

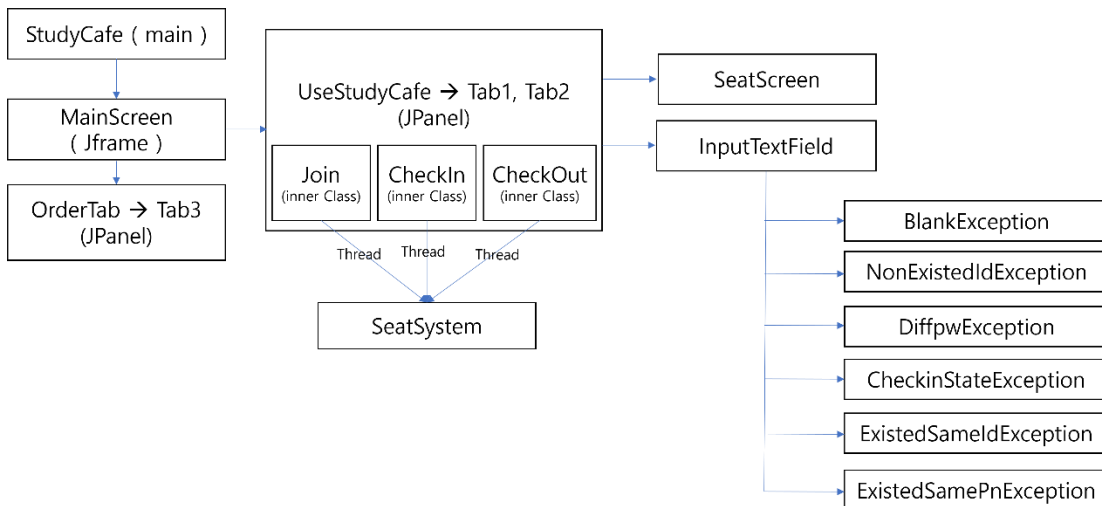
서예빈 1912739 Term Project

텀 프로젝트로 스터디카페 키오스크 기능을 구현하여 보았다. 전체적으로 입실과 퇴실, 그리고 음료 주문 기능이 있다. 입실과 퇴실 과정에는 로그인이 필요하며, 회원이 아닌 경우 회원 가입을 할 수 있도록 구성하였다.

입실 시 로그인이 되면 빈 좌석 중 원하는 좌석을 선택하여 입실 할 수 있고, 퇴실 시 로그인이 되면 본인이 입실하였던 좌석에서 퇴실 처리가 되고, 그 좌석은 빈 자리가 된다.

음료주문 기능에서는 원하는 음료 최대 5가지를 선택할 수 있도록 하였다.

[StudyCafe 객체 구조]



[StudyCafe.java]

```
1 class StudyCafe {
2     public static void main(String[] args) {
3         MainScreen MS = new MainScreen();
4     }
5 }
```

[MainScreen.java]

```
1 import java.awt.*;
2 import javax.swing.*;
3
4 class MainScreen extends JFrame{
5     JTabbedPane tab;
6     OrderTab order;
7     UseStudyCafe checkin;
8     JPanel checkout;
9     Font font = new Font("HY영서M", Font.PLAIN, 15);
10
11     public MainScreen() {
12         setTitle("서예빈 1912739 스터디카페 키오스크");
13         tab = new JTabbedPane();
14         checkin = new UseStudyCafe();
15         checkout = checkin.ckoutlogin;
16         order = new OrderTab();
17
18         tab.addTab("입실", checkin);
19         tab.addTab("퇴실", checkout);
20         tab.addTab("메뉴 주문", order);
21         tab.setBackgroundAt(0, new Color(255, 200, 200));
22         tab.setBackgroundAt(1, new Color(255, 250, 150));
23         tab.setBackgroundAt(2, new Color(185, 255, 180));
24
25         tab.setFont(font);
26
27         getContentPane().add(tab);
28         getContentPane().setBackground(Color.white);
29         setSize(800, 700);
30         setVisible(true);
31     }
32 }
```

⇒ 스터디 카페 키오스크를 '입실', '퇴실', '메뉴 주문' 탭으로 구성함.

입실 탭에는 UseStudyCafe 클래스(JPanel)를, 퇴실 탭에는 UseStudyCafe 클래스의 ckoutlogin JPanel을, 메뉴 주문 탭에는 OrderTab 클래스(JPanel)으로 구성함.

