GOVT.DEGREE COLLEGE
ARMOOR (T.S.)

Student Study Project Report
on

EMAIL SYSTEM

For the Academic Year 2020-21

Submitted by:

CH.AKHILA
Roll No: 19055010468009
CH.NAVEEN
Roll No: 19055010468010
CH.PRIYANKA
Roll No: 19055010468008
U.SUPRIYA
Roll No: 19055010468029
U.GOUTHAMI
Roll No: 19055010468028

Guided by:

Lecturer. J.SUDEEP KUMAR
Department of computer science

Head of the Department:

Asst.Prof. A.RAJESH
Department of Chemistry

Table of contents

ABSTRACT

ABOUT THE PROJECT

Businesses usually adopt a common sense approach when it comes to spending the IT budget with cost being a major influence in purchasing decisions. However, when it comes to choosing a mail server, many businesses seem to pay less attention to the costs and, as a result, end up spending far more money than is necessary.

Today, email is absolutely mission-critical. Communication and collaboration keep your business running. Email and electronically enabled collaboration have become so embedded in normal day-to-day operations that many businesses simply could not function without them. Many businesses, however, have found that the cost of providing employees with the latest in messaging and collaboration technology is rapidly escalating. To meet modern business needs, mail servers have had to become more complex – and with that additional complexity come additional management burdens and costs. Furthermore, some mail servers have an upgrade process that is both extremely complex and extremely costly and which may necessitate the purchase of replacement server hardware. Combined, these factors place a considerable drain on corporate resources. The problem is especially severe for small and medium sized businesses (SMBs) which usually do not have access to the same financial or technical resources as large enterprises.

Email-System Is An Application Program That Sends Electronic Message From One Computer To Another. We will be developing our Project to keep in Mind the Problem of Organization and also We Will Try To Minimize Cost for Organization.

Introduction

INTRODUCTION

This project deals with the Mailing System. This project is having different modules like new User creation form named it as a Sign-Up form and already existing user can logged into the Mailing System named it as a Sign In form.

Email System(or Internet Mailing System) which has been privatized and is existing in different forms like Hotmail. Free mail. Cyber mail, Mainly The project will give the easy way to create a new account and sending mails with free of cost.

Advantages:

The user of Email System is given a unique login id and must give the correct password. It gives total security for us. So unauthorized user can't allow to see our messages. Even if the user forgets his/her password reminding facility by which the user can recollect the password and log into the system.

The advantage of the this system is it's security feature allowing only registered users to access the system and preventing any hackers, unauthorized users.

Existing System with Limitations:

Presenting Intranet Mailing is manually providing services to employees of departments of an Organization. Employees have to go departments to know some particular information. Sometimes information is passed by manually between departments. This manual system will take time to pass the information and sometimes it causes lost of information also. This causes lost of employee time also.

Proposed System Features:

Now a day the organizations are growing fastly and are increasing in size also. So there organizations are divided into departments. In the fast growing world the information is need as fast as possible. This can be accomplished by passing the information quickly. Quick passing of mails is not possible in load manual systems. Because in manual system the mails are passed through persons from one department to another. But it takes mush time and risk also. This leads the inconsistency of information. So we need a system, which is both quick and accurate. This can be achieved by mailing system.

In present organization structure most of the work is done using software applications. In order to improve service for customers we need effective applications. Similarly considering need of work flow we need intranet mailing application.

SPECIFICATION REQUIREMENT

SPECIFICATION REQUIREMENT

Requirement analysis for web applications encompasses three major tasks: formulation, requirements gathering and analysis modeling. During formulation, the basic motivation and goals for the web application are identified, and the categories of users are defined. In the requirements gathering phase, the content and functional requirements are listed and interaction scenarios written from end-user's point-of-view are developed. This intent is to establish a basic understanding of why the web application is built, who will use it, and what problems it will solve for its users.

Software requirement Specification:

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system, which enables the user to interact with several physical hardware devices.

The minimum software requirement specifications for developing this project are as follows:

Operating System	:	**Window 2000, XP,Win 7,Win 8.**
Presentation layer	:	**Java, Servlets.**
Database	:	**Oracle**
Documentation Tool	:	**Ms Office**

Hardware Requirement Specification:

The collection of internal electronic circuits and external physical devices used in building a computer is called the Hardware. The minimum hardware requirement specifications for developing this project are as follows:

Processor	**:** Standard processor with a speed of 1.6 GHz or more
RAM	**:** 256 MB RAM or more
Hard Disk	**:** 20 GB or more
Monitor	**:** Standard color monitor

TECHNOLOGIES

USED

TECHNOLOGIES USED

Servlets:-

A servlet is a java programming language class that is used to extend the capabilities of servers that host applications access via a request-response programming mode. Servlets are Java technology's answer to Common Gateway Interface (CGI) Programming. They are programs that run on a Web server, acting as middle layer between request coming from a Web browser or other HTTP client and databases of applications on the HTTP server.

Read any data sent by the user: This data usually entered in a form on a Web page, but could also come from a java applet or a custom HTTP client program.

Look up any other information about the request that is embedded in the HTTP request: This information includes details about browser capabilities, cookies, the host name of the requesting client, and so froth.

Generate the results: This process may require talking to a database, executing an RMI or CORBA call, invoking a legacy application, or computing the response directly.

Format the results inside a document: In most cases, this involves embedding the information inside an HTML page.

Set the appropriate HTTP response parameters: This means telling the browser what type of document is being returned (e.g.HTML), setting cookies and caching parameters, and other such tasks.

Send the document back to the client: This document may be sent in text format (HTML), binary format (GIF images), or even in a compressed format like gzip that is layered on top of some other underlying format.

The Javax.servlet and javax.servlet.http packages provide interfaces and classes for writing servlets. All servlets must implement the Servlet interface, which defines life-cycle methods. When implementing a generic service, you can use or extend the GenericServlet class provided with the java Servlet API. The HttpServlet classes provide methods, such as doGet and do Post, for handling HTTP-specific services.

To be a servlet, a class should extend HTTPServlet and override doGet or do Post (or both), depending on whether the data is being sent by GET or by POST. These methods take two arguments: An HttpServletRequest and an HttpServletResponse.The HttpServletRequest have methods that let you find out about incoming information such as FORM data, HTTP request headers, and the like. Finally, note that doGet and do Post are called by the service method, and sometimes you may want to override service directly.

Servlet Life Cycle:

The life cycle of a servlet is controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the container performs the following steps.

1. If an instance of the servlet does not exist, the Web container:

⇨ Loads the servlet class.

Creates an instance of the Servlet class.

Initializes the servlet instance by calling the init method.

2. Invokes the service method, passing request and response objects.

If the container needs to remove the servlet, it finalizes the servlet by calling the servlet's destroy method.

Cookies

Cookies are small bits of textual information that a Web server sends to a browser and that the browser returns unchanged when visiting the same Web site or domain later

Browsers generally only accept 20 cookies per site and 300 cookies total, and each cookie is limited to 4KB, cookies cannot be used to fill up someone's disk or launch other denial of service attacks.

The Servlet Cookie API

To send cookies to the client, a servlet would create one or more cookies with the appropriate names and values via new Cookie (name, value), set any desired optional attributes via cookie.setXxx,and add the cookies to the response headers via response.addCookie(cookie).To read incoming cookies, call request.getCookies(), which returns an array of Cookie objects.

Session Management

Many applications require that a series of requests from a client be associated with one another. Sessions are represented by an Http Session object. A session can be accessed by calling the get Session () method of a request object. This method returns the current session associated with this request, or, if the request does not have a session, it creates one. The timeout period can be accessed by using a session's [get\set] Max Inactive Interval methods.

Session Tracking

A Web container can use several methods to associate a session with a user, all of which involve passing an identifier between the client and the server. The identifier can be maintained on the client as a cookie, or the Web component can include the identifier in every URL that is returned to the client.

In fact, on many servers, they use cookies if the browser supports them, but automatically revert to URL-rewriting when cookies are unsupported or explicitly disabled.

The Session Tracking API

Using sessions in servlets is quite straightforward, and involves looking up the session object associated with the current request, creating a new session object when necessary, looking up information associated with a session, storing information in a session, and discarding completed or abandoned sessions.

Oracle 10g Express Edition

Oracle® Database Express Edition

1 Logging in as the Database Administrator

The first thing you need to do is to log in as the Oracle Database XE Administrator. Follow these steps:

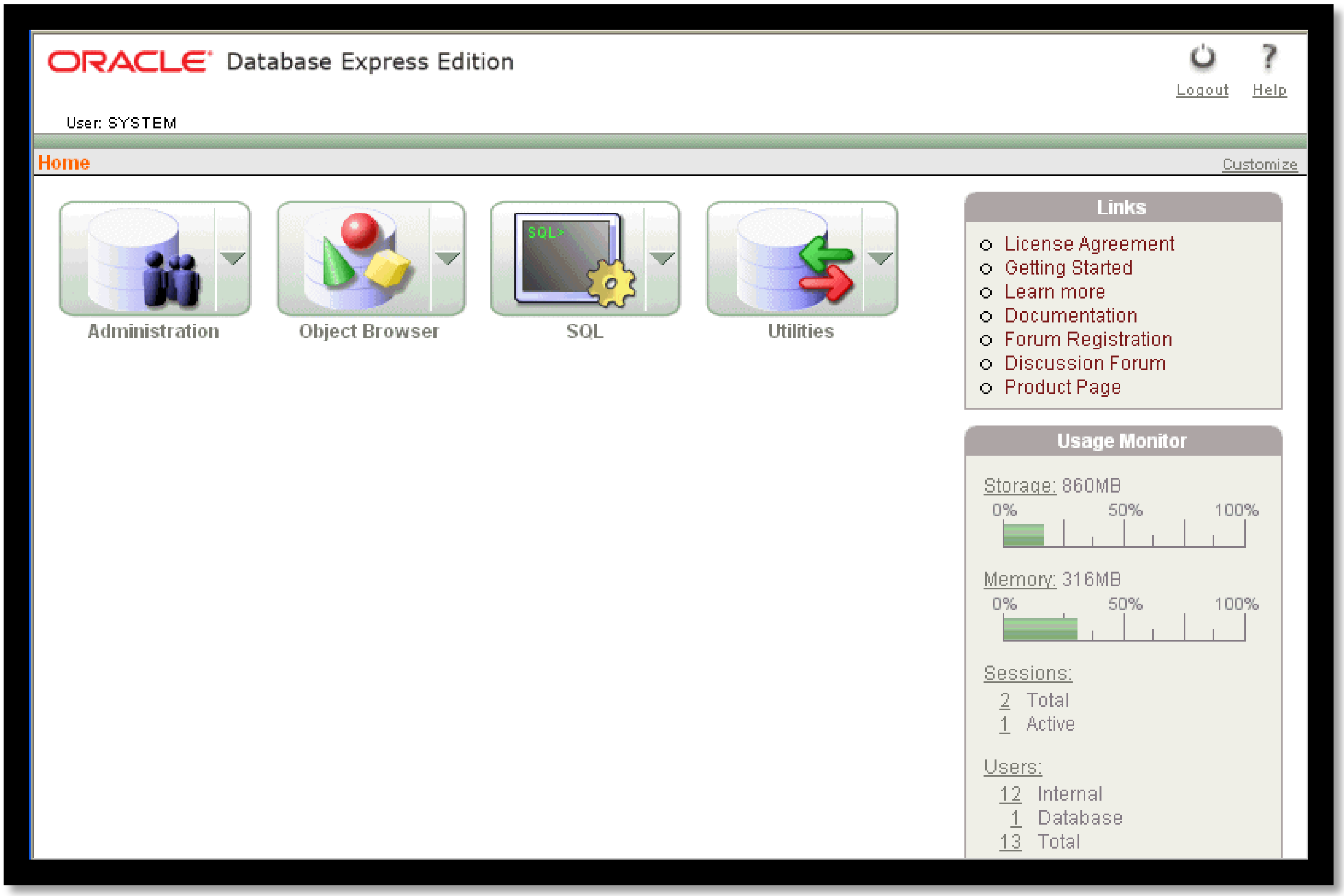
1. Open the Database Home Page login window:
    - On Windows, from the **Start** menu, select **Programs** (or **All Programs**), then **Oracle Database 10g Express Edition**, and then **Go To Database Home Page**.
    - On Linux, click the **Application** menu (on Gnome) or the **K** menu (on KDE), then point to **Oracle Database 10g Express Edition**, and then **Go To Database Home Page**.

At the Database Home Page login window, enter the following information:

- **Username:** Enter system for the user name.
- **Password:** Enter the password that was specified when Oracle Database XE was installed.

Click **Login**.

The Oracle Database XE home page appears.



