



20091909 박혜빈

20091930 전소라

20115579 김현구

목 차

1. 개 요

2. 배경 및 목적

3. 구현 과정

4. 결 과

개요

➡ JAVA와 데이터베이스 연동을 이용한 나만의 공간 제공 프로그램

- 다이어리 : 날짜 별 간단메모를 저장하는 기능
- 주소록 : 지인의 연락처 및 주소 등을 저장, 검색하는 기능
- 사진첩 : 개인의 사진을 저장하고 볼 수 있는 기능
- 채팅 : 같은 서버 접속 시 친구와 채팅을 할 수 있는 기능
- 쪽지 : 친구에게 쪽지를 보낼 수 있는 기능

배경 및 목적

SNS 시대에 맞춰 커뮤니케이션이 가능하고, 개인의 공간으로도
활용할 수 있는 프로그램

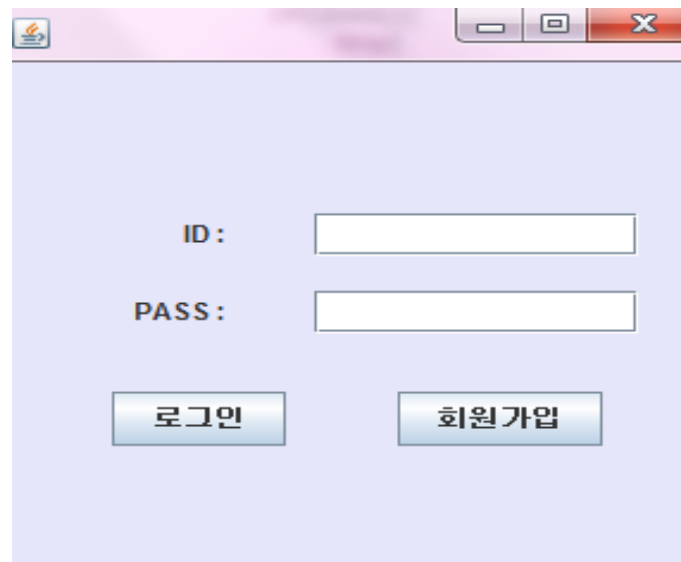
다이어리, 주소록, 사진첩, 채팅, 쪽지 기능을 제공하는 프로그램



구현과정

로그인

```
btnNewButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        DBConnection dc = new DBConnection();  
        int i=dc.confirm(textField.getText(),textField_1.getText());  
        if(i==1){  
            try{  
                InetAddress ip = InetAddress.getLocalHost();  
                con=dc.getConnection();  
                String query="insert into send values(?,?)";  
                pstmt=con.prepareStatement(query);  
                pstmt.setString(1,textField.getText());  
                pstmt.setString(2,ip.getHostAddress());  
                pstmt.executeUpdate();  
                new TabbedPaneEx(textField.getText(),new JJJcalendar(), new Address(),new ChatClient(textField.getText())  
                ,new DatagramClient(textField.getText()),new ButtonAndImageLabelFrame(textField.getText()));  
                JJJcalendar jc= new JJJcalendar();  
                jc.receive_Id(textField.getText());  
                log.this.dispose();  
            }catch(Exception ea){  
                ea.printStackTrace();  
            }  
        }else if(i==2){  
            JOptionPane.showMessageDialog(log.this,"아호나 패스워드가 틀렸습니다..");  
        }else{  
            JOptionPane.showMessageDialog(log.this,"아호나 패스워드가 틀렸습니다..");  
        }  
    }  
});
```



회원가입

```
int i=dc.confirm2(textField_1.getText());
if(i==1){
    try{
        con=dc.getConnection();
        String query="insert into member values (?,?,?,?,?)";
        pstmt=con.prepareStatement(query);
        pstmt.setString(1,textField.getText());
        pstmt.setString(2,textField_1.getText());
        pstmt.setString(3,textField_2.getText());
        pstmt.setString(4,textField_3.getText());
        pstmt.setString(5,textField_4.getText());
        pstmt.executeUpdate();
        userid=textField_1.getText();
        JOptionPane.showMessageDialog(member.this,"가입을 축하합니다");
        String query2="create table ";
        query2+= textField_1.getText();
        query2+=" (text varchar2(2000),year varchar2(10),month varchar2(10),day varchar2(10))";
        pstmt=con.prepareStatement(query2);
        pstmt.executeUpdate();
        con=dc.getConnection();
        String query3="create table ";
        query3+= textField_1.getText();
        query3+="pic";
        query3+=" (filename varchar2(100))";
        pstmt=con.prepareStatement(query3);
        pstmt.executeUpdate();
        member.this.dispose();
    }
}
```