[UseStudyCafe.java]

```
1 import java.awt.*;
2 import javax.swing.*;
3 import java.awt.event.*;
4 import java.sql.Timestamp;
5 import java.text.SimpleDateFormat;
6
7 class UseStudyCafe extends JPanel implements ActionListener
8 {
9     JButton login, signup, loginbtn, lgcancel, signupbtn, sucancel, selectcancel, ok, loginbtn2, lgcancel2, findid, findpw, findid2, findpw2;
10    JPanel loginscreen, signupscreen, checkininit, loginscreen2, ckoutlogin;
11    JLabel lgidl, lgpw1, suidl, supw1, supwck1, suname1, suphenum1, lgidl2, lgpw12;
12    JTextField lgidtf, suidtf, sunametf, suphenumtf, lgidtf2;
13    JPasswordField lgpwtf, supwtf, supwck, lgpwtf2;
14
15    GridBagConstraints constraint = new GridBagConstraints();
16    GridBagLayout gridbag = new GridBagLayout();
17
18    SeatScreen seatscreen;
19    SeatSystem seatsystem = new SeatSystem();
20    Join joinmem = new Join(seatssystem);
21    CheckIn checkin = new CheckIn(seatssystem);
22    CheckOut checkout = new CheckOut(seatssystem);
23
24    String inputname, inputid, inputpw, inputpwck, inputpn;
25    int index, answer;
26    int memnum = 0;
27
28    boolean checkinstart = false;
29    boolean joinstart = false;
30    boolean checkoutstart = false;
31
32    Color bg = new Color(230,225,255);
33    Color pink = new Color(255,150,150);
34    Color blue = new Color(130,170,250);
35    Color yellow = new Color(255,250,150);
36    Color gray = new Color(230,230,230);
37
38    Font bigFont = new Font("HY영체",Font.PLAIN,20);
39    Font smallFont = new Font("HY영체",Font.PLAIN,15);
40    Font smallestfont = new Font("HY영체",Font.PLAIN,10);
41
42    InputTextField itf = new InputTextField(); // Exception
43
44    SimpleDateFormat sdf = new SimpleDateFormat("yyyy년 MM월 dd일 hh시 mm분 ");
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
```

```
public UseStudyCafe() {
```

```
    //입실 초기화면
    login = new JButton("로그인");
    login.addActionListener(this);
    login.setFont(bigFont);
    login.setBackground(pink);
    signup = new JButton("회원가입");
    signup.addActionListener(this);
    signup.setFont(bigFont);
    signup.setBackground(blue);
    checkininit = new JPanel();
    checkininit.add(login);
    checkininit.add(signup);
    checkininit.setBackground(bg);
```

⇒ checkininit : 입실 탭의 초기화면 (로그인 버튼 / 회원 가입 버튼)

```
    add(checkininit);
    setBackground(bg);
```

```
    //로그인 화면 ( 입실 )
    loginscreen = new JPanel();
    loginscreen.setLayout(gridbag);
    constraint.fill = GridBagConstraints.BOTH;
    loginscreen.setBackground(bg);
```

```
    lgidl = new JLabel("ID");
    lgidtf = new JTextField(15);
    lgpw1 = new JLabel("PW");
    lgpwtf = new JPasswordField(15);
    lgpwtf.setEchoChar('*');
```