Description of the illustration gs_home_page.gif

2 Unlocking the Sample User Account

To create your application, you need to log in as a database user. Oracle Database XE comes with a sample database user called HR. This user owns a number of database tables in a sample schema that can be used to create applications for a fictional Human Resources department. However, for security reasons, this user's account is locked. You need to unlock this account before you can build a sample application.

To unlock the sample user account:

1. Make sure you are still logged on as the database administrator, as described in the previous section.
2. Click the **Administration** icon, and then click **Database Users**.

3. Click the **HR** schema icon to display the user information for HR.

Description of the illustration gs_hr_icon.gif

4. Under Manage Database User, enter the following settings:
    - **Password** and **Confirm Password**: Enter hr for the password.
    - **Account Status**: Select **Unlocked**.
    - **Roles**: Ensure that both **CONNECT** and **RESOURCE** are enabled.

Click **Alter User**.

Now you are ready to create your first application.

3 Logging in as the Sample User Account

To log in as the sample user account:

1. Log out from the database administrator account by clicking **Logout** in the upper right corner of the Database Home Page.
2. In the window, click **Login**.
3. In the Login window, enter hr for both the user name and password.
4. Click **Login**.

    The Database Home Page appears.

4 Creating a Simple Application

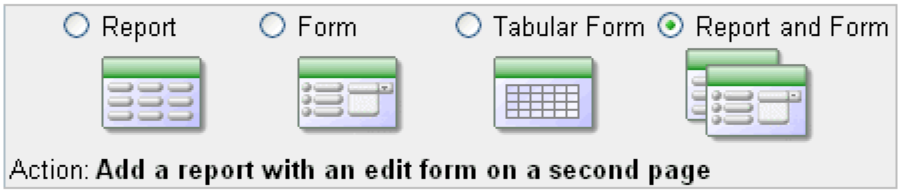
Creating an application is an easy way to view and edit your database data. You create this application based on the EMPLOYEES table, which is part of the HR schema.

To create an application based on the EMPLOYEES table:

1. On the Database Home Page, click the **Application Builder** icon.
2. Click the **Create** button.
3. Under Create Application, select **Create Application** and click **Next**.
4. Under Create Application:
    a. **Name**: Enter MyApp.
    b. Accept the remaining defaults.
    c. Click **Next**.

    Next, add pages to your application.

5. Under Add Page:
    a. For Select Page Type, select **Report and Form**.



○ Report ○ Form ○ Tabular Form ● Report and Form

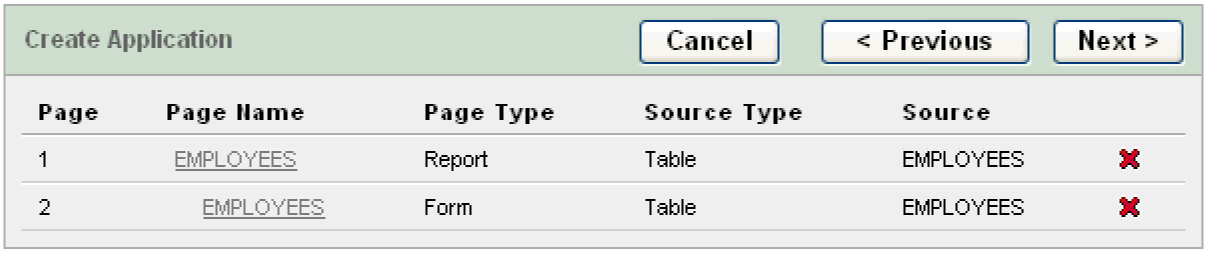
Action: Add a report with an edit form on a second page

Description of the illustration gs_report_and_form.gif

Notice that **Action** describes the type of page you are adding.

    b. Next to the **Table Name** field, click the up arrow, and then select **EMPLOYEES** from the Search Dialog window.
    c. Click **Add Page**.

Two new pages display at the top of the page, under Create Application.



Page	Page Name	Page Type	Source Type	Source	
1	EMPLOYEES	Report	Table	EMPLOYEES	✖
2	EMPLOYEES	Form	Table	EMPLOYEES	✖

Description of the illustration gs_two_new_pages.gif

    d. Click **Next**.
6. On the Tabs panel, accept the default (**One Level of Tabs**) and click **Next**.
7. On the Shared Components panel, accept the default (**No**) and click **Next**.

This option enables you to import shared components from another application. Shared components are common elements that can display or be applied on any page within an application.

8. For Authentication Scheme, Language, and User Language Preference Derived From, accept the defaults and click **Next**.
9. For the theme, select **Theme 2** click **Next**.

Themes are collections of templates that you can use to define the layout and style of an entire application.

10. Confirm your selections. To return to a previous wizard page, click **Previous**. To accept your selections, click **Create**.

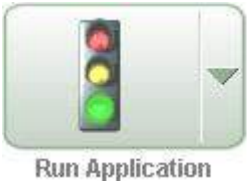
    After you click **Create**, the following message displays at the top of the page:

    Application created successfully.

5 Running Your New Application

To run your application:

1. Click the **Run Application** icon.

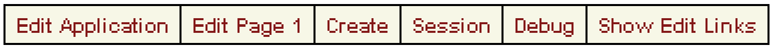
   Description of the illustration gs_run_ico_sm.gif

2. In the log in page, enter hr for both the **User Name** and **Password**.

   Your application appears, showing the EMPLOYEES table.

3. Explore your application.

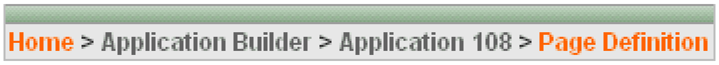
   You can query the EMPLOYEES table, if you want. To manage the application, use the Developer toolbar at the bottom on the page.

   Description of the illustration gs_d_toolbar.gif

   The Developer toolbar offers a quick way to edit the current page, create a new page, control, or component, view session state, or toggle debugging or edit links on and off.

4. To exit your application and return to Application Builder, click **Edit Page 1** on the Developer toolbar.
5. To return to the Database Home Page, select the **Home** breadcrumb at the top of the page.

   
   Home > Application Builder > Application 108 > Page Definition

   Description of the illustration gs_bread_myapp.gif

Congratulations! You have just created your first application using Oracle Database XE.

6 Using the Oracle Database XE Menus

You can use the Oracle Database XE menus to perform basic functions with Oracle Database XE. To see the menus, do the following:

- On Windows, from the **Start** menu, select **Programs** (or **All Programs**) and then **Oracle Database 10g Express Edition**.
- On Linux, click the **Application** menu (on Gnome) or the **K** menu (on KDE) and then point to **Oracle Database 10g Express Edition**.

The following menu items are available:

- **Get Help**: Displays the following selections:
    - **Go To Online Forum**: Displays the online forum for discussions about Oracle Database XE.
    - **Read Documentation**: Displays the Oracle Database XE documentation library on the Internet.
    - **Read Online Help**: Displays the Oracle Database XE online help. This help is only available if the database is started.
    - **Register For Online Forum**: Allows you to register for the Oracle Database XE online forum.
- **Backup Database:** In NOARCHIVELOG mode (the default), shuts down the database, backs it up, and then restarts it. In ARCHIVELOG mode, performs an online backup of the database. For more information on backups, refer to *Oracle Database Express Edition 2 Day DBA*.
- **Get Started:** Link to this tutorial.
- **Go To Database Home Page**: Displays the Oracle Database XE Home Page in your default browser. "Logging in as the Database Administrator" explains how to log into this home page as a database administrator.
- **Restore Database:** Shuts down and then restores the database to the most recent backup. For more information on restoring a database, refer to *Oracle Database Express Edition 2 Day DBA*.
- **Run SQL Command Line**: Starts the SQL Command Line utility for Oracle Database XE. To connect to the database, issue the following command at the SQL prompt that appears:
- connect *username*/*password*
-

where *username* is the user name, such as sys, system, or another account name, and *password* is the password that was assigned when Oracle Database XE was installed. The get help, you can enter the commandhelp at the SQL prompt, once you have connected to the database.

- **Start Database**: Starts Oracle Database XE. By default, the database is started for you after installation and every time your computer is restarted. However, if you think the database is not running you can use this menu item to start it.
- **Stop Database**: Stops Oracle Database XE.

Tomcat:-

Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process.

Tomcat 5 implements the Servlet 2.4 and Java Server Pages 2.0 specifications and includes many additional features that make it a useful platform for developing and deploying web applications and web services.

Directories and files:

$CATALINA_HOME represents the root of your Tomcat installation. When we say, "This information can be found in your $CATALINA_HOME/README.txt file" we mean to look at the README.txt file at the root of your Tomcat install.

These are some of the key tomcat directories, all relative to **$CATALINA_HOME**:

- **/bin** - Startup, shutdown, and other scripts. The *.sh files (for Unix systems) are functional duplicates of the *.bat files (for Windows systems). Since the Win32 command-line lacks certain functionality, there are some additional files in here.

- **/conf** - Configuration files and related DTDs. The most important file in here is server.xml. It is the main configuration file for the container.
- **/logs** - Log files are here by default.
- **/webapps** - This is where your webapps go.

INSTALLING TOMCAT

Installing Tomcat on Windows can be done easily using the Windows installer. **Installation as a service**: Tomcat will be installed as a Windows NT/2k/XP service no matter what setting is selected. Using the checkbox on the component page sets the service as "auto" startup, so that Tomcat is automatically startup when Windows starts. For optimal security, the service should be affected a separate user, with reduced permissions (see the Windows Services administration tool and its documentation).

- **Java location**: The installer will use the registry or the JAVA_HOME environment variable to determine the base path of the JDK or a JRE. If only a JRE (or an incorrect path) is specified, Tomcat will run but will be unable to compile JSP pages at runtime. Either all webapps will need to be precompiled (this can be easily done using the Tomcat deployed), or the lib\tools. Jar file from a JDK installation must be copied to the common\lib path of the Tomcat installation.

Architecture Overview

Server: In the Tomcat world, a Server represents the whole container. A **Server** element represents the entire Catalina servlet container. Therefore, it must be the single outermost element in the conf/server.xml configuration file. Its attributes represent the characteristics of the servlet container as a whole. Tomcat provides a default implementation of the Server interface, and this is rarely customized by users.

Engine

An Engine represents request processing pipeline for a specific Service. As a Service may have multiple Connectors, the Engine received and processes all requests from these connectors, handing the response back to the appropriate connector for transmission to the client.

Host

A Host is an association of a network name, e.g. www.yourcompany.com, to the Tomcat server. An Engine may contain multiple hosts, and the Host element also supports network aliases such as yourcompany.com and abc.yourcompany.com. Users rarely create custom Hosts because the Standard Host implementation provides significant additional functionality.

Service

A Service is an intermediate component which lives inside a Server and ties one or more Connectors to exactly one Engine. The Service element is rarely customized by users, as the default implementation is simple and sufficient: service interface.

Connector

A Connector handles communications with the client. There are multiple connectors available with Tomcat, all of which implement the Connector interface These include the Coyote connector which is used for most HTTP traffic, especially when running Tomcat as a standalone server, and the JK2 connector which implements the AJP protocol used when connecting Tomcat to an Apache HTTPD server. Creating a customized connector is a significant effort.

Context

A Context represents a web application. A Host may contain multiple contexts, each with a unique path. The Context interface may be implemented to create custom Contexts, but this is rarely the case because the Standard Context provides significant additional

functionality.

BEHAVIORAL DESCRIPTION

BEHAVIORAL DESCRIPTION

Data Flow:

There are 2 types of Dfd's they are

1. Context Level DFD
2. Top Level DFD

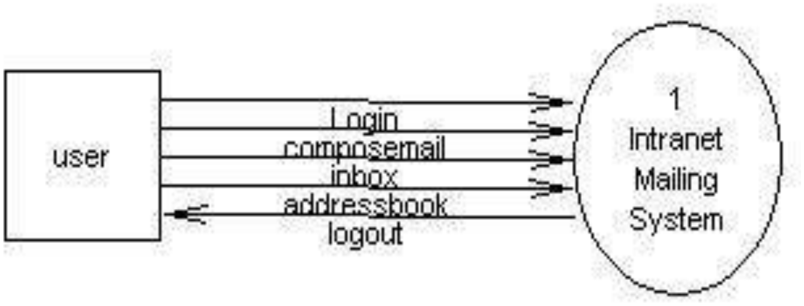
Context Level DFD:

In the Context Level the whole system is shown as a single process.

- No data stores are shown.
- Inputs to the overall system are shown together with data sources (as External entities).
- Outputs from the overall system are shown together with their destinations (as External entities).

