

# PmangPlus SDK 결제 가이드

SDK Version:	1.3
작성일:	2011-09-08

## 목차

1	문서 개요 .....	1
2	PmangPlus 결제 개요 .....	1
3	상품과 가상계좌.....	2
3.1	상품.....	2
3.2	가상계좌 .....	3
4	In App Purchase .....	5
5	게임 서버 연동.....	6
6	연동.....	8
6.1	정보 등록 .....	8
6.2	SDK 기능 호출 .....	8
6.2.1	Android SDK .....	8
6.2.2	iOS SDK.....	11

# PmangPlus SDK 결제 가이드

## 1 문서 개요

본 문서는 PmangPlus SDK가 제공하는 결제 기능과 SDK를 통해 결제 기능을 구현하는 방법을 설명한다.

## 2 PmangPlus 결제 개요

PmangPlus의 결제 기능은 크게 1. 타 사업자가 제공하는 In-App purchase 구매 상품 관리, 2. 직접 구매, 3. 가상 화폐 기능을 제공한다. 단, 사용할 수 있는 기능은 앱이 배포될 마켓과 플랫폼에 따라 제약을 받을 수 있다. 다음은 각 기능에 대한 간략한 설명이다.

1. IAP(in-app purchase) 구매 상품 관리: PmangPlus를 이용해 타 사업자를 통해 구매한 상품을 관리할 수 있음.
2. 직접 구매: 타 사업자의 In-App purchase 기능이 아닌, PmangPlus가 직접 제공하는 WebView를 이용해 핸드폰 결제를 할 수 있음. 구매한 상품은 In-App purchase 구매 상품과 동일하게 관리된다.
3. 가상 화폐: 앱 별 가상 화폐 시스템을 제공한다. 이를 위해 1. 계좌 관리, 2. 계좌 충전, 3. 계좌를 이용한 상품 구매, 4. 계좌 충전/계좌를 통한 구매 이력 조회 기능을 제공한다.

IAP 구매 상품 관리/직접 구매	가상 화폐
<ul style="list-style-type: none"> <li>• receipt 등록 (IAP)</li> <li>• 직접 구매 I/F(Webview)</li> <li>• 상품 목록 조회</li> <li>• 상품 소비</li> <li>• 충전/구매 이력 조회(직접구매)</li> </ul>	<ul style="list-style-type: none"> <li>• 가상 화폐 충전 I/F(Webview)</li> <li>• 가상 화폐를 이용한 상품 구매</li> <li>• 가상 화폐 충전 이력 조회</li> <li>• 가상 화폐 소비 이력 조회</li> </ul>

그림 1 PmangPlus의 결제 기능

## 3 상품과 가상계좌

### 3.1 상품

PmangPlus의 상품은 그 특성에 따라 크게 1. 일반 상품, 2. 가상계좌 충전 상품으로 나뉜다. 일반 상품은 앱 내에서 효력을 발생시키는 상품이며, 가상계좌 충전 상품은 가상계좌의 현금성 포인트를 충전하는 특수한 용도의 상품이다. (상품권을 생각하면 된다.)

일반 상품은 사용 유형에 따라 1. 소비형, 2. 가입형, 3. 비소비형(영구형) 상품으로 나뉜다. 다음은 일반 상품의 세부 유형 별 설명이다.

1. 소비형 상품은 주어진 개수가 있어 사용 시 마다 개수를 차감시키고, 개수가 0이 되면 더 이상 사용할 수 없는 상품이다. (예: 99개 들이 폭탄) 개수를 모두 소진한 상품은 상품 목록에서 사라진다.
2. 가입형 상품은 특정 기간 동안 효력이 발생하는 상품이다. (예: 30일 무적 망또) 효력 기간이 만료된 상품은 상품 목록에서 사라진다.
3. 비소비형 상품은 한번 구매하면 영구적으로 효력이 발생하는 상품이다. (예: 확장 스테이지 입장권) 한번 구매한 비소비형 상품은 상품 목록에서 사라지지 않는다.

일반 상품과 달리 가상 계좌 충전 상품은 별도의 세부 유형을 갖지 않는다.

