ADVANCED JAVA PROGRAMMING PROJECT REPORT
BTECH IT SEVENTH SEMESTER

ALUMNI PORTAL

PROJECT REPORT

Submitted by

Siddharth Sahay — Reg no: 120911294

Suruchi Khetarpal — Reg no: 120911326

DEPARTMENT OF INFORMATION & COMMUNICATION TECHNOLOGY

MANIPAL INSTITUTE OF TECHNOLOGY

(A Constituent College of Manipal University)

MANIPAL – 576104, KARNATAKA, INDIA

November 2015

ACKNOWLEDGEMENT

At the very outset, I would like to give the first honors to the Almighty who gave me the wisdom and knowledge to complete this project. We also express our gratitude to **Ms. Anju Raveendran and Ms. Sangeetha TS,** Faculty Advisor and Project Guide for providing us with adequate facilities, ways and means by which we were able to complete this project.

ABSTRACT

An alumni association consists of the graduated/former and current students of the institution. The objective of this application is to allow old and new students of a university or college to communicate with each other. It will allow students to know about each other and their activities. Alumni will be able to log in and post experiences about their current work, location, opportunities, etc. Current students will be able to log in and view these posts in reverse chronological order, i.e. newest first. They may also do a live chat with alumni who are online, if any. There is also an option to create and manage events. The admin will have the right to block/unblock a particular user. A user when created is initially blocked. Additionally, there is a section for latest news related to the college that can be posted by the college authorities/admin.

CONTENTS

INTRODUCTION

The alumni portal is for an association of graduates and the current students of the institution organized according the year of graduation as well as the current location of the person. It is a platform for networking, reunions and institute events. The application begins with a login/register page for the user and goes to the home page which is the main timeline. The user gets various options for communicating with the alumni/current students. As well as viewing their profiles. The user can search for others by categories such as year and location. Announcements are made by the administrator and posts by the users. The users may create events for the institution and view the list of attendees. A centralized database is used which updates information in all tables.

LANGUAGE DESCRIPTION

The underlying programming language used for the portal is Java. Java Servlets have been used for web servers and hosted on Glassfish 4.0. The web page styling has been done by Cascading Style Sheets (CSS) and Java Server Pages has been used based on HTML and XML for dynamically generated web pages. Remote method Invocation (RMI) has been implemented for invoking methods running on another object. The database used is MySQL and the communication protocol used is Transmission Control Protocol (TCP) written in java language.

DATABASE DESCRIPTION

MySQL has been used as the database management system. The tables along with the attributes are:-

Admin(adminID primary key)

Student(studID primary key, name, dateOfBirth, yearOfGraduation, currentCity, emailAddress, phoneNumber)

Login(studID primary key, password, role, blocked)

Inbox(messageID primary key, studID, from, message)

Feed(postID primary key, post, studID)

Announcement(annID primary key, post)

Attending(eventID primary key, studID primary key)

Event(eventID primary key, eventName, location, time, date)

PROJECT DESCRIPTION

The alumni portal has been created for networking among the former and current students in an institution. The project is based on Java language with servlets used for web servers and JSP used for the dynamic creation of web pages. A login page opens up first on running the project. Non-registered users sign up as either students or alumnus. On logging in, servlets along with JSP pages are embedded together for the basic home page of the user. The timeline of the user consists of the news feed by all students and alumnus as well as options of viewing inbox, going to profile page, searching users by name, location and year of graduation, creating events and specifying whether interested in attending it or not. The administrator makes announcements after being verified and has the option for blocking or unblocking a registered user. The inbox functionality of the project has been implemented using the transmission control protocol (TCP) in java and Remote Method Invocation (RMI) is used for generating new ports for the connection establishment, announcement id generation, and post/feed id incrementing. The database model MySQL has been used for underlying tables.

PRODUCT FUNCTIONS.

Registration: The non-registered users have the option for registering on the alumni portal by providing their basic information along with the password. All newly registered users are blocked until verified and unblocked by the administrator.

Login: The user has to login to use the alumni portal application. A user can login by providing his ID and password as specified while signing up. After logging in, the student/alumni has access to all the functionalities of the application.

Blocking/Unblocking user: Each user just after registration is blocked until the administrator unblocks him/her. The administrator has access to this functionality where he/she can block/unblock a specific user by toggling a button next to the user's name.

Making Announcements: The administrator posts announcements which are important and have to be conveyed to all the students as well as the alumni. This functionality is available only to the administrator.

View Profile: All the users: student, alumni and administrator have the functionality to view their own or someone else's profile by clicking onto their name. On clicking the button, the control flows to the profile of the user which displays a send message option along with their details and events attending.

Create and Manage Events: The users have the option of creating various institution events specifying the date, time and location. The users need to announce whether they will be attending the event or not by toggling a button next to the event description. This functionality is not available to the administrator.