⇒ loginscreen : checkininit에서 로그인을 눌렀을 때 (입실 시 로그인 화면)

```
    loginbtn = new JButton("로그인");
    loginbtn.addActionListener(this);
    loginbtn.setFont(smallFont);
    loginbtn.setBackground(pink);
    lgcancel = new JButton("취소");
    lgcancel.addActionListener(this);
    lgcancel.setFont(smallFont);
    lgcancel.setBackground(Color.WHITE);
```

```
    findid = new JButton("ID 찾기");
    findid.addActionListener(this);
    findid.setFont(smallestfont);
    findid.setBackground(gray);
```

```
    findpw = new JButton("PW 찾기");
    findpw.addActionListener(this);
    findpw.setFont(smallestfont);
```

```

97 findpw.setBackground(gray);
98
99 layoutP(loginscreen,lgidl,0,0,1,1);
100 layoutP(loginscreen,lgpw1,0,1,1,1);
101 layoutP(loginscreen,lgidtf,1,0,2,1);
102 layoutP(loginscreen,lgpwtf,1,1,2,1);
103 layoutP(loginscreen,loginbtn,1,2,1,1);
104 layoutP(loginscreen,lgcancel,2,2,1,1);
105 layoutP(loginscreen,findid,3,0,1,1);
106 layoutP(loginscreen,findpw,3,1,1,1);
107
108 //로그인 화면 ( 퇴실 )
109 loginscreen2 = new JPanel();
110 loginscreen2.setBackground(bg);
111 loginscreen2.setLayout(gridbag);
112 constraint.fill = GridBagConstraints.BOTH;
113
114 lgidl2 = new JLabel("ID");
115 lgidtf2 = new JTextField(15);
116 lgpw12 = new JLabel("PW");
117 lgpwtf2 = new JPasswordField(15);
118 lgpwtf2.setEchoChar('*');
119
120 loginbtn2 = new JButton("로그인");
121 loginbtn2.addActionListener(this);
122 loginbtn2.setFont(smallfont);
123 loginbtn2.setBackground(yellow);
124
125 lgcancel2 = new JButton("취소");
126 lgcancel2.addActionListener(this);
127 lgcancel2.setFont(smallfont);
128 lgcancel2.setBackground(Color.WHITE);
129
130 findid2 = new JButton("ID 찾기");
131 findid2.addActionListener(this);
132 findid2.setFont(smallestfont);
133 findid2.setBackground(gray);
134
135 findpw2 = new JButton("PW 찾기");
136 findpw2.addActionListener(this);
137 findpw2.setFont(smallestfont);
138 findpw2.setBackground(gray);
139
140 layoutP(loginscreen2,lgidl2,0,0,1,1);
141 layoutP(loginscreen2,lgpw12,0,1,1,1);
142 layoutP(loginscreen2,lgidtf2,1,0,2,1);
143 layoutP(loginscreen2,lgpwtf2,1,1,2,1);
144 layoutP(loginscreen2,loginbtn2,1,2,1,1);
145 layoutP(loginscreen2,lgcancel2,2,2,1,1);
146 layoutP(loginscreen2,findid2,3,0,1,1);
147 layoutP(loginscreen2,findpw2,3,1,1,1);
148
149 ckoutlogin = new JPanel(new BorderLayout());
150 ckoutlogin.setBackground(bg);
151 ckoutlogin.add(loginscreen2,BorderLayout.NORTH);
152
153 //회원 가입 화면 ( 입실 )
154 signupscreen = new JPanel();
155 signupscreen.setLayout(gridbag);
156 signupscreen.setBackground(bg);
157
158 suidl = new JLabel("ID");
159 suidtf = new JTextField(15);
160 supw1 = new JLabel("PW");
161 supwtf = new JPasswordField(15);
162 supwtf.setEchoChar('*');
163 supwck1 = new JLabel("PW 확인");
164 supwck = new JPasswordField(15);
165 supwck.setEchoChar('*');
166 sunamel = new JLabel("이름");
167 sunametf = new JTextField(15);
168 suphonenum1 = new JLabel("전화번호");
169 suphonenumtf = new JTextField(15);
170
171 signubtn = new JButton("회원가입");
172 signubtn.addActionListener(this);
173 signubtn.setFont(smallfont);
174 signubtn.setBackground(blue);
175
176 sucancel = new JButton("취소");
177 sucancel.addActionListener(this);
178 sucancel.setFont(smallfont);
179 sucancel.setBackground(Color.WHITE);
180
181 layoutP(signupscreen,sunamel,0,0,1,1);
182 layoutP(signupscreen,suidl,0,1,1,1);
183 layoutP(signupscreen,supw1,0,2,1,1);
184 layoutP(signupscreen,supwck1,0,3,1,1);
185 layoutP(signupscreen,suphonenum1,0,4,1,1);
186 layoutP(signupscreen,sunametf,1,0,2,1);
187 layoutP(signupscreen,suidtf,1,1,2,1);
188 layoutP(signupscreen,supwtf,1,2,2,1);
189 layoutP(signupscreen,supwck,1,3,2,1);
190 layoutP(signupscreen,suphonenumtf,1,4,2,1);
191 layoutP(signupscreen,signubtn,1,5,1,1);
192 layoutP(signupscreen,sucancel,2,5,1,1);
193
194 ok = new JButton("확인");
195 ok.addActionListener(this);
196 ok.setFont(smallfont);
197 ok.setBackground(new Color(200,215,180));
198
199 selectcancel = new JButton("취소 및 로그아웃");
200 selectcancel.addActionListener(this);
201 selectcancel.setFont(smallfont);
202 selectcancel.setBackground(Color.WHITE);
203
204
205 public void actionPerformed(ActionEvent e){
206     Object o = e.getSource();
207     if(o == login) {
208         removeAll();
209         add(loginscreen);
210         revalidate();
211         repaint();

```

⇒ layoutP로 loginscreen(→ gridbaglayout)에 컴포넌트 배치
(layoutP : 첫번째 인자로 받은 JPanel에 배치하는 함수)

⇒ loginscreen2 : 퇴실할 때의 로그인 화면

⇒ 위 loginscreen2를 새로운 JPanel인 ckoutlogin의 NORTH에 추가
ckoutlogin이 퇴실 탭의 화면

⇒ signupscreen : checkininit에서 회원가입 눌렀을 때

⇒ seatsscreen에 추가할 확인 버튼 (좌석 선택 후 입실 처리)

⇒ seatsscreen에 추가할 취소 및 로그아웃 버튼

⇒ 초기화면에서 로그인 버튼을 누르면 화면의 요소들이 사라지고,

```

212 日 } else if(o == signup) {
213      removeAll();
214      add(signupscreen);
215      revalidate();
216      repaint();
217 日 } else if(o == loginbtn){
218      inputid = (String)lgidtf.getText();
219      inputpw = (String)lgpwtf.getText();
220 日 try {
221          itf.checktfblank(inputid);
222          itf.checktfblank(inputpw);
223          itf.checkexistedid(seatsystem.idarray,inputid);
224          index = seatsystem.idarray.indexOf(inputid);
225          itf.checkdiffpw(seatsystem.pwarray.get(index),inputpw);
226          itf.checkstate(seatsystem.memberstate.get(index));
227
228          seatscreen = new SeatScreen(seatsystem.seatcheck);
229          seatscreen.layout(selectcancel,8,11,3,1);
230          seatscreen.layout(ok,7,11,1,1);
231          removeAll();
232          add(seatscreen);
233          revalidate();
234          repaint();
235          lgidtf.setText(null);
236          lgpwtf.setText(null);

```

loginscreen을 나타냄

⇒ 초기화면에서 회원가입을 누르면 signupscreen이 나타남.