4.1.1 DFD:

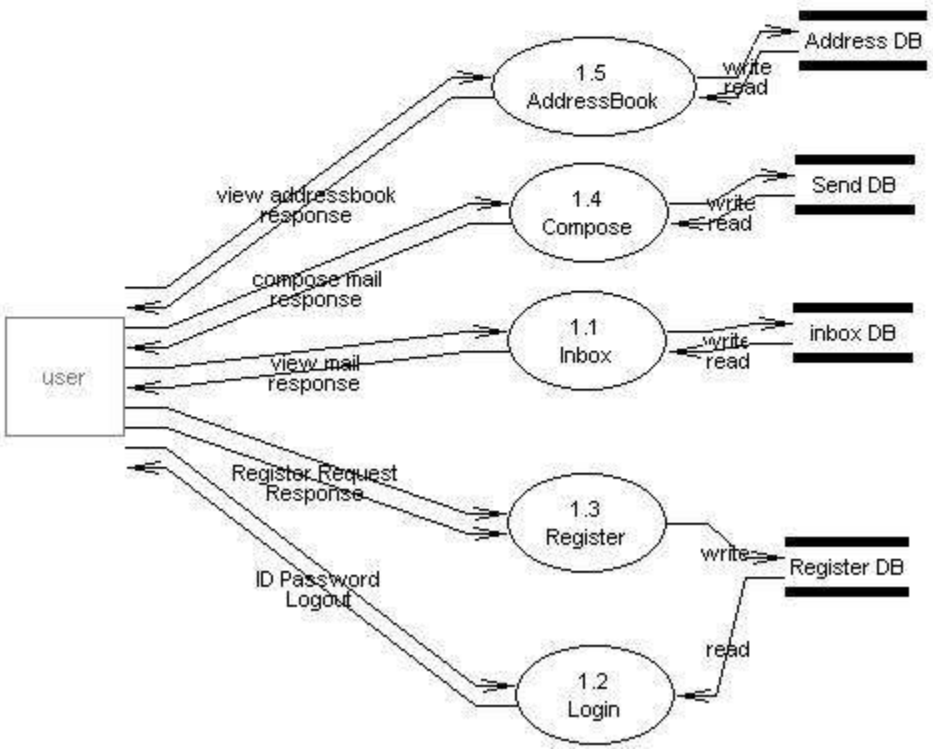
[1,1]



Top Level DFD:

The Top Level DFD gives the overview of the whole system identifying the major system processes and data flow. This level focuses on the single process that is drawn in the context diagram by ‘Zooming in’ on its contents and illustrates what it does in more detail.

[1,1]



4.1.2. Use Case Documentation:

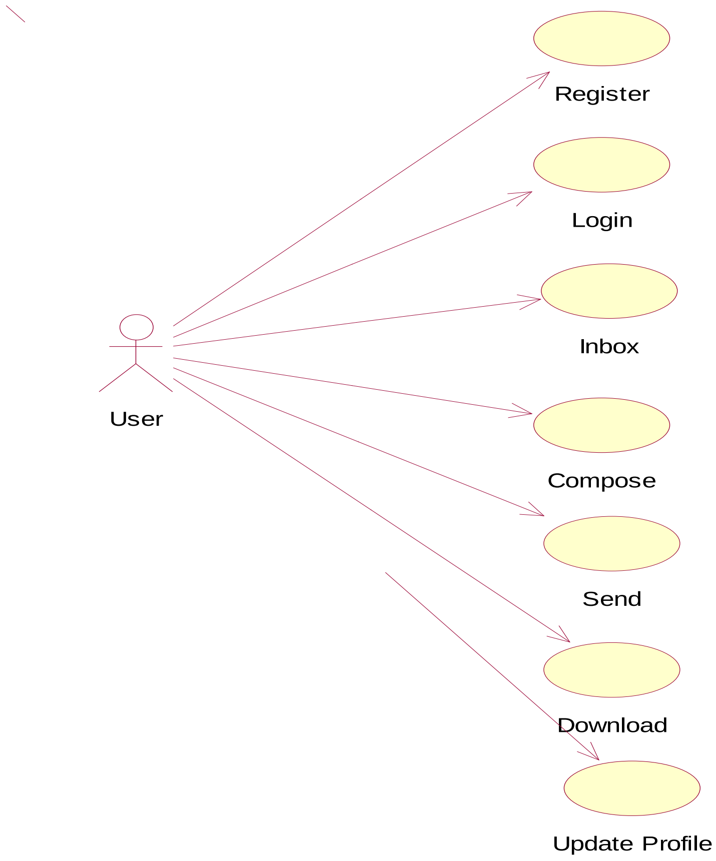
Use Case Diagram

- A use case diagram is a diagram that shows a set of use cases and actors and relationships.

Contents

- Use case commonly contain
  - Use cases
  - Actors
  - Dependency, generalization and association relationships

Overall Use Case



Process Flow

Activity Diagrams:

Activity Diagram:

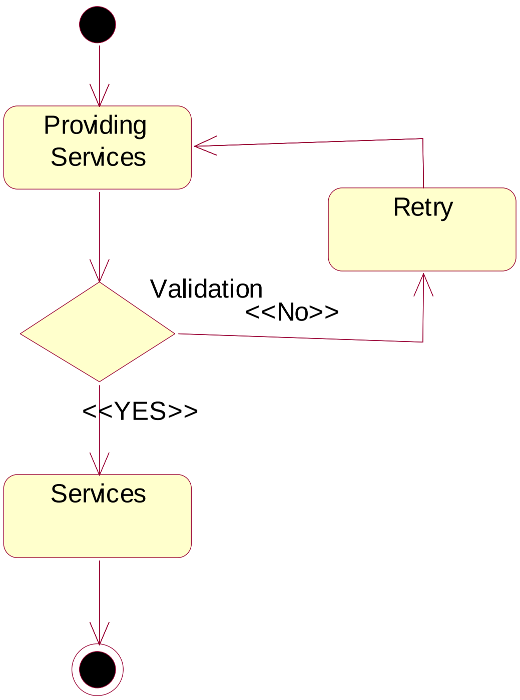
- An activity diagram shows the flow from activity to activity. An activity is an ongoing non- atomic execution within a state machine.
- Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in state of the system or the return of a value.

  Activity diagrams commonly contain

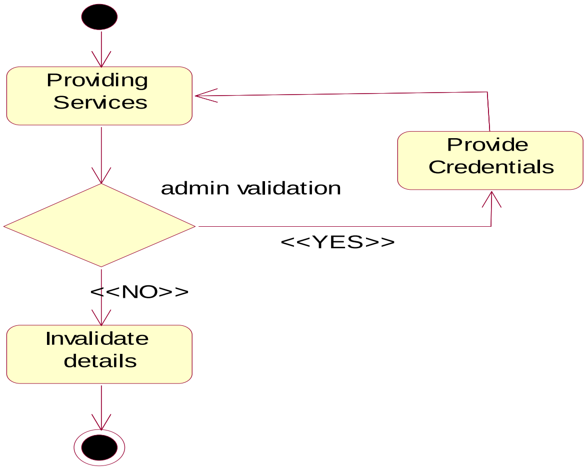
  - Activity states and action states
  - Transitions
  - Objects

- Like all other diagrams, activity diagrams may contain notes and constrains.

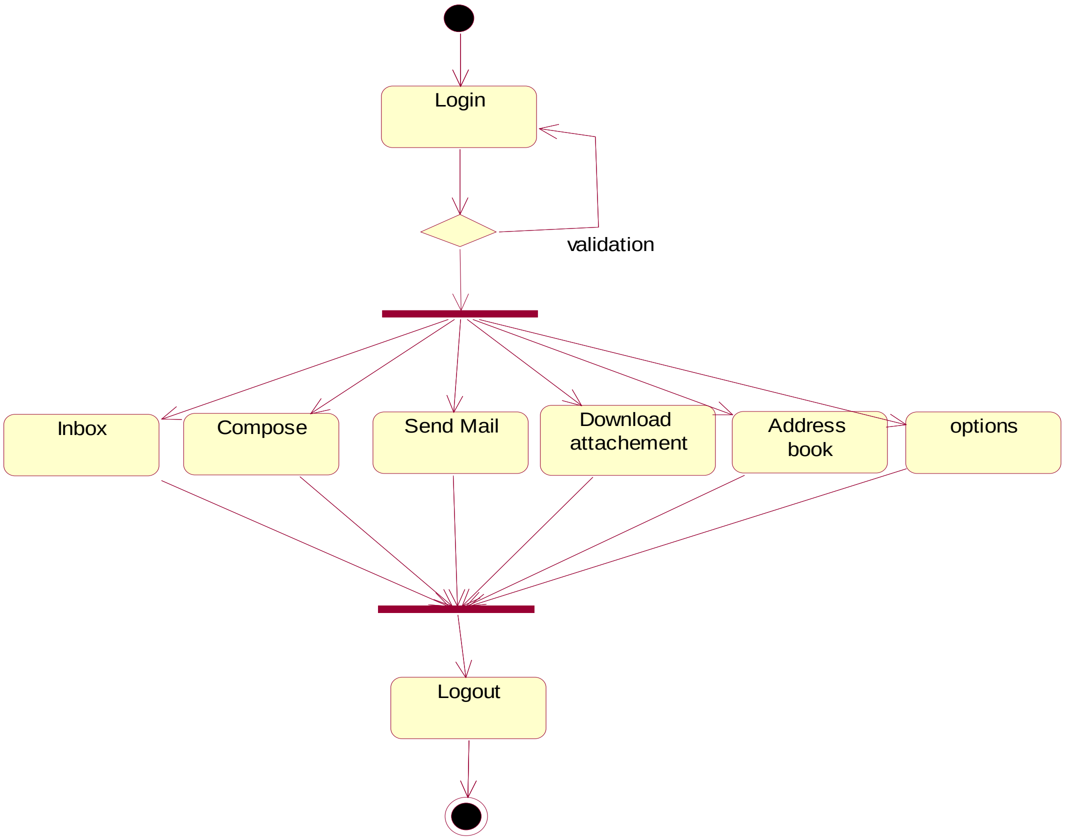
Login Process



Registration Process :



User Activity:



SYSTEM DESIGN

SYSTEM DESIGN

The main focus of the analysis phase of Software development is on "What needs to be done". The objects discovered during the analysis can serve as the framework or Design. The class's attributes, methods and association identified during analysis must be designed for implementation language. New classes must be introduced to store intermediate results during the program execution.

Emphasis shifts from the application domain o implementation and computer such as user interfaces or view layer and access layer. During analysis, we look at the physical entities or business objects in the system, that is, which players and how they cooperate to do the work of the application. These objects represent tangible elements of the business.

During the Design phase, we elevate the model into logical entities, some of which might relate more to the computer domain as people or employees. Here his goal is to design the classes that we need to implement the system the difference is that, at this level we focus on the view and access classes, such as how to maintain information or the best way o interact with a user or present information.

Design process:

During the design phase the classes identified in object-oriented analysis Must be revisited with a shift focus to their implementation. New classes or attribute and Methods must be an added for implementation purposes and user interfaces. The object-oriented design process consists of the following activities:

1. Apply design axioms to design classes, their attributes, methods, associations, structure

And protocols Refine and complete the static UML class diagram by adding details to the UML diagram. This step consists of following activities. *Refine attributes *Design methods and protocols by utilizing a UML activity diagram to represent the method's algorithms.

*Refine associations between classes

*Refine class hierarchy and design with inheritance

*Iterate and refine again

2. Design the access layer

- Create mirror classes: For every business class identified and created. For
  example, if there are three business classes, create three access layer classes.

- Identify access layer class relationships.
- Simplify classes and their relationships: The main goal here is to eliminate
  redundant classes and structures.

  *Redundant classes: Do not keep two classes that perform similar translate results

  activities. Simply select one and eliminate the other.

  *Method classes: Revisit the classes that consist of only one or two methods to see if they can be eliminated or combined with existing classes.

- Iterate and refine again.

Define the view layer classes

- Design the macro level user interface, identifying view layer objects.
- Design the micro level user interface, which includes these activities:
  - * Design the view layer objects by applying the design axioms and corollaries.
  - * Built a prototype of the view layer interface.
    - Test usability and user satisfaction
    - Iterate and refine.

3. Iterate refine the whole design process. From the class diagram, you can begin to extrapolate which classes you will have to built and which existing classes you can reuse. As you do this, also begin this, also begin thinking about the inheritance structure. If you have several classes that seem relates but have specific differences.

Design also must be traceable across requirements, analysis, design from the Requirements model.