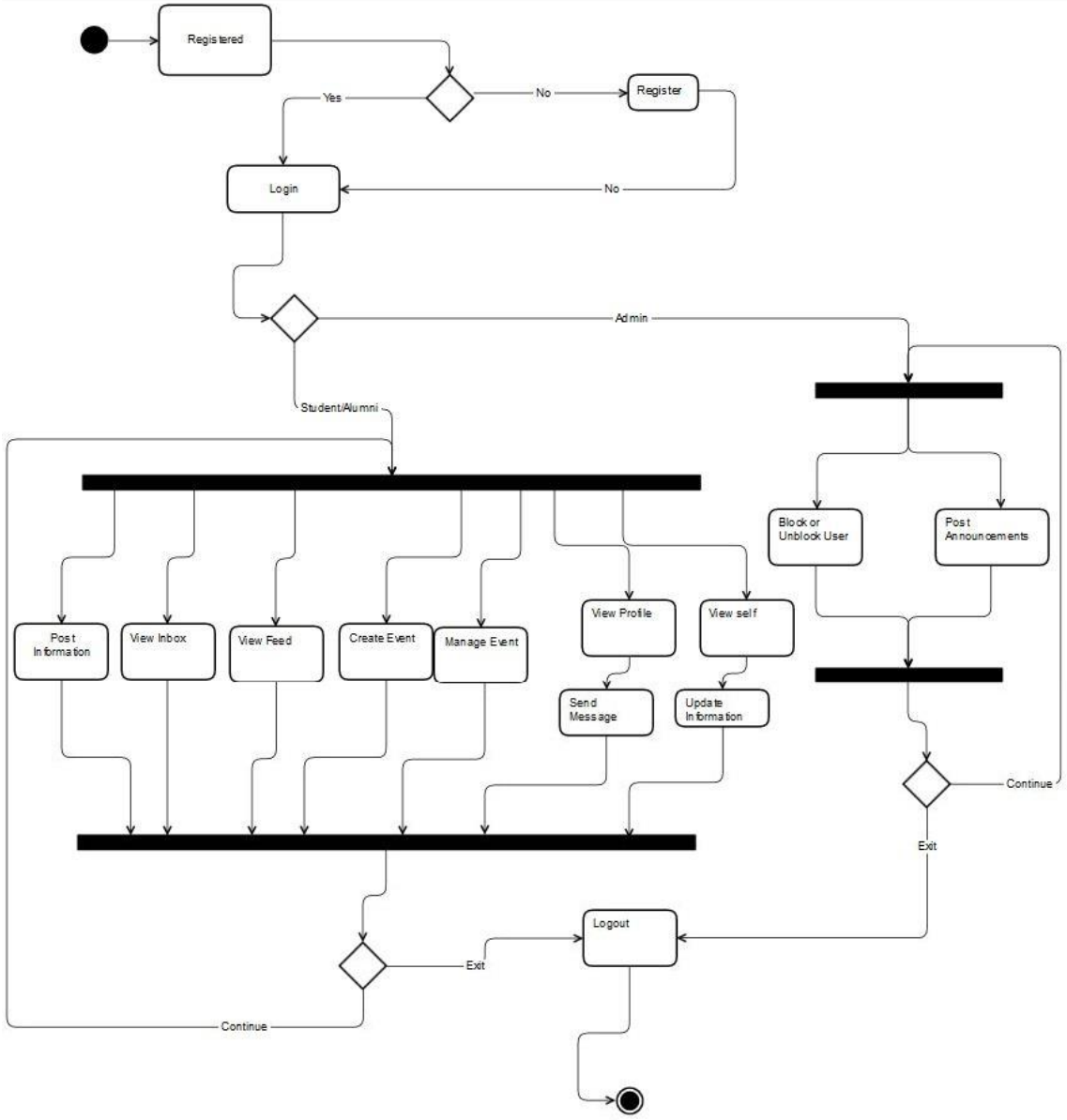
View Feed: The users can view the feed posted by other users on their home pages. The feed gets updated whenever a new one is posted by a user.

View Inbox: The inbox functionality is based on the Transmission Control Protocol(TCP) where communication is connection oriented and a user can view the messages sent by others to them. New messages are posted on top.

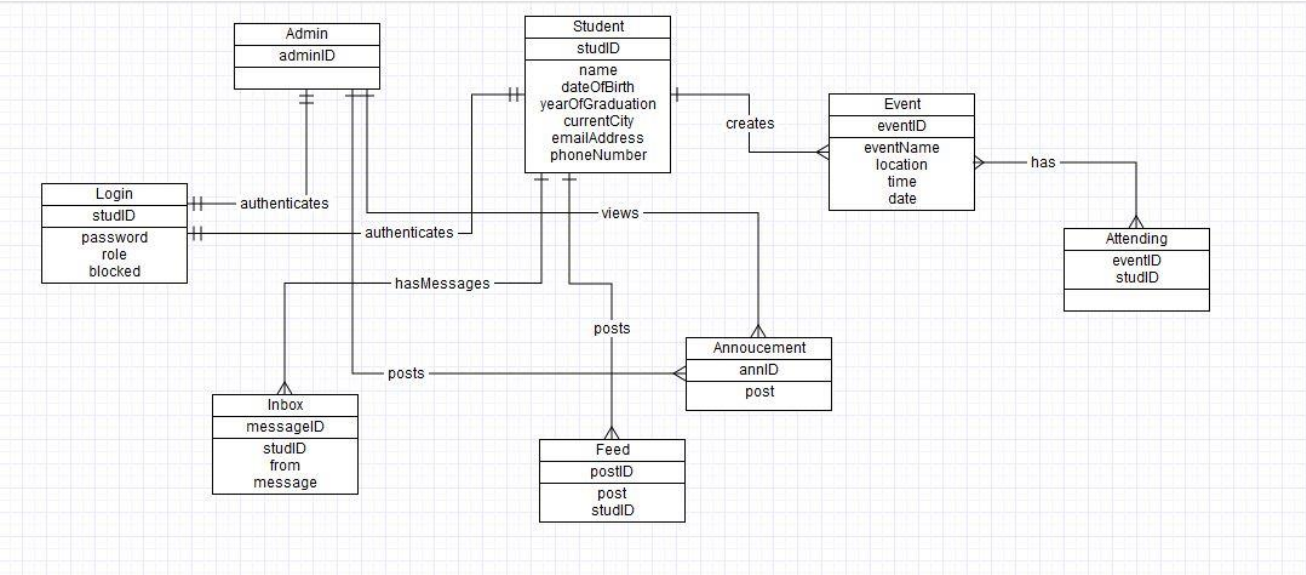
Post Feed: The users can post whatever they wish to which gets stored in the database and displayed to all other users.
Send Message: The users as well as administrator can send messages to each other via TCP. This option is displayed on a user's timeline/home page.
Update Information: The updating functionality is provided to the users while viewing their profile. A user cannot update another user's profile and a message is displayed on doing so. The user's updated information is displayed when the page reloads.
Logout: The user currently using the application can logout from his/her profile by clicking on the Logout button and will not be able to access any functionality unless logged in again.

