tables available

**Perform SQL Query**

• Create query string

– $query = ‘SQL formatted string’

– $query = ‘SELECT \* FROM table’

• Submit query to database for processing

– $result = mysql\_query($query);

– For UPDATE, DELETE, DROP, etc, returns TRUE or FALSE

– For SELECT, SHOW, DESCRIBE or EXPLAIN, $result is an identifier for the results, and does not contain the results themselves

• $result is called a “resource” in this case

• A result of FALSE indicates an error

• If there is an error

– mysql\_error() returns error string from last MySQL call

**Process Results**

• Many functions exist to work with database results

• mysql\_num\_rows()

– Number of rows in the result set

– Useful for iterating over result set

• mysql\_fetch\_array()

– Returns a result row as an array

– Can be associative or numeric or both (default)

– $row = mysql\_fetch\_array($result);

– $row[‘column name’] :: value comes from database row with specified column name

– $row[0] :: value comes from first field in result set

**Process Results Loop**

• Easy loop for processing results:

$result = mysql\_query($qstring);

$num\_rows = mysql\_num\_rows($result);

for ($i=0; $I<$num\_rows;$i++){

$row = mysql\_fetch\_array($result);

// take action on database results here

}

**Why MYSQL?**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**Structured Query Language (SQL)**

* SQL (pronounced SEQUEL) is the programming language that defines and manipulates the database. SQL databases are relational

**CHAPTER 4**

**IMPLEMENTATION RESULTS**

**CHAPTER 5**

**Conclusion**

**5.1 Conclusion:**

This electronic billing system assists in automating the existing manual system. This is a paperless work. It can be monitored and controlled remotely. It reduces the man power required and provides accurate information. During the course of my dissertation, I found out that a computer aided result management system makes information management much more convenient and efficient; This is opposed to the manual method, which is stress less and timesaving process.

**CHAPTER 6**

**APPENDICES**

**CODE**

**ACCEPT.JSP**

<%@ page import="java.sql.\*" %>

<%

String id=request.getParameter("id");

String status="accepted";

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ebilling", "root" , "root");

PreparedStatement ps1=con.prepareStatement("update userreg set status=? where id='"+id+"'");

ps1.setString(1,status);

ps1.executeUpdate();

response.sendRedirect("acceptusers.jsp?msg= status updated successfully");

}

catch(Exception e)

{

e.printStackTrace();

}

%>

Acceptusers.jsp

<%@ page import="java.sql.\*" %>

<%

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ebilling", "root" , "root");

PreparedStatement ps1=con.prepareStatement("select \* from userreg where status='notaccepted'");

ResultSet rs=ps1.executeQuery();

%>

<div class="acceptusers">

<%@include file="adminmenu.jsp" %>

<div class="w3ls-news-grids">

<div class="w3-welcome-heading col-sm-12">

<h3 > ACCEPT USERS</h3>

</div>

<div class="news-right">

<table class="table" >

<tr>

<th>Id</th>

<th>Name</th>

<th>Phone No</th>

<th>Email</th>

<th>Account no</th>

<th>Address</th>

<th>Accept</th>

<th>Reject</th>

</tr>

<%while(rs.next())

{%>

<tr>

<td><font color="green"><%=rs.getString(1)%></font></td>

<td ><font color="green"><%=rs.getString(2)%></font></td>

<td ><font color="green"><%=rs.getString(4)%></font></td>

<td ><font color="green"><%=rs.getString(5)%></font></td>

<td ><font color="green"><%=rs.getString(6)%></font></td>

<td><font color="green"><%=rs.getString(7)%></font></td>

<td ><a href="accept.jsp?id=<%=rs.getString("id")%>">accept</a>

<td><a href="reject.jsp?id=<%=rs.getString("id")%>">reject</a>

</tr>

<%

}

}

catch(Exception e)

{

e.printStackTrace();

}

%>

</table>

</div>

</div>

</div>

</div>

<%@include file="footer.jsp" %>

**Adminhome.jsp**

<%

String id=request.getParameter("id");

System.out.println(id);

String bname=request.getParameter("bname");

System.out.println(bname);%>

<%@ page import="java.sql.\*" %>

<div class="admin">

<%@include file="adminmenu.jsp" %>

<div class="w3ls-news-grids">

<div class="news-right">

<div class="col-md-5 news-right-grid">

<div class="agile-news-info">

<img src="assets/imageb.png" alt=" " class="img-responsive">

<div class="agileinfo-news-button">

</div>

</div>

</div>

<div class="w3ls-news-grids">

<div class="news-right">

<div class="col-md-7 news-right-grid">

<div class="col-md-6 news-right-grid">

<div class="agile-news-info">

<img src="assets/imageb.png" alt=" " class="img-responsive">

</div>

</div>

<div class="col-md-6 news-right-grid">

<div class="agile-news-info">

<img src="assets/imageb.png" alt=" " class="img-responsive">

</div>

</div><div class="clearfix"> </div><br>

<div class="col-md-6 news-right-grid">

<div class="agile-news-info">

<img src="assets/imageb.png" alt=" " class="img-responsive">

</div>

</div>

<div class="col-md-6 news-right-grid">

<div class="agile-news-info">

<img src="assets/imageb.png" alt=" " class="img-responsive">

</div>

</div>

<div class="clearfix"> </div>

</div>

<div class="clearfix"> </div>

</div>

</div>

</div>

</div>

</div>

</div>

<%@include file="footer.jsp" %>

**CHAPTER 7**

**REFERENCES**

Books

* “An Integrated Approach to Software Engineering”- PankajJalote
* “Database Management System” - Elmasri and Navathe
* Advanced Java Programming - Dietel and Dietel
* Mastering JAVA 2 - John Zukowski
* Java Server Programming - Apress
* Software Engineering - Roger S Pressman
* Análysis & Design of InformationSystems – Senn

**Web Resources:**

[1] www.java2share.com

[2] www.way2java.com

[3] www.java2s.com

[4] www.roseindia.net

[5] www.javatpoint.com

[6] www.ajava2all.com