



ID

비밀번호

ID찾기

로그인

회원가입

퇴실



회원가입

ID

비밀번호

이름


전화번호

회원가입완료

회원가입

메시지

×



회원가입이 완료되었습니다.

확인

전화번호

010

회원가입완료



로그인



ID

한글

비밀번호

ID찾기

메시지



로그인 성공

확인



회원정보

ID	이름	전화번호	남은시간
한글	한글	010	00:00:00

시간충전

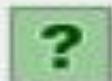
좌석선택

로그아웃

회원정보

ID	이름	전화번호	남은시간
한글	한글	010	00:00:00

입력



충전 시간을 입력하세요.

5

확인

취소

시간충전

좌석선택

로그아웃



회원정보

ID	이름	전화번호	남은시간
한글	한글	010	05:00:00

시간충전

좌석선택

로그아웃



1

2

3

4

5

6

7

11

12

13

14

18

19

20

8

9

10

15

16

17

취소

확인



로그인



ID

한글

비밀번호 메시지



퇴실이 완료되었습니다.

ID찾기

확인

퇴실

1

2

3

4

5

6

7

8

9

10

11

12

15

16

17

13

14

18

19

20

취소

확인

```
// 로그인 액션
loginBtn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        conn = DB.getConnection();
        String username = tfUsername.getText();
        String password = tfPassword.getText();
        MemberDao dao = MemberDao.getInstance();
        int result = dao.findByUsernameAndPassword(username, password);
        if (result == 1) {
            // 로그인 성공 메시지
            JOptionPane.showMessageDialog(null, "로그인 성공");
            // 회원 정보 리스트 화면 이동과 동시에 username,password 세션값으로 넘김.
            MemberListFrame mlf = new MemberListFrame(username, password);
            // 현재 화면 종료

            dispose();
        } else {
            try {
                JOptionPane.showMessageDialog(null, "로그인 실패");

                if (username.equals("")) {
                    JOptionPane.showMessageDialog(null, "아이디를 입력하세요.");
                } else {
                    JOptionPane.showMessageDialog(null, "아이디가 틀렸습니다.");
                }

                if (password.equals("")) {
                    JOptionPane.showMessageDialog(null, "비밀번호를 입력하세요.");
                } else {
                    JOptionPane.showMessageDialog(null, "비밀번호가 틀렸습니다.");
                }
            }
        }
    }
});
```

```

if (result == 1) {
    JOptionPane.showMessageDialog(null, "회원가입이 완료되었습니다.");
    try {
        conn.setAutoCommit(false);
        pstmt = conn.prepareStatement("update time set id= ? where num = (select max(num) from time)");
        pstmt.setString(1, tfUsername.getText());
        pstmt.executeUpdate();
        pstmt.close();
        conn.commit();

        pstmt = conn.prepareStatement("update timemanager set id= ? where num = (select max(num) from timemanager)");
        pstmt.setString(1, tfUsername.getText());
        pstmt.executeUpdate();
        pstmt.close();
        conn.commit();
    } catch (SQLException e1) {
    }

    dispose();
} else{
    try {
        JOptionPane.showMessageDialog(null, "회원가입이 실패하였습니다.");
        String ID = tfUsername.getText();
        String PHONE = tfPhone.getText();
        String DB_ID = null;
        String DB_PHONE = null;
        pstmt = conn.prepareStatement("select username from member where username = ?");
        pstmt.setString(1, ID);
        rs = pstmt.executeQuery();
        while (rs.next()) {
            DB_ID = rs.getString("username");
        }

        if(ID.equals(DB_ID)) {
            JOptionPane.showMessageDialog(null, "존재하는 아이디입니다.");
        }

        pstmt = conn.prepareStatement("select Phone from member where Phone = ?");
        pstmt.setString(1, PHONE);
        rs = pstmt.executeQuery();
        while (rs.next()) {
            DB_PHONE = rs.getString("phone");
        }
        if(PHONE.equals(DB_PHONE)) {
            JOptionPane.showMessageDialog(null, "존재하는 전화번호 입니다.");
        }
        else {
            JOptionPane.showMessageDialog(null, "전화번호를 입력하세요.");
        }
    }
}

```

```
btnOK.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        try {  
            conn.setAutoCommit(false);  
            pstmt = conn.prepareStatement("update timemanager set seat = ? where ID = ?");  
            pstmt.setInt(1, Action.num);  
            pstmt.setString(2, ID);  
            pstmt.executeUpdate();  
            pstmt.close();  
            conn.commit();  
  
            pstmt = conn.prepareStatement("update timemanager set time1 = systimestamp where ID = ?");  
            pstmt.setString(1, ID);  
            pstmt.executeUpdate();  
            pstmt.close();  
            conn.commit();  
            jf.dispose();  
  
        } catch (SQLException e1) {  
            e1.printStackTrace();  
        }  
    }  
});
```

```
import javax.swing.JButton;
import javax.swing.JOptionPane;

public class Action implements ActionListener {

    int k=0;

    static int num;

    @Override
    public void actionPerformed(ActionEvent e) {
        JButton btn = (JButton) e.getSource();
        String seatnum = btn.getText();

        num = Integer.parseInt(seatnum);

        k++;

        if(k==1) {
            btn.setBackground(Color.cyan);
            btn.setEnabled(false);
        }
        else {
            JOptionPane.showMessageDialog(null, "더 이상 선택할 수 없습니다.");
            k=1;
        }
    }
}
```

// 과석선택

```
btnSelSeat.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            pstmt = conn.prepareStatement("select seat from timemanager where ID = ?");
            pstmt.setString(1, ID);
            rs = pstmt.executeQuery();
            while (rs.next()) {
                seat1 = rs.getInt("seat");
            }

        } catch (Exception e2) {

        }
        if (seat1 == 0) {
            Seat st = new Seat(ID);
        }
    }
});
```

```
btnAddTime.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        String num = null; // 충전할 시간
        num = JOptionPane.showInputDialog("충전 시간을 입력하세요.", "");
        try {
            pstmt = conn.prepareStatement("select time from time where ID = ?");
            pstmt.setString(1, ID);
            rs = pstmt.executeQuery();
            while (rs.next()) {
                TimeFunction.myTime = rs.getInt("time");
            }
            TimeFunction.addTime(Integer.parseInt(num)*3600);
        } catch (SQLException e1) {

        } catch (NumberFormatException e2) {

        }

        try {
            pstmt = conn.prepareStatement("update time set time= ? where ID = ?");
            pstmt.setInt(1, TimeFunction.myTime);
            pstmt.setString(2, ID);
            pstmt.executeUpdate();
        }
```