

DEVELOPMENT OF A TASK MANAGEMENT EMAIL SYSTEM

by

Ezeanwu P.Tochi , PG/MSC/11/59578

Being a MSC project report submitted in partial fulfillment of the requirements for the award of a Masters degree in Computer Science of the University of Nigeria.

Supervisor: Dr. (Mr.) GAM Ikekeonwu

Department Of Computer Science

University of Nigeria, Nsukka

September, 2014

Certification

We hereby declare that the work presented herein was done by me, and not by a third party. Should I be convicted of having cheated in this work. I shall accept my verdict of the University

Ezeanwu P. Tochi, PG/MSc/2011/59578

Approval

This project report is approved for submission.

Dr. (Mr.) GAM Ikekeonwu

Dedication

To God Almighty

For his infinite mercy, grace and strength to finish this work.

To Dr. Ezeanwu Achike Steven

For his constant encouragement and support

To Nne & Achike Ezeanwu

For their love and understanding.

Acknowledgement

My gratitude goes to God Almighty, the source of my existence who was ever present in times of need.

I acknowledge my dearest husband, Dr. Achike Ezeanwu for his words of encouragement and advice throughout this Masters Degree programme and especially throughout the project work.

I want to thank my fellow MSc colleagues especially Chinenye, Tobi, Odinaka and Chinyere for the sacrifices they willingly made for me in my areas of difficulty.

I want to say thank you to my Project Supervisor Dr. GAM Ikekeonwu for his advice, support and encouragement.

God bless you all.

Abstract

This project work analyzed the relevance of the Email in modern business practices and organization. As SMEs and large firms expand, they are faced with the problem of managing numerous staff dispersed in diverse field offices. There is a need to send up-to-date information about their products to a larger clientele. The proposed system design provides a management tool that handles employee task management, internal communication and information dissemination. The new system design made use of email to alert employees of task allocated to them with deadlines; broadcast confidential information to staff according their access privileges, give access to inter-office communication through a network e.t.c. Email applications generally can be run on premises, hosted or hybrid model (Hosted and On-premises). The new system design proposes a hybrid approach where the advantages of the both delivery models can be utilized. The Structured System Analysis and Design Methodology (SSADM) is used to develop the application. The new system is a web application with 3-tier architecture. The Front-end was designed using Html; the middle tier was designed using Java and JSP while the Mysql database was used at the data tier. The developer made use of the Tomcat JEE (Java Enterprise Edition) container as deployment container and web server. Java Mail library was also used.

TABLE OF CONTENTS

	Pages
Title Page -----	i
Approval Page -----	ii
Certification Page -----	iii
Dedication -----	iv
Acknowledgement -----	v
Abstract -----	vi
Table of contents -----	vii
List of Figures -----	viii
List of Tables -----	ix
CHAPTER ONE: INTRODUCTION	
1.1 Introduction -----	1
1.2 Statement of Problem -----	2
1.3 Objectives of the Study -----	2
1.4 Scope of the Study -----	3
1.5 Significance of the Study -----	3
1.6 Limitation of the Study -----	3

CHAPTER TWO: LITERATURE REVIEW

2.0 Introduction	5
2.1 Types of Email Systems	5
2.2 Review of Related Literature	7
2.3 Role of Email in business growth and management	14
2.4 Theoretical background	16

CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN

3.0 Introduction	24
3.1 Describe the existing system	24
3.2 Analysis of the proposed system	29
3.3 Design of the proposed system	29
3.4 System Architecture	32

CHAPTER FOUR: SYSTEM IMPLEMENTATION

4.0 Introduction	34
4.1 Choice of Development Environment	34
4.2 Implementation Environment	35

CHAPTER FIVE: SUMMARY AND CONCLUSION

5.0 Summary	37
5.1 Areas of Application of work	38
5.2 Suggestions for further Study	38

5.3 Recommendations í í í í í í í í í í í í í í í í í í í .38

5.4 Conclusion í .39

References

Appendix

List of Figures

Fig 2.1	A block diagram of an html form used in the project	9
Fig 3.1	The DFD level 0 Diagram	22
Fig 3.2	Detailed dataflow diagram	23
Fig 3.3	Flowchart of the system	24
Fig 3.4	ERD	25
Fig 3.5	ELD	25
Fig 3.6	Process description	27
Fig 3.7	System Architecture	28
Fig 4.0	Implementation architecture	29
Fig 4.1	Assign task	30
Fig 4.2	Snapshot of assign task panel	31
Fig 4.3	Snapshot showing email delivery	32
Fig 4.3	Snapshot showing submission to database	33

Fig 4.4	Snapshot- Mail is delivered-A view of the receiver's inbox.	37
---------	---	----

List of tables

Fig 3.1	Task Table in the database	32
Fig 3.2	Inbox	32
Fig 3.3	Task	32

CHAPTER 1: Introduction

This chapter introduces the general background of the project and gives an overview of the problem the project intends to solve with the proposed system design. It gives details of the objectives, scope, limitation, significance and definition of relevant terms.

1.1 Background of The Study

The E-mail is a universally accepted system; it is a more convenient way of communication. Many Organizations have discovered the relevance of email in a corporate environment. It is an undeniable fact that most businesses depend on email more than any other digital communication medium. [1].

E-mail functionality has progressed a long way from its original roots of simple text-based access to sophisticated content. At the same time, the number of users with e-mail access has grown exponentially. [2].

Large and medium firms in Nigeria make use of the email in their daily activities. Our main concern is on how growing Small Medium Enterprises (SMEs) and large firms can utilize the email to solve organizational problems.

Recent Researches have shown the important role of SMEs in the development of developing nations of which Nigeria is inclusive. [3] Large Firms on the other hand are the major employers in Nigeria.

Every thriving SME has the capacity to grow into a large corporation, which means more remote offices, more staff and more clientele. When it does, it would be faced with some challenges. Some of which are staff management, task management, internal communication, product marketing e.t.c [4],[5]

This project illustrates the development of a simple email application that can be used by medium and large organization to solve the above stated problems.

1.2 Statement of Problem

- Managers face the difficulty of managing projects among numerous staff in dispersed field offices.
- Manual handling and management of multiple tasks in a project amongst numerous employees in diverse locations.
- In a fast growing organization Information handling needs to be controlled such that confidential data can only be viewed and handled by the top management
- There is also a need to send up-to-date information about their products to a larger clientele.
- The need for dual email archiving system.

1.3 Objectives of The Study

The aim of this work is to develop a system that should be able to:

- Enable project manager create and manage projects
- Assign task activities to employees
- Broadcast messages to employees

- Manage and secure employee data
- Hybrid email archiving

1.4 Scope of The Study

The scope of the work is limited to the GUI interface design, database design and the business logic code implementation.

1.5 Significance of Study

Email helps to organize electronic documents sent via mail.

There is a need for business organizations to take control of company data so as to avoid leakage of sensitive or highly confidential information.

Organizations without any form of email management are faced with issues of accumulated emails, space limitation and the problem of finding storage space for messages and their attachments.

Daily allocation of tasks and deadlines on each task can control employees who are most times actively online during business hours.

All business organizations cannot do without proper information dissemination and transfer. This is needed for planning, organizing, leading and controlling.

Email archiving on premises ensures that emails with sensitive information are archived on premises.

The Reports generated from this system can be used for employee appraisal

CHAPTER TWO: **Literature Review**

2.0 Introduction

The early email was like a simple file directory, it was used by placing a message in another user's directory in a spot where they could see it when they are logged in. From a directory structure it was designed as a system that could allow messages to be sent to several users. The first email system was the MAILBOX used at Massachusetts institute of technology.[1] It was used to send messages to various users of the same computer before the advent of the internet. As the internet evolved the email was developed to send messages from a single computer to multiple computers. Internet standard protocols were introduced to checkmate forgery of email addresses . As internet standards began to mature the POP (Post Office Protocol) became an important standard to allow users to develop email systems that would work with each other. One of the first commercial email system EUDORA was developed.[7]

With the world wide web email systems have been made available in web interfaces by providers such as Yahoo and Hotmail. Furthermore, Integrated email systems are used in business enterprises to meet the growing need of business communication and handle the problem of increasing volumes of inbound mail.[7]

2.1 Types of Email Systems

Email systems have evolved over the years, from simple directory system to a stand-alone system, a web based system and finally to an integrated system

offering email services. The listing below shows the classification of email systems and examples;

2.1.1 LAN Based Email System:

Eudora and Pegasus Mail were produced in the 80s. Eudora was the first commercial product to be widely used in a local Area network (LAN). Email messages were strictly between computers in the local network.

2.1.2 Internet or Web based Email System:

Starting from the 1990s, companies have been taking advantage of the internet and the email by providing email over the web. The key advantage of this service is that the user need not carry anything as he moves about since one can check email anywhere with a web browser. One disadvantage with this system is that they do not always have as many features as a standalone system and the user may have to pay for extra features. Some web based systems are hotmail, Yahoo, gmail e.t.c

2.1.3 Mobile Email System:

Push mail is another name for this type of email system. It is simply email accessed on mobile devices. Smartphones and more modern electronic gadget have applications that deliver emails to the device and the user can access it on the go. E,g of these devices are Blackberry, iphone, tablets and notes. One disadvantage of this type of email is in the way the email is rendered on the device.

2.1.4 Integrated Email System:

An Email system can be integrated with a wide range of custom applications. It can be embedded to corporate softwares.e.g ERP (Enterprise resource planning), CRM

system, project management systems. Echomail was the first integrated email system.

2.2 Review of Related Literature

The Internet evolved with great speed during the 1990s, it was once an obscure academic system but in recent times it has become a mainstay of the developed world's daily routines of communication, shopping, travel, entertainment, and business. The millions of ordinary people who rush to connect their computers to the network are mostly driven by a desire to use two particular kinds of program: Email systems and the World Wide Web.[13]

On like the computer system, the operation of the Internet does not lie in hardware or software but in protocols: protocols are the agreed upon rules by which computer programs communicate with each other. When computers communicate on the Internet, it is these protocols that determine what is automatic and what is impossible; what is easy to accomplish and what requires complex or cumbersome special procedures. But the protocols that define the Internet were designed by and for a group of users very different from that of the commercial Internet.