DESIGN

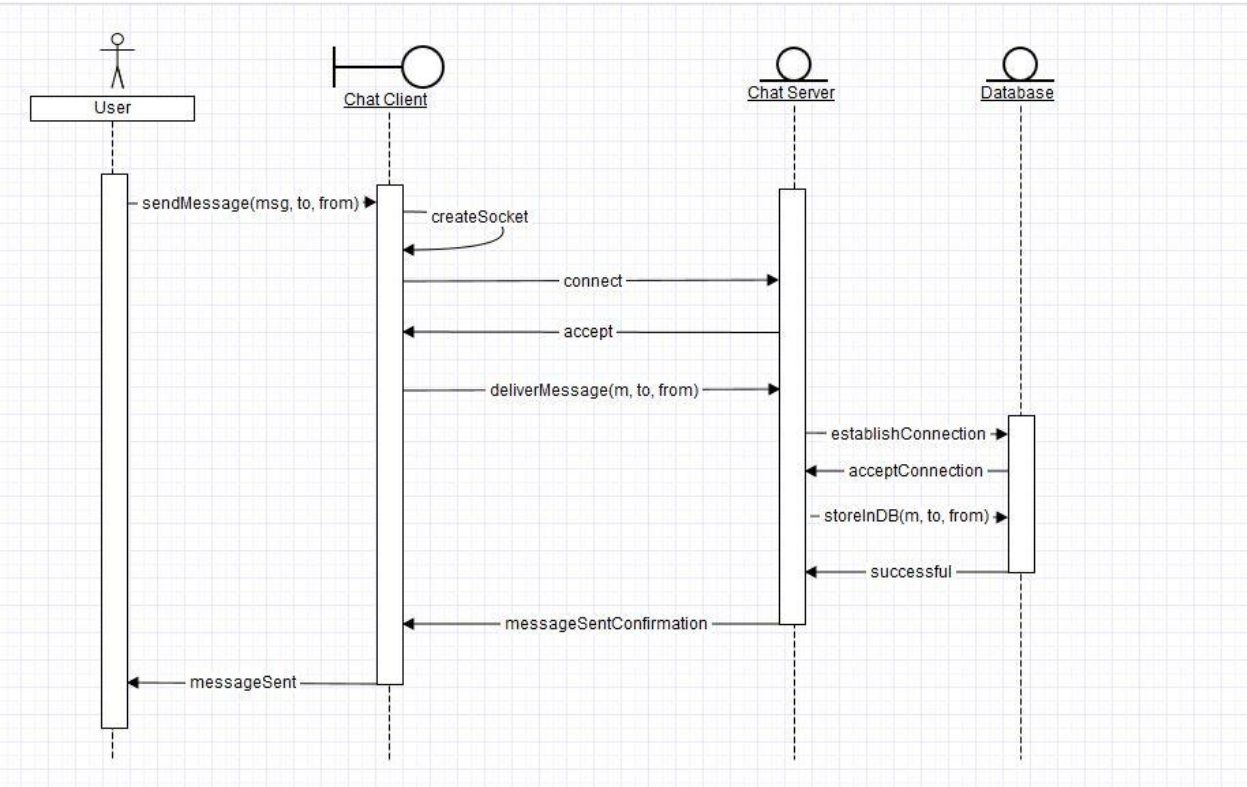
Activity Diagram Alumni Portal, v2



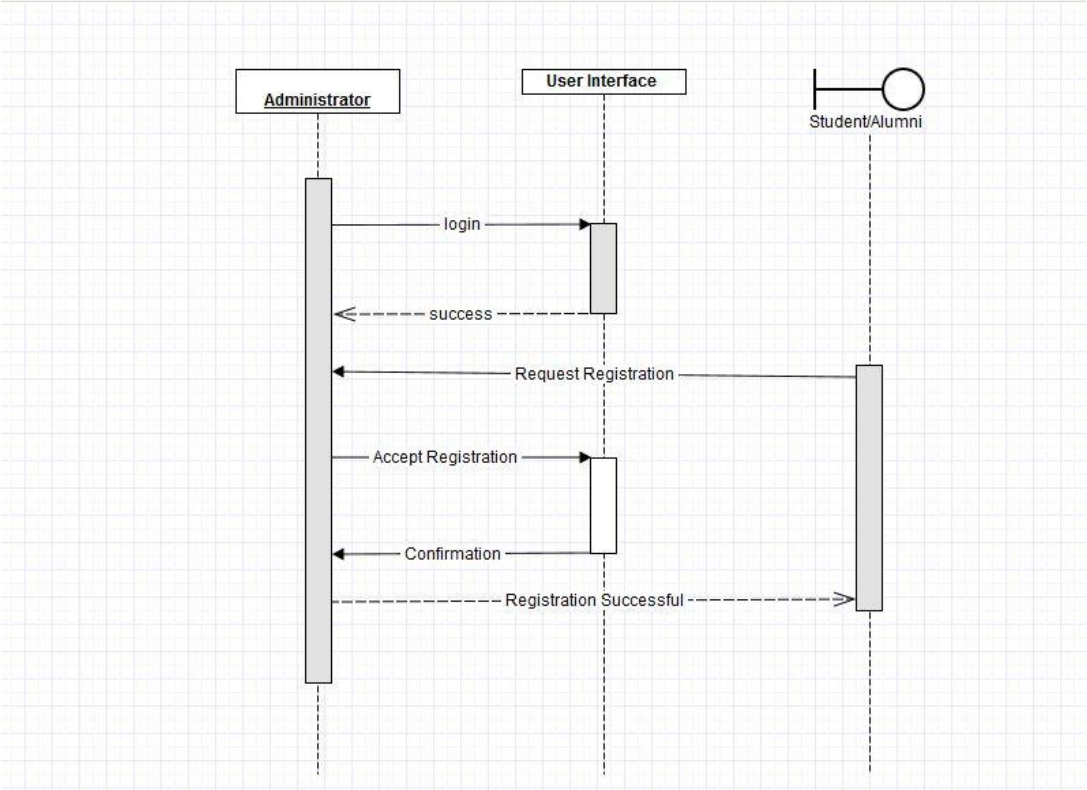
ER Diagram Alumni Portal, v2



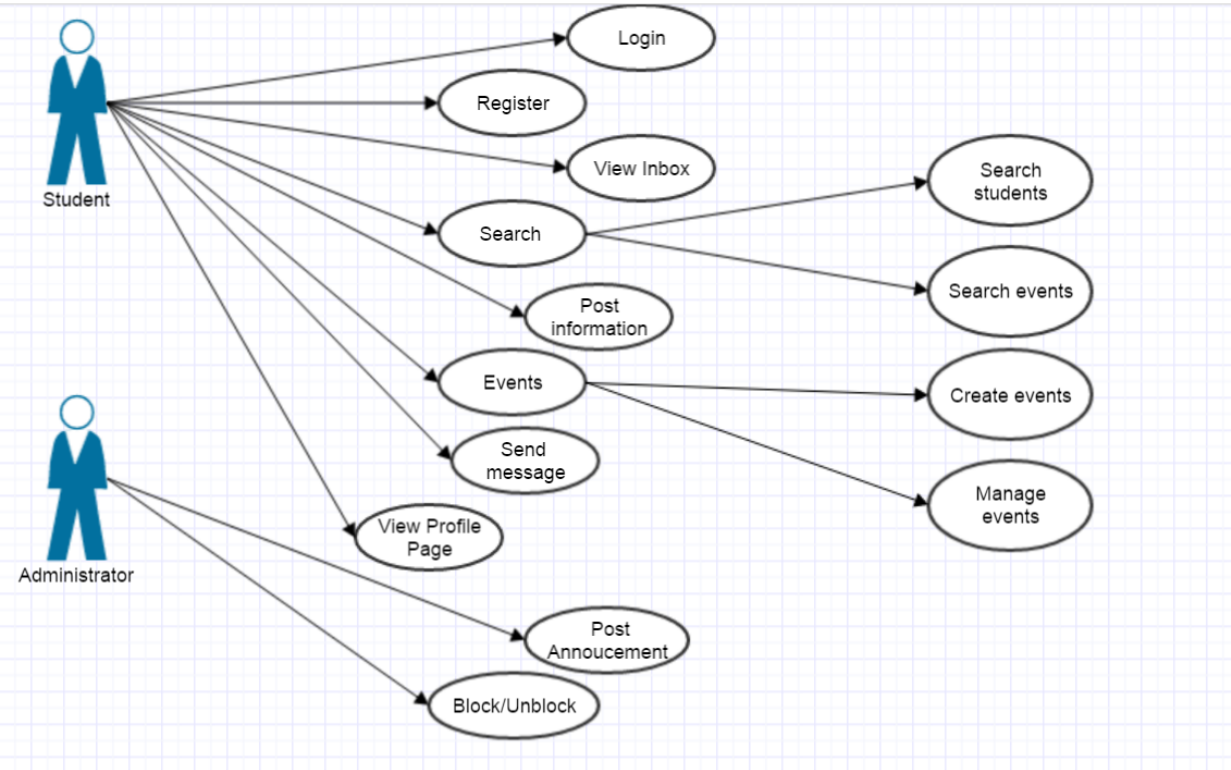
Sequence Diagram for Chat Alumni Portal, v1



***Sequence Diagram for Registration Acceptance, v2**



Use Case Alumni Portal, v1



IMPLEMENTATION

HomePage.java:-

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author student
 */
@WebServlet(urlPatterns = {"/HomePage_Student"})
public class HomePage_Student extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        doPost(request, response);

    }
```

```
    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the
left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        //processRequest(request, response);
        PrintWriter out = response.getWriter();
        response.setContentType("text/html;charset-UTF-8");

