

```

1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.core.content.ContextCompat;
5
6 import android.content.Intent;
7 import android.os.Bundle;
8 import android.text.Editable;
9 import android.text.TextWatcher;
10 import android.view.View;
11 import android.widget.Button;
12 import android.widget.EditText;
13 import android.widget.ImageButton;
14 import android.widget.ListView;
15 import android.widget.TextView;
16 import android.widget.Toast;
17
18 import com.android.volley.RequestQueue;
19 import com.android.volley.Response;
20 import com.android.volley.toolbox.Volley;
21
22 import org.json.JSONArray;
23 import org.json.JSONException;
24 import org.json.JSONObject;
25
26 /**
27 * 경매장 화면
28 * 메인 화면에서 계정을 선택 한 뒤 화면
29 * 경매장에서 아이템을 검색, 구매 할 아이템을 선택, 내 아이템 화면으로 이동, 판매중인 아이템 화면으로
30 * 이동 할 수 있게 하는 액티비티
31 */
32 public class AuctionScreen extends AppCompatActivity {
33     private Button moveSellingButton, moveMyitemButton, buyingbutton; // 판매중인
34     아이템 화면 버튼, 내 아이템 화면 버튼, 구매하기 버튼
35     private ImageButton backButton; // 뒤로가기 버튼
36     private TextView UserCharName; // 유저 게임 닉네임 텍스트뷰
37     private GameAccountInfo accountInfo; // 계정 정보에 대한 객체
38
39     ListView listview = null;
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.screen_auction);
44         ListViewAdapter adapter; // 어댑터 선언
45         Intent account_info = getIntent(); // 로그인 계정 정보 받기
46         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info"); // account_info라는 key로 GameAccountInfo 객체를 받아옴
47         String myGameName = accountInfo.getGameName(); // 게임 이름 정보 가져오기
48         String myCharName = accountInfo.getCharacterName(); // 게임 캐릭터 닉네임 정보
49         가져오기
50
51         // Adapter 생성
52         adapter = new ListViewAdapter() ;
53
54         // 리스트뷰 참조 및 Adapter달기
55         listview = (ListView) findViewById(R.id.auctionList);
56         listview.setAdapter(adapter);

```

```
File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/AuctionScreen.java

57
58         //경매장 리스트 정보를 DB에서 받아와 리스트뷰에 보여줌
59         Response.Listener<String> responseListener = new Response.Listener<String>
60     >() {
61             @Override
62             public void onResponse(String response) { // onResponse 함수 재정의
63                 try { // 만약 반응이 왔고 정상적이라면,
64                     JSONObject jsonResponse = new JSONObject(response); // json
65                     형태로 받기 위해 객체 준비
66                     JSONArray games = jsonResponse.getJSONArray("GameAuction");
67                     // GameAuction key 값으로 배열을 받음
68                     for (int i = 0; i < games.length(); i++) { // games의 길이만큼
69                         반복문 수행
70                         JSONObject item = games.getJSONObject(i); // games의
71                         데이터를 가져옴
72                         String UniNumber = item.getString("UniNum"); // UniNum
73                         데이터 저장
74                         String RegisterNumber = item.getString("RegisterNumber");
75                         ); // RegisterNumber 데이터 저장
76                         String ItemName = item.getString("ItemName"); // ItemName
77                         데이터 저장
78                         String Price = item.getString("Price"); // Price 데이터
79                         저장
80                         adapter.addAuctionItem(ContextCompat.getDrawable(
81                             AuctionScreen.this,
82                             R.drawable.ic_baseline_account_box_24), ItemName,
83                             Price, UniNumber, RegisterNumber); //리스트뷰에 있는 아이템의 이름, 가격, UniNumber,
84                             RegisterNumber 정보 저장
85                         }
86                         됨을 알림
87                     }
88                     adapter.notifyDataSetChanged(); // 반복문 수행 후에 리스트가 업데이트
89                 } catch (JSONException e) { // 접속 오류가 난 것이라면
90                     e.printStackTrace(); // 오류 추적
91                     // 토스트 메세지를 띄워줌
92                     Toast.makeText(getApplicationContext(), "no connection",
93                     Toast.LENGTH_SHORT).show();
94                 }
95             };
96             String purl = "http://ualsgur98.dothome.co.kr/GameAuction.php"; // 통신할
97             php 주소
98             // 데이터 전송을 위한 데이터 세팅
99             PHPRequest validateRequest = new PHPRequest( purl, myGameName,
100             responseListener);
101             RequestQueue queue = Volley.newRequestQueue(AuctionScreen.this); // 큐를
102             생성
103             queue.add(validateRequest); // 큐에 추가
104
105
106             //EditText(검색창)에 검색된 값을 받아 함수를 호출
107             EditText editTextFilter = (EditText)findViewById(R.id.editTextFilter) ;
108             editTextFilter.addTextChangedListener(new TextWatcher() { // 검색창 입력
109                 이벤트 생성
110                 @Override
111                 public void afterTextChanged(Editable edit) { // afterTextChanged 함수
112                     재정의
113                     String filterText = edit.toString() ; // 입력값 받기
114                     ((ListAdapter)listview.getAdapter()).getFilter().filter(
115                         filterText); //ListView를 통하지 않고 Adapter로부터 직접 Filter 객체의 참조를 가져와서
116                         filter() 함수를 호출 하여 텍스트 팝업이 안뜨게 함
117                 }
118             });
119         }
120     }
121 }
```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/AuctionScreen.java

```

96             }
97         @Override
98         public void beforeTextChanged(CharSequence s, int start, int count,
99             int after) {
100            }
101        @Override
102        public void onTextChanged(CharSequence s, int start, int before, int
103             count) {
104            }
105        });
106        //리스트 뷰에 있는 아이템을 터치하고 구매하기 버튼을 누르면 BuyingScreen 창 띄우기
107        buyingbutton = findViewById(R.id.buyButton);
108        buyingbutton.setOnClickListener(new View.OnClickListener() { // 구매하기
버튼 클릭 이벤트 생성
109        @Override
110        public void onClick(View view) { // onClick 함수 재정의
111            int pos = listview.getCheckedItemPosition(); // 리스트뷰 위치데이터
변수
112            if (pos > -1) { // pos는 0이상이므로 그 이상일 때만 작동
113                ListViewitem item = (ListViewitem) adapter.getItem(pos); //
터치한 아이템의 위치값 받아옴
114                String uninumber = item.getUninumber(); //아이템의 Uninumber
받아옴
115                String regnum = item.getRegnumber(); //아이템의 RegisterNumber
받아옴
116                // AuctionScreen에서 BuyingScreen으로 가는 intent 준비
117                Intent intent = new Intent(AuctionScreen.this, BuyingScreen.
class);
118                계정 정보 보내줌
119                intent.putExtra("account_info", accountInfo); // 구매하기 화면에
Uninumber 보내줌
120                intent.putExtra("myuninum", uninumber); // 구매하기 화면에
RegisterNumber 보내줌
121                intent.putExtra("myregnum", regnum); // 구매하기 화면에
122                startActivity(intent); // BuyingScreen 실행
123                finish(); // 현재 액티비티 종료
124            }
125        });
126    }
127    backButton = findViewById(R.id.back_button); // 뒤로가기 버튼 계정 선택 화면으로
이동
128    backButton.setOnClickListener(new View.OnClickListener() { //뒤로가기 버튼
클릭 이벤트 생성
129        @Override
130        public void onClick(View view) { // onClick 함수 재정의
131            // AuctionScreen에서 MainActivity으로 가는 intent 정의
132            Intent intent = new Intent(AuctionScreen.this, MainActivity.class
);
133            넣음
134            intent.putExtra("account_info", accountInfo); // intent에 데이터를
135            startActivity(intent); // intent 실행
136            finish(); // 현재 액티비티 종료
137        });
138    }
139    //판매중, 내 아이템 이동버튼 구현
140    moveSellingButton = (Button)findViewById(R.id.moveSellingButton); //

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/AuctionScreen.java

```
140 판매중인 아이템으로 이동버튼 생성
141     moveMyitemButton = (Button)findViewById(R.id.moveMyitemButton); // 내
아이템으로 이동버튼 생성
142
143     moveSellingButton.setOnClickListener(new View.OnClickListener(){ //
판매중인 아이템 이동 버튼 클릭 이벤트 생성
144         @Override
145         public void onClick(View v) { // onClick 함수 재정의
146             // AuctionScreen에서 SellingScreen으로 가는 intent 준비
147             Intent intent = new Intent(getApplicationContext(),
SellingItemScreen.class);
148             intent.putExtra("account_info", accountInfo); // intent에 게임 계정
데이터 추가
149             startActivity(intent); // SellingItemScreen 실행
150             finish(); // 현재 액티비티 종료
151         }
152     });
153     moveMyitemButton.setOnClickListener(new View.OnClickListener(){ //
내
아이템 이동 버튼 클릭 이벤트 생성
154         @Override
155         public void onClick(View v) {
156             // AuctionScreen에서 ItemCheckScreen으로 가는 intent 준비
157             Intent intent = new Intent(getApplicationContext(),
ItemCheckScreen.class);
158             intent.putExtra("account_info", accountInfo); // intent에 게임 계정
데이터 추가
159             startActivity(intent); // ItemCheckScreen 실행
160             finish(); // 현재 액티비티 종료
161         }
162     });
163
164 }
165 }
```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/BuyingScreen.java

```
1 package com.example.teamauction;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.view.MotionEvent;
6 import android.view.View;
7 import android.view.Window;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.TextView;
11 import android.widget.Toast;
12
13 import androidx.appcompat.app.AppCompatActivity;
14
15 import com.android.volley.RequestQueue;
16 import com.android.volley.Response;
17 import com.android.volley.toolbox.Volley;
18
19 import org.json.JSONArray;
20 import org.json.JSONException;
21 import org.json.JSONObject;
22
23 /**
24 * 구매하기 화면
25 * 경매장 화면에서 리스트뷰 목록중 구매할 아이템을 누르고 구매하기 버튼을 클릭한 뒤 화면
26 * 구매하기를 실질적으로 할 수 있게 하는 액티비티
27 */
28
29 public class BuyingScreen extends AppCompatActivity {
30     private EditText editTextquantity; //수량을 입력할 수 있는 에디트 텍스트 뷰
31     private GameAccountInfo accountInfo; // 계정 정보에 대한 객체
32     private TextView ItemName, ItemInfo, ItemPrice; // 아이템 이름, 정보, 가격이 들어갈
33     텍스트 뷰
34     Button yes_btn, no_btn; // 확인, 취소 버튼
35     private String DBItemName; // DB에서 받아온 아이템 이름
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39         setContentView(R.layout.screen_buying);
40
41         Intent intent = getIntent(); // 로그인 계정 정보 받기
42         accountInfo = (GameAccountInfo) intent.getSerializableExtra("account_info");
43     } // intent라는 key로 GameAccountInfo 객체를 받아옴
44         String Uninum = intent.getExtras().getString("myuninum"); //구매하려는 아이템의
45         Uninum 받아옴
46         String Regnum = intent.getExtras().getString("myregnum"); //구매하려는 아이템의
47         Regnum 받아옴
48
49         Intent account_info = getIntent(); // 로그인 계정 정보 받기
50         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info");
51         String myGameName = accountInfo.getGameName(); // 게임 이름 정보 가져오기
52         String myCharName = accountInfo.getCharacterName(); // 게임 캐릭터 닉네임 정보
53         가져오기
54
55         ItemName = findViewById(R.id.buyingItemName); //게임 아이템 이름 들어갈 공간 생성
56         ItemInfo = findViewById(R.id.buyingItemInfo); //게임 아이템 고유정보 들어갈 공간
57         생성
58         ItemPrice = findViewById(R.id.buyCostBox); //게임 아이템 가격정보 들어갈 공간 생성
```