다음은 상품의 주요 속성이다.

속성 명	설명	예	속성 유효 대상
APP ID	상품이 등록된 PmangPlus APP ID	1	공통
상품 유형	일반 상품과 가상계좌 충전 상품 구분	일반상품/충전상품	공통
사용 유형	일반 상품의 세부 사용 유형.	소비형/ 가입형/ 비소비형	일반상품
상품 명	상품 이름	폭탄아이템	공통
상품 상세	상품의 상세 설명	폭탄이 터집니다.	공통
부가세 포함 가격	부가세를 포함한 가격. 소수점을 포함함	1100.00	공통
부가세 제외 가격	부가세를 제외한 가격. 소수점을 포함함	1000.00	공통
통화	가격의 통화 단위.	KRW	공통
유효개수	소비형 일반상품의 잔여 사용 회수	30	소비형 일반상품
유효기간	가입형 일반상품의 잔여 사용 기간(일)	30	가입형 일반상품
IAP ID	외부 사업자의 IAP를 통해 구매한 receipt과 매핑하기 위한 IAP 상품 식별번호	com.neowiz.tapsonic.0	IAP 등록형 일반상품
구입 가능 계좌유형 ID	가상 계좌의 금액을 소비해서 구매하는 일반 상품일 경우 비용을 차감할 계좌의 계좌 유형 ID.	1	가상계좌를 통해 구매하는 일반상품
충전 대상 계좌유형 ID	충전상품을 구매했을 때 금액이 충전될 계좌 유형 ID.	1	충전상품
충전타입 별 금액	충전상품을 구매했을 때 충전될 금액. 포인트 타입 별로 구분된다.	현금성:1000.00 비현금성:0.00	충전상품

## 3.2 가상계좌

가상 계좌는 사용자가 상품을 구매할 때 쓸 수 있는 포인트를 보관하는 계좌이다. 가상계좌의 포인트 타입은 현금성 포인트, 비현금성 포인트, 마일리지 포인트의 세 가지 이다.

- 현금성 포인트: 가상계좌 충전상품을 구매해서 획득한 포인트이다.
- 비현금성 포인트: 관리자가 제공하거나, 특정 조건을 만족했을 때 앱이 부여하는 포인트이다.
- 마일리지 포인트: 상품 구매 등의 활동으로 앱이 부여하는 포인트이다.

가상계좌는 사용자 별, 앱 별 생성된다. 하나의 계좌엔 현금성, 비현금성, 마일리지 포인트가 모두 쌓일 수 있다. 차감 순서는 마일리지 -> 비현금성 -> 현금성 포인트의 순서이며, **특정 타입의 포인트를 선택적으로 차감할 순 없다.**

### 중요

가상 계좌는 3가지 유형의 포인트를 모두 담을 수 있다. 그러나 가상계좌의 포인트를 이용해 상품을 구매할 경우, 특정 유형의 포인트만을 선택적으로 차감할 수 없다. 따라서 특정 유형의 포인트로만 상품을 구매하도록 강제하려면 포인트 유형 별로 계좌유형을 지정해야 한다. (ex. 계좌유형ID가 1인 계좌엔 현금성만, ID 2엔 비현금성만, 3엔 마일리지만 적립함.) 이렇게 하면, 사용자는 포인트 유형 별 별도의 계좌를 가지게 된다.

상품을 등록할 땐 포인트 유형 별로 정의된 계좌유형을 지정한다. 즉, 현금성 포인트로만 구매 가능한 상품은 계좌유형ID를 1로, 비현금성 포인트로만 구매 가능한 상품은 계좌유형ID를 2로 지정한다.

다음은 가상계좌의 주요 속성이다.

속성 명	설명	예
APP ID	계좌가 등록된 PmangPlus APP ID	1
MEMBER ID	계좌를 보유한 PmangPlus 사용자 ID	1
ACCOUNT ID	계좌 별 고유 ID.	1
계좌유형 ID	계좌의 유형 ID. 앱 별 복수의 계좌 유형 ID를 가질 수 있다.	1
포인트 유형 별 금액	계좌에 적립된 포인트 유형 별 금액.	CASH: 10000 (현금성) POINT: 300 (비현금성) MILEAGE: 400 (마일리지)

사용자, 상품, 가상계좌의 관계를 도식화하면 다음과 같다.