이름

아이디

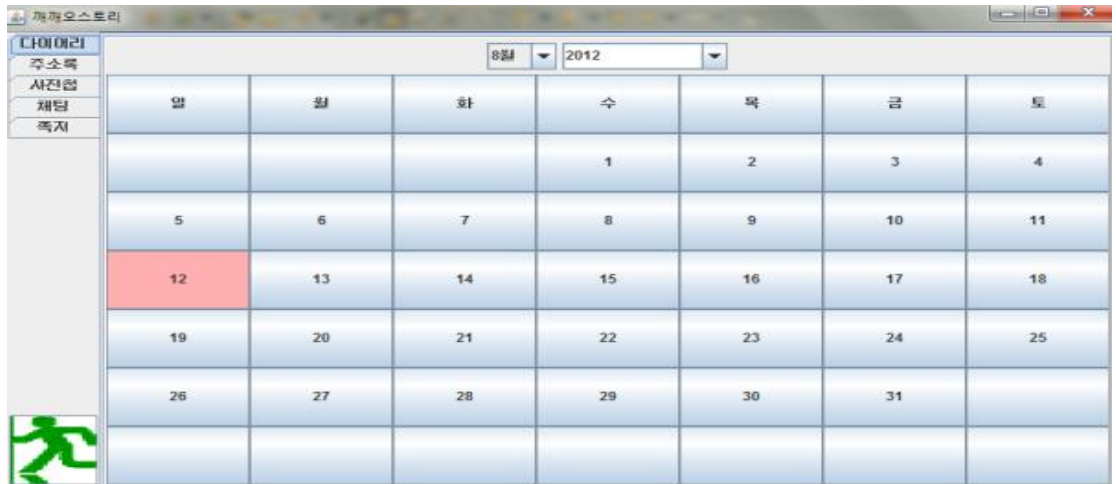
비밀번호

주소

전화번호

메인화면

```
JJJcalendar jjc;  
Address addr;  
ChatClient chat;  
DatagramClient mess;  
ButtonAndImageLabelFrame pic;
```



```
TabbedPaneEx(JJJcalendar jjc, Address addr, ChatClient chat, DatagramClient mess, ButtonAndImageLabelFrame pic) {  
    this.jjc=jjc;  
    this.addr=addr;  
    this.chat=chat;  
    this.mess=mess;  
    this.pic=pic;
```

```
JTabbedPane createTabbedPane() {
```

```
    JTabbedPane pane = new JTabbedPane(JTabbedPane.LEFT); // 왼쪽 객체를 생성한다.  
    pane.addTab("다이어리", jjc); // 첫번째 탭  
    pane.addTab("주소록", addr); // 세번째 탭  
    pane.addTab("사진첩", pic);  
    pane.addTab("채팅", chat);  
    pane.addTab("쪽지", mess);
```

```
    return pane;
```

```
}
```


메인화면

```
add(Button2); //로그아웃버튼
Button2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        TabbedPaneEx.this.dispose();
        try{
            DBConnection dc = new DBConnection();
            con=dc.getConnection();
            String query="delete from send where userid=?";
            pstmt=con.prepareStatement(query);
            pstmt.setString(1,TabbedPaneEx.this.id);
            pstmt.executeUpdate();
        }catch(Exception e2){
            e2.printStackTrace();
        }

        new log();
    }
});
```