```
File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/BuyingScreen.java
54         //구매하려는 아이템 정보를 DB에서 받아와 화면에 보여줌
55         Response.Listener<String> responseListener = new Response.Listener<String>()
56     >() {
57             @Override
58             public void onResponse(String response) { // onResponse 함수 재정의
59                 try { // 만약 반응이 왔고 정상적이라면
60                     JSONObject jsonResponse = new JSONObject(response); // json
61                     형태로 받기 위해 객체 준비
62                     boolean success = jsonResponse.getBoolean("success"); // //
63                     success라는 key 값으로 boolean value를 받음
64                     if (success) { // success가 참이라면
65                         String BuyItemName = jsonResponse.getString("ItemName");
66                     ); // ItemName 데이터 저장
67                     String BuyItemInfo = jsonResponse.getString("ItemInfo");
68                     // ItemInfo 데이터 저장
69                     ItemName.setText(BuyItemName); //구매하려는 아이템 이름으로 변경
70                     ItemInfo.setText(BuyItemInfo); //구매하려는 아이템 정보로 변경
71                     DBItemName = jsonResponse.getString("ItemName");// //
72                     ItemName 데이터 저장
73                     } else { // success가 false일 때
74                         // 토스트 메세지를 띄워줌
75                         Toast.makeText(getApplicationContext(), "서버와 연결이
76                         끊겼습니다", Toast.LENGTH_SHORT).show();
77                     return;
78                 }
79             } catch (JSONException e) { // 접속 오류가 난 것이라면
80                 e.printStackTrace(); // 오류 추적
81                 // 토스트 메세지를 띄워줌
82                 Toast.makeText(getApplicationContext(), "no connection",
83                 Toast.LENGTH_SHORT).show();
84             }
85         };
86         String purl = "http://ualsgur98.dothome.co.kr/BuyItemInfo.php"; // 통신할
87         php 주소
88         // 데이터 전송을 위한 데이터 세팅
89         PHPRequest validateRequest = new PHPRequest(purl, myGameName, Uninum,
90         responseListener);
91         RequestQueue queue = Volley.newRequestQueue(BuyingScreen.this); // 큐를
92         생성
93         queue.add(validateRequest); // 큐에 추가
94
95
96         //구매 버튼을 눌렀을때 값 저장, 내 아이템 화면으로 돌아감
97         Response.Listener<String> responseListener2 = new Response.Listener<
98         String>() { // 구매 버튼 클릭 이벤트 생성
99             @Override
100            public void onResponse(String response) {
101                try {
102                    JSONObject jsonResponse = new JSONObject(response); // json
103                    형태로 받기 위해 객체 준비
104                    boolean success = jsonResponse.getBoolean("success"); // //
105                    success라는 key 값으로 boolean value를 받음
106                    if (success) { // 서버 통신
107                        String cost = jsonResponse.getString("Price"); // Price
108                        데이터 저장
109                        ItemPrice.setText(cost); // DB에서 받아온 가격으로 변경
110                    } else { // success가 false일 때
111                        // 토스트 메세지를 띄워줌
112                    }
113                }
114            }
115        };
116    }
117
118    @Override
119    protected void onCreate(Bundle savedInstanceState) {
120        super.onCreate(savedInstanceState);
121        setContentView(R.layout.activity_buying_screen);
122
123        // 초기화
124        ItemName = findViewById(R.id.item_name);
125        ItemInfo = findViewById(R.id.item_info);
126        ItemPrice = findViewById(R.id.item_price);
127        ItemPrice.setHint("가격");
128
129        // 버튼
130        BuyButton = findViewById(R.id.buy_button);
131        BuyButton.setOnClickListener(this);
132
133        // 툴바
134        Toolbar toolbar = findViewById(R.id.toolbar);
135        setSupportActionBar(toolbar);
136        getSupportActionBar().setTitle("구매");
137        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
138
139        // 페더
140        DrawerLayout drawerLayout = findViewById(R.id.drawer_layout);
141        NavigationView navigationView = findViewById(R.id.nav_view);
142        navigationView.setNavigationItemSelectedListener(this);
143
144        // 페더 콘텐츠 설정
145        View headerView = navigationView.getHeaderView(0);
146        TextView textView = headerView.findViewById(R.id.nav_header_text);
147        textView.setText("Shin");
148
149        // 페더 콘텐츠 설정
150        View footerView = navigationView.getFooterView(0);
151        TextView textView2 = footerView.findViewById(R.id.nav_footer_text);
152        textView2.setText("Footer");
153
154        // 페더 콘텐츠 설정
155        View middleView = navigationView.getMiddleView(0);
156        TextView textView3 = middleView.findViewById(R.id.nav_middle_text);
157        textView3.setText("Middle");
158
159        // 페더 콘텐츠 설정
160        View bottomView = navigationView.getBottomView(0);
161        TextView textView4 = bottomView.findViewById(R.id.nav_bottom_text);
162        textView4.setText("Bottom");
163
164        // 페더 콘텐츠 설정
165        View topView = navigationView.getTopView(0);
166        TextView textView5 = topView.findViewById(R.id.nav_top_text);
167        textView5.setText("Top");
168
169        // 페더 콘텐츠 설정
170        View leftView = navigationView.getLeftView(0);
171        TextView textView6 = leftView.findViewById(R.id.nav_left_text);
172        textView6.setText("Left");
173
174        // 페더 콘텐츠 설정
175        View rightView = navigationView.getRightView(0);
176        TextView textView7 = rightView.findViewById(R.id.nav_right_text);
177        textView7.setText("Right");
178
179        // 페더 콘텐츠 설정
180        View centerView = navigationView.getCenterView(0);
181        TextView textView8 = centerView.findViewById(R.id.nav_center_text);
182        textView8.setText("Center");
183
184        // 페더 콘텐츠 설정
185        View bottomLeftView = navigationView.getBottomLeftView(0);
186        TextView textView9 = bottomLeftView.findViewById(R.id.nav_bottom_left_text);
187        textView9.setText("Bottom Left");
188
189        // 페더 콘텐츠 설정
190        View bottomRightView = navigationView.getBottomRightView(0);
191        TextView textView10 = bottomRightView.findViewById(R.id.nav_bottom_right_text);
192        textView10.setText("Bottom Right");
193
194        // 페더 콘텐츠 설정
195        View topLeftView = navigationView.getTopLeftView(0);
196        TextView textView11 = topLeftView.findViewById(R.id.nav_top_left_text);
197        textView11.setText("Top Left");
198
199        // 페더 콘텐츠 설정
200        View topRightView = navigationView.getTopRightView(0);
201        TextView textView12 = topRightView.findViewById(R.id.nav_top_right_text);
202        textView12.setText("Top Right");
203
204        // 페더 콘텐츠 설정
205        View middleLeftView = navigationView.getMiddleLeftView(0);
206        TextView textView13 = middleLeftView.findViewById(R.id.nav_middle_left_text);
207        textView13.setText("Middle Left");
208
209        // 페더 콘텐츠 설정
210        View middleRightView = navigationView.getMiddleRightView(0);
211        TextView textView14 = middleRightView.findViewById(R.id.nav_middle_right_text);
212        textView14.setText("Middle Right");
213
214        // 페더 콘텐츠 설정
215        View bottomCenterView = navigationView.getBottomCenterView(0);
216        TextView textView15 = bottomCenterView.findViewById(R.id.nav_bottom_center_text);
217        textView15.setText("Bottom Center");
218
219        // 페더 콘텐츠 설정
220        View topCenterView = navigationView.getTopCenterView(0);
221        TextView textView16 = topCenterView.findViewById(R.id.nav_top_center_text);
222        textView16.setText("Top Center");
223
224        // 페더 콘텐츠 설정
225        View leftCenterView = navigationView.getLeftCenterView(0);
226        TextView textView17 = leftCenterView.findViewById(R.id.nav_left_center_text);
227        textView17.setText("Left Center");
228
229        // 페더 콘텐츠 설정
230        View rightCenterView = navigationView.getRightCenterView(0);
231        TextView textView18 = rightCenterView.findViewById(R.id.nav_right_center_text);
232        textView18.setText("Right Center");
233
234        // 페더 콘텐츠 설정
235        View centerCenterView = navigationView.getCenterCenterView(0);
236        TextView textView19 = centerCenterView.findViewById(R.id.nav_center_center_text);
237        textView19.setText("Center Center");
238
239        // 페더 콘텐츠 설정
240        View bottomBottomView = navigationView.getBottomBottomView(0);
241        TextView textView20 = bottomBottomView.findViewById(R.id.nav_bottom_bottom_text);
242        textView20.setText("Bottom Bottom");
243
244        // 페더 콘텐츠 설정
245        View topBottomView = navigationView.getTopBottomView(0);
246        TextView textView21 = topBottomView.findViewById(R.id.nav_top_bottom_text);
247        textView21.setText("Top Bottom");
248
249        // 페더 콘텐츠 설정
250        View leftBottomView = navigationView.getLeftBottomView(0);
251        TextView textView22 = leftBottomView.findViewById(R.id.nav_left_bottom_text);
252        textView22.setText("Left Bottom");
253
254        // 페더 콘텐츠 설정
255        View rightBottomView = navigationView.getRightBottomView(0);
256        TextView textView23 = rightBottomView.findViewById(R.id.nav_right_bottom_text);
257        textView23.setText("Right Bottom");
258
259        // 페더 콘텐츠 설정
260        View centerBottomView = navigationView.getCenterBottomView(0);
261        TextView textView24 = centerBottomView.findViewById(R.id.nav_center_bottom_text);
262        textView24.setText("Center Bottom");
263
264        // 페더 콘텐츠 설정
265        View bottomLeftBottomView = navigationView.getBottomLeftBottomView(0);
266        TextView textView25 = bottomLeftBottomView.findViewById(R.id.nav_bottom_left_bottom_text);
267        textView25.setText("Bottom Left Bottom");
268
269        // 페더 콘텐츠 설정
270        View bottomRightBottomView = navigationView.getBottomRightBottomView(0);
271        TextView textView26 = bottomRightBottomView.findViewById(R.id.nav_bottom_right_bottom_text);
272        textView26.setText("Bottom Right Bottom");
273
274        // 페더 콘텐츠 설정
275        View topLeftBottomView = navigationView.getTopLeftBottomView(0);
276        TextView textView27 = topLeftBottomView.findViewById(R.id.nav_top_left_bottom_text);
277        textView27.setText("Top Left Bottom");
278
279        // 페더 콘텐츠 설정
280        View topRightBottomView = navigationView.getTopRightBottomView(0);
281        TextView textView28 = topRightBottomView.findViewById(R.id.nav_top_right_bottom_text);
282        textView28.setText("Top Right Bottom");
283
284        // 페더 콘텐츠 설정
285        View middleLeftBottomView = navigationView.getMiddleLeftBottomView(0);
286        TextView textView29 = middleLeftBottomView.findViewById(R.id.nav_middle_left_bottom_text);
287        textView29.setText("Middle Left Bottom");
288
289        // 페더 콘텐츠 설정
290        View middleRightBottomView = navigationView.getMiddleRightBottomView(0);
291        TextView textView30 = middleRightBottomView.findViewById(R.id.nav_middle_right_bottom_text);
292        textView30.setText("Middle Right Bottom");
293
294        // 페더 콘텐츠 설정
295        View bottomCenterBottomView = navigationView.getBottomCenterBottomView(0);
296        TextView textView31 = bottomCenterBottomView.findViewById(R.id.nav_bottom_center_bottom_text);
297        textView31.setText("Bottom Center Bottom");
298
299        // 페더 콘텐츠 설정
300        View topCenterBottomView = navigationView.getTopCenterBottomView(0);
301        TextView textView32 = topCenterBottomView.findViewById(R.id.nav_top_center_bottom_text);
302        textView32.setText("Top Center Bottom");
303
304        // 페더 콘텐츠 설정
305        View leftCenterBottomView = navigationView.getLeftCenterBottomView(0);
306        TextView textView33 = leftCenterBottomView.findViewById(R.id.nav_left_center_bottom_text);
307        textView33.setText("Left Center Bottom");
308
309        // 페더 콘텐츠 설정
310        View rightCenterBottomView = navigationView.getRightCenterBottomView(0);
311        TextView textView34 = rightCenterBottomView.findViewById(R.id.nav_right_center_bottom_text);
312        textView34.setText("Right Center Bottom");
313
314        // 페더 콘텐츠 설정
315        View centerCenterBottomView = navigationView.getCenterCenterBottomView(0);
316        TextView textView35 = centerCenterBottomView.findViewById(R.id.nav_center_center_bottom_text);
317        textView35.setText("Center Center Bottom");
318
319        // 페더 콘텐츠 설정
320        View bottomBottomBottomView = navigationView.getBottomBottomBottomView(0);
321        TextView textView36 = bottomBottomBottomView.findViewById(R.id.nav_bottom_bottom_bottom_text);
322        textView36.setText("Bottom Bottom Bottom");
323
324        // 페더 콘텐츠 설정
325        View topBottomBottomView = navigationView.getTopBottomBottomView(0);
326        TextView textView37 = topBottomBottomView.findViewById(R.id.nav_top_bottom_bottom_text);
327        textView37.setText("Top Bottom Bottom");
328
329        // 페더 콘텐츠 설정
330        View leftBottomBottomView = navigationView.getLeftBottomBottomView(0);
331        TextView textView38 = leftBottomBottomView.findViewById(R.id.nav_left_bottom_bottom_text);
332        textView38.setText("Left Bottom Bottom");
333
334        // 페더 콘텐츠 설정
335        View rightBottomBottomView = navigationView.getRightBottomBottomView(0);
336        TextView textView39 = rightBottomBottomView.findViewById(R.id.nav_right_bottom_bottom_text);
337        textView39.setText("Right Bottom Bottom");
338
339        // 페더 콘텐츠 설정
340        View centerBottomBottomView = navigationView.getCenterBottomBottomView(0);
341        TextView textView40 = centerBottomBottomView.findViewById(R.id.nav_center_bottom_bottom_text);
342        textView40.setText("Center Bottom Bottom");
343
344        // 페더 콘텐츠 설정
345        View bottomLeftBottomBottomView = navigationView.getBottomLeftBottomBottomView(0);
346        TextView textView41 = bottomLeftBottomBottomView.findViewById(R.id.nav_bottom_left_bottom_bottom_text);
347        textView41.setText("Bottom Left Bottom Bottom");
348
349        // 페더 콘텐츠 설정
350        View bottomRightBottomBottomView = navigationView.getBottomRightBottomBottomView(0);
351        TextView textView42 = bottomRightBottomBottomView.findViewById(R.id.nav_bottom_right_bottom_bottom_text);
352        textView42.setText("Bottom Right Bottom Bottom");
353
354        // 페더 콘텐츠 설정
355        View topLeftBottomBottomView = navigationView.getTopLeftBottomBottomView(0);
356        TextView textView43 = topLeftBottomBottomView.findViewById(R.id.nav_top_left_bottom_bottom_text);
357        textView43.setText("Top Left Bottom Bottom");
358
359        // 페더 콘텐츠 설정
360        View topRightBottomBottomView = navigationView.getTopRightBottomBottomView(0);
361        TextView textView44 = topRightBottomBottomView.findViewById(R.id.nav_top_right_bottom_bottom_text);
362        textView44.setText("Top Right Bottom Bottom");
363
364        // 페더 콘텐츠 설정
365        View middleLeftBottomBottomView = navigationView.getMiddleLeftBottomBottomView(0);
366        TextView textView45 = middleLeftBottomBottomView.findViewById(R.id.nav_middle_left_bottom_bottom_text);
367        textView45.setText("Middle Left Bottom Bottom");
368
369        // 페더 콘텐츠 설정
370        View middleRightBottomBottomView = navigationView.getMiddleRightBottomBottomView(0);
371        TextView textView46 = middleRightBottomBottomView.findViewById(R.id.nav_middle_right_bottom_bottom_text);
372        textView46.setText("Middle Right Bottom Bottom");
373
374        // 페더 콘텐츠 설정
375        View bottomCenterBottomBottomView = navigationView.getBottomCenterBottomBottomView(0);
376        TextView textView47 = bottomCenterBottomBottomView.findViewById(R.id.nav_bottom_center_bottom_bottom_text);
377        textView47.setText("Bottom Center Bottom Bottom");
378
379        // 페더 콘텐츠 설정
380        View topCenterBottomBottomView = navigationView.getTopCenterBottomBottomView(0);
381        TextView textView48 = topCenterBottomBottomView.findViewById(R.id.nav_top_center_bottom_bottom_text);
382        textView48.setText("Top Center Bottom Bottom");
383
384        // 페더 콘텐츠 설정
385        View leftCenterBottomBottomView = navigationView.getLeftCenterBottomBottomView(0);
386        TextView textView49 = leftCenterBottomBottomView.findViewById(R.id.nav_left_center_bottom_bottom_text);
387        textView49.setText("Left Center Bottom Bottom");
388
389        // 페더 콘텐츠 설정
390        View rightCenterBottomBottomView = navigationView.getRightCenterBottomBottomView(0);
391        TextView textView50 = rightCenterBottomBottomView.findViewById(R.id.nav_right_center_bottom_bottom_text);
392        textView50.setText("Right Center Bottom Bottom");
393
394        // 페더 콘텐츠 설정
395        View centerCenterBottomBottomView = navigationView.getCenterCenterBottomBottomView(0);
396        TextView textView51 = centerCenterBottomBottomView.findViewById(R.id.nav_center_center_bottom_bottom_text);
397        textView51.setText("Center Center Bottom Bottom");
398
399        // 페더 콘텐츠 설정
400        View bottomBottomBottomBottomView = navigationView.getBottomBottomBottomBottomView(0);
401        TextView textView52 = bottomBottomBottomBottomView.findViewById(R.id.nav_bottom_bottom_bottom_bottom_text);
402        textView52.setText("Bottom Bottom Bottom Bottom");
403
404        // 페더 콘텐츠 설정
405        View topBottomBottomBottomView = navigationView.getTopBottomBottomBottomView(0);
406        TextView textView53 = topBottomBottomBottomView.findViewById(R.id.nav_top_bottom_bottom_bottom_text);
407        textView53.setText("Top Bottom Bottom Bottom");
408
409        // 페더 콘텐츠 설정
410        View leftBottomBottomBottomView = navigationView.getLeftBottomBottomBottomView(0);
411        TextView textView54 = leftBottomBottomBottomView.findViewById(R.id.nav_left_bottom_bottom_bottom_text);
412        textView54.setText("Left Bottom Bottom Bottom");
413
414        // 페더 콘텐츠 설정
415        View rightBottomBottomBottomView = navigationView.getRightBottomBottomBottomView(0);
416        TextView textView55 = rightBottomBottomBottomView.findViewById(R.id.nav_right_bottom_bottom_bottom_text);
417        textView55.setText("Right Bottom Bottom Bottom");
418
419        // 페더 콘텐츠 설정
420        View centerBottomBottomBottomView = navigationView.getCenterBottomBottomBottomView(0);
421        TextView textView56 = centerBottomBottomBottomView.findViewById(R.id.nav_center_bottom_bottom_bottom_text);
422        textView56.setText("Center Bottom Bottom Bottom");
423
424        // 페더 콘텐츠 설정
425        View bottomLeftBottomBottomBottomView = navigationView.getBottomLeftBottomBottomBottomView(0);
426        TextView textView57 = bottomLeftBottomBottomBottomView.findViewById(R.id.nav_bottom_left_bottom_bottom_bottom_text);
427        textView57.setText("Bottom Left Bottom Bottom Bottom");
428
429        // 페더 콘텐츠 설정
430        View bottomRightBottomBottomBottomView = navigationView.getBottomRightBottomBottomBottomView(0);
431        TextView textView58 = bottomRightBottomBottomBottomView.findViewById(R.id.nav_bottom_right_bottom_bottom_bottom_text);
432        textView58.setText("Bottom Right Bottom Bottom Bottom");
433
434        // 페더 콘텐츠 설정
435        View topLeftBottomBottomBottomView = navigationView.getTopLeftBottomBottomBottomView(0);
436        TextView textView59 = topLeftBottomBottomBottomView.findViewById(R.id.nav_top_left_bottom_bottom_bottom_text);
437        textView59.setText("Top Left Bottom Bottom Bottom");
438
439        // 페더 콘텐츠 설정
440        View topRightBottomBottomBottomView = navigationView.getTopRightBottomBottomBottomView(0);
441        TextView textView60 = topRightBottomBottomBottomView.findViewById(R.id.nav_top_right_bottom_bottom_bottom_text);
442        textView60.setText("Top Right Bottom Bottom Bottom");
443
444        // 페더 콘텐츠 설정
445        View middleLeftBottomBottomBottomView = navigationView.getMiddleLeftBottomBottomBottomView(0);
446        TextView textView61 = middleLeftBottomBottomBottomView.findViewById(R.id.nav_middle_left_bottom_bottom_bottom_text);
447        textView61.setText("Middle Left Bottom Bottom Bottom");
448
449        // 페더 콘텐츠 설정
450        View middleRightBottomBottomBottomView = navigationView.getMiddleRightBottomBottomBottomView(0);
451        TextView textView62 = middleRightBottomBottomBottomView.findViewById(R.id.nav_middle_right_bottom_bottom_bottom_text);
452        textView62.setText("Middle Right Bottom Bottom Bottom");
453
454        // 페더 콘텐츠 설정
455        View bottomCenterBottomBottomBottomView = navigationView.getBottomCenterBottomBottomBottomView(0);
456        TextView textView63 = bottomCenterBottomBottomBottomView.findViewById(R.id.nav_bottom_center_bottom_bottom_bottom_text);
457        textView63.setText("Bottom Center Bottom Bottom Bottom");
458
459        // 페더 콘텐츠 설정
460        View topCenterBottomBottomBottomView = navigationView.getTopCenterBottomBottomBottomView(0);
461        TextView textView64 = topCenterBottomBottomBottomView.findViewById(R.id.nav_top_center_bottom_bottom_bottom_text);
462        textView64.setText("Top Center Bottom Bottom Bottom");
463
464        // 페더 콘텐츠 설정
465        View leftCenterBottomBottomBottomView = navigationView.getLeftCenterBottomBottomBottomView(0);
466        TextView textView65 = leftCenterBottomBottomBottomView.findViewById(R.id.nav_left_center_bottom_bottom_bottom_text);
467        textView65.setText("Left Center Bottom Bottom Bottom");
468
469        // 페더 콘텐츠 설정
470        View rightCenterBottomBottomBottomView = navigationView.getRightCenterBottomBottomBottomView(0);
471        TextView textView66 = rightCenterBottomBottomBottomView.findViewById(R.id.nav_right_center_bottom_bottom_bottom_text);
472        textView66.setText("Right Center Bottom Bottom Bottom");
473
474        // 페더 콘텐츠 설정
475        View centerCenterBottomBottomBottomView = navigationView.getCenterCenterBottomBottomBottomView(0);
476        TextView textView67 = centerCenterBottomBottomBottomView.findViewById(R.id.nav_center_center_bottom_bottom_bottom_text);
477        textView67.setText("Center Center Bottom Bottom Bottom");
478
479        // 페더 콘텐츠 설정
480        View bottomBottomBottomBottomBottomView = navigationView.getBottomBottomBottomBottomBottomView(0);
481        TextView textView68 = bottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_bottom_bottom_bottom_bottom_text);
482        textView68.setText("Bottom Bottom Bottom Bottom Bottom");
483
484        // 페더 콘텐츠 설정
485        View topBottomBottomBottomBottomView = navigationView.getTopBottomBottomBottomBottomView(0);
486        TextView textView69 = topBottomBottomBottomBottomView.findViewById(R.id.nav_top_bottom_bottom_bottom_bottom_text);
487        textView69.setText("Top Bottom Bottom Bottom Bottom");
488
489        // 페더 콘텐츠 설정
490        View leftBottomBottomBottomBottomView = navigationView.getLeftBottomBottomBottomBottomView(0);
491        TextView textView70 = leftBottomBottomBottomBottomView.findViewById(R.id.nav_left_bottom_bottom_bottom_bottom_text);
492        textView70.setText("Left Bottom Bottom Bottom Bottom");
493
494        // 페더 콘텐츠 설정
495        View rightBottomBottomBottomBottomView = navigationView.getRightBottomBottomBottomBottomView(0);
496        TextView textView71 = rightBottomBottomBottomBottomView.findViewById(R.id.nav_right_bottom_bottom_bottom_bottom_text);
497        textView71.setText("Right Bottom Bottom Bottom Bottom");
498
499        // 페더 콘텐츠 설정
500        View centerBottomBottomBottomBottomView = navigationView.getCenterBottomBottomBottomBottomView(0);
501        TextView textView72 = centerBottomBottomBottomBottomView.findViewById(R.id.nav_center_bottom_bottom_bottom_bottom_text);
502        textView72.setText("Center Bottom Bottom Bottom Bottom");
503
504        // 페더 콘텐츠 설정
505        View bottomLeftBottomBottomBottomBottomView = navigationView.getBottomLeftBottomBottomBottomBottomView(0);
506        TextView textView73 = bottomLeftBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_left_bottom_bottom_bottom_bottom_text);
507        textView73.setText("Bottom Left Bottom Bottom Bottom Bottom");
508
509        // 페더 콘텐츠 설정
510        View bottomRightBottomBottomBottomBottomView = navigationView.getBottomRightBottomBottomBottomBottomView(0);
511        TextView textView74 = bottomRightBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_right_bottom_bottom_bottom_bottom_text);
512        textView74.setText("Bottom Right Bottom Bottom Bottom Bottom");
513
514        // 페더 콘텐츠 설정
515        View topLeftBottomBottomBottomBottomView = navigationView.getTopLeftBottomBottomBottomBottomView(0);
516        TextView textView75 = topLeftBottomBottomBottomBottomView.findViewById(R.id.nav_top_left_bottom_bottom_bottom_bottom_text);
517        textView75.setText("Top Left Bottom Bottom Bottom Bottom");
518
519        // 페더 콘텐츠 설정
520        View topRightBottomBottomBottomBottomView = navigationView.getTopRightBottomBottomBottomBottomView(0);
521        TextView textView76 = topRightBottomBottomBottomBottomView.findViewById(R.id.nav_top_right_bottom_bottom_bottom_bottom_text);
522        textView76.setText("Top Right Bottom Bottom Bottom Bottom");
523
524        // 페더 콘텐츠 설정
525        View middleLeftBottomBottomBottomBottomView = navigationView.getMiddleLeftBottomBottomBottomBottomView(0);
526        TextView textView77 = middleLeftBottomBottomBottomBottomView.findViewById(R.id.nav_middle_left_bottom_bottom_bottom_bottom_text);
527        textView77.setText("Middle Left Bottom Bottom Bottom Bottom");
528
529        // 페더 콘텐츠 설정
530        View middleRightBottomBottomBottomBottomView = navigationView.getMiddleRightBottomBottomBottomBottomView(0);
531        TextView textView78 = middleRightBottomBottomBottomBottomView.findViewById(R.id.nav_middle_right_bottom_bottom_bottom_bottom_text);
532        textView78.setText("Middle Right Bottom Bottom Bottom Bottom");
533
534        // 페더 콘텐츠 설정
535        View bottomCenterBottomBottomBottomBottomView = navigationView.getBottomCenterBottomBottomBottomBottomView(0);
536        TextView textView79 = bottomCenterBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_center_bottom_bottom_bottom_bottom_text);
537        textView79.setText("Bottom Center Bottom Bottom Bottom Bottom");
538
539        // 페더 콘텐츠 설정
540        View topCenterBottomBottomBottomBottomView = navigationView.getTopCenterBottomBottomBottomBottomView(0);
541        TextView textView80 = topCenterBottomBottomBottomBottomView.findViewById(R.id.nav_top_center_bottom_bottom_bottom_bottom_text);
542        textView80.setText("Top Center Bottom Bottom Bottom Bottom");
543
544        // 페더 콘텐츠 설정
545        View leftCenterBottomBottomBottomBottomView = navigationView.getLeftCenterBottomBottomBottomBottomView(0);
546        TextView textView81 = leftCenterBottomBottomBottomBottomView.findViewById(R.id.nav_left_center_bottom_bottom_bottom_bottom_text);
547        textView81.setText("Left Center Bottom Bottom Bottom Bottom");
548
549        // 페더 콘텐츠 설정
550        View rightCenterBottomBottomBottomBottomView = navigationView.getRightCenterBottomBottomBottomBottomView(0);
551        TextView textView82 = rightCenterBottomBottomBottomBottomView.findViewById(R.id.nav_right_center_bottom_bottom_bottom_bottom_text);
552        textView82.setText("Right Center Bottom Bottom Bottom Bottom");
553
554        // 페더 콘텐츠 설정
555        View centerCenterBottomBottomBottomBottomView = navigationView.getCenterCenterBottomBottomBottomBottomView(0);
556        TextView textView83 = centerCenterBottomBottomBottomBottomView.findViewById(R.id.nav_center_center_bottom_bottom_bottom_bottom_text);
557        textView83.setText("Center Center Bottom Bottom Bottom Bottom");
558
559        // 페더 콘텐츠 설정
560        View bottomBottomBottomBottomBottomBottomView = navigationView.getBottomBottomBottomBottomBottomBottomView(0);
561        TextView textView84 = bottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_bottom_bottom_bottom_bottom_bottom_text);
562        textView84.setText("Bottom Bottom Bottom Bottom Bottom Bottom");
563
564        // 페더 콘텐츠 설정
565        View topBottomBottomBottomBottomBottomView = navigationView.getTopBottomBottomBottomBottomBottomView(0);
566        TextView textView85 = topBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_bottom_bottom_bottom_bottom_bottom_text);
567        textView85.setText("Top Bottom Bottom Bottom Bottom Bottom");
568
569        // 페더 콘텐츠 설정
570        View leftBottomBottomBottomBottomBottomView = navigationView.getLeftBottomBottomBottomBottomBottomView(0);
571        TextView textView86 = leftBottomBottomBottomBottomBottomView.findViewById(R.id.nav_left_bottom_bottom_bottom_bottom_bottom_text);
572        textView86.setText("Left Bottom Bottom Bottom Bottom Bottom");
573
574        // 페더 콘텐츠 설정
575        View rightBottomBottomBottomBottomBottomView = navigationView.getRightBottomBottomBottomBottomBottomView(0);
576        TextView textView87 = rightBottomBottomBottomBottomBottomView.findViewById(R.id.nav_right_bottom_bottom_bottom_bottom_bottom_text);
577        textView87.setText("Right Bottom Bottom Bottom Bottom Bottom");
578
579        // 페더 콘텐츠 설정
580        View centerBottomBottomBottomBottomBottomView = navigationView.getCenterBottomBottomBottomBottomBottomView(0);
581        TextView textView88 = centerBottomBottomBottomBottomBottomView.findViewById(R.id.nav_center_bottom_bottom_bottom_bottom_bottom_text);
582        textView88.setText("Center Bottom Bottom Bottom Bottom Bottom");
583
584        // 페더 콘텐츠 설정
585        View bottomLeftBottomBottomBottomBottomBottomView = navigationView.getBottomLeftBottomBottomBottomBottomBottomView(0);
586        TextView textView89 = bottomLeftBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_left_bottom_bottom_bottom_bottom_bottom_text);
587        textView89.setText("Bottom Left Bottom Bottom Bottom Bottom Bottom");
588
589        // 페더 콘텐츠 설정
590        View bottomRightBottomBottomBottomBottomBottomView = navigationView.getBottomRightBottomBottomBottomBottomBottomView(0);
591        TextView textView90 = bottomRightBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_right_bottom_bottom_bottom_bottom_bottom_text);
592        textView90.setText("Bottom Right Bottom Bottom Bottom Bottom Bottom");
593
594        // 페더 콘텐츠 설정
595        View topLeftBottomBottomBottomBottomBottomView = navigationView.getTopLeftBottomBottomBottomBottomBottomView(0);
596        TextView textView91 = topLeftBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_left_bottom_bottom_bottom_bottom_bottom_text);
597        textView91.setText("Top Left Bottom Bottom Bottom Bottom Bottom");
598
599        // 페더 콘텐츠 설정
600        View topRightBottomBottomBottomBottomBottomView = navigationView.getTopRightBottomBottomBottomBottomBottomView(0);
601        TextView textView92 = topRightBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_right_bottom_bottom_bottom_bottom_bottom_text);
602        textView92.setText("Top Right Bottom Bottom Bottom Bottom Bottom");
603
604        // 페더 콘텐츠 설정
605        View middleLeftBottomBottomBottomBottomBottomView = navigationView.getMiddleLeftBottomBottomBottomBottomBottomView(0);
606        TextView textView93 = middleLeftBottomBottomBottomBottomBottomView.findViewById(R.id.nav_middle_left_bottom_bottom_bottom_bottom_bottom_text);
607        textView93.setText("Middle Left Bottom Bottom Bottom Bottom Bottom");
608
609        // 페더 콘텐츠 설정
610        View middleRightBottomBottomBottomBottomBottomView = navigationView.getMiddleRightBottomBottomBottomBottomBottomView(0);
611        TextView textView94 = middleRightBottomBottomBottomBottomBottomView.findViewById(R.id.nav_middle_right_bottom_bottom_bottom_bottom_bottom_text);
612        textView94.setText("Middle Right Bottom Bottom Bottom Bottom Bottom");
613
614        // 페더 콘텐츠 설정
615        View bottomCenterBottomBottomBottomBottomBottomView = navigationView.getBottomCenterBottomBottomBottomBottomBottomView(0);
616        TextView textView95 = bottomCenterBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_center_bottom_bottom_bottom_bottom_bottom_text);
617        textView95.setText("Bottom Center Bottom Bottom Bottom Bottom Bottom");
618
619        // 페더 콘텐츠 설정
620        View topCenterBottomBottomBottomBottomBottomView = navigationView.getTopCenterBottomBottomBottomBottomBottomView(0);
621        TextView textView96 = topCenterBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_center_bottom_bottom_bottom_bottom_bottom_text);
622        textView96.setText("Top Center Bottom Bottom Bottom Bottom Bottom");
623
624        // 페더 콘텐츠 설정
625        View leftCenterBottomBottomBottomBottomBottomView = navigationView.getLeftCenterBottomBottomBottomBottomBottomView(0);
626        TextView textView97 = leftCenterBottomBottomBottomBottomBottomView.findViewById(R.id.nav_left_center_bottom_bottom_bottom_bottom_bottom_text);
627        textView97.setText("Left Center Bottom Bottom Bottom Bottom Bottom");
628
629        // 페더 콘텐츠 설정
630        View rightCenterBottomBottomBottomBottomBottomView = navigationView.getRightCenterBottomBottomBottomBottomBottomView(0);
631        TextView textView98 = rightCenterBottomBottomBottomBottomBottomView.findViewById(R.id.nav_right_center_bottom_bottom_bottom_bottom_bottom_text);
632        textView98.setText("Right Center Bottom Bottom Bottom Bottom Bottom");
633
634        // 페더 콘텐츠 설정
635        View centerCenterBottomBottomBottomBottomBottomView = navigationView.getCenterCenterBottomBottomBottomBottomBottomView(0);
636        TextView textView99 = centerCenterBottomBottomBottomBottomBottomView.findViewById(R.id.nav_center_center_bottom_bottom_bottom_bottom_bottom_text);
637        textView99.setText("Center Center Bottom Bottom Bottom Bottom Bottom");
638
639        // 페더 콘텐츠 설정
640        View bottomBottomBottomBottomBottomBottomBottomView = navigationView.getBottomBottomBottomBottomBottomBottomBottomView(0);
641        TextView textView100 = bottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
642        textView100.setText("Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
643
644        // 페더 콘텐츠 설정
645        View topBottomBottomBottomBottomBottomBottomView = navigationView.getTopBottomBottomBottomBottomBottomBottomView(0);
646        TextView textView101 = topBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_bottom_bottom_bottom_bottom_bottom_bottom_text);
647        textView101.setText("Top Bottom Bottom Bottom Bottom Bottom Bottom");
648
649        // 페더 콘텐츠 설정
650        View leftBottomBottomBottomBottomBottomBottomView = navigationView.getLeftBottomBottomBottomBottomBottomBottomView(0);
651        TextView textView102 = leftBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_left_bottom_bottom_bottom_bottom_bottom_bottom_text);
652        textView102.setText("Left Bottom Bottom Bottom Bottom Bottom Bottom");
653
654        // 페더 콘텐츠 설정
655        View rightBottomBottomBottomBottomBottomBottomView = navigationView.getRightBottomBottomBottomBottomBottomBottomView(0);
656        TextView textView103 = rightBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_right_bottom_bottom_bottom_bottom_bottom_bottom_text);
657        textView103.setText("Right Bottom Bottom Bottom Bottom Bottom Bottom");
658
659        // 페더 콘텐츠 설정
660        View centerBottomBottomBottomBottomBottomBottomView = navigationView.getCenterBottomBottomBottomBottomBottomBottomView(0);
661        TextView textView104 = centerBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_center_bottom_bottom_bottom_bottom_bottom_bottom_text);
662        textView104.setText("Center Bottom Bottom Bottom Bottom Bottom Bottom");
663
664        // 페더 콘텐츠 설정
665        View bottomLeftBottomBottomBottomBottomBottomBottomView = navigationView.getBottomLeftBottomBottomBottomBottomBottomBottomView(0);
666        TextView textView105 = bottomLeftBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_left_bottom_bottom_bottom_bottom_bottom_bottom_text);
667        textView105.setText("Bottom Left Bottom Bottom Bottom Bottom Bottom Bottom");
668
669        // 페더 콘텐츠 설정
670        View bottomRightBottomBottomBottomBottomBottomBottomView = navigationView.getBottomRightBottomBottomBottomBottomBottomBottomView(0);
671        TextView textView106 = bottomRightBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_right_bottom_bottom_bottom_bottom_bottom_bottom_text);
672        textView106.setText("Bottom Right Bottom Bottom Bottom Bottom Bottom Bottom");
673
674        // 페더 콘텐츠 설정
675        View topLeftBottomBottomBottomBottomBottomBottomView = navigationView.getTopLeftBottomBottomBottomBottomBottomBottomView(0);
676        TextView textView107 = topLeftBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_left_bottom_bottom_bottom_bottom_bottom_bottom_text);
677        textView107.setText("Top Left Bottom Bottom Bottom Bottom Bottom Bottom");
678
679        // 페더 콘텐츠 설정
680        View topRightBottomBottomBottomBottomBottomBottomView = navigationView.getTopRightBottomBottomBottomBottomBottomBottomView(0);
681        TextView textView108 = topRightBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_right_bottom_bottom_bottom_bottom_bottom_bottom_text);
682        textView108.setText("Top Right Bottom Bottom Bottom Bottom Bottom Bottom");
683
684        // 페더 콘텐츠 설정
685        View middleLeftBottomBottomBottomBottomBottomBottomView = navigationView.getMiddleLeftBottomBottomBottomBottomBottomBottomView(0);
686        TextView textView109 = middleLeftBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_middle_left_bottom_bottom_bottom_bottom_bottom_bottom_text);
687        textView109.setText("Middle Left Bottom Bottom Bottom Bottom Bottom Bottom");
688
689        // 페더 콘텐츠 설정
690        View middleRightBottomBottomBottomBottomBottomBottomView = navigationView.getMiddleRightBottomBottomBottomBottomBottomBottomView(0);
691        TextView textView110 = middleRightBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_middle_right_bottom_bottom_bottom_bottom_bottom_bottom_text);
692        textView110.setText("Middle Right Bottom Bottom Bottom Bottom Bottom Bottom");
693
694        // 페더 콘텐츠 설정
695        View bottomCenterBottomBottomBottomBottomBottomBottomView = navigationView.getBottomCenterBottomBottomBottomBottomBottomBottomView(0);
696        TextView textView111 = bottomCenterBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_center_bottom_bottom_bottom_bottom_bottom_bottom_text);
697        textView111.setText("Bottom Center Bottom Bottom Bottom Bottom Bottom Bottom");
698
699        // 페더 콘텐츠 설정
700        View topCenterBottomBottomBottomBottomBottomBottomView = navigationView.getTopCenterBottomBottomBottomBottomBottomBottomView(0);
701        TextView textView112 = topCenterBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_center_bottom_bottom_bottom_bottom_bottom_bottom_text);
702        textView112.setText("Top Center Bottom Bottom Bottom Bottom Bottom Bottom");
703
704        // 페더 콘텐츠 설정
705        View leftCenterBottomBottomBottomBottomBottomBottomView = navigationView.getLeftCenterBottomBottomBottomBottomBottomBottomView(0);
706        TextView textView113 = leftCenterBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_left_center_bottom_bottom_bottom_bottom_bottom_bottom_text);
707        textView113.setText("Left Center Bottom Bottom Bottom Bottom Bottom Bottom");
708
709        // 페더 콘텐츠 설정
710        View rightCenterBottomBottomBottomBottomBottomBottomView = navigationView.getRightCenterBottomBottomBottomBottomBottomBottomView(0);
711        TextView textView114 = rightCenterBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_right_center_bottom_bottom_bottom_bottom_bottom_bottom_text);
712        textView114.setText("Right Center Bottom Bottom Bottom Bottom Bottom Bottom");
713
714        // 페더 콘텐츠 설정
715        View centerCenterBottomBottomBottomBottomBottomBottomView = navigationView.getCenterCenterBottomBottomBottomBottomBottomBottomView(0);
716        TextView textView115 = centerCenterBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_center_center_bottom_bottom_bottom_bottom_bottom_bottom_text);
717        textView115.setText("Center Center Bottom Bottom Bottom Bottom Bottom Bottom");
718
719        // 페더 콘텐츠 설정
720        View bottomBottomBottomBottomBottomBottomBottomBottomView = navigationView.getBottomBottomBottomBottomBottomBottomBottomBottomView(0);
721        TextView textView116 = bottomBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
722        textView116.setText("Bottom Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
723
724        // 페더 콘텐츠 설정
725        View topBottomBottomBottomBottomBottomBottomBottomView = navigationView.getTopBottomBottomBottomBottomBottomBottomBottomView(0);
726        TextView textView117 = topBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
727        textView117.setText("Top Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
728
729        // 페더 콘텐츠 설정
730        View leftBottomBottomBottomBottomBottomBottomBottomView = navigationView.getLeftBottomBottomBottomBottomBottomBottomBottomView(0);
731        TextView textView118 = leftBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_left_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
732        textView118.setText("Left Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
733
734        // 페더 콘텐츠 설정
735        View rightBottomBottomBottomBottomBottomBottomBottomView = navigationView.getRightBottomBottomBottomBottomBottomBottomBottomView(0);
736        TextView textView119 = rightBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_right_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
737        textView119.setText("Right Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
738
739        // 페더 콘텐츠 설정
740        View centerBottomBottomBottomBottomBottomBottomBottomView = navigationView.getCenterBottomBottomBottomBottomBottomBottomBottomView(0);
741        TextView textView120 = centerBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_center_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
742        textView120.setText("Center Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
743
744        // 페더 콘텐츠 설정
745        View bottomLeftBottomBottomBottomBottomBottomBottomBottomView = navigationView.getBottomLeftBottomBottomBottomBottomBottomBottomBottomView(0);
746        TextView textView121 = bottomLeftBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_left_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
747        textView121.setText("Bottom Left Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
748
749        // 페더 콘텐츠 설정
750        View bottomRightBottomBottomBottomBottomBottomBottomBottomView = navigationView.getBottomRightBottomBottomBottomBottomBottomBottomBottomView(0);
751        TextView textView122 = bottomRightBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_right_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
752        textView122.setText("Bottom Right Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
753
754        // 페더 콘텐츠 설정
755        View topLeftBottomBottomBottomBottomBottomBottomBottomView = navigationView.getTopLeftBottomBottomBottomBottomBottomBottomBottomView(0);
756        TextView textView123 = topLeftBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_left_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
757        textView123.setText("Top Left Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
758
759        // 페더 콘텐츠 설정
760        View topRightBottomBottomBottomBottomBottomBottomBottomView = navigationView.getTopRightBottomBottomBottomBottomBottomBottomBottomView(0);
761        TextView textView124 = topRightBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_right_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
762        textView124.setText("Top Right Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
763
764        // 페더 콘텐츠 설정
765        View middleLeftBottomBottomBottomBottomBottomBottomBottomView = navigationView.getMiddleLeftBottomBottomBottomBottomBottomBottomBottomView(0);
766        TextView textView125 = middleLeftBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_middle_left_bottom_bottom_bottom_bottom_bottom_bottom_bottom_text);
767        textView125.setText("Middle Left Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
768
769        // 페더 콘텐츠 설정
770        View middleRightBottomBottomBottomBottomBottomBottomBottomView = navigationView.getMiddleRightBottomBottomBottomBottomBottomBottomBottomView(0);
771        TextView textView126 = middleRightBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_middle_right_bottom_bottom_bottom_bottom_bottom_bottom_text);
772        textView126.setText("Middle Right Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
773
774        // 페더 콘텐츠 설정
775        View bottomCenterBottomBottomBottomBottomBottomBottomBottomView = navigationView.getBottomCenterBottomBottomBottomBottomBottomBottomBottomView(0);
776        TextView textView127 = bottomCenterBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_bottom_center_bottom_bottom_bottom_bottom_bottom_bottom_text);
777        textView127.setText("Bottom Center Bottom Bottom Bottom Bottom Bottom Bottom Bottom");
778
779        // 페더 콘텐츠 설정
780        View topCenterBottomBottomBottomBottomBottomBottomBottomView = navigationView.getTopCenterBottomBottomBottomBottomBottomBottomBottomView(0);
781        TextView textView128 = topCenterBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_top_center_bottom_bottom_bottom_bottom_bottom_text);
782        textView128.setText("Top Center Bottom Bottom Bottom Bottom Bottom Bottom");
783
784        // 페더 콘텐츠 설정
785        View leftCenterBottomBottomBottomBottomBottomBottomBottomView = navigationView.getLeftCenterBottomBottomBottomBottomBottomBottomBottomView(0);
786        TextView textView129 = leftCenterBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_left_center_bottom_bottom_bottom_bottom_bottom_text);
787        textView129.setText("Left Center Bottom Bottom Bottom Bottom Bottom");
788
789        // 페더 콘텐츠 설정
790        View rightCenterBottomBottomBottomBottomBottomBottomBottomView = navigationView.getRightCenterBottomBottomBottomBottomBottomBottomBottomView(0);
791        TextView textView130 = rightCenterBottomBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_right_center_bottom_bottom_bottom_bottom_text);
792        textView130.setText("Right Center Bottom Bottom Bottom");
793
794        // 페더 콘텐츠 설정
795        View centerCenterBottomBottomBottomBottomBottomBottomBottomView = navigationView.getCenterCenterBottomBottomBottomBottomBottomBottomView(0);
796        TextView textView131 = centerCenterBottomBottomBottomBottomBottomBottomView.findViewById(R.id.nav_center_center_bottom_bottom_bottom_bottom_text);
797        textView131.setText("Center Center Bottom Bottom");
798
799        // 페더 콘텐츠 설정
800        View bottomBottomBottomBottomBottomBottomBottomBottomBottomView = navigationView.getBottomBottomBottomBottomBottomBottomBottomBottomView(0);
801        TextView textView
```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/BuyingScreen.java

99                     Toast.makeText(getApplicationContext(), "서버와 연결이
100                    끊겼습니다", Toast.LENGTH_SHORT).show();
100                    return;
101                }
102            } catch (JSONException e) { // 접속 오류가 난 것이라면
103                e.printStackTrace(); // 오류 추적
104                // 토스트 메세지를 띄워줌
105                Toast.makeText(getApplicationContext(), "no connection",
106                Toast.LENGTH_SHORT).show();
106            }
107        }
108    };
109    String purl2 = "http://ualsgur98.dothome.co.kr/CostInfo.php"; // 통신할 php
주소
110    // 데이터 전송을 위한 데이터 세팅
111    PHPRequest validateRequest2 = new PHPRequest(purl2, myGameName, Regnum,
responseListener2);
112    RequestQueue queue2 = Volley.newRequestQueue(BuyingScreen.this); // 큐를
생성
113    queue2.add(validateRequest2); // 큐에 추가
114
115
116    //UI 객체생성
117    yes_btn = findViewById(R.id.buy_check_yes);
118    no_btn = findViewById(R.id.buy_check_no);
119
120    editTextquantity = findViewById(R.id.buyQuantityBox); //게임 아이템 수량
입력할 공간 생성
121
122    //예 버튼을 눌렀을때 값 저장
123    yes_btn.setOnClickListener(new View.OnClickListener() {
124        @Override
125        public void onClick(View view) {
126            String getTextQuan = editTextquantity.getText().toString(); //
입력받은 수량의 값을 저장
127
128            //구매하려는 아이템 정보 DB에 보냄
129            Response.Listener<String> responseListener3 = new Response.
Listener<String>() {
130                @Override
131                public void onResponse(String response) { // onResponse 함수
재정의
132                    try { // 만약 반응이 왔고 정상적이라면
133                        JSONObject jsonResponse = new JSONObject(response);
// json 형태로 받기 위해 객체 준비
134                        boolean success = jsonResponse.getBoolean("success");
}; // success라는 key 값으로 boolean value를 받음
135                        if (success) { // success가 참이라면
136                            // 토스트 메세지를 띄워줌
137                            Toast.makeText(getApplicationContext(), "
구매하셨습니다", Toast.LENGTH_SHORT).show();
138                        }
139                        else { // success가 false일 때
140                            // 토스트 메세지를 띄워줌
141                            Toast.makeText(getApplicationContext(), "서버와
연결이 끊겼습니다", Toast.LENGTH_SHORT).show();
142                        }
143                    }
144                } catch (JSONException e) { // 접속 오류가 난 것이라면
e.printStackTrace(); // 오류 추적
145            }

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/BuyingScreen.java
146                                // 토스트 메세지를 띄워줌
147                                Toast.makeText(getApplicationContext(), "no
148 connection", Toast.LENGTH_SHORT).show();
149                            }
150                        }
151                        String purl3 = "http://ualsgur98.dothome.co.kr/Buy.php"; // 통신할
php 주소
152                                // 데이터 전송을 위한 데이터 세팅
153                                PHPRequest validateRequest3 = new PHPRequest(purl3, myGameName,
Regnum, Uninum, getTextQuan, myCharName, DBItemName, responseListener3);
154                                RequestQueue queue3 = Volley.newRequestQueue(BuyingScreen.this);
// 큐를 생성
155                                queue3.add(validateRequest3); // 큐에 추가
156                                // BuyingScreen에서 AuctionScreen으로 가는 intent 준비
157                                Intent intent = new Intent(getApplicationContext(), AuctionScreen
.class);
158                                intent.putExtra("account_info", accountInfo); // intent에 게임 계정
데이터 추가
159                                startActivity(intent); // AuctionScreen 실행
160                                finish(); // 현재 액티비티 종료
161                            }
162                        });
163
164                    //아니오 버튼을 눌렀을때 구매 취소, 경매장 화면으로 다시 돌아감
165                    no_btn.setOnClickListener(new View.OnClickListener() { //아니오 버튼 클릭
이벤트 생성
166                        @Override
167                        public void onClick(View view) {
168                            // 토스트 메세지를 띄워줌
169                            Toast.makeText(getApplicationContext(), "구매를 취소하셨습니다.", Toast.LENGTH_SHORT).show();
170                            // BuyingScreen에서 AuctionScreen으로 가는 intent 준비
171                            Intent intent = new Intent(getApplicationContext(), AuctionScreen
.class);
172                            intent.putExtra("account_info", accountInfo); // intent에 게임 계정
데이터 추가
173                            startActivity(intent); // AuctionScreen 실행
174                            finish(); // 현재 액티비티 종료
175                        }
176                    });
177                }
178
179
180 }
181
182

