

Customer Support Ticketing Service

Kandregula Someswara Prudhvi

¹(Computer Science and Engineering ,GITAM University ,Bengaluru ,KA,India -561023)

Date of Submission: 01-09-2022

Date of Acceptance: 10-09-2022

ABSTRACT : Due to increase in demand of online services number of new comers are increasing daily to the world of internet. they many face issues with the product or service. or they may need clarification regarding something. answering each and every one in old fashion way of calling or texting may not be feasible every kind of companies has different sections that help the method of paperwork concerning personnel and maintenance. However, some of this paperwork might be overlooked or delayed, which could lead to slow down the outcome of the organization. So, they have to enhance their system work to save lots of time, effort and make an efficient mechanism to follow requests and tickets of any organization or customer. As a solution to that, we have come with the idea of the " customer ticketing service." The application aim is to support, facilitate, and solve problems regarding time, effort, priority and type. It is a modern way to solve problems.

KEYWORDS: Ticket Tracking,

INTRODUCTION

Project background

Customer ticketing system is the system which help both customers and the agents in productivity. It not merely detects the issues but provides the complete information regarding issues detected. Customer ticketing System ensures the user of it who needs to know about a provide information regarding the identified issue.

When a customer raises tickets, it will be assigned to one of the developers/Support Staff. The admin manager is responsible for adding users to respective field of work.

Problem Statement

Customer Support Ticketing System can dramatically increase the productivity and accountability of individual employees by providing a documented workflow and positive feedback for good performance.

1.3 Objectives

The main objective for developing this project is an Customer Support Tracking system for issues. This project help to provide Information of trouble tickets regarding general issues such as email, viruses, network issues. But as the company has grown, managing these spreadsheets has become tedious. This application has a good appearance and is very easy to operate. It saves our time and money. This project provides a lot of features to manage in very well manner. This project contains a lot of advance modules which makes the backend system very powerful.

Literature Review:

The development of a system based on the smart case for help desk operations is increased around the world with the use of modern technological systems. However, it requires the availability of the modern electronic system to arrange, follow-up, and resolve problems separately by describing the problem and the statement of importance by numbers, and track the problem to be solved in the best way without the need to refer to it again and again .

In 2005, González et al. designed a helpdesk system called Knowledge management-centric help desk to minimize the required time to solve problems and improve the helpdesk system functionality.

Automation of helpdesk using case-based reasoning is a system concerned with solving customers' problems. Through Case-Based Reasoning (CBR). It classifies the problem and makes it easier for the competent authority to without any delay; the manager can keep track of all the problems and return to it at any time . Furthermore, an automated Help Desk was developed to provide a variety of online resources for the clients for the aim of resolving problems related to IT.

Smartphones are massively used among people, regardless of their age, position, and interest. An experiential study was carried to find the impact of smartphone applications on forest

products industry and academia in the USA. The study concluded that people use a smartphone as a help tool in their job. Another study was conducted by Martínez et al. to test the influence of smartphone applications in medical education. The use of smartphone applications enhanced the students' academic performance. This indicates the importance and influence of smartphone applications in improving lifestyle. However, it is essential to understand how people are accepting the use of knowledge management system and how the factors of individual and organizational influence the people use of such system.

WORKING AND IMPLEMENTATION

Determination of broad requirements of Project :

Authenticate User:

It first activates the login form. Here it will Username and Password are matching with the one present in database or not. If not we will get a warning message. After Successful login the system will display the menu. We will be preparing activity log for if there is any failure.

Products:

List of Products:

After successful login the user can see the list of tickets and the details of it. Here the user can modify(add/delete) it.

Product Versions:

It is not possible to resolve the ticket at single attempt. So, here we will divide it into different submodules and according to that we will keep deadlines for that.

Product Users:

In order to complete the project we need users data. That data we will be storing in the database for future reference. Later each user is allocated with some tasks based on his performance he we will be getting the review, rating based on his we will try to assign him to the next ticket. We will store all the user data in the database.

Ticket Details:

Here in this module the user has the facility to add new ticket or he can modify the old ticket. As number of tickets we receive may be large the system is capable of handling all those tickets. It was designed in such a way it can handle large number of tickets simultaneously. The user has the facility that he can filter the **issues based** on priority. Once the user applies the priority it will

reflect in the database and the tickets will be fetched from database based on priority(severity of the defect).

Ticket History:

Ticket Assignee:

This displays the list of users for whom the issue is assigned for resolution. As the issue need to be resolved for completing the product several user are assigned to find a solution for the issue. The user can add this issue to a new user or he can modify the existing user details.