⇒ loginscreen의 로그인 버튼을 누르면 현재 입력창에 있는 값들을 String으로 저장하고, try/catch문을 실행함으로써 try문에서 예외가 발생하면 catch문 실행

- jtextfield 가 공백인지 확인
- 입력한 id가 존재하는 id인지 확인
- 존재한다면 (예외발생x) 그 아이디가 저장된 값의 index를 index에 저장
- 그 index의 pw와 입력창의 pw가 같은지 확인
- Index의 회원이 현재 입실 상태인지 확인 (이미 입실 상태일 경우 재입실 불가)
- 위의 과정에서 예외 발생하지 않았다면 seatscreen을 띄우고, 위에서 정의한 확인버튼과 취소 및 로그아웃 버튼을 추가
- Id와 pw의 텍스트 필드 초기화

```

237 日 } catch (BlankException be) { JOptionPane.showMessageDialog(loginscreen,"빈칸을 채워주세요."); }
238      catch (NonExistedIdException neie) { JOptionPane.showMessageDialog(loginscreen,"존재하지 않는 아이디 입니다.");}
239      catch (DiffpwException de) { JOptionPane.showMessageDialog(loginscreen,"비밀번호가 맞지 않습니다.");}
240      catch (CheckinStateException cise) { JOptionPane.showMessageDialog(loginscreen,"이미 입실 상태 입니다.");}

```

- 위의 try문에서 예외가 발생한 경우 경고창을 나타냄.

```

241 日 } else if ( o == signupbtn ) {
242      inputname = (String)sunametf.getText();
243      inputid = (String)suidtf.getText();
244      inputpw = (String)supwtf.getText();
245      inputpwck = (String)supwck.getText();
246      inputpn = (String)suphonenumtf.getText();
247 日 try {
248          itf.checktfblank(inputname);
249          itf.checktfblank(inputid);
250          itf.checktfblank(inputpw);
251          itf.checktfblank(inputpwck);
252          itf.checktfblank(inputpn);
253          itf.checksameid(seatsystem.idarray,inputid);
254          itf.checkdiffpw(inputpw, inputpwck);
255          itf.checksamepn(seatsystem.phonenumarray,inputpn);
256 日 if (joinstart == false) {
257              joinmem.start();
258              joinstart = true;
259          } else joinmem.run();
260          JOptionPane.showMessageDialog(signupscreen,"회원가입 완료");
261          removeAll();
262          add(loginscreen);
263          revalidate();
264          repaint();
265          sunametf.setText(null);
266          suidtf.setText(null);
267          supwtf.setText(null);
268          supwck.setText(null);
269          suphonenumtf.setText(null);
270 日 } catch (BlankException be) { JOptionPane.showMessageDialog(signupscreen,"빈칸을 채워주세요."); }
271      catch (ExistedSameIdException esie) { JOptionPane.showMessageDialog(signupscreen,"동일한 아이디가 존재합니다."); }
272      catch (DiffpwException esie) { JOptionPane.showMessageDialog(signupscreen,"비밀번호 확인을 다시 해주세요."); }
273      catch (ExistedSamePnException espe) { JOptionPane.showMessageDialog(signupscreen,"동일한 번호로 가입한 아이디가 존재합니다."); }
274 日 } else if(o == lgcancel || o == sucancel ){
275      removeAll();
276      add(checkininit);
277      revalidate();
278      repaint();
279      sunametf.setText(null);
280      suidtf.setText(null);
281      supwtf.setText(null);
282      supwck.setText(null);
283      suphonenumtf.setText(null);
284      lgidtf.setText(null);
285      lgpwtf.setText(null);
286 日 } else if(o == ok) {
287      if(seatscreen.selected == -1) JOptionPane.showMessageDialog(seatscreen,"과석을 선택해주세요.");
288 日 else {
289      if (checkinstart == false) {
290          checkin.start();
291          checkinstart = true;
292      } else checkin.run();

```

⇒ 마찬가지로 회원가입 창에서 회원가입 버튼을 누르면 각종 예외가 발생하는지 확인

⇒ 입력창에서 예외가 발생하지 않으면 Join 스레드 발생

⇒ 회원가입창이나 입실의 로그인 창에서 취소를 누르면 현재 나타나 있는 창은 지우고 입실 초기화면이 나옴. 그리고 입력창 모두 초기화

```

293         Timestamp citime = new Timestamp(System.currentTimeMillis());
294         JOptionPane.showMessageDialog(seatscreen,sdf.format(citime)+seatsystem.namearray.get(index)+"님, 입실 완료");
295         removeAll();
296         add(checkininit);
297         revalidate();
298         repaint();
299         seatscreen.selected = -1;

```

⇒ seatscreen에서 ok창을 눌렀을 때 seatscreen.selected가 -1이라면 (좌석 버튼을 누르지 않은 상태) 좌석을 선택해 주세요 라는 메시지 창이 뜬.

⇒ 좌석 버튼을 선택 후 ok버튼을 눌렀다면 CheckIn Thread 수행하고, 현재 시각, 회원의 이름과 함께 입실 완료 메시지가 나타남.