        String firstname = (String)request.getAttribute("user");
        String password = (String)request.getAttribute("password");
        System.out.println(firstname+password);
        try{
            String people = (String)request.getAttribute("people");
            request.setAttribute("people", people);
        }catch(Exception e ){
            System.out.println("Did not receive value for 'people'");
        }

        try{
            String sql="select * from login where studentid='"+firstname+"' and
password='"+password+"';";
            String dept;
            System.out.println(sql);
```

```
        //String sql="select * from student where firstname='Sid' and lastname='Sahay'";
            try{
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn =
(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/alumniportal","root","s
tudent");
                Statement stmt = conn.createStatement();
                ResultSet rs = stmt.executeQuery(sql);
                if(!rs.first()){
                   String message = "Invalid user. Please try again.";
                   request.setAttribute("message", message); // This will be available as
${message}
                   request.getRequestDispatcher("/index.jsp").forward(request, response);
                   return;
                }else{
                   if(firstname.equals("admin") && password.equals("admin")){
                     response.sendRedirect("HomePage_Admin");
                     return;
                   }
                }
                sql = "select blocked from login where studentid='"+firstname+"'";
                rs = stmt.executeQuery(sql);
                rs.first();
                if(rs.getString("blocked").equals("1")){
                   String message = "Your account has been blocked. Please try again later.";
                   request.setAttribute("message", message);
                   request.getRequestDispatcher("/index.jsp").forward(request, response);
                   return;
                }
                System.out.println("REACHED HERE");
                //TODO - retrieve news feed, annoucements, username,

                sql="select feed.post as f, student.name as n from feed inner join student on
feed.studentid=student.studentid order by postid desc;";
                rs = stmt.executeQuery(sql);
                String feed="";
                while(rs.next()){
                   feed = feed + "<h4>" +rs.getString("n") + "</h4>";
                   feed = feed + rs.getString("f") +" <br><br>";
                }
                sql = "select post from annoucement order by annid desc;";
                        rs = stmt.executeQuery(sql);
                        String ann="";
                        while(rs.next()){
                                ann = ann + "
          <bold>" +
```

```
rs.getString("post") + "</bold>
          ";
                    }
                    sql = "select name from student where studentid='"+firstname+"';";
                    rs = stmt.executeQuery(sql);
                    rs.first();
                    System.out.println(ann);


                    request.setAttribute("userid", firstname);
                    request.setAttribute("username", rs.getString("name"));
                    request.setAttribute("feed", feed);
                            request.setAttribute("annoucement", ann);

                    request.getRequestDispatcher("/homepage.jsp").forward(request, response);



                }catch(Exception e){
                    System.out.println(e);
                    //JOptionPane.showMessageDialog(this, e.getMessage());
                }
            out.println("</body></html>");
        }finally{
            out.close();
        }
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Send_Message.java:-

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
```

```
 */

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author USER
 */
@WebServlet(urlPatterns = {"/Send_Message"})
public class Send_Message extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        response.setContentType("text/html;charset-UTF-8");
        String tostudentname = (String)request.getParameter("tostudentname");
        String fromstudentid = (String)request.getParameter("fromstudentid");
        String message = (String)request.getParameter("message");



            try{
                    Class.forName("com.mysql.jdbc.Driver");
                    Connection conn =
(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/alumniportal","root","s
tudent");
```

```
                Statement stmt = conn.createStatement();
                ResultSet rs;
                System.out.println("REACHED HERE TWO");

                String sql="select studentid from student where name='"+tostudentname+"';";
                rs = stmt.executeQuery(sql);
                System.out.println(sql);
                rs.first();
                String tostudentid = rs.getString("studentid");

                TCP_Client client = new TCP_Client();

                String messageFromServer =
client.callServer(tostudentid+"###"+fromstudentid+"###"+message);


                sql="select * from student where studentid='"+tostudentid+"';";

                rs = stmt.executeQuery(sql);
                System.out.println("REACHED HERE TWO");

                rs.first();

                request.setAttribute("dob", rs.getString("dob"));
                request.setAttribute("yog", rs.getString("yearofgrad"));
                request.setAttribute("city", rs.getString("currentcity"));
                request.setAttribute("phonenum", rs.getString("phonenum"));
                request.setAttribute("email", rs.getString("emailaddress"));


                sql = "select post from annoucement order by annid desc;";
                        rs = stmt.executeQuery(sql);
                        String ann="";
                        while(rs.next()){
                                ann = ann +
"          <bold>" +
rs.getString("post") + "</bold>
           ";
                }

                request.setAttribute("annoucement", ann);

                sql = "select name from student where studentid='"+tostudentid+"';";
                rs = stmt.executeQuery(sql);
                rs.first();
                request.setAttribute("name",rs.getString("name"));
```

```
                sql=" select eventname, time, date from event where eventid in (select eventid
from attending where studentid='"+tostudentid+"')";
                rs = stmt.executeQuery(sql);
                        String events="";
                        while(rs.next()){
                                events = events + "<bold><u><font size='4' >" +
rs.getString("eventname") + "</font></u></bold>
<br>"+rs.getString("time")+"<br>"+rs.getString("date")+"<br><br>";
                }
                if(events.equals(""))events="This person is not attending any upcoming events.";
                sql="select role from login where studentid='"+tostudentid+"';";
                rs = stmt.executeQuery(sql);
                rs.first();
                String t2="Now an ", t = rs.getString("role");

                if(t.equals("student"))t2="Currently a ";


                request.setAttribute("student",
t2+(""+(rs.getString("role").charAt(0))).toUpperCase()+rs.getString("role").substring(1));
                request.setAttribute("message", messageFromServer);
                request.setAttribute("events", events);
                request.setAttribute("fromstudentid", fromstudentid);
                request.setAttribute("studentid", tostudentid);
                request.getRequestDispatcher("/profilepage.jsp").forward(request, response);