DESIGN AXIOMS

Axioms are a fundamental truth that always is observed to be valid and for which there is no counter example or exception. Such explains that axioms may be hypothesized form a large number of observations by nothing the common phenomena shared by all cases; they cannot be proven or derived, but they can be invalidated by counter examples or exceptions. A theorem is a proposition that may not be self-evident but can be proven from accepted axioms. If therefore, is equivalent to a law or principle. A corollary is a proposition that follows from an axioms or another proposition that has been proven. Again, corollary is shown to be valid or not valid in the same manner as a theorem. In the two important axioms axiom 1 deals with relationships between system components and axiom 2 deals with the complexity of design.

The following the two important axioms:

Axiom 1: The independence axiom, which maintain the independence of the components.

Axiom 2: The information axioms that maintain the information content of the design.

Axioms1 states that, during the design process, as we go from requirement and use case to a system component, each component must satisfy that requirement without affecting other requirements.

An axiom 2 is concerned with simplicity. Scientific theoreticians often rely on a general rule known as Occam's razor, after William of Occam. He says, "The best theory explains the known facts with a minimum amount of complexity and maximum simplicity and straightforwardness."

The best designs usually involve the least complex code but not necessarily the fewest number of classes or methods. Minimizing complexity should be the goal, because that produces the most easily maintained and enhanced application. In an object-oriented system, the best way to minimize complexity is to use inheritance and the systems built in classes and to add as little as possible to what already is there.

From the two design axioms, many corollaries may be derived as a direct consequence of the axioms. These corollaries may be more useful in marking specific design decisions, since they can be applied to actual situations.

1. Uncoupled design with less information content: Highly cohesive objects can improve coupling because only a minimal amount of essential information need be passed between objects. The degree or strength of coupling between two components is measured by the amount and complexity of information transmitted between them.
2. Single purpose: Each class must have single, clearly defined purposes.
3. Large number of simple classes: Keeping the classes simple allows reusability. Large and complex classes are too specialized to be reused.
4. Strong mapping: There must be a strong association between the physical system and logical design. During the design phase, we need to design this class, design its methods, its association with other objects. So a strong mapping links classes should be identified.
5. Standardization: promote standardization by designing interchangeable and reusing existing classes or components.
6. Design with inheritance: Common behavior must be moved to super classes. The super class-sub class structure must make logical sense.

Refining attributes and methods:

Attributes identified in object oriented analyzed must be refined in the design phase. In the analysis phase, the name of the attributes was sufficient. But in the design phase, detailed information must be added to the model. The three basic types of attributes are:

1. Single valued attributes: This has only value or state.
2. Multiplicity or multivalue attributes: This has a collection of many values at any point in time.
3. Instance connection attributes: This is required to provide the mapping needed by an object to fulfill its responsibilities.

UML attribute presentation:

Visibility name: type-expression=initial-value

Visibility indicates either public visibility or protected visibility or private visibility. The public visibility indicates that the attribute can be accessible to all classes. The protected visibility indicates that the accessibility is given to the subclasses and operations of the class. The private visibility indicates that the accessibility can be given only to the operations of the class only.

Type expression is a language dependent specification of the implementation type of an attribute. Initial value is a language dependent expression for the initial value is optional.

5.1 Sequence and collaboration diagrams

Sequence Diagram

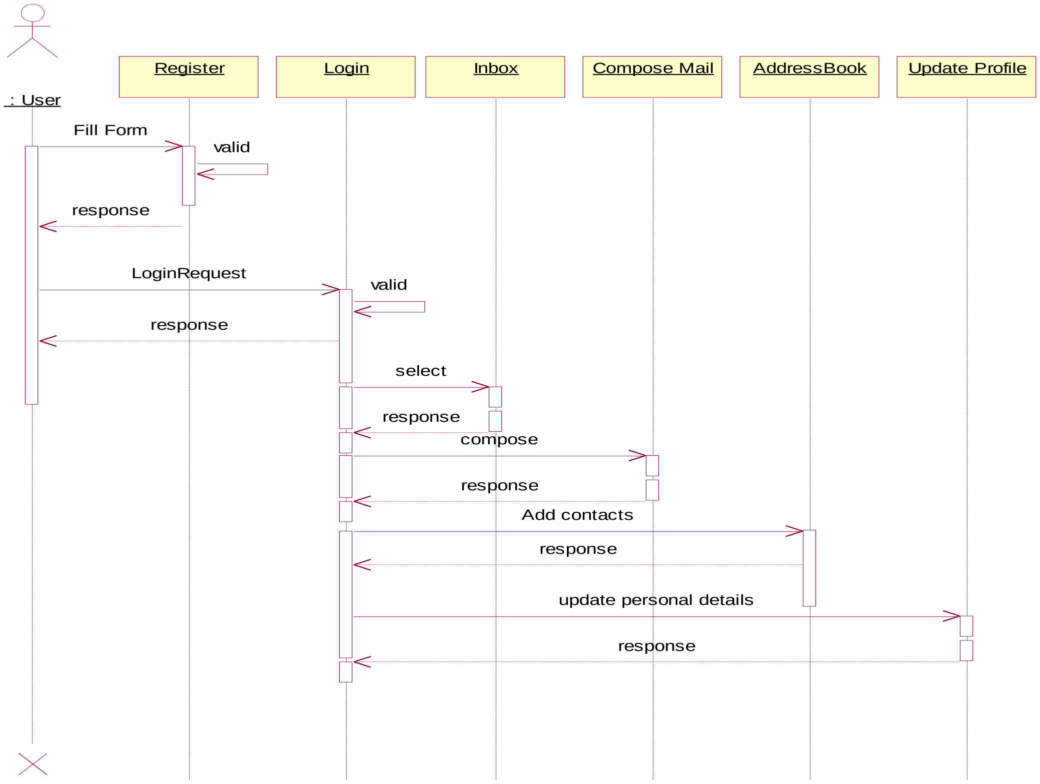
- An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
- A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
- Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis.

Contents

- Sequence diagrams commonly contain the following:
    - Objects
    - Links
    - Messages

Like all other diagrams, sequence diagrams may contain notes and constrains

Sequence:


: User
Register
Login
Inbox
Compose Mail
AddressBook
Update Profile
Fill Form
valid
response
LoginRequest
valid
response
select
response
compose
response
Add contacts
response
update personal details
response

Collaboration Diagram

- Collaboration is a society of classes, interfaces, and other elements that work together to provide some cooperative behavior that’s bigger than the sum of all its parts.

- Collaboration is also the specification of how an element, such as a classifier or an operation, is realized by a set of classifiers and associations playing specific roles used in a specific way

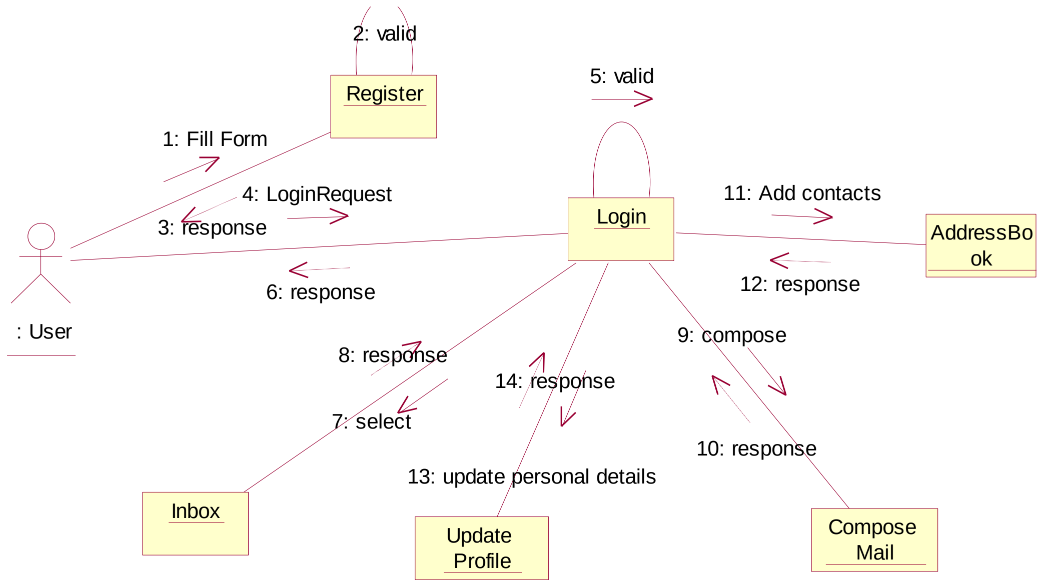
Contents

Collaboration diagrams commonly contain the following:

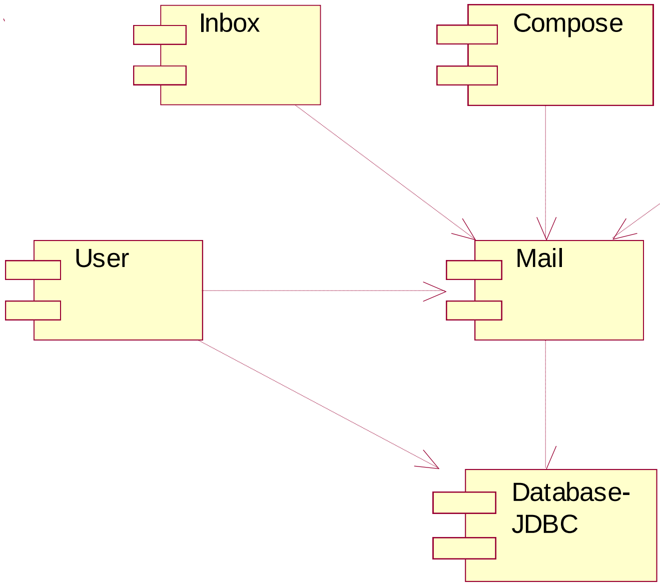
- Objects
- Links
- Messages

Like all other diagrams, sequence diagrams may contain notes and constrains.

Collaboration



Component Diagram:



Deployment Diagram

- A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them.

- Graphically, a deployment diagram is collection of vertices and arcs.

Contents

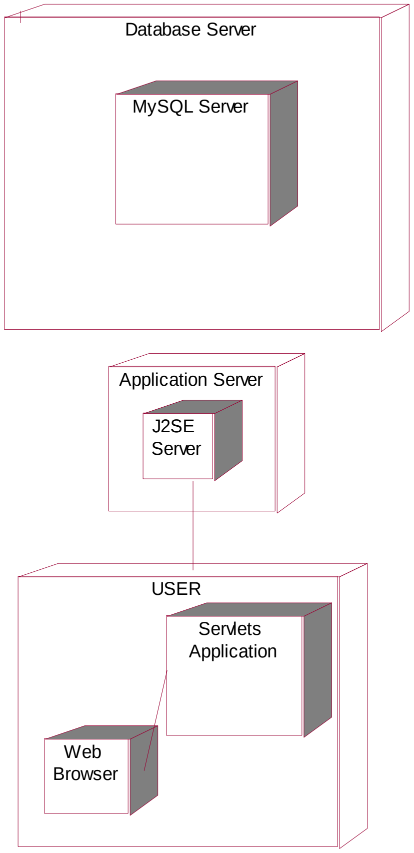
- Deployment diagram commonly contain the following things:

    - Nodes
    - Dependency and association relationships

- Like all other diagrams, deployment diagrams may contain notes and constraints.

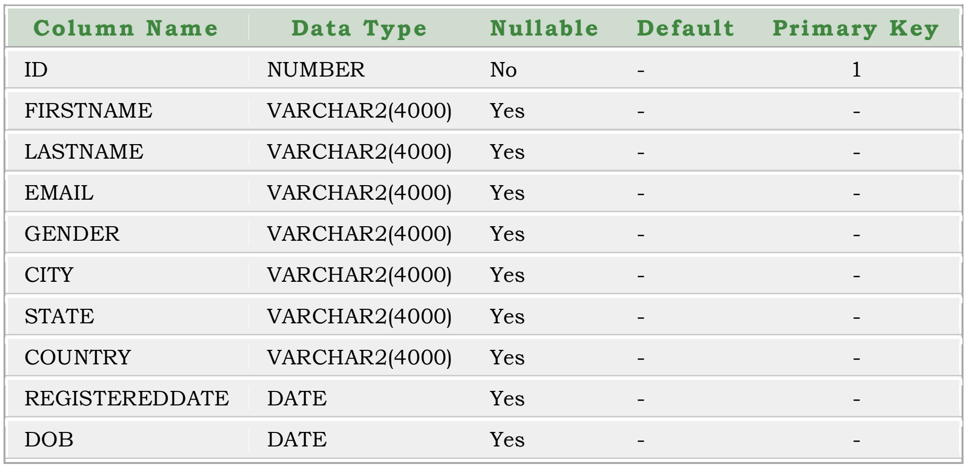
- Deployment diagrams may also contain components, each of which must live on some node.

- Deployment diagrams may also contain packages or subsystems, both of which are used to group elements of your model into larger chunks.