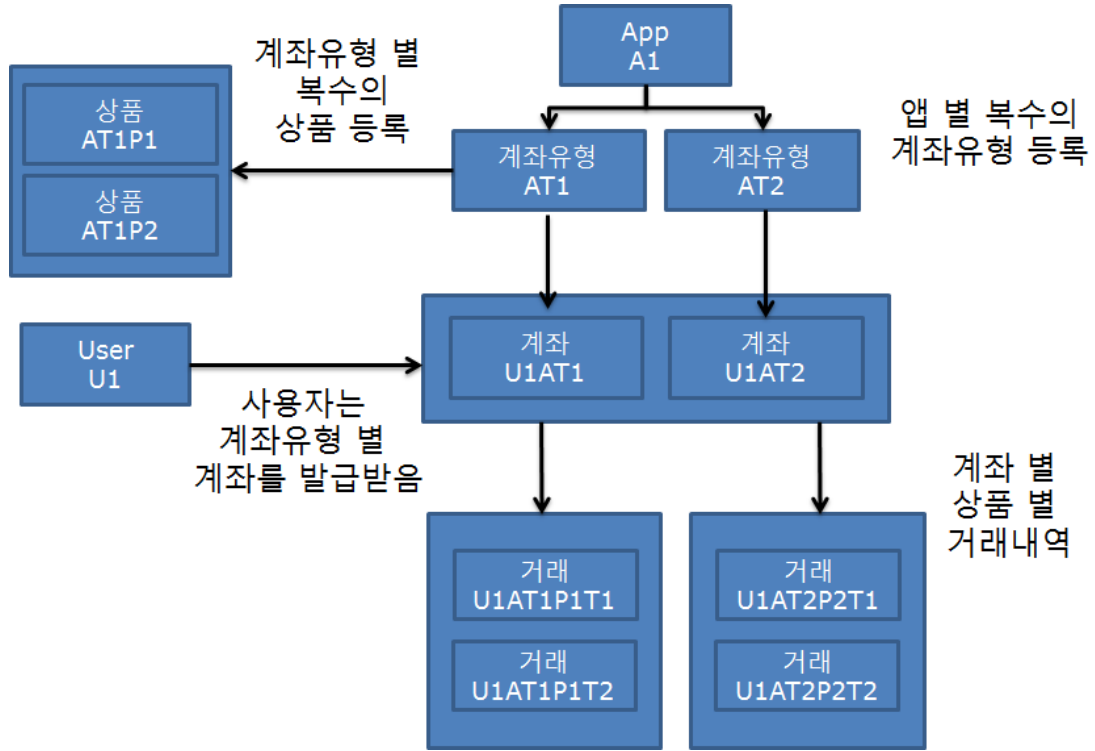


그림 2 사용자/상품/가상계좌 간 상관관계

## 4 In App Purchase

Apple의 App store나 SK의 T Store 등의 플랫폼은 독자적인 In App Purchase 모듈을 제공한다. PmangPlus에선 이런 외부 사업자의 IAP로 구매한 상품의 등록, 사용 I/F를 제공한다.

### 참고

현재는 App Store만 연동이 가능하다. 안드로이드 앱의 경우, 구글 Market이 제공하는 IAP가 아닌, PmangPlus가 직접 제공하는 WebView를 이용해 결제를 해야 한다.

## 5 게임 서버 연동

별도의 서버를 운용하지 않는 StandAlone 형태의 앱의 결제 처리 흐름은 다음과 같다.