Ticket Attachments:

This gives a list of attachments for a particular issue. The issue can be of any type it can be a database issue or a GUI issue. So while you add a issue you need to provide with the details of issue. So the file attachments can be a document, database file or an image file. All then attachments are stored in a location along with the size and type of the file. Here the user can add a new attachment or can change the details of existing files.

Ticket Tracking:

Track Hierarchy:

All the issues saved in the database will have a particular hierarchy. There might be issues which can be related to the earlier issues saved in the database so our system is provided with a hierarchy. And user can add child nodes in this hierarchy or he can modify the existing values of the nodes. This hierarchy is based on the parent child relation ship between the issues. Here we will maintain all the details of the tickets. We will be storing previous tickets solutions. As the ticket needs several methods, techniques to solve the issue. It is important to store the history of the **ticket**

Track Resolution:

This displays a list of all solutions provided by the users allotted to a issue. This stores the action type and the necessary resolution provided by the user.

Track Resources:

This displays list of resources allotted to the project. As the issues need to be resolved resources are provided for the issues. These Resources will be the resources allotted to the project. The resources are allotted based on the rating of the employee.

Search: Our system provides with the feature of advanced search technique. Generally Number of

issues for a project increased tremendously so if we want to know about a particular issue It takes much amount of time. With the search screen provided one can filter the issue's base on priority, product, severity, database and type of operating system. He can also list the issues between particular time based on the start date and end date. After Searching it displays a list of issues. From this list the user can modify the existing issues or can add a new issue.

Admin

Users: All the users of this system are displayed in this module. One can add new user or can update the details of an existing user. Here the password provided by the user is encrypted before saving them to the database for proper security. This module saves the details like address, phone and email.

Configuration: All the Values that we are using in this system are configurable. Values like status, priority and others can be added dynamically on the screen. Suppose if we limit these fields by hot coding them and if the user wants to add a new value again he has to come to the developer of the product. So In order to avoid this it is provided with the feature of adding values from the screen. And the user can change the status to In Active whenever he wants.

Log View

In order for the efficient Tracking of the system logs are maintained. As the logs will be in vast it will be a problem for user for checking the database. The Log View module can be searched based on the user and Records between a start date and end date. '

Log Out

In this once the user clicks on Log out First the session variable is killed and then the system is redirected to the login page.

Prepare Logs

At all the stages, whenever user performs an operation by clicking a button, automatically the Tracking System logs the activity

Introduction to System Analysis:

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers.

In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Existing System:

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application.

Lack of security of data.
More man power.
Time consuming.
Consumes large volume of pare work.
Needs manual calculations.
No direct role for the higher officials

Proposed System:

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

Security of data.
Ensure data accuracy.
Proper control of the higher officials.
Minimize manual data entry.
Minimum time needed for the various processing.
Greater efficiency.
Better service.
User friendliness and interactive.
Minimum time required.

User Characteristics:

We have 2 levels of users
a) User module: This is a normal level of user who will be very few numbers of functionality for website
b) Administration module: This user is an admin type who has full rights on the system.

Assumptions and Dependencies:

All the data entered will be correct and up to date. This software package is developed using JSP as front end which is supported by Apache Server system. MySQL as the back end.

a) User interface:
HTML has been used for developing the User Layout for the system.
JavaScript has been used for creating all the validations and client-side scripting functionality.
CSS has been used for designing the web pages of the system.

Communication Protocol :

Following protocols are required to be permitted on the server side

a)HTTP incoming request

Functional Requirements:

The system runs of apache server so it is needed that server must have apache server version 2.0 available.

We have used JSP for server-side scripting so the current version of JSP must be available on the server.

MySQL database has been used for storing the data of the website.

HTML has been used for creating the layout of the web application.

CSS has been used for creating the designing of the webpages.

JavaScript scripting language has been implemented on the system for performing all of the Client Side Server Validation.

Classes and Objects of the Project:

Login Class: Used for performing all the operations of the login functionality.

Page Class: Class for managing all the operations of the page.

Traffic Class: Class for managing the traffic of the website.

IP Class: It has been used for storing all the IPS which hits the website.

Users Class: Class for managing all the user operations.

Permission Class: This class has been used for managing all the permissions level of operations.

Non Functional Requirements:

Performance: System should be able handle multiple users at a time using any of the web browsers.

Reliability: Database updating should follow transaction processing to avoid data inconsistency.

Availability: The project will be deployed on a public shared server so it will be available all the time and will be accessible anywhere of the world using internet.

Security: We have implemented a lot of security mechanism to avoid to hack the system by outer world.

