

BaroPAM App/Web 가이드

목차

목차.....	0
1. BaroPAM 앱	1
1.1 BaroPAM 앱 설치.....	1
1.2 BaroPAM 앱 사용(메뉴선택)	19
1.3 BaroPAM 앱 사용(애플리케이션정보).....	20
1.4 BaroPAM 앱 사용(본인확인)	28
1.5 BaroPAM 앱 사용(서버정보)	42
1.6 BaroPAM 앱 사용(환경설정)	50
1.7 BaroPAM 앱 사용(정보공유).....	68
1.8 BaroPAM 앱 FAQ.....	71
1.9 BaroPAM 연동 API.....	72
2. BaroPAM 웹 사이트.....	113
2.1 BaroPAM 웹 사용(회원가입)	113
2.2 BaroPAM 웹 사용(로그인)	118
2.3 BaroPAM 웹 사용(애플리케이션정보관리)	121
2.4 BaroPAM 웹 사용(서버정보관리)	130
2.5 BaroPAM 웹 사용(PIN 정보관리)	141
2.6 BaroPAM 웹 사용(접속로그관리)	144
2.7 BaroPAM 웹 사용(Abort & Logout)	145
3. About BaroPAM	146

1. BaroPAM 앱

정보자산에 로그인 시 "Verification code"에 입력할 **일회용 인증키**의 생성기인 "BaroPAM" 앱의 안드로이드 폰용 다운로드(<https://play.google.com/store/apps/details?id=com.baro.pam>)는 구글의 "Play 스토어"에서 아이폰 용은 애플의 "App store"에서 가능하며, 설치하는 일반 앱의 설치와 동일하다.



"BaroPAM" 앱은 Android 6.0 (Marshmallow) API 23, iOS 13.0 이상에서 사용 가능하며, 가로보기 모드를 지원하지 않는다.

Android용 "BaroPAM" 앱을 구글의 "Play 스토어"에서 다운로드 받아 설치할 수 없는 경우 다음과 같은 주소에서 apk를 다운로드 받아서 수동으로 "BaroPAM" 앱을 설치해야 한다.

한글버전: http://nuriapp.com/apk/app_baropam.apk

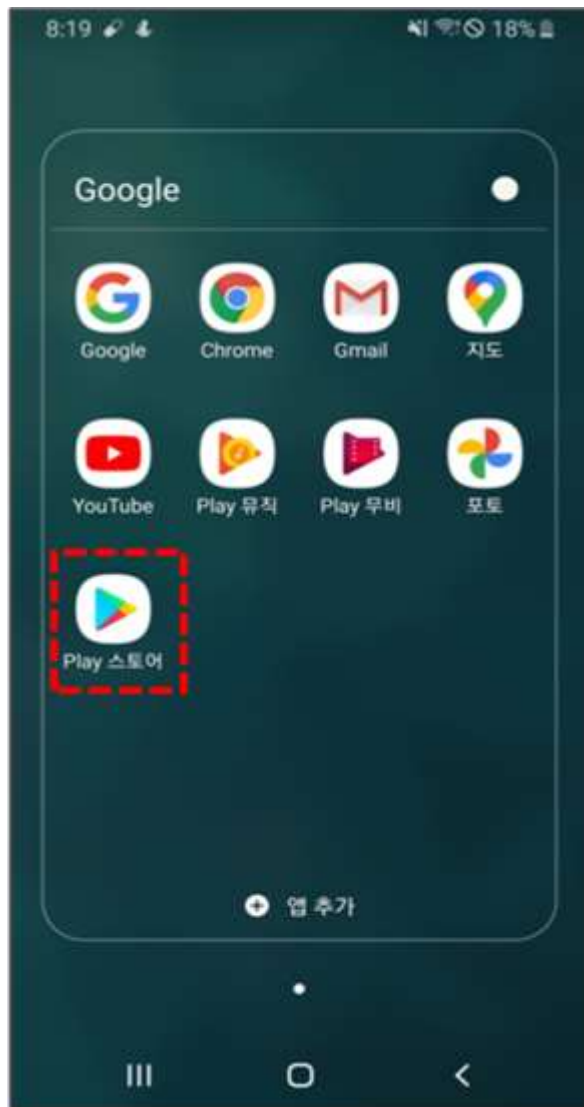
영문버전: http://nuriapp.com/apk/app_baropams.apk

1.1 BaroPAM 앱 설치

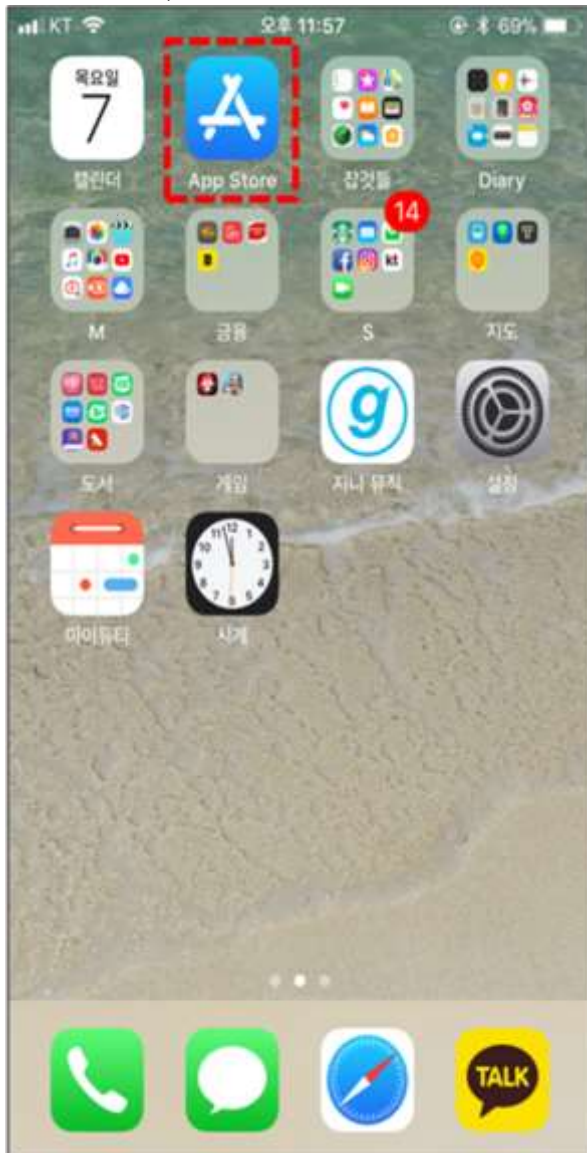
"BaroPAM" 앱의 다운로드 및 설치하는 다음과 같은 절차로 진행된다.

1. 스마트 폰에서 "Play 스토어" 또는 "App Store" 아이콘을 클릭한다.

(안드로이드 폰인 경우)

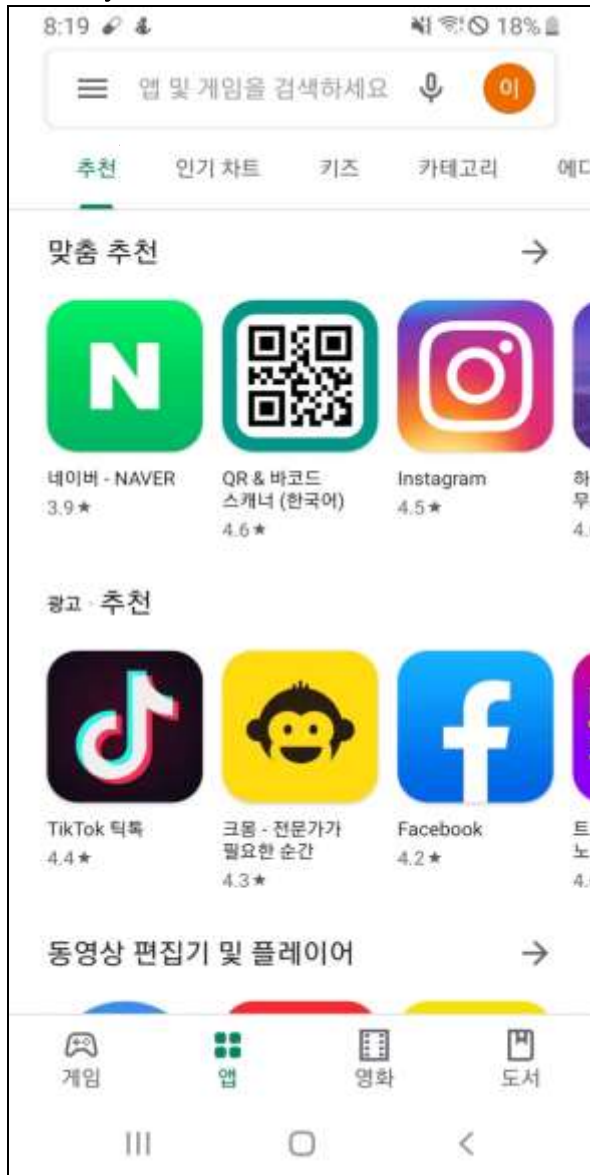


아이폰인 경우)

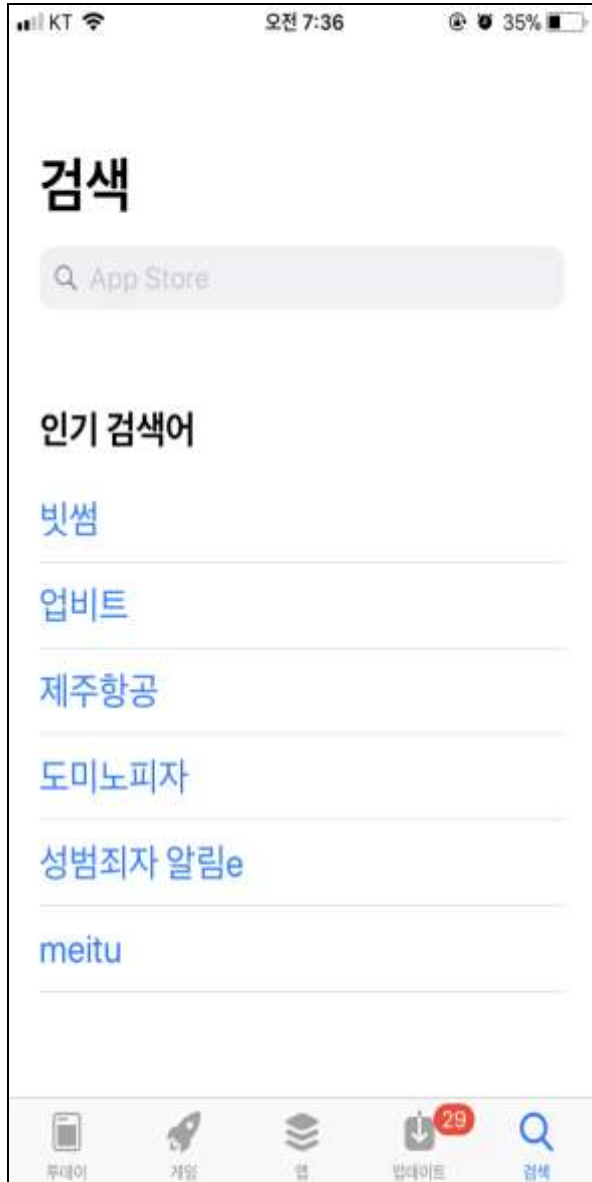


2. 구글 "Pay 스토어" 또는 애플 "App Store"가 다음과 같이 활성화된다.

