# `VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590018, Karnataka, India

**A MINI PROJECT REPORT**

**ON**

# "EMPLOYEE CHAT DATABASE SYSTEM"

**A Mini Project Report Submitted in partial fulfillment of the requirement for the degree of**

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by**


| | |
|---|---|
| **PRACHTI J B** | **1RG17CS009** |
| **PRAJWAL B MANI** | **1RG17CS037** |


**Under the Guidance of**


**Mrs. PUSHPLATA DUBEY**

Asst. Prof. Dept. of CSE
RGIT, Bengaluru-32


Department of Computer Science & Engineering

**RAJIV GANDHI INSTITUTE OF TECHNOLOGY**

Cholanagar, R.T. Nagar Post, Bengaluru-560036

**2019-2020**

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

## (Affiliated To Visvesvaraya Technological University)

**Cholanagar, R.T.Nagar Post, Bangalore-560032**

## Department of Computer Science Engineering

# CERTIFICATE

This is to certify that the Mini Project Report entitled **"EMPLOYEE CHAT DATABASE SYSTEM"** is a bonafide work carried out by **Ms. Prachiti J B (1RG17CS009) and Mr. Prajwal B Mani (1RG17CS037)** in partial fulfillment for the award of Bachelor of Engineering in Computer Science Engineering under Visvesvaraya Technological University, Belgavi, during the year 2019-2020. It is certified that all corrections/suggestions given for Internal Assessment have been incorporated in the report. This Mini Project report has been approved as it satisfies the academic requirements in respect of mini project.

| | |
|---|---|
| **Signature of Guide** | **Signature of HOD** |
| **Mrs. Pushplata Dubey** | **Mrs. Arudra A** |
| **Asst.Professor** | **Professor & HOD** |
| **Dept. of CSE** | **Dept. of CSE,** |
| **RGIT, Bengaluru** | **RGIT, Bengaluru** |

## External Viva

| **Name of the Examiners** | **Signature With Date** |
|---|---|
| 1. _____ | |
| 2. _____ | |

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**JNANA SANGAMA, BELAGAVI-590018**

## RAJIV GANDHI INSTITUTE OF TECHNOLOGY

### DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## DECLARATION

We hereby declare that the mini project entitled "EMPLOYEE CHAT DATABASE SYSTEM" **Visvesvaraya Technological University, Belagavi** during the academic year **2019-2020**, is record of an original work done by us under the guidance of **Mrs. Pusplata Dubey, Assistant Professor, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bengaluru** and this mini project is submitted in the partial fulfillment of requirements for the award of the degree of **Bachelor of Engineering in Computer Science** & **Engineering**. The results embodied in this thesis have not been submitted to any other University or Institute for award of any degree or diploma.

**PRACHITI J B (1RG17CS009)**

**PRAJWAL B MANI (1RG17CS037)**

# ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude and respect to the **Rajiv Gandhi Institute of Technology, Bengaluru** for providing us an opportunity to carry out our project work.

We express our sincere regards and thanks to **Dr. NAGRAJ A M, Principal, RGIT, Bengaluru** and **Mrs. Arudra A, Associate Professor and Head, Department of Computer Science & Engineering, RGIT, Bengaluru,** for her encouragement and support throughout the Project.

With profound sense of gratitude, we acknowledge the guidance and support extended by **Mrs. Pushplata Dubey, Asst. Prof, Department of Computer Science & Engineering, RGIT, Bengaluru.** Her incessant encouragement and valuable technical support have been of immense help in realizing this project. Her guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication and hard work.

We also extend our thanks to the entire faculty of the Department of CSE, RGIT, Bengaluru, who have encouraged us throughout the course of Bachelor Degree.

**PRACHITI J B (1RG17CS009)**

**PRAJWAL B MANI (1RG17CS037)**

# ABSTRACT

The main objective of this application is to ensure a effective communication between the employees via texts using client connection with the help of a local host server MySQL. This also keeps a record of all the resource managers controlling a particular project with a fixed number of employees assigned to each manager. This application has a history feature which shows every employee who has logged in. This also enables the employee to search for a desired message which is displayed along with the number of times it has been sent. This application provides the deadlines for the respective projects as well. There is a lot of scope in the current working models. This application is developed for a user friendly environment and to have a better time management.

# TABLE OF CONTENTS

# LIST OF TABLES AND FIGURES

# INTRODUCTION

# Chapter 1

# INTRODUCTION

The detailed description of the proposed project and developed application is as follows:

## 1.1 Project Overview

The is a computer based application which helps one end user to communicate with 'n' number of users via text. This keeps a record of all the messages sent by the users and gives the detailed information of the projects and the employees working for a particular project. This application also gives access to the resource manager who controls the project and is assigned to a group of users.

## 1.2 Project Description

The proposed system ensures that the user has an effective communication with the user on the other side and an efficient mechanism benefiting the users. The main objective of this application is that it works on the concept of TCP client/server communication which provides an user friendly environment, problem solving and a trustworthy system for the storage of data. It seeks to avoid redundant data entry using the framework of a modular design to cater the specific needs. The application aims at ensuring a user friendly design with the goal to facilitate system wide integration to increase efficiency and ensuring uniform services. The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of MY-SQL Server and all the user interfaces have been designed using the technologies. The database connectivity is planned using the "SQL Connection" methodology. The standards of security and data protective mechanism have been given a big choice for proper usage.

The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff. The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system.

The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. The MY-SQL server was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using java i.e. Eclipse 4.8 (Photon).

At all proper levels high care was taken to check that the system manages the data consistency with proper validations. The database connectivity was planned using the latest "SQL Connection" technology provided by Microsoft Corporation. The authentication and authorization were crosschecked at all the relevant stages.