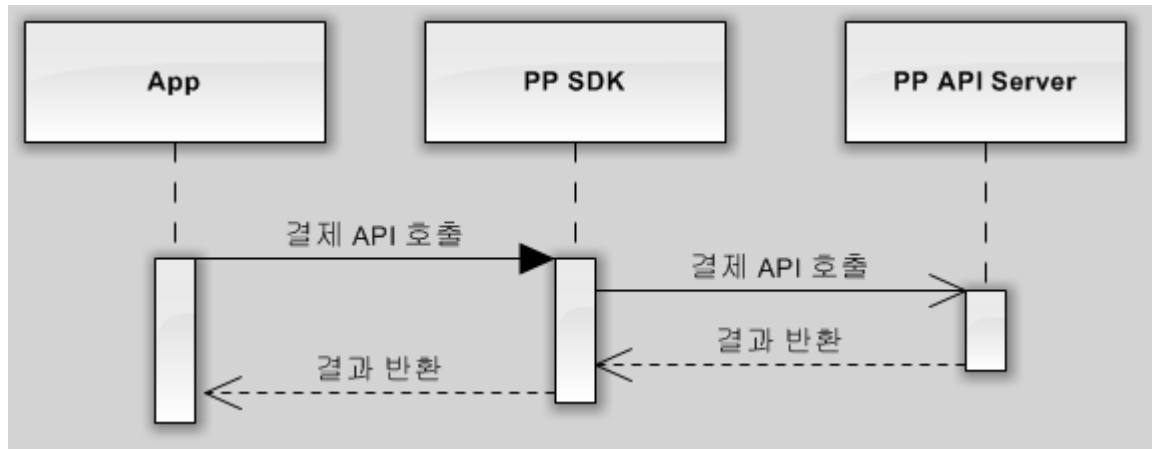


그림 3 StanaAlone 앱의 결제 처리(성공)

그러나 앱이 서버를 운영하며(앱 서버), 앱 서버가 결제된 상품의 효력을 관리하는 경우, 결제 후 상품 효력 지급 요청을 앱 서버에 통보해야 한다. PP SDK의 경우, API 서버에서 앱 서버로 상품 효력 지급 요청을 HTTP GET 방식으로 보내고, 결과는 HTTP response 코드로 받는다. 호출할 URL은 PmangPlus의 관리자 웹 페이지에서 설정한다. PP API 서버는 결제에 성공한 경우 이 URL을 HTTP GET 방식으로 호출한다. 상품 효력 지급 요청을 위해선 인증을 위한 gameAuth, certKey가 필요하다.