구글 "Pay 스토어"인 경우)



애플 "App Store"인 경우)

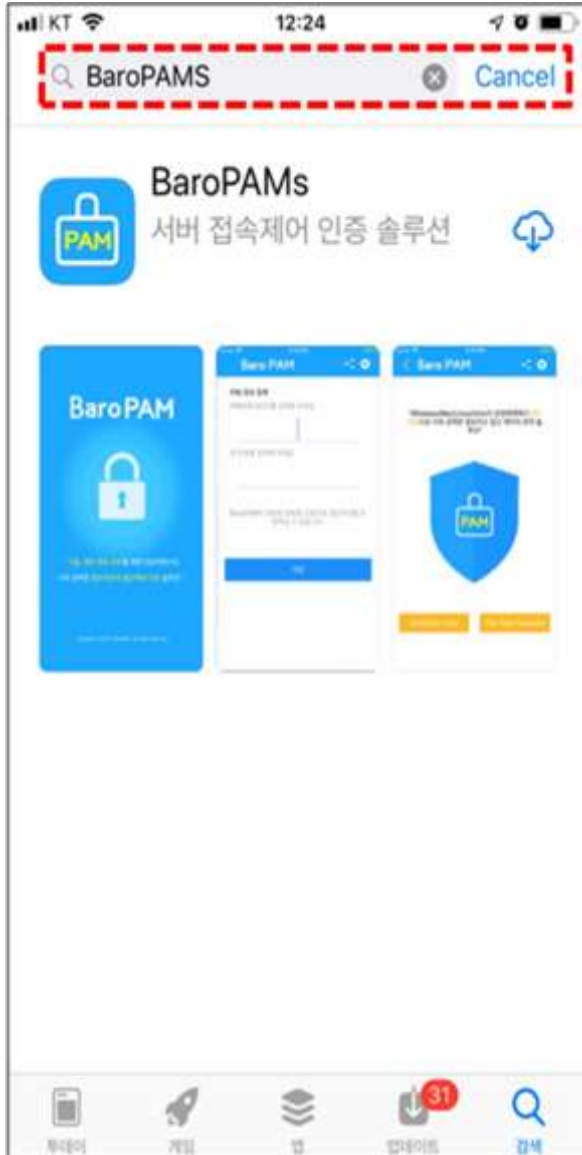


3. 상단의 검색어 란에 "baropam 또는 바로팜"을 다음과 같이 입력하면 "바로팜, BaroPAMs" 앱이 검색되어 나타난다.

구글 "Pay 스토어"인 경우)



애플 "App Store"인 경우)



4. "BaroPAM" 앱의 정보를 확인한 후 "설치" 또는 "받기" 버튼을 클릭한다.

구글 "Pay 스토어"인 경우)



애플 "App Store"인 경우)

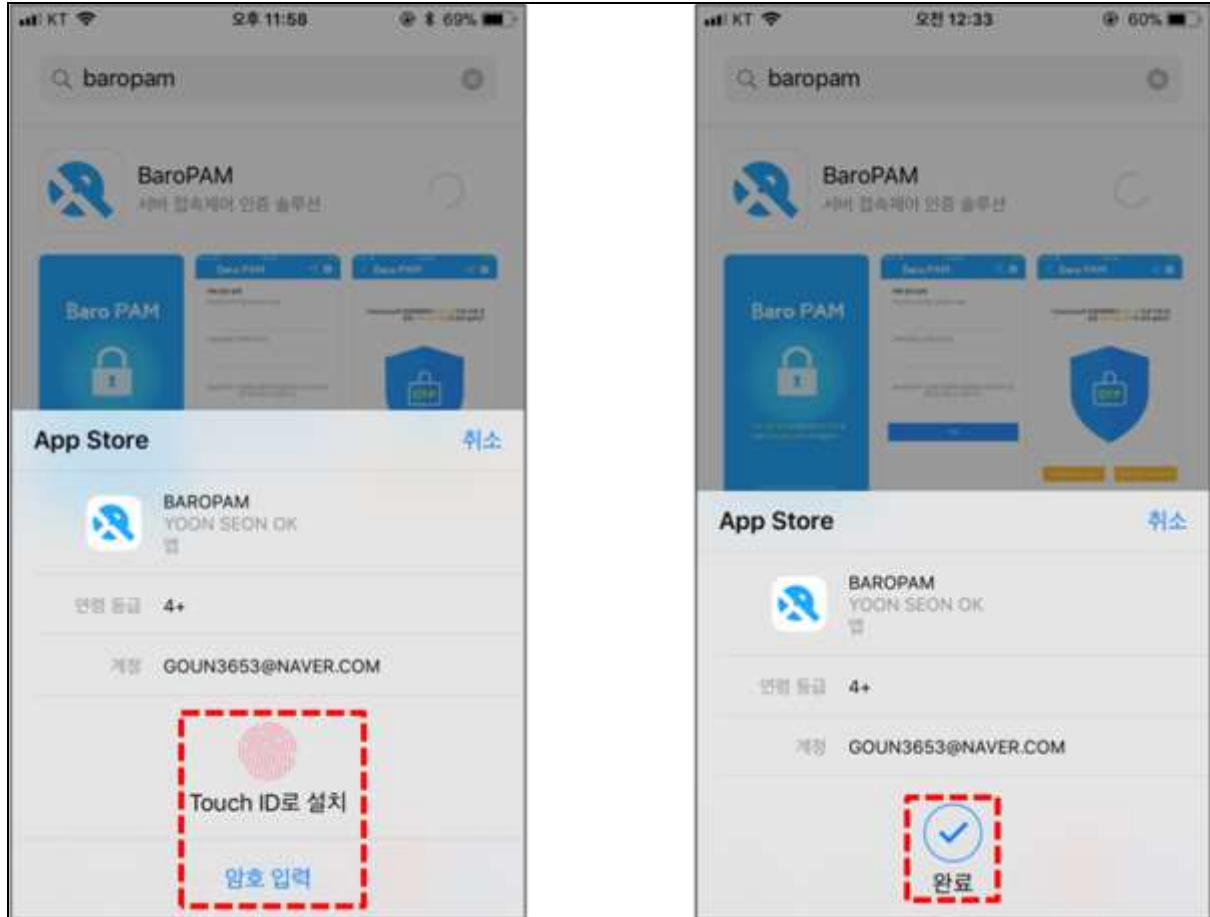


5. "BaroPAM" 앱의 다운로드 후 설치가 다음과 같이 진행된다.

구글 "Pay 스토어"인 경우)



애플 "App Store"인 경우는 다음과 같이 본인을 완료한 후 설치가 진행된다.



6. "BaroPAM" 앱의 설치가 정상적으로 완료되면, 스마트 폰에 다음과 같은 "바로팜, BaroPAMS" 아이콘이 추가된다.

구글 "Pay 스토어"인 경우)



애플 "App Store"인 경우)



"BaroPAM" 앱을 설치한 후 iOS12 부터는 더욱 편리한 **암호 자동 완성 기능**을 반드시 설정(아이폰의 "설정" -> "암호" -> "암호 자동 완성" -> "허용")해야 한다.

7. 안드로이드 폰인 경우 "BaroPAM" 앱의 권한을 설정하기 위해서 다음과 같이 "설정" 아이콘을 클릭한다.
(아이폰인 경우 8~11 단계는 진행할 필요가 없다.)

안드로이드 폰인 경우)



안드로이드 폰인 경우 권한을 설정하지 않으면 **일회용 인증키**가 생성되지 않는다.

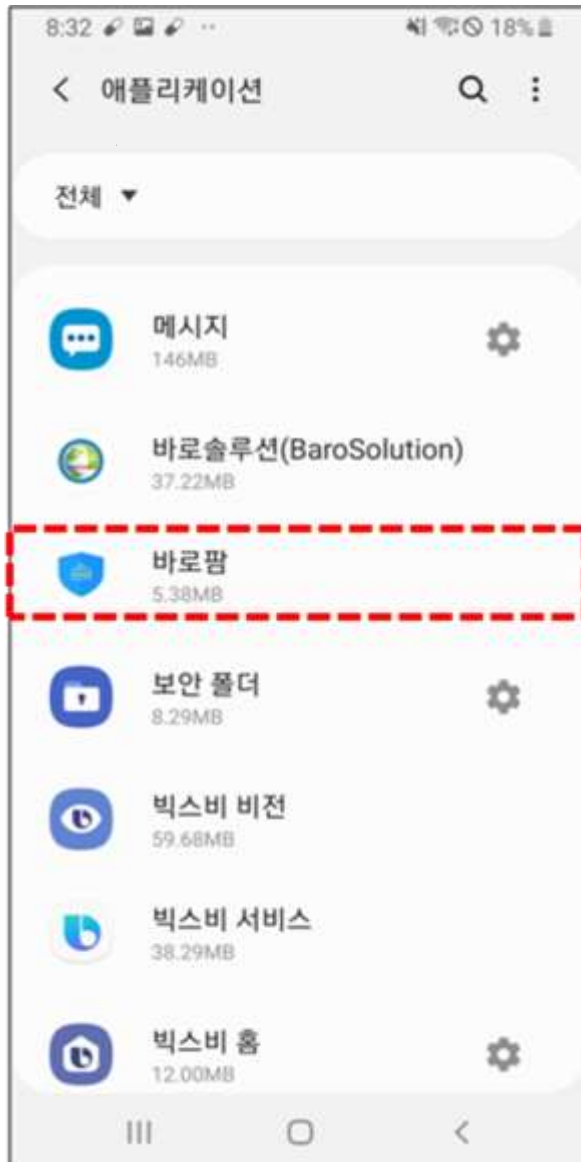
8. "설정" 화면의 아이콘 중에 다음과 같이 "애플리케이션" 아이콘을 클릭한다.

(안드로이드 폰인 경우)



9. "애플리케이션" 화면에서 다음과 같이 "바로팜" 아이콘을 클릭한다.