## 1.3 Scope of The Project

As for the scope, the project will be tested as the program is being developed. A database for the users registered will be developed and tested, a menu will be developed and tested, a client/server interface will be developed and tested, GUI's will be developed and tested, for the user's benefits. When the application is near completion, more testing will be done in order to make it less buggy and more user friendly. Since the application is written in java, while developing it practice of different techniques in java and working on it as much as possible will help in inculcating java skills which are useful for the near future.

## 1.4 Aim of The Project

The main aim of our project is that it gives the ability to chat with whoever is online on the application. These users are employees itself who are working under a resource manager who is assigned for a project under a unique PID. This application mainly serves as a nodal point which is used to store information of the users and also it keeps track of various functionalities such as online history which will be tracked every time the user refreshes the frame. Measures are taken as well to reduce the number of frames as much as possible.

# REQUIREMENT SPECIFICATION

# Chapter 2

# REQUIREMENT SPECIFICATION

## 2.1 Details of Software and Languages

### 2.1.1 Introduction to MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL basically has the following features:

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

### 2.1.2 Introduction to Eclipse

Eclipse IDE is a modular, standards-based integrated development environment (IDE), written in the Java programming language. The Eclipse project consists of a full- featured open source (IDE) written in the Java programming language and a rich client application platform, which can be used as a generic framework to build any kind of application.
Eclipse IDE runs on the Java SE Development Kit(JDK) which consists of the Java Runtime Environment and developer tools for compiling, debugging and running applications written in the Java language.

### 2.1.3 Introduction to Java

Java is a general-purpose computer-programming language that is concurrent, classe- based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled java code can run on all platforms that support java without the need for recompilation. Java application are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

### 2.1.4 Introduction to XAMPP

XAMPP is an Open Source web server with all the tools and language support built-in to it. It's ready to use and makes it easier for developers to take their code and host it locally and test the same. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes

Everything needed to set up a web server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. In XAMPP – Apache is a Server application, MySQL is Database, and PHP is Server Side Scripting Language and all these are included as an extractable/executable file.

## 2.2 System Requirements

### 2.2.1 User Requirements

- Easy to understand and should be simple.
- The built in functions should be utilized to the maximum extent.
- Database applications shoulssd be used.

### 2.2.2 Software Requirements:

- OS : Windows10 OS
- Frontend : Core java
- Data Base Server : MySQL
- IDE : Eclipse Java 2019-09

### 2.2.3 Hardware Requirements:

- A visual display such as a monitor
- Input devices such as Keyboard and Mouse
- APUs or Intel Core Processor having a clock speed of at least 2.0 GHz
- At least 100 MB of additional free memory or RAM
- At least 80 GB Hard Disk Drive

# SYSTEM DESIGN

# CHAPTER 3

# SYSTEM DESIGN

## 3.1 Schema Diagram

A database schema is the skeleton that represents the logical view of the entire database. It defines how the data is organized and how the relation among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designer who design the schema to help the programmer understand the database and make it useful.

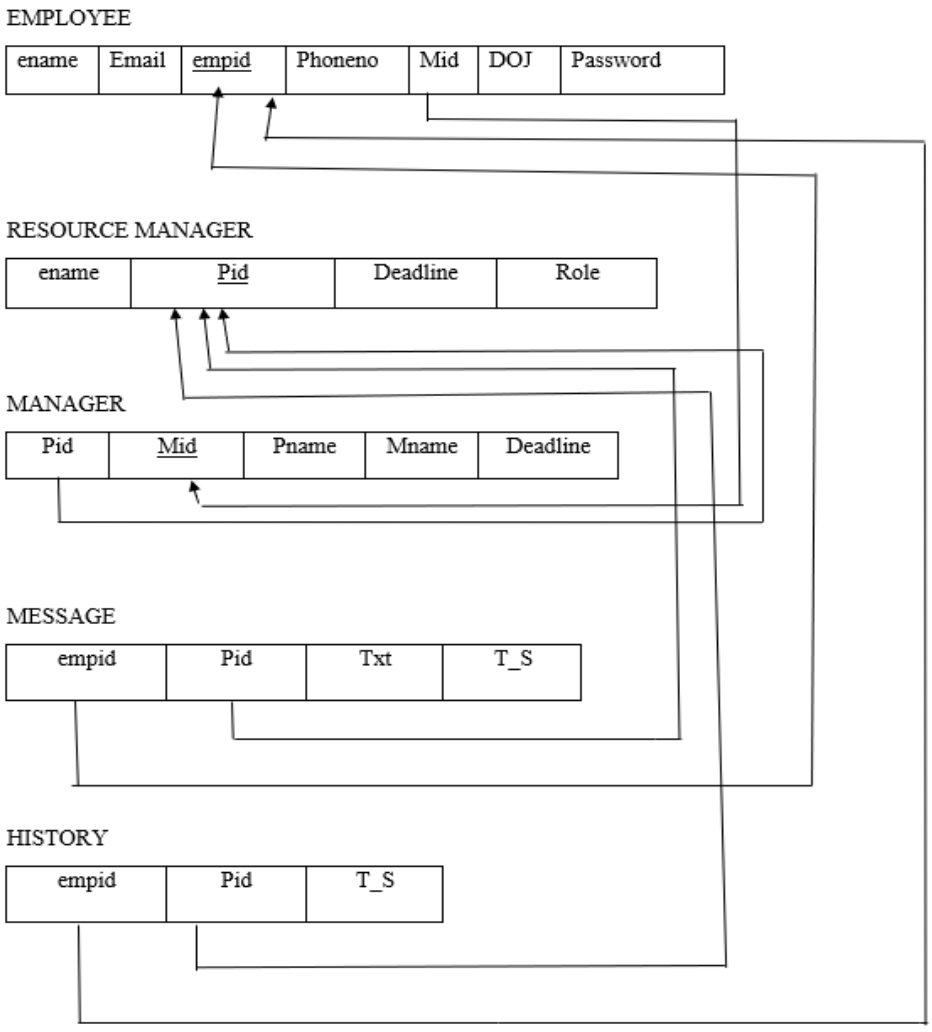


**Figure 3.1 : Schema diagram of employee chat database system**

## 3.2 Tables

### 3.2.1 : Employee Table

| Name | Null? | Type |
|---|---|---|
| Empid | NOT NULL | Int(15) |
| Ename | | Varchar(25) |
| Email | | Varchar(25) |
| Phoneno | | Int(15) |
| DOJ | | Date |
| Password | | Varchar(25) |
| MEid | | Int(10) |