Database Server
MySQL Server
Application Server
J2SE
Server
USER
Servlets
Application
Web
Browser

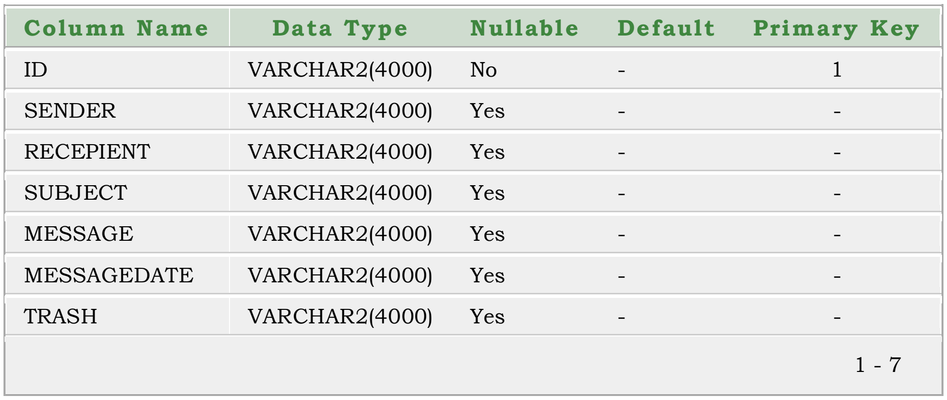
ORACLE DATA BASE TABLES

Table Name:- sssitmail_users



Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
FIRSTNAME	VARCHAR2(4000)	Yes	-	-
LASTNAME	VARCHAR2(4000)	Yes	-	-
EMAIL	VARCHAR2(4000)	Yes	-	-
GENDER	VARCHAR2(4000)	Yes	-	-
CITY	VARCHAR2(4000)	Yes	-	-
STATE	VARCHAR2(4000)	Yes	-	-
COUNTRY	VARCHAR2(4000)	Yes	-	-
REGISTEREDDATE	DATE	Yes	-	-
DOB	DATE	Yes	-	-

Table Name:- sssitmail_message



Column Name	Data Type	Nullable	Default	Primary Key
ID	VARCHAR2(4000)	No	-	1
SENDER	VARCHAR2(4000)	Yes	-	-
RECEPIENT	VARCHAR2(4000)	Yes	-	-
SUBJECT	VARCHAR2(4000)	Yes	-	-
MESSAGE	VARCHAR2(4000)	Yes	-	-
MESSAGEDATE	VARCHAR2(4000)	Yes	-	-
TRASH	VARCHAR2(4000)	Yes	-	-

1 - 7

TESTING AND IMPLEMENTATION

TESTING AND IMPLEMENTATION

- **Testing Methodologies**
  - Black box Testing:
  - White box Testing.
  - Gray Box Testing.

- **Levels of Testing**
  - Unit Testing.
  - Module Testing.
  - Integration Testing.
  - System Testing.
  - User Acceptance Testing.

- **Types Of Testing**
  - Smoke Testing.
  - Sanitary Testing.
  - Regression Testing.
  - Re-Testing.
  - Static Testing.
  - Dynamic Testing.
  - Alpha-Testing.
  - Beta-Testing.
  - Compatibility Testing.
  - Installation Testing.
  - Adhoc Testing.

- **TCD (Test Case Documentation)**
- **STLC**
  - Test Planning.
  - Test Development.
  - Test Execution.
  - Result Analysis.
  - Bug-Tracing.
  - Reporting.

- **Microsoft Windows – Standards**
- **Manual Testing**
- **Automation Testing (Tools)**
  - Win Runner.
  - Test Director

Testing:

- The process of executing a system with the intent of finding an error.
- Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
- Quality is defined as justification of the requirements
- Defect is nothing but deviation from the requirements
- Defect is nothing but bug.
- Testing --- The presence of bugs
- Testing can demonstrate the presence of bugs, but not their absence
- Debugging and Testing are not the same thing!
- Testing is a systematic attempt to break a program or the AUT

- Debugging is the art or method of uncovering why the script /program did not execute properly.

Testing Methodologies:

- **Black box Testing**: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.
  Usually Test Engineers are involved in the black box testing.
- **White box Testing**: is the testing process in which tester can perform testing on an application with having internal structural knowledge.
  Usually The Developers are involved in white box testing.
- **Gray Box Testing**: is the process in which the combination of black box and white box techniques are used.

STLC (SOFTWARE TESTING LIFE CYCLE)

Test Planning: **1.**Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way.

2. Objective of testing,

3. Areas that need to be tested,

4. Areas that should not be tested,

5. Scheduling Resource Planning,

7. Areas to be automated, various testing tools used

Test Development: **1.** Test case Development (check list)

2. Test Procedure preparation. (Description of the test cases)

Test Execution: **1.** Implementation of test cases. Observing the result.

Result Analysis: **1.** Expected value: is nothing but expected behavior Of application.

2. Actual value: is nothing but actual behavior of the application

Bug Tracing: Collect all the failed cases, prepare documents.

Reporting: Prepare document (status of the application)

Types of Testing:

- **Smoke Testing**: is the process of initial testing in which tester looks for the availability of all the functionality of the application in order to perform detailed testing on them. (Main check is for available forms)
- **Sanity Testing:** is a type of testing that is conducted on an application initially to check for the proper behavior of an application that is to check all the functionality are available before the detailed testing is conducted by on them.
- **Regression Testing:** is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before,

is once again tested whenever some new change is added in order to check whether the existing functionality remains same.

- **Re-Testing:** is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environments issues if at all any defects are there.
- **Static Testing:** is the testing, which is performed on an application when it is not been executed. ex: GUI, Document Testing
- **Dynamic Testing:** is the testing which is performed on an application when it is being executed. ex: Functional testing.
- **Alpha Testing:** it is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.
- **Beta-Testing:** it is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.
- **Compatibility testing:** it is the testing process in which usually the products are tested on the environments with different combinations of databases (application servers, browsers…etc) In order to check how far the product is compatible with all these environments platform combination.
- **Installation Testing:** it is the process of testing in which the tester try to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.
- **Adhoc Testing:** Adhoc Testing is the process of testing in which unlike the formal testing where in test case document is used, with out that test case document testing can be done of an application, to cover that testing of the future which are not covered in that test case document. Also it is intended to perform GUI testing which may involve the cosmotic issues.

TCD (Test Case Document):

Test Case Document Contains

- **Test Scope (or) Test objective**
- **Test Scenario**
- **Test Procedure**
- **Test case**

This is the sample test case document for the Acadamic details of student project:

Test scope:

- Test coverage is provided for the screen " Acadamic status entry" form of a student module of university management system application
- Areas of the application to be tested

Test Scenario:

- When the office personals use this screen for the marks entry, calculate the status details, saving the information on student's basis and quit the form.

Test Procedure:

- The procedure for testing this screen is planned in such a way that the data entry, status calculation functionality, saving and quitting operations are tested in terms of Gui testing, Positive testing, Negative testing using the corresponding Gui test cases, Positive test cases, Negative test cases respectively

Test Cases:

- Template for Test Case

T.C.No	Description	Exp	Act	Result

Guidelines for Test Cases:

1. **GUI Test Cases:**

- Total no of features that need to be check
- Look & Feel
- Look for Default values if at all any (date & Time, if at all any require)
- Look for spell check

Example for GUI Test cases:

T.C. No	Description	Expected value	Actual value	Result
1	Check for all the features in the screen	The screen must contain all the features		
2	Check for the alignmen of the objects as per the validations	The alignment should be in proper way		

2. **Positive Test Cases:**

- The positive flow of the functionality must be considered
- Valid inputs must be used for testing
- Must have the positive perception to verify whether the requirements are justified.

Example for Positive Test cases:

T.C. No	Description	Expected value	Actual value	Result
1	Check for the date Time Auto Display	The date and time of the system must be displayed		
2	Enter the valid Roll no into the student roll no field	It should accept		

3. Negative Test Cases:

- Must have negative perception.
- Invalid inputs must be used for test.

Example for Negative Test cases:

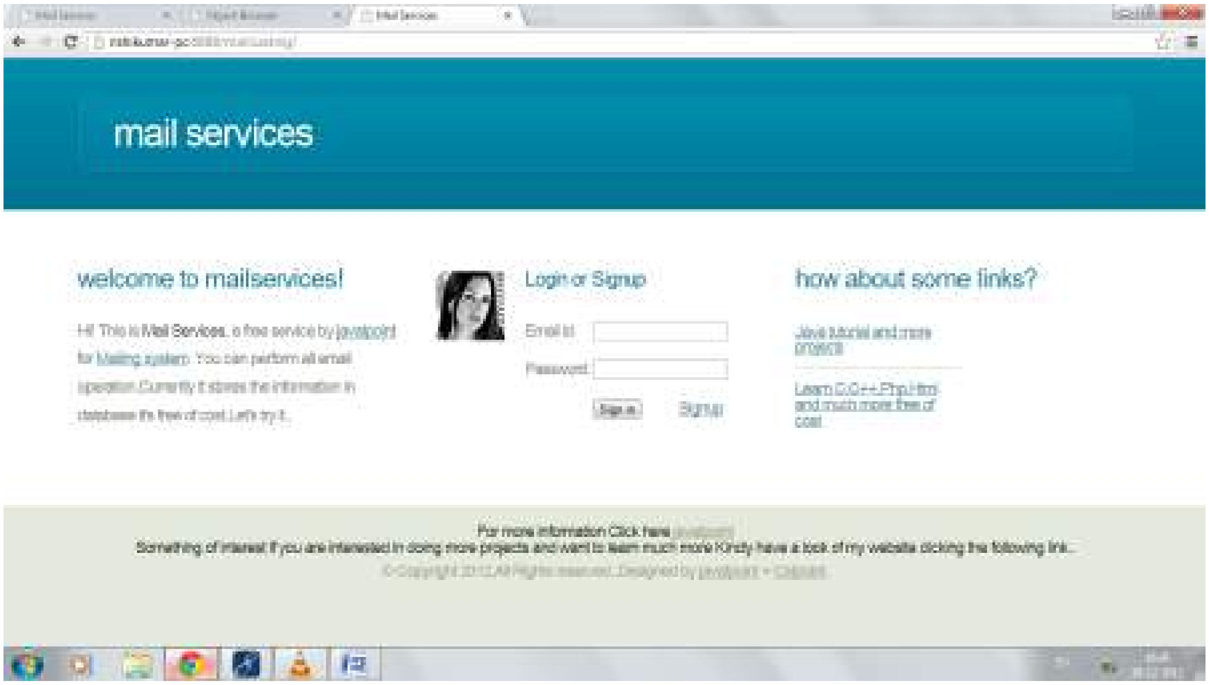
T.C. No	Description	Expected value	Actual value	Result
1	Try to modify the information in date and time	Modification should not be allow		
2	Enter invalid data in to the student details form, click on Save	It should not accept invalid data, save should not allow		

GUI

OUTPUT RESULTS

How Mail casting Project Works?

Welcome Page



welcome to mailservices!

Hi! This is Mail Services, a free service by javatpoint for Mailing system. You can perform all email operation. Currently it stores the information in database its free of cost.Let's try it.

Login or Signup

how about some links?

Java tutorial and more projects

Learn C,C++,Php,Html and much more free of cost

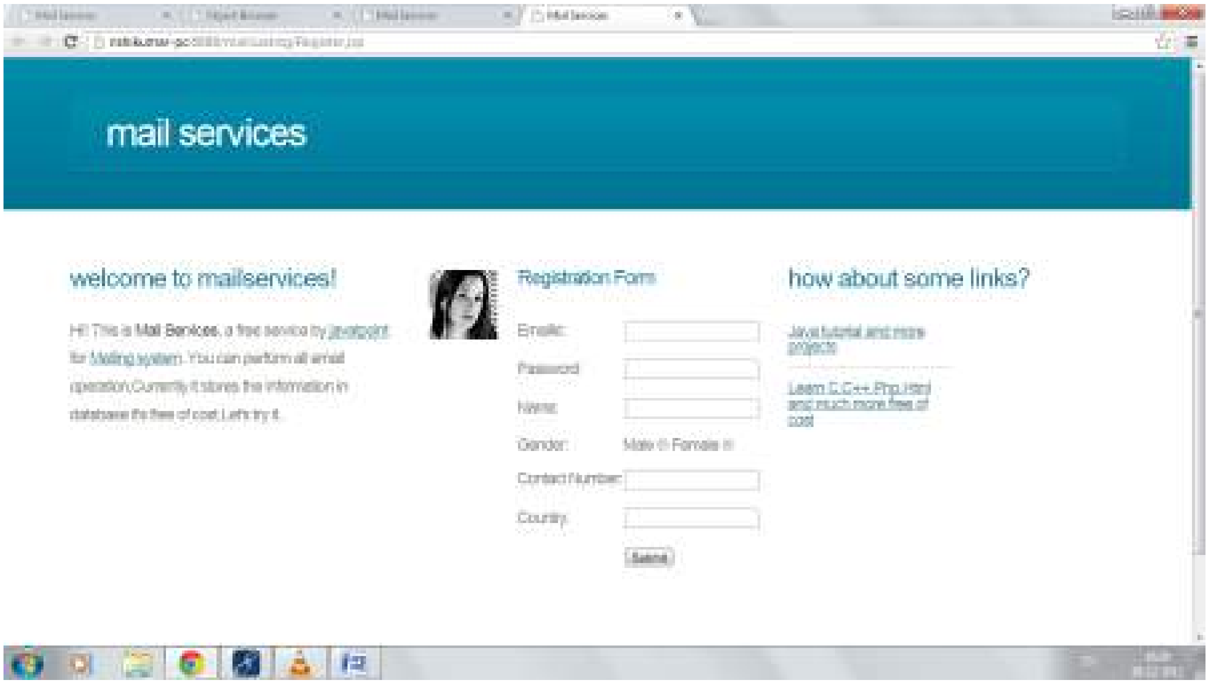
For more information Click here javatpoint

Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link...