안드로이드 폰인 경우)



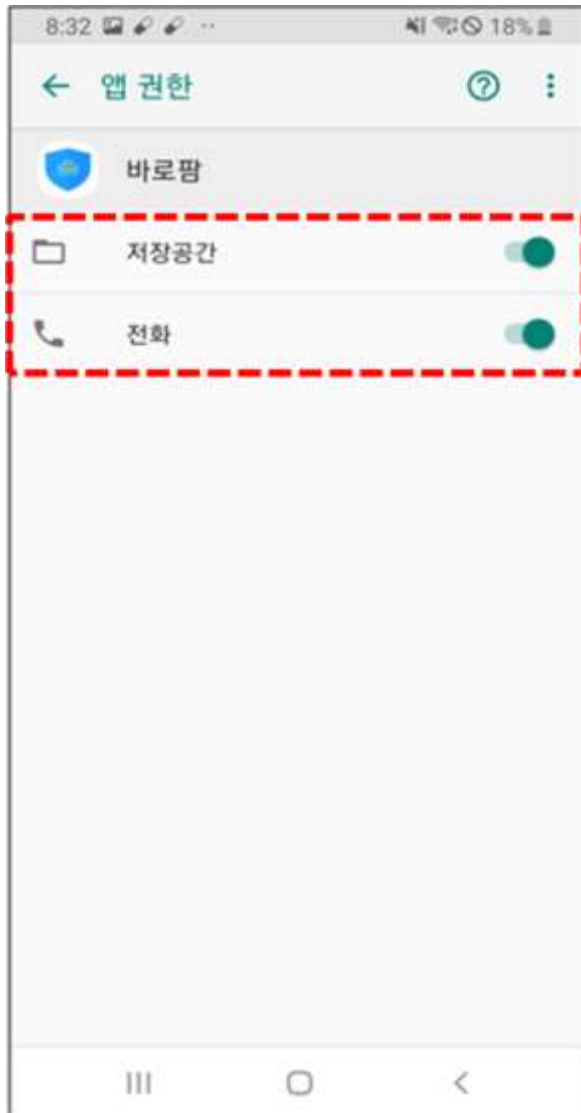
10. "바로팜" 애플리케이션 정보 화면에서 "권한"을 선택하면 "앱 권한"을 설정하는 화면이 나타난다.

(안드로이드 폰인 경우)



11. "바로팜" 앱 권한 화면에서 "저장공간"과 "전화"를 off에서 on으로 다음과 같이 "앱 권한"을 설정한다.

안드로이드 폰인 경우)



"BaroPAM" 앱은 시간 동기화 방식을 사용하므로 폰의 시간 설정이 매우 중요하므로 시간을 자동으로 다음과 같이 설정해야 한다.

안드로이드 폰인 경우: "설정" -> "일반" -> "날짜 및 시간" -> "날짜 및 자동 설정"과 "시간대 자동 설정" -> "허용"
 아이폰인 경우: "설정" -> "날짜 및 시간" -> "자동으로 설정" -> "허용"

"BaroPAM" 앱은 크게 본인확인, 메뉴선택, 애플리케이션 정보, 서버 정보, 환경설정, 정보공유 등으로 구성되어 있다.

1.2 BaroPAM 앱 사용(메뉴선택)

1. "BaroPAM" 앱이 활성화되면 다음과 같이, 다음과 같이 "인증 코드"와 "일회용 인증키"를 선택하는 화면이 나타난다.



"인증 코드"는 PC나 서버 같은 정보자산을 로그인할 때 2차 인증(추가 인증)으로 사용하는 일회용 인증키를 생성하는 경우 사용되며, "일회용 인증키"는 애플리케이션 로그인 화면의 비밀번호를 일회용 인증키로 대체 또는 추가 인증하는 경우에 사용된다.

먼저, "일회용 인증키" 버튼을 클릭한 경우 애플리케이션의 로그인 정보가 존재한 경우는 "애플리케이션 정보 목록" 화면으로 애플리케이션의 로그인에 대한 정보가 존재하지 않은 경우는 "애플리케이션 정보 등록" 화면이 나타난다.

1.3 BaroPAM 앱 사용(애플리케이션정보)

1. "메뉴선택" 화면에 "**일회용 인증키**" 버튼을 클릭한 경우 애플리케이션의 로그인 정보가 존재한 경우는 "애플리케이션 정보 목록" 화면이 나타나며, 애플리케이션의 로그인 정보가 존재하지 않은 경우는 "애플리케이션 정보 등록" 화면이 나타난다.

■시스템명

로그인할 시스템명(System name)은 필수 입력 항목으로 최소 1자리 이상 최대 30자리까지 입력할 수 있다.

■아이디

로그인 아이디(Login-ID)는 필수 입력 항목으로 최소 1자리 이상 최대 50자리까지 입력할 수 있다.

■생성주기(3~60초)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, BaroPAM 검증모듈의 **일회용 인증키**의 생성주기(Auth key cycle time)와 앱에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

2. 애플리케이션의 로그인 정보를 입력한 후 "Save" 버튼을 클릭한다.

The screenshot shows a mobile application interface for 'BaroPAM'. At the top, there's a blue header with a back arrow, the text '바로팜', and a settings gear icon. Below the header, the title '애플리케이션 정보 등록' (Application Information Registration) is displayed. The form contains three input fields: '시스템명' (System Name) with the value 'BaroPAM', '아이디' (ID) with the value 'mc529@nurit.co.kr', and '생성주기(3~60초)' (Auth Key Cycle Time) with the value '20'. At the bottom of the form, there is a blue button labeled 'Save', which is highlighted with a red dashed border. The bottom of the screen shows a standard Android navigation bar with three icons.

"Save" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 애플리케이션의 로그인 정보를 저장하는 작업을 진행한다.

시스템명(System name)이 입력하지 않은 경우 "시스템명을 입력하세요."라는 메시지가 화면에 나타난다.

로그인 아이디(Login-ID)가 입력하지 않은 경우 "아이디를 입력하세요."라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 "인증키 생성주기를 입력 또는 확인해주세요."라는 메시지가 화면에 나타난다.

3. 애플리케이션의 로그인 정보 저장이 정상적으로 완료되지 못하면 "애플리케이션 정보 등록" 화면이 지속되며, 정상적으로 완료되면 다음과 같이 "애플리케이션 정보 목록" 화면이 나타난다.



애플리케이션의 로그인 정보를 검색하고 싶은 경우 검색어를 입력한 다음 "Search" 버튼을 클릭하면 검색된 애플리케이션의 로그인 정보가 화면에 나타난다.

애플리케이션의 로그인 정보를 신규로 추가해야 할 경우 "New" 버튼을 클릭하면 애플리케이션의 로그인 정보를 등록하는 화면이 나타난다.

4. **2차 인증키(일회용 인증키)**를 생성해야 하는 경우 "애플리케이션 정보 목록"에서 애플리케이션 정보 항목을 다음과 같이 클릭한다.

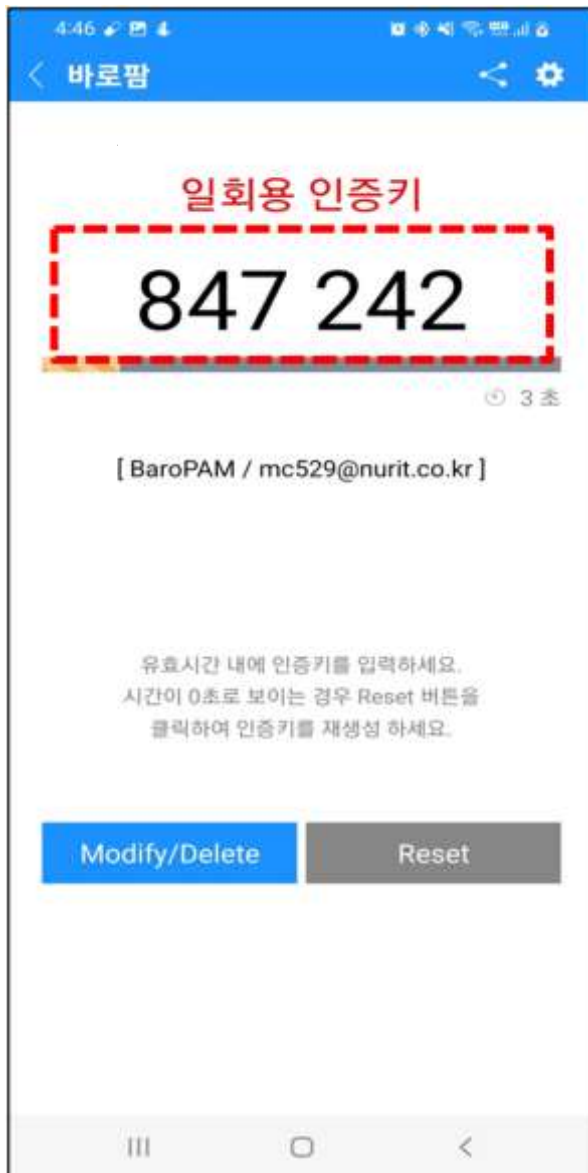


"애플리케이션 정보 목록"에서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 애플리케이션의 로그인 정보 항목을 클릭한다.

그러면, 본인확인 절차를 하지 않은 경우에는 안드로이드 폰과 달리 아이폰(iPhone)은 자체 보안 때문에 기기정보를 얻을 수가 없어서 아이폰인 경우에만 타인의 폰번호를 부정으로 사용하지 못하도록 하기 위해서 별도의 본인확인 기능을 적용할 필요가 있다.

그래서, "BaroPAM" 앱에서는 자체 알고리즘을 적용하여 자체적으로 본인확인 절차("1.4 BaroPAM 앱 사용 (본인확인)" 참조)를 실행하는 화면이 호출된다.

5. 그러면 다음과 같이 애플리케이션의 로그인 정보를 바탕으로 **2차 인증키(일회용 인증키)**를 생성하는 화면이 나타난다.



애플리케이션 로그인 정보의 수정/삭제를 진행해야 하는 경우 "Modify/Delete" 버튼을 클릭하면 된다.

동일한 인증키 생성주기(Auth key cycle time) 동안 "Reset" 버튼은 비활성화되며, 인증키 생성주기(Auth key cycle time)가 지나면 "Reset" 버튼은 활성화시킨다.

활성화된 "Reset" 버튼을 클릭하면 새로운 **2차 인증키(일회용 인증키)**가 생성되는데, **일회용 인증키**의 생성주기(Auth key cycle time) 동안은 동일한 **2차 인증키(일회용 인증키)**가 발생한다.

참고) 생성된 **일회용 인증키**인 "847 242" 숫자를 클릭하면 "Reset" 버튼의 클릭과 동일하게 **일회용 인증키**를 재생성한다.

6. 애플리케이션의 로그인 정보를 수정(Modify) 또는 삭제>Delete) 해야 할 경우 "Modify/Delete" 버튼을 클릭하면 다음과 같은 화면이 나타난다.

■시스템명

로그인할 시스템명(System name)은 필수 입력 항목으로 최소 1자리 이상 최대 30자리까지 입력할 수 있다.

■아이디

로그인 아이디(Login-ID)는 필수 입력 항목으로 최소 1자리 이상 최대 50자리까지 입력할 수 있다.