다이어리

```
year = yearChoice.getSelectedItem().toString();
month = monthChoice.getSelectedItem().toString().replace("월", "");
day = e.getActionCommand();
```

```
try{
    DBConnection dc = new DBConnection();
    con=dc.getConnection();
    String query="select text from ";
    query+=id;
    query+= " where year=? and month =? and day=?";
    pstmt=con.prepareStatement(query);
    pstmt.setString(1,year);
    pstmt.setString(2,month);
    pstmt.setString(3,day);
    rs=pstmt.executeQuery();
    while(rs.next()){
        textPane.setText(rs.getString(1));
    }
}
```

```
}catch(Exception ea){
    ea.printStackTrace();
}
b_save.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try{
            DBConnection dc = new DBConnection();
            con=dc.getConnection();
            System.out.println(id);
            String query="insert into ";
            query+= id;
            query+=" values (?, ?, ?, ?)";
            pstmt=con.prepareStatement(query);
            pstmt.setString(1,textPane.getText());
            pstmt.setString(2,year);
            pstmt.setString(3,month);
            pstmt.setString(4,day);
            pstmt.executeUpdate();
        }
```

다이어리

주소록

사진첩

채팅

쪽지

8월 2012

일	월	화	수	목	금	토
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

주소록

```
public void actionPerformed(ActionEvent e) {
    ResultSet rs = null;
    Component c = (Component) e.getSource();
    try {
        if(c == add) {
            if(cmd != ADD)
                setEnable(ADD);
            else {
                String vname = name.getText().trim();
                String vuserid = userid.getText().trim();
                if(vname == null || vname == null ||
                    vname.length() == 0 || vname.length() == 0)
                    return ;
                String vtel = tel.getText().trim();
                String vaddress = address.getText().trim();
                String vbirth = birth.getText().trim();
                String sql = "insert into people values(?,?,?,?,?)";
                PreparedStatement stat=conn.prepareStatement(sql);
                stat.setString(1, vname);
                stat.setString(2, vuserid);
                stat.setString(3, vtel);
                stat.setString(4, vaddress);
                stat.setString(5, vbirth);
                stat.executeUpdate();
                setEnable(NONE);
                clear();
                cmd = NONE;
                setEditable(false);
            }
        }
    }
```

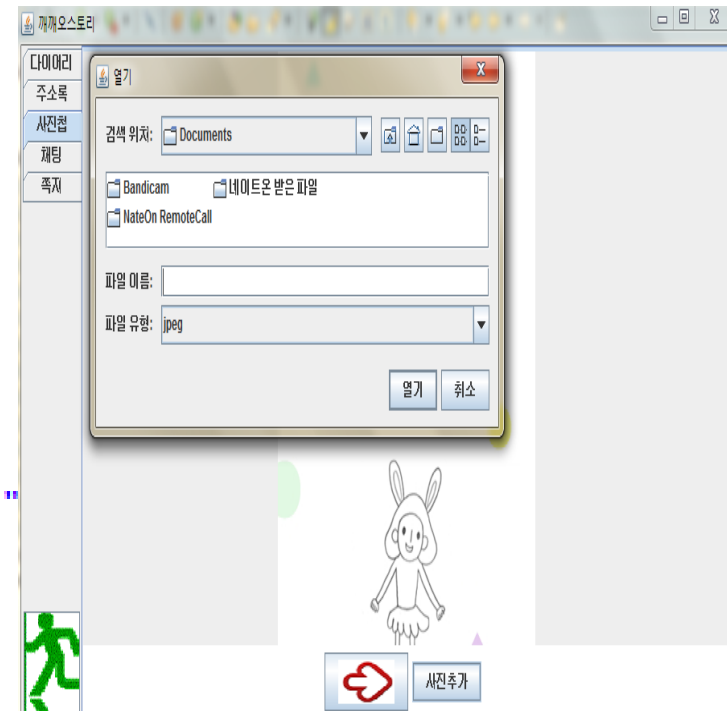
개요오스토리

주소록	이름	USERID	전화번호	주소	생년월일
사진첨	이름	ID	전화번호	주소	생년월일
재입					
확인	김방구	hero	01022223333	대전유성구	동안
	박해봉	zzz	01011117777	대전	940213
	전골	jsr	01012345678	익산	870517