The fundamental data transmission protocol suite of the Internet, TCP/IP, was developed during the late 1970s. Simple Mail Transfer Protocol (SMTP) the main protocol for sending email on the internet, surfaced in the early 1980s. And although the World Wide Web was not created until the early 1990s, its protocols were modeled on and designed to work with the existing standards of the Internet.

Since the ARPANET was replaced by the Internet, the Email remains a key network application. And from the early 1980s onward, specially created protocols for Internet email ensured that people using different email programs and different

types of computer were able to exchange messages freely. Mail was relayed to the recipient's machine using SMTP. As an application protocol this ran over the Internet's native TCP/IP, meaning that all the complexity of routing messages and verifying their successful transmission was taken care of by TCP/IP. SMTP had to do little more than use Internet's existing infrastructure to contact the recipient's mail server and then transmit the name of the sender, name of the recipient, and the text of the message itself. Email servers were connected to the Internet, and were registered with the Internet's domain name servers as the place to which all email addressed to recipients in a particular domain. The first widely used SMTP message delivery software was Sendmail, and today SMTP is supported by virtually all email clients to send outgoing messages. While SMTP could originally transmit only plain (7-bit encoded) text, it was later supplemented with the MIME (Multipurpose Internet Mail Extensions) standard for encoding file attachments and non-Western alphabets.[13]

Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks. SMTP was first defined by RFC 821 (1982), eventually declared STD 10 and last updated by RFC 5321 (2008) which includes the Extended SMTP (ESMTP) additions, and is the protocol in widespread use today.[7] SMTP uses TCP port 25. The protocol for new submissions (MSA) is effectively the same as SMTP, but it uses port 587 instead. SMTP connections secured by SSL are known by the shorthand SMTPS, though SMTPS is not a protocol in its own right.

While electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use SMTP only for sending messages to a mail server for relaying. For receiving messages, client

applications usually use either the Post Office Protocol (POP) or the Internet Message Access Protocol (IMAP) or a proprietary system (such as Microsoft Exchange or Lotus Notes/Domino) to access their mail box accounts on a mail server.[12].

With the technical knowledge of the email and its protocols, we relate the application of the email to business development. The following findings were made:

In information societies which consist of developed and developing countries, economic priorities are given to information production and usage. Information technology (IT) and email is critical to the success of most businesses. In recent years, there has been a widespread interest in understanding the factors that affect IT usage in SMEs, they gain competitive advantage through the use of information and communications technology. In order to be ahead of your competitors you have to get to your customer's inbox first[6].

Every organization that uses email would have to make a decision on which email system would meet its organizational needs. Researches and results of survey show that though most SMEs and large organization may want to be able to manage and archive their email in house but they also desire to reap some of the benefits of the hosted email[8].

A recent survey on email archiving commissioned by GFI Software found that a growing number of organizations are considering or would consider a third-party hosted email archiving service. A total of 18% of those organizations that already use an email archiving solution, have opted for a hosted service, while 38% are open to using such a service. At the same time, 51% of those surveyed said they

would still only use an on-premise email archiving solution. Apart from the shocking statistic that more than 63% do not archive their email.[9]

According to Osterman Research Surveys the following findings were documented:

One important benefit of an on premises email solution is that it permits complete control over the custody of data. A company using a cloud provider may not be able to exercise this level of control over how and where their data is managed, resulting in legal or regulatory problems. This is a critical consideration in a variety of situations.

For example:

Personal data held by European companies, such as personnel records or customer financial information, cannot be transferred out of the European Economic Area (EEA) ó the 27 countries that belong to the European Union ó without violating the EU Data Protection Directive (95/46/EC). The only US companies permitted to transfer such data outside of the EEA are those that are self-certified under the Safe Harbor framework, a rigorous set of requirements designed to protect the integrity of sensitive data about residents of the EEA. If a non-certified company in the EEA uses a cloud-based provider that hosts data outside of the EEA, that company could be in violation of the EU Data Protection Directive if it cannot control the location of its data storage[11].

Singapore's banking laws require additional due diligence when using an outsourced provider of any kind outside the country. For example, regulations issued by the Monetary Authority of Singapore (MAS) require that banks are to

take appropriate due diligence measures, including the assessment of the track record, reputation, financial soundness of the service provider and its ability to safeguard the confidentiality of information entrusted to it. Further, MAS requires that all outsourcing arrangements are to be appropriately documented by means of a written outsourcing agreement.

Swiss laws prevent banks in the country from sharing data outside the country, making use of cloud providers unless they can be assured that the content they store in the cloud never leaves Switzerland. [21]

There are some very high volume email-generating applications that are often better managed using on premises infrastructure, particularly when communications are highly time sensitive or when confidential information is being sent. For example, some brokerage houses send out very large volumes of email early each trading day to customers and prospects so an inability to send high volumes of email during a tight time window means that fewer customers are informed of opportunities which, in turn, lead to fewer investments by these customers.

Many organizations also find that some of their email-enabled applications simply cannot migrate well to the cloud because it would be too costly to rewrite them for cloud-based delivery. These may be legacy applications that would have to be completely re-written if they were transferred to the cloud, or applications that are important to a particular function, but not to the enterprise as a whole.

Osterman Research surveys show that the majority of decision makers prefer to maintain the responsibility of sending emails inside the organization. It believes that most mid-sized and large organizations will ultimately decide how best to use a combination of solutions, not one or the other.[8]

There are a number of benefits associated with using hosted email. These benefits include:

More predictable cost of ownership because of the long-term contracts that are normally established by mid-sized and large companies that opt to use a third-party solution to host email for at least some of their users.

Potentially lower cost of ownership because IT staff are not required to maintain servers and other infrastructure devoted to email management.

The cloud and on-premises systems are not mutually exclusive delivery models ó many organizations are using both models in combination, realizing the advantages of both. This project proposes a hybrid delivery approach to cub the problems of the existing system.

The proposed system proposes a scenario whereby an on-premises email system will be used for their corporate headquarters to serve several thousand users with a team of dedicated IT staff members. For field offices, where there might be only 10 or 20 employees at each location, the cost of employing IT staff might be prohibitive and a hosted email solution might be more appropriate.

Some organizations opt for a cloud-based solution for its employees that use email, but maintain dedicated, on- premises systems for very high-volume and time-sensitive email broadcasts for its customers.

On-premises email systems could be used for the primary email system, but a cloud-based email continuity/disaster recovery system could be used to maintain availability during outages of the primary system.

The primary email functionality could be maintained using on-premises infrastructure, but all ancillary capabilities for anti-spam, anti-virus, anti-malware and email archiving could be maintained in the cloud. [11]

Some of the companies that are using such a hybrid approach are stated below:

GlaxoSmithKline has migrated its users to Microsoft's cloud-based BPOS and ForeFront offerings, while still maintaining an email backbone for some of its email functionality.

One of the largest biotech companies is using Google Gmail and Google Postini Services for email and email security, respectively, but also using Send mail for its email backbone. [11]

After determining which delivery model is most preferred, the organization has to decide which exchange server is ideal. Microsoft Exchange Server is the most popular exchange server in the world. An Exchange server is a software product designed to make emailing faster and more efficient. As email is now a critical function in the business environment, a great deal of business is now conducted by email, many companies are finding that using Microsoft Exchange Server has increased the reliability of their email functions. [21]

One of the biggest benefits that Microsoft Exchange Server provides to its users is the enhanced security features included in the software package. There are many different threats that can compromise the security of a software package today and Microsoft Exchange Server works to neutralize those threats, leaving the users less vulnerable to hackers, viruses, and spam. It also works to ensure that emails remain confidential and are not compromised by outside sources. This enhanced security allows companies to feel confident that their email system is protected at all times and allows employees and customers to utilize the email system to its full potential.

Microsoft Exchange Server also allows businesses to be more efficient, by making it a great deal easier to find and fix any issues that may arise. It also has improved monitoring capabilities so that any issues that do occur can be detected quickly before it disrupts business by a great amount. Many tasks that are needed to keep Microsoft Exchange Server operating properly have been automated, allowing the administrators more time to deal with more pressing issues. There are many benefits to using Microsoft Exchange Server and many businesses are discovering that this is a program that they cannot do without.

Other functions of Microsoft exchange are as follows:

The exchange supports IMAP, POP and web clients including Microsoft Outlook.

The exchange lets users share information either on premises with an email client or on the web.

2.3 Role of Email in Business Growth and Management

Information technology has changed the way businesses are run today. Business owners have learnt to work smarter making use of tools this technology provides. One way it has caused change is in the way information is dispersed to many people. The conventional approach to communication and information dissemination is by holding series of meetings and face to face contacts. It poses a big problem when the organization becomes larger with several field offices and numerous staff dispersed in different locations. As a business expands, their clientele base increases. Any business owner who does not communicate regularly with his/her customers may risk losing them.[6]

The Email as an e-messaging system has proven to be best option to meet organizational needs. Though other forms of messaging exist but it is observed that the email is more beneficial to business owners. As a communication tool, more time is spent using email than the telephone, instant messaging and social media combined. Because of the ease with which it can be used to send files, email is the primary content transport system in most organizations. It is still the chief way that many users collaborate with one another on projects and documents both inside and outside their organizations.[7]

Other advantages are quick information dissemination, marketing of products and archiving information that is relevant to the organization. It can also be used internally to handle communication between the top level management and their subordinates and to also ease task management.

2.4 Theoretical Background

The major technologies used in this project are web technologies (JSP, Html, and Java), Relational database technology and a Web server (Apache Tomcat). The site developed is a collection of static and dynamic web documents. The static web documents are written in html while the dynamic web documents are the Jsp web pages with Java codes embedded in it. The JSP web pages were used to insert dynamic content; it's dynamic because it is generated by the JSP scriptlets and the result can be viewed on a web server. JSP web pages enable us to enter Java code through a Jsp scriptlet into html tags.