■생성주기(3~60초)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, BaroPAM 검증모듈의 **일회용 인증키**의 생성주기(Auth key cycle time)와 앱에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 애플리케이션을 로그인

할 수 없는 경우가 발생할 수 있다.

"Delete" 버튼을 클릭하면 해당 애플리케이션의 로그인 정보를 삭제하는 작업을 진행한다.

"Modify" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 애플리케이션의 로그인 정보를 수정하는 작업을 진행한다.

시스템명(System name)이 입력하지 않은 경우 "시스템명을 입력하세요."라는 메시지가 화면에 나타난다.

로그인-ID(Login-ID)가 입력하지 않은 경우 "아이디를 입력하세요."라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 "인증키 생성주기를 입력 또는 확인해주세요."라는 메시지가 화면에 나타난다.

애플리케이션의 로그인 정보 수정(Modify) 또는 삭제>Delete)가 정상적으로 완료되지 못하면 "애플리케이션 정보 수정/삭제" 화면이 지속되며, 정상적으로 완료되면 "애플리케이션 정보 목록" 화면이 나타난다.

1.4 BaroPAM 앱 사용(본인확인)

1. 아이폰(iPhone)인 경우

안드로이드 폰과 달리 아이폰(iPhone)폰의 기기정보를 얻지 못해서 "애플리케이션 정보 목록" 화면에서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 애플리케이션의 로그인 정보 항목을 선택했을 때 "일회용 인증키" 생성 화면으로 이동하지 않은 경우가 발생할 수 있다.

또한, 타인의 폰번호를 부정으로 사용하지 못하도록 하기 위해서 별도의 본인확인 기능을 적용할 필요가 있는데, "BaroPAM" 앱에서는 자체 알고리즘을 적용하여 자체적으로 본인확인 절차를 다음과 같이 실행하고 있다.

1) "애플리케이션 정보 목록"에서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 애플리케이션의 로그인 정보 항목을 클릭을 클릭한다.

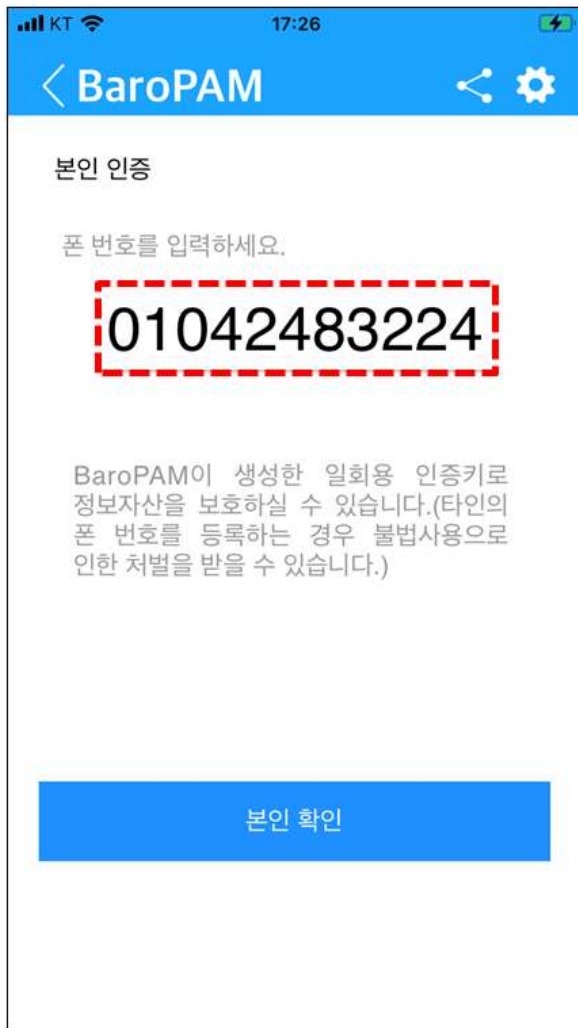


2) 본인확인 절차를 수행하지 않은 경우 다음과 같이 본인확인을 위한 화면이 나타난다.



The image shows a mobile app interface for 'BaroPAM'. At the top, there's a blue header with a back arrow, the text 'BaroPAM', and a settings gear icon. Below the header, the title '본인 인증' (Self-authentication) is displayed. Underneath, a prompt says '폰 번호를 입력하세요.' (Enter your phone number). A horizontal line indicates where to enter the number. Below this, a paragraph of text reads: 'BaroPAM이 생성한 일회용 인증키로 정보자산을 보호하실 수 있습니다.(타인의 폰 번호를 등록하는 경우 불법사용으로 인한 처벌을 받을 수 있습니다.)' (You can protect your information assets with a one-time authentication key generated by BaroPAM. (If you register someone else's phone number, you may be punished for illegal use.)). At the bottom, there is a large blue button with the text '본인 확인' (Self-check).

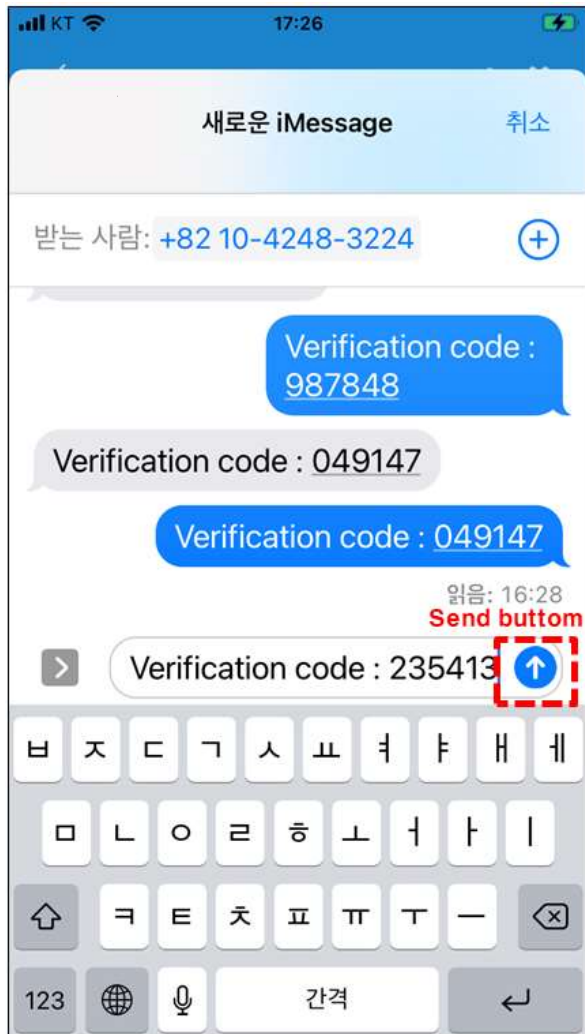
3) 아이폰의 폰번호를 숫자로 입력하고 "본인 확인" 버튼을 클릭한다.



"본인 확인" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 본인 폰번호에 대한 확인 작업을 진행한다.

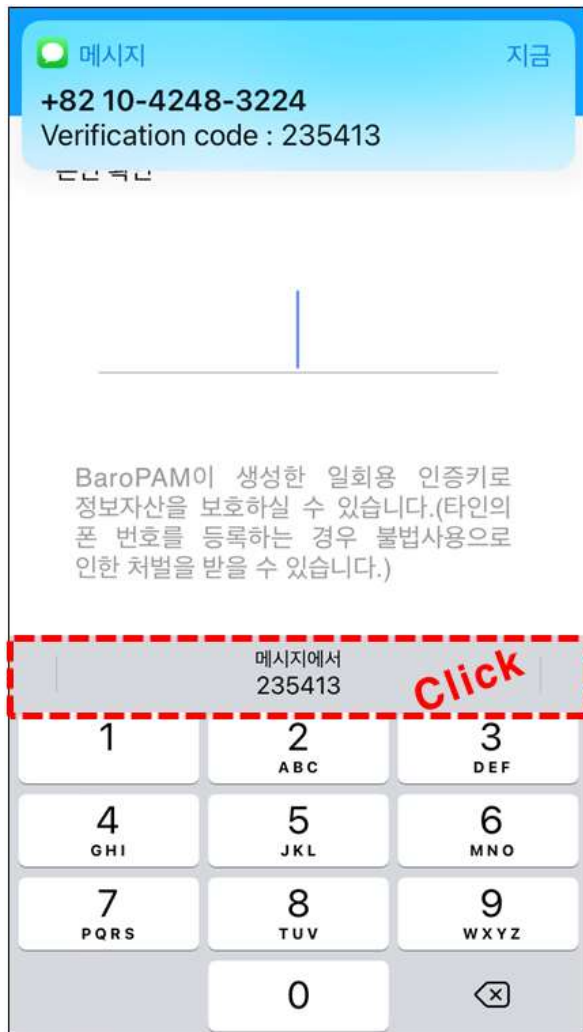
만약, 폰번호가 8자리 이하인 경우 "폰 번호 8자리 이상을 입력하세요."라는 메시지가 화면에 나타난다.

4) 정상적으로 완료되면 본인의 폰번호인지 확인 작업을 진행하지 위하여 내부 알고리즘에 의하여 **OTA key(One-Time Authentication key)**를 생성한 다음 생성한 **OTA key**를 입력 폰번호로 전송하기 위하여 다음 화면에서 "SMS 전송" 버튼 클릭한다.



만약, 타인의 폰번호를 입력하는 경우 불법 사용으로 인한 처벌을 받을 수 있다.