추가 삭제 검색 전체 취소

사진첩

```
public ButtonAndImageLabelFrame(String id) {  
    this.setLayout(new BorderLayout());  
    this.id = id;  
    DBConnection dc = new DBConnection();  
    con=dc.getConnection();  
    System.out.println(id);  
    try {  
        String query="select filename from "  
        query+=id;  
        query+="pic";  
  
        System.out.println(id);  
        pstmt=con.prepareStatement(query);  
        rs=pstmt.executeQuery();  
        int i=0;  
        while(rs.next()){  
            images[i] = new ImageIcon("d:/"+rs.getString(1));  
            i++;  
        }  
  
    } catch (SQLException e2) {  
        e2.printStackTrace();  
    }  
}
```



사진첩

```
jbt_open.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        if(arg0.getSource() == jbt_open){  
            if(jfc.showOpenDialog(MenuPanel.this) == JFileChooser.APPROVE_OPTION){  
                jlb.setText("선택한 파일 : " + jfc.getSelectedFile().toString());  
                int data = 0;  
                DBConnection dc = new DBConnection();  
                try {  
                    in = new FileInputStream(new File(jfc.getSelectedFile().toString()));  
                    File file2= jfc.getSelectedFile();  
                    String dbname = file2.getName();  
                    out = new FileOutputStream("d://" + file2.getName(), true);  
                    while((i=in.read()) != -1) {  
                        out.write(i);  
                        len++;  
                    }  
                    in.close();  
                    out.close();  
                    System.out.println(len + " bytes are copied...Successfully!");  
                    con=dc.getConnection();  
                    String query="insert into ";  
                    query+=id;  
                    query+="pic ";  
                    query+="values(?)";  
                    System.out.println(id);  
                    pstmt=con.prepareStatement(query);  
                    pstmt.setString(1,file2.getName());  
                    pstmt.executeUpdate();  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        }  
    }  
});
```

사진첩

```
        try {

            String query="select filename from ";
            query+=id;
            query+="pic";
            System.out.println(id);
            pstmt=con.prepareStatement(query);
            rs=pstmt.executeQuery();
            int i=0;
            while(rs.next()){
                images[i] = new ImageIcon("d:/" + rs.getString(1));
                i++;
            }

        } catch (SQLException e2) {
            e2.printStackTrace();
        }

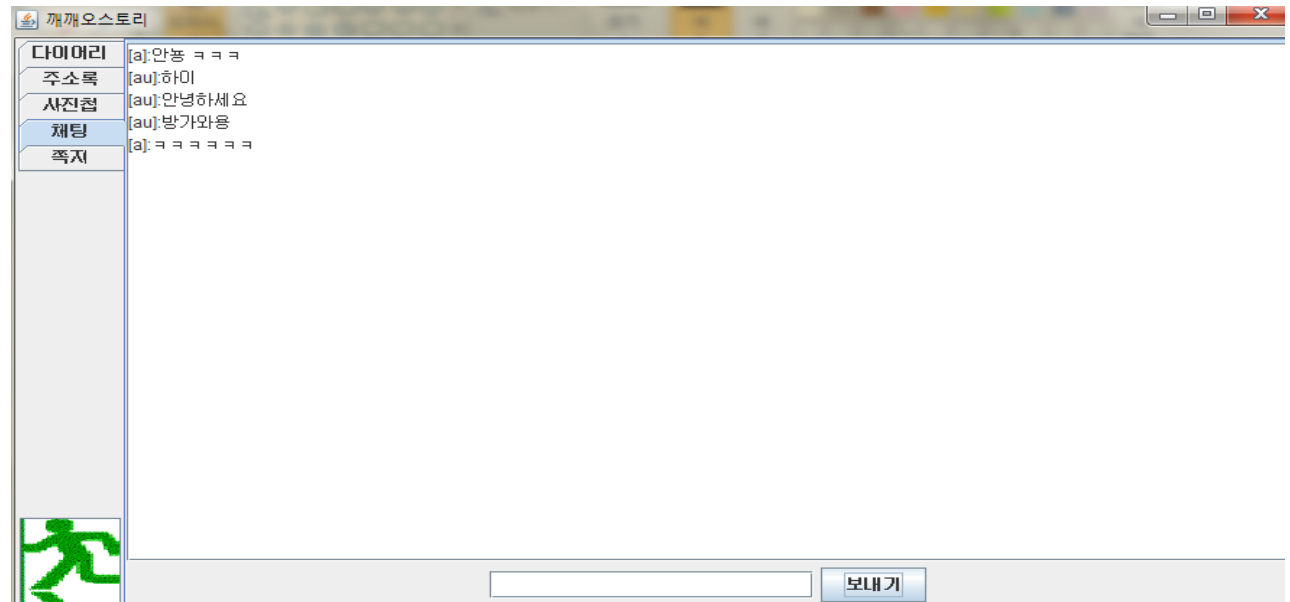
    }

});

rightBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(images[currentId]==null)
            currentId=0;
        imageLabel.setIcon(images[currentId]); // 이미지 레이블에 이미지 변경
        currentId ++;
    }
});
```