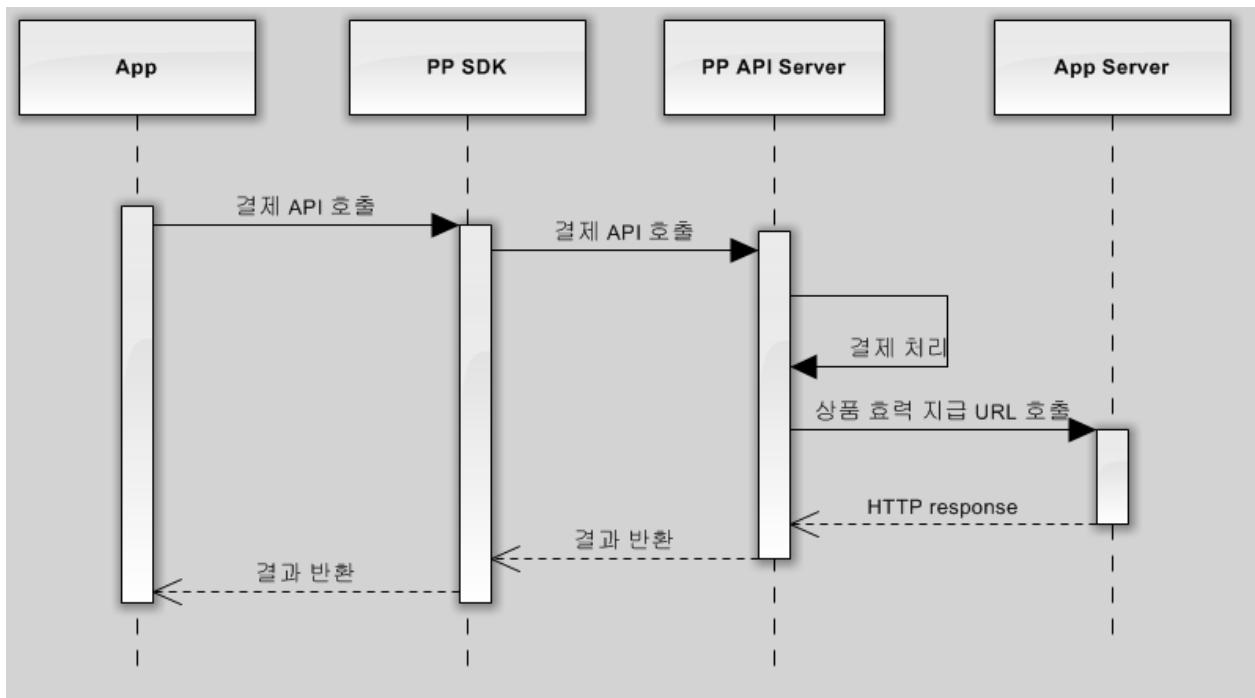


그림 4 게임 서버를 운영하는 앱의 결제 처리(성공)



상품 효력 지급 실패 여부는 HTTP status code로 판단한다. PP API 서버는 HTTP status 코드가 200이 아닌 경우 효력 지급 실패로 판단한다. 이 경우 앱은 PP SDK의 API를 호출해서 상품 효력 지급을 수행해야 한다. 상세한 내용은 본 문서의 6.2 SDK 기능 호출을 참고한다.

#### 중요

앱 서버는 상품 효력 지급에 실패한 경우, 앱이 효력 지급 재요청을 할 수 있도록 200 이외의 HTTP status 코드를 반환해야 한다.

## 6 연동

PmangPlus의 결제 기능을 이용하기 위한 연동 절차는 다음과 같다.

1. 앱 정보 등록: Admin 기능을 이용해 상품, 계좌유형 정보를 등록한다.
2. SDK 기능 호출: 플랫폼 별로 제공되는 PmangPlus SDK의 기능을 이용해 구매/결제를 수행한다.

### 6.1 정보 등록

결제 기능을 이용하기 위해 사전에 등록해야 하는 정보는 다음과 같다. 가상계좌는 선택적으로 사용할 수 있으므로 가상계좌를 사용하지 않는 앱은 별도로 가상계좌 관련 정보를 등록하지 않아도 된다.

구분	정보	상세	필수 여부
가상계좌	계좌유형	앱에서 사용할 계좌유형을 등록한다.	가상계좌 사용 시
가상계좌	계좌충전 상품	가상계좌에 포인트를 충전할 수 있는 계좌충전 상품을 등록한다.	가상계좌 사용 시
구매	상품	<p>상품명, 상품 유형 등의 상품 세부 정보를 등록한다.</p> <p>IAP 상품의 경우, IAP 플랫폼에 등록된 상품 ID를 등록한다.</p> <p>가상계좌 구매 상품의 경우, 포인트를 차감할 계좌유형 ID를 등록한다.</p>	O

### 6.2 SDK 기능 호출

SDK 기능 호출 방식은 제공하는 SDK에 따라 달라지므로 앱의 플랫폼에 맞는 방식을 선택한다.

#### 6.2.1 Android SDK

Android SDK는 결제 기능을 포함하고 있으므로 별도의 라이브러리를 설치할 필요는 없다.

결제 기능은 다른 기능들과 동일하게 com.pmangplus.ui.PP를 통해 제공되므로, PP의 javadoc을 참고한다. 다만 WebView를 이용해 사용자의 입력을 받는 기능은 내부적으로 Activity.startActivityForResult() 메서드를 호출해서 결제 Activity를 띄우며, onActivityResult() 메서드를 통해 결제 결과를 전달하기 때문에 기능을 사용하는 Activity에서 onActivityResult() 메서드를 override해야 한다. 결제 결과는 onActivityResult에 ResultCode와 Intent로 전달된다. openPurchaseItemView() 와 openPurchaseCashView() 가 이에 해당한다.

onActivityResult() 로 전달되는 ResultCode와 내용은 다음과 같다.