⇒ 또한 입실 초기화면으로 돌아감. (seatscreen.selected의 값은 -1로 초기화)

```

300     }
301     } else if(o == loginbtn2){
302         String inputid = (String)lgidtf2.getText();
303         String inputpw = (String)lgpwtf2.getText();
304         try {
305             itf.checktfblank(inputid);
306             itf.checktfblank(inputpw);
307             itf.checkexistedid(seatsystem.idarray,inputid);
308             index = seatsystem.idarray.indexOf(inputid);
309             itf.checkdiffpw(seatsystem.pwarray.get(index),inputpw);
310             itf.checkoutstate(seatsystem.memberstate.get(index));
311
312             String comsg = seatsystem.namearray.get(index)+"님, 퇴실하시겠습니까? [ 좌석번호 : "+(seatsystem.memberseat.get(index)+1)+" ]";
313             answer = JOptionPane.showConfirmDialog(loginscreen2, comsg, "퇴실 확인", JOptionPane.YES_NO_OPTION);
314
315             if(answer == JOptionPane.YES_OPTION) {
316                 if (checkoutstart == false) {
317                     checkout.start();
318                     checkoutstart = true;
319                 } else checkout.run();
320                 lgidtf2.setText(null);
321                 lgpwtf2.setText(null);
322                 Timestamp cotime = new Timestamp(System.currentTimeMillis());
323                 JOptionPane.showMessageDialog(loginscreen2,sdf.format(cotime)+seatsystem.namearray.get(index)+"님, 퇴실 완료");
324             }
325         } catch (BlankException be) { JOptionPane.showMessageDialog(loginscreen,"빈칸을 채워주세요."); }
326         catch (NonExistedIdException neie) { JOptionPane.showMessageDialog(loginscreen,"존재하지 않는 아이디입니다."); }
327         catch (DiffpwException de) { JOptionPane.showMessageDialog(loginscreen,"비밀번호가 맞지 않습니다."); }
328         catch (CheckinStateException cise) { JOptionPane.showMessageDialog(loginscreen,"이미 퇴실 상태입니다."); }
329     } else if(o == lgcancel2){
330         lgidtf2.setText(null);
331         lgpwtf2.setText(null);
332     } else if( o == selectcancel ) {
333         seatscreen.selected = -1;
334         removeAll();
335         add(loginscreen);
336         revalidate();
337         repaint();
338     } else if ( o == findid || o == findid2 ) {
339         String id;
340         if(o == findid) id = JOptionPane.showInputDialog(loginscreen, "전화번호를 입력하세요", "아이디 찾기", JOptionPane.OK_CANCEL_OPTION);
341         else id = JOptionPane.showInputDialog(loginscreen2, "전화번호를 입력하세요", "아이디 찾기", JOptionPane.OK_CANCEL_OPTION);
342         try {
343             itf.checkexistedid(seatsystem.phonenumarray,id);
344             index = seatsystem.phonenumarray.indexOf(id);
345             if(o == findid) JOptionPane.showMessageDialog(loginscreen,"해당 번호로 가입된 아이디는 [" +seatsystem.idarray.get(index)+"] 입니다.");
346             else JOptionPane.showMessageDialog(loginscreen2,"해당 번호로 가입된 아이디는 [" +seatsystem.idarray.get(index)+"] 입니다.");
347         } catch (NonExistedIdException neie) {
348             if(o == findid) JOptionPane.showMessageDialog(loginscreen,"해당 번호로 가입된 아이디는 없습니다.");
349             else JOptionPane.showMessageDialog(loginscreen2,"해당 번호로 가입된 아이디는 없습니다.");
350         }
351     } else if ( o == findpw || o == findpw2 ) {
352         String pw;
353         if( o == findpw ) pw = JOptionPane.showInputDialog(loginscreen, "ID를 입력하세요", "비밀번호 찾기", JOptionPane.OK_CANCEL_OPTION);
354         else pw = JOptionPane.showInputDialog(loginscreen2, "ID를 입력하세요", "비밀번호 찾기", JOptionPane.OK_CANCEL_OPTION);
355         try {
356             itf.checkexistedid(seatsystem.idarray,pw);
357             index = seatsystem.idarray.indexOf(pw);
358             if( o == findpw ) JOptionPane.showMessageDialog(loginscreen,"해당 아이디의 비밀번호는 가입시 입력된 전화번호로 발송됩니다.");
359             else JOptionPane.showMessageDialog(loginscreen2,"해당 아이디의 비밀번호는 가입시 입력된 전화번호로 발송됩니다.");
360         } catch (NonExistedIdException neie) {
361             if(o == findpw ) JOptionPane.showMessageDialog(loginscreen,"존재하지 않는 아이디 입니다.");
362             else JOptionPane.showMessageDialog(loginscreen2,"존재하지 않는 아이디 입니다.");
363         }
364     }
365 }

```

⇒ 퇴실 창에서의 로그인 버튼을 누르면 입실창과 마찬가지로 try/catch문 실행 (textfield에서 예외가 발생하지 않았으면 퇴실 확인 메시지가 뜨고 여기서 Yes버튼을 누르면 퇴실 처리와 함께 퇴실 처리 완료 메시지가 뜬.)

⇒ findid : 입실 로그인창의 아이디 찾기 버튼 / findid2 : 퇴실 로그인창의 아이디 찾기

메시지 창에 전화번호를 입력하고 id에 저장하면 try/catch문을 통해 phonenumarray에 있는지 확인 후 있으면 아이디를 메시지창에 나타내고, 없으면 "해당 번호로 가입된 아이디는 없습니다." 라는 메시지 창을 띄움.