© Copyright 2012.All Rights reserved. Designed by javatpoint + [illegible]

Steps:

1. **If you are a registered user kindly log in, if you aren't please signup first.**
2. **Signup:**


mail services

welcome to mailservices!

Hi! This is Mail Services, a free service by javatpoint for Mailing system. You can perform all email operation.Currently it stores the information in database it's free of cost.Let's try it.

Registration Form

Email:

Password:

Name:

Gender: Male ○ Female ○

Contact Number:

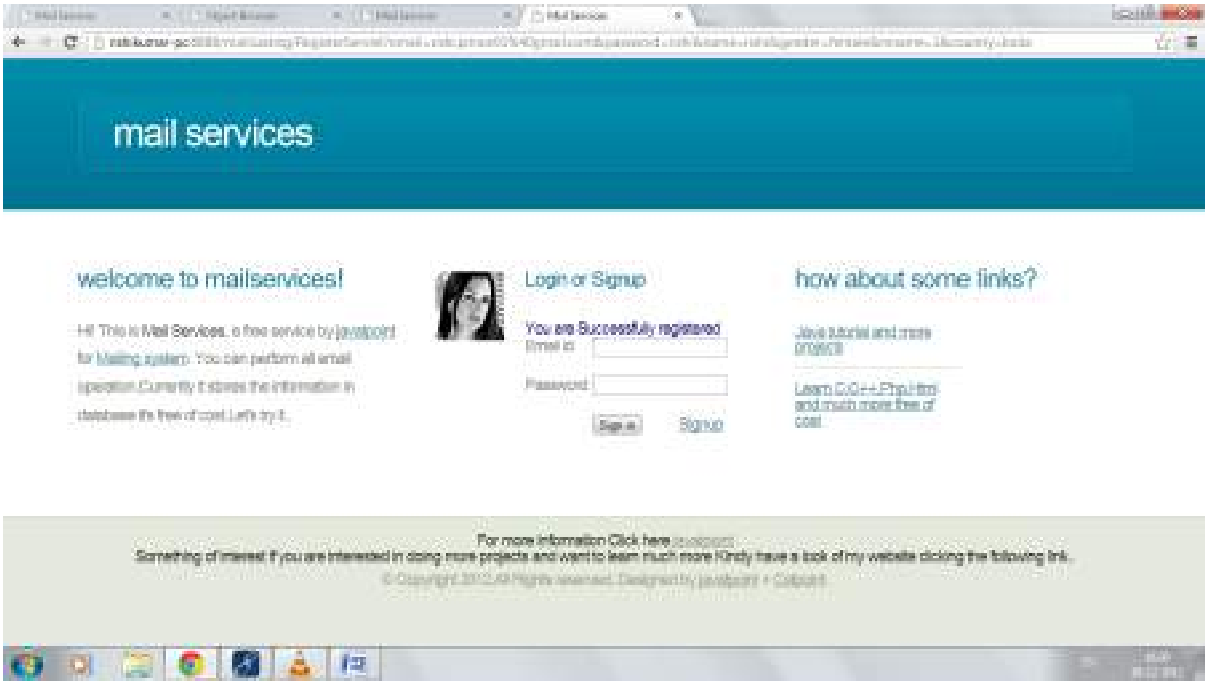
Country:

how about some links?

Java tutorial and more projects

Learn C,C++,Php,Html and much more free of cost

On successful registration:


mail services

welcome to mailservices!

Hi! This is Mail Services, a free service by javatpoint for Mailing system. You can perform all email operation.Currently it stores the information in database it's free of cost.Let's try it.

Login or Signup

You are Successfully registered

Email id:

Password:

how about some links?

Java tutorial and more projects

Learn C,C++,Php,Html and much more free of cost

For more information Click here javatpoint

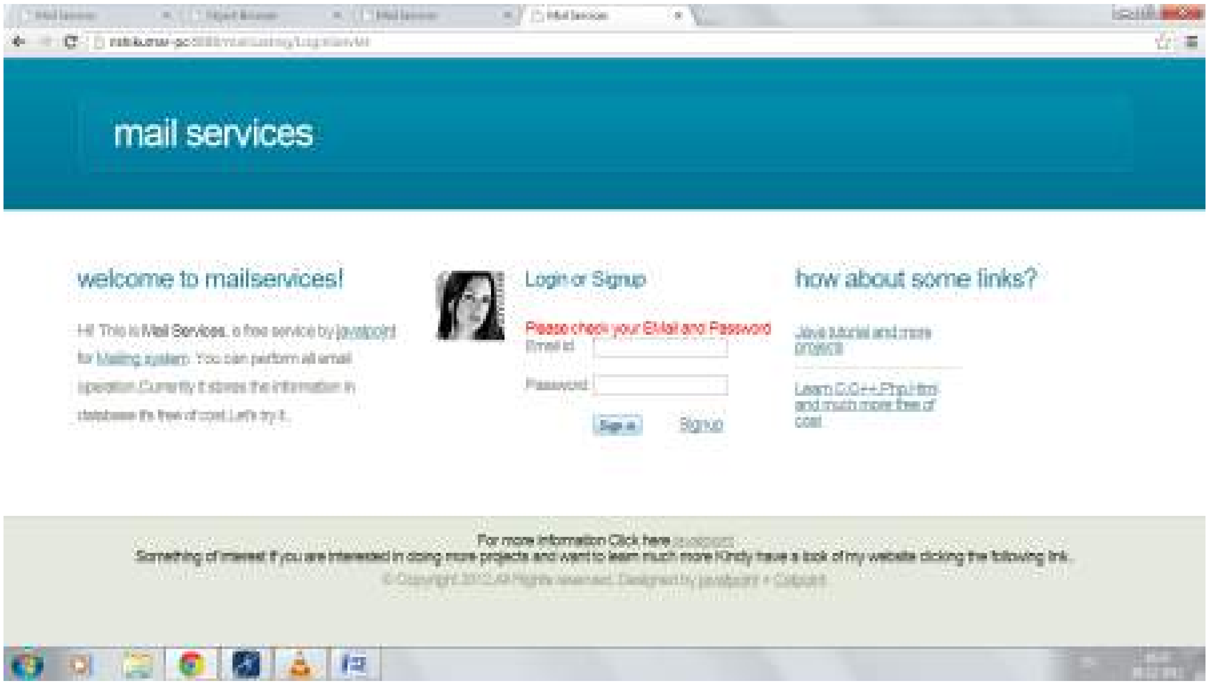
Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link.

© Copyright 2012 All Rights reserved. Designed by javatpoint + Colpoint

3. Login:

Login through above page or welcome page.

On filling Incorrect Detail:



mail services

welcome to mailservices!

Hi! This is **Mail Services**, a free service by javatpoint for Mailing system. You can perform all email operation.Currently it stores the information in database its free of cost.Let's try it..

Login or Signup

Please check your EMail and Password

Email id

Password

Sign in Signup

how about some links?

Java tutorial and more projects

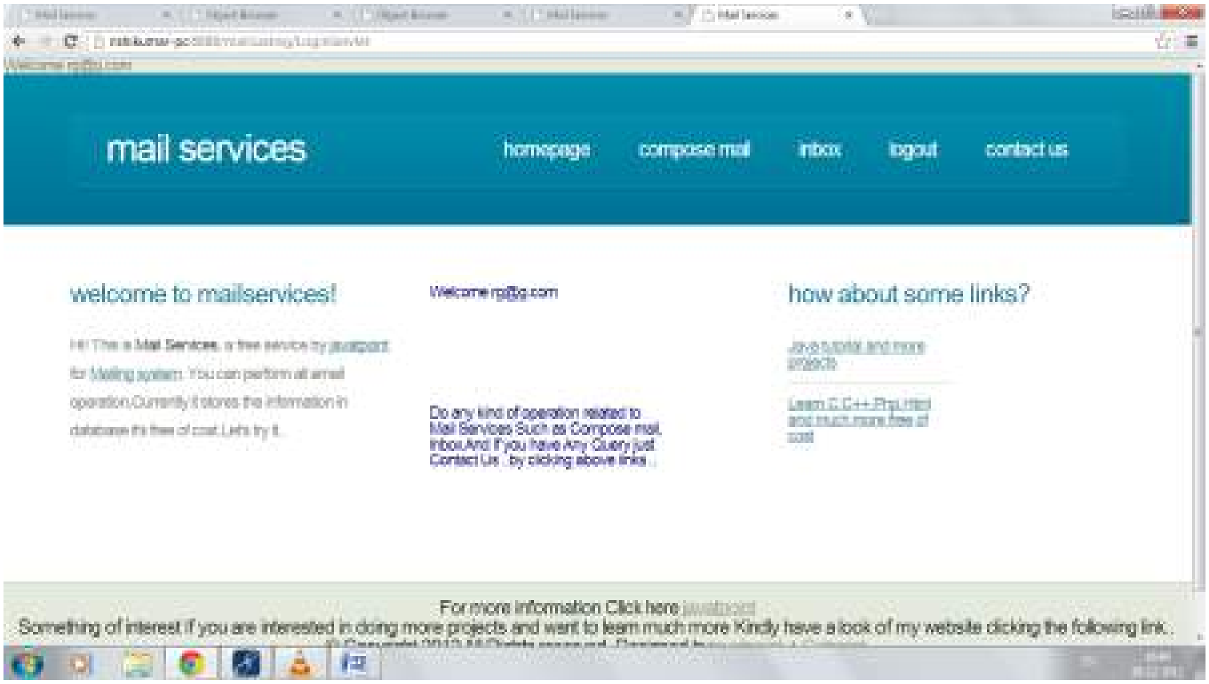
Learn C,C++,Php,Html and much more free of cost

For more information Click here javatpoint

Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link.

© Copyright 2012.All Rights reserved. Designed by javatpoint + Codepoint

On filling Correct Detail:



Welcome rg@g.com

mail services

homepage compose mail inbox logout contact us

welcome to mailservices!

Hi! This is **Mail Services**, a free service by javatpoint for Mailing system. You can perform all email operation.Currently it stores the information in database its free of cost.Let's try it..

Welcome rg@g.com

Do any kind of operation related to Mail Services Such as Compose mail, Inbox.And If you have Any Query just Contact Us ..by clicking above links ..

how about some links?