### 3.2.2 : Message Table

| Name | Null? | Type |
|---|---|---|
| Empid | | Int(15) |
| PId | | Int(15) |
| Txt | | Text |
| T_S | | Date |

### 3.2.3 : History Table

| Name | Null? | Type |
|---|---|---|
| Empid | | Int(15) |
| PId | | Int(15) |
| T_S | | Date |

### 3.2.4 : Manager Table

| Name | Null? | Type |
|---|---|---|
| MEid | NOT NULL | Int(10) |
| Mname | | Varchar(25) |
| Pname | | Varchar(25) |
| PId | | Int(15) |
| Deadline | | Date |

### 3.2.5 : Resource Manager Table

| Name | Null? | Type |
|---|---|---|
| Ename | | Varchar(25) |
| PId | NOT NULL | Int(15) |
| Role | | Varchar(25) |
| Deadline | | Date |

## 3.3 ER Diagram

ER diagram shows the diagrammatic representation of entities and attributes with primary key and shows the association and the participation to their entity.

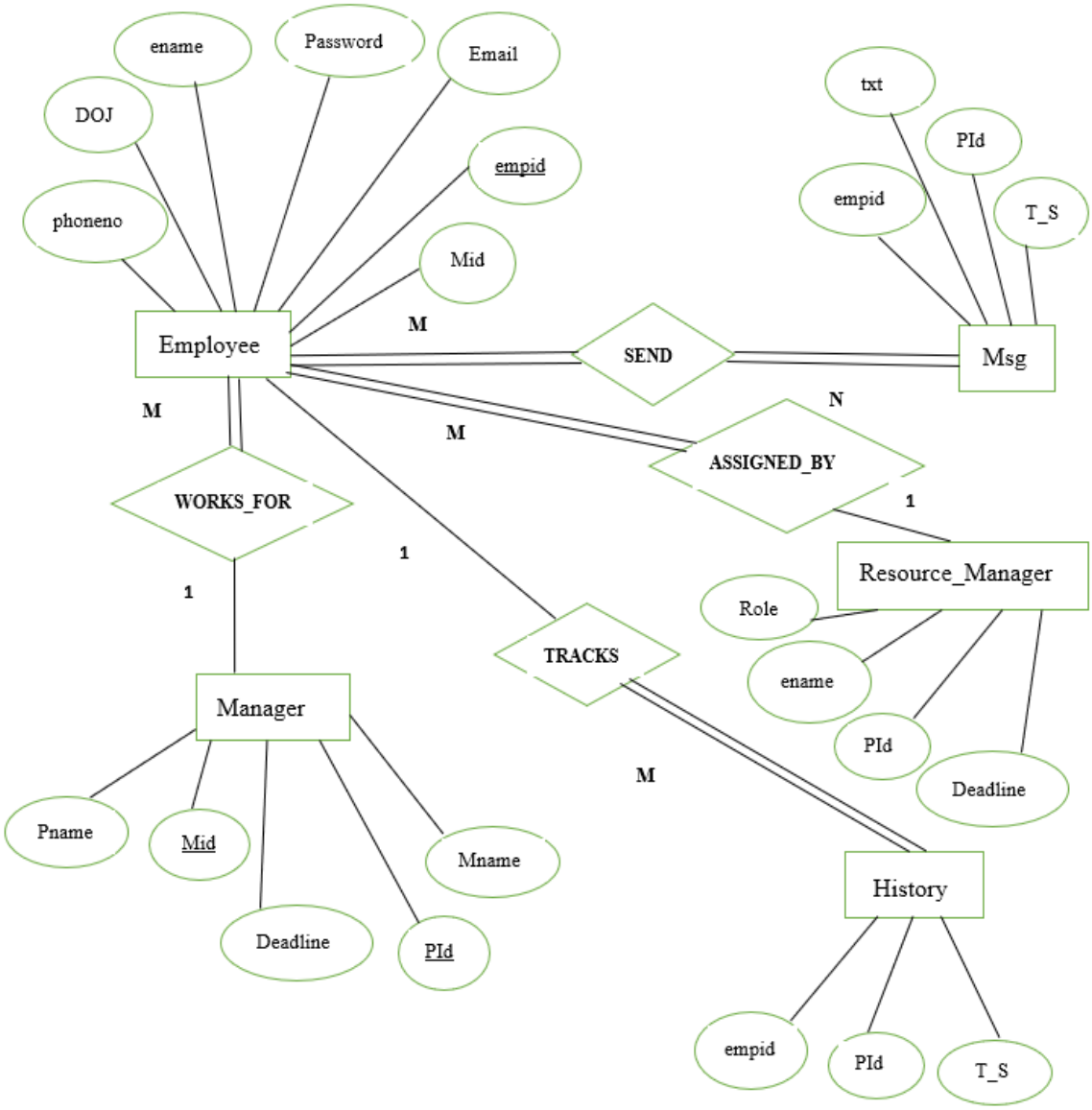


**Figure 3.2: Entity Relationship diagram of employee chat database system**

# SYSTEM IMPLEMENTATION

# CHAPTER 4

# SYSTEM IMPLEMENTATION

## 4.1 Employee login page

```
btnSignin.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
String sql="select * from employee where Empid =? and Password= ? ";
try {
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1,txtEnterUserid.getText());
pstmt.setString(2, new String(passwordField.getPassword()));
ResultSet rs = pstmt.executeQuery();
if(rs.next()) {
            msg c=new msg(Integer.parseInt(txtEnterUserid.getText()) );
            c.setVisible(true);
            dispose();
            }else {
JOptionPane.showMessageDialog(null, "Invalid Credentials",
login",JOptionPane.ERROR_MESSAGE);}
conn.close();
}
catch(Exception e1) {
e1.printStackTrace();
}String sql1="INSERT INTO histroy(Empid, pid, ts) VALUES(?,?,?)";
```

```
try {
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement pstmt = conn.prepareStatement(sql1);
pstmt.setInt(1, Integer.parseInt(txtEnterUserid.getText()));
pstmt.setInt(2,  comboBox.getSelectedItem());
DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
Date date = new Date();
pstmt.setDate(3,new java.sql.Date(date.getTime()));
pstmt.executeUpdate();}
catch(Exception e1) {
e1.printStackTrace();
}}});
```