채팅

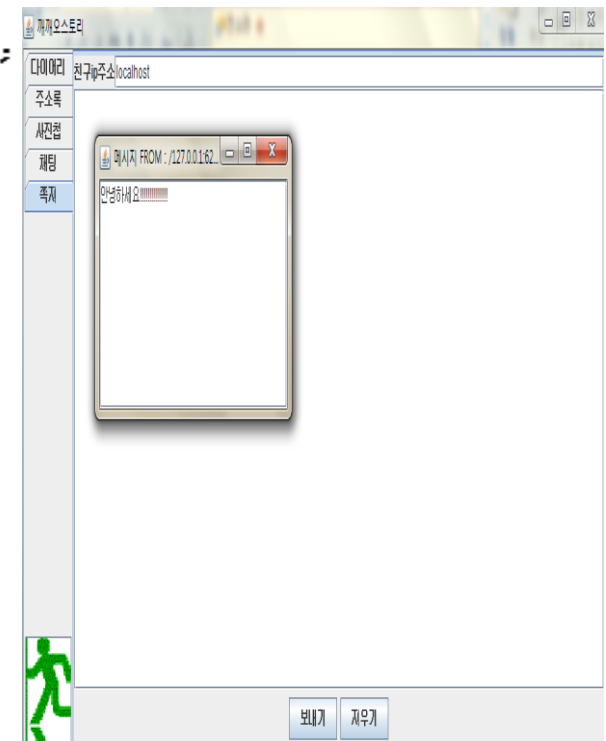
```
try{
    new TabbedPaneEx(new JJJcalendar(), new Address(),new ChatClient(textField.getText()),new DatagramClient(),
        new ButtonAndImageLabelFrame(textField.getText()));
    JJJcalendar jc= new JJJcalendar();
    jc.receive_Id(textField.getText());
    log.this.dispose();
}catch(Exception ea){
    ea.printStackTrace();
}
```



꼭 지

```
public void actionPerformed(ActionEvent e) {
    Object b = e.getSource();
    if(b == send) {
        String host = server.getText();
        if(host == null || host.trim().length() == 0)
            return;
        DBConnection dc = new DBConnection();
        try{
            con=dc.getConnection();
            String query="select ipad from send ";
            query+= "where userid=?";
            pstmt=con.prepareStatement(query);
            pstmt.setString(1,server.getText());
            rs=pstmt.executeQuery();

            if(rs.next()){
                rs.getString(1);
            }
        }catch(Exception ea){
            ea.printStackTrace();
        }
        String msg = input.getText();
        try {
            sendMessage(rs.getString(1), msg);
            input.setText("");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    } else if(b == clear) {
        input.setText("");
    }
}
```



꼭 지

```
public class DatagramServer {
    public static void main(String[] args) {
        DatagramSocket socket;
        byte[] buf = new byte[2048];
        DatagramPacket packet;
        InetAddress address;
        int port;
        try {
            socket = new DatagramSocket(2000);
            System.out.println("DatagramServer is ready.");

            while (true) {
                packet = new DatagramPacket(buf, buf.length);
                socket.receive(packet);
                int len = packet.getLength();
                address = packet.getAddress();
                port = packet.getPort();
                new MessageWindow(new String(buf, 0, len),
                                   address.toString() + ":" + port);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

결과

www.Bandicam.co.kr

이름

아이디 중복확인

비밀번호

주소

전화번호

회원가입

회원가입

www.Bandicam.co.kr

8월 2012

일	월	화	수	목	금	토
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

구현