In Jsp, the start and end tag are `<%` and `%>`. This shows that the content of these tags is a Jsp or Java syntax. Jsp web pages can contain html elements, Java elements and Jsp actions and commands.

The following is an example of a Jsp element used in this project

1. This is an html code used to create a form in the project.

```
<% @ page language="Java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @page import="toppackage.SendMail" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
```

```

<title>Manager tool</title>
</head>
<body>
<h1>Assign Task Page-Manager Tool</h1>
<form action="save.JSP" method="post">
  <table border="0" width="35%" align="center">
    <tr>
      <td width="50%">Task Title </td>
      <td><input type="text" name="title" size="50"/></td>
    </tr>
    <tr>
      <td>Employee </td>
      <td><input type="text" name="employee" size="50"/></td>
    </tr>
    <tr>
      <td>Date </td>
      <td><input type="text" name="date" size="50"/></td>
    </tr><tr>
      <td>Email </td>
      <td><input type="text" name="email" size="50"/></td>
    </tr><tr>
      <td>Deadline</td>
      <td><input type="text" name="deadline" size="50"/></td>
    </tr>
    <tr>
      <td>Task </td>

```

```

        <td><textarea rows="10" cols="39" name="task"
size="250"></textarea> </td>
    </tr>
    <tr>
        <td colspan="2" align="center"><input type="submit"
value="Send"/></td>
    </tr>
</table>

</form>
</body>
</html>

```

This html form was written in a Jsp webpage. It creates a form that would call the save.jsp webpage to perform a connection to the database to save the entries from the form.

The Jsp webpage with the html form looks this way when rendered on a web server.

Figure 2.1: A block diagram of an html form used in the project

Task Title	<input type="text"/>
Employee	<input type="text"/>
Date	<input type="text"/>
Email	<input type="text"/>
Deadline	<input type="text"/>

Task

2. The save.jsp collects the entries from the form. It uses the `getParameter()` Jsp method and the `õgetö` action. This code below shows how it is used.

```
<%
```

```
String Employee=request.getParameter("employee");
```

```
String Task=request.getParameter("task");
```

```
String Email=request.getParameter("email");
```

```
%>
```

It gets the entries from the form and assigns them to a variable string.

3. The next thing the save.jsp does is to make a connection to the mysql database and save the entries in the database. Below is the code that makes the connection to the database. The database has already been created using PhpMyAdmin and is named `õtestö` and the entries are sent to a database table called `õtaksö`

```
try
```

```
{
```

```

String url = "jdbc:mysql://localhost/test";
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection(url,"root","");
Statement st=con.createStatement();
st.executeUpdate("insert into tasks (tasktitle,date,deadline,employee,task,email)
values('"+Tasktitle+"','"+Date+"','"+Deadline+"','"+Employee+"','"+Task+"','"+
Email+"')");
}
catch(Exception ex)
{
out.println(ex.getMessage());
}

```

It uses the try and catch statement to initiate a connection to the database and would print out an error message if it encounters an error. After it initiates the connection to the database and it's successful, it inserts the data it was holding in the variable strings into the different fields in the table (tasks) of the database (test).

4. The next thing the save.jsp does for us is to send a mail after submitting the content to the database. It does this by importing a Java class into the save.jsp webpage.

```
<% @page import="toppackage.SendMail" %>
```

The syntax above tells us we are importing a Java class called SendMail into this Jsp webpage and it belongs to a package called toppackage.

5. The Java class SendMail hosts a Java method called sending(). The import statement allows the sending method to be called and the mail sent. The sending method was declared as follows:

```
public void sending(String host, String port,  
    final String username, final String password, String toAddress,  
    String subject, String message) throws  
    MessagingException {  
  
    Properties props = new Properties() ;  
    props. put ( "mail.smtp.auth" , "true" ) ;  
    props. put ( "mail.smtp.starttls.enable" , "true" ) ;  
    props. put ( "mail.smtp.host" , host);  
    props. put ( "mail.smtp.port" , port);  
    Session session = Session. getInstance ( props,  
    new Javax. mail . Authenticator () {  
    protected PasswordAuthentication getPasswordAuthentication () {  
    return new PasswordAuthentication ( username, password ) ;  
    }  
    });
```

The code supplies the parameters required to make a connection to a Smtplib server that would enable the mail to be transported.

6. After this is done, a try and catch statement is used to set up the structure of the mail, the subject, body and sender. And finally the mail is sent.

```
try {
```

```

Message messageM = new MimeMessage ( session) ;
messageM. setFrom ( new InternetAddress (username)) ;
messageM. setRecipients ( Message. RecipientType . TO ,
InternetAddress. parse ( toAddress)) ;
messageM. setSubject( subject) ;
messageM. setText ( message ) ;
Transport. send ( messageM) ;
result = "You have succeeded";
System . out . println ( result ) ;
        } catch ( MessagingException e ) {
throw new RuntimeException ( e ) ;
    }
}
}
}

```

7. When the sending() method is called, it requires the user to enter the host, port, username, password, toAddress, subject and message.

8. To use the sending() an object of the SendMail class was created to enable the method to be called. Calling the method sends the mail automatically.

See below:

```
SendMail obj = new SendMail();
```

```
Obj.Sending().
```


CHAPTER THREE: **SYSTEM ANALYSIS AND DESIGN**

3.0 Introduction

The software development approach used in the design of our system is the structured software analysis and design methodology (SSADM).

SSADM divides an application development into modules, stages, steps and tasks. It covers aspects of the life cycle of a system from feasibility study stage to the production of the physical design.

In detail, SSADM sets out a waterfall view of a system development, in which there are a series of steps, each of which leads to the next step. The SSADM method involves the application of a series of analysis, documentation and design tasks. Stages of the SSADM are as follows:

1. Feasibility study
2. Investigation of the current environment
3. Business systems options
4. Definition of requirements
5. Technical systems options
6. Logical design
7. Physical design

3.1 Describe the existing system

The following processes are observed in the existing:

1. Manager has to meet with all staff daily to share the task for the day
2. Task sharing and management is usually verbally or paper based
3. Project planning requires chains of meetings between some staff
4. Inter-office meetings

After careful analysis, the following problems were identified in the system

1. Informal mode of communication
2. Decentralized information dissemination thereby causing an information gap between the top and the subordinates
3. Time Wasting: Series of meetings to realise proper task management to staff
4. Improper follow up: Series of inter-branch meetings to follow up with events happening in each branch.
5. Manual Task allocation and planning: Paper based task allocation

3.2 Analysis of the proposed system

There are tools used in SSADM to create a model for the proposed system. The following tools were used to model the proposed system: *System Dataflow Diagram (DFD)*, *system flowchart*, *Entity relationship diagram*, *Entity Life History*