```

```
1 package com.example.teamauction;
2
3 import android.content.Context;
4 import android.util.AttributeSet;
5 import android.widget.CheckBox;
6 import android.widget.Checkable;
7 import android.widget.LinearLayout;
8
9 /**
10  * 체크박스에 관해서 직접 설정하기 위해 만든 클래스
11  * LinearLayout을 상속받아 체크박스를 재정의 해준다
12 */
13 public class CheckableLinearLayout extends LinearLayout implements Checkable {
14     // 만약 CheckBox가 아닌 View를 추가한다면 아래의 변수 사용 가능.
15     // private boolean mIsChecked ;
16
17     public CheckableLinearLayout(Context context, AttributeSet attrs) {
18         super(context, attrs);
19
20         // mIsChecked = false ;
21     }
22
23     @Override
24     public boolean isChecked() {
25         CheckBox cb = (CheckBox) findViewById(R.id.list_checkbox) ;
26
27         return cb.isChecked() ;
28         // return mIsChecked ;
29     }
30
31     @Override
32     public void toggle() {
33         CheckBox cb = (CheckBox) findViewById(R.id.list_checkbox) ;
34
35         setChecked(cb.isChecked() ? false : true) ;
36         // setChecked(mIsChecked ? false : true) ;
37     }
38
39     @Override
40     public void setChecked(boolean checked) {
41         CheckBox cb = (CheckBox) findViewById(R.id.list_checkbox) ;
42
43         if (cb.isChecked() != checked) {
44             cb.setChecked(checked) ;
45         }
46
47         // CheckBox 가 아닌 View의 상태 변경.
48     }
49 }
50 }
```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/DeleteScreen.java

```
1 package com.example.teamauction;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;
9 import android.widget.Toast;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 import com.android.volley.RequestQueue;
14 import com.android.volley.Response;
15 import com.android.volley.toolbox.Volley;
16
17 import org.json.JSONException;
18 import org.json.JSONObject;
19
20 /**
21 * 삭제하기 화면
22 * 판매중인 아이템 화면에서 리스트뷰 목록중 삭제할 아이템을 누르고 삭제하기 버튼을 클릭한 뒤 화면
23 * 삭제하기를 실질적으로 할 수 있게 하는 액티비티
24 */
25
26 public class DeleteScreen extends AppCompatActivity {
27     private GameAccountInfo accountInfo; // 계정 정보에 대한 객체
28     private TextView ItemName, ItemInfo, ItemPrice, ItemQuen; // 아이템 이름, 정보,
29     가격, 수량이 들어갈 텍스트 뷰
30     Button yes_btn, no_btn; // 확인, 취소 버튼
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.screen_delete);
35
36         Intent intent = getIntent(); // 로그인 계정 정보 받기
37         accountInfo = (GameAccountInfo) intent.getSerializableExtra("account_info");
38     } // intent라는 key로 GameAccountInfo 객체를 받아옴
39         String Uninum = intent.getExtras().getString("myuninum"); // 구매하려는 아이템의
40         Uninum 받아옴
41         String Regnum = intent.getExtras().getString("myregnum"); // 구매하려는 아이템의
42         Regnum 받아옴
43
44         String myGameName = accountInfo.getGameName(); // 게임 이름 정보 가져오기
45         String myCharName = accountInfo.getCharacterName(); // 게임 캐릭터 닉네임 정보
46         가져오기
47
48         ItemName = findViewById(R.id.dropItemName); // 게임 아이템 이름 들어갈 공간 생성
49         ItemInfo = findViewById(R.id.dropItemInfo); // 게임 아이템 고유정보 들어갈 공간
50         생성
51         ItemPrice = findViewById(R.id.dropCostBox); // 게임 아이템 가격정보 들어갈 공간
52         생성
53         ItemQuen = findViewById(R.id.dropQuantityBox); // 게임 아이템 가격정보 들어갈 공간
54         생성
55
56         // 삭제하려는 아이템 정보를 DB에서 받아와 화면에 보여줌
57         Response.Listener<String> responseListener = new Response.Listener<String>
58         >() {
59             @Override
60             public void onResponse(String response) { // onResponse 함수 재정의
```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/DeleteScreen.java

```

52         try { // 만약 반응이 왔고 정상적이라면
53             JSONObject jsonResponse = new JSONObject(response); // json
54             형태로 받기 위해 객체 준비
55             boolean success = jsonResponse.getBoolean("success"); // 
56             success라는 key 값으로 boolean value를 받음
57             if (success) { // success가 참이라면
58                 String DropItemName = jsonResponse.getString("ItemName")
59             ); // ItemName 데이터 저장
60                 String DropItemInfo = jsonResponse.getString("ItemInfo"
61             ); // ItemInfo 데이터 저장
62                 itemName.setText(DropItemName); // 삭제하려는 아이템 이름으로
63                 변경
64                 itemInfo.setText(DropItemInfo); // 삭제하려는 아이템 정보로 변경
65             } else { // success가 false일 때
66                 // 토스트 메세지를 띄워줌
67                 Toast.makeText(getApplicationContext(), "서버와 연결이
68                 끊겼습니다", Toast.LENGTH_SHORT).show();
69             }
70         }
71     };
72     String purl = "http://ualsgur98.dothome.co.kr/Del.php"; // 통신할 php 주소
73     // 데이터 전송을 위한 데이터 세팅
74     PHPRequest validateRequest = new PHPRequest(purl, myGameName, Uninum,
75     responseListener);
76     RequestQueue queue = Volley.newRequestQueue(DeleteScreen.this); // 큐를
77     생성
78     queue.add(validateRequest); // 큐에 추가
79
80     // 삭제 버튼을 눌렀을때 값 저장, 내 아이템 화면으로 돌아감
81     Response.Listener<String> responseListener2 = new Response.Listener<
82     String>() { // 삭제 버튼 클릭 이벤트 생성
83         @Override
84         public void onResponse(String response) { // onResponse 함수 재정의
85             try { // 만약 반응이 왔고 정상적이라면
86                 JSONObject jsonResponse = new JSONObject(response); // json
87                 형태로 보내기 위해 객체 준비
88                 boolean success = jsonResponse.getBoolean("success"); // 
89                 success라는 key 값으로 boolean value를 받음
90                 if (success) { // success가 참이라면
91                     String cost = jsonResponse.getString("Price"); // Price
92                     데이터 저장
93                     String quen = jsonResponse.getString("Quantity"); // 
94                     데이터 저장
95                     ItemPrice.setText(cost); // DB에서 받아온 가격으로 변경
96                     ItemQuen.setText(quen); // DB에서 받아온 수량으로 변경
97                 } else { // success가 false일 때
98                     // 토스트 메세지를 띄워줌
99                     Toast.makeText(getApplicationContext(), "서버와 연결이
100                    끊겼습니다", Toast.LENGTH_SHORT).show();
101                }
102            }
103        }
104    }
105    catch (JSONException e) { // 접속 오류가 난 것이라면
106        e.printStackTrace(); // 오류 추적
107    }
108}

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/DeleteScreen.java

97                     // 토스트 메세지를 띄워줌
98                     Toast.makeText(getApplicationContext(), "no connection",
99                         Toast.LENGTH_SHORT).show();
100                 }
101             );
102             String purl2 = "http://ualsgur98.dothome.co.kr/DelCostInfo.php"; // 통신할
    php 주소
103             // 데이터 전송을 위한 데이터 세팅
104             PHPRequest validateRequest2 = new PHPRequest(purl2, myGameName, Regnum,
    responseListener2);
105             RequestQueue queue2 = Volley.newRequestQueue(DeleteScreen.this); // 큐를
    생성
106             queue2.add(validateRequest2); // 큐에 추가
107
108             //UI 객체생성
109             yes_btn = findViewById(R.id.drop_check_yes);
110             no_btn = findViewById(R.id.drop_check_no);
111
112
113             //삭제 버튼을 눌렀을때 값 저장
114             yes_btn.setOnClickListener(new View.OnClickListener() {
115                 @Override
116                 public void onClick(View view) {
117                     Response.Listener<String> responseListener3 = new Response.
    Listener<String>() {
118                         @Override
119                         public void onResponse(String response) { // onResponse 함수
    재정의
120                             try { // 만약 반응이 왔고 정상적이라면
121                                 JSONObject jsonResponse = new JSONObject(response);
    // json 형태로 받기 위해 객체 준비
122                                 boolean success = jsonResponse.getBoolean("success");
    ); // success라는 key 값으로 boolean value를 받음
123                                 if (success) { // success가 참이라면
124                                     Toast.makeText(getApplicationContext(), ""
    삭제하였습니다.", Toast.LENGTH_SHORT).show();
    }
125                                 }
126                                 else { // success가 false일 때
127                                     // 토스트 메세지를 띄워줌
128                                     Toast.makeText(getApplicationContext(), "서버와
    연결이 끊겼습니다", Toast.LENGTH_SHORT).show();
    }
129                                 return;
    }
130                             } catch (JSONException e) { // 접속 오류가 난 것이라면
131                                 e.printStackTrace(); // 오류 추적
    // 토스트 메세지를 띄워줌
132                                 Toast.makeText(getApplicationContext(), "no
    connection", Toast.LENGTH_SHORT).show();
    }
133                             }
134                         };
    }
135                     );
    }
136                 );
    }
137             );
    }
138             String purl3 = "http://ualsgur98.dothome.co.kr/Delete.php"; // 통신할
    php 주소
139             // 데이터 전송을 위한 데이터 세팅
140             PHPRequest validateRequest3 = new PHPRequest(purl3, Regnum,
    myGameName, myCharName, responseListener3);
141             RequestQueue queue3 = Volley.newRequestQueue(DeleteScreen.this);
    // 큐를 생성
142             queue3.add(validateRequest3); // 큐에 추가

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/DeleteScreen.java