ResultCode	발생 상황	Intent 내용
Activity.RESULT_OK	결제가 성공하고 게임 서버와의 통신도 성공함	extra에 data라는 이름의 PurchaseResultBase 객체가 저장됨
Activity.RESULT_CANCELED	결제 실패 혹은 사용자의 결제 취소	null
UIHelper.RESULT_CODE_UNKNOWN	서버로부터 응답을 전달받지 못해 결제의 성공/실패 여부를 확인하지 못함	null
UIHelper. RESULT_CODE_ERR_MOBILE_GAMEAUTH	결제는 성공했으나 게임 서버와의 통신은 실패함. 게임 서버와 통신이 필요하지 않은 앱에선 발생하지 않음.	null

**참고** UIHelper 클래스는 com.pmangplus.ui.internal 패키지에 들어있다.

이를 도식화한 그림은 아래와 같다.

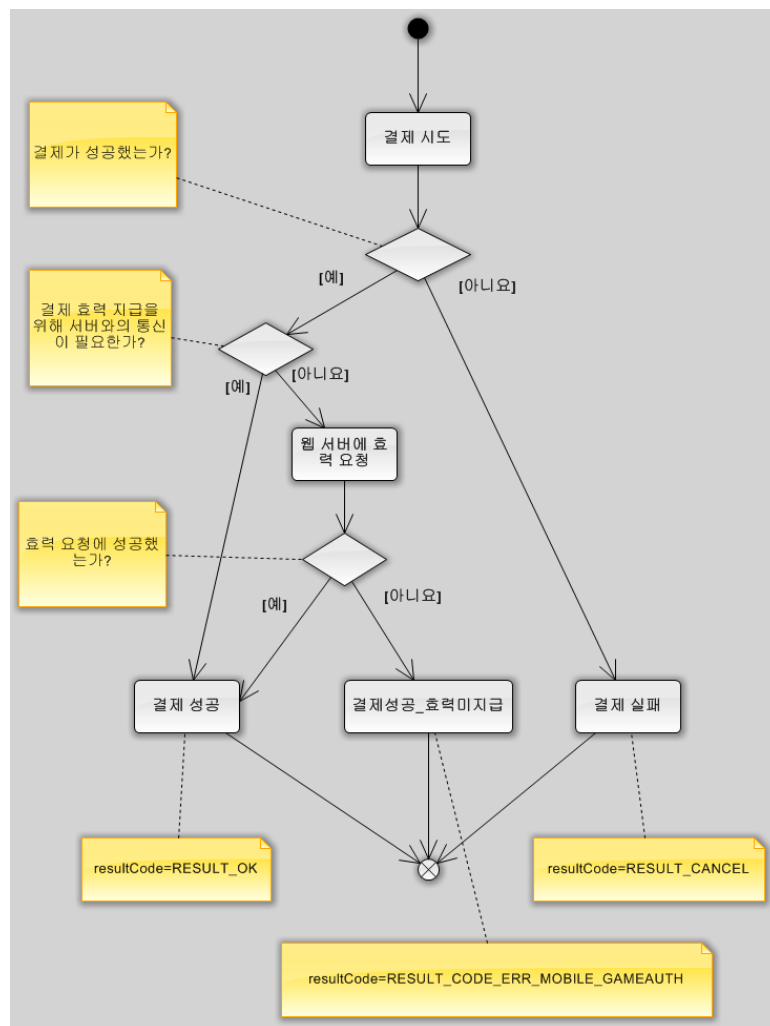


그림 5 결제 처리 결과에 따른 ResultCode

결제에 성공한 경우, 즉 ResultCode가 Activity.RESULT\_OK인 경우엔 Intent에 data라는 이름의 결과 객체가 전달된다. 객체의 타입은 결제 유형에 따라 충전상품 구매의 경우

com.pmangplus.core.model.CashProductPurchaseResult가 전달되고, 일반상품 구매의 경우

com.pmangplus.core.model.AppProductPurchaseResult가 전달된다.

결제의 성공/실패 여부를 확인할 수 없는 경우, 즉 ResultCode가 com.pmangplus.ui.internal. UIHelper.