## 4.1 Employee signup page

```
public void mouseClicked(MouseEvent e) {
String sql="INSERT INTO employee(Ename, Empid, Email, Phoneno, MEid, Password,
DOJ) VALUES (?,?,?,?,?,?,?) ";
try {
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1,txtName.getText());
pstmt.setInt(2, Integer.parseInt(txtEmpid.getText()));
pstmt.setString(3, txtEmailId.getText());
pstmt.setLong(4, Long.parseLong( txtPhoneNo.getText()));
pstmt.setInt(5, Integer.parseInt(txtMempid.getText()));
pstmt.setString(6, txtEnterPassword.getText());
```

```
DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
Date date = new Date();

System.out.println(dateFormat.format(date));
pstmt.setDate(7,new java.sql.Date(date.getTime()));
pstmt.executeUpdate();
msg c=new msg( Integer.parseInt(txtEmpid.getText()));
c.setVisible(true)
setVisible(false);
dispose();
}
catch(Exception e1) {
e1.printStackTrace();
}}});
```

## 4.2 Reset password page

```
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1,textField.getText());
ResultSet rs = pstmt.executeQuery();
if(rs.next()) {
if((Arrays.equals(passwordField.getPassword(), passwordField_1.getPassword()))    ) {
String sql2="UPDATE employee SET Password =? WHERE Eid=?";
Class.forName("com.mysql.jdbc.Driver");
```

```
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement pstmt1 = conn1.prepareStatement(sql2);
pstmt1.setInt(1, Integer.parseInt( textField.getText()) );
pstmt1.setString(2,new String(passwordField.getPassword()));

pstmt1.executeUpdate();

login l=new login();
setVisible(false);
l.setVisible(true);
}
else{
JOptionPane.showMessageDialog(null, "Passwords Doesnt match",
 match",JOptionPane.ERROR_MESSAGE);}}
else {

JOptionPane.showMessageDialog(null, "Invalid Eid",

JOptionPane.ERROR_MESSAGE);

}}
catch(Exception e1){
      e1.printStackTrace();
}}});
```

## 4.3 Online page

```
button.addMouseListener(new MouseAdapter() {
public void mouseClicked(MouseEvent e) {
String sql="Select ename From employee Where empid in(select Empid from histroy
where ts=?)";
```

```
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement ps = conn.prepareStatement(sql);
DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
Date date = new Date();
ps.setDate(1,new java.sql.Date(date.getTime()));
ResultSet rs = ps.executeQuery();
textArea_1.setText("");
while(rs.next()){
textArea_1.append(rs.getString(1)+"\n");}
}
catch(Exception e1){
e1.printStackTrace();
}}});
```

## 4.4 Message page

```
btnNewButton.addMouseListener(new MouseAdapter() {@Override
public void mouseClicked(MouseEvent e) {
String sql3="SELECT employee.Ename, `txt`, `time` FROM `msg` INNER JOIN
employee WHERE msg.Empid=employee.Empid";
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement ps = conn.prepareStatement(sql3);
ResultSet rs = ps.executeQuery();
while(rs.next()){
```

```
textArea.append(rs.getString(1)+":"+rs.getString(2)+"\t\t\t\t"+rs.getString(3)+"\n");
textArea.append("-----------------------------------------------\n");
}}
catch(Exception e1){
e1.printStackTrace();
}}});
btnNewButton.addMouseListener(new MouseAdapter() {@Override
public void mouseClicked(MouseEvent e) {
String sql3="SELECT employee.Ename, `txt`, `time` FROM `msg` INNER JOIN
employee WHERE msg.Empid=employee.Empid";
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement ps = conn.prepareStatement(sql3);
ResultSet rs = ps.executeQuery();
while(rs.next()){
textArea.append(rs.getString(1)+":"+rs.getString(2)+"\t\t\t\t"+rs.getString(3)+"\n");
textArea.append("-----------------------------------------------\n");
}}
catch(Exception e1) {
e1.printStackTrace();
}}});
```

## 4.5 Resource Manager page

```
btnAdd.addMouseListener(new MouseAdapter() {
public void mouseClicked(MouseEvent e) {
String sql="INSERT INTO rs(Ename, pid, role, deadline) VALUES (?,?,?,?)";
```

```
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1, jc.getSelectedItem().toString());
pstmt.setInt(2,  (int) comboBox.getSelectedItem());
pstmt.setString(3, comboBox_1.getSelectedItem().toString());
pstmt.setString(4, textField.getText() );
pstmt.executeUpdate();
JOptionPane.showMessageDialog(null,"Successfully
Updated.","Alert",JOptionPane.INFORMATION_MESSAGE);
}
catch(Exception e1){
e1.printStackTrace();
}}});
```

## 4.1 Resource Manager view page

```
btnView.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
String sql="SELECT manger.Mid,manger.PId,manger.pname,manger.Mname,rs.deadline
FROM manger NATURAL JOIN rs GROUP BY manger.PId" ;
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement ps = conn.prepareStatement(sql);
ResultSet rs = ps.executeQuery();
```

```
table.setModel(DbUtils.resultSetToTableModel(rs));
}
catch(Exception e1){
e1.printStackTrace();
}}});
```