```
143                     // 토스트 메세지를 띄워줌
144                     Toast.makeText(getApplicationContext(), "판매 중인 아이템을
145                     삭제하셨습니다.", Toast.LENGTH_SHORT).show();
146                     // DeleteScreen에서 SellingItemScreen으로 가는 intent 준비
147                     Intent intent = new Intent(DeleteScreen.this, SellingItemScreen.
148                     class);
149                     intent.putExtra("account_info", accountInfo); // intent에 게임 계정
데이터 추가
150                     startActivity(intent); // SellingItemScreen 실행
151                     finish(); // 현재 액티비티 종료
152                 }
153             });
154         });
155     });
156     no_btn.setOnClickListener(new View.OnClickListener() {
157         // 아니오 버튼을 눌렀을때 구매 취소, 판매중인 아이템 화면으로 다시 돌아감
158         @Override
159         public void onClick(View view) {
160             // 토스트 메세지를 띄워줌
161             Toast.makeText(getApplicationContext(), "삭제를 취소하셨습니다.",
162             Toast.LENGTH_SHORT).show();
163             // DeleteScreen에서 SellingItemScreen으로 가는 intent 준비
164             Intent intent = new Intent(DeleteScreen.this, SellingItemScreen.
165             class);
166             intent.putExtra("account_info", accountInfo); // intent에 게임 계정
데이터 추가
167             startActivity(intent); // SellingItemScreen 실행
168             finish(); // 현재 액티비티 종료
169         }
170     });
171 }
```

```

1 package com.example.teamauction;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;
9 import android.widget.Toast;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 import com.android.volley.RequestQueue;
14 import com.android.volley.Response;
15 import com.android.volley.toolbox.Volley;
16
17 import org.json.JSONException;
18 import org.json.JSONObject;
19
20 /**
21 * 수정하기 화면
22 * 판매중인 아이템 화면에서 리스트뷰 목록중 수정할 아이템을 누르고 수정하기 버튼을 클릭한 뒤 화면
23 * 수정하기를 실질적으로 할 수 있게 하는 액티비티
24 */
25
26 public class FixScreen extends AppCompatActivity {
27     private EditText editTextcost, editTextquantity; //가격, 수량을 입력할 수 있는 에디트
28     텍스트 뷰
29     private GameAccountInfo accountInfo; // 계정 정보에 대한 객체
30     private TextView ItemName, ItemInfo; // 아이템 이름, 정보가 들어갈 텍스트 뷰
31     Button yes_btn, no_btn; // 확인, 취소 버튼
32
33     @Override
34     protected void onCreate(Bundle savedInstanceState) {
35         super.onCreate(savedInstanceState);
36         setContentView(R.layout.screen_fix);
37
38         Intent intent = getIntent(); // 로그인 계정 정보 받기
39         accountInfo = (GameAccountInfo) intent.getSerializableExtra("account_info");
40     // intent라는 key로 GameAccountInfo 객체를 받아옴
41         String myGameName = accountInfo.getGameName(); // 게임 이름 정보 가져오기
42         String myCharName = accountInfo.getCharacterName(); // 게임 캐릭터 닉네임 정보
43         가져오기
44         String Uninum = intent.getExtras().getString("myuninum"); //구매하려는 아이템의
45         Uninum 받아옴
46         String Regnum = intent.getExtras().getString("myregnum"); //구매하려는 아이템의
47         Regnum 받아옴
48
49         //UI 객체생성
50         yes_btn = findViewById(R.id.fix_check_yes);
51         no_btn = findViewById(R.id.fix_check_no);
52
53         editTextcost = findViewById(R.id.fixCostBox); //수정하려는 가격 입력하는 공간
54         생성
55         editTextquantity = findViewById(R.id.fixQuantityBox); //수정하려는 수량
56         입력하는 공간 생성
57
58         ItemName = findViewById(R.id.FixItemName); //게임 아이템 이름 들어갈 공간 생성
59         ItemInfo = findViewById(R.id.FixItemInfo); //게임 아이템 고유정보 들어갈 공간 생성
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
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
637
638
639
639
640
641
642
643
644
645
645
646
647
647
648
649
649
650
651
652
653
654
655
655
656
657
657
658
659
659
660
661
662
663
664
664
665
666
666
667
667
668
668
669
669
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/FixScreen.java
54      //수정하려는 아이템 정보를 DB에서 받아와 화면에 보여줌
55      Response.Listener<String> responseListener = new Response.Listener<String>
56      >() {
57          @Override
58          public void onResponse(String response) { // onResponse 함수 재정의
59              try {// 만약 반응이 왔고 정상적이라면
60                  JSONObject jsonResponse = new JSONObject(response); // json
61                  형태로 받기 위해 객체 준비
62                  boolean success = jsonResponse.getBoolean("success"); //
63                  success라는 key 값으로 boolean value를 받음
64                  if (success) { // success가 참이라면
65                      String BuyItemName = jsonResponse.getString("ItemName");
66                      // ItemName 데이터 저장
67                      String BuyItemInfo = jsonResponse.getString( "ItemInfo" );
68                  } // ItemInfo 데이터 저장
69                  itemName.setText(BuyItemName); //수정하려는 아이템 이름으로 변경
70                  itemInfo.setText(BuyItemInfo); //수정하려는 아이템 정보로 변경
71              } else {//success가 false일 때
72                  // 토스트 메세지를 띄워줌
73                  Toast.makeText(getApplicationContext(), "서버와 연결이
74                  끊겼습니다", Toast.LENGTH_SHORT).show();
75                  return;
76              }
77          } catch (JSONException e) { // 접속 오류가 난 것이라면
78              e.printStackTrace(); // 오류 추적
79              // 토스트 메세지를 띄워줌
80              Toast.makeText(getApplicationContext(), "no connection",
81              Toast.LENGTH_SHORT).show();
82          }
83      }
84      String purl = "http://ualsgur98.dothome.co.kr/FixItemInfo.php"; // 통신할
85      php 주소
86      // 데이터 전송을 위한 데이터 세팅
87      PHPRequest validateRequest = new PHPRequest(purl, myGameName, Uninum,
88      responseListener);
89      RequestQueue queue = Volley.newRequestQueue(FixScreen.this); // 큐를 생성
90      queue.add(validateRequest); // 큐에 추가
91
92      //수정 버튼을 눌렀을때 값 저장, 내 아이템 화면으로 돌아감
93      yes_btn.setOnClickListener(new View.OnClickListener() { // 수정 버튼 클릭
94          이벤트 생성
95          @Override
96          public void onClick(View view) {
97              String get_textcost = editTextcost.getText().toString(); //입력받은
98              가격의 값을 저장
99              String get_textquan = editTextquantity.getText().toString(); //
100             입력받은 수량의 값을 저장
101             //수정하려는 아이템 정보 DB에 보냄
102             Response.Listener<String> responseListener3 = new Response.
103             Listener<String>() {
104                 @Override
105                 public void onResponse(String response) { // onResponse 함수
106                     재정의
107                     try {
108                         JSONObject jsonResponse = new JSONObject(response);
109                         // json 형태로 받기 위해 객체 준비
110                         boolean success = jsonResponse.getBoolean("success");
111                     } // success라는 key 값으로 boolean value를 받음
112                     if (success) { // success가 참이라면
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/FixScreen.java

98                     Toast.makeText(getApplicationContext(), " "
99             수정하였습니다.", Toast.LENGTH_SHORT).show();
100            } else { // success가 false일 때
101                // 토스트 메세지를 띄워줌
102                Toast.makeText(getApplicationContext(), "서버와
103                    연결이 끊겼습니다", Toast.LENGTH_SHORT).show();
104                return;
105            }
106        } catch (JSONException e) { // 접속 오류가 난 것이라면
107            e.printStackTrace(); // 오류 추적
108            // 토스트 메세지를 띄워줌
109            Toast.makeText(getApplicationContext(), "no
110                connection", Toast.LENGTH_SHORT).show();
111        }
112    }
113    String purl3 = "http://ualsgur98.dothome.co.kr/fix.php"; //통신할
114    php 주소
115    // 데이터 전송을 위한 데이터 세팅
116    PHPRequest validateRequest3 = new PHPRequest(purl3, Regnum,
117    myGameName, myCharName, get_textcost, get_textquan, responseListener3);
118    RequestQueue queue3 = Volley.newRequestQueue(FixScreen.this); // //
119    큐를 생성
120    queue3.add(validateRequest3); // 큐에 추가
121
122    // 토스트 메세지를 띄워줌
123    Toast.makeText(getApplicationContext(), "판매 중인 아이템을
124        수정하셨습니다.", Toast.LENGTH_SHORT).show();
125    // FixScreen에서 SellingItemScreen으로 가는 intent 준비
126    Intent intent = new Intent(FixScreen.this, SellingItemScreen.
127        class); // intent에 게임 계정 데이터 추가
128    intent.putExtra("account_info", accountInfo); // intent에 게임 계정
129    데이터 추가
130    startActivity(intent); // SellingItemScreen 실행
131    finish(); // 현재 액티비티 종료
132}
133}
134}
135}
136}
137}
138}
139}
140}
141}
142}

```

```
1 package com.example.teamauction;
2
3 import java.io.Serializable;
4
5 /**
6  * Intent 객체 전달을 위한 클래스
7  * 계정 정보를 저장하기 위한 클래스이기도 함
8 */
9 public class GameAccountInfo implements Serializable {
10     private String LoginID;           // 계정 아이디
11     private String LoginPW;          // 계정 비밀번호
12     private String gameName;         // 게임 이름
13     private String characterName;    // 캐릭터 이름
14     private String gamePublisherID;  // 게임 아이디
15     private String gamePublisherPW; // 게임 비밀번호
16
17     public GameAccountInfo() { }     // 생성자
18
19     public void setGameName(String gameName) {
20         this.gameName = gameName;
21     }
22     public void setCharacterName(String characterName) {
23         this.characterName = characterName;
24     }
25     public void setGamePublisherID(String gamePublisherID) { this.gamePublisherID
= gamePublisherID; }
26     public void setGamePublisherPW(String gamePublisherPW) { this.gamePublisherPW
= gamePublisherPW; }
27     public void setLoginID(String loginID) { this.LoginID = loginID; }
28     public void setLoginPW(String loginPW) { this.LoginPW = loginPW; }
29
30     public String getGameName() {
31         return gameName;
32     }
33     public String getCharacterName() { return characterName; }
34     public String getGamePublisherID() {
35         return gamePublisherID;
36     }
37     public String getGamePublisherPW() {
38         return gamePublisherPW;
39     }
40     public String getLoginID() { return LoginID; }
41     public String getLoginPW() { return LoginPW; }
42
43     public void resetData() {           // 필요한 데이터만 남기고 초기화
44         this.gameName = "";
45         this.gamePublisherID = "";
46         this.gamePublisherPW = "";
47         this.characterName = "";
48     }
49 }
50 }
```

```

1 package com.example.teamauction;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.ImageButton;
8 import android.widget.ListView;
9 import android.widget.Toast;
10
11 import androidx.appcompat.app.AppCompatActivity;
12 import androidx.core.content.ContextCompat;
13
14 import com.android.volley.RequestQueue;
15 import com.android.volley.Response;
16 import com.android.volley.toolbox.Volley;
17
18 import org.json.JSONArray;
19 import org.json.JSONException;
20 import org.json.JSONObject;
21
22 /**
23 * 게임 로그인을 완료한 상황에서 나오는 액티비티
24 * 게임 계정 내에 있는 게임 캐릭터를 선택할 수 있다.
25 */
26 public class GameChActivity extends AppCompatActivity {
27
28     private GameAccountInfo accountInfo;      // 계정 정보를 받기 위한 GameAccountInfo
29     객체 생성
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_character_select);
35
36         Intent account_info = getIntent(); // intent를 받아옴
37         // account_info라는 key로 GameAccountInfo 객체를 받아옴
38         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info");
39
40         ImageButton CanalAdda = findViewById(R.id.cancel_account_add); // 뒤로가기
41         버튼
42         CanalAdda.setOnClickListener(new View.OnClickListener() {           // 뒤로가기
43             버튼 클릭 이벤트 생성
44             @Override
45             public void onClick(View view) {          // onClick 함수 재정의
46                 // GameChActivity에서 GameLoginActivity으로 가는 intent 정의
47                 Intent intent = new Intent(GameChActivity.this, GameLoginActivity.
48                 class);
49                 intent.putExtra("account_info", accountInfo); // intent에 데이터를
50                 넣음
51                 startActivity(intent); // intent 실행
52                 finish(); // 현재 액티비티 종료
53             }
54         });
55
56         ListView listview;          // 리스트뷰 변수
57         ListViewAdapter adapter; // 리스트뷰 어댑터
58
59         // Adapter 생성

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/GameChActivity.java

55         adapter = new ListViewAdapter();      // 리스트뷰 어댑터 생성
56
57         // 리스트뷰 참조 및 Adapter 달기
58         listview = (ListView) findViewById(R.id.listview2);
59         listview.setAdapter(adapter);    // 리스트뷰 어댑터 설정
60
61         // volley 통신을 수행하려고 응답대기를 준비
62         Response.Listener<String> responseListener = new Response.Listener<String>
63         >() {
64             @Override
65             public void onResponse(String response) {           // onResponse 함수
66                 // 정상적이라면,
67                 try {
68                     JSONObject jsonResponse = new JSONObject(response
69                     );           // json 형태로 받기 위해 객체 준비
70                     JSONArray Ids = jsonResponse.getJSONArray("CharacterList"
71                     );   // CharacterList key 값으로 배열을 받음
72                     for (int i = 0; i < Ids.length(); i++) {     // Ids 길이만큼
73                         JSONObject item = Ids.getJSONObject(i); // Ids의 데이터를
74                         가져옴
75                         String characterName = item.getString("gameNickname"
76                         );   // gameNickname 데이터 저장
77                         String characterMony = item.getString("gameMoney"
78                         );   // gameMoney 데이터 저장
79                         // 리스트뷰에 데이터 추가
80                         adapter.addItem(ContextCompat.getDrawable(GameChActivity.
81                         this,
82                             R.drawable.ic_baseline_account_box_24),
83                         characterName, characterMony);
84                     }
85                     adapter.notifyDataSetChanged();      // 반복문 수행 후에 리스트가
86                     업데이트 됨을 알림
87                 } catch (JSONException e) { // 접속 오류가 난 것이라면
88                     e.printStackTrace();        // 오류 추적
89                     // 토스트 메세지를 띄워줌
90                     Toast.makeText(getApplicationContext(), "no connection",
91                     Toast.LENGTH_SHORT).show();
92                 }
93             };
94
95             String purl = "http://ualsgur98.dothome.co.kr/CharacterList.php"
96             ;           // 통신할 php 주소
97             // 데이터 전송을 위한 데이터 세팅
98             PHPRequest validateRequest = new PHPRequest( purl, accountInfo.
99             getGameName(),
100                accountInfo.getGamePublisherID(), accountInfo.getGamePublisherPW
101                (), responseListener);
102                RequestQueue queue = Volley.newRequestQueue(GameChActivity.this);    // 큐를 생성
103                queue.add(validateRequest);      // 큐에 추가
104
105                Button selectCharacter = findViewById(R.id.select_game_character
106                );           // 캐릭터 선택 버튼
107                selectCharacter.setOnClickListener(new View.OnClickListener
108                () {
109                    // 선택버튼 클릭 이벤트 생성
110                    @Override
111                    public void onClick(View view

```