Figure 3.1: The DFD level 0 concept diagram

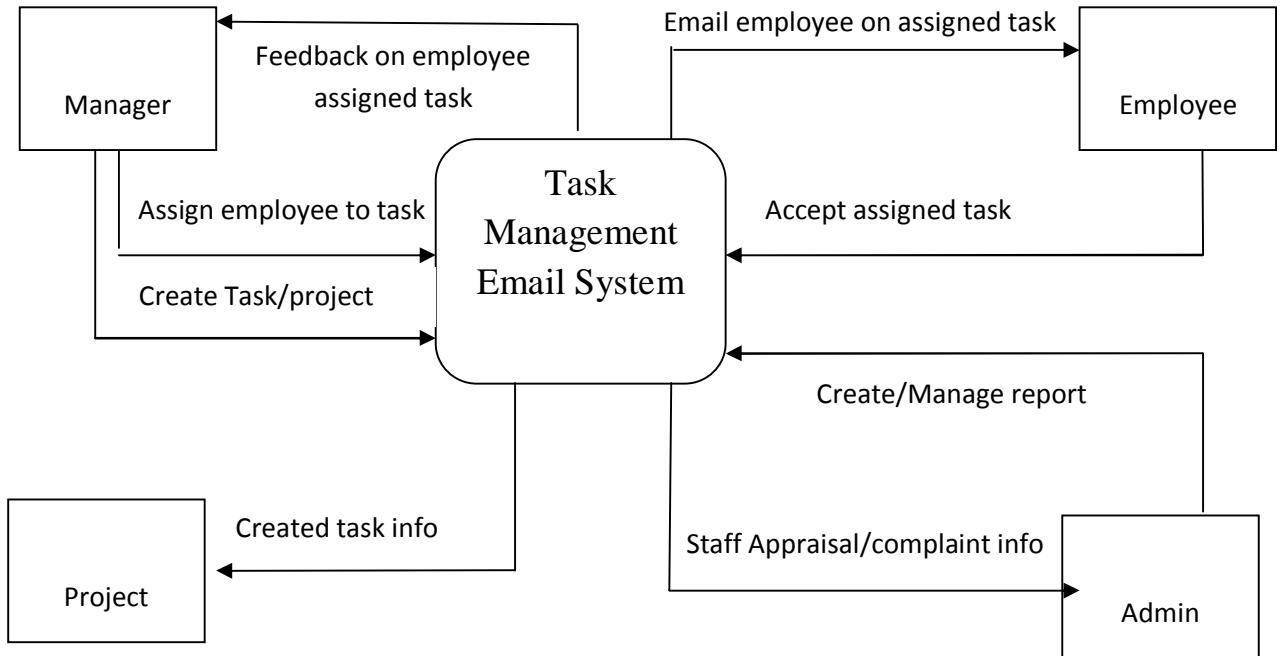


Figure 3.2 Detailed DFD Diagram

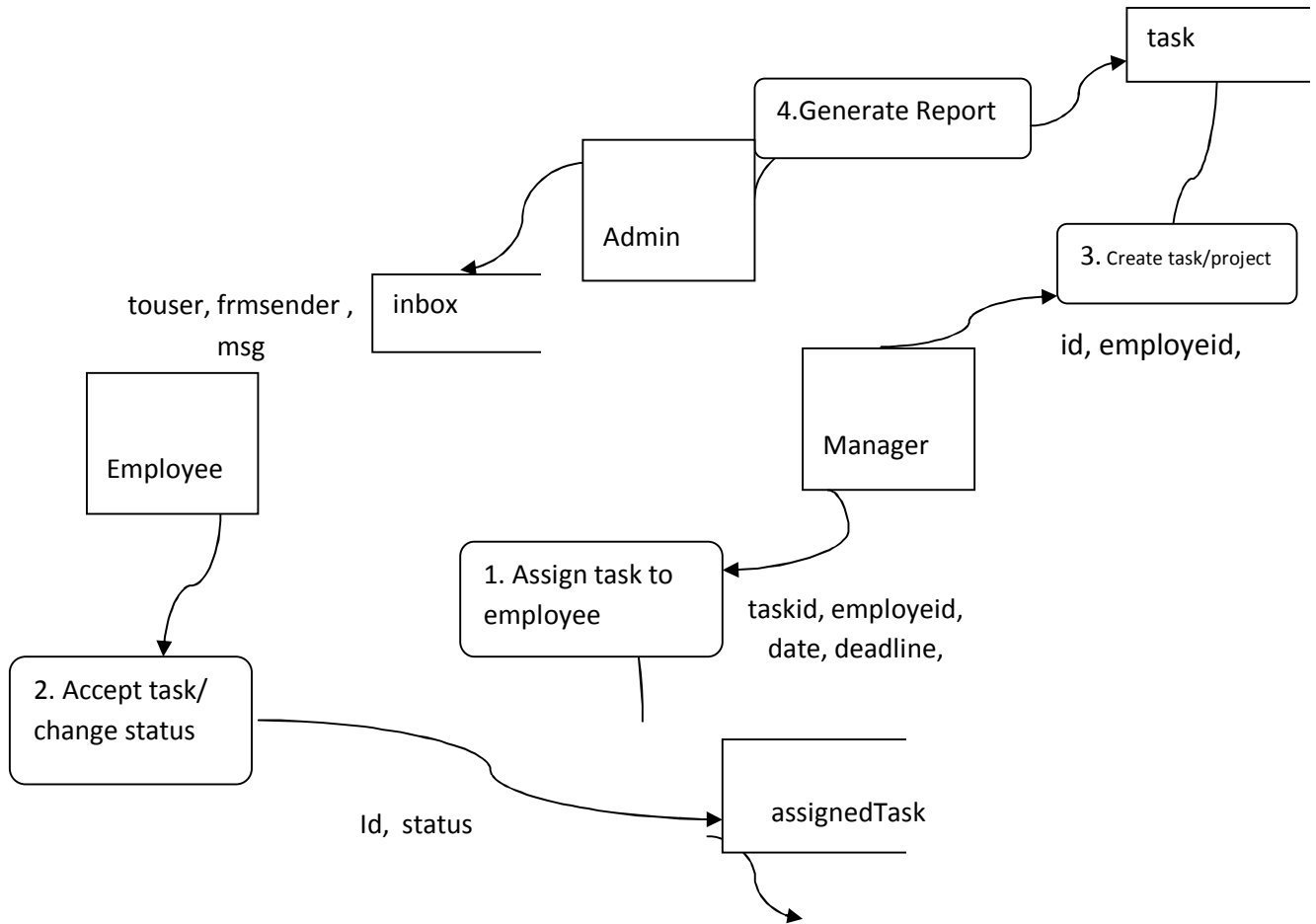


Figure 3.3: Flowchart of the system

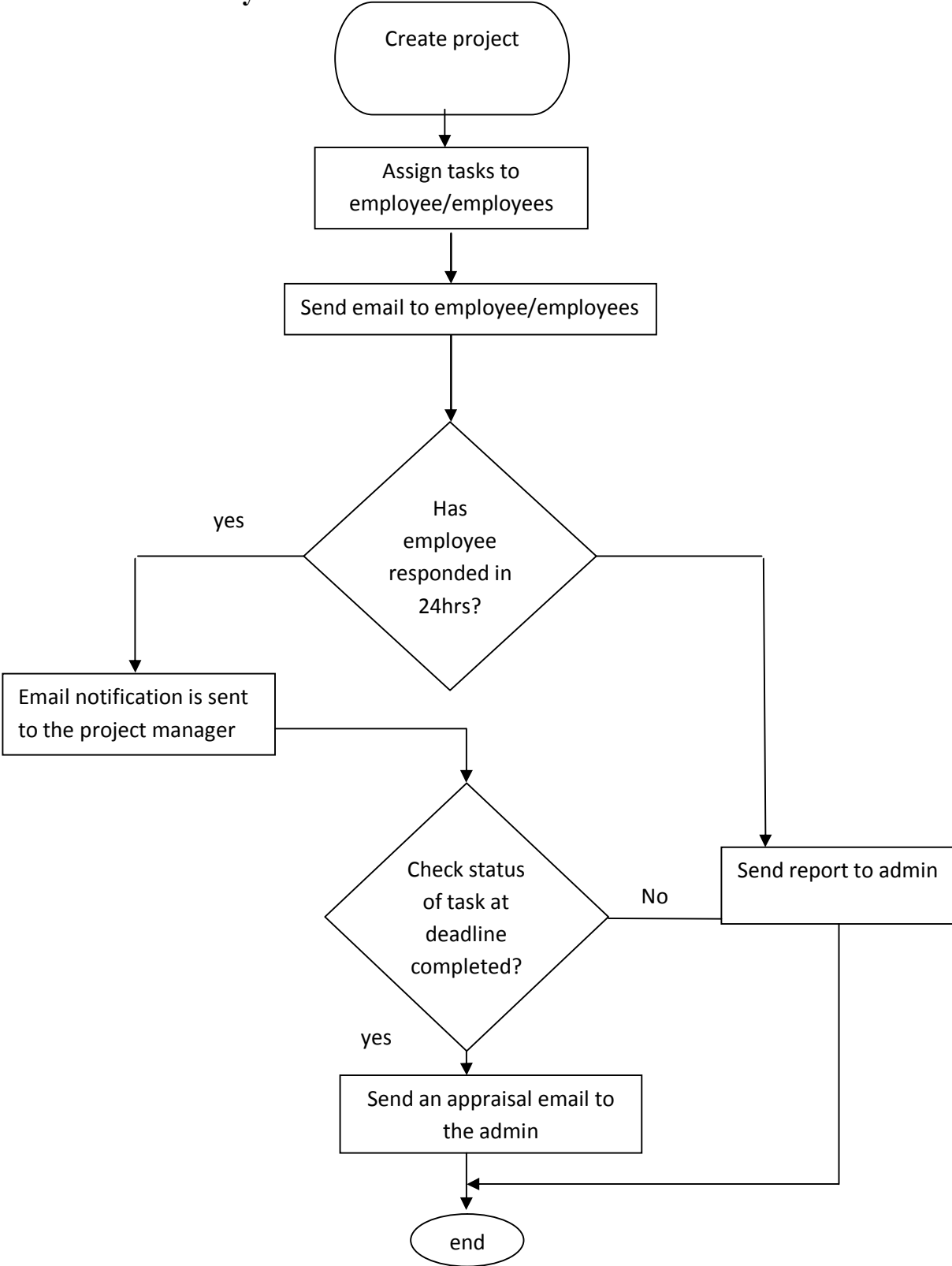


Figure 3.4 Entity Relationship Diagram(ERD)

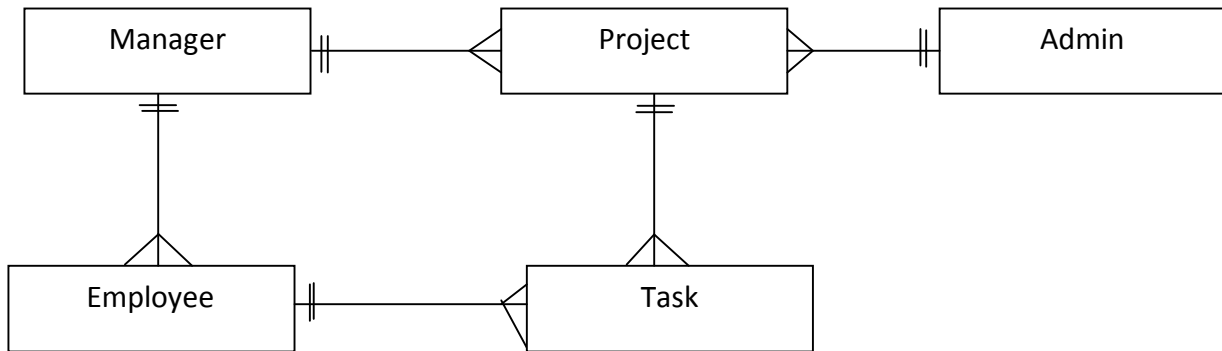
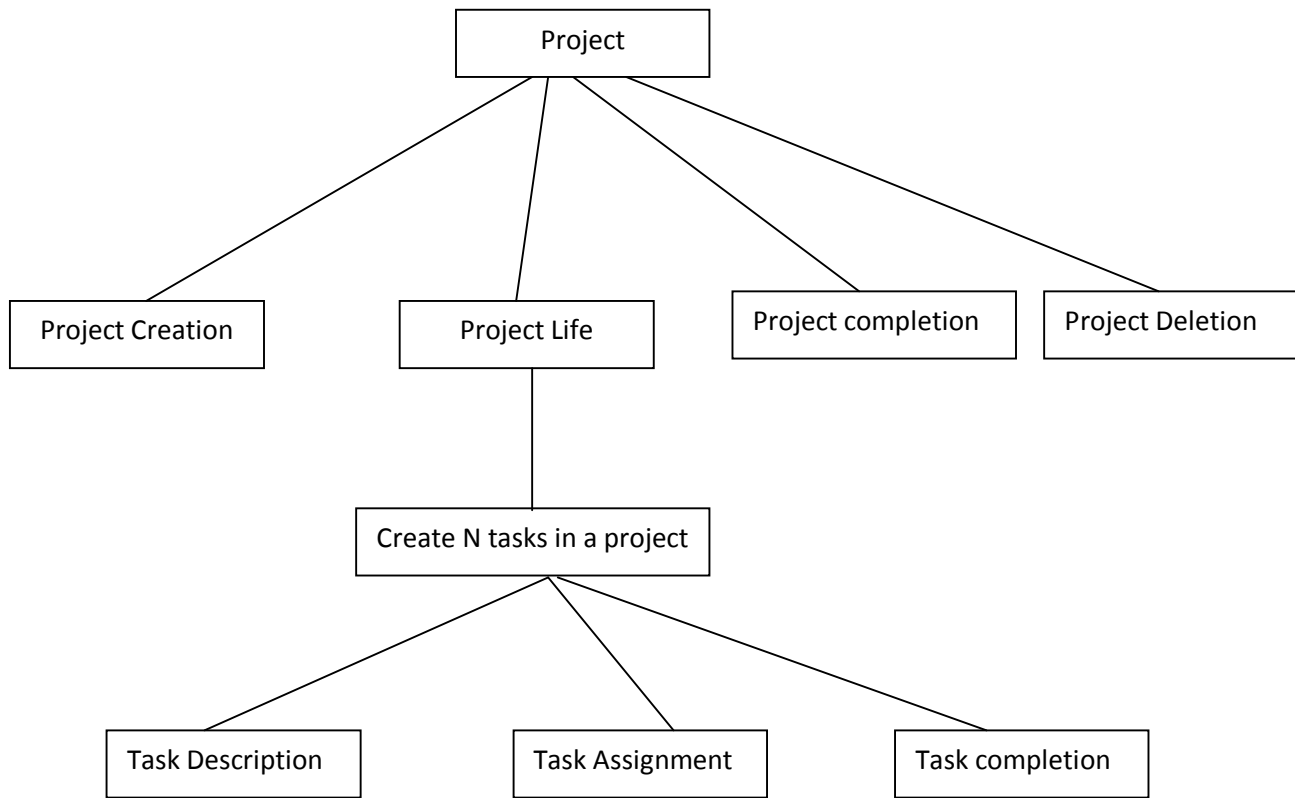