            }catch(Exception e){
                System.out.println(e);
                //JOptionPane.showMessageDialog(this, e.getMessage());
            }



    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the
left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
```

```
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

TCP_Client.java:-

```
import java.io.*;
import java.net.*;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;

class MyRunnable implements Runnable {

    private int port;
    public MyRunnable(int port) {
        this.port = port;
```

```
    }

    @Override
    public void run() {
        TCP_Server server = new TCP_Server();
        try {
            server.makeServer(port);
        } catch (Exception ex) {
            Logger.getLogger(MyRunnable.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
}

public class TCP_Client{

public String callServer(String message)throws Exception{


    RMI_Client client = new RMI_Client();
    int port = client.callServerRMI2();
    System.out.println("THIS IS THE NEW PORT: "+port);

    MyRunnable myRunnable = new MyRunnable(port);
    Thread t = new Thread(myRunnable);
    t.start();


    Socket s=new Socket("localhost",port);
    DataInputStream din=new DataInputStream(s.getInputStream());
    DataOutputStream dout=new DataOutputStream(s.getOutputStream());

    String str2="";
    dout.writeUTF(message);
    dout.flush();
    str2=din.readUTF();
    System.out.println("Server says: "+str2);

    dout.close();
    s.close();

    if(str2.equals("success")){
        return "Your message has been sent.";
    }else{
        return "Your message was not sent. Try again later.";
    }
```

```
}
}
```

TCP_Server.java:-

```
import java.io.*;
import java.net.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Timer;

class TCP_Server{

public void makeServer(int port)throws Exception{

   System.out.println("Running Server, before ServerSocket");

   ServerSocket ss=new ServerSocket(port);
   Socket s=ss.accept();

   System.out.println("Running the TCP_Server");

   DataInputStream din=new DataInputStream(s.getInputStream());
   DataOutputStream dout=new DataOutputStream(s.getOutputStream());


   String messageold="",sendToClient="";
   String[] message = new String[3];

   messageold=din.readUTF();
   message = messageold.split("###");
   for(int i=0;i<3;i++)System.out.print(message[i]);

   //add message to database
   RMI_Client client = new RMI_Client();


   try{
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn =
(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/alumniportal","root","s
tudent");
                Statement stmt = conn.createStatement();
                ResultSet rs;
```

```
                String sql = "select max(messageid) as max from inbox;";
                System.out.println(sql);
                rs = stmt.executeQuery(sql);
                rs.first();
                sql = "insert into inbox values('"+client.callServerRMI(rs.getString("max"))+"',
'"+message[0]+"', '"+message[1]+"', '"+message[2]+"');";

                stmt.executeUpdate(sql);

                sendToClient = "success";



            }catch(Exception e){
                System.out.println(e);
                 sendToClient = "failure";
            }finally{
                 dout.writeUTF(sendToClient);
                dout.flush();
            }

                din.close();
                s.close();
                ss.close();

}
}
```

RMI_ServerIntf.java:-

```
import java.rmi.*;

public interface RMI_ServerIntf extends Remote
{
   String increment(String val) throws RemoteException;
   int newPort() throws RemoteException;
}
```

RMI_ServerImpl.java:-

```
import java.rmi.*;
import java.rmi.server.*;
```

```
public class RMI_ServerImpl extends UnicastRemoteObject implements RMI_ServerIntf{
   public RMI_ServerImpl() throws RemoteException{

   }

   public String increment(String val) throws RemoteException{
      int old = Integer.parseInt(val);
      return ""+(old+1);
   }
   public int newPort() throws RemoteException{
      int randomNumber = ( int )( Math.random() * 9999 );

      if( randomNumber <= 3000 ) {
         randomNumber = randomNumber + 1000;
      }
      System.out.println(randomNumber);
      return randomNumber;


   }
}
```

RMI_Server.java:-

```
import java.net.*;

import java.rmi.*;


public class RMI_Server{

   public static void main(String args[]){

      try{

         RMI_ServerImpl addServerImpl = new RMI_ServerImpl();
         Naming.rebind("RMI_Server", addServerImpl);

      }catch(Exception e){

         e.printStackTrace();

      }
```