```

95 ) {
96     int pos = listView.getCheckedItemPosition
97     // 리스트뷰 위치데이터 변수
98     if (pos > -1) { // pos는 0이상이므로 그 이상일 때만 작동
99         ListViewitem item = (ListViewitem) adapter.getItem(pos)
100        // 리스트뷰의 아이템을 가져옴
101        accountInfo.setCharacterName(item.getText
102        // GameAccountInfo 객체에 캐릭터 이름 저장
103    Listener<String>() {
104        @Override
105        public void onResponse(String response
106    ) {
107        // onResponse 함수 재정의
108        try
109        {
110            // 만약 반응이 왔고 정상적이라면,
111            JSONObject jsonResponse = new JSONObject(response
112        ); // json 형태로 받기 위해 객체 준비
113        boolean success = jsonResponse.getBoolean("success"); // success라는 key 값으로 boolean value를 받음
114        if(success
115        ) {
116            // success가 참이라면
117            // 토스트 메세지를 띄워주고
118            Toast.makeText(getApplicationContext(), "계정이
119            정상적으로 연동되었습니다", Toast.LENGTH_SHORT).show();
120            // GameChActivity에서 MainActivity으로 가는
121            intent 생성
122            Intent intent = new Intent(GameChActivity.
123            this, MainActivity.class);
124            accountInfo.resetData(); // 필요한 정보를
125            빼고는 비워줌
126            intent.putExtra("account_info", accountInfo
127            );
128            startActivity(intent); // intent 실행
129            finish(); // 현재 액티비티 종료
130        }
131        else { // success가 참이 아니라면
132            // 토스트 메세지를 띄워줌
133            Toast.makeText(getApplicationContext(), "서버
134            접속 오류", Toast.LENGTH_SHORT).show();
135        }
136        } catch (JSONException e) { // 접속 오류가 난 것이라면
137            e.printStackTrace(); // 오류 추적
138            // 토스트 메세지를 띄워줌
139            Toast.makeText(getApplicationContext(), "no
140            connection", Toast.LENGTH_SHORT).show();
141        }
142    }
143    }
144    );
145
146    String purl = "http://ualsgur98.dothome.co.kr/
147    MyGameAndCharacter.php"; // 통신할 php 주소
148    // 데이터 전송을 위한 데이터 세팅
149    PHPRequest validateRequest = new PHPRequest( purl,
150    accountInfo.getLoginID(), accountInfo.getLoginPW(), accountInfo.getGameName(),
151    accountInfo.getGamePublisherID(), accountInfo.
152    getGamePublisherPW(), accountInfo.getCharacterName(), responseListener);
153    RequestQueue queue = Volley.newRequestQueue(GameChActivity.
154    this); // 큐를 생성

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/GameChActivity.java

```
135         queue.add(validateRequest);      // 큐에 추가
136     }
137     else { // pos가 -1 이하 이므로 리스트뷰 영역을 터치한 적이 없음
138         // 토스트 메세지를 띄워줌
139         Toast.makeText(GameChActivity.this, "캐릭터를 선택해주세요", Toast
140 .LENGTH_SHORT).show();
141     }
142 }
143 });
144 }
145 }
```

```

1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.core.content.ContextCompat;
5
6 import android.content.Intent;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.ImageButton;
11 import android.widget.ListView;
12 import android.widget.Toast;
13
14 import com.android.volley.RequestQueue;
15 import com.android.volley.Response;
16 import com.android.volley.toolbox.Volley;
17
18 import org.json.JSONArray;
19 import org.json.JSONException;
20 import org.json.JSONObject;
21
22 /**
23 * 게임 목록이 나오는 액티비티
24 * 연동 가능한 게임 목록이 나오고 선택할 수 있다
25 */
26 public class GameListActivity extends AppCompatActivity {
27
28     private ListViewAdapter adapter;           // 리스트뷰 어댑터
29     private ListView listview;                 // 리스트뷰
30     private GameAccountInfo accountInfo;      // 계정 정보를 받기 위한 GameAccountInfo
 객체 생성
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_add_account);
36
37         Intent account_info = getIntent(); // intent를 받아옴
38         // account_info라는 key로 GameAccountInfo 객체를 받아옴
39         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info");
40
41         ImageButton go_main = findViewById(R.id.back_account); // 뒤로가기 버튼
42         go_main.setOnClickListener(new View.OnClickListener() { // 뒤로가기 버튼
 클릭 이벤트 생성
43             @Override
44             public void onClick(View view) { // onClick 함수 재정의
45                 // GameListActivity MainActivity으로 가는 intent 정의
46                 Intent intent = new Intent(GameListActivity.this, MainActivity.
 class);
47                 intent.putExtra("account_info", accountInfo); // intent에 데이터를
 넣음
48                 startActivity(intent); // intent 실행
49                 finish(); // 현재 액티비티 종료
50             }
51         });
52
53         adapter = new ListViewAdapter(); // Adapter 생성
54
55         // 리스트뷰 참조 및 Adapter달기

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/GameListActivity.java

56         listview = (ListView) findViewById(R.id.listview_choose_game);
57         listview.setAdapter(adapter);
58
59         // volley 통신을 수행하려고 응답대기를 준비
60         Response.Listener<String> responseListener = new Response.Listener<String>
61             >() {
62                 @Override
63                 public void onResponse(String response)
64                     // onResponse 함수 재정의
65                     try { // 만약 반응이 왔고 정상적이라면,
66                         JSONObject jsonResponse = new JSONObject(response)
67                     }; // json 형태로 받기 위해 객체 준비
68                     JSONArray games = jsonResponse.getJSONArray("GameList"); // //
69                     GameList key 값으로 배열을 받음
70                     for (int i = 0; i < games.length(); i++) { // //
71                         games 길이만큼 반복문 수행
72                         JSONObject item = games.getJSONObject(i); // //
73                         String gameName = item.getString("GameName"); // GameName
74                         데이터 저장
75                         // 리스트뷰에 데이터 추가
76                         adapter.addItem(ContextCompat.getDrawable(
77                             GameListActivity.this,
78                                 R.drawable.ic_baseline_account_box_24), gameName
79 , "name");
80                     }
81                     adapter.notifyDataSetChanged(); // 반복문 수행 후에 리스트가
82                     업데이트 됨을 알림
83                     } catch (JSONException e) { // 접속 오류가 난 것이라면
84                         e.printStackTrace(); // 오류 추적
85                         // 토스트 메세지를 띄워줌
86                         Toast.makeText(getApplicationContext(), "no connection",
87                             Toast.LENGTH_SHORT).show();
88                     }
89             };
90             String purl = "http://ualsgur98.dothome.co.kr/GameList.php"
91             ; // 통신할 php 주소
92             PHPRequest validateRequest = new PHPRequest( purl, responseListener
93             ); // 데이터 전송을 위한 데이터 세팅
94             RequestQueue queue = Volley.newRequestQueue(GameListActivity.this); // //
95             큐를 생성
96             queue.add(validateRequest); // 큐에 추가
97
98             Button publisher_login = findViewById(R.id.login_button_publisher
99             ); // 로그인 버튼
100            publisher_login.setOnClickListener(new View.OnClickListener
101             () {
102                 // 로그인 클릭 이벤트 생성
103                 @Override
104                 public void onClick(View view
105             ) {
106                     // onClick 함수 재정의
107                     int pos = listview.getCheckedItemPosition
108                     (); // 리스트뷰 위치데이터 변수
109                     if (pos > -1) { // pos는 0이상이므로 그 이상일 때만 작동
110                         ListViewItem item = (ListViewItem) adapter.getItem(pos
111                     ); // 리스트뷰의 아이템을 가져옴
112                         accountInfo.setGameName(item.getText
113                     ); // GameAccountInfo 객체에 게임 이름 저장
114                     // GameListActivity에서 GameLoginActivity으로 가는 intent 생성
115                     Intent intent = new Intent(GameListActivity.this,

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/GameListActivity.java

```
95 GameLoginActivity.class);
96         intent.putExtra("account_info", accountInfo); // 
97         intent에 데이터 추가
98         startActivity(intent); // intent 실행
99         finish(); // 현재 액티비티 종료
100     }
101     else { // pos가 -1 이하 이므로 리스트뷰 영역을 터치한 적이 없음
102         // 토스트 메세지를 띄워줌
103         Toast.makeText(GameListActivity.this, "게임을 선택해주세요", Toast.LENGTH_SHORT).show();
104     }
105 }
106 });
107 }
108 }
109 }
```

```

1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.ImageButton;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import com.android.volley.RequestQueue;
15 import com.android.volley.Response;
16 import com.android.volley.toolbox.Volley;
17
18 import org.json.JSONException;
19 import org.json.JSONObject;
20
21 /**
22 * 게임사 로그인을 하는 엑티비티
23 * 게임 목록을 선택한 후 나오는 엑티비티이며
24 * 게임사 로그인을 통해 게임 캐릭터를 선택할 수 있도록 도와주는 엑티비티이다
25 */
26 public class GameLoginActivity extends AppCompatActivity {
27     private EditText game_id, game_pw; // 게임 로그인 아이디, 비밀번호 입력 필드
28     private GameAccountInfo accountInfo; // 계정 정보를 받기 위한 GameAccountInfo
29     객체 생성
30     TextView gameName; // 게임 이름 텍스트뷰
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_publisher_login);
35
36         Intent account_info = getIntent(); // intent를 받아옴
37         // account_info라는 key로 GameAccountInfo 객체를 받아옴
38         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info");
39
40         gameName = findViewById(R.id.game_name); // 게임 이름 텍스트뷰
41         gameName.setText(accountInfo.getGameName()); // 게임 이름 설정
42
43         game_id = findViewById(R.id.login_publisher_emailbox); // 아이디 입력 필드
44         game_pw = findViewById(R.id.login_publisher_pwbox); // 비밀번호 입력 필드
45
46         Button publisher_login_btn = findViewById(R.id.login_publisher_button);
47         // 로그인 버튼
48         publisher_login_btn.setOnClickListener(new View.OnClickListener() {
49             // 로그인 버튼 클릭 이벤트 생성
50             @Override
51             public void onClick(View view) { // onClick 함수
52                 // 재정의
53                 String gameID = game_id.getText().toString(); // 아이디 입력 필드에서
54                 String 형태로 데이터를 가져옴
55                 String gamePW = game_pw.getText().toString(); // 비밀번호 입력
56                 필드에서 String 형태로 데이터를 가져옴
57
58                 if(gameID.equals("")) { // 만약 값이 비어있다면
59                     // 토스트 메세지를 띄워주고

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/GameLoginActivity.java

54             Toast.makeText(getApplicationContext(), "아이디를 입력해주세요",
55             Toast.LENGTH_SHORT).show();
56         return; // 이벤트 종료
57     }
58
59         // volley 통신을 수행하려고 응답대기를 준비
60         Response.Listener<String> responseListener = new Response.
61     Listener<String>() {
62             @Override
63             public void onResponse(String response
64                     // onResponse 함수 재정의
65                     try { // 만약 반응이 왔고 정상적이라면,
66                         JSONObject jsonObject = new JSONObject(response
67                     ); // json 형태로 받기 위해 객체 준비
68                     boolean success = jsonObject.getBoolean("success"
69                     ); // GameList key 값으로 배열을 받음
70
71                     if (success) { // success가 참이라면
72                         String jsonID = jsonObject.getString("gameID"
73                     ); // gameID 데이터 저장
74                         String jsonPW = jsonObject.getString("gamePW"
75                     ); // gamePW 데이터 저장
76                         accountInfo.setGamePublisherID(jsonID
77                     ); // GameAccountInfo 객체에 아이디 저장
78                         accountInfo.setGamePublisherPW(jsonPW
79                     ); // GameAccountInfo 객체에 비밀번호 저장
80                         // 토스트 메세지를 띄워줌
81                         Toast.makeText(getApplicationContext(), jsonID+
82                             "님 환영합니다", Toast.LENGTH_SHORT).show();
83                         // GameLoginActivity에서 GameChActivity으로 가는
84                         intent 생성
85                         Intent intent = new Intent(GameLoginActivity.this
86                             , GameChActivity.class);
87                         intent.putExtra("account_info", accountInfo
88                     ); // intent에 데이터를 넣음
89                         startActivity(intent); // intent 실행
90                         finish(); // 현재 액티비티 종료
91
92                     } else { // success가 false일 때
93                         // 토스트 메세지를 띄워주고
94                         Toast.makeText(getApplicationContext(), "서버와
95                             연결에 실패했습니다", Toast.LENGTH_SHORT).show();
96                     return; // 이벤트 종료
97                 }
98             }
99         };
100         String purl = "http://ualsgur98.dothome.co.kr/gamelogin.php"
101             // 통신할 php 주소
102             // 데이터 전송을 위한 데이터 세팅
103             PHPRequest validateRequest = new PHPRequest( purl, accountInfo.
104             getGameName(),
105                 gameID, gamePW, responseListener);
106             RequestQueue queue = Volley.newRequestQueue(GameLoginActivity.

```

```
96 this); // 큐를 생성
97         queue.add(validateRequest); // 큐에 추가
98     }
99 }
100
101    ImageButton back_add_publisher = findViewById(R.id.back_add_account);
// 뒤로가기 버튼
102    back_add_publisher.setOnClickListener(new View.OnClickListener() {
// 뒤로가기 버튼 클릭 이벤트 생성
103    @Override
104    public void onClick(View view) {
// onClick 함수 재정의
105        // GameLoginActivity에서 GameListActivity으로 가는 intent 생성
106        Intent intent = new Intent(GameLoginActivity.this,
GameListActivity.class);
107        intent.putExtra("account_info", accountInfo); // intent에 데이터를 넣음
108        startActivity(intent); // intent 실행
109        finish(); // 현재 액티비티 종료
110    }
111 });
112
113 }
114 }
```

```

1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.core.content.ContextCompat;
5
6 import android.content.Intent;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.ImageButton;
11 import android.widget.ListView;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import com.android.volley.RequestQueue;
16 import com.android.volley.Response;
17 import com.android.volley.toolbox.Volley;
18
19 import org.json.JSONArray;
20 import org.json.JSONException;
21 import org.json.JSONObject;
22
23 /**
24 * 내 아이템 화면
25 * 경매장 화면에서 내 아이템 아이템 버튼을 클릭한 뒤 화면
26 * 내 아이템 중 판매 할 아이템을 선택할 수 있게 하는 액티비티
27 */
28
29 public class ItemCheckScreen extends AppCompatActivity {
30     private GameAccountInfo accountInfo; // 계정 정보에 대한 객체
31     private TextView UserCharName; // 유저 게임 닉네임 텍스트뷰
32     private ImageButton backButton; // 뒤로가기 버튼
33     private TextView money; // 유저 게임 돈 텍스트뷰
34     private Button sellingbutton; //판매하기 버튼
35
36     ListView listview = null;
37     @Override
38     protected void onCreate(Bundle savedInstanceState) {
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.screen_check_item);
41         ListViewAdapter adapter; // 어댑터 선언
42         Intent account_info = getIntent(); // 로그인 계정 정보 받기
43         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info"); // account_info라는 key로 GameAccountInfo 객체를 받아옴
44         String myGameName = accountInfo.getGameName(); // 게임 이름 정보 가져오기
45         String myCharName = accountInfo.getCharacterName(); // 게임 캐릭터 닉네임 정보
46         가져오기
47         UserCharName = findViewById(R.id.MyCharName); //유저 닉네임 들어갈 공간 생성
48         UserCharName.setText(myCharName); //로그인한 계정 캐릭터 이름으로 변경
49
50         // Adapter 생성
51         adapter = new ListViewAdapter() ;
52
53         // 리스트뷰 참조 및 Adapter달기
54         listview = (ListView) findViewById(R.id.my_item_list);
55         listview.setAdapter(adapter);
56
57         //내 아이템 정보를 DB에서 받아와 리스트뷰에 보여줌
58         Response.Listener<String> responseListener = new Response.Listener<String>

```

```

58 >() {
59         @Override
60         public void onResponse(String response) { // onResponse 함수 재정의
61             try { // 만약 반응이 왔고 정상적이라면,
62                 JSONObject jsonResponse = new JSONObject(response); // json
63                 형태로 받기 위해 객체 준비
64                 JSONArray games = jsonResponse.getJSONArray("UserInfo"); // UserInfo
65                 Selling 값으로 배열을 받음
66                 for (int i = 0; i < games.length(); i++) { // games의 길이만큼
67                     반복문 수행
68                     JSONObject item = games.getJSONObject(i); // games의
69                     데이터를 가져옴
70                     저장
71                     데이터 저장
72                     ItemQuantity 데이터 저장
73                     adapter.addAuctionItem(ContextCompat.getDrawable(
74                         ItemCheckScreen.this,
75                         R.drawable.ic_baseline_account_box_24), ItemName,
76                         ItemQuantity, UniNum, ""); //리스트뷰에 있는 아이템의 이름, 가격, UniNumber,
77                         RegisterNumber 정보 저장
78                     }
79                     adapter.notifyDataSetChanged(); // 반복문 수행 후에 리스트가 업데이트
80                     됨을 알림
81                     } catch (JSONException e) { // 접속 오류가 난 것이라면
82                         e.printStackTrace(); // 오류 추적
83                         // 토스트 메세지를 띄워줌
84                         Toast.makeText(getApplicationContext(), "no connection",
85                         Toast.LENGTH_SHORT).show();
86                     }
87                     }
88                     String purl = "http://ualsgur98.dothome.co.kr/UserInfo.php"; // 통신할 php
89                     주소
90                     // 데이터 전송을 위한 데이터 세팅
91                     PHPRequest validateRequest = new PHPRequest( purl, myGameName, myCharName,
92                     responseListener);
93                     RequestQueue queue = Volley.newRequestQueue(ItemCheckScreen.this); // 큐를
94                     생성
95                     queue.add(validateRequest); // 큐에 추가
96                     //내 돈 띄워주기 구현
97                     Response.Listener<String> responseListener1 = new Response.Listener<
String>() {
98                         @Override
99                         public void onResponse(String response) { // onResponse 함수 재정의
100                             try { // 만약 반응이 왔고 정상적이라면
101                                 JSONObject jsonResponse = new JSONObject(response); // json
102                                 형태로 받기 위해 객체 준비
103                                 JSONArray games = jsonResponse.getJSONArray("Money"); // Money
104                                 Money 값으로 배열을 받음
105                                 JSONObject item = games.getJSONObject(0); // games의 데이터를
106                                 가져옴
107                                 데이터 저장
108                                 String Money1 = item.getString("gameMoney"); // gameMoney
109                                 //DB에서 받아온 내 돈으로 변경
110                                 money = findViewById(R.id.moneyBox); //돈이 들어갈 공간 생성
111                                 money.setText(Money1); //유저의 돈으로 변경

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/ItemCheckScreen.java

98                         adapter.notifyDataSetChanged(); // 반복문 수행 후에 리스트가 업데이트
    됨을 알림
99                     } catch (JSONException e) { // 접속 오류가 난 것이라면
100                         e.printStackTrace(); // 오류 추적
101                         // 토스트 메세지를 띄워줌
102                         Toast.makeText(getApplicationContext(), "no connection",
103                             Toast.LENGTH_SHORT).show();
104                     }
105                 };
106             String purl1 = "http://ualsgur98.dothome.co.kr/Money.php"; // 통신할 php
    주소
107             // 데이터 전송을 위한 데이터 세팅
108             PHPRequest validateRequest1 = new PHPRequest( purl1, myGameName,
    myCharName, responseListener1);
109             RequestQueue queue1 = Volley.newRequestQueue(ItemCheckScreen.this); //
    큐를 생성
110             queue1.add(validateRequest1); // 큐에 추가
111
112
113
114         backButton = findViewById(R.id.back_auctionScreen); // 뒤로가기 버튼 경매장
    화면으로 이동
115         backButton.setOnClickListener(new View.OnClickListener() { // 뒤로가기 버튼
    클릭 이벤트 생성
116             @Override
117             public void onClick(View view) { // onClick 함수 재정의
    // ItemCheckScreen에서 AuctionScreen으로 가는 intent 준비
118                 Intent intent = new Intent(ItemCheckScreen.this, AuctionScreen.
    class);
119                 intent.putExtra("account_info", accountInfo); // 경매장 화면에 계정
    정보 보내줌
120                 startActivity(intent); // AuctionScreen 실행
121                 finish(); // 현재 액티비티 종료
122             }
123         });
124
125
126
127
128         //리스트 뷰에 있는 아이템을 터치하고 판매하기 버튼을 누르면 SellingScreen창 띄우기
129         sellingbutton = findViewById(R.id.sellButton);
130         sellingbutton.setOnClickListener(new View.OnClickListener() { // 판매하기
    버튼 클릭 이벤트 생성
131             @Override
132             public void onClick(View view) { // onClick 함수 재정의
133                 int pos = listView.getCheckedItemPosition(); // 리스트뷰 위치데이터
    변수
134                 if (pos > -1) { // pos는 0이상이므로 그 이상일 때만 작동
135                     ListViewitem item = (ListViewitem) adapter.getItem(pos); //
    터치한 아이템의 위치값 받아옴
136                     String uninumber = item.getUninumber(); //아이템의 Uninumber
    받아옴
137                     String regnum = item.getRegnumber(); //아이템의 RegisterNumber
    받아옴
138                     // ItemCheckScreen에서 SellingScreen으로 가는 intent 준비
139                     Intent intent = new Intent(ItemCheckScreen.this,
    SellingScreen.class);
140                     intent.putExtra("account_info", accountInfo); // 구매하기 화면에
    계정 정보 보내줌
141                     intent.putExtra("myuninum", uninumber); // 구매하기 화면에