Figure 3.5: Entity Life Diagram (ELD)



3.3 Design of the proposed system

The proposed application is designed to allow organizations manage task between project managers, top management executives and their subordinates. The system works by sub-dividing projects into series of individual tasks.

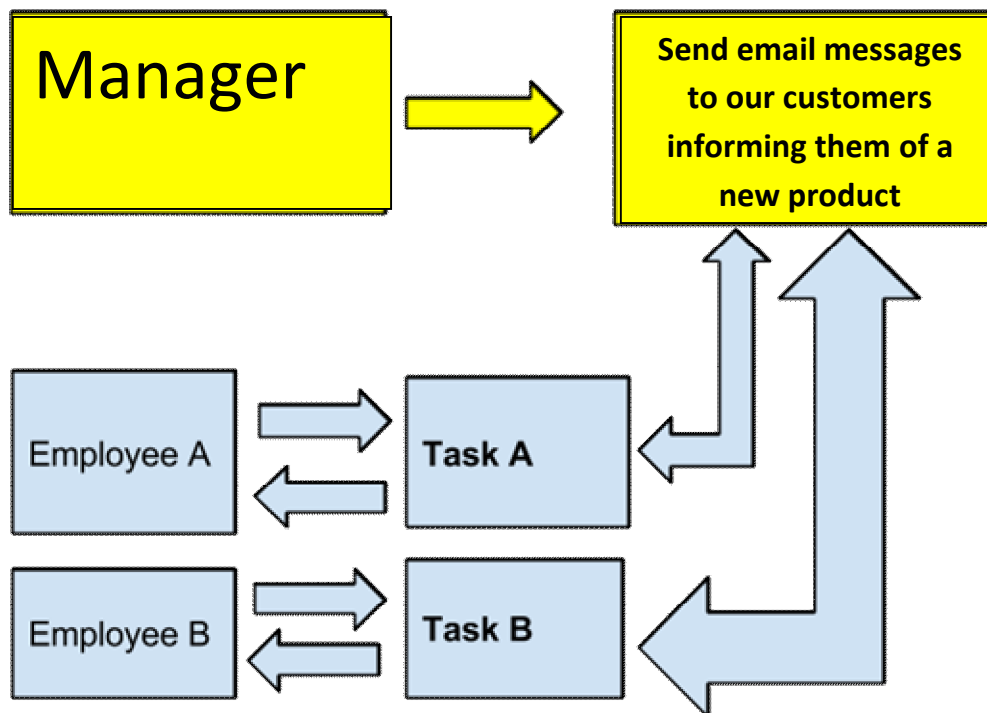


Figure 3.6: Process description

An employer or project co-ordinator creates a project. He/She then assigns the projects to an individual. He/She may decide to sub-divide the project into individual tasks.

A project may consist of series of tasks. Each task can be handled separately by an individual. Each task in a project has a timeframe within which it must be completed.

The employee may update the status of the project and can send messages to the

project manager or employer with updates on the state of his/her own tasks.

This process allows easy review and appraisal of employees.

If an employee is unable to complete a task within a given timeframe, the project may be assigned to another employee.

There would be a timeline to monitor the progress of the project over the stipulated project time frame. The system uses email as the messaging medium to allow employees and employer/project managers exchange information. Once an employee updates a task status, the employer/project manager gets an email. The same applies to the employee who gets an email when the employer/project manager comments on the status of the project.

3.3.1 Database Design

The database management system used in this project is MySQL. The required entities (tables) include the following: Task, Admin, Employee, Inbox, Users e.t.c

Table 1: assignedtask

Column	Type	Null	Default
Id	int(11)	No	
Taskid	int(11)	Yes	<i>NULL</i>
Employeeid	int(11)	Yes	<i>NULL</i>
Managerid	int(11)	Yes	<i>NULL</i>

Table 2: inbox

Column	Type	Null	Default
Id	int(11)	No	
Touser	varchar(255)	Yes	<i>NULL</i>
Fromsender	varchar(255)	Yes	<i>NULL</i>
Subject	varchar(255)	Yes	<i>NULL</i>
Message	longtext	Yes	<i>NULL</i>
Msgdate	datetime	Yes	<i>NULL</i>
Status	int(11)	Yes	<i>NULL</i>

Table 3: Task

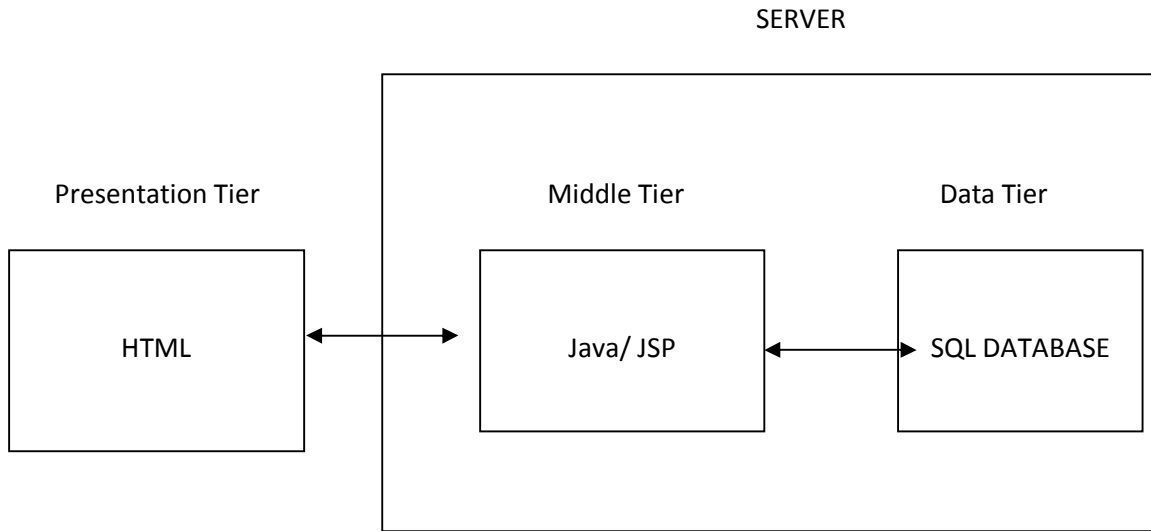
Column	Type	Null	Default
Id	int(11)	No	
Employeeid	int(11)	No	
Tasktitle	varchar(200)	Yes	<i>NULL</i>

Task	Text	Yes	<i>NULL</i>
Date	Date	Yes	<i>NULL</i>
Deadline	Date	Yes	<i>NULL</i>
Status	int(11)	Yes	<i>NULL</i>
Comments	Text	Yes	<i>NULL</i>

3.5 System Architecture

The proposed system has 3-tier architecture. The tiers are the presentation, middle tier and data tier. The presentation tier is the user interface also called the front end which is the email client. It was designed using html. The middle tier is the business logic is designed using Java and Java server pages and it runs on the server. The data tier is the part of the system that is responsible for storing data which is the database. The database management system used in this system is the MySQL database.

Figure 1.0: The System Architecture of the proposed system



CHAPTER FOUR: SYSTEM IMPLEMENTATION

4.0 Introduction

The application was developed using Java (JSP) as the programming language and MySQL (DBMS) as the database technology (Back end database). The front-end was done using HTML so that it can be readily accessible from all browsers.

The application makes use of the Tomcat JEE (Java Enterprise Edition) container as deployment container and web server. Java Mail library was also used.

4.1 Choice of development Environment

The reason for choice of technology includes:

- a. Enterprise grade with support for concurrent users.
- b. Stable and Tested Technology Used In production environments
- c. The technology allows for easy scalability. Example if we had more users we won't need to redevelop the application rather we would just need to upgrade our hardware. A common challenge of enterprise applications is scalability (handle increase in user traffic or workload), in the case of a higher number of users than initially planned for, the application would be re-developed. But this is not the case with our choice of technologies.