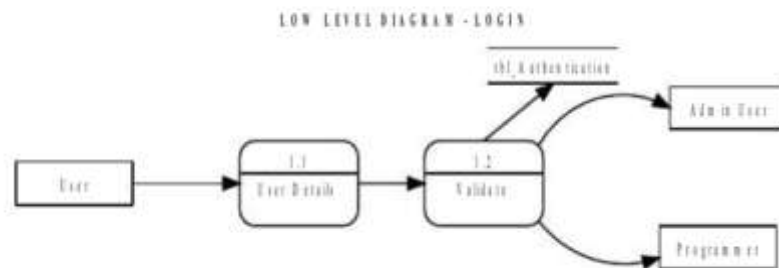
Maintainability: It is very easy to maintain the system. The system has been developed on JSP so anyone who has the knowledge of JSP, can easily maintain the system

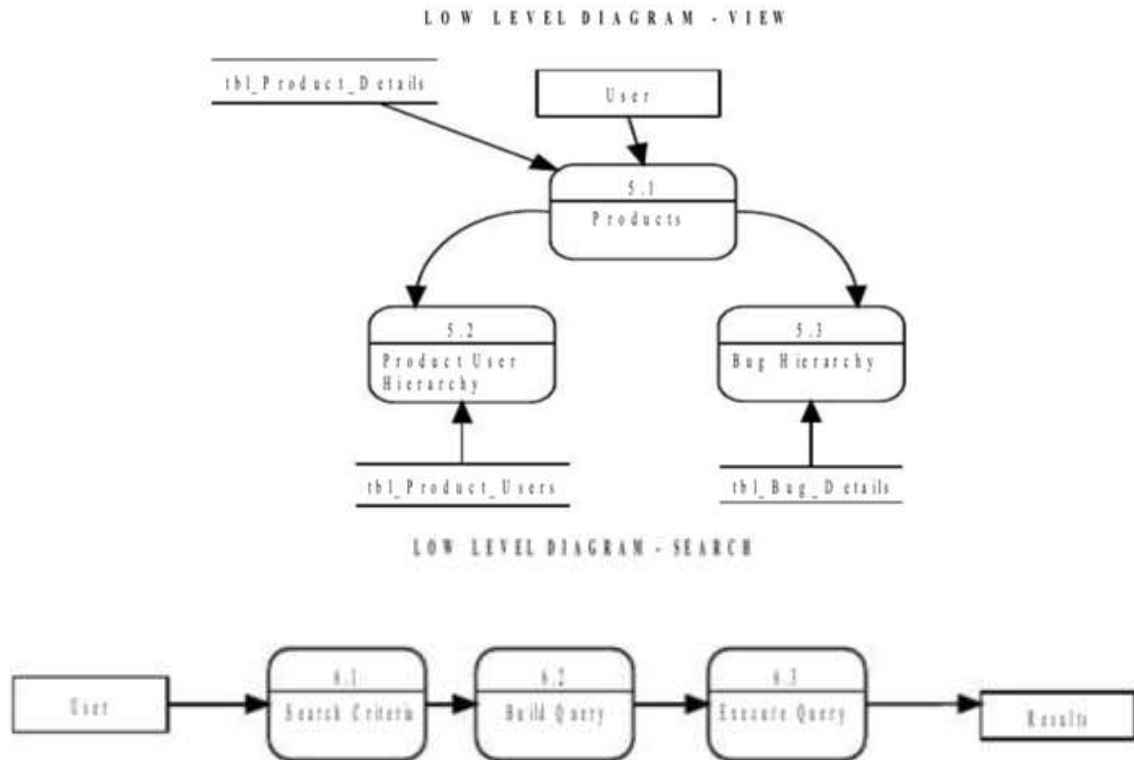
Portability: Yes this system is portable and we can switch the servers very easily.

Browser Compatibility: The project being web based required compatibility with at least the popular web browsers. Microsoft Windows XP and above, Linux and Macintosh being the current popular operating system and Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari and Google Chrome being the currently popular web browser.

Low Level Design:

Low Level Design:



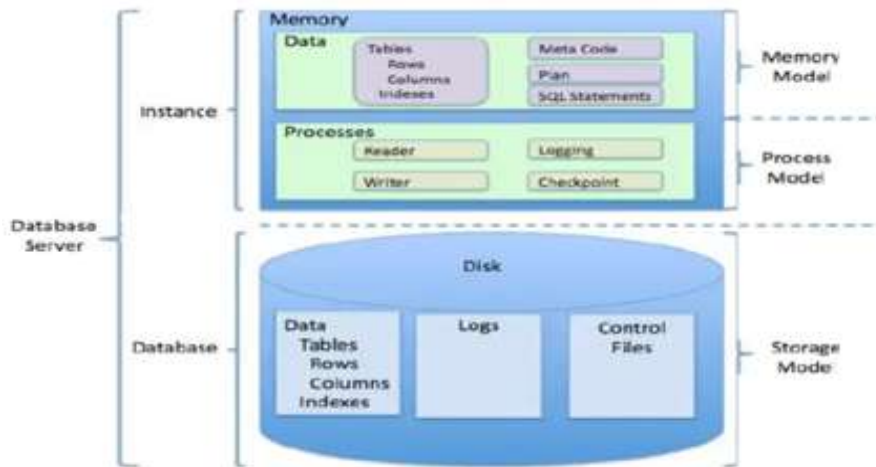


Database:

Relational Database Management System (RDBMS) This is an RDBMS based project which is currently using MySQL for all the transaction statements. MySQL is an opensource RDBMS System.

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMSs have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use. However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object relational impedance mismatch in relational database, and XML databases.



Implementation Methodology:

```
CREATE TABLE IF NOT EXISTS `login` (
  `login_id` int(11) NOT NULL AUTO_INCREMENT,
  `login_emp_id` int(11) NOT NULL,
  `login_user` varchar(255) NOT NULL,
  `login_password` varchar(255) NOT NULL,
  `login_level` int(11) NOT NULL DEFAULT '3',
  `login_email` varchar(100) NOT NULL,
  PRIMARY KEY (`login_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=29 ;
```

```
CREATE TABLE IF NOT EXISTS `employee` (
  `employee_id` int(11) NOT NULL AUTO_INCREMENT,
  `employee_manager_id` varchar(255) NOT NULL,
  `employee_role` varchar(255) NOT NULL,
  `employee_sal` varchar(100) NOT NULL,
  `employee_first_name` varchar(100) NOT NULL,
  `employee_middle_name` varchar(100) NOT NULL,
  `employee_last_name` varchar(100) NOT NULL,
  `employee_gender` varchar(100) NOT NULL,
  `employee_address` varchar(100) NOT NULL,
  `employee_village` varchar(100) NOT NULL,
  `employee_state` varchar(100) NOT NULL,
  `employee_country` varchar(100) NOT NULL,
  `employee_landline` varchar(100) NOT NULL,
  `employee_mobile` varchar(100) NOT NULL,
  `employee_email` varchar(100) NOT NULL,
  `employee_status` varchar(255) NOT NULL,
  `employee_department` varchar(255) NOT NULL,
  `employee_dob` varchar(255) NOT NULL,
  `employee_nationalty` varchar(255) NOT NULL,
  PRIMARY KEY (`employee_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1022 ;
```

```
CREATE TABLE IF NOT EXISTS `issue` (  
  `issue_id` int(11) NOT NULL AUTO_INCREMENT,  
  `issue_title` varchar(255) NOT NULL,  
  `issue_reported_id` varchar(255) NOT NULL,  
  `issue_type_id` varchar(255) NOT NULL,  
  `issue_description` varchar(255) NOT NULL,  
  `issue_product_version` varchar(255) NOT NULL,  
  `issue_date_raised` varchar(255) NOT NULL,  
  `issue_status_id` varchar(255) NOT NULL,  
  `issue_assigned_id` varchar(255) NOT NULL,  
  `issue_date_closed` varchar(255) NOT NULL,  
  `issue_priority` varchar(255) NOT NULL,  
  PRIMARY KEY (`issue_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
CREATE TABLE IF NOT EXISTS `department` (  
  `dept_id` int(11) NOT NULL AUTO_INCREMENT,  
  `dept_name` varchar(255) NOT NULL,  
  PRIMARY KEY (`dept_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