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/ItemCheckScreen.java

```
141 Uninumber 보내줌
142             startActivity(intent); // SellingScreen 실행
143             finish(); // 현재 액티비티 종료
144         }
145     });
146 }
147 }
148 }
149 }
```

```

1 package com.example.teamauction;
2
3 import android.content.Context;
4 import android.graphics.drawable.Drawable;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.BaseAdapter;
9 import android.widget.Filter;
10 import android.widget.Filterable;
11 import android.widget.ImageView;
12 import android.widget.TextView;
13
14 import java.util.ArrayList;
15 /**
16 * 리스트뷰를 보여주는 클래스
17 * 리스트를 구성하는 요소와 보여주는 요소를 지정할 수 있음
18 */
19 public class ListViewAdapter extends BaseAdapter implements Filterable {
20     // Adapter에 추가된 데이터를 저장하기 위한 ArrayList
21     private ArrayList<ListViewItem> listViewItemList = new ArrayList<ListViewItem>();
22
23     // 필터링된 결과 데이터를 저장하기 위한 ArrayList. 최초에는 전체 리스트 보유.
24     private ArrayList<ListViewItem> filteredItemList = listViewItemList;
25     Filter listFilter;
26     // ListViewAdapter의 생성자
27     public ListViewAdapter() {
28     }
29
30     // Adapter에 사용되는 데이터의 개수를 리턴
31     @Override
32     public int getCount() {
33         return filteredItemList.size();
34     }
35
36     // position에 위치한 데이터를 화면에 출력하는데 사용될 View를 리턴
37     @Override
38     public View getView(int position, View convertView, ViewGroup parent) {
39         final int pos = position; // 위치 값 받음
40         final Context context = parent.getContext();
41
42         // "listview_item" Layout을 inflate하여 convertView 참조 획득.
43         if (convertView == null) {
44             LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
45             convertView = inflater.inflate(R.layout.listview_item, parent, false);
46         }
47
48         // 화면에 표시될 View(Layout이 inflate된)으로부터 위젯에 대한 참조 획득
49         ImageView iconImageView = (ImageView) convertView.findViewById(R.id.list_imageview); // 이미지의 위치 값 참조 획득
50         TextView textTextView = (TextView) convertView.findViewById(R.id.list_textview1); // 첫번째 텍스트뷰의 위치 값 참조 획득
51         TextView textViewView = (TextView) convertView.findViewById(R.id.list_textview2); // 두번째 텍스트뷰의 위치 값 참조 획득
52
53         // Data Set(filteredItemList)에서 position에 위치한 데이터 참조 획득
54         ListViewItem listItem = filteredItemList.get(position);
55

```

```

56         // 아이템 내 각 위젯에 데이터 반영
57         iconImageView.setImageResource(listViewItem.getIcon()); // 이미지 값 반영
58         textViewText.setText(listViewItem.getText()); // 첫번째 텍스트뷰 텍스트 값 반영
59         textViewMassage.setText(listViewItem.getMessage()); // 두번째 텍스트뷰 텍스트 값
60         반영
61     }
62 }
63
64     // 지정한 위치(position)에 있는 데이터와 관계된 아이템(row)의 ID를 리턴
65     @Override
66     public long getItemId(int position) {
67         return position ;
68     }
69
70     // 지정한 위치(position)에 있는 데이터 리턴
71     @Override
72     public Object getItem(int position) {
73         return filteredItemList.get(position) ;
74     }
75
76     // 리스트뷰 아이템 데이터 추가를 위한 함수
77     public void addItem(Drawable icon, String text, String msg) {
78         ListViewItem item = new ListViewItem();
79         item.setIcon(icon); //이미지 데이터 추가
80         item.setText(text); //첫번째 텍스트 데이터 추가
81         item.setMessage(msg); //두번째 텍스트 데이터 추가
82         listViewItemList.add(item);
83     }
84
85     //getFilter() 함수를 override
86     @Override
87     public Filter getFilter() {
88         if (listFilter == null) {
89             listFilter = new ListFilter() ;
90         }
91         return listFilter ;
92     }
93
94     //Adapter 내부에 커스텀 Filter 클래스를 정의하고 검색창 기능구현
95     private class ListFilter extends Filter {
96         @Override
97         protected FilterResults performFiltering(CharSequence constraint) {
98             FilterResults results = new FilterResults() ;
99             if (constraint == null || constraint.length() == 0) { //내용물이 없을
100                 경우
101                     results.values = listViewItemList ;
102                     results.count = listViewItemList.size() ;
103             } else {
104                 ArrayList<ListViewItem> itemList = new ArrayList<ListViewItem>()
105                         >() ;
106                 for (ListViewItem item : listViewItemList) {
107                     if (item.getText().toUpperCase().contains(constraint.toString()
108                         .toUpperCase()) ||
109                         item.getMessage().toUpperCase().contains(constraint.
110                         toString().toUpperCase())))
111                     {
112                         itemList.add(item) ; //리스트뷰에 추가
113                     }
114                 }
115             }
116         }
117     }

```

```
File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/ListViewAdapter.java
111         results.values = itemList ;
112         results.count = itemList.size() ;
113     }
114     return results; // 결과값 리턴
115 }
116
117     @Override
118     protected void publishResults(CharSequence constraint, FilterResults
results) {
119
120         //리스트뷰에 데이터 업데이트
121         filteredItemList = (ArrayList<ListViewItem>) results.values ;
122
123         if (results.count > 0) {
124             notifyDataSetChanged() ;
125         } else {
126             notifyDataSetInvalidated() ;
127         }
128     }
129 }
130
131     //DB에서 가져온 값 아이템 데이터 추가를 위한 함수
132     public void addAuctionItem(Drawable icon, String text, String msg, String
uninum, String regnum) {
133         ListViewItem item = new ListViewItem();
134         item.setIcon(icon);
135         item.setText(text);
136         item.setMassage(msg);
137         item.setUninumber(uninum);
138         item.setRegnumber(regnum);
139         listViewItemList.add(item);
140     }
141 }
142
143
```

```

1 package com.example.teamauction;
2
3 import android.content.Context;
4 import android.graphics.drawable.Drawable;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.BaseAdapter;
9 import android.widget.Filter;
10 import android.widget.Filterable;
11 import android.widget.ImageView;
12 import android.widget.TextView;
13
14 import java.util.ArrayList;
15 /**
16 * 리스트뷰를 보여주는 클래스
17 * 리스트를 구성하는 요소와 보여주는 요소를 지정할 수 있음
18 */
19 public class ListViewAdapter extends BaseAdapter implements Filterable {
20     // Adapter에 추가된 데이터를 저장하기 위한 ArrayList
21     private ArrayList<ListViewItem> listViewItemList = new ArrayList<ListViewItem>();
22
23     // 필터링된 결과 데이터를 저장하기 위한 ArrayList. 최초에는 전체 리스트 보유.
24     private ArrayList<ListViewItem> filteredItemList = listViewItemList;
25     Filter listFilter;
26     // ListViewAdapter의 생성자
27     public ListViewAdapter() {
28     }
29
30     // Adapter에 사용되는 데이터의 개수를 리턴
31     @Override
32     public int getCount() {
33         return filteredItemList.size();
34     }
35
36     // position에 위치한 데이터를 화면에 출력하는데 사용될 View를 리턴
37     @Override
38     public View getView(int position, View convertView, ViewGroup parent) {
39         final int pos = position; // 위치 값 받음
40         final Context context = parent.getContext();
41
42         // "listview_item" Layout을 inflate하여 convertView 참조 획득.
43         if (convertView == null) {
44             LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
45             convertView = inflater.inflate(R.layout.listview_item, parent, false);
46         }
47
48         // 화면에 표시될 View(Layout이 inflate된)으로부터 위젯에 대한 참조 획득
49         ImageView iconImageView = (ImageView) convertView.findViewById(R.id.list_imageview); // 이미지의 위치 값 참조 획득
50         TextView textTextView = (TextView) convertView.findViewById(R.id.list_textview1); // 첫번째 텍스트뷰의 위치 값 참조 획득
51         TextView textViewView = (TextView) convertView.findViewById(R.id.list_textview2); // 두번째 텍스트뷰의 위치 값 참조 획득
52
53         // Data Set(filteredItemList)에서 position에 위치한 데이터 참조 획득
54         ListViewItem listItem = filteredItemList.get(position);
55

```

```

56         // 아이템 내 각 위젯에 데이터 반영
57         iconImageView.setImageResource(listViewItem.getIcon()); // 이미지 값 반영
58         textViewText.setText(listViewItem.getText()); // 첫번째 텍스트뷰 텍스트 값 반영
59         textViewMassage.setText(listViewItem.getMessage()); // 두번째 텍스트뷰 텍스트 값
60         반영
61     }
62 }
63
64     // 지정한 위치(position)에 있는 데이터와 관계된 아이템(row)의 ID를 리턴
65     @Override
66     public long getItemId(int position) {
67         return position ;
68     }
69
70     // 지정한 위치(position)에 있는 데이터 리턴
71     @Override
72     public Object getItem(int position) {
73         return filteredItemList.get(position) ;
74     }
75
76     // 리스트뷰 아이템 데이터 추가를 위한 함수
77     public void addItem(Drawable icon, String text, String msg) {
78         ListViewItem item = new ListViewItem();
79         item.setIcon(icon); //이미지 데이터 추가
80         item.setText(text); //첫번째 텍스트 데이터 추가
81         item.setMessage(msg); //두번째 텍스트 데이터 추가
82         listViewItemList.add(item);
83     }
84
85     //getFilter() 함수를 override
86     @Override
87     public Filter getFilter() {
88         if (listFilter == null) {
89             listFilter = new ListFilter() ;
90         }
91         return listFilter ;
92     }
93
94     //Adapter 내부에 커스텀 Filter 클래스를 정의하고 검색창 기능구현
95     private class ListFilter extends Filter {
96         @Override
97         protected FilterResults performFiltering(CharSequence constraint) {
98             FilterResults results = new FilterResults() ;
99             if (constraint == null || constraint.length() == 0) { //내용물이 없을
100                 경우
101                     results.values = listViewItemList ;
102                     results.count = listViewItemList.size() ;
103             } else {
104                 ArrayList<ListViewItem> itemList = new ArrayList<ListViewItem>()
105                         >() ;
106                 for (ListViewItem item : listViewItemList) {
107                     if (item.getText().toUpperCase().contains(constraint.toString()
108                         .toUpperCase()) ||
109                         item.getMessage().toUpperCase().contains(constraint.
110                         toString().toUpperCase())))
111                     {
112                         itemList.add(item) ; //리스트뷰에 추가
113                     }
114                 }
115             }
116         }
117     }

```

```
File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/ListViewAdapter.java
111         results.values = itemList ;
112         results.count = itemList.size() ;
113     }
114     return results; // 결과값 리턴
115 }
116
117     @Override
118     protected void publishResults(CharSequence constraint, FilterResults
results) {
119
120         //리스트뷰에 데이터 업데이트
121         filteredItemList = (ArrayList<ListViewItem>) results.values ;
122
123         if (results.count > 0) {
124             notifyDataSetChanged() ;
125         } else {
126             notifyDataSetInvalidated() ;
127         }
128     }
129 }
130
131     //DB에서 가져온 값 아이템 데이터 추가를 위한 함수
132     public void addAuctionItem(Drawable icon, String text, String msg, String
uninum, String regnum) {
133         ListViewItem item = new ListViewItem();
134         item.setIcon(icon);
135         item.setText(text);
136         item.setMassage(msg);
137         item.setUninumber(uninum);
138         item.setRegnumber(regnum);
139         listViewItemList.add(item);
140     }
141 }
142
143
```

```

1 package com.example.teamauction;
2
3 import androidx.activity.result.ActivityResult;
4 import androidx.activity.result.ActivityResultCallback;
5 import androidx.activity.result.ActivityResultLauncher;
6 import androidx.activity.result.contract.ActivityResultContracts;
7 import androidx.appcompat.app.AppCompatActivity;
8 import androidx.core.content.ContextCompat;
9 import androidx.recyclerview.widget.LinearLayoutManager;
10 import androidx.recyclerview.widget.RecyclerView;
11
12 import android.content.Intent;
13 import android.os.Bundle;
14 import android.util.Log;
15 import android.view.View;
16 import android.widget.Button;
17 import android.widget.ImageButton;
18 import android.widget.ImageView;
19 import android.widget.Toast;
20
21 import com.android.volley.RequestQueue;
22 import com.android.volley.Response;
23 import com.android.volley.toolbox.Volley;
24
25 import org.json.JSONArray;
26 import org.json.JSONException;
27 import org.json.JSONObject;
28
29 import java.util.ArrayList;
30
31 /**
32 * 메인화면
33 * 앱 실행시 첫화면
34 * 회원로그인 혹은 회원가입 페이지로 이동할 수 있게 해주는 엑티비티
35 */
36 public class MainActivity extends AppCompatActivity {
37     RecyclerView mRecyclerView = null; // 리사이클러뷰 정의
38     RecyclerViewAdapter mAdapter = null; // 리사이클러뷰 어댑터 정의
39     ArrayList<ListViewItem> mList = new ArrayList<ListViewItem>(); // 리사이클러뷰
40     안에 들어갈 객체 정의
41     private ActivityResultLauncher<Intent> mStartForResult; // 엑티비티 콜백
42     변수, intent에 대한 응답을 받아줌
43     private int last_pos; // 리사이클러뷰 안에 정의된 체크박스의
44     마지막 위치 변수
45     private GameAccountInfo accountInfo; // 계정 정보에 대한 객체
46
47     @Override
48     protected void onCreate(Bundle savedInstanceState) {
49         super.onCreate(savedInstanceState);
50         setContentView(R.layout.activity_main);
51
52         Intent account_info = getIntent(); // intent를 받아옴
53         // account_info라는 key로 GameAccountInfo 객체를 받아옴
54         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info");
55
56         //액티비티 콜백 함수
57         mStartForResult = registerForActivityResult(
58             new ActivityResultContracts.StartActivityForResult(),
59             new ActivityResultCallback<ActivityResult>() {

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainActivity.java

57         @Override
58         public void onActivityResult(ActivityResult result) {
59             // 다른 액티비티에서 받아온 결과
60             if (result.getResultCode() == RESULT_OK) {
61                 Intent intent = result.getData();
62                 // intent로 부터 데이터를 가져옴
63                 int code_get = intent.getIntExtra("result", 0); // 그
64                 데이터 안의 result라는 key로 정의된 데이터를 꺼냄
65
66                 if (code_get == 0){ // 그 코드가 0이라면
67                     // 토스트 메세지를 띄움
68                     Toast.makeText(MainActivity.this, getString(R.
69                         string.cancel_logout), Toast.LENGTH_SHORT).show();
70                 }
71                 else if (code_get == 1){ // 그 코드가 1이라면
72                     // 토스트 메세지를 띄우고
73                     Toast.makeText(MainActivity.this, getString(R.
74                         string.confirm_logout), Toast.LENGTH_SHORT).show();
75                     // 로그인 화면으로 갈 준비를 한 후
76                     Intent back_login_page = new Intent(MainActivity.
77                         this, MainLoginActivity.class);
78                     startActivity(back_login_page); // intent 실행
79                     finish(); // 그리고 현재 액티비티를 종료
80                 }
81                 else if (code_get == 2){ // 그 코드가 2이라면
82                     // 토스트 메세지를 띄움
83                     Toast.makeText(MainActivity.this, getString(R.
84                         string.cancel_access_account), Toast.LENGTH_SHORT).show();
85                 }
86                 else if (code_get == 3){ // 그 코드가 3이라면
87                     // 토스트 메세지를 띄움
88                     Toast.makeText(MainActivity.this, getString(R.
89                         string.confirm_access_account), Toast.LENGTH_SHORT).show();
90                 }
91             }
92             else { // 그 외의 코드가 들어왔다면
93                 }
94             }
95         }
96     });
97     go_logout.setOnClickListener(new View.OnClickListener() { // 뒤로가기 버튼
98         @Override
99         public void onClick(View v) { // onClick 함수 재정의
100             // MainActivity에서 PopupActivity를 띄우는 intent 준비
101             Intent intent = new Intent(MainActivity.this, PopupActivity.class
102             );
103             intent.putExtra("data", getString(R.string.wanttologout)); // intent에 데이터를 넣음
104             mStartForResult.launch(intent); // mStartForResult를 통한 intent
105             실행
106         }
107     });
108
109     ImageView add_account = findViewById(R.id.add_account); // 새로운 연동
110     계정 추가 버튼
111     add_account.setOnClickListener(new View.OnClickListener() { // 계정연동 버튼
112

```

```

102    클릭 이벤트 생성
103        @Override
104            public void onClick(View view) { // onClick 함수 재정의
105                // MainActivity에서 GameListActivity으로 가는 intent 준비
106                Intent intent = new Intent(MainActivity.this, GameListActivity.
107                class);
107                intent.putExtra("account_info", accountInfo); // intent에 데이터를
108                넣음
108                startActivity(intent); // intent 실행
109            }
110        );
111
112        // 리사이클러뷰에 LinearLayoutManager 객체 지정.
113        mRecyclerView = findViewById(R.id.linked_id_list);
114        mRecyclerView.setLayoutManager(new LinearLayoutManager(this)); // 생성 위치 지정
115        mRecyclerView.addItemDecoration(new RecyclerViewDecoration(10)); // 높이 지정
116
117        // 리사이클러뷰에 SimpleTextAdapter 객체 지정.
118        RecyclerViewAdapter mAdapter = new RecyclerViewAdapter(mList);
119        mRecyclerView.setAdapter(mAdapter); // 어댑터 설정
120        mAdapter.setOnItemClickListener(new RecyclerViewAdapter.
120        OnItemClickListener() {
121            @Override
122                public void onItemClick(View v, int position) { // onItemClick 함수
122                재정의
123                    //Toast.makeText(v.getContext(), "touched"+position, Toast.
123                    LENGTH_SHORT).show();
124                    // PopupActivity를 띄우는 intent 생성
125                    Intent intent = new Intent(MainActivity.this, PopupActivity.class
125                    );
126                    intent.putExtra("data", getString(R.string.access_account));
126                    // intent에 데이터 추가
127                    mStartForResult.launch(intent); // mStartForResult를 통한
127                    intent 실행
128                }
129            });
130
131        Button go_login = findViewById(R.id.login_account); // 로그인 버튼
132        go_login.setOnClickListener(new View.OnClickListener() { // 로그인 버튼
132        클릭 이벤트 생성
133            @Override
134                public void onClick(View view) { // onClick 함수
134                재정의
135                    last_pos = RecyclerViewAdapter.lastCheckedPos; // 마지막 위치를 받음
136                    ListViewItem item = mList.get(last_pos); // 그 위치로 데이터를
136                    가져옴
137                    // 토스트 메세지를 띄워주고
138                    Toast.makeText(MainActivity.this, item.getText()+" " +
139                        item.getMessage() +" data access", Toast.LENGTH_SHORT).
139                    show();
140                    // AuctionScreen으로 갈 intent 생성 후
141                    Intent intent = new Intent(MainActivity.this, AuctionScreen.class
141                    );
142                    accountInfo.setGameName(item.getText()); // GameAccountInfo 객체에 게임이름 저장
143                    accountInfo.setCharacterName(item.getMessage()); // GameAccountInfo 객체에 게임 캐릭터이름 저장
144                    intent.putExtra("account_info", accountInfo); // intent에 데이터

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainActivity.java

```

144 추가
145         startActivity(intent);      // AuctionScreen 실행
146     }
147 }
148
149     // volley 통신을 수행하려고 응답대기를 준비
150     Response.Listener<String> responseListener = new Response.Listener<String>
151     >() {
152         @Override
153         public void onResponse(String response) {          // onResponse 함수
154             재정의
155             try {    // 만약 반응이 왔고 정상적이라면,
156                 boolean CheckFirst = true;                  // 첫번째 데이터인 경우
157                 미리 체크해두기 위한 boolean
158                 JSONObject jsonResponse = new JSONObject(response)
159             );           // json 형태로 받기 위해 객체 준비
160                 JSONArray Ids = jsonResponse.getJSONArray("MyGameList"); // MyGameList key 값으로 배열을 받음
161
162                 for (int i = 0; i < Ids.length(); i++) {        // Ids 길이만큼
163                     반복문 수행
164                     JSONObject item = Ids.getJSONObject(i); // Ids의
165                     데이터를 가져옴
166                     String gameName = item.getString("gameName");
167                     // gameName 데이터 저장
168                     String gameNickname = item.getString("gameNickname");
169                     // gameNickname 데이터 저장
170                     데이터 추가
171                     CheckFirst = false; // 첫번째 수행 후에는 다 false
172                     }
173                     mAdapter.notifyDataSetChanged(); // 반복문 수행 후에 리스트가
174                     업데이트 됨을 알림
175                     } catch (JSONException e) { // 접속 오류가 난 것이라면
176                         e.printStackTrace(); // 오류 추적
177                         // 토스트 메세지를 띄워줌
178                         Toast.makeText(getApplicationContext(), "no connection",
179                         Toast.LENGTH_SHORT).show();
180                     }
181                 }
182             }
183         }
184     }
185     RequestQueue queue = Volley.newRequestQueue(MainActivity.this); // 큐를
186     생성
187     queue.add(validateRequest); // 큐에 추가
188 }
189
190     public void addItem(String title, String desc, Boolean checkbox) { // 리사이클러뷰에 데이터 추가를 위한 함수
191         ListViewItem item = new ListViewItem(); // ListViewItem 클래스로부터 객체
192         생성
193         item.setIcon(icon); // 이미지 아이콘
194         item.setText(title); // 텍스트1(게임이름)
195         item.setMassage(desc); // 텍스트2(게임캐릭터이름)
196         item.setSelected(checkbox); // 체크박스