RESULT\_CODE\_UNKNOWN인 경우는 결제 요청을 서버로 보냈으나 서버에서 결과를 받아오지 못한 경우에 해당한다. 이 경우 앱은 다시 서버로 요청을 넘겨 결제 성공/실패 여부를 확인해야 한다.

ResultCode 중 UIHelper.RESULT\_CODE\_ERR\_MOBILE\_GAMEAUTH 는 결제 결과를 게임 서버에 알려주는 경우, PmangPlus의 API가 게임 서버와 통신하지 못한 경우에만 발생한다. 게임 서버와 통신이 필요하다면

openPurchaseItemView() 의 매개변수 gameAuthUrl에 서버 URL을 넘겨야 한다. gameAuthUrl의 값이 null이 아닐 경우, PmangPlus API는 전달받은 URL을 HTTP 호출하며, HTTP response code가 200이 아닐 경우

UIHelper.RESULT\_CODE\_ERR\_MOBILE\_GAMEAUTH 를 반환한다. 이 경우 이미 결제는 이뤄졌기 때문에 앱은 효력 지급 요청 API인 validatePayinfo() 를 호출해야 한다.

다음 코드는 onActivityResult의 구현 예시이다.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    String msg="";

    if (requestCode == REQ_CODE_PAY) {
        if (resultCode == Activity.RESULT_OK) {
            PurchaseResultBase result = (PurchaseResultBase) data.getSerializableExtra("data");
            msg = "ok~ data:" + result;
        } else if (resultCode == Activity.RESULT_CANCELED) {
            msg = "err: canceled";
        } else if (resultCode == UIHelper.RESULT_CODE_ERR_MOBILE_GAMEAUTH) {
            msg = "err : connection to game server failed";
        } else if (resultCode == UIHelper.RESULT_CODE_UNKNOWN) {
            msg = "err : didn't receive result";
        }
        Toast.makeText(PaymentActivity.this, msg , Toast.LENGTH_LONG).show();
    }
}
```

그림 6 결제 화면 표출 예시

일반적인 API 사용 flow는 다음과 같다. 개별 API의 내용은 javadoc을 참고한다.

#### 1. 가상계좌를 사용하지 않는 경우

##### A. 상품 확인/사용

- i. PP.listProducts() 를 이용해 등록된 전체 상품 목록을 조회한다.
- ii. PP.getProductInfo() 를 이용해 특정 등록 상품의 정보를 조회한다.
- iii. PP.listPurchasedProduct() 를 이용해 사용자가 이미 구매한 상품 목록을 조회한다.
- iv. PP.consumeProduct() 를 이용해 구매한 상품 중 소비형 상품의 잔여 개수를 1만큼 차감한다.

##### B. 상품 구매

- i. PP.listProducts() 를 이용해 등록된 전체 상품 목록을 조회한다.
- ii. 상품 중 구매할 상품을 선택한 뒤, PP.openPurchaseItemView() 를 호출해서 결제&구매 WebView를

연다.

- iii. **구매에 성공한 경우, PP.listPurchasedProduct() 를 이용해 사용자가 이미 구매한 상품 목록을 조회한다.**
- iv. PP.getPayinfo() 를 이용해 특정 트랜잭션의 결과를 확인한다.
- v. PP.consumeProduct() 를 이용해 구매한 상품 중 소비형 상품의 잔여 개수를 1만큼 차감한다.

#### C. 앱 서버의 효력 적용

- i. PP.listInvalidPayinfo() 를 이용해 구매는 성공했으나 효력 지급엔 실패한 구매 내역을 조회한다.
- ii. PP.validatePayinfo() 를 이용해 효력 지급에 실패한 구매의 효력 지급을 요청한다.