⇒ 마찬가지로 findpw와 findpw2에 대해서도 아이디를 입력받아 idarray와 비교함. 해당 아이디가 없으면 "존재하지 않는 아이디 입니다."라는 메시지를 띄우고, 있으면 저장된 번호로 발송한다는 메시지를 띄움.

```

366     public void layout(Component obj, int x, int y, int width, int height) {
367         constraint.gridx = x;
368         constraint.gridy = y;
369         constraint.gridwidth = width;
370         constraint.gridheight = height;
371         add(obj, constraint);
372     }
373 }

```

```

374 public void layout(JPanel p, Component obj, int x, int y, int width, int height) {
375     constraint.gridx = x;
376     constraint.gridy = y;
377     constraint.gridwidth = width;
378     constraint.gridheight = height;
379     p.add(obj, constraint);
380 }
381
382 class Join extends Thread {
383     public Join(SeatSystem s) {
384         seatSystem = s;
385     }
386
387     public void run() {
388         seatSystem.join(inputname, inputid, inputpw, inputpn);
389     }
390 }
391
392 class CheckIn extends Thread {
393     public CheckIn(SeatSystem s) {
394         seatSystem = s;
395     }
396
397     public void run() {
398         seatSystem.ckin(seatScreen.selected, index);
399     }
400 }
401
402
403
404 class CheckOut extends Thread {
405     public CheckOut(SeatSystem s) {
406         seatSystem = s;
407     }
408
409     public void run() {
410         seatSystem.ckout(seatSystem.memberseat.get(index), index);
411     }
412 }
413

```

⇒ Join, CheckIn, CheckOut 스레드

(각각 seatSystem 클래스의 함수를 호출)

[SeatSystem.java]

```

1 import java.util.ArrayList;
2
3 class SeatSystem{
4     static int seatcheck[] = new int[41];
5     ArrayList<String> namearray = new ArrayList<>();
6     ArrayList<String> idarray = new ArrayList<>();
7     ArrayList<String> pwarray = new ArrayList<>();
8     ArrayList<String> phonenumarray = new ArrayList<>();
9     ArrayList<Integer> memberstate = new ArrayList<Integer>(); //각 인덱스의 회원이 입실 상태인가?
10    ArrayList<Integer> memberseat = new ArrayList<Integer>(); //각 인덱스의 회원의 좌석 ( 입실상태라면 좌석번호, 퇴실상태라면 -1 )
11
12    public SeatSystem(){ // SeatSystem()이 처음 호출되면 seatcheck배열의 모든 값을 0으로 초기화 ( 모든 좌석이 비어있음 )
13        for(int i=0; i<seatcheck.length; i++){
14            seatcheck[i] = 0;
15        }
16    }
17
18    public synchronized void ckin(int seatnum, int index) { //입실 ( 선택한 좌석 번호, 회원 리스트에서 현재 로그인된 사람의 인덱스를 매개변수로 받음 )
19        seatcheck[seatnum] = 1; //선택된 좌석을 입실상태로 바꿈 ( 0 : 공석, 1 : 입실 )
20        memberstate.set(index,1); //index 회원은 현재 입실상태
21        memberseat.set(index,seatnum); //index회원의 좌석 : seatnum
22    }
23
24    public synchronized void ckout(int seatnum, int index) { //퇴실 ( 입실했던 좌석번호, 회원의 인덱스를 매개변수로 받음 )
25        seatcheck[seatnum] = 0; // seatnum 좌석은 공석
26        memberstate.set(index,0); // index 회원은 퇴실함 -> 0
27        memberseat.set(index,-1); // index 회원이 퇴실함으로써 회원의 좌석은 없어짐 -> -1 ( 좌석 번호가 0부터 시작이므로 -1 )
28    }
29
30    public synchronized void join(String name, String id, String pw, String pnum) { //회원가입 ( 이름, id, pw, 전화번호를 매개변수로 받음 )
31        // 각 인자를 arraylist에 추가하고, 아직 입실 전이므로 memberstate는 0, memberseat는 -1
32        namearray.add(name);
33        idarray.add(id);
34        pwarray.add(pw);
35        phonenumarray.add(pnum);
36        memberstate.add(0);
37        memberseat.add(-1);
38    }
39 }

```

- ⇒ SeatSystem이 처음 호출 되면 모든 좌석이 공석이 되므로 seatcheck 배열은 모두 0으로 초기화
- ⇒ ckin 함수 : CheckIn 스레드 실행시 호출되는 함수 (사용자가 선택한 좌석번호와, 그 사용자의 회원list의 index가 인자로 입력되며, seatnum번의 좌석은 입실상태이므로 1, memberstate(→인덱스에 해당하는 회원의 입실, 퇴실 여부)도 1, memberseat(→인덱스에 해당하는 회원이 앉아있는 좌석)은 seatnum으로 설정.)
- ⇒ ckout 함수 : CheckOut 스레드 실행시 호출되는 함수 (seatnum은 공석이 되므로 0, 회원 또한 퇴실하므로 memberstate의 인덱스 값은 0, 회원이 앉아있는 좌석도 없어지므로 -1로 초기화)
- ⇒ join 함수 : Join 스레드 실행시 호출되는 함수 (새로운 회원이 입력한 name, id, pw, pnum을 각 arraylist에 등록하며, 아직 입실상태는 아니므로 memberstate는 0으로, memberseat는 -1로 초기화)