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainActivity.java

```
187
188     mList.add(item);           // 리스트에 아이템 추가
189 }
190
191 }
```

File - /Users/shin/AndroidStudioProjects/Team Auction/app/src/main/java/com/example/team auction/MainLoginActivity.java

```
1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.ImageButton;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import com.android.volley.RequestQueue;
15 import com.android.volley.Response;
16 import com.android.volley.toolbox.Volley;
17
18 import org.json.JSONException;
19 import org.json.JSONObject;
20
21 /**
22 * 회원 로그인화면 액티비티
23 * 회원은 로그인하여 경매장 서비스를 이용할 수 있다
24 * 계정이 기억나지 않는 경우 비밀번호 찾기 액티비티로 이동 가능하고
25 * 계정이 없을 경우 계정 만들기 액티비티로 넘어갈 수 있다
26 */
27 public class MainLoginActivity extends AppCompatActivity {
28
29     private EditText login_id, login_pw; // 로그인을 위한 아이디와 비밀번호를 입력받는
필드
30     private GameAccountInfo account_data; // 계정 정보를 갖고 있기 위한 객체
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_login);
36
37         login_id = findViewById(R.id.login_idbox); // 아이디 입력 필드
38         login_pw = findViewById(R.id.login_pwbox); // 비밀번호 입력 필드
39
40         ImageButton back_button = findViewById(R.id.back_login); // 뒤로가기를 위한
이미지 버튼
41         back_button.setOnClickListener(new View.OnClickListener() { // 이미지 버튼에
대한 이벤트 생성
42             @Override
43             public void onClick(View view) { // onClick 함수 재정의
44                 // MainLoginActivity에서 MainStartActivity으로 가기 위한 intent 생성
45                 Intent intent = new Intent(MainLoginActivity.this,
MainStartActivity.class);
46                 startActivity(intent); // intent 실행
47                 finish(); // 현재 화면 종료
48             }
49         });
50         account_data = new GameAccountInfo(); // GameAccountInfo 객체 생성
51         Button button = findViewById(R.id.login_button); // 로그인 버튼
52         button.setOnClickListener(new View.OnClickListener() { // 로그인 버튼에 대한
이벤트 생성
53             @Override
54             public void onClick(View view) { // onClick 함수 재정의
55             }
56         });
57     }
58 }
```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainLoginActivity.java

56         String userID = login_id.getText().toString();           // 아이디
57         입력필드 String 형태로 가져오기
58         String userPW = login_pw.getText().toString();           // 비밀번호
59         입력필드 String 형태로 가져오기
60
61         if(userID.equals("")) {          // 아이디 입력필드가 비어있다면
62             // 토스트 메세지를 띄워주고
63             Toast.makeText(getApplicationContext(), "아이디를 입력해주세요",
64             Toast.LENGTH_SHORT).show();
64             return; // 이벤트 종료
65         }
66         // volley 통신을 수행하려고 응답대기를 준비
67         Response.Listener<String> responseListener = new Response.
68         Listener<String>() {
69             @Override
70             public void onResponse(String response) {           //
71                 // onResponse 함수 재정의
72                 try { // 만약 반응이 왔고 정상적이라면,
73                     JSONObject jsonObject = new JSONObject(response
74 );           //json 형태로 받기 위해 객체 준비
75                     boolean success = jsonObject.getBoolean("success"
76 ); // success라는 key 값으로 boolean value를 받음
77
78                     if (success) { // success가 참이라면
79                         String jsonID = jsonObject.getString("userID"
80 );           // jsonObject 객체로부터 userID를 가져옴
81                         String jsonPW = jsonObject.getString("userPassword");
82                         // jsonObject 객체로부터 userPassword 가져옴
83                         // 토스트 메세지를 띄워주고
84                         Toast.makeText(getApplicationContext(), jsonID+
85                             "님 환영합니다", Toast.LENGTH_SHORT).show();
86                         // MainLoginActivity에서 MainActivity으로 가는
87                         Intent intent = new Intent(MainLoginActivity.this
88 , MainActivity.class);
89                         account_data.setLoginID(jsonID); // 아이디를
90                         account_data.setLoginPW(jsonPW); // 비밀번호를
91                         intent.putExtra("account_info", account_data);
92                         startActivity(intent); //>MainActivity 실행
93                         finish(); // 현재 액티비티 종료
94
95                     } else { // success가 false일 때
96                         // 토스트 메세지를 띄워주고
97                         Toast.makeText(getApplicationContext(), "서버와
연결이 끊겼습니다", Toast.LENGTH_SHORT).show();
98                         return; // 이벤트 종료
99                     }
100
101                 } catch (JSONException e) { // 로그인 오류가 난 것이라면
102                     e.printStackTrace(); // 에러 추적하고
103                     // 토스트 메세지를 띄워준다
104                     Toast.makeText(getApplicationContext(), "등록되지 않은
계정입니다", Toast.LENGTH_SHORT).show();
105                 }
106             }
107         }

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainLoginActivity.java

```
98         };
99
100        String purl = "http://ualsgur98.dothome.co.kr/Login.php"
101           ;                                // 통신할 php 주소
102           PHPRequest validateRequest = new PHPRequest( purl, userID, userPW
103 , responseListener);    // 데이터 전송을 위한 데이터 세팅
104           RequestQueue queue = Volley.newRequestQueue(MainLoginActivity.
105 this);                                // 큐를 생성
106           queue.add(validateRequest); // 큐에 추가
107       }
108   });
109
110  TextView return_signup_textview = findViewById(R.id.return_signup);
111  // 회원가입 텍스트뷰
112  return_signup_textview.setOnClickListener(new View.OnClickListener() {
113     // 텍스트뷰 클릭 이벤트 생성
114     @Override
115     public void onClick(View view) {          // onClick 함수 재정의
116         // MainLoginActivity에서 MainSignupActivity으로 가는 intent 생성
117         Intent intent = new Intent(MainLoginActivity.this,
118         MainSignupActivity.class);
119         startActivity(intent);                // intent 실행
120         finish();                           // 현재 액티비티 종료
121     }
122   });
123
124  TextView return_find_password = findViewById(R.id.go_fpassword_text);
125  // 비밀번호 찾기 텍스트뷰
126  return_find_password.setOnClickListener(new View.OnClickListener() {
127     // 텍스트뷰 클릭 이벤트 생성
128     @Override
129     public void onClick(View view) {          // onClick 함수 재정의
130         // MainLoginActivity에서 MainPasswordActivity으로 가는 intent 생성
131         Intent intent = new Intent(MainLoginActivity.this,
132         MainPasswordActivity.class);
133         startActivity(intent);                // intent 실행
134         finish();                           // 현재 액티비티 종료
135     }
136   });
137 }
```

```

1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.ImageButton;
10 /**
11 * 비밀번호 찾는 액티비티
12 * 비밀번호가 기억나지 않는 경우 자신이 등록했던 이메일로 비밀번호를 재발급 받을 수 있다
13 */
14 public class MainPasswordActivity extends AppCompatActivity {
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_password);
20
21         ImageButton back_button = findViewById(R.id.back_password); // 뒤로가기
22         버튼
23         back_button.setOnClickListener(new View.OnClickListener() { // 버튼 클릭
24             이벤트 생성
25             @Override
26             public void onClick(View view) { // onClick 함수 재정의
27                 // MainPasswordActivity에서 MainLoginActivity으로 가는 intent 생성
28                 Intent intent = new Intent(MainPasswordActivity.this,
29                 MainLoginActivity.class);
30                 startActivity(intent); // intent 실행
31                 finish(); // 현재 액티비티 종료
32             }
33         });
34
35         Button send_email = findViewById(R.id.find_pw_button); // 이메일
36         전송 버튼
37         send_email.setOnClickListener(new View.OnClickListener() { // 버튼 클릭
38             이벤트 생성
39             @Override
40             public void onClick(View view) { // onClick 함수 재정의
41                 // MainPasswordActivity에서 MainLoginActivity으로 가는 intent 생성
42                 Intent intent = new Intent(MainPasswordActivity.this,
43                 MainLoginActivity.class);
44                 startActivity(intent); // intent 실행
45                 finish(); // 현재 액티비티 종료
46             }
47         });
48     }
49 }

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainSignupActivity.java

```
1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.app.AlertDialog;
6 import android.content.Intent;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ImageButton;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import com.android.volley.RequestQueue;
16 import com.android.volley.Response;
17 import com.android.volley.toolbox.Volley;
18
19 import org.json.JSONException;
20 import org.json.JSONObject;
21
22 /**
23 * 회원가입 액티비티
24 * 사용자가 회원가입을 진행할 수 있다
25 */
26 public class MainSignupActivity extends AppCompatActivity {
27
28     // 회원가입에 필요한 데이터를 입력받는 필드들
29     private EditText join_email, join_password, join_pwck, join_name, join_phone,
30     join_id;
31     // 아이디 중복검사시 성공 여부를 받는 boolean 값
32     private boolean validate = false;
33     // 경고창을 띄우는 변수
34     private AlertDialog dialog;
35
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39         setContentView(R.layout.activity_signup);
40
41         join_name = findViewById(R.id.signup_name); // 이름을 받는 필드
42         join_phone = findViewById(R.id.signup_phone); // 핸드폰 번호를 받는
43         필드
44         join_email = findViewById( R.id.signup_email ); // 이메일을 받는 필드
45         join_id = findViewById(R.id.signup_id); // 아이디를 받는 필드
46         join_password = findViewById( R.id.signup_password ); // 비밀번호를 받는 필드
47         join_pwck = findViewById(R.id.confirm_pwbox); // 비밀번호 재확인하는
48         필드
49
50         ImageButton back_button = findViewById(R.id.back_signup); // 이미지 버튼을
51         정의
52         back_button.setOnClickListener(new View.OnClickListener() { // 이미지 버튼을
53             클릭시 이벤트를 활성화 시킴
54             @Override
55             public void onClick(View view) { // onClick을 재정의 하여 이벤트 수행
56                 // 이전 화면으로 가도록 하는 intent
57                 Intent intent = new Intent(MainSignupActivity.this,
58                 MainStartActivity.class);
59                 startActivity(intent); // intent 실행
60                 finish(); // 현재 액티비티는 종료시킴
61             }
62         });
63     }
64 }
```

```

55         }
56     });
57
58     Button checkIdButton = findViewById(R.id.check_id);           // 아이디 중복검사를
      수행하는 버튼
59     checkIdButton.setOnClickListener(new View.OnClickListener() {    // 중복검사
      버튼 클릭시 이벤트를 활성화
60         @Override
61         public void onClick(View view) {        // onClick 함수를 재정의 함
62             String userID = join_id.getText().toString();    // 아이디를
      String으로 받음
63             if (validate) { return; }           // 만약 validate가 참이라면 이벤트 종료
64             if (userID.equals("")) {          // 만약 값이 비어있다면
65                 // 알림창 준비
66                 AlertDialog.Builder builder = new AlertDialog.Builder(
      MainSignupActivity.this);
67                 // 알림창 세팅
68                 dialog = builder.setMessage("아이디를 입력하세요.");
      setPositiveButton("확인", null).create();
69                 // 알림창 띄우기
70                 dialog.show();
71                 // 그리고 이벤트 종료
72             }
73         }
74         Response.Listener<String> responseListener = new Response.
      Listener<String>() { // volley 통신을 수행하려고 응답대기를 준비
75             @Override
76             public void onResponse(String response) { // onResponse
      재정의
77                 try { // 만약 반응이 왔고 정상적이라면,
78                     JSONObject jsonResponse = new JSONObject(response
      );
59                 //json 형태로 받기 위해 객체 준비
79                     boolean success = jsonResponse.getBoolean("success"
      );
59                 // success라는 key 값으로 boolean value를 받음
80
81                     if (success) { // success가 참이라면
82                         // 알림창 준비
83                         AlertDialog.Builder builder = new AlertDialog.
      Builder(MainSignupActivity.this);
84                         // 알림창 세팅
85                         dialog = builder.setMessage("사용할 수 있는
      아이디입니다.").setPositiveButton("확인", null).create();
86                         // 알림창 띄우기
87                         dialog.show();
88                         join_id.setEnabled(false); //아이디값 고정
89                         checkIdButton.setVisibility(View.GONE); //버튼
      비활성화
90                         validate = true;           //검증 완료
91                     } else {
92                         // 알림창 준비
93                         AlertDialog.Builder builder = new AlertDialog.
      Builder(MainSignupActivity.this);
94                         // 알림창 세팅
95                         dialog = builder.setMessage("이미 존재하는 아이디입니다
      .").setNegativeButton("확인", null).create();
96                         // 알림창 띄우기
97                         dialog.show();
98                         // 입력 필드 비워주기
99                         join_id.setText(null);
100                    }
}

```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainSignupActivity.java

101             } catch (JSONException e) { // 반응이 왔으나 오류라면,
102                 e.printStackTrace(); // 에러를 출력해줌
103             }
104         }
105     };
106     String purl = "http://ualsgur98.dothome.co.kr/UserValidate.php"
107 ;           // 통신할 php 주소
108     PHPRequest validateRequest = new PHPRequest( purl, userID,
109     responseListener); // 데이터 전송을 위한 데이터 세팅
110     RequestQueue queue = Volley.newRequestQueue(MainSignupActivity.
111     this); // 큐를 생성
112     queue.add(validateRequest); // 큐에 추가
113 }
114
115     Button SignUpButton = findViewById(R.id.signup_button); // 회원가입
116     버튼
117     SignUpButton.setOnClickListener(new View.OnClickListener() { // 회원가입
118         버튼을 눌렀을 때 이벤트 생성
119         @Override
120         public void onClick(View view) {
121             String userName = join_name.getText().toString(); // 이름
122             입력 필드에서 String 형태로 데이터를 가져옴
123             String userPhone = join_phone.getText().toString(); // 핸드폰번호 입력 필드에서 String 형태로 데이터를 가져옴
124             String userEmail = join_email.getText().toString(); // 이메일
125             입력 필드에서 String 형태로 데이터를 가져옴
126             String userID = join_id.getText().toString(); // 아이디
127             입력 필드에서 String 형태로 데이터를 가져옴
128             String userPass = join_password.getText().toString(); // 비밀번호
129             입력 필드에서 String 형태로 데이터를 가져옴
130             String userPwck = join_pwck.getText().toString(); // 비밀번호
131             검증 입력 필드에서 String 형태로 데이터를 가져옴
132
133             Response.Listener<String> responseListener = new Response.
134             Listener<String>() { // volley 통신을 수행하려고 응답대기를 준비
135                 @Override
136                 public void onResponse(String response) { // onResponse 재정의
137                     try {
138                         JSONObject json0bject = new JSONObject(response
139 ); // json 형태로 받기 위해 객체 준비
140                         boolean success = json0bject.getBoolean("success"
141 ); // success라는 key 값으로 boolean value를 받음
142
143                         if (success) { // success가 참이라면
144                             if(userPass.equals(userPwck)){ // 비밀번호 검증 입력
145                                 필드에서 비밀번호와 같다면
146                                     // 토스트 메세지를 띄워주고
147                                     Toast.makeText(getApplicationContext(), "계정이
148                                     정상적으로 등록되었습니다", Toast.LENGTH_SHORT).show();
149
150                                     // MainStartActivity로 가는 Intent를 생성
151                                     Intent intent = new Intent(MainSignupActivity
152 .this, MainStartActivity.class);
153                                     startActivity(intent); // 새 액티비티를 띄워줌
154                                     finish(); // 그리고 현재 액티비티는 종료
155
156                                 }
157                                 else{ // 만약 비밀번호 검증 필드에서 같지
158                                     않았다면
159                                     // 토스트 메세지를 띄워주고
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
221
```

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/MainSignupActivity.java

142                     Toast.makeText(getApplicationContext(), "비밀번호가 같지 않습니다", Toast.LENGTH_SHORT).show();
143                     return; // 이벤트 종료
144                 }
145             }
146         } else { // php로 부터 success가 도착하지 않았다면
147             // 토스트 메세지를 띄워주고
148             Toast.makeText(getApplicationContext(), "계정 생성에 실패했습니다", Toast.LENGTH_SHORT).show();
149             return; // 이벤트 종료
150         }
151     }
152 } catch (JSONException e) { // 만약 중간에 오류가 생겼다면,
153     e.printStackTrace(); // 오류 추적
154 }
155 }
156 }
157 };
158
159 String purl = "http://ualsgur98.dothome.co.kr/Register.php";
// 통신할 php 주소
160 // 데이터 전송을 위한 데이터 세팅
161 PHPRequest registerRequest = new PHPRequest(purl, userName,
userPhone, userEmail, userID, userPass, responseListener);
162 RequestQueue queue = Volley.newRequestQueue( MainSignupActivity.
this ); // 큐를 생성
163 queue.add( registerRequest ); // 큐에 추가
164 }
165 });
166
167 TextView goLoginButton = findViewById(R.id.go_login_txt); // 로그인화면으로 바로 이동
168 goLoginButton.setOnClickListener(new View.OnClickListener() { // 텍스트뷰에 터치 이벤트를 생성
169     @Override
170     public void onClick(View view) { // onClick 함수 재정의
171         // MainSignupActivity에서 MainLoginActivity으로 가는 intent 생성
172         Intent intent = new Intent(MainSignupActivity.this,
MainLoginActivity.class);
173         startActivity(intent); // intent 실행
174         finish(); // 현재 액티비티 종료
175     }
176 });
177 }
178 }

```

```

1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9
10 /**
11 * 앱 실행시 첫화면
12 * 로그인과 회원가입을 선택할 수 있는 액티비티
13 */
14 public class MainStartActivity extends AppCompatActivity {
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_start);
20
21         Button goLoginButton = findViewById(R.id.go_login); // 로그인화면 전환 버튼
22         goLoginButton.setOnClickListener(new View.OnClickListener() { // 로그인 전환
버튼 클릭시 이벤트 발생
23             @Override
24             public void onClick(View view) { // 버튼에 대한 함수 재정의
25                 // Intent로 액티비티 간 데이터 전송
26                 Intent intent = new Intent(MainStartActivity.this,
MainLoginActivity.class);
27                 startActivity(intent); // 로그인 액티비티 실행
28             }
29         });
30
31         Button goSignupButton = findViewById(R.id.go_signup); // 회원가입 화면 전환
버튼
32         goSignupButton.setOnClickListener(new View.OnClickListener() { // 회원가입
버튼 클릭시 이벤트 발생
33             @Override
34             public void onClick(View view) { // 버튼에 대한 함수 재정의
35                 // Intent로 액티비티 간 데이터 전송
36                 Intent intent = new Intent(MainStartActivity.this,
MainSignupActivity.class);
37                 startActivity(intent); // 회원가입 액티비티 실행
38             }
39         });
40     }
41 }

```

```
1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9
10 /**
11 * 앱 실행시 첫화면
12 * 로그인과 회원가입을 선택할 수 있는 액티비티
13 */
14 public class MainStartActivity extends AppCompatActivity {
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_start);
20
21         Button goLoginButton = findViewById(R.id.go_login); // 로그인화면 전환 버튼
22         goLoginButton.setOnClickListener(new View.OnClickListener() { // 로그인 전환
23             버튼 클릭시 이벤트 발생
24             @Override
25                 public void onClick(View view) { // 버튼에 대한 함수 재정의
26                     // Intent로 액티비티 간 데이터 전송
27                     Intent intent = new Intent(MainStartActivity.this,
28                         MainLoginActivity.class);
29                     startActivity(intent); // 로그인 액티비티 실행
30                 }
31         });
32
33         Button goSignupButton = findViewById(R.id.go_signup); // 회원가입 화면 전환
34         버튼
35             goSignupButton.setOnClickListener(new View.OnClickListener() { // 회원가입
36                 버튼 클릭시 이벤트 발생
37                 @Override
38                     public void onClick(View view) { // 버튼에 대한 함수 재정의
39                         // Intent로 액티비티 간 데이터 전송
40                         Intent intent = new Intent(MainStartActivity.this,
41                             MainSignupActivity.class);
42                         startActivity(intent); // 회원가입 액티비티 실행
43                     }
44             });
45     }
46 }
```

```

1 package com.example.teamauction;
2
3 import com.android.volley.AuthFailureError;
4 import com.android.volley.Response;
5 import com.android.volley.toolbox.StringRequest;
6
7 import java.util.HashMap;
8 import java.util.Map;
9
10 /**
11 * php 통신을 할 때 쓰는 클래스
12 * 생성자를 통해 여러 형태로 데이터를 저장할 수 있다
13 */
14 public class PHPRequest extends StringRequest {
15
16     private Map<String, String> map;
17
18     // 계정 추가
19     public PHPRequest(String URL, String userID, String userPW, String gameName,
20                       String gameID, String gamePW, String userCh, Response.
21 Listener<String> listener) {
22         super(Method.POST, URL, listener, null);
23
24         map = new HashMap<>();           // 해시맵 생성
25         map.put("userID", userID);       // 계정 아이디 추가
26         map.put("userPassword", userPW); // 계정 비밀번호 추가
27         map.put("gameName", gameName);   // 게임 이름 추가
28         map.put("gameID", gameID);      // 게임 아이디 추가
29         map.put("gamePW", gamePW);      // 게임 비밀번호 추가
30         map.put("gameNickname", userCh); // 게임 캐릭터이름 추가
31     }
32
33     // 회원가입
34     public PHPRequest(String URL, String userName, String userPhone, String
35     userEmail, String userID, String userPassword, Response.Listener<String> listener
36     ) {
37         super(Method.POST, URL, listener, null);
38
39         map = new HashMap<>();           // 해시맵 생성
40         map.put("userName", userName);   // 계정 이름 추가
41         map.put("userPhone", userPhone); // 계정 전화번호 추가
42         map.put("userEmail", userEmail); // 계정 이메일 추가
43         map.put("userID", userID);      // 계정 아이디 추가
44         map.put("userPassword", userPassword); // 계정 비밀번호 추가
45     }
46
47     // 게임 넘겨주기
48     public PHPRequest(String URL, String gameName, String gameID, String gamePW,
49     Response.Listener<String> listener) {
50         super(Method.POST, URL, listener, null);
51
52         map = new HashMap<>();           // 해시맵 생성
53         map.put("gameName", gameName);   // 계정 아이디 추가
54         map.put("gameID", gameID);      // 게임 아이디 추가
55         map.put("gamePW", gamePW);      // 게임 비밀번호 추가
56     }
57
58     // 로그인, 판매중인, 내아이템
59     public PHPRequest(String URL, String userID, String userPW, Response.Listener<
60     String> listener) {

```