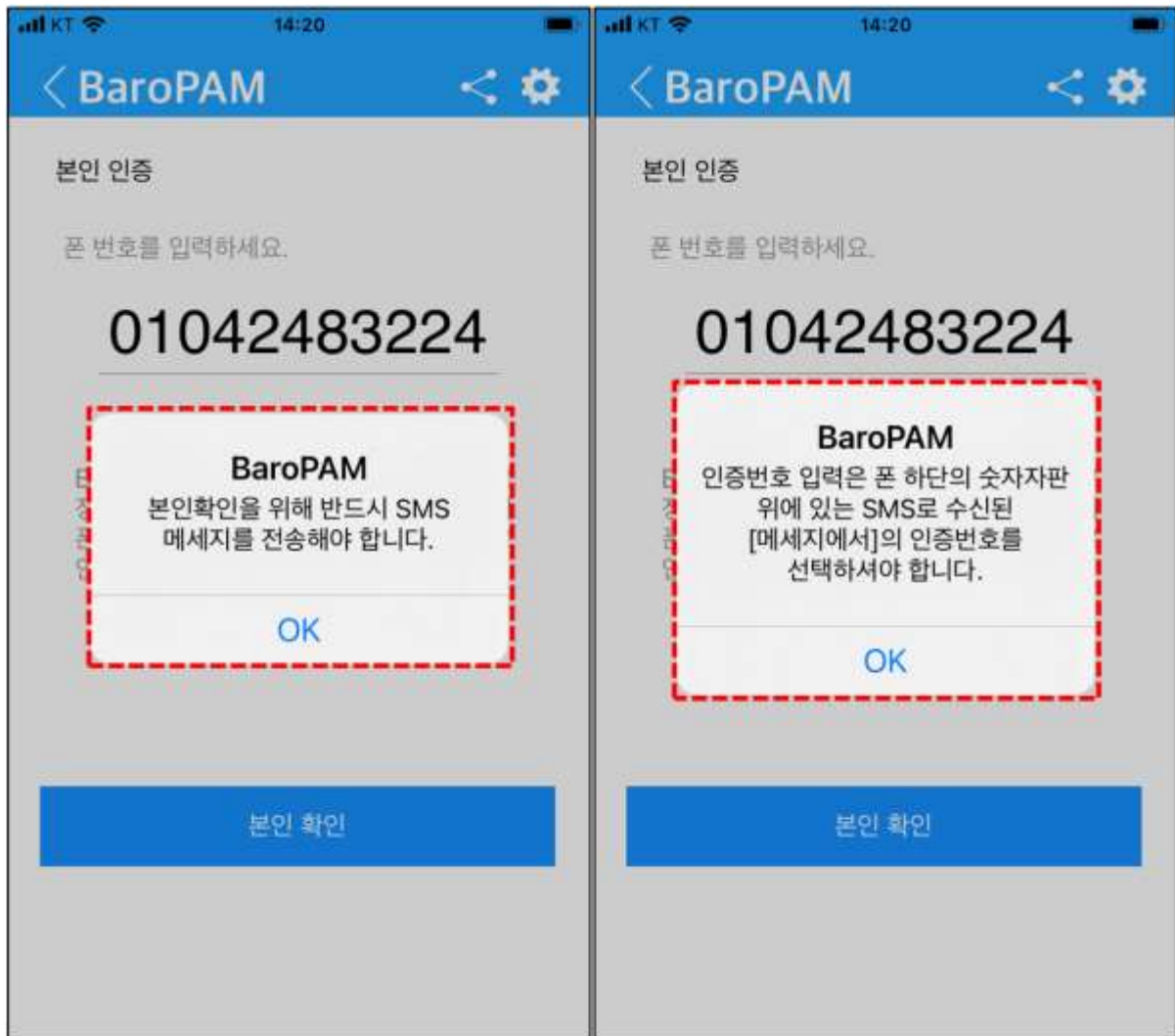
5) 그러면 다음과 같이 화면에 SMS로 전송한 **OTA key**가 하단의 키펀 위에 나타난다.



키펀 위에 표시된 **OTA key**를 클릭하면 내부 **OTA key** 검증모듈에 의하여 검증작업 후 폰 번호를 저장하는 작업을 진행한다.

위 화면처럼 키펀 위에 **OTA key**가 표시되지 않으면 iOS12 부터는 더욱 편리한 **암호 자동 완성 기능**을 반드시 설정(아이폰의 "설정" -> "암호" -> "암호 자동 완성" -> "허용")한 후 본인확인 작업을 진행해야 한다.

만약, SMS 전송을 하지 않고 **OTA key**를 직접 입력시 "본인확인을 위해 반드시 SMS 메시지를 전송해야 합니다." 또는 "인증번호 입력은 폰 하단의 숫자자판 위에 있는 SMS로 수신된 [메세지에서]의 인증번호를 선택하셔야 합니다."라는 메시지가 화면에 나타난다.



참고) SMS로 전송한 OTA key가 수신은 되었는데 키판 위에 표시되지 않거나 SMS로 전송한 OTA key가 수신되지 않은 경우

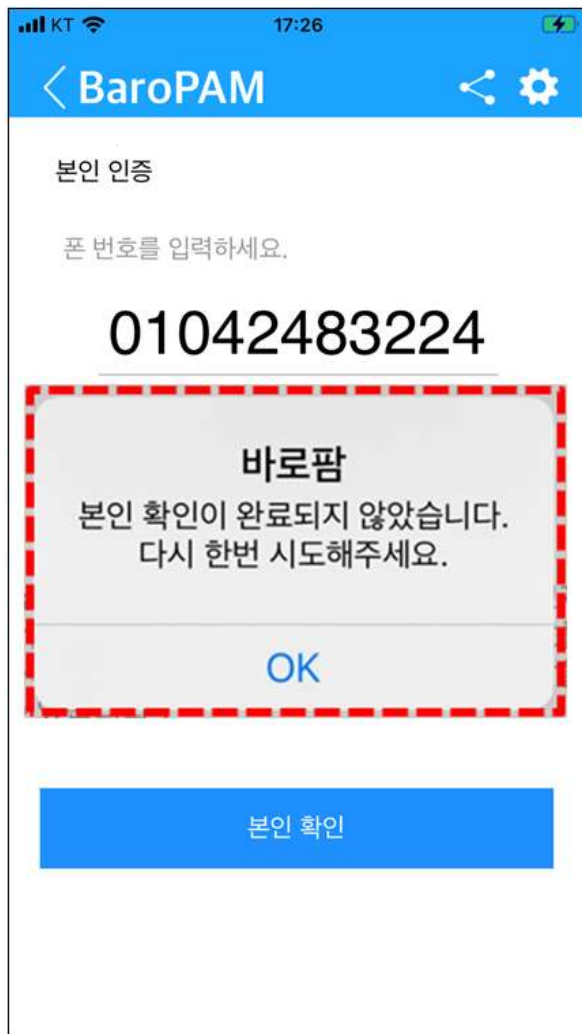


아이폰의 "암호 자동 완성 기능"이 설정되지 않아서 발생한다. "BaroPAM" 앱을 설치한 후 iOS12 부터는 더욱 편리한 암호 자동 완성 기능을 반드시 설정해야 한다. (아이폰의 "설정" -> "암호" -> "암호 자동 완성" -> "허용")

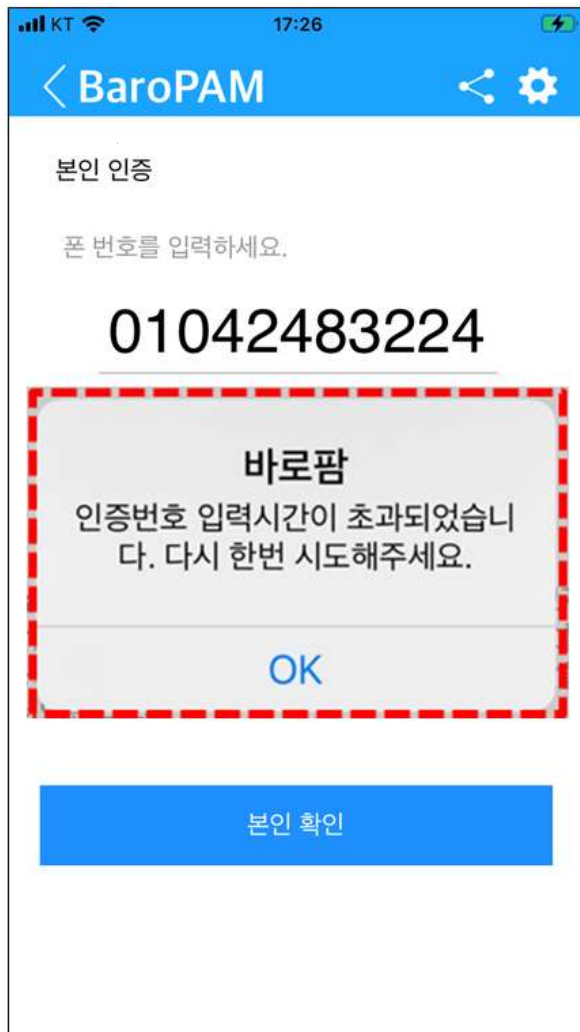
만약, 긴급 발행한 임시 인증코드인 OTA key 검증에 실패한 경우는 다음과 같은 메시지가 화면에 나타나며, 긴급 발행한 임시 인증코드인 OTA key 검증을 다시 진행해야 한다.



6) 만약, 본인확인 작업에 실패한 경우는 다음과 같은 메시지가 화면에 나타나며, 본인확인 작업을 다시 진행해야 한다.



7) 본인확인 작업 중 **OTA key** 입력시간이 **10초를 초과**한 경우 다음과 같은 메시지가 화면에 나타나며, 본인확인 작업을 다시 진행해야 한다.



2. 안드로이드 폰인 경우

안드로이드 폰의 기기정보를 얻지 못해서 "애플리케이션 정보 목록" 화면에서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 애플리케이션의 로그인 정보 항목을 선택했을 때 "**일회용 인증키**" 생성 화면으로 이동하지 않은 경우가 발생할 수 있다.

또한, 타인의 폰번호를 부정으로 사용하지 못하도록 하기 위해서 별도의 본인확인 기능을 적용할 필요가 있는데, "BaroPAM" 앱에서는 자체 알고리즘을 적용하여 자체적으로 본인확인 절차를 다음과 같이 실행하고 있다.

1) "BaroPAM" 앱의 환경설정은 다음과 같은 "환경설정" 아이콘을 클릭하면 된다.



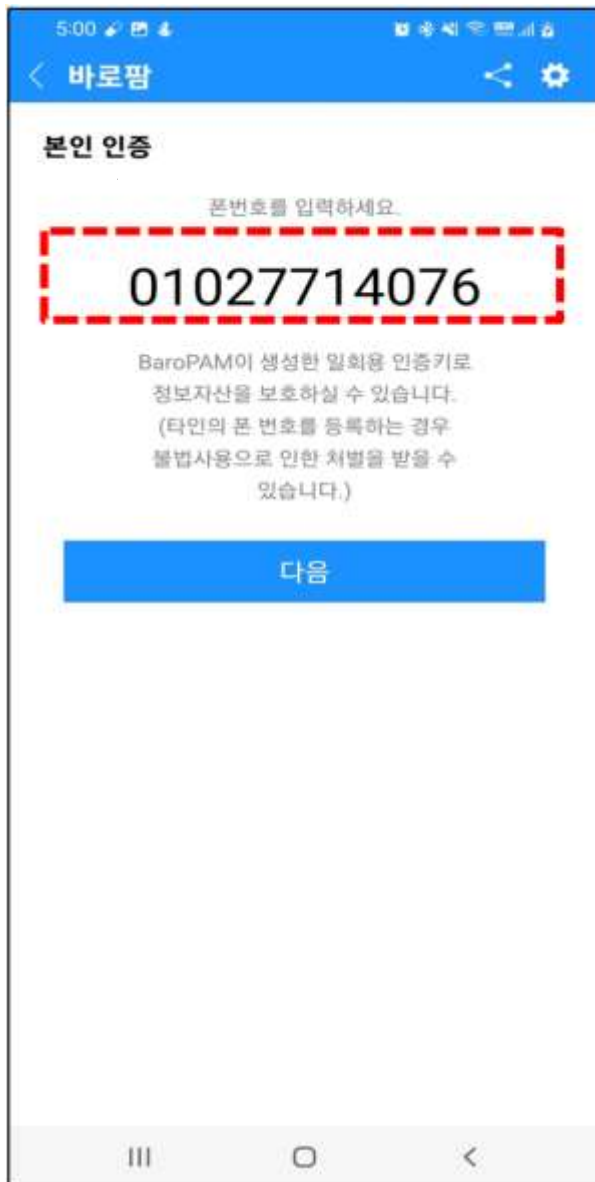
2) 그러면 다음과 같이 "환경설정" 화면이 나타난다.