[SeatScreen.java]

```

1  import java.awt.*;
2  import javax.swing.*;
3  import java.awt.event.*;
4
5  class SeatScreen extends JPanel implements ActionListener {
6      GridBagConstraints constraint;
7      GridBagLayout gridbag = new GridBagLayout();
8      JButton seatbtn[] = new JButton[41];
9      String[] seatnum = {"1","2","3","4","5","6","7","8","9","10",
10         "11","12","13","14","15","16","17","18","19","20",
11         "21","22","23","24","25","26","27","28","29","30",
12         "31","32","33","34","35","36","37","38","39","40","41"};
13
14     static int selected = -1;
15     String input;
16     int[] seatstate;
17     JButton cafe, lounge, entrance, locker1, locker2;
18     Font smallfont = new Font("HYUNSE",Font.PLAIN,15);
19
20     public SeatScreen(int[] seatarray) {
21         seatstate = seatarray;
22         setLayout(gridbag);
23         constraint = new GridBagConstraints();
24         constraint.fill = GridBagConstraints.BOTH;
25         setBackground(new Color(230,225,255));
26
27         for (int i=0; i<seatbtn.length; i++){
28             seatbtn[i] = new JButton(seatnum[i]);
29             seatbtn[i].addActionListener(this);
30             seatbtn[i].setFont(smallfont);
31             seatbtn[i].setBackground(new Color(210,255,130));
32
33             //좌석 버튼 배치
34             if(i<4) layout(seatbtn[i],0,4-i,1,1);
35             else if(i<13) layout(seatbtn[i],1-3,0,1,1);
36             else if(i<17) layout(seatbtn[i],10,i-12,1,1);
37             else if(i<20) layout(seatbtn[i],i-15,2,1,1);
38             else if(i<23) layout(seatbtn[i],i-18,3,1,1);
39             else if(i<26) layout(seatbtn[i],i-17,2,1,1);
40             else if(i<29) layout(seatbtn[i],i-20,3,1,1);
41             else if(i<32) layout(seatbtn[i],i-27,5,1,1);
42             else if(i<35) layout(seatbtn[i],i-30,6,1,1);
43             else if(i<38) layout(seatbtn[i],i-29,5,1,1);
44             else layout(seatbtn[i],i-32,6,1,1);
45
46             if (seatstate[i] == 1) seatbtn[i].setEnabled(false); // 이미 입실상태인 좌석 선택불가
47             else seatbtn[i].setEnabled(true);
48         }
49
50         cafe = new JButton("카페");
51         cafe.setEnabled(false);
52
53         lounge = new JButton("휴게실");
54         lounge.setEnabled(false);
55
56         entrance = new JButton("입구");
57         entrance.setEnabled(false);
58
59         locker1 = new JButton("사물함");
60         locker1.setEnabled(false);
61
62         locker2 = new JButton("사물함");
63         locker2.setEnabled(false);
64
65         layout(cafe,8,8,3,3);
66         layout(lounge,0,8,3,3);
67         layout(entrance,4,10,3,1);
68         layout(locker1,0,5,1,3);
69         layout(locker2,10,5,1,3);
70     }
71
72     public void actionPerformed(ActionEvent e){
73         Object o = e.getSource();
74         input = e.getActionCommand();
75         selected = Integer.parseInt(input)-1;
76
77         for (int i=0;i<seatbtn.length;i++) {
78             if(seatstate[i] == 0) { //비어있는 좌석에 한해서 선택한 좌석을 제외한 모든 좌석 클릭 가능
79                 if( o == seatbtn[i] ) seatbtn[i].setEnabled(false);
80                 else seatbtn[i].setEnabled(true);
81             }
82         }
83     }
84
85     public void layout(Component obj, int x, int y, int width, int height) {
86         constraint.gridx = x;
87         constraint.gridy = y;
88         constraint.gridwidth = width;
89         constraint.gridheight = height;
90         add(obj, constraint);
91     }
92 }

```

⇒ selected : 회원이 선택한 좌석의 번호

번호는 0~41이므로 -1로 초기화

⇒ SeatScreen 생성자는 seatarray 매개변수로 받음 (seatarray : 좌석들이 공석인지 아닌지를 확인할 수 있는 배열 → seatssystem의 seatcheck)

⇒ for문을 활용하여 모든 좌석 버튼 생성

⇒ 해당 좌석이 입실 상태라면 선택할 수 없음.

⇒ 나머지 구조물에 대해서도 버튼을 생성 하였고 클릭할 수 없도록 설정.