4.2 Implementation Architecture

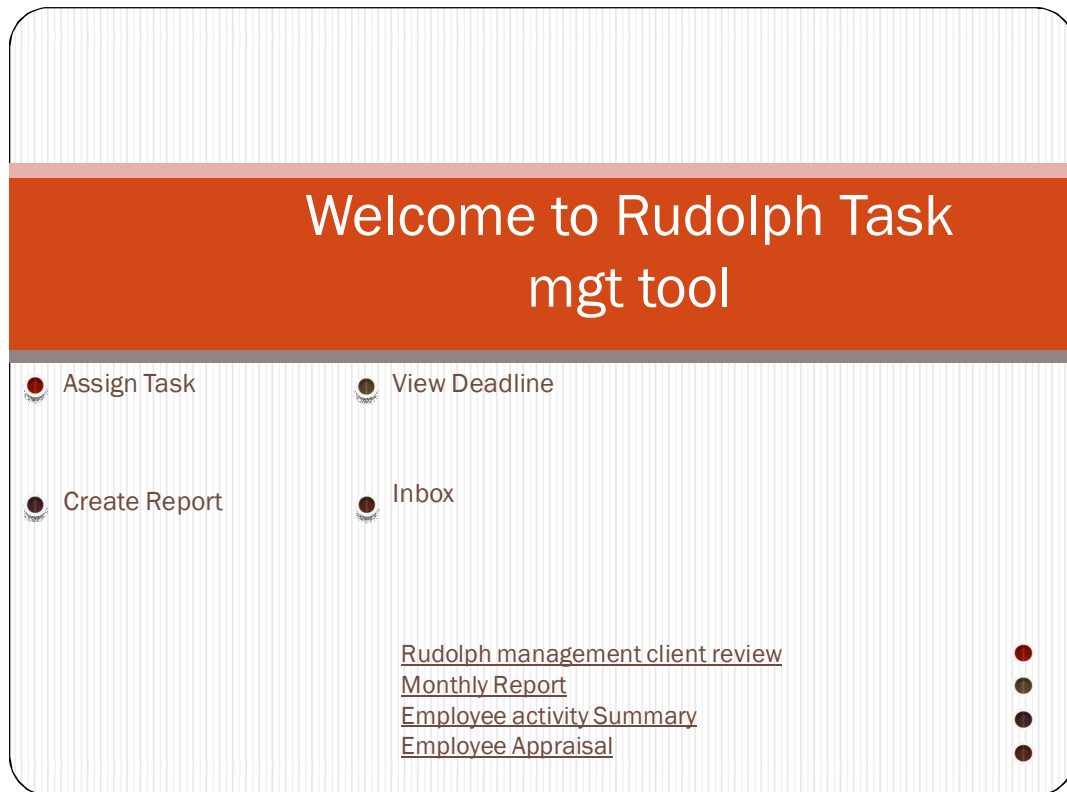


Figure 4.1 implementation architecture

The form consists of a large rectangular container. Inside, there are six input fields stacked vertically, each with a label to its left: 'Task Title:', 'Employee:', 'Date:', 'Email:', 'Deadline:', and 'Task :'. The 'Task :' field is significantly larger than the others. Below the container is a rounded rectangular button labeled 'Send'.

Figure 4.2: Assign task page

CHAPTER 5: Summary, Conclusion & Recommendation

5.0 Summary

In the course of the system design and implementation, the developer designed the system as an easy to use application. The Homepage of the administrative section of the software has links that would enable the project manager assign tasks, view deadlines, create reports and view feedback from staff on completed projects through the inbox.

Regular staff on the other hand, have their own view with restricted links that limits them to certain information and actions. For instance only a top management can access the management view. The developer was able to create these views in order to control access and information.

The new system design made use of email notification to alert employees of task allocated to them with deadlines; broadcast confidential information to staff according their access privileges.

The hybrid archiving option chosen by the developer is to ensure that a local server is maintained on premises that would house local company data. The software is structured in such a way that only a management staff can approve sensitive data to be uploaded to a remote server.

The choice of development tools was chosen after a thorough analysis of how best the system can run. The developer decided to create the application as a 3-tier application. She made use of JSP to create dynamic pages and html to create static pages, The JSP pages enables java codes to be run and content displayed on the JSP pages. Apache Tomcat was used as deployment container and web server, and finally Mysql database was used.

After the deployment and testing of the proposed system the developer was able to

- I. Successfully deploy the software application and use the forms to assign task to staff, of whom email applications were sent immediately the form submitted.
- II. Access the inbox to see updates of people's task and their deadlines
- III. View monthly report on the activities of staff.
- IV. Send mass information to all staff in the organization in just a second.
- V. Upload data to a remote data after it had been certified fit for online transfer.
- VI. Regular staff meetings were replaced by detailed emails messages copying all the staff involved.

5.1 Areas of Application of Work

This project is applicable to small medium and large business organizations who are in need of a control system that can help solve internally communication issues, task management and email management as well as archiving solution to email management.

5.2 Suggestions for Further Study

This project work was done within a limited time range and deployed with the development tools chosen by the developer to be run locally in an organization on

a local server but when required to upload to remote server the internet was used. The Software could be designed to using other development tools and can be embedded to existing corporate software.

5.3 Recommendations

In software development there is always a need for future modification; the software was developed in such a way that it can be deployed to existing software and used alongside for the smooth running of an organization. I therefore recommend this software to be deployed as an embedded application and an existing software.

5.4 Conclusion

The use of an Email system in businesses and organizations is rapidly becoming a mainstay in modern business practises. The email can be utilized in areas of task management and internal and remote communication. This project work has the view that the proposed system when deployed in an organization can be help cub problems that occur during business expansion.

REFERENCES

- [1] Steve Chew, "Evolution of Email": Microsoft Exchange Conference (MEC), 31st March 2014. Austin
- [2] PC World Communications, April 2000. <http://www.cworld.com/pcworldtoday/article/0,1510,16045+1+0,00.html>.
- [3] E. C. Gbandi, PhD, G. Amisshah, PhD, "Financing Options For Small and Medium Enterprises (smes) in nigeria". *European Scientific Journal January 2014* edition vol.10, No 1 ISSN: 1857 ó 7881 (Print) e - ISSN 1857- 7431
- [4] B. A NGWU ONUGU (FIMC, FICA), "A Small and Medium Enterprises (SMEs) in Nigeria: Problems and Prospects": PH.D Dissertation, St. Clements University, Switzerland. 2005
- [5] Prof. Banji Oyelaran-Oyeyinka, "SME: Issues, Challenges and Prospects" FSS 2020 International Conference. The Open University, Nigeria ()
- [6] Sona Mardikyan, "Analyzing the Usage of IT in SMEs" IBIMA Publishing: Communications of the IBIMA, <http://www.ibimapublishing.com/journals/CIBIMA/cibima.html>, Vol. 2010 (2010), Article ID 208609, DOI: 10.5171/2010.208609
- [7] David Wood, "Programming Internet Email" [eBook] ,O'Reilly Media Publishing, August 1999 ISBN 1-56592-479-7
- [8] Osterman Research, "Comparing the Cost of Leading Email Systems" Whitepaper ©2011.
- [9] Randy Franklin Smith "61% of SMES use Email Archiving in-house - What about others" Email Security Whitepapers. (2013)
- [10] Richard Flynn, "Different Emails Systems explained", blog. Thursday 22 December 2005. <http://richardflynn.net/blog/view/different-email-systems-explained>
- [11] Osterman Research "The TCO of Enterprise Email" Whitepaper ©2012
- [12] William Aspray, Paul Ceruzzi, Thomas Haigh. "Protocols for Profit: Web and Email Technologies as Product and Infrastructure", MIT Press, 2007.

- [13] Enric Olive "Open networks" ©2010
- [14] Cisco Systems "Remote-Access VPNs: Business Productivity, Deployment, and Security Considerations" 2008, Whitepaper <http://www.cisco.com/go/sslvpn>
- [15] Charles Edge 318 TECH JOURNAL, The VPN September 23rd, 2009.
<http://techjournal.318.com/general-technology/the-vpn/>
- [16] Paul Ferguson, "What is a VPN" The Internet Protocol Journal- Volume 1 No 1, *Cisco Systems and Geoff Huston, Telstra*
- [17] EPA, "Exchange: Hosted or On-Premises" Whitepaper.
<http://www.epacloud.com/>
- [18] Rogers S. Pressman, "Software Engineering: A Practitioner's Approach". McGraw-Hill series in Computer science 6th Edition. Publisher: Elizabeth A. Jones, 2005
- [19] Richard Flynn, "Different Email Systems explained", blog. Thursday 22 December 2005. <http://richardflynn.net/blog/view/different-email-systems-explained>
- [20] Osterman Research, "Why the cloud is not killing the On-premises email market". White Paper. April 2011
- [21] Amy Nutt "What are the benefits of using exchange server" Articleezine Magazine

APPENDIX

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="toppackage.SendMail" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Manager tool</title>
</head>
<body>
    <h1>Assign Task Page-Manager Tool</h1>

    <form action="ave.jsp" method="post">
        <table border="0" width="35%" align="center">
            <tr>
                <td width="50%">Task Title </td>
                <td><input type="text" name="title"
size="50"/></td>
            </tr>
            <tr>
                <td>Employee </td>
                <td><input type="text" name="employee"
size="50"/></td>
            </tr>
            <tr>
```

```

        <td>Date </td>
        <td><input type="text" name="date"
size="50"/></td>
    </tr><tr>
        <td>Email </td>
        <td><input type="text" name="email"
size="50"/></td>
    </tr><tr>
        <td>Deadline</td>
        <td><input type="text" name="deadline"
size="50"/></td>
    </tr>
    <tr>
        <td>Task </td>
        <td><textarea rows="10" cols="39" name="task"
size="250"></textarea> </td>
    </tr>
    <tr>
        <td colspan="2" align="center"><input
type="submit" value="Send"/></td>
    </tr>
</table>