3) "본인 인증" 항목을 클릭하면 다음과 같은 "본인 인증" 화면이 나타난다.



4) 본인의 폰번호를 숫자로 입력하고 "다음" 버튼을 클릭한다.



"다음" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 본인확인 작업을 진행하기 위하여 내부 알고리즘에 의하여 생성한 **승인번호**를 "본인 확인" 화면으로 전달한다.

폰번호가 8자리 이하인 경우 "폰 번호 8자리 이상을 입력하세요."라는 메시지가 화면에 나타난다.

만약, 타인의 폰번호를 입력하는 경우 불법 사용으로 인한 처벌을 받을 수 있다.

5) 전달받은 **승인번호**를 확인한 후 "본인 확인" 버튼을 클릭한다.



승인번호를 확인한 후 "본인 확인" 버튼을 클릭하면 내부 알고리즘에 의하여 **승인번호**를 검증한 후 폰 번호를 저장하는 작업을 진행한다.

본인확인 작업이 성공한 경우 "메뉴 선택" 화면으로 이동하지만 실패한 경우는 "본인 인증" 화면으로 이동한다.

1.5 BaroPAM 앱 사용(서버정보)

1. 정보자산에 대한 정보가 설정되어 있지 않은 경우("BaroPAM" 앱을 처음 사용하는 경우) 다음과 같이 정보자산에 대한 정보를 등록하는 화면이 나타난다.

■서버명

BaroPAM이 운영될 서버명(Server name)은 필수 입력 항목으로 최소 1자리 이상 최대 30자리까지 입력할 수 있다.

■Secure key

정보자산별 또는 계정별로 부여되는 Secure key는 필수 입력 항목으로 벤더에 요청하여 부여받은 것을 입력해야 한다.

벤더에서 부여받지 않은 임의의 "Secure key"를 입력하는 경우 잘못된 **일회용 인증키**가 부여되어 정보자산에 로그인 할 수 없는 경우가 발생할 수 있다.

만약, 정보자산에 설정한 Secure key와 앱에서 지정한 Secure key가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

■생성주기(3~60초)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, 정보자산에 설정한 **일회용 인증키**의 생성주기(Auth key cycle time)와 앱에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

2. 정보자산에 대한 정보를 입력한 후 "Save" 버튼을 클릭한다.

"Save" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 정보자산에 대한 정보를 저장하는 작업을 진행한다.

정보자산명(Server name)이 입력하지 않은 경우 "서버명을 입력하세요."라는 메시지가 화면에 나타난다.

Secure key가 입력하지 않은 경우 "Secure key를 입력하세요."라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 "인증키 생성주기를 입력 또는 확인해주세요."라는 메시지가 화면에 나타난다.

3. 정보자산에 대한 정보 저장이 정상적으로 완료되지 못하면 "서버 정보 등록" 화면이 지속되며, 정상적으로 완료되면 다음과 같이 "서버 정보 목록" 화면이 나타난다.



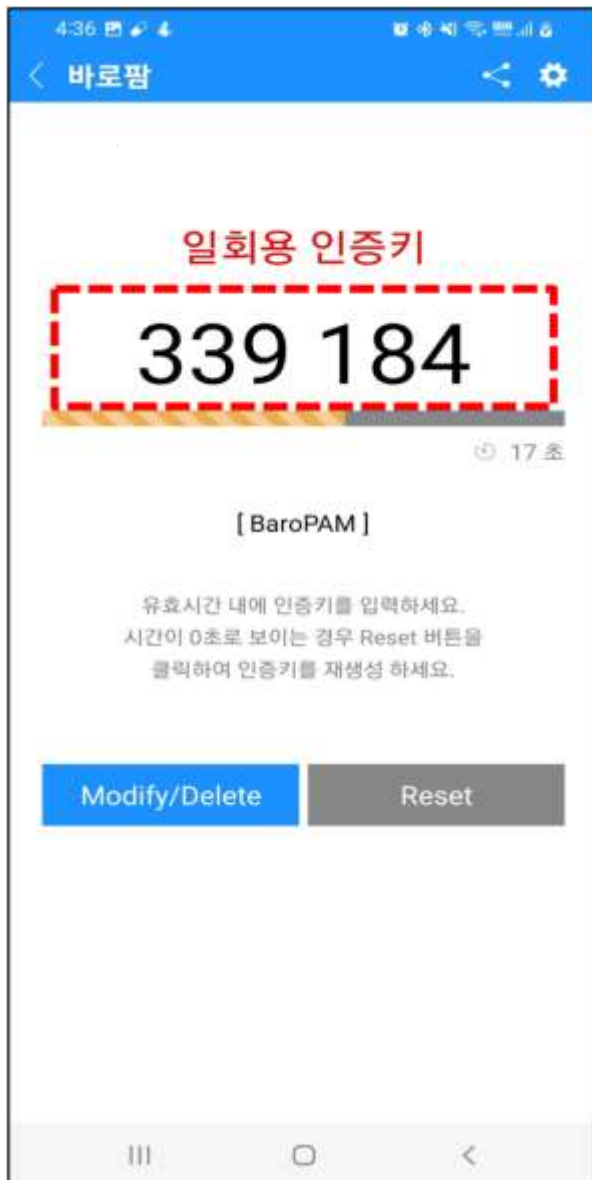
정보자산에 대한 정보를 검색하고 싶은 경우 검색어를 입력한 다음 "Search" 버튼을 클릭하면 검색된 정보 자산에 대한 정보가 화면에 나타난다.

정보자산에 대한 정보를 신규로 추가해야 할 경우 "New" 버튼을 클릭하면 정보자산에 대한 정보를 등록하는 화면이 나타난다.

4. **2차 인증키(일회용 인증키)**를 생성해야 하는 경우 "서버 정보 목록"에서 정보자산에 대한 정보 항목을 다음과 같이 클릭한다.



5. 그러면 다음과 같이 정보자산에 대한 정보를 바탕으로 **2차 인증키(일회용 인증키)**를 생성하는 화면이 나타난다.



정보자산에 대한 정보의 수정/삭제를 진행해야 하는 경우 "Modify/Delete" 버튼을 클릭하면 된다.

동일한 인증키 생성주기(Auth key cycle time) 동안 "Reset" 버튼은 비활성화되며, 인증키 생성주기(Auth key cycle time)가 지나면 "Reset" 버튼은 활성화시킨다.

활성화된 "Reset" 버튼을 클릭하면 새로운 **2차 인증키(일회용 인증키)**가 생성되는데, **일회용 인증키**의 생성주기(Auth key cycle time) 동안은 동일한 **2차 인증키(일회용 인증키)**가 발생한다.

참고) 생성된 **일회용 인증키**인 "339 184" 숫자를 클릭하면 "Reset" 버튼의 클릭과 동일하게 **일회용 인증키**를 재생성한다.

6. 정보자산에 대한 정보를 수정(Modify) 또는 삭제>Delete) 해야 할 경우 "Modify/Delete" 버튼을 클릭하면 다음과 같은 화면이 나타난다.

■서버명

BaroPAM이 운영될 정보자산명(Server name)은 필수 입력 항목으로 최소 1자리 이상 최대 30자리까지 입력할 수 있다.

■Secure key

정보자산별 또는 계정별로 부여되는 Secure key는 필수 입력 항목으로 벤더에 요청하여 부여받은 것을 입력해야 한다.

벤더에서 부여받지 않은 임의의 "Secure key"를 입력하는 경우 잘못된 **일회용 인증키**가 부여되어 정보자산에 로그인 할 수 없는 경우가 발생할 수 있다.

만약, 정보자산에 설정한 Secure key와 앱에서 지정한 Secure key가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

■생성주기(3~60초)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, 정보자산에 설정한 **일회용 인증키**의 생성주기(Auth key cycle time)와 앱에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

"Delete" 버튼을 클릭하면 해당 정보자산에 대한 정보를 삭제하는 작업을 진행한다.

"Modify" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 정보자산에 대한 정보를 수정하는 작업을 진행한다.

정보자산명(Server name)이 입력하지 않은 경우 "서버명을 입력하세요."라는 메시지가 화면에 나타난다.

Secure key가 입력하지 않은 경우 "Secure key를 입력하세요."라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 "인증키 생성주기를 입력 또는 확인해주세요."라는 메시지가 화면에 나타난다.

정보자산에 대한 정보 수정(Modify) 또는 삭제>Delete)가 정상적으로 완료되지 못하면 "서버 정보 수정/삭제" 화면이 지속되며, 정상적으로 완료되면 "서버 정보 목록" 화면이 나타난다.

1.6 BaroPAM 앱 사용(환경설정)

1. "BaroPAM" 앱의 환경설정은 다음과 같은 "환경설정" 아이콘을 클릭하면 된다.



2. 그러면 다음과 같이 "환경설정" 화면이 나타난다.



3. "BaroPAM 업데이트 하기" 항목을 클릭하면 다음과 같은 "Google Play Store" 화면이 나타난다.



"업데이트" 버튼을 클릭하면 "BaroPAM" 앱의 업데이트 작업 이 진행된다. "업데이트" 버튼이 없는 경우는 업데이트할 정보가 없는 경우를 의미한다.

4. "BaroPAM 별점주기" 항목을 클릭하면 다음과 같은 "Google Play Store"의 별점주기 화면이 나타난다.