Home | About Us | Contact Us | User Login | Admin Login

CUSTOMER TICKETING SERVICE

customer ticketing service

HOME ABOUT US USER LOGIN ADMINISTRATOR LOGIN CONTACT US

User-Login Form

Username *

Password *

[Forgot Password](#)

Home | About Us | Contact Us | User Login | Admin Login

CUSTOMER TICKETING SERVICE

customer ticketing service

Search Here

HOME ABOUT US USER LOGIN ADMINISTRATOR LOGIN CONTACT US

Administrator Login Form

Username *

Password *

Login Cancel

[Forgot Password](#)

Welcome prodlvi a | Logout

CUSTOMER TICKETING SERVICE

customer ticketing service

Search Here

HOME ABOUT US MY ACCOUNT • ADD NEW • REPORTS CHANGE PASSWORD CONTACT US LOGOUT

Welcome to Issue Tracking System:

- Add User
- Add Issue
- Add Software Versions
- User Report
- Issue Report
- Software Report
- Change Password
- Logout

Welcome pmulhvi | Logout

CUSTOMER TICKETING SERVICE

customer ticketing service

[HOME](#) [ABOUT US](#) [MY ACCOUNT](#) [+ ADD NEW](#) [+ REPORTS](#) [CHANGE PASSWORD](#) [CONTACT US](#) [LOGOUT](#)

Registration Form

Solution* <input type="text" value="Please Select"/>	State* <input type="text" value="Please Select"/>	Department* <input type="text" value="Please Select"/>
First Name* <input type="text"/>	Middle Name* <input type="text"/>	Last Name <input type="text"/>
Gender <input type="text" value="Please Select"/>	Date of Birth <input type="text"/>	Nationality <input type="text"/>
E-mail <input type="text"/>	Landline <input type="text"/>	Mobile <input type="text"/>

Address Details

Address <input type="text"/>	Village/City/Town <input type="text"/>	State <input type="text" value="Please Select"/>
Country <input type="text" value="Please Select"/>		

Role & Manager

Role* <input type="text" value="Please Select"/>	Manager Code* <input type="text" value="Please Select"/>
---	---

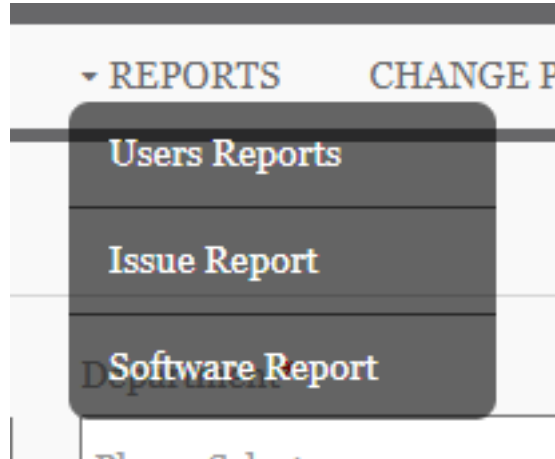
▼ ADD NEW ▼ REPORTS

Add User

Add Issue/Feature

Add Version **Department***

cf ▼ Please Select



Welcome prudhvi | Logout

CUSTOMER TICKETING SERVICE

customer ticketing service

Search Here

HOME ABOUT US MY ACCOUNT ADD NEW REPORTS CHANGE PASSWORD CONTACT US LOGOUT

Empoloyee Report

Employee Code	Name	Mobile	Department	Status	Email	Action
1001	prudhvi	09876543212	IT Department	Consultant	kaushal@gmail.com	Select Action
1022	sai	888573493	Java Developement	Consultant	kprudhvi25@gmail.com	Select Action
1023	prudhvi	77684593	IT Department	Permanent Employee	kprudhvi5@gmail.com	Select Action
1024	KING	77684593	Java Developement	Permanent Employee	kprudhvi26@gmail.com	Select Action