## 4.8 Search page

```
button.addMouseListener(new MouseAdapter() {
public void mouseClicked(MouseEvent e) {
String sql="{ call smsgs(?,?) } ";
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn = (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
CallableStatement stmt = (CallableStatement) conn.prepareCall(sql) ;
stmt.setString(1, textField.getText());
stmt.registerOutParameter(2, java.sql.Types.INTEGER);
stmt.executeUpdate();
int count=stmt.getInt(2);
textField_1.setText("Total messages displayed:"+count);
}
catch(Exception e1){
e1.printStackTrace();
}
String sql1="SELECT employee.Ename, `txt`, `time` FROM `msg1` INNER JOIN employee WHERE msg1.Empid=employee.Empid";
try{
```

```
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement ps = conn.prepareStatement(sql1);
ResultSet rs = ps.executeQuery();
while(rs.next()){
textArea.append(rs.getString(1)+":"+rs.getString(2)+"\t\t\t\t"+rs.getString(3)+"\n");
textArea.append("-----------------------------------------------\n");
}}
catch(Exception e1){
e1.printStackTrace();
}}});
```

## 4.9 To check for wrong resource manager entries

```
comboBox1.addMouseListener(new MouseAdapter() {
public void mouseClicked(MouseEvent e) {
String sql="SELECT `Ename`FROM `rs` WHERE `pid`=0;";
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/chat","root","");
PreparedStatement pstmt = conn.prepareStatement(sql);
 ResultSet rs = pstmt.executeQuery();
 while(rs.next()) {comboBox1.addItem(rs.getString(1));}
 }
 catch(Exception e1){
 e1.printStackTrace();
```

}}});

## 4.10 Trigger Code

```
CREATE TRIGGER `rsc` BEFORE INSERT ON `rs`
 FOR EACH ROW IF((SELECT COUNT(*) FROM rs WHERE role LIKE '%Software architect%' AND pid=pid)>2) THEN
 SET NEW.pid=0;
SET NEW.role='Unknown';
SET NEW.deadline='0000-00-00';
END IF
```

## 4.11 Stored Procedure

```
 CREATE TRIGGER `rsc` BEFORE INSERT ON `rs`
 FOR EACH ROW IF((SELECT COUNT(*) FROM rs WHERE role LIKE '%Software architect%' AND pid=pid)>2) THEN
 SET NEW.pid=0;
 SET NEW.role='Unknown';
 SET NEW.deadline='0000-00-00';
END IF
```

# SCREENSHOTS

# CHAPTER 5

# SCREENSHOTS

## 5.1 Employee Login Page

Employee is asked to enter empid and password. Only when the empid and password matches, the employee is logged in. This page facilitates selection of the particular project ID and also if not employee asks for the verified resource manager

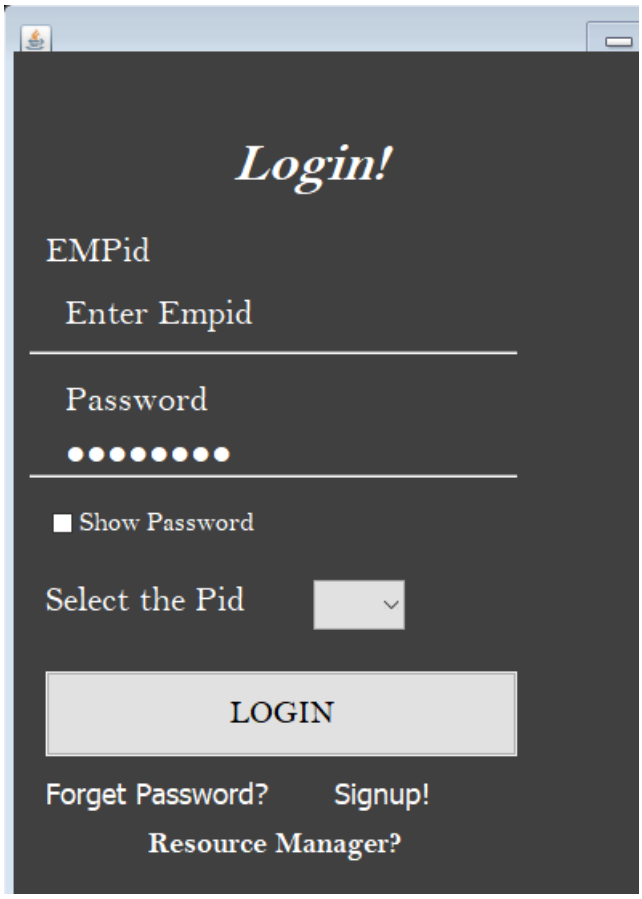


**Figure 5.1 : Screenshot of Employee Login Page**

## 5.2 Employee Sign Page

This page allows the employees to signup in their respective chat logs.