</form>
</body>
</html>

<%@ page language="java"
    contentType="text/html; charset=windows-1256"
    pageEncoding="windows-1256"
%>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=windows-1256">
    <title>Login Page</title>
  </head>

  <body>
    <form action="LoginServlet">

      Please enter your username
      <input type="text" name="un"/><br>

      Please enter your password
      <input type="text" name="pw"/>

      <input type="submit" value="submit">

    </form>
  </body>
</html>

```

```

package toppackage;

import java.util.Properties ;
import javax.mail.Message ;
import javax.mail.MessagingException ;

```

```

import javax.mail.PasswordAuthentication ;
import javax.mail.Session ;
import javax.mail.Transport ;
import javax.mail.internet.InternetAddress ;
import javax.mail.internet.MimeMessage ;

public class SendMail {

    public String result;

    public void sending(String host, String port,
        final String username, final String password, String
toAddress,
        String subject, String message) throws
        MessagingException {

        Properties props = new Properties() ;
        props.put ( "mail.smtp.auth" , "true" ) ;
        props.put ( "mail.smtp.starttls.enable" , "true" ) ;
        props.put ( "mail.smtp.host" , host);
        props.put ( "mail.smtp.port" , port);
        Session session = Session.getInstance ( props,
        new javax.mail . Authenticator () {
        protected PasswordAuthentication getPasswordAuthentication
() {
        return new PasswordAuthentication ( username, password ) ;

```

```

    }
    });
    try {
        Message messageM = new MimeMessage ( session ) ;
        messageM. setFrom ( new InetAddress ( username ) ) ;
        messageM. setRecipients ( Message. RecipientType . TO ,
        InetAddress. parse ( toAddress ) ) ;
        messageM. setSubject( subject ) ;
        messageM. setText ( message ) ;
        Transport. send ( messageM ) ;
        result = "You have succeeded";
        System . out . println ( result ) ;
        } catch ( MessagingException e ) {
        throw new RuntimeException ( e ) ;
        }
    }
}
}
}

<%@ page language="java" contentType="text/html; charset=ISO-
8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.*"%>
<%@page import="toppackage.SendMail" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-1">
<title>Saving process</title>
</head>
<body>

```


<%

```
String Tasktitle=request.getParameter("title");
String Date=request.getParameter("date");
String Deadline=request.getParameter("deadline");
String Employee=request.getParameter("employee");
String Task=request.getParameter("task");
String Email=request.getParameter("email");
try
{
String url = "jdbc:mysql://localhost/test";
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection(url,"root","");
Statement st=con.createStatement();
st.executeUpdate("insert into tasks
(tasktitle,date,deadline,employee,task,email)
values('"+Tasktitle+"','"+Date+"','"+Deadline+"','"+Employee+"',
'"+Task+"','"+Email+"')");
}
catch(Exception ex)
{
out.println(ex.getMessage());
}

try
{
String mSql;
ResultSet rs;
String url = "jdbc:mysql://localhost/test";
```

```

Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection(url, "root", "");
Statement st=con.createStatement();
mSql="SELECT * FROM tasks where email='"+Email+"'";
rs=st.executeQuery(mSql);
if(rs.next()){

    String CopiedEmail= rs.getString("email");
    String CopiedTasktitle=rs.getString("tasktitle");
    String CopiedContent= rs.getString("task");
    out.println("Email Notification has been sent to " +
CopiedEmail);

    SendMail obj = new SendMail();

obj.sending("smtp.gmail.com", "587", "tochiezeanwu@gmail.com", "ezu
zu86", CopiedEmail, CopiedTasktitle, CopiedContent);
}

}
catch(Exception ex)
{
out.println(ex.getMessage());
}

%></body>
</html>

package toppackage;

```

```
import java.sql.*;
//import java.util.*;

public class ConnectionManager {

    static Connection con;

    static String url;

    public static Connection getConnection()
    {

        try
        {
            String url = "jdbc:mysql://localhost/test";
            // assuming "DataSource" is your DataSource name

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

            try
            {
                con = DriverManager.getConnection(url,"root","");

                // assuming your SQL Server's username is "username"
                // and password is "password"
```

```

    }

    catch (SQLException ex)
    {
        ex.printStackTrace();
    }
}

catch(ClassNotFoundException e)
{
    System.out.println(e);
}

return con;
}
}

package toppackage;

//import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

```

```

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {

/**
 *
 */
    private static final long serialVersionUID = 1L;

    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException,
        java.io.IOException {

    try
    {

        UserBean user = new UserBean();
        user.setUserName(request.getParameter("un"));
        user.setPassword(request.getParameter("pw"));

        user = UserDAO.login(user);
    }
}

```

```
    if (user.isValid())
    {

        HttpSession session = request.getSession(true);
        session.setAttribute("SessionUser",user);
        response.sendRedirect("UserLogged.jsp"); //logged-in
page
    }

    else

        response.sendRedirect("InvalidLogin.jsp"); //error
page
    }

catch (Throwable theException)
{
    System.out.println(theException);
}

}

}
```

```
package toppackage;
```

```

import java.util.Properties ;
import javax.mail.Message ;
import javax.mail.MessagingException ;
import javax.mail.PasswordAuthentication ;
import javax.mail.Session ;
import javax.mail.Transport ;
import javax.mail.internet.InternetAddress ;
import javax.mail.internet.MimeMessage ;

    public class SendMail {

        public String result;

        public void sending(String host, String port,
            final String username, final String password, String
toAddress,
            String subject, String message) throws
            MessagingException {

            Properties props = new Properties() ;
            props.put ( "mail.smtp.auth" , "true" ) ;
            props.put ( "mail.smtp.starttls.enable" , "true" ) ;
            props.put ( "mail.smtp.host" , host);

```

```

props. put ( "mail.smtp.port" , port);
Session session = Session. getInstance ( props,
new javax. mail . Authenticator () {
protected PasswordAuthentication getPasswordAuthentication
() {
return new PasswordAuthentication ( username, password ) ;
}
}) ;
try {
Message messageM = new MimeMessage ( session) ;
messageM. setFrom ( new InternetAddress (username)) ;
messageM. setRecipients ( Message. RecipientType . TO ,
InternetAddress. parse ( toAddress)) ;
messageM. setSubject( subject) ;
messageM. setText ( message ) ;
Transport. send ( messageM) ;
result = "You have succeeded";
System . out . println ( result ) ;
} catch ( MessagingException e ) {
throw new RuntimeException ( e ) ;
}
}
}
}

```



```
package toppackage;

public class UserBean {

    private String username;
    private String password;
    private String firstName;
    private String lastName;
    public boolean valid;

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String newFirstName) {
        firstName = newFirstName;
    }

    public String getLastName() {
        return lastName;
    }
}
```

```
public void setLastName(String newLastName) {  
    lastName = newLastName;  
}
```

```
public String getPassword() {  
    return password;  
}
```

```
public void setPassword(String newPassword) {  
    password = newPassword;  
}
```

```
public String getUsername() {  
    return username;  
}
```

```
public void setUsername(String newUsername) {  
    username = newUsername;  
}
```

```
public boolean isValid() {  
    return valid;  
}
```

```

    }

    public void setValid(boolean newValid) {
        valid = newValid;
    }
}

package toppackage;

//import java.text.*;
//import java.util.*;
import java.sql.*;

public class UserDAO
{
    static Connection currentCon = null;
    static ResultSet rs = null;

    public static UserBean login(UserBean bean) {

        //preparing some objects for connection
        Statement stmt = null;

```

```

String username = bean.getUsername();
String password = bean.getPassword();

String searchQuery =
    "select * from users where username='"
        + username
        + "' AND password='"
        + password
        + "'";

// "System.out.println" prints in the console; Normally used
to trace the process

System.out.println("Your user name is " + username);
System.out.println("Your password is " + password);
System.out.println("Query: "+searchQuery);

try
{
    //connect to DB
    currentCon = DriverManager.getConnection();
    stmt=currentCon.createStatement();
    rs = stmt.executeQuery(searchQuery);
    boolean more = rs.next();
}

```

```

        // if user does not exist set the isValid variable to
false
        if (!more)
        {
            System.out.println("Sorry, you are not a registered
user! Please sign up first");
            bean.setValid(false);
        }

        //if user exists set the isValid variable to true
else if (more)
        {
            String firstName = rs.getString("fname");
            String lastName = rs.getString("lname");

            System.out.println("Welcome " + firstName);
            bean.setFirstName(firstName);
            bean.setLastName(lastName);
            bean.setValid(true);
        }
    }

    catch (Exception ex)
    {
        System.out.println("Log In failed: An Exception has
occurred! " + ex);
    }

```

```
}

//some exception handling
finally
{
    if (rs != null)    {
        try {
            rs.close();
        } catch (Exception e) {}
        rs = null;
    }

    if (stmt != null) {
        try {
            stmt.close();
        } catch (Exception e) {}
        stmt = null;
    }

    if (currentCon != null) {
        try {
            currentCon.close();
        } catch (Exception e) {
        }
    }
}
```

```
        currentCon = null;
    }
}

return bean;