1025	ARIVAND	77684593	Java Developement	Permanent Employee	kprudhvi27@gmail.com	Select Action
1026	kadharsir	77684593	Java Developement	Permanent Employee	kprudhvi125@gmail.com	Select Action
1027	padma	77684593	HR Department	Permanent Employee	kprudhvi511@gmail.com	Select Action
1028	nag	77684593	IT Department	customer	kprudhvi2335@gmail.com	Select Action
1029	PRUDHVI25	77684593	IT Department	Permanent Employee	kprudhvi1255@gmail.com	Select Action
1030	prudhvi	77684593	Java Developement	Permanent Employee	kprudhvi20@gmail.com	Select Action
1031	sravan	77684593	HR Department	Permanent Employee	kprudhvi645@gmail.com	Select Action
1032	prudhviasfr	77684593	Java Developement	Permanent Employee	kprudhvi265@gmail.com	Select Action

CUSTOMER TICKETING SERVICE
 customer ticketing service

Welcome prudhvi s | Logout

Search Here

HOME ABOUT US MY ACCOUNT - ADD NEW - REPORTS CHANGE PASSWORD CONTACT US LOGOUT

Issue Report

ID	Title	Priority	Type	Status	Assigned to	Action
1	hsddjhfgsdkjf	High Priority	Bug	Resolved	prudhvi v	Edit
7	not able to get	High Priority	new issue	Released	KING G	Edit
8	not able to get	High Priority	new issue	Released	ARIVAND H	Edit
9	how to sleep	High Priority	new issue	Released	kadharsir s	Edit
10	how to drink	High Priority	new issue	Released	kadharsir s	Edit
11	how to play	High Priority	internal issue	Released	KING G	Edit
12	happy	High Priority	new issue	Released	kadharsir s	Edit
13	how to drink	High Priority	new issue	Released	KING G	Edit
15	how to drink	High Priority	new issue	Released	prudhvi K	Edit
16	how to drinkdfthydt	High Priority	new issue	Released	KING G	Edit
17	how to eat	High Priority	new issue	Released	prudhvi K	Edit
18	sgertwert	High Priority	new issue	Released	sal f	Edit
19	how to fly	High Priority	new issue	Released	kadharsir s	Edit
20	how to drink juice	High Priority	new issue	Released	KING G	Edit
21	hweiqe	High Priority	new issue	Released	nag d	Edit
22	not able to get	High Priority	new issue	Assigned	nag d	Edit
23	how to assign a new issue to agent	High Priority	new issue	Released	PRUDHVI25 s	Edit
24	hsddjhfgsdkjf	High Priority	new issue	Released	prudhvi v	Edit
2	Error in Login	Low Priority	Bug	Resolved	ARIVAND H	Edit
14	not able to get	Low Priority	new issue	Released	ARIVAND H	Edit
25	fjerfwerfg	Information	new issue	Released	prudhvi asfr s	Edit

CUSTOMER TICKETING SERVICE
 customer ticketing service

HOME ABOUT US MY ACCOUNT - ADD NEW - REPORTS CHANGE PASSWORD

Issue Creation Form

Title*

Assigned To*

Status*

Date*

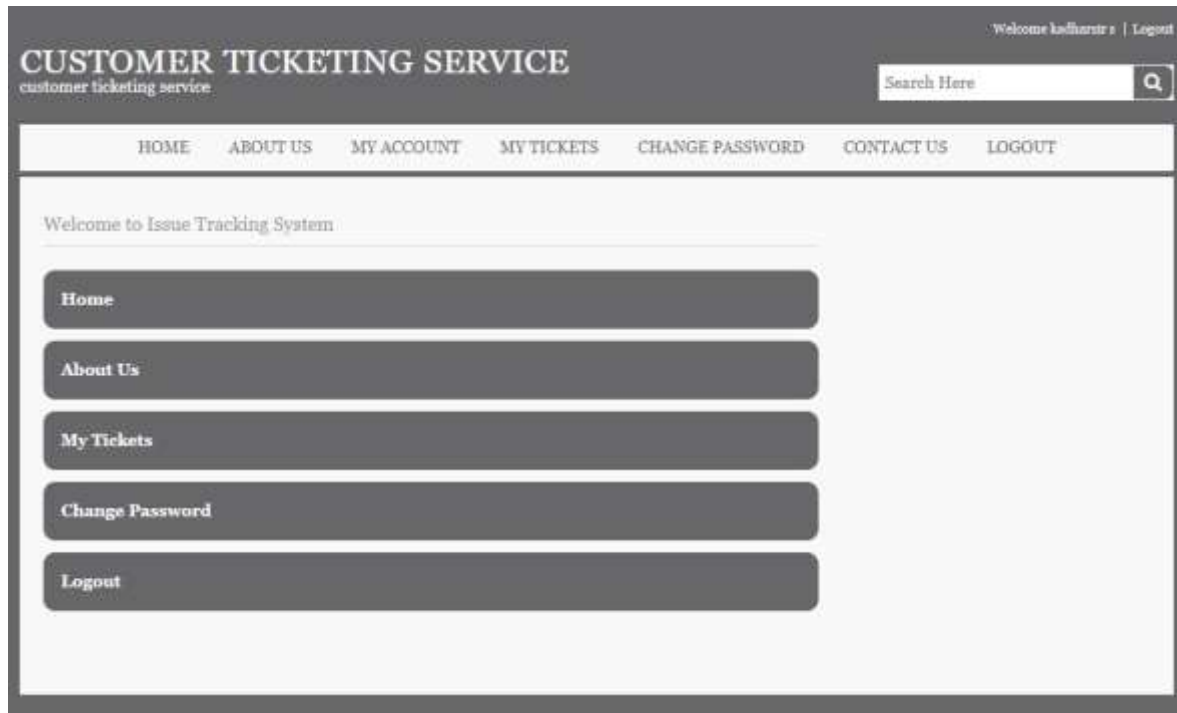
Description*

Issue Type*

Product Version*

Priority*

Date Closed*



Logic of employee