```
56         super(Method.POST, URL, listener, null);
57
58         map = new HashMap<>();           // 해시맵 생성
59         map.put("userID", userID);      // 계정 아이디 추가
60         map.put("userPassword", userPW); // 계정 비밀번호 추가
61     }
62
63     // 아이디 검사
64     public PHPRequest(String URL, String userID, Response.Listener<String>
65     listener) {
66         super(Method.POST, URL, listener, null);
67
68         map = new HashMap<>();           // 해시맵 생성
69         map.put("userID", userID);      // 계정 아이디 추가
70     }
71
72     // 게임 목록
73     public PHPRequest(String URL, Response.Listener<String> listener) {
74         super(Method.POST, URL, listener, null);
75
76         map = new HashMap<>();           // 해시맵 생성
77     }
78
79     // 구매하기 기능
80     public PHPRequest(String URL, String GameName, String RegisterNumber, String
81     ItemQuantity, String Buyer, Response.Listener<String> listener ) {
82         super(Method.POST, URL, listener, null);
83         map = new HashMap<>();           // 해시맵 생성
84         map.put("gameName", GameName); // 게임 닉네임 추가
85         map.put("RegisterNumber", RegisterNumber); // RegisterNumber 추가
86         map.put("Quantity", ItemQuantity); // 수량 추가
87         map.put("Buyer", Buyer); // 구매자 추가
88     }
89
90     //public PHPRequest(String URL, String )
91     @Override
92     protected Map<String, String>getParams() throws AuthFailureError {
93         return map;           // 해시맵 리턴
94     }
```

```

1 package com.example.teamauction;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.util.Log;
8 import android.view.MotionEvent;
9 import android.view.View;
10 import android.view.Window;
11 import android.widget.Button;
12 import android.widget.TextView;
13
14 /**
15 * 팝업창을 구성하는 액티비티형 dialog
16 * 원하는 항목을 집어넣고 원하는 결과를 유도할 수 있다
17 */
18 public class PopupActivity extends Activity {
19
20     TextView txtText; // 어떤 알림인지 알려주기 위한 텍스트 뷰
21     Button yes_btn, no_btn; // 확인, 취소 버튼
22     int request_num_yes, request_num_no; // 보내줄 intent 값
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26
27         //타이틀바 없애기
28         requestWindowFeature(Window.FEATURE_NO_TITLE);
29         setContentView(R.layout.activity_popup);
30
31         //UI 객체생성
32         txtText = findViewById(R.id.txtText);
33         yes_btn = findViewById(R.id.check_yes);
34         no_btn = findViewById(R.id.check_no);
35
36         //데이터 가져오기
37         Intent intent = getIntent();
38         String data = intent.getStringExtra("data");
39         txtText.setText(data);
40
41         if(data.equals(getString(R.string.wanttologout))) {
42             request_num_yes = 1; // 로그아웃할 때
43             request_num_no = 0; // 로그아웃 취소할 때
44         }
45         else if(data.equals(getString(R.string.access_account))) {
46             request_num_yes = 3; // 로그인 할 때
47             request_num_no = 2; // 로그인 취소할 때
48         }
49
50         yes_btn.setOnClickListener(new View.OnClickListener() { // 이벤트 생성
51             @Override // onClick 재정의 해서 함수 실행
52             public void onClick(View view) {
53                 mOnClose(request_num_yes);
54             }
55         });
56
57         no_btn.setOnClickListener(new View.OnClickListener() { // 이벤트 생성
58             @Override // onClick 재정의 해서 함수 실행
59             public void onClick(View view) {
60                 mOnClose(request_num_no);

```

```
61         }
62     });
63
64     }
65
66     //확인 버튼 클릭
67     public void mOnClose(int i){
68         //데이터 전달하기
69         Intent intent = new Intent();
70         intent.putExtra("result", i);
71         setResult(RESULT_OK, intent);
72
73         //액티비티(팝업) 닫기
74         finish();
75     }
76
77     @Override
78     public boolean onTouchEvent(MotionEvent event) {
79         //바깥레이어 클릭시 안닫히게
80         if(event.getAction()== MotionEvent.ACTION_OUTSIDE){
81             return false;
82         }
83         return true;
84     }
85
86     @Override
87     public void onBackPressed() {
88         //안드로이드 백버튼 막기
89         return;
90     }
91
92 }
```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/RecyclerViewAdapter.java

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/RecyclerViewAdapter.java

59                     int pos = getAdapterPosition();           // 현재 위치 받음
60                     if (pos != RecyclerView.NO_POSITION) { // 리사이클러뷰 내의
61                         위치라면
62                         if (mListener!=null) { mListener.onItemClick(view, pos
63                         );
64                     }
65                     deleteBox.setOnClickListener(new View.OnClickListener() { // 삭제
66                         버튼 클릭 이벤트 생성
67                         @Override
68                         public void onClick(View view) {           // onClick 함수 재정의
69                             // 알림창 생성
70                             AlertDialog.Builder oDialog = new AlertDialog.Builder(view.
71                             getContext());
72                             // 알림창 구성
73                             oDialog.setMessage("정말로 계정을 삭제하시겠어요?")
74                             .setTitle("계정 삭제 알림")
75                             .setPositiveButton("아니오", new DialogInterface.
76                             OnClickListener() {
77                                 @Override
78                                 public void onClick(DialogInterface dialog, int
79                                 which) {
80                                     Toast.makeText(view.getContext(), "계정 삭제를
81                                     취소했습니다", Toast.LENGTH_LONG).show();
82                                 }
83                             })
84                             .setNeutralButton("예", new DialogInterface.
85                             OnClickListener() {
86                                 public void onClick(DialogInterface dialog, int
87                                 which) {
88                                     Toast.makeText(view.getContext(), "계정을
89                                     삭제했습니다", Toast.LENGTH_LONG).show();
90                                     removeAt(getAdapterPosition());
91                                 }
92                             })
93                             .setCancelable(false) // 백버튼으로 팝업창이 닫히지 않도록
94                             한다.
95                             .show();                      // 알림창 띄우기
96                         }
97                     );
98                 }
99             }
100             // 생성자에서 데이터 리스트 객체를 전달받음.
101             RecyclerViewAdapter(ArrayList<ListViewItem> list) {
102                 mData = list;
103             }
104             // onCreateViewHolder() - 아이템 뷰를 위한 뷰홀더 객체 생성하여 리턴.
105             @Override
106             public RecyclerViewAdapter.ViewHolder onCreateViewHolder(ViewGroup parent,
107             int viewType) {
108                 Context context = parent.getContext();           // 부모 객체 받음
109                 LayoutInflator inflater = (LayoutInflator) context.getSystemService(
110                 Context.LAYOUT_INFLATER_SERVICE);
111                 View view = inflater.inflate(R.layout.recyclerview_item, parent, false);
112                 RecyclerViewAdapter.ViewHolder vh = new RecyclerViewAdapter.ViewHolder(

```

```
105 view);
106
107     return vh;
108 }
109
110 // onBindViewHolder() - position에 해당하는 데이터를 뷰홀더의 아이템뷰에 표시.
111 @Override
112 public void onBindViewHolder(RecyclerViewAdapter.ViewHolder holder, int
position) {
113     ListViewItem item = mData.get(position) ;
114     //holder.icon.setImageDrawable(item.getIcon()) ;
115     holder.title.setText(item.getText()) ;
116     holder.desc.setText(item.getMessage()) ;
117
118     BoxChanger(holder, position);
119 }
120
121
122 // getItemCount() - 전체 데이터 갯수 리턴.
123 @Override
124 public int getItemCount() {
125     return mData.size();
126 }
127
128 private void BoxChanger(RecyclerViewAdapter.ViewHolder holder, int position
) {
129     ListViewItem item = mData.get(position) ;
130     holder.checkBox.setChecked(item.getSelected());
131     holder.checkBox.setTag(new Integer(position));
132
133     //for default check in first item
134     if(position == 0 && mData.get(0).getSelected() && holder.checkBox.
isChecked()) {
135         lastChecked = holder.checkBox;
136         lastCheckedPos = 0;
137     }
138
139     holder.checkBox.setOnClickListener(new View.OnClickListener() {
140         @Override
141         public void onClick(View v) {
142             CheckBox cb = (CheckBox)v;
143             int clickedPos = ((Integer)cb.getTag()).intValue();
144
145             if(cb.isChecked()) {
146                 if(lastChecked != null) {
147                     lastChecked.setChecked(false);
148                     mData.get(lastCheckedPos).setSelected(false);
149                 }
150
151                 lastChecked = cb;
152                 lastCheckedPos = clickedPos;
153             }
154             else
155                 lastChecked = null;
156
157             mData.get(clickedPos).setSelected(cb.isChecked());
158         }
159     });
160 }
161 }
```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/RecyclerViewAdapter.java

```
162     public void removeAt(int position) {  
163         mData.remove(position);  
164         notifyItemRemoved(position);  
165         notifyItemRangeChanged(position, getItemCount());  
166     }  
167  
168 }  
169  
170
```

```
1 package com.example.teamauction;
2
3 import android.graphics.Rect;
4 import android.view.View;
5
6 import androidx.annotation.NonNull;
7 import androidx.recyclerview.widget.RecyclerView;
8 /**
9  * 리사이클러뷰 간격에 관련된 클래스
10 */
11
12 public class RecyclerViewDecoration extends RecyclerView.ItemDecoration {
13
14     private final int divHeight; // 아이템 간의 간격
15
16     public RecyclerViewDecoration(int divHeight) {
17         this.divHeight = divHeight;
18     }
19
20     @Override
21     public void getItemOffsets(@NonNull Rect outRect, @NonNull View view, @NonNull
22     RecyclerView parent, @NonNull RecyclerView.State state) {
23         super.getItemOffsets(outRect, view, parent, state); // 부모 상속
24         if (parent.getChildAdapterPosition(view) != parent.getAdapter().
25             getItemCount() - 1){ // 가장 위는 빼고
26             outRect.bottom = divHeight;
27         }
28     }
29 }
```

```

1 package com.example.teamauction;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.core.content.ContextCompat;
5
6 import android.content.Intent;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.ImageButton;
11 import android.widget.ListView;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import com.android.volley.RequestQueue;
16 import com.android.volley.Response;
17 import com.android.volley.toolbox.Volley;
18
19 import org.json.JSONArray;
20 import org.json.JSONException;
21 import org.json.JSONObject;
22 import org.w3c.dom.Text;
23
24
25 /**
26 * 판매중인 아이템 화면
27 * 경매장 화면에서 판매중인 아이템 버튼을 클릭한 뒤 화면
28 * 판매중인 아이템중 삭제, 수정을 할 아이템을 선택 할 수 있게 하는 액티비티
29 */
30
31 public class SellingItemScreen extends AppCompatActivity {
32     private GameAccountInfo accountInfo; // 계정 정보에 대한 객체
33     private TextView UserCharName; // 유저 게임 닉네임 텍스트뷰
34     private ImageButton backButton; // 뒤로가기 버튼
35     private TextView money; // 유저 게임 돈 텍스트뷰
36
37     ListView listview = null;
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.screen_item_selling);
42
43         ListViewAdapter adapter; // 어댑터 선언
44         Intent account_info = getIntent(); // 로그인 계정 정보 받기
45         accountInfo = (GameAccountInfo) account_info.getSerializableExtra("account_info"); // account_info라는 key로 GameAccountInfo 객체를 받아옴
46         String myGameName = accountInfo.getGameName(); // 게임 이름 정보 가져오기
47         String myCharName = accountInfo.getCharacterName(); // 게임 캐릭터 닉네임 정보
48         가져오기
49         UserCharName = findViewById(R.id.MyUserCharName); //유저 닉네임 들어갈 공간 생성
50         UserCharName.setText(myCharName); //로그인한 계정 캐릭터 이름으로 변경
51
52         // Adapter 생성
53         adapter = new ListViewAdapter() ;
54
55         // 리스트뷰 참조 및 Adapter달기
56         listview = (ListView) findViewById(R.id.selling_list);
57         listview.setAdapter(adapter);
58

```



```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/SellingItemScreen.java

99                     money = findViewById(R.id.myMoneyBox); //돈이 들어갈 공간 생성
100                    money.setText(Money1); //유저의 돈으로 변경
101                    adapter.notifyDataSetChanged(); // 반복문 수행 후에 리스트가 업데이트
    됨을 알림
102                } catch (JSONException e) { // 접속 오류가 난 것이라면
103                    e.printStackTrace(); // 오류 추적
104                    // 토스트 메세지를 띄워줌
105                    Toast.makeText(getApplicationContext(), "no connection",
    Toast.LENGTH_SHORT).show();
106                }
107            }
108        };
109        String purl1 = "http://ualsgur98.dothome.co.kr/Money.php"; // 통신할 php
    주소
110        // 데이터 전송을 위한 데이터 세팅
111        PHPRequest validateRequest1 = new PHPRequest( purl1, myGameName,
    myCharName, responseListener1);
112        RequestQueue queue1 = Volley.newRequestQueue(SellingItemScreen.this); //
    큐를 생성
113        queue1.add(validateRequest1); // 큐에 추가
114
115
116        backButton = findViewById(R.id.back_auctionScreen2); // 뒤로가기 버튼 경매장
    화면으로 이동
117        backButton.setOnClickListener(new View.OnClickListener() { //뒤로가기 버튼
    클릭 이벤트 생성
118            @Override
119            public void onClick(View view) { // onClick 함수 재정의
120                // SellingScreen에서 AuctionScreen으로 가는 intent 준비
121                Intent intent = new Intent(SellingItemScreen.this, AuctionScreen.
    class);
122                intent.putExtra("account_info", accountInfo); // 경매장 화면에 계정
    정보 보내줌
123                startActivity(intent); // AuctionScreen 실행
124                finish(); // 현재 액티비티 종료
125            }
126        });
127
128        //리스트 뷰에 있는 아이템을 터치하고 판매하기 버튼을 누르면 DeleteScreen창 띄우기
129        Button dropingbutton = findViewById(R.id.dropButton);
130        dropingbutton.setOnClickListener(new View.OnClickListener() { // 구매하기
    버튼 클릭 이벤트 생성
131            @Override
132            public void onClick(View view) { // onClick 함수 재정의
133                int pos = listView.getCheckedItemPosition(); // 리스트뷰 위치데이터
    변수
134                if (pos > -1) { // pos는 0이상이므로 그 이상일 때만 작동
135                    ListViewitem item = (ListViewitem) adapter.getItem(pos); //
    터치한 아이템의 위치값 받아옴
136                    String uninumber = item.getUninumber(); //아이템의 Uninumber
    받아옴
137                    String regnum = item.getRegnumber(); //아이템의 RegisterNumber
    받아옴
138                    // SellingItemScreen에서 DeleteScreen으로 가는 intent 준비
139                    Intent intent = new Intent(SellingItemScreen.this,
    DeleteScreen.class);
140                    intent.putExtra("account_info", accountInfo); // 판매하기 화면에
    계정 정보 보내줌
141                    intent.putExtra("myuninum", uninumber); // 판매하기 화면에
    Uninumber 보내줌

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/SellingItemScreen.java

```

142             intent.putExtra("myregnum", regnum); // 판매하기 화면에
143             RegisterNumber 보내줌
144             startActivity(intent); // DeleteScreen 실행
145             finish(); // 현재 액티비티 종료
146         }
147     });
148
149     //리스트 뷰에 있는 아이템을 터치하고 판매하기 버튼을 누르면 FixScreen창 띄우기
150     Button fixingbutton = findViewById(R.id.fixButton);
151     fixingbutton.setOnClickListener(new View.OnClickListener() { // 구매하기
버튼 클릭 이벤트 생성
152         @Override
153         public void onClick(View view) { // onClick 함수 재정의
154             int pos = listView.getCheckedItemPosition(); // 리스트뷰 위치데이터
변수
155             if (pos > -1) { // pos는 0이상이므로 그 이상일 때만 작동
156                 ListViewItem item = (ListViewItem) adapter.getItem(pos); //
터치한 아이템의 위치값 받아옴
157                 String uninumber = item.getUninumber(); //아이템의 Uninumber
받아옴
158                 String regnum = item.getRegnumber(); //아이템의 RegisterNumber
받아옴
159                 // SellingItemScreen에서 FixScreen으로 가는 intent 준비
160                 Intent intent = new Intent(SellingItemScreen.this, FixScreen.
class);
161                 계정 정보 보내줌
162                 intent.putExtra("account_info", accountInfo); // 삭제하기 화면에
163                 Uninumber 보내줌
164                 intent.putExtra("myuninum", uninumber); // 삭제하기 화면에
165                 RegisterNumber 보내줌
166                 intent.putExtra("myregnum", regnum); // 삭제하기 화면에
167                 startActivity(intent); // FixScreen 실행
168             }
169         });
170     }
171

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/SellingScreen.java

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/SellingScreen.java

53         //판매하려는 아이템 정보를 DB에서 받아와 화면에 보여줌
54         Response.Listener<String> responseListener = new Response.Listener<String>
55     >() {
56             @Override
57             public void onResponse(String response) { // onResponse 함수 재정의
58                 try { // 만약 반응이 왔고 정상적이라면
59                     JSONObject jsonResponse = new JSONObject(response); // json
60                     형태로 받기 위해 객체 준비
61                     boolean success = jsonResponse.getBoolean("success"); //
62                     success라는 key 값으로 boolean value를 받음
63                     if (success) { // success가 참이라면
64                         String BuyItemName = jsonResponse.getString("ItemName");
65                     ); // ItemName 데이터 저장
66                         String BuyItemInfo = jsonResponse.getString("ItemInfo");
67                     // ItemInfo 데이터 저장
68                         ItemName.setText(BuyItemName); //판매하려는 아이템 이름으로 변경
69                         ItemInfo.setText(BuyItemInfo); //판매하려는 아이템 정보로 변경
70                     } else { //success가 false일 때
71                         // 토스트 메세지를 띄워줌
72                         Toast.makeText(getApplicationContext(), "서버와 연결이
73                         끊겼습니다", Toast.LENGTH_SHORT).show();
74                         return;
75                     }
76                 } catch (JSONException e) { // 접속 오류가 난 것이라면
77                     e.printStackTrace(); // 오류 추적
78                     // 토스트 메세지를 띄워줌
79                     Toast.makeText(getApplicationContext(), "no connection",
80                     Toast.LENGTH_SHORT).show();
81                 }
82             }
83             String purl = "http://ualsgur98.dothome.co.kr/SellItemInfo.php"; // 통신할
84             php 주소
85             // 데이터 전송을 위한 데이터 세팅
86             PHPRequest validateRequest = new PHPRequest(purl, myGameName, Uninum,
87             responseListener);
88             RequestQueue queue = Volley.newRequestQueue(SellingScreen.this); // 큐를
89             생성
90             queue.add(validateRequest); // 큐에 추가
91             //판매 버튼을 눌렀을때 값 저장, 내 아이템 화면으로 돌아감
92             yes_btn.setOnClickListener(new View.OnClickListener() { // 판매 버튼 클릭
93                 이벤트 생성
94                 @Override
95                 public void onClick(View view) {
96                     String get_textcost = editTextcost.getText().toString(); //입력받은
97                     가격의 값을 저장
98                     String get_textquan = editTextquantity.getText().toString(); //
99                     입력받은 수량의 값을 저장
100                     //판매하려는 아이템 정보 DB에 보냄
101                     Response.Listener<String> responseListener3 = new Response.
102                     Listener<String>() {
103                         @Override
104                         public void onResponse(String response) { // onResponse 함수
105                             재정의
106                             try { // 만약 반응이 왔고 정상적이라면
107                                 JSONObject jsonResponse = new JSONObject(response);
108                                 // json 형태로 보내기 위해 객체 준비
109                                 boolean success = jsonResponse.getBoolean("success")

```

File - /Users/shin/AndroidStudioProjects/TeamAuction/app/src/main/java/com/example/teamauction/SellingScreen.java

```

96 ); // success라는 key 값으로 boolean value를 받음
97             if (success) { // success가 참이라면
98                 Toast.makeText(getApplicationContext(), "
99                     판매하셨습니다.", Toast.LENGTH_SHORT).show();
100                } else { // success가 false일 때
101                    // 토스트 메세지를 띄워줌
102                    Toast.makeText(getApplicationContext(), "서버와
103                     연결이 끊겼습니다", Toast.LENGTH_SHORT).show();
104                }
105            }
106        }
107    }
108 }
109 }
110 }
111 String purl3 = "http://ualsgur98.dothome.co.kr/Trade.php"; //
112 //통신할 php 주소
113 // 데이터 전송을 위한 데이터 세팅
114 PHPRequest validateRequest3 = new PHPRequest(purl3, myGameName,
115 myCharName, get_textcost, get_textquan, Uninum, responseListener3);
116 RequestQueue queue3 = Volley.newRequestQueue(SellingScreen.this
117 ); // 큐를 생성
118 queue3.add(validateRequest3); // 큐에 추가
119
120 //Toast.makeText(getApplicationContext(), get_textcost, Toast.
121 LENGTH_SHORT).show();
122 // SellingScreen에서 ItemCheckScreen으로 가는 intent 준비
123 Intent intent = new Intent(SellingScreen.this, ItemCheckScreen.
124 class);
125
126 데이터 추가
127 startActivity(intent); // ItemCheckScreen 실행
128 finish(); // 현재 액티비티 종료
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }

```

143

144