⇒ 좌석 버튼을 눌렀다면 해당 버튼의 값 -1의 값을 정수로 selected에 저장함 (좌석 번호 배열의 인덱스는 0부터 이므로)

⇒ 좌석 버튼을 누르면 해당 버튼은 누를 수 없고 나머지 버튼만 누를 수 있음 (자신이 누른 버튼을 제외한 버튼 클릭 가능)

[BlankException.java]

```

1 public class BlankException extends Exception{
2     public BlankException() {
3         super("빈칸 있음");
4     }
5 }

```

[NonExistedIdException.java]

```
1 public class NonExistedIdException extends Exception{
2     public NonExistedIdException() {
3         super("존재하지 않는 아이디");
4     }
5 }
```

[DiffpwException.java]

```
1 public class DiffpwException extends Exception{
2     public DiffpwException() {
3         super("잘못된 패스워드");
4     }
5 }
```

[CheckinStateException.java]

```
1 public class CheckinStateException extends Exception{
2     public CheckinStateException() {
3         super("입실/퇴실 상태");
4     }
5 }
```

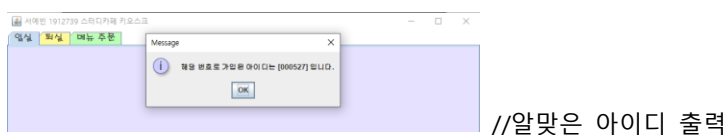
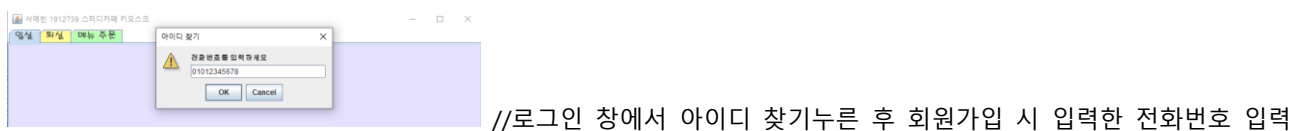
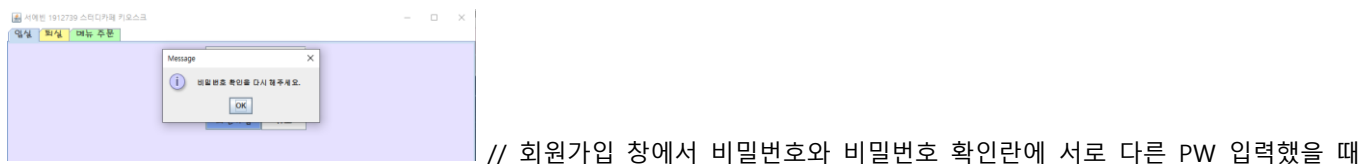
[ExistedSameIdException.java]

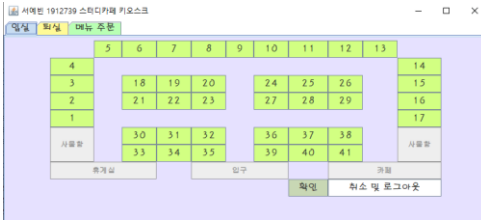
```
1 public class ExistedSameIdException extends Exception{
2     public ExistedSameIdException() {
3         super("동일한 아이디 존재");
4     }
5 }
```

[ExistedSamePnException.java]

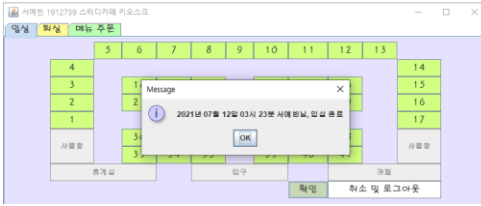
```
1 public class ExistedSamePnException extends Exception{
2     public ExistedSamePnException() {
3         super("동일한 번호로 가입한 아이디 존재");
4     }
5 }
```

[실행 결과]

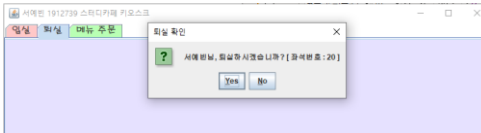




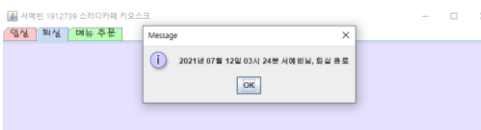
// 로그인 성공시 좌석선택화면



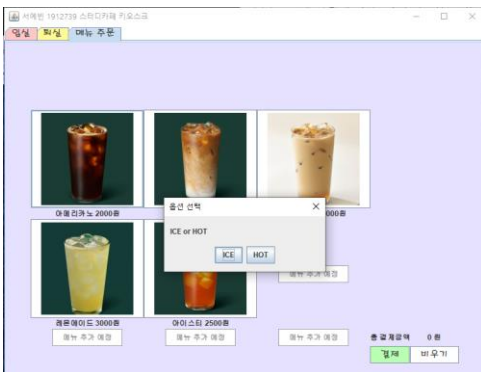
// 좌석(20번) 선택 후 확인 버튼 클릭



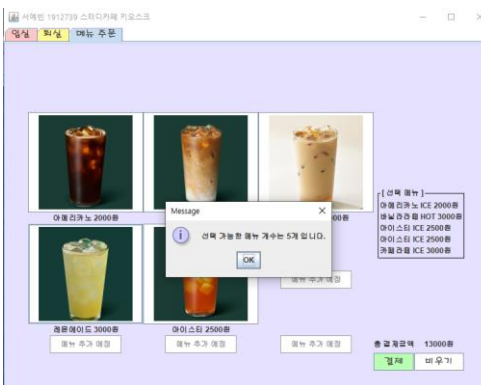
// 퇴실 탭에서 아이디, 비밀번호 입력



// Yes 클릭



//메뉴 주문 탭에서 음료 버튼 클릭



// 메뉴 5개를 담고 한 개 더 클릭 했을 때



//비우기 버튼 눌렀을 때.