```
Employee employeeDetails = new Employee();
int employee_id = Integer.parseInt(request.getParameter ("employee_id"));
String male = "";
String female = "";
HashMap Values = employeeDetails.getEmployeeDetails(employee_id);
if(Values.get("employee_gender").equals("Male"))
{
    male = "Selected";
}
else if(Values.get("employee_gender").equals("Female"))
{
    female = "Selected";
}
```

```
Login login_obj = new Login();
Employee employee_obj = new Employee();

/////Check Username and Password for the
Employee /////
if((request.getParameter("employee_id")).equals("")
)) {
if(login_obj.checkUsernameExists(request.getParam
eter("employee_user"),1) >= 1) {
response.sendRedirect("../employee.jsp?employee_
id=0&type=error&msg=Username already exists.
Kindly choose another !!!!!!!");
return;
}
}
```

```
if(login_obj.checkUsernameExists(request.getParam
eter("employee_email"),2) >= 1) {
response.sendRedirect("../employee.jsp?employee_
id=0&type=error&msg=Email ID already exists.
Kindly choose another !!!!!!!");
return;
}
}

if((request.getParameter("act")).equals("Save"))
{
HashMap employee_data = new HashMap();
String managerID =
request.getParameter("employee_manager_id");
```

```

String          roleID          =          {
request.getParameter("employee_role");    out.print(employee_obj.saveEmployee(employee_d
ata));
if(managerID == null) {                  if(session.getAttribute("login_level") == null)
managerID = "1001";                       {
}                                           response.sendRedirect("../login.jsp?msg=You have
registered successfully.<br>Login toaccess your
account.");
if(roleID == null) {                      }
roleID = "3";                             }
}                                           else
{
employee_data.put("employee_id",request.getPara
meter("employee_id"));                    response.sendRedirect("../report-
employee.jsp?msg=Employee          Saved
Successfully");
employee_data.put("employee_user",request.getPar
ameter("employee_user"));                }
employee_data.put("employee_password",request.
getParameter("employee_password"));      }else
employee_data.put("employee_sal",request.getPara
meter("employee_sal"));                    {
employee_data.put("employee_first_name",request
.getParameter("employee_first_name"));    employee_data.put("employee_id",request.getPara
meter("employee_id"));
employee_data.put("employee_middle_name",requ
est.getParameter("employee_middle_name")); out.print(employee_obj.updateEmployee(employee
_data));
employee_data.put("employee_last_name",request.
getParameter("employee_last_name"));    if(request.getParameter("employee_id").equals(sess
ion.getAttribute("login_emp_id")))
{
employee_data.put("employee_gender",request.get
Parameter("employee_gender"));           response.sendRedirect("../employee.jsp?employee_
id="+session.getAttribute("login_emp_id")"&msg=
Account Updated Successfully")
employee_data.put("employee_address",request.get
Parameter("employee_address"));         }
employee_data.put("employee_village",request.get
Parameter("employee_village"));         else
employee_data.put("employee_state",request.getPa
rameter("employee_state"));             {
employee_data.put("employee_country",request.ge
tParameter("employee_country"));        response.sendRedirect("../report-
employee.jsp?msg=Employee          Updated
Successfully");}
employee_data.put("employee_landline",request.ge
tParameter("employee_landline"));       }
employee_data.put("employee_mobile",request.get
Parameter("employee_mobile"));          if((request.getParameter("act")).equals("delete"))
{
employee_data.put("employee_email",request.getP
arameter("employee_email"));            int          employee_id          =
Integer.parseInt(request.getParameter("employee_i
d"));
employee_data.put("employee_status",request.getP
arameter("employee_status"));           if(employee_obj.deleteEmployee(employee_id))
response.sendRedirect("../register.jsp?msg=Record
Deleted Successfully");
employee_data.put("employee_deparment",request.
getParameter("employee_deparment"));    else
employee_data.put("employee_dob",request.getPar
ameter("employee_dob"));                response.sendRedirect("../register.jsp?msg=Proble
m in Deletion.");
employee_data.put("employee_nationalty",request.
getParameter("employee_nationalty")    }
employee_data.put("employee_manager_id",manag
erID);
employee_data.put("employee_role",roleID);

if((request.getParameter("employee_id")).equals(""))
{
Issue issueObj = new Issue();
String emp_id = "0";
emp_id          =          (String)
session.getAttribute("login_emp_id");
if((request.getParameter("act")).equals("Save"))
{

```



```

HashMap results = new HashMap();
results.put("issue_title",request.getParameter("issue_title"));
results.put("issue_reported_id",emp_id);
results.put("issue_type_id",request.getParameter("issue_type_id"));
results.put("issue_product_version",request.getParameter("issue_product_version"));
results.put("issue_description",request.getParameter("issue_description"));
results.put("issue_date_raised",request.getParameter("issue_date_raised"));
results.put("issue_status_id",request.getParameter("issue_status_id"));
results.put("issue_assigned_id",request.getParameter("issue_assigned_id"));
results.put("issue_date_closed",request.getParameter("issue_date_closed"));
results.put("issue_priority",request.getParameter("issue_priority"));
results.put("issue_id",request.getParameter("issue_id"));

if((request.getParameter("issue_id")).equals(""))
{
out.println(issueObj.saveIssue(results));
response.sendRedirect("../report-issue.jsp?emp_id="+emp_id+"&msg=Issue Saved Successfully");
}
else
{
results.put("issue_id",request.getParameter("issue_id"));
out.println(issueObj.updateIssue(results));
response.sendRedirect("../report-issue.jsp?emp_id=0&msg=Issue Updated Successfully");
}
}
%>

```