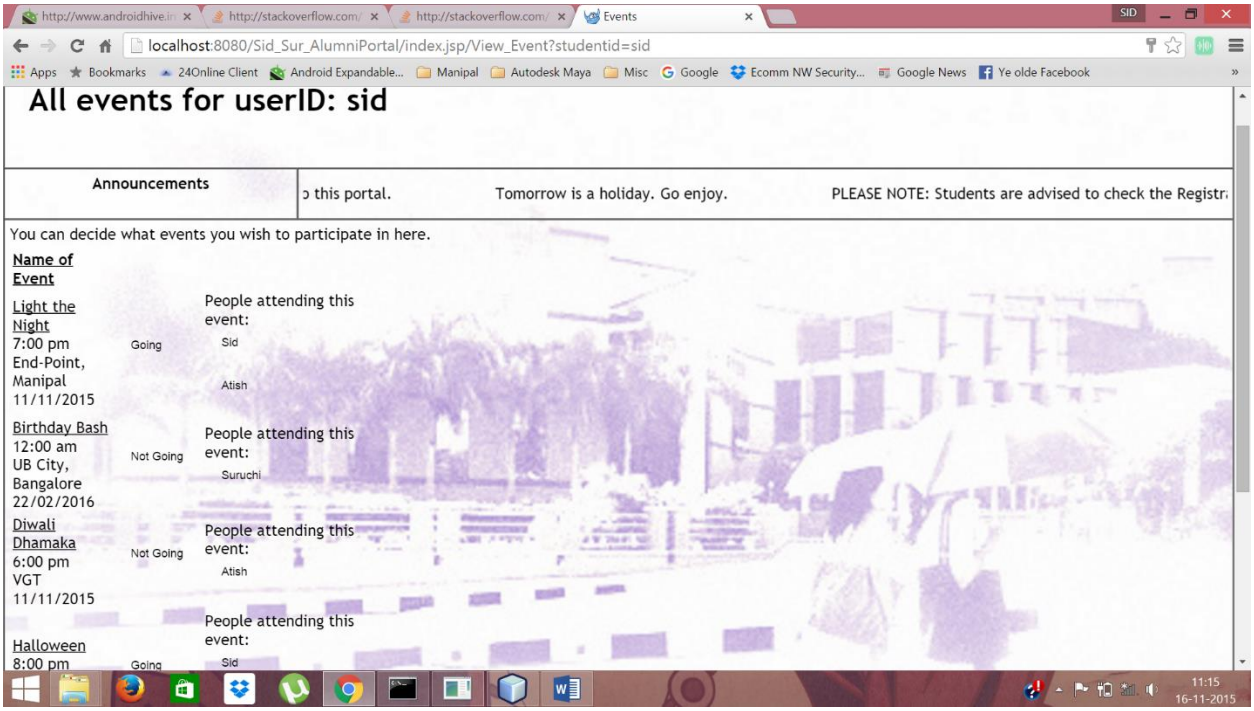
```
}
}
```

RMI_Client.java:-

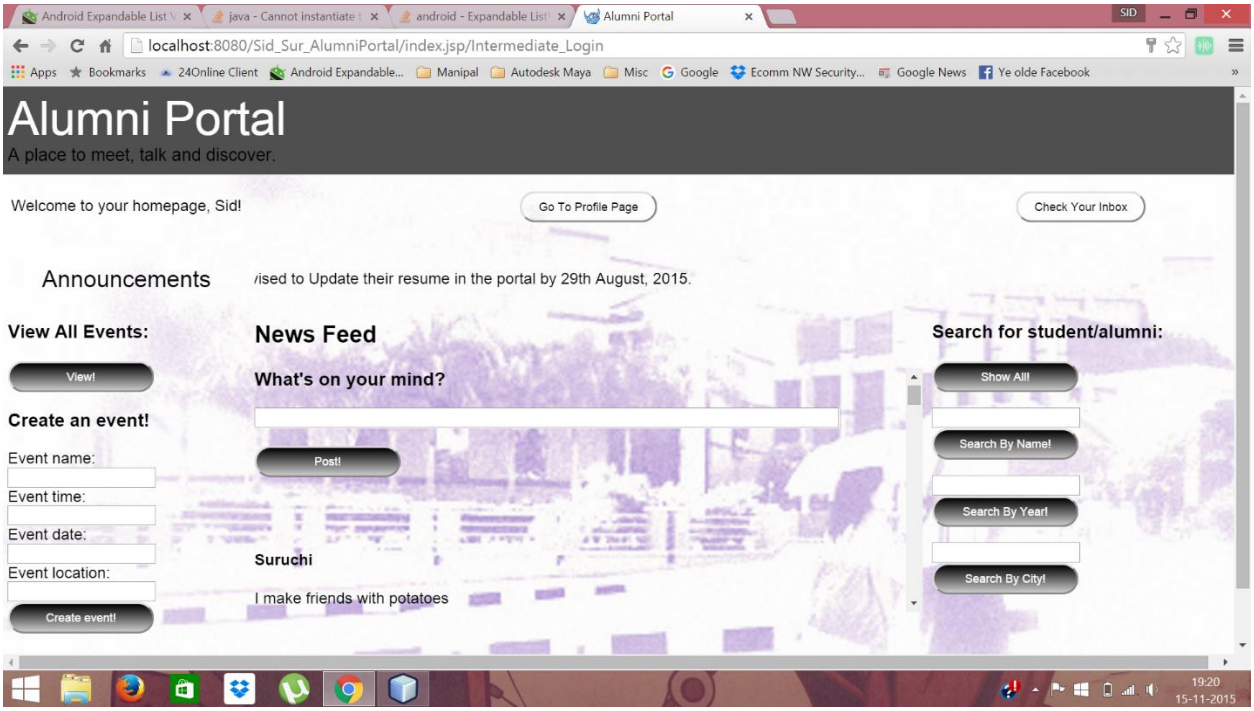
```
import java.awt.Desktop;
import java.io.File;
import java.rmi.*;

public class RMI_Client{
   public int callServerRMI2(){
      try{
      //Runtime.getRuntime().exec("cmd /c ./RMI_Commands.bat", null, new File("./"));
         //Desktop.getDesktop().open(new File("C:\\Users\\USER\\Documents\\Sid\\College Stuff\\Sem 7 IT\\Advanced Java Programming - AJP Lab\\Sid_Sur_AlumniPortal\\src\\java\\RMI_Commands.bat"));

         String addServerURL = "rmi://localhost/RMI_Server";
         RMI_ServerIntf addServerIntf = (RMI_ServerIntf)Naming.lookup(addServerURL);

         return addServerIntf.newPort();

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