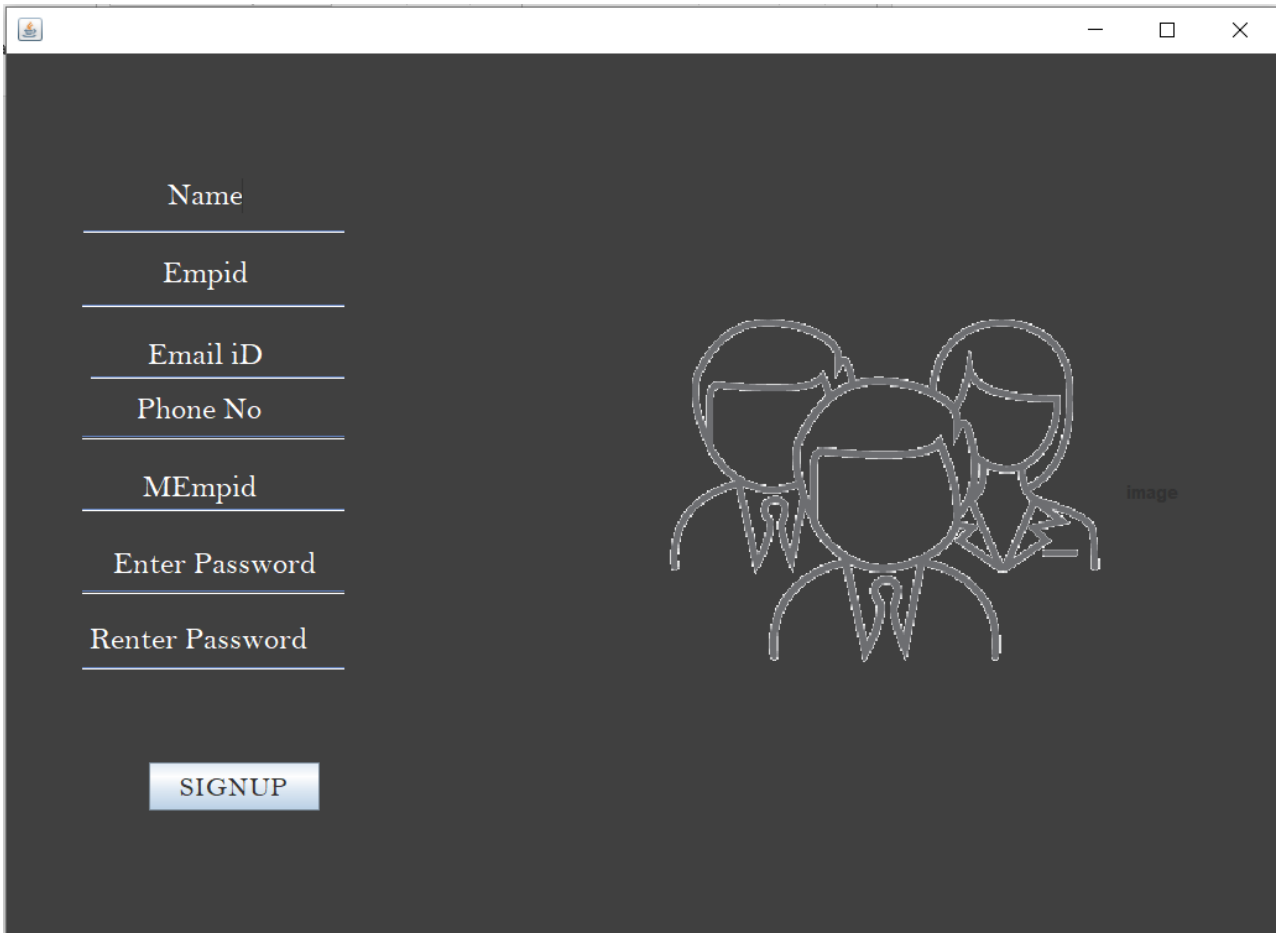


**Figure 5.2 : Screenshot of Employee Signup Page**

## 5.3 Password Reset Page

This page helps the employee to change the password in case it is required, if the password entered is invalid, the updations are not acceptable.

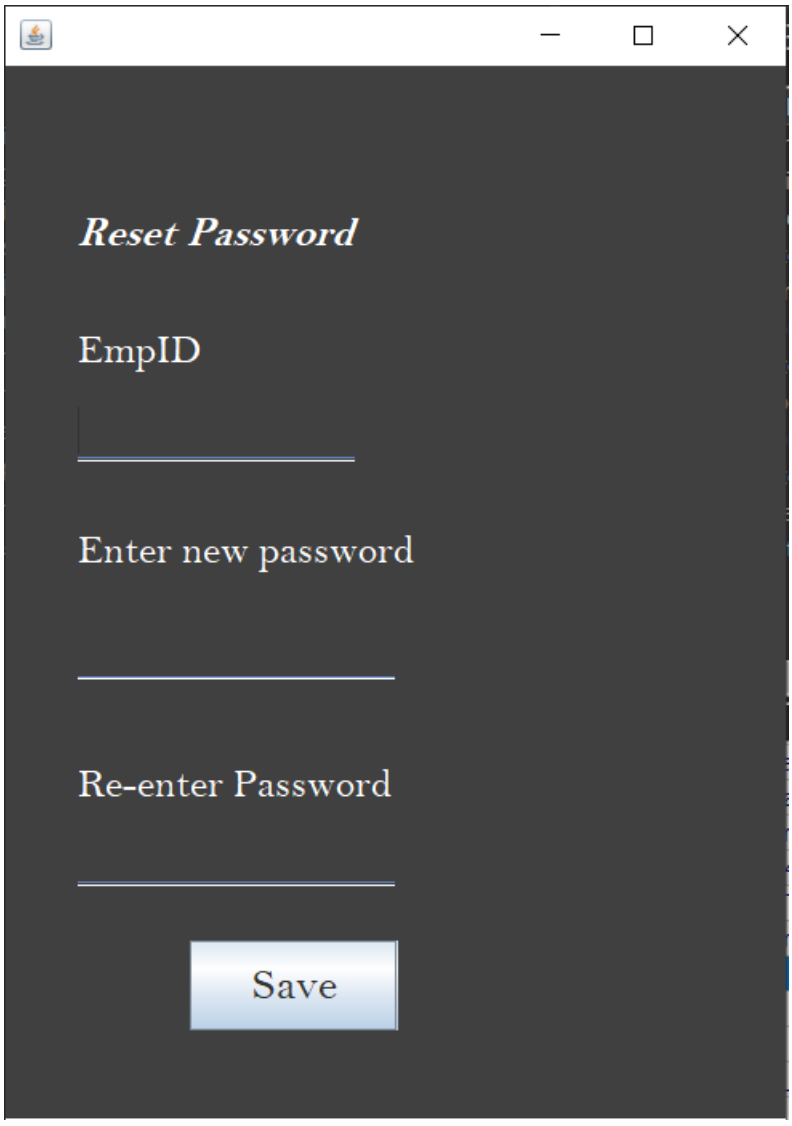


**Figure 5.3: Screenshots of Password Reset Page**

## 5.4 Message Page

This page facilitates a user friendly framework which gives the details of the employees currently online. This also includes a search panel which helps the employee to find the required message from the chat logs.