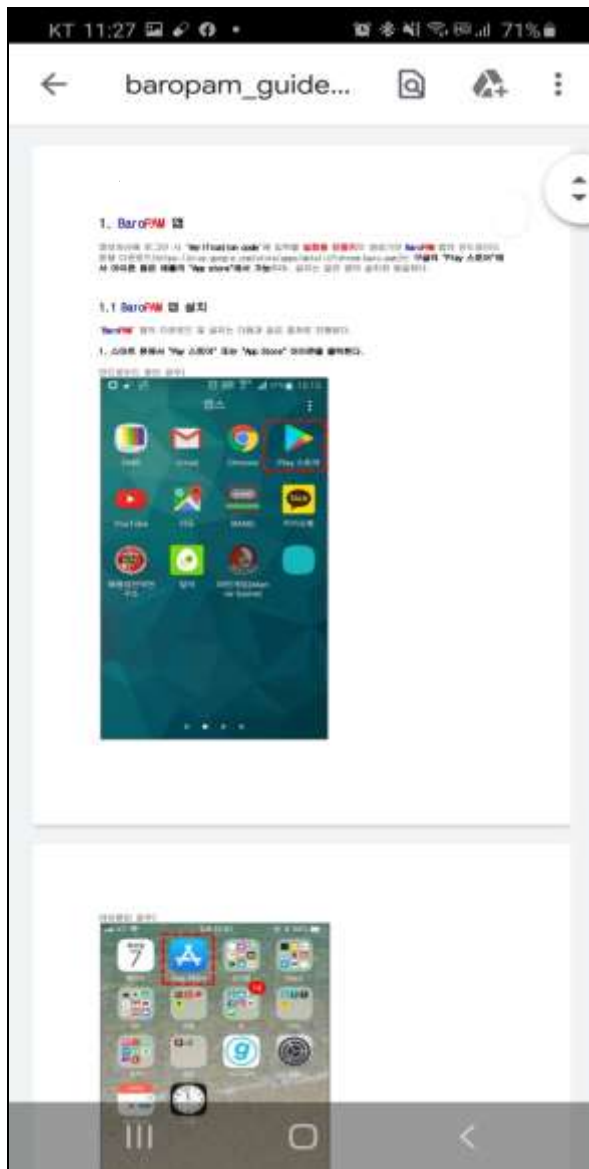
"BaroPAM" 앱의 활성을 위해 별점 및 댓글을 작성하여 제출하면 앱 활성화에 도움이 된다.

5. "BaroPAM 소개서" 항목을 클릭하면 다음과 같은 "BaroPAM 솔루션 소개서" 화면이 나타난다.



"BaroPAM 솔루션 소개서"는 pdf 파일(baropam_intro_kr.pdf) 형태로 제공됩니다.

6. "BaroPAM 가이드" 항목을 클릭하면 다음과 같은 "BaroPAM 앱 사용 설명서" 화면이 나타난다.



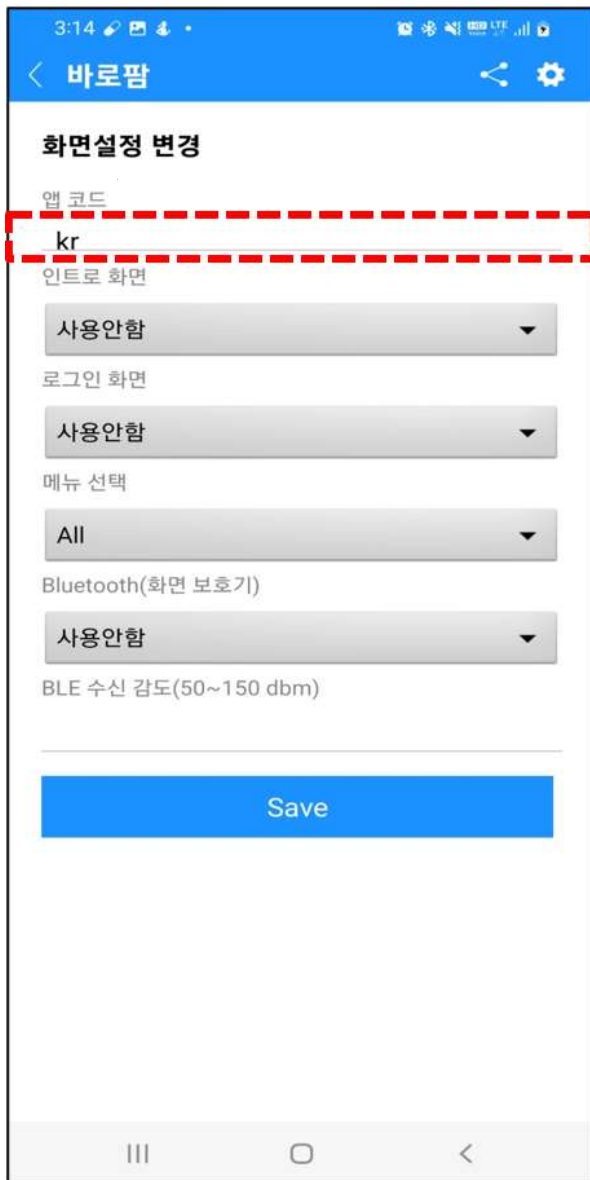
"BaroPAM 앱 사용 설명서"는 pdf 파일(baropam_guide_kr.pdf) 형태로 제공된다.

7. "BaroPAM 영업문의" 항목을 클릭하면 다음과 같은 "전화걸기" 화면이 나타난다.



"전화걸기" 버튼을 클릭하여 "BaroPAM"에 대한 영업 및 전반적인 사항을 문의할 수 있다.

8. "화면설정 변경" 항목을 클릭하면 다음과 같은 "화면설정 변경" 화면이 나타난다.



■ 앱 코드

벤더에서 부여받은 앱 코드를 입력한다. 입력하지 않으면 기본적으로 한글 모드인 "kr"이 지정되며, 입력한 앱코드에 맞게 "BaroPAM" 앱이 전환된다.

예) 앱 코드

한글 모드: kr
영문 모드: en
중국 모드: cn
일본 모드: jp

■ 인트로 화면

"BaroPAM" 앱을 실행 시 인트로 화면의 사용여부를 선택한다. 인트로 화면을 사용할 경우는 "사용함", 인트로 화면을 사용하지 않을 경우는 "사용안함"을 선택한다.

인트로 화면을 사용할 경우 "BaroPAM" 앱을 실행 시, "BaroPAM" 앱이 활성화되면 "BaroPAM"의 인트로 화면이 2초간 활성화되어 다음과 같이 지속된다.



■로그인 화면

"BaroPAM" 앱을 실행 시 로그인 화면의 사용여부를 선택한다. 로그인 화면의 사용여부를 선택한다. 로그인 화면을 사용하지 않은 경우는 "사용안함", PIN 번호를 사용하는 경우는 "PIN 번호 사용함", 지문이나 얼굴인식을 사용하는 경우는 "생체인증 사용함"을 선택한다.

로그인 화면으로 지문이나 얼굴 인식 같은 생체인증을 사용하는 경우에는 "폰설정 -> 생체 인식 및 보안 -> 지문과 얼굴인식"에서 지문과 얼굴인식을 먼저 등록한 후 "BaroPAM" 앱 실행 시 지문이나 얼굴인식을 통하여 "BaroPAM" 앱에 로그인 할 수 있다.

만약, 로그인 화면으로 PIN 번호를 사용할 경우 다음과 같이 "로그인" 화면이 나타난다.



"로그인" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 PIN 번호(PIN Number)가 맞는지를 확인하는 작업을 진행한다.

만약, PIN 번호(PIN Number)가 8자리가 아닌 경우 "PIN 번호 8자리를 입력하세요."라는 메시지가 화면에 나타난다.

설정된 PIN 번호(PIN Number)와 틀린 경우 "PIN번호가 틀립니다. 다시 확인 후 입력하세요."라는 메시지가 화면에 나타난다.

PIN번호를 5회 연속 실패한 경우 "Access denied."라는 메시지가 화면에 나타나고 3초 뒤에 앱을 종료한다.

만약, "BaroPAM" 앱을 실행 시 인트로 화면을 사용할 경우는 인트로 화면이 먼저 2초간 지속된 뒤 로그인 화면으로 이동하며, 로그인 화면을 사용하는 경우는 로그인 화면이 정상적으로 진행한 뒤 메뉴선택 화면이 이동한다.

■메뉴 선택

사용할 메뉴를 선택한다. "메뉴"를 선택하는 화면을 사용하여 서버(정보자산 등) 및 애플리케이션의 로그

인 정보 모두를 사용하는 경우는 "All"를 선택할 경우는, 서버 정보만 사용하는 경우는 "인증 코드", 애플리케이션의 로그인 정보만 사용하는 경우는 "일회용 인증키"를 선택한다.

■Bluetooth(화면 보호기)

Windows와 개방형OS(하모니카OS, 구름OS, TMaxOS)의 Desktop PC에 정보 보호를 위하여 화면 보호기를 설정하는데, 화면 보호기 잠금 방지 및 해제 기능을 사용하는 경우에 선택한다. 화면 보호기 잠금 방지 및 해제 기능을 사용하지 않을 경우는 "사용함", 화면 보호기 잠금 방지 및 해제 기능을 사용하지 않을 경우는 "사용안함"을 선택한다.

저장하는 작업이 정상적으로 완료되지 못하면 "화면설정 변경" 화면이 지속되며, 정상적으로 완료되면 인트로 화면을 사용하는 경우 "인트로" 화면이, 사용하지 않으면 "메뉴선택" 화면이 나타난다.

9. "본인 인증" 항목을 클릭하면 다음과 같은 "본인 인증" 화면이 나타난다.



1) 본인의 폰번호를 숫자로 입력하고 "다음" 버튼을 클릭한다.



"다음" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 본인확인 작업을 진행하기 위하여 내부 알고리즘에 의하여 생성한 **승인번호**를 "본인 확인" 화면으로 전달한다.

폰번호가 8자리 이하인 경우 "폰 번호 8자리 이상을 입력하세요."라는 메시지가 화면에 나타난다.

만약, 타인의 폰번호를 입력하는 경우 불법 사용으로 인한 처벌을 받을 수 있다.

2) 전달받은 **승인번호**를 확인한 후 "본인 확인" 버튼을 클릭한다.

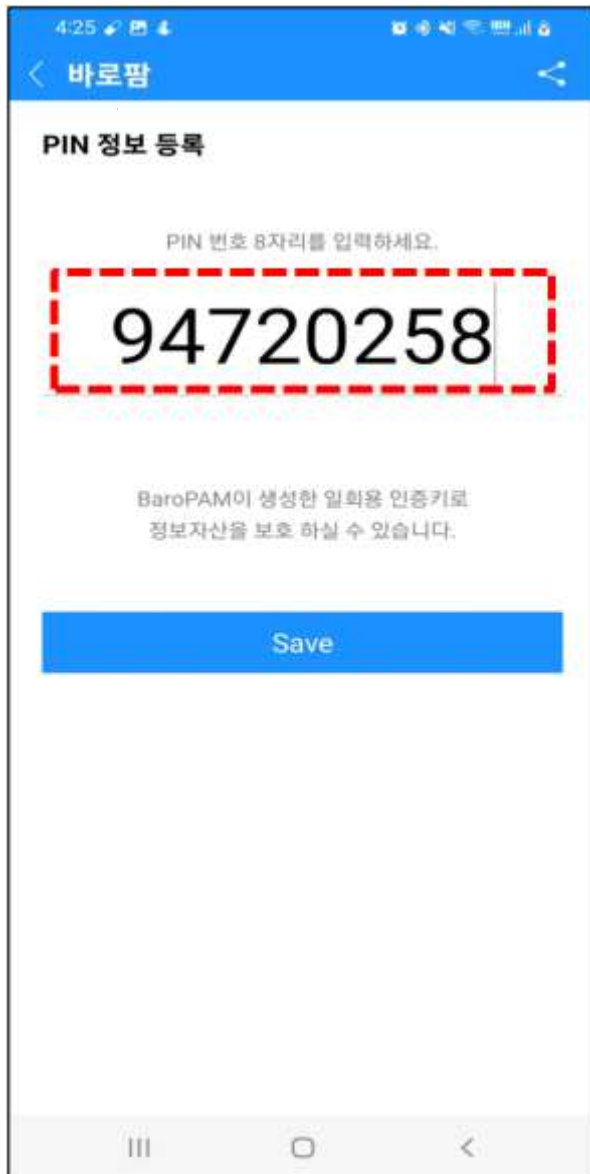


승인번호를 확인한 후 "본인 확인" 버튼을 클릭하면 내부 알고리즘에 의하여 **승인번호**를 검증한 후 폰 번호를 저장하는 작업을 진행한다.