### 2. 가상계좌를 사용하는 경우

#### A. 가상계좌 조회

- i. PP.listVaAccounts() 를 이용해 사용자 계좌 내역을 조회한다.

#### B. 가상계좌 충전

- i. PP.listVaProducts() 를 이용해 등록된 전체 충전상품 목록을 조회한다.
- ii. PP.getVaProductInfo() 를 이용해 특정 등록 상품의 정보를 조회한다.
- iii. 상품 중 구매할 충전상품을 선택한 뒤, PP.openPurchaseCashView() 를 호출해서 결제&구매 WebView를 연다.
- iv. **구매에 성공한 경우, PP.listVaAccounts() 를 이용해 사용자 계좌 내역을 조회한다.**

#### C. 가상계좌 금액을 이용한 상품 구매

- i. PP.listVaAccounts() 를 이용해 사용자 계좌 내역을 조회한다.
- ii. PP.listProducts() 를 이용해 등록된 전체 상품 목록을 조회한다.
- iii. 구매하고자 하는 상품의 계좌유형 ID와 사용자 계좌의 계좌유형 ID를 확인한다. 계좌에 잔액이 부족할 경우 구매에 실패하며, 계좌유형 ID가 같지 않은 경우에도 구매에 실패한다.
- iv. 정보를 확인했다면 PP.purchaseVaProduct() 를 이용해 구매를 시도한다.
- v. **구매에 성공한 경우, PP.listPurchasedProduct() 를 이용해 사용자가 이미 구매한 상품 목록을 조회한다.**

#### 중요

WebView나 가상계좌 금액을 이용한 결제/구매에 성공한 경우 callback 객체나 onActivityResult를 통해 결과를 보내주지만, 반드시 별도의 API 호출을 다시 해서 서버로부터 결과를 받아와야 한다. 실제 구매가 이뤄졌으나 네트워크 오류로 구매 결과를 보내주지 못한 경우가 생길 수 있기 때문이다.

## 6.2.2 iOS SDK

iOS SDK에선 애플의 정책 때문에 In App Purchase만 지원한다. 앱 개발자는 애플이 제공하는 Store Kit을 이용해 애플 서버와 연동되는 결제 모듈을 직접 구현해야 한다. PmangPlus SDK가 Store Kit을 직접 다룰 경우, 앱 개발자가 별도의 Store Kit 모듈을 구현했을 때 중복 구현에 따른 예상치 못한 결과가 발생할 수 있기 때문이다. 애플이 제공하는 In App Purchase에 관한 기본 정보는 다음 경로에서 확인할 수 있다.

**참고자료: In App Purchase Programming Guide**

<http://developer.apple.com/library/iOS/#documentation/NetworkingInternet/Conceptual/StoreKitGuide/Introduction/Introduction.html>

Store Kit과 PmangPlus의 API의 호출 순서는 다음과 같다.

1. PmangPlus 서버로부터 상품 목록을 수신한다.
  - A. requestInAppProduct 메서드는 해당 앱에 연계된 모든 상품 목록을 가져온다.
  - B. requestInAppProductId: 메서드는 특정 Product Id의 상품 정보를 가져온다.
2. 상품 목록에서 구매 가능한 대상인지 확인하기 위해 PmangPlus 서버로부터 구매 내역과 결제 이력을 수신한다.
  - A. requestInAppPurchases 메서드는 현재 효력이 있는 구매 상품 목록을 가져온다.
  - B. requestInAppTransactions 메서드는 구매한 상품의 결제 정보를 가져온다.
3. 확인된 상품의 정보로 Store Kit의 API를 이용해 실제 구매를 요청한다.
4. 완료된 구매 정보를 PmangPlus 서버에 전달한다
  - A. paymentInAppProductId:withReceipt: 메서드는 구매한 상품의 Product Id를 영수증(Receipt)과 함께 서버로 전달한다.
5. Consumable 상품의 경우 1회 사용할 때마다 PmangPlus 서버에 사용 사실을 전달한다.
  - A. updateInAppProductId:useConsume: 메서드는 구매한 Consumable 상품의 Product Id와 구입 내역의 Buy Id를 서버에 전달해 앞으로 사용 가능한 횟수를 1회 차감한다.

PmangPlus가 제공하는 각 메서드와 그에 연계된 delegate의 자세한 내용은 별도로 제공되는 문서인 "PmangPlus SDK for iOS API Reference"를 참고한다.