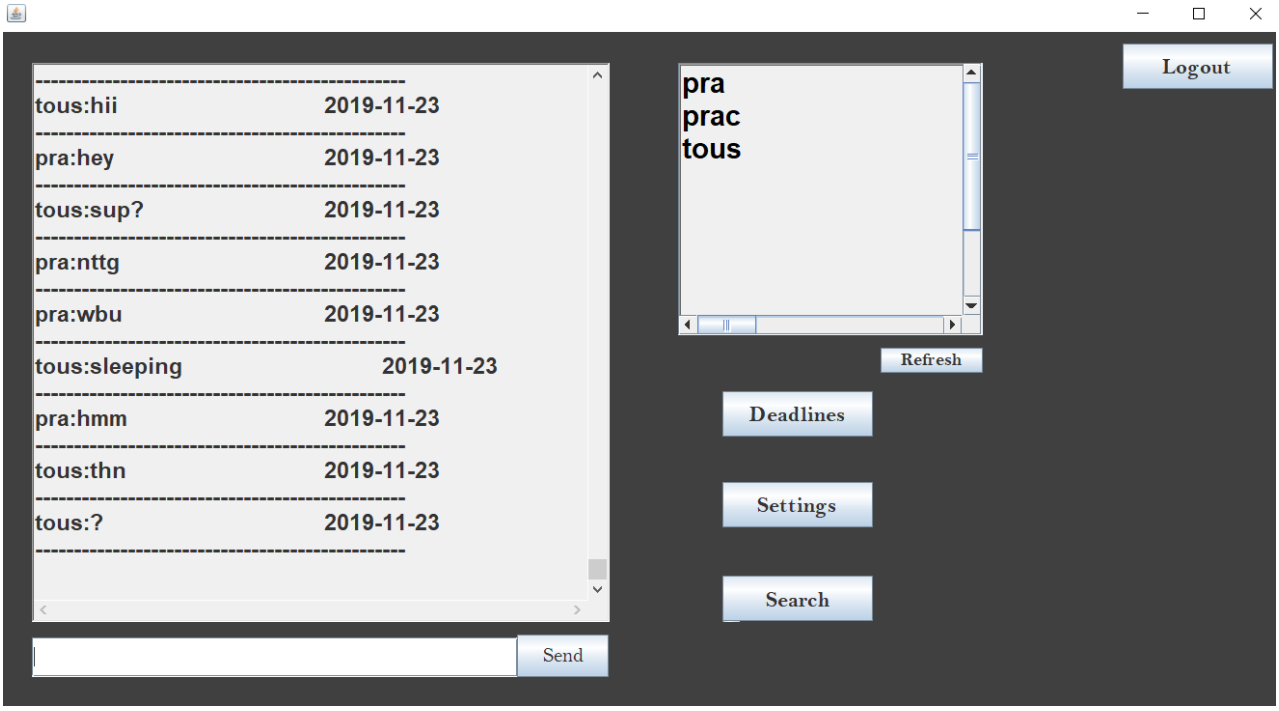


**Figure 5.4 : Screenshot of Message Page**

## 5.5 Message Search Page

The message search page is useful in case the employees wants to know the whereabouts of a particular message along with the date. This page also has a feature as it helps the employee know the number of times a particular message was sent.