}
}
```

APPENDIX B

Software Testing

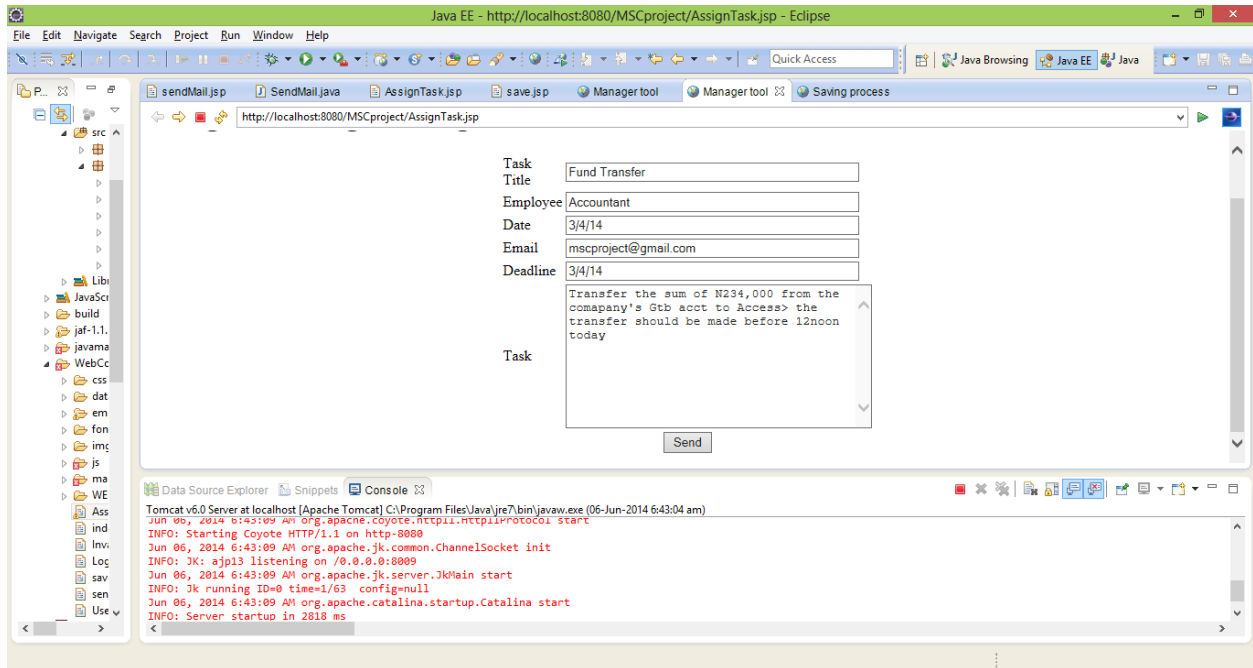


Figure 4.3: Snapshot of assign task page

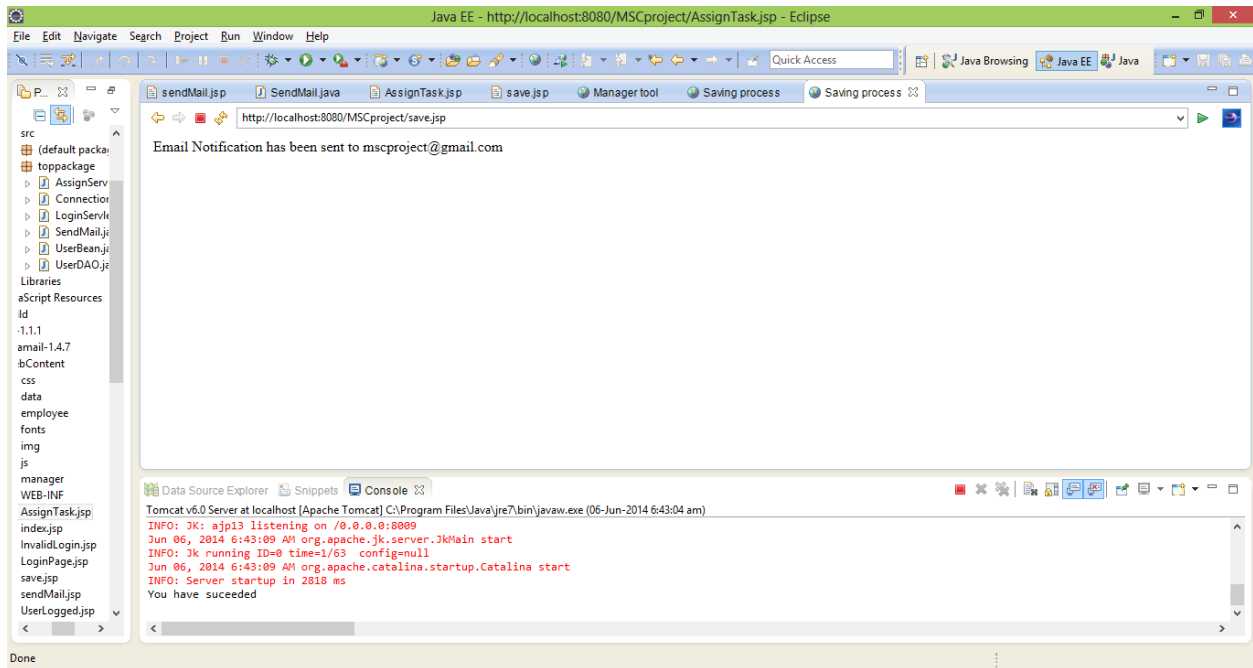
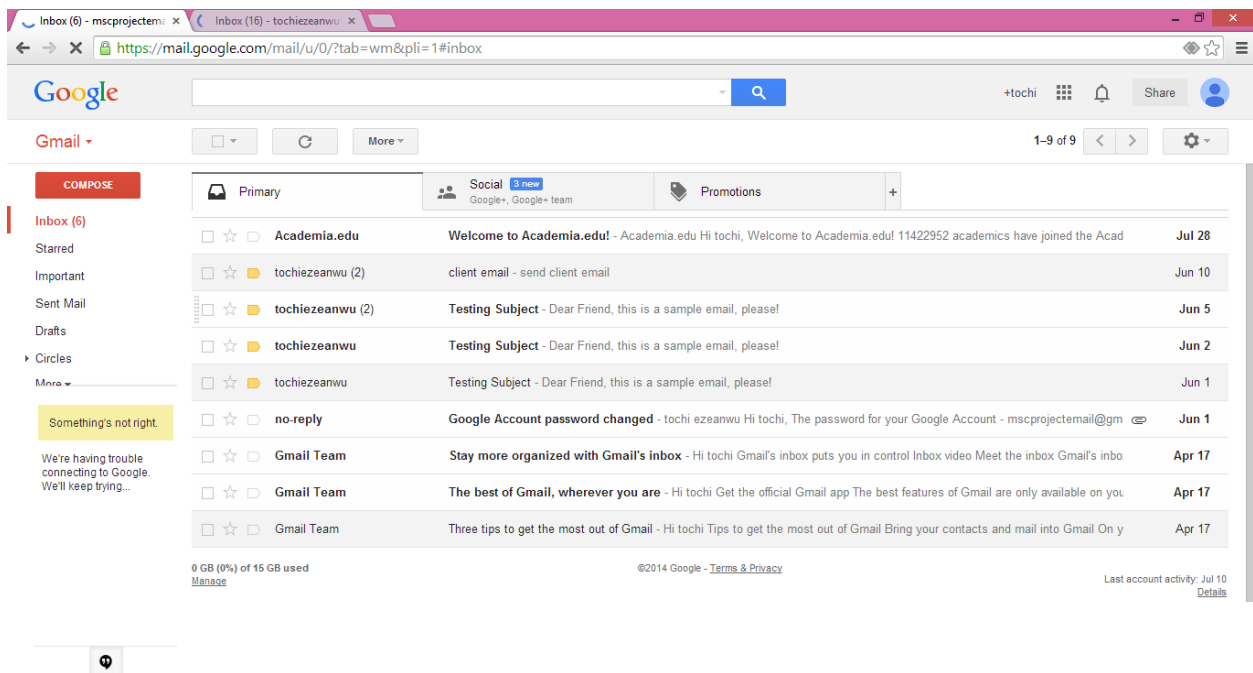


Figure 4.4: Snapshot showing email is sent



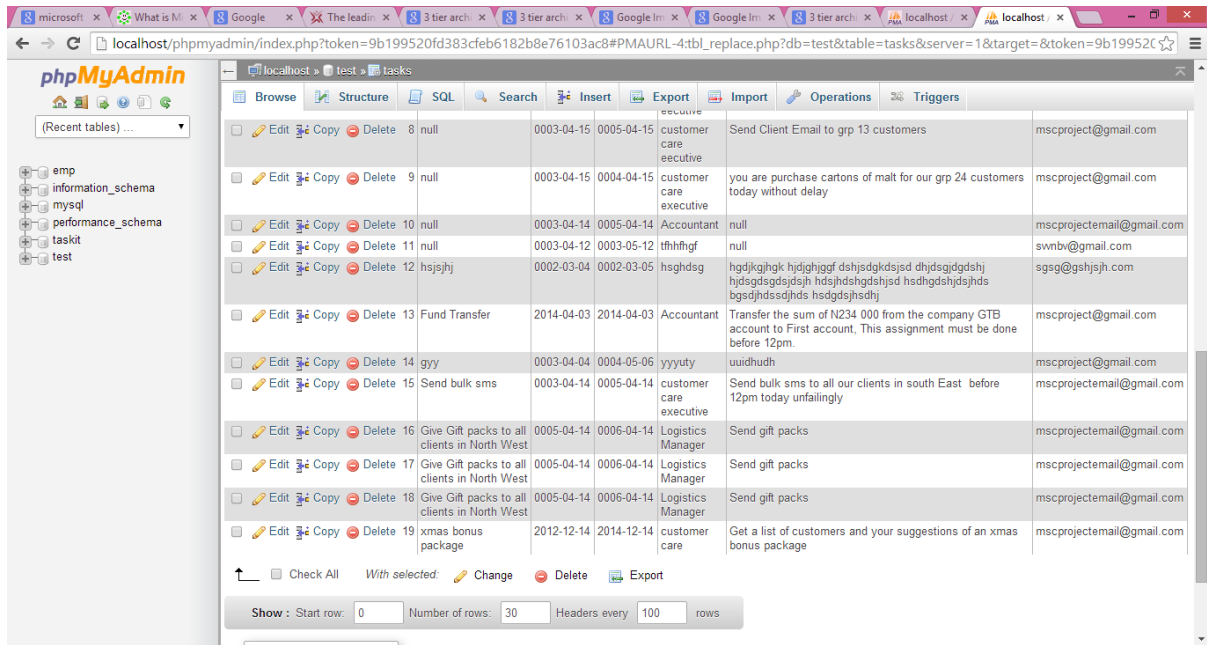


Figure 4.5: Snapshot showing submission into the database