```
    return 9999;
  }
  public String callServerRMI(String value){
    try{
      //Runtime.getRuntime().exec(new String[] { "cmd.exe", "/c", "./RMI_Commands.bat" }
);
      //Runtime.getRuntime().exec("cmd.exe", "/c", "./RMI_Commands.bat");
      //File file = new File("./RMI_Commands.bat");
      //Desktop.getDesktop().open(file);
      //Runtime.getRuntime().exec("cmd /c .\\RMI_Commands.bat");
      //Runtime.getRuntime().exec("cmd /c ./RMI_Commands.bat", null, new File("./"));
      //Runtime.getRuntime().exec(new String[] {"cmd.exe", "/C", "./RMI_Commands.bat"});
      //Desktop.getDesktop().open(new File("C:\\Users\\USER\\Documents\\Sid\\College
Stuff\\Sem 7 IT\\Advanced Java Programming - AJP
Lab\\Sid_Sur_AlumniPortal\\src\\java\\RMI_Commands.bat"));

      String addServerURL = "rmi://localhost/RMI_Server";
      RMI_ServerIntf addServerIntf = (RMI_ServerIntf)Naming.lookup(addServerURL);

      return addServerIntf.increment(value);

    }catch(Exception e){
      e.printStackTrace();
    }
    return null;
  }}
```

EXPERIMENTAL RESULTS



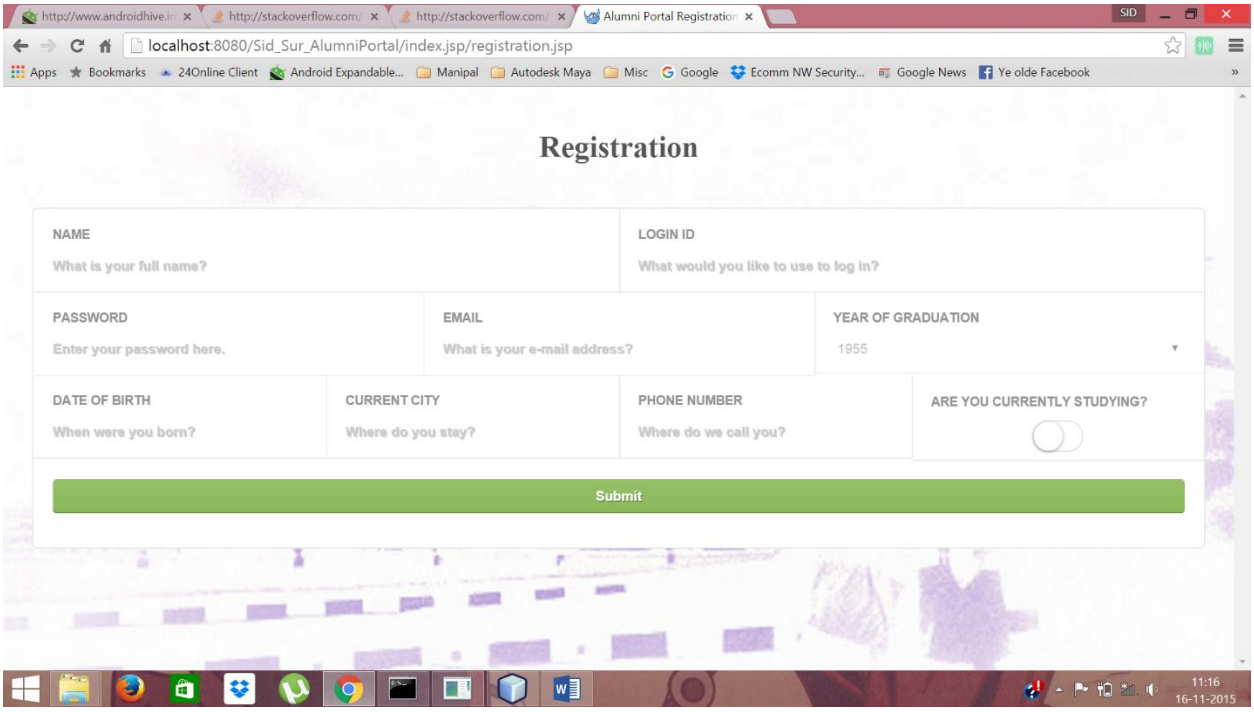
Events page for a user. The user can choose which events he wants to go for. Also, he can view the profile page of other people attending particular events.



This is what the user sees when he logs in. The current news feed is shown, the user can post his own ‘status’ to add it to the feed. User can search for other students/alumnus on the portal. User can also create events and as view existing events.

On this page, the user can view messages that have been sent to him.

This is the profile page of a user. It is a publicly accessible page, i.e. anyone can view anyone's profile page.



This is the registration page, where a user can register. After a user registers, the administrator confirms that the user has been a part of the organization and then unblocks his account, making it accessible by the user.

CONCLUSION

The Alumni Portal has been created for communication between alumni as well as the current students in an institution. It can help connect people from an organization better. This portal can be used to search for people, and even maintain connections with them. This is a form of social media that can be implemented in a college.

FUTURE WORK

The application can be deployed on the cloud for remote usage. It can even be deployed on a local intranet of an organization to be accessed from within itself.

Currently, we are using Servlets and JSPs to pass information from one page to another, to upgrade this form of data transfer, we can use JSON files for more sophisticated data passing.