Java tutorial and more projects

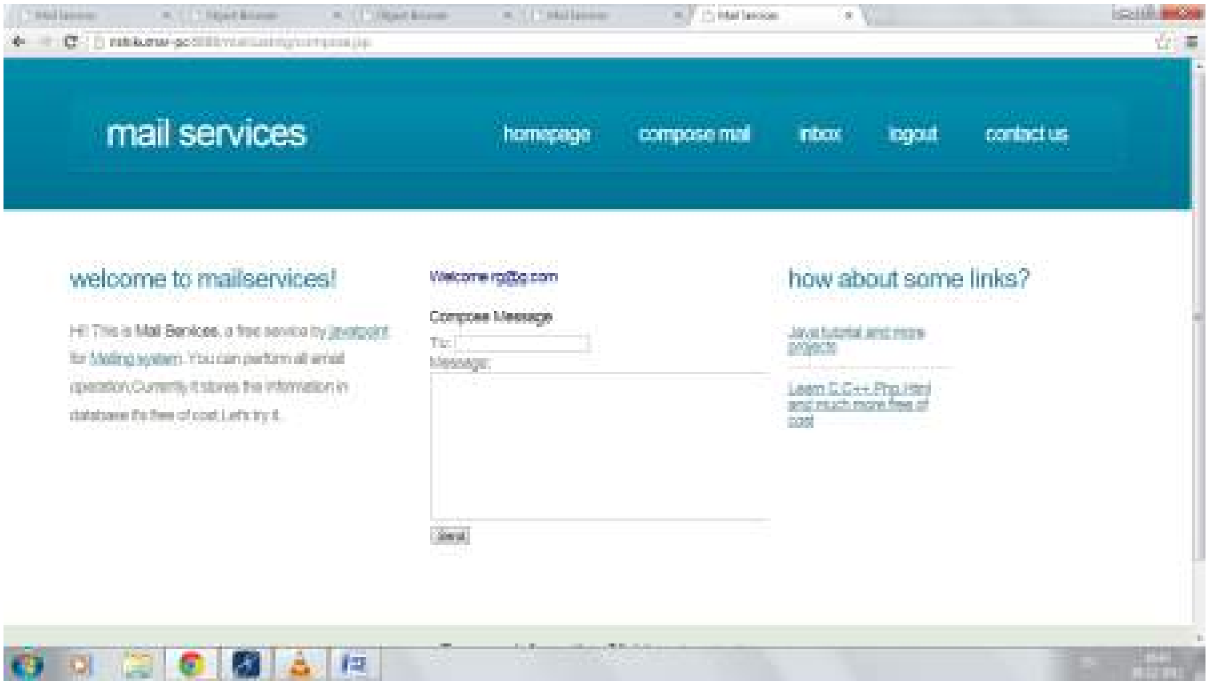
Learn C,C++,Php,Html and much more free of cost

For more information Click here javatpoint

Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link.

4. **Authorized user can compose mail, can check received mail and can contact to us.**

5. **Composing mail:**



welcome to mailservices!

Hi! This is Mail Services, a free service by javatpoint for Mailing system. You can perform all email operation.Currently it stores the information in database it's free of cost.Let's try it.

Welcome rg@g.com

Compose Message

To:

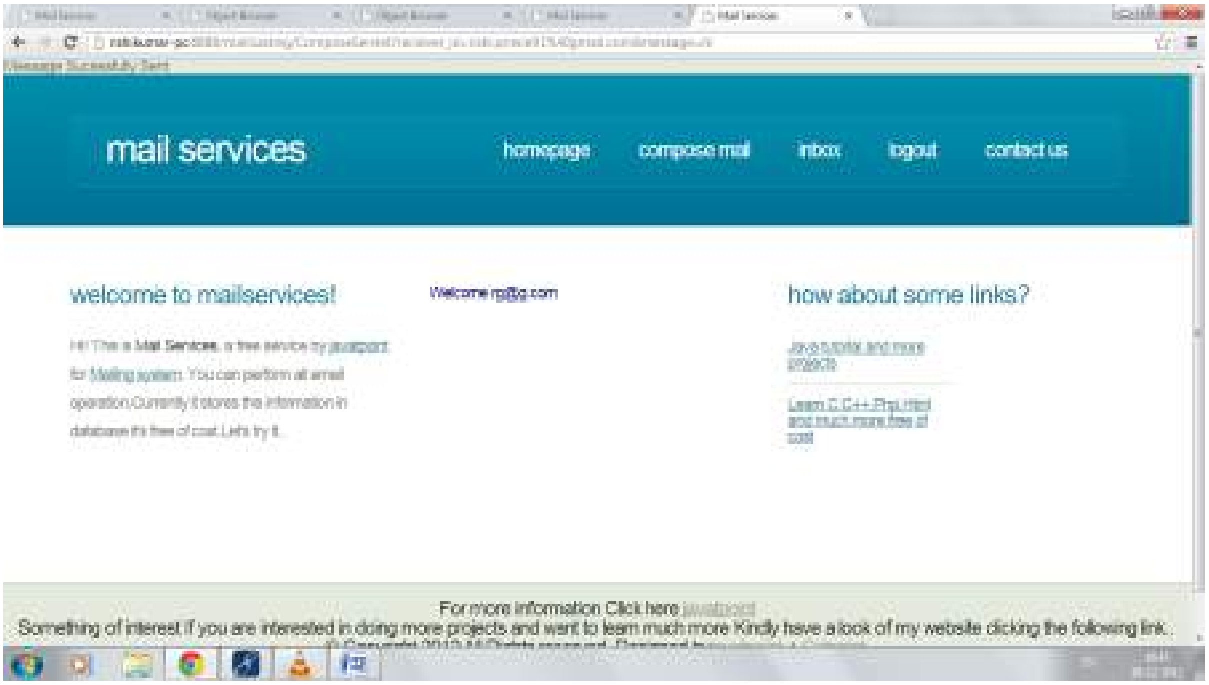
Message:

how about some links?

Java tutorial and more projects

Learn C,C++,Php,Html and much more free of cost

After sending mail:



Message Successfully Sent

welcome to mailservices!

Hi! This is Mail Services, a free service by javatpoint for Mailing system. You can perform all email operation.Currently it stores the information in database it's free of cost.Let's try it.

Welcome rg@g.com

how about some links?

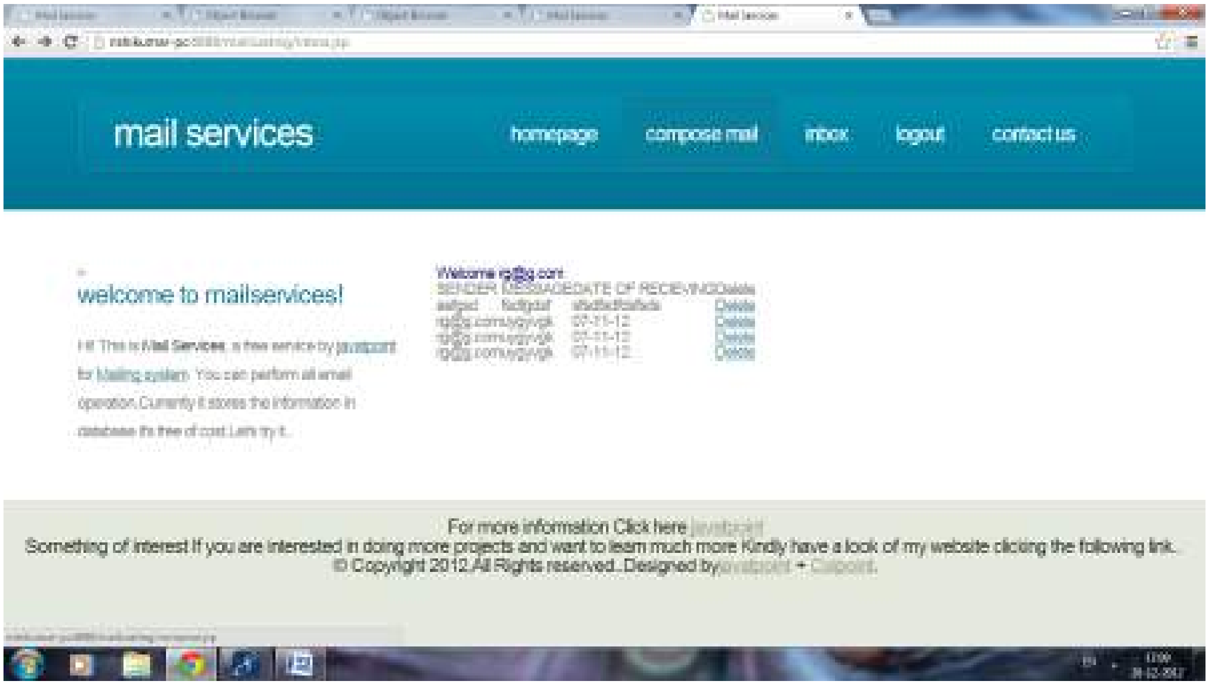
Java tutorial and more projects

Learn C,C++,Php,Html and much more free of cost

For more information Click here javatpoint

Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link.

6. Inbox:



welcome to mailservices!

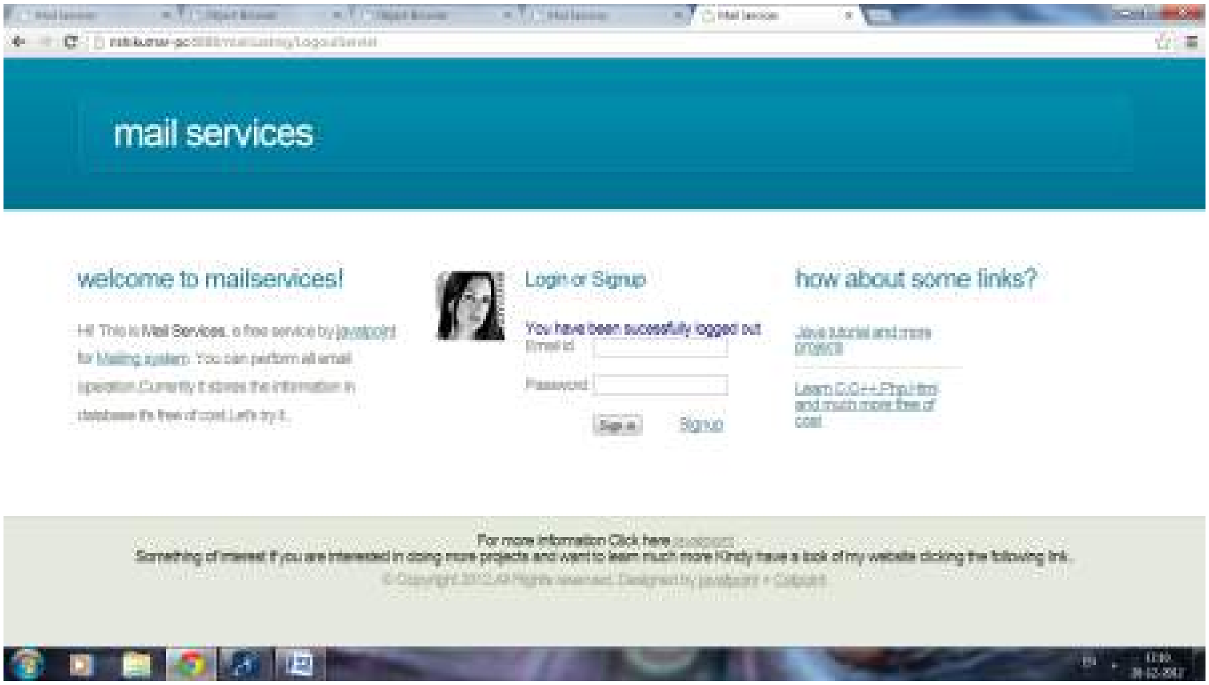
Hi! This is **Mail Services**, a free service by javatpoint for Mailing system. You can perform all email operation. Currently it stores the information in database its free of cost. Let's try it...

Welcome rg@g.com

SENDER	MESSAGE	DATE OF RECIEVING	Delete
asfgsd	fsdfgsdf	afsdfsdfsdfsds	Delete
rg@g.com	uygyvgk	07-11-12	Delete
rg@g.com	uygyvgk	07-11-12	Delete
rg@g.com	uygyvgk	07-11-12	Delete

For more information Click here javatpoint
Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link.
© Copyright 2012.All Rights reserved..Designed by javatpoint + Catpoint.

7. Logging out:



welcome to mailservices!

Hi! This is **Mail Services**, a free service by javatpoint for Mailing system. You can perform all email operation. Currently it stores the information in database its free of cost. Let's try it...