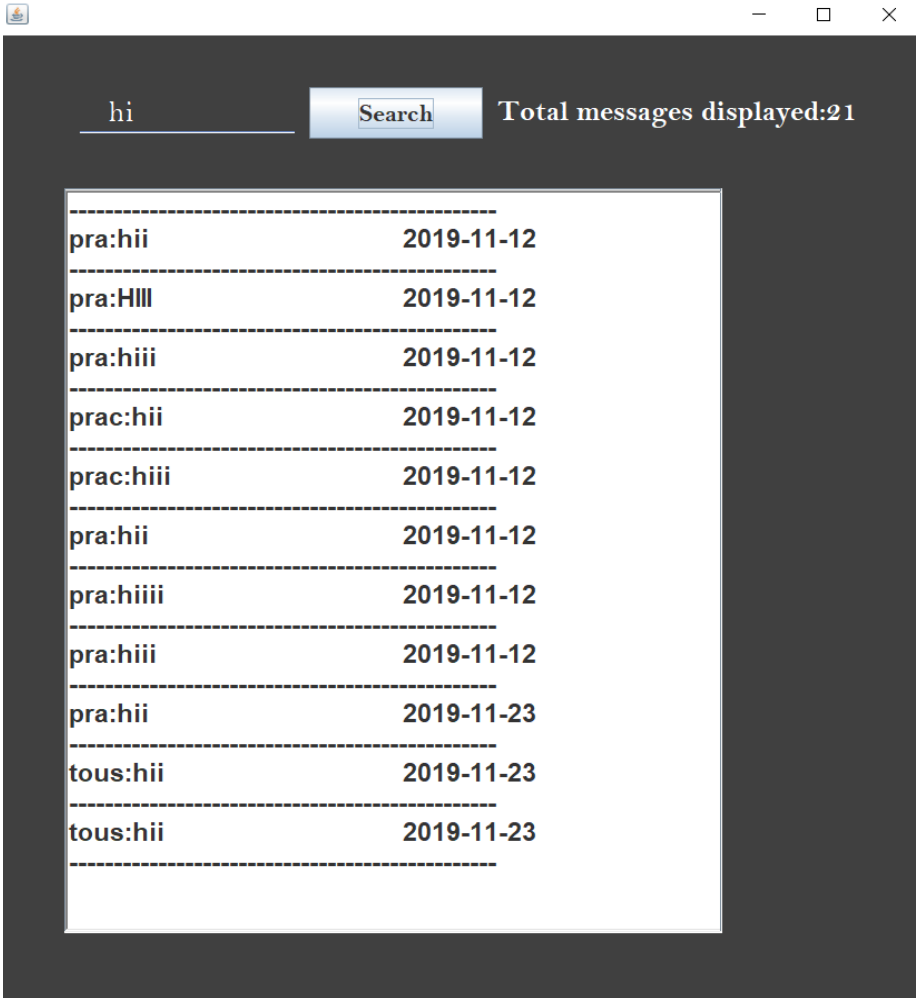


**Figure 5.5 : Screenshot of Message Search Page**

## 5.6 Resource Manager Page

This page specifically is made for resource manager where he/she will select their employees for the projects assigned. For example from the screenshot, it is clearly stated that manager Madhu works for project Softsys, If the entered data is invalid, it will be highlighted.

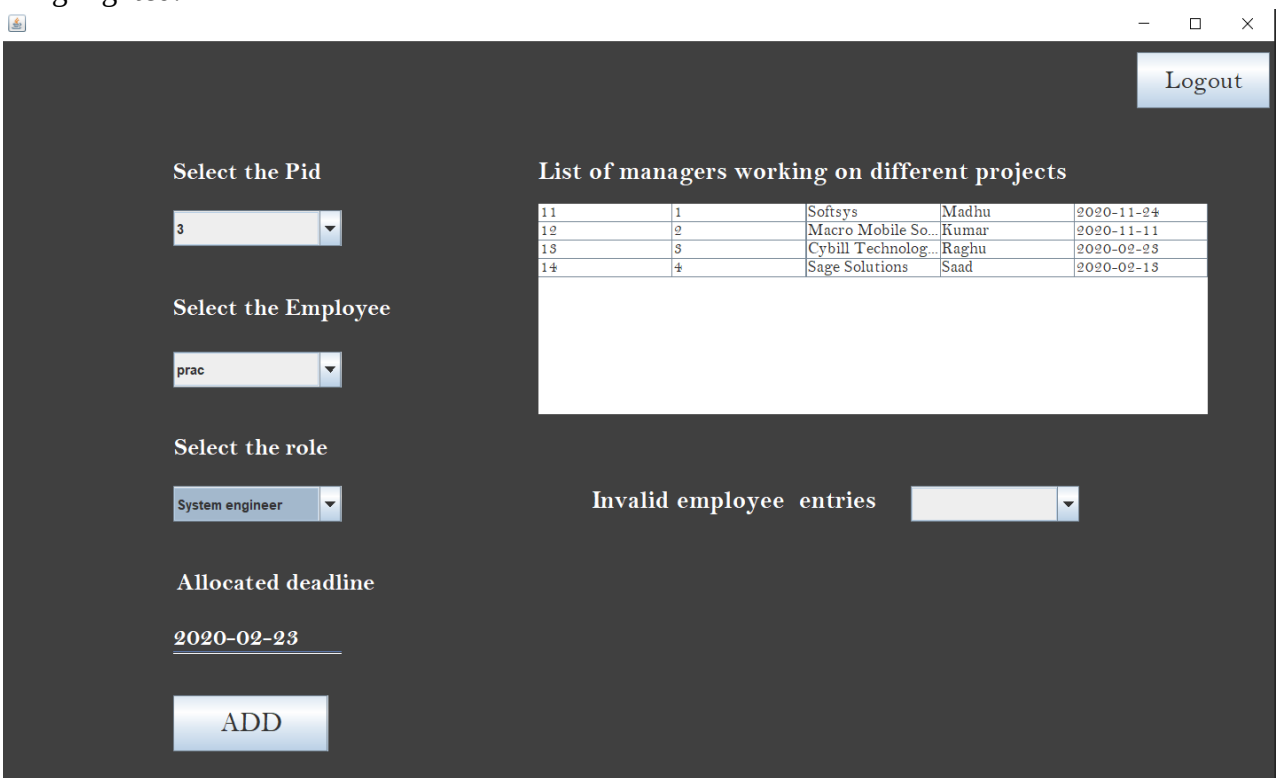


**Figure 5.6: Screenshot of Resource Manager Page**

# CONCLUSION

# CHAPTER 6

# CONCLUSION

This application as a whole makes sure that the employees have a user friendly environment in order to have an effective conversation via texts with the other employees. The concept of client/server comes into play here, as concerned with the network connection. This also includes the connection made between the SQL queries and the java code by the use of several connection objects. The framework is such that it gives a clear view to the employee about the flow of how the applications runs. And also it has many features such as a message search panel which also displays number of texts for the desired search, settings to do any kind of changes to the data stored, also the resource manager gets his/her own panel to select employees so as to work for the assigned projects. Basic troubleshoots have been applied in places of need. But majorly the focus is on the database which stores the relevant data of the employees and is stored in the localhost server MySQL.

## 6.1 FUTURE ENHANCEMENT:

It can be extended as web application We can provides the user with a good and efficient UI. Even the user can even send files, audio and video files including emojis'. Also the network services can be extended to audio and video conference calls and extra security protocol can be added.

# BI BLIOGRAPHY

## Textbook referred:


[1] Database systems Models, Languages, Design and Application Programming, Ramez Elmasri and Shamkant B Navathe, 7th Edition, 2017, Pearson.

[2] Database management systems, Ramakrishna, and Gehrke, 3rd Edition, 2014, McGraw Hill.

[3] Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, McGraw

Hill, 2013.

[4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.


## Websites referred:


1. https://stackoverflow.com
2. https://www.oracle.com
3. https://github.com
4. http://www.w3schools.com