```

Login login_obj = new Login();
Employee employeeDetails = new Employee();

if((request.getParameter("act")).equals("chk_login"))
{
if(login_obj.checkLogin(request.getParameter("login_user"),request.getParameter("login_password")))
{
HashMap Values =
login_obj.getLoginDetails(request.getParameter("l

```

```

ogin_user"),request.getParameter("login_password"));

```

```

HashMap employeeValues =
employeeDetails.getEmployeeDetails(Integer.parseInt(Values.get("login_emp_id").toString()));
session.setAttribute("login_id",Values.get("login_id"));
session.setAttribute("login_emp_id",Values.get("login_emp_id"));
session.setAttribute("login_user",Values.get("login_user"));
session.setAttribute("login_level",Values.get("login_level"));
session.setAttribute("login_name",employeeValues.get("employee_first_name")+
"+employeeValues.get("employee_last_name"));
response.sendRedirect("../login-home.jsp?msg=You are login successfully.");
}
else
{
response.sendRedirect("../login.jsp?msg=Invalid User/Password. Please try again.....");
}
}
if((request.getParameter("act")).equals("change_password"))
{
if(login_obj.changePassword(request.getParameter("old_password"),
request.getParameter("new_password"),Integer.parseInt((String) session.getAttribute("login_id"))))
{
response.sendRedirect("../change-password.jsp?msg=Your Password has been changed successfully.");
}
else
{
response.sendRedirect("../change-password.jsp?msg=Old Password doesnot match. Please try again.....");
}
}
if((request.getParameter("act")).equals("logout"))
{
session.setAttribute("login_id",null);
session.setAttribute("login_level",null);
response.sendRedirect("../login.jsp?msg=You are logout successfully.....");
}
}
%>

```

```

Login login_obj = new Login();
if((request.getParameter("act")).equals("Save"))

```

```
{
HashMap login_data = new HashMap();
login_data.put("login_user",request.getParameter("
login_user"));
login_data.put("login_password",request.getParam
eter("login_password"));
login_data.put("login_email",request.getParameter(
"login_email"));
login_data.put("login_sal",request.getParameter("lo
gin_sal"));
login_data.put("login_first_name",request.getPara
meter("login_first_name"));
login_data.put("login_middle_name",request.getPa
rameter("login_middle_name"));
login_data.put("login_last_name",request.getParam
eter("login_last_name"));
login_data.put("login_gender",request.getParamete
r("login_gender"));
login_data.put("login_address",request.getParamet
er("login_address"));
login_data.put("login_village",request.getParamete
r("login_village"));
login_data.put("login_state",request.getParameter("
login_state"));
login_data.put("login_country",request.getParamet
er("login_country"));
login_data.put("login_landline",request.getParamet
er("login_landline"));
login_data.put("login_mobile",request.getParamete
r("login_mobile"));

if((request.getParameter("login_id")).equals(""))
{
login_obj.save_login(login_data);
if(request.getParameter("login_id").equals(""))
{
response.sendRedirect("../login.jsp?msg=You have
registered successfully.<br>Login to access your
account.");
}
else
{
response.sendRedirect("../report-
customer.jsp?msg=Customer Saved Successfully");
}
}
else
{
login_data.put("login_id",request.getParameter("lo
gin_id"));
login_obj.update_login(login_data);
if(request.getParameter("login_id").equals(session.
getAttribute("login_id")))
{
response.sendRedirect("../register.jsp?login_id=1&
msg=Account Updated Successfully");
```

```
}
else
{
response.sendRedirect("../report-
customer.jsp?msg=Customer Updated
Successfully");
}
}
}
}
if((request.getParameter("act")).equals("delete"))
{
int login_id =
Integer.parseInt(request.getParameter("login_id"));
if(login_obj.delete_login(login_id))
response.sendRedirect("../register.jsp?msg=Record
Deleted Successfully");
else
response.sendRedirect("../register.jsp?msg=Proble
m in Deletion.");
}
```

REFERENCES AND BIBLIOGRAPHY:

<http://www.bluedart.com/>
<http://www.JSP.net/>
<http://www.tutorialspoint.com/mysql/>
<httpd.apache.org/docs/2.0/misc/tutorials.html>