Login or Signup

You have been sucessfully logged out
Email id
Password

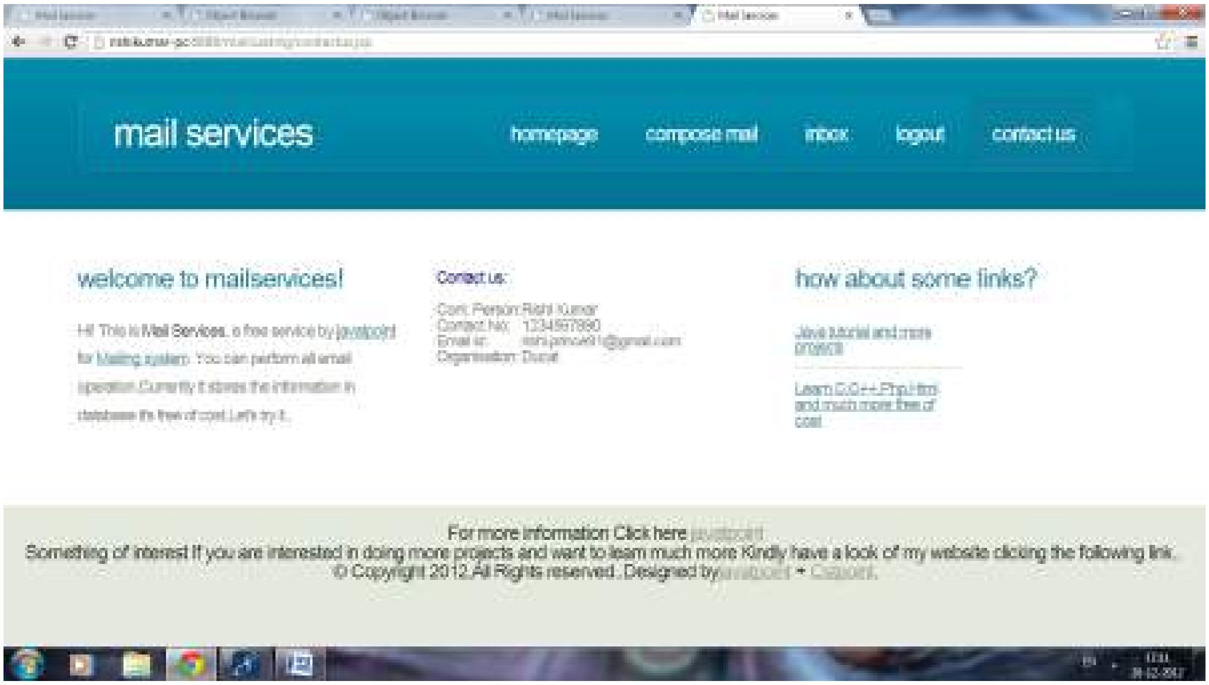
how about some links?

Java tutorial and more projects

Learn C,C++,Php,Html and much more free of cost

For more information Click here javatpoint
Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link.
© Copyright 2012.All Rights reserved..Designed by javatpoint + Catpoint

8. Contact us:



welcome to mailservices!

Hi! This is Mail Services, a free service by javatpoint for Mailing system. You can perform all email operation.Currently it stores the information in database its free of cost.Let's try it.

Contact us:

Cont. Person: Rishi Kumar
Contact No: 1234567890
Email at: rishi.prince91@gmail.com
Organisation: Ducat

how about some links?

Java tutorial and more projects

Learn C,C++,Php,Html and much more free of cost

For more information Click here javatpoint
Something of interest If you are interested in doing more projects and want to learn much more Kindly have a look of my website clicking the following link.
© Copyright 2012.All Rights reserved . Designed by javatpoint • Cstpoint.

Coding

File Name: ComposeMailServlet.java

```
      package com.javatpoint;


      import java.io.IOException;

      import java.io.PrintWriter;


      import javax.servlet.ServletException;

      import javax.servlet.http.HttpServlet;

      import javax.servlet.http.HttpServletRequest;

      import javax.servlet.http.HttpServletResponse;

      import javax.servlet.http.HttpSession;

      public class ComposeMailServlet extends HttpServlet {

            protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

                  response.setContentType("text/html");

                  PrintWriter out=response.getWriter();



      request.getRequestDispatcher("header.html").include(request, response);


                  HttpSession session=request.getSession(false);

                  if(session!=null){

                  String sender=(String)session.getAttribute("email");

                  String recipient=request.getParameter("to");

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

                  String message=request.getParameter("message");
```

```
                int i=MailDao.save(sender, recipient, subject, message);

                if(i>0){

                      out.print("message successfully sent!");

                }


                }else{

                      response.sendRedirect("loginerror.html");

                }


                out.close();

          }


    }
```

File Name: ConnectionProvider.java

```
    package com.javatpoint;


    import java.sql.*;


    public class ConnectionProvider {

    public static Connection getConnection(){

          Connection con=null;

          try{

                      Class.forName("oracle.jdbc.driver.OracleDriver");

    con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","
system","oracle");


          }catch(Exception e){e.printStackTrace();}

          return con;
```

```
      }

      }
```

File Name: DateFormatter.java

```
      package com.javatpoint;


      import java.text.SimpleDateFormat;


      public class DateFormatter {


            public static java.sql.Date formatdate(String stringdate)throws
Exception{

                  SimpleDateFormat formatter=new SimpleDateFormat("yyyy-MM-
dd");

                  java.util.Date utildate=formatter.parse(stringdate);

                  java.sql.Date sqldate=new
java.sql.Date(utildate.getTime());

                  return sqldate;

            }

      }
```

File Name: InboxServlet.java

```
            package com.javatpoint;

            import java.io.IOException;

      import java.io.PrintWriter;


      import javax.servlet.ServletException;

      import javax.servlet.http.HttpServlet;

      import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;


public class InboxServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();


request.getRequestDispatcher("header.html").include(request, response);



        out.close();
    }


    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request,response)
    }
}
```

File Name:LoginDao.java

```
package com.javatpoint;
import java.sql.*;


public class LoginDao {


    public static String validate(String email,String password){
        String name=null;
```

```
                  try{

                        Connection con=ConnectionProvider.getConnection();

                        PreparedStatement      ps=con.prepareStatement("select
firstname from sssitmail_users where email=? and password=?");

                        ps.setString(1,email);

                        ps.setString(2,password);

                        ResultSet rs=ps.executeQuery();

                        if(rs.next()){

                              name=rs.getString(1);

                        }

                  }catch(Exception e){e.printStackTrace();}

                  return name;

            }

      }
```

File Name: LoginServlet.java

```
      package com.javatpoint;


      import java.io.IOException;

      import java.io.PrintWriter;


      import javax.servlet.ServletException;

      import javax.servlet.http.HttpServlet;

      import javax.servlet.http.HttpServletRequest;

      import javax.servlet.http.HttpServletResponse;

      import javax.servlet.http.HttpSession;


      public class LoginServlet extends HttpServlet {
            protected       void       doPost(HttpServletRequest       request,
HttpServletResponse response) throws ServletException, IOException {

                  response.setContentType("text/html");

                  PrintWriter out=response.getWriter();



                  String
email=request.getParameter("email").concat("@sssitmail.com");

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



                  String name=LoginDao.validate(email, password);

                  if(name!=null && !name.equals("")){

                        HttpSession session=request.getSession();

                        session.setAttribute("name",name);

                        session.setAttribute("email",email);

                        response.sendRedirect("InboxServlet");

                  }else{


      request.getRequestDispatcher("loginerror.html").forward(request,
response);

                  }



                  out.close();

            }



      }
```

File Name: MailDao.java

```
      package com.javatpoint;

      import java.sql.*;
      public class MailDao {


            public static int save(String sender,String recipient,String
subject,String message){

                  int status=0;

                  try{

                  Connection con=ConnectionProvider.getConnection();


                  PreparedStatement ps=con.prepareStatement("insert into
sssitmail_message(sender,recipient,subject,message,messagedate,trash)
values(?,?,?,?,?,?)");

                  ps.setString(1,sender);

                  ps.setString(2,recipient);

                  ps.setString(3,subject);

                  ps.setString(4,message);


                  java.util.Date
utildate=java.util.Calendar.getInstance().getTime();

                  java.sql.Date sqldate=new
java.sql.Date(utildate.getTime());


                  ps.setDate(5,sqldate);

                  ps.setString(6,"no");

                  status=ps.executeUpdate();


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


                  return status;

            }

      }
```

File Name: RegisterDao.java

```
             package com.javatpoint;

             import java.sql.*;

             public class RegisterDao {


            public static int save(String firstname,String lastname,String
email,String password,String gender,Date sqldob,String city,String
state,String country,Date sqlcurrentdate){

                  int status=0;


                  try{

                        Connection con=ConnectionProvider.getConnection();

                        PreparedStatement ps=con.prepareStatement("insert
into
sssitmail_users(firstname,lastname,email,gender,dob,city,state,country,regist
ereddate,password) values(?,?,?,?,?,?,?,?,?,?)");

                        ps.setString(1,firstname);

                        ps.setString(2,lastname);

                        ps.setString(3,email);

                        ps.setString(4,gender);

                        ps.setDate(5,sqldob);

                        ps.setString(6,city);

                        ps.setString(7,state);

                        ps.setString(8,country);

                        ps.setDate(9,sqlcurrentdate);

                        ps.setString(10,password);


                        status=ps.executeUpdate();

                  }catch(Exception e){System.out.println(e);}
```

```
                              return status;

            }

      }
```

File Name: RegisterServlet.java

```
           package com.javatpoint;

            import java.io.IOException;

      import java.io.PrintWriter;

      import java.util.Calendar;


      import javax.servlet.ServletException;

      import javax.servlet.http.HttpServlet;

      import javax.servlet.http.HttpServletRequest;

      import javax.servlet.http.HttpServletResponse;


      public class RegisterServlet extends HttpServlet {

            protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

                  response.setContentType("text/html");

                  PrintWriter out=response.getWriter();

                  out.print("<span style='color:red;font-size: 30px;font-
family: sans-serif;'>CompanyMailer</span><hr/>");


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

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

                  String
email=request.getParameter("email").concat("@sssitmail.com");

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

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

                  String dob=request.getParameter("dob");
                  String city=request.getParameter("city");

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

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

                  java.sql.Date sqldob=null;

                  try{

                        sqldob=DateFormatter.formatdate(dob);

                  }catch(Exception e){out.print(e);}


                  java.util.Date
currentdate=Calendar.getInstance().getTime();

                  java.sql.Date sqlcurrentdate=new
java.sql.Date(currentdate.getTime());


                  int i=RegisterDao.save(firstname, lastname, email,
password, gender, sqldob, city, state, country, sqlcurrentdate);

                  if(i>0){

                        out.print("you are successfully registered!<br/>");


      request.getRequestDispatcher("loginform.html").include(request,
response);

                  }

                  else{

                        out.print("registration failed!");

                  }


                  out.close();

            }


      }
```

File Name:- LogOutServlet.java

```java
package com.javatpoint;


import java.io.IOException;

import java.io.PrintWriter;


import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class LogoutServlet extends HttpServlet {

      protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

            response.setContentType("text/html");

            PrintWriter out=response.getWriter();


            out.print("<span style='color:red;font-size: 30px;font-family: sans-serif;'>CompanyMailer</span><hr/>");


            request.getSession(false).invalidate();

            //out.print("<p>You are successfully logged out!</p>");


            //request.getRequestDispatcher("loginform.html").include(request, response);


            response.sendRedirect("index.html");

      }


}
```

File Name:- LogOutServlet.java

```
package com.javatpoint;


import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;


import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class SentMailServlet extends HttpServlet {
      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            response.setContentType("text/html");
            PrintWriter out=response.getWriter();


            request.getRequestDispatcher("header.html").include(request,
response);


            HttpSession session=request.getSession(false);
            String name=(String)session.getAttribute("name");
            String email=(String)session.getAttribute("email");


            out.print("<span style='float:right'>Hi,"+name+"</span>");
            out.print("<h2>Inbox</h2>");
            try{

                  Connection con=ConnectionProvider.getConnection();

                  PreparedStatement ps=con.prepareStatement("select * from
sssitmail_message where sender=? and trash=? order by id desc");

                  ps.setString(1,email);

                  ps.setString(2,"no");

                  ResultSet rs=ps.executeQuery();

                  out.print("<table border='1' cellpadding='4'>");


      out.print("<tr><th>No.</th><th>Recipient:Subject</th><th>Message
Date</th></tr>");

                  int count=1;

                  while(rs.next()){

                        out.print("<tr><td>"+(count++)+"</td><td><a
href='InboxServletMessage?id="+rs.getInt(1)+"'>"+rs.getString("recipient")+":
"+rs.getString("subject")+"</a></td><td>"+rs.getDate("messagedate")+"</td></t
r>");

                  }

                  con.close();

                  out.print("<table>");;


            }catch(Exception e){out.print(e);}


            out.close();


      }


}
```

Conclusion

CONCLUSION

The project titled as **“Email System" has** been designed with much care, with the intention easier and the more complexity involved is presented in a simple and lucid style.

The advantages of the mailing System are

1. Security

2.Cost effective (may be Free of cost)

3.Less Mailing Time

4.Gift Incentives and many more...

The Intranet Mailing System works in a similar fashion as that of an Intranet Mailing System, there is no need to get an internet connection for this mailing system. The various branches of the organization can be connected to a single host server and then an employee of one branch can send a message to an employee of another branch through server.

The users of Intranet Mailing are given a unique user-id and password is hided .So it gives security also.

BIBLIOGRAPHY

BIBLIOGRAPHY

- For Java installation
- https://www.java.com/en/download/
- For Oracle DataBase installation
- http://www.oracle.com/index.html
- Reference websites
- www.javatpoint.com
- www.w3schools.com
- http://www.tutorialspoint.com/java/index.htm
- Reference Books
- Thinking in java
- OCJP Certified Programmer for Java
- Learn Java in Eassy Steps
- Complete reference Java