본인확인 작업이 성공한 경우 "메뉴 선택" 화면으로 이동하지만 실패한 경우는 "본인 인증" 화면으로 이동한다.

10. "PIN 번호 변경" 항목을 클릭하면 "화면설정 변경"에서 "로그인 화면" 사용함을 선택한 경우에 다음과 같은 "PIN 번호 변경 또는 PIN번호 등록" 화면이 나타난다.

PIN 번호가 등록되어 있지 않은 경우)

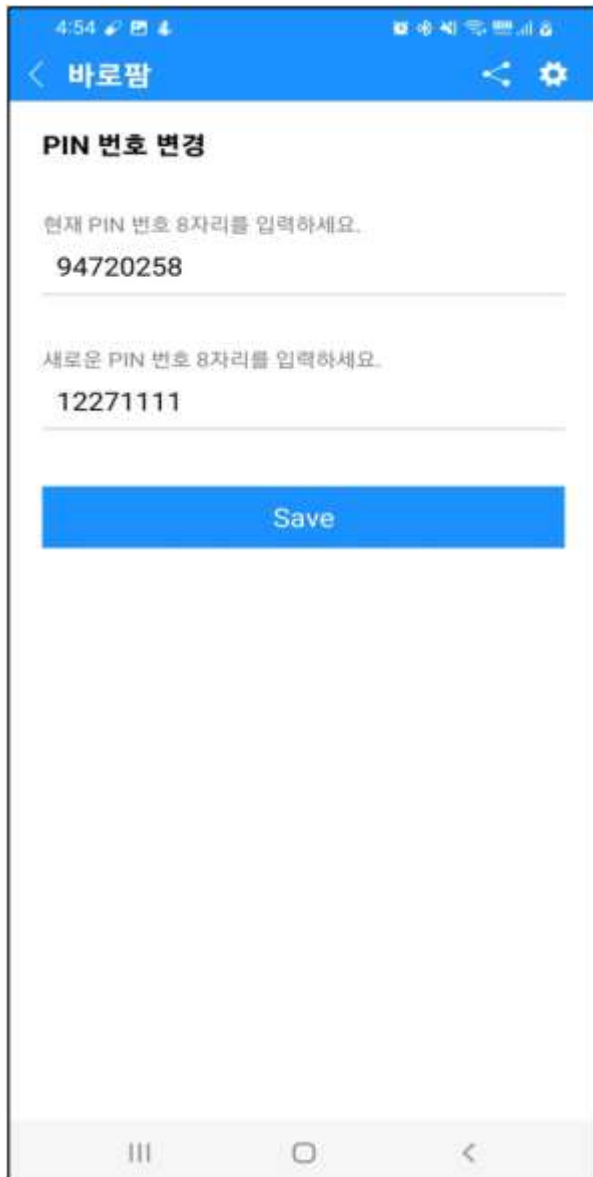


"Save" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 PIN 번호(PIN Number)를 저장하는 작업을 진행한다.

만약, PIN 번호(PIN Number)가 8자리가 아닌 경우 "PIN 번호 8자리를 입력하세요."라는 메시지가 화면에 나타난다.

PIN 번호(PIN Number) 저장이 정상적으로 완료되지 못하면 "PIN 번호 등록" 화면이 지속되며, 정상적으로 완료되면 인트로 화면을 사용하는 경우 "인트로" 화면이, 사용하지 않으면 "메뉴선택" 화면이 나타난다.

PIN번호가 등록된 경우)



■ 현재 PIN번호 8자리를 입력하세요.

현재 설정된 PIN 번호(PIN Number)를 입력한다.

■ 새로운 PIN번호 8자리를 입력하세요.

새로운 설정할 PIN 번호(PIN Number)를 입력한다.

"Save" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 새로운 PIN 번호(PIN Number)를 저장하는 작업을 진행한다.

만약, PIN 번호(PIN Number)가 8자리가 아닌 경우 "PIN 번호 8자리를 입력하세요."라는 메시지가 화면에 나타난다.

현재 설정된 PIN 번호(PIN Number)와 새로 입력한 PIN 번호(PIN Number)가 다른 경우 "현재 PIN번호가 틀립니다. 다시 확인 후 입력하세요."라는 메시지가 화면에 나타난다.

현재 설정된 PIN 번호(PIN Number)와 새로운 PIN 번호(PIN Number)가 동일한 경우 "현재 PIN번호와 새로운

PIN번호가 동일합니다. 다시 확인 후 입력하세요."라는 메시지가 화면에 나타난다.

만약, 새로운 PIN 번호(PIN Number)가 8자리가 아닌 경우 "새로운 PIN 번호 8자리를 입력하세요."라는 메시지가 화면에 나타난다.

새로운 PIN 번호(PIN Number) 저장에 정상적으로 완료되지 못하면 "PIN 번호 변경" 화면이 지속되며, 정상적으로 완료되면 인트로 화면을 사용하는 경우 "인트로" 화면이, 사용하지 않으면 "메뉴선택" 화면이 나타난다.

11. "환경설정 초기화" 항목을 클릭하면 다음과 같은 "환경설정 초기화" 화면이 나타난다.



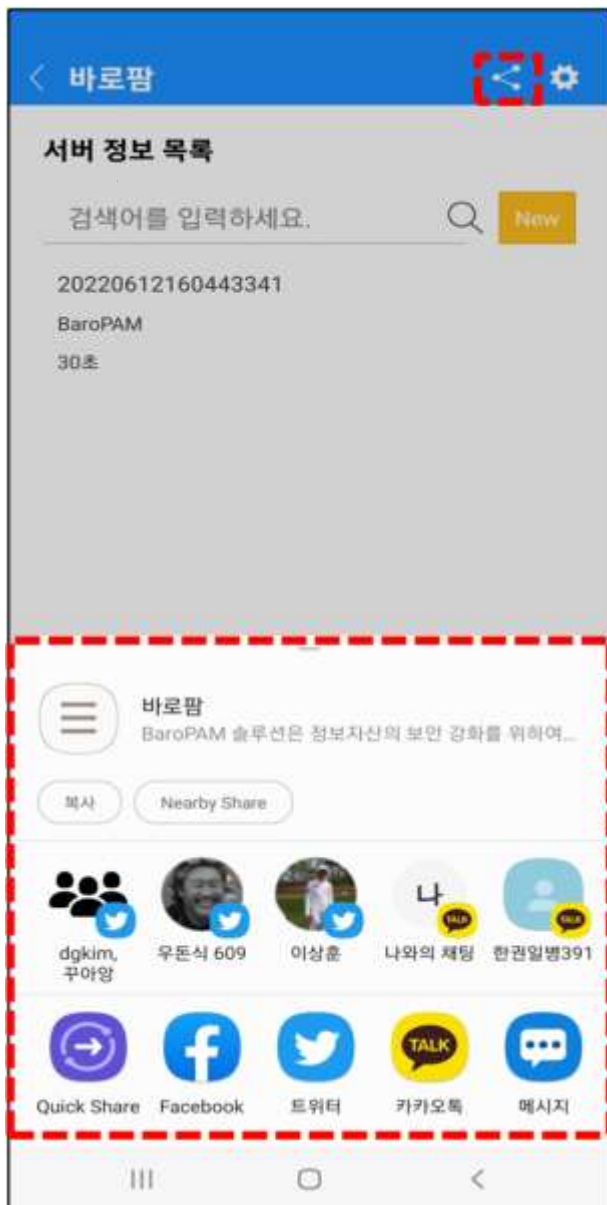
"Yes" 버튼을 클릭하면 "BaroPAM" 앱에서 설정한 환경 정보를 초기화하는 작업을 진행한다.

1.7 BaroPAM 앱 사용(정보공유)

1. "BaroPAM" 앱의 SNS를 이용한 공유는 다음과 같은 "BaroPAM 공유" 아이콘을 클릭하면 된다.



2. 그러면 다음과 같이 "BaroPAM 공유" 할 수 있는 화면이 나타난다.



3. 트위터에 트윗을 하고자 할 경우 "트윗하기" 아이콘을 클릭하면 다음과 같은 화면이 나타난다.



"트윗하기" 아이콘을 클릭하면 기 설정된 트위터 계정으로 트윗 작업이 진행된다.

1.8 BaroPAM 앱 FAQ

현상 : 아이폰인 경우 "BaroPAM" 앱을 설치한 후 본인확인 시 Keypad 위에 Verification code (메시지에서 123456)가 표시되지 않는 경우

원인 : 아이폰의 "암호 자동 완성 기능"이 설정되지 않아서 발생

조치 : "BaroPAM" 앱을 설치한 후 iOS12 부터는 더욱 편리한 암호 자동 완성 기능을 반드시 설정 (아이폰의 "설정" -> "암호" -> "암호 자동 완성" -> "허용")

현상 : 안드로이드폰 또는 아이폰의 날짜와 시간이 현재 시간과 차이가 발생하여 "일회용 인증키"가 맞지 않은 경우

원인 : 안드로이드폰 또는 아이폰의 날짜와 시간을 네트워크에서 제공하는 시간을 사용하지 않아서 발생.

조치 : 안드로이드폰인 경우는 폰의 "설정" -> "일반" -> "날짜 및 시간" -> "날짜 및 자동 설정"과 "시간대 자동 설정" -> "허용"
아이폰인 경우는 폰의 "설정" -> "날짜 및 시간" -> "자동으로 설정" -> "허용"

현상 : 특정유형은 본인 폰번호로 메시지를 보낸 경우 발신은 되는데 수신이 안되는 경우
아이폰에서 본인 폰번호로 메시지 전송이 안되는 경우

원인 : 메시지 관련 아이폰 설정 상의 문제

조치 : 아이폰을 reboot 한 후에 해결

우선, 본인의 iPhone로 메시지가 전송이 안되는 경우를 확인하기 위해 다음을 확인해 주십시오.

1. 설정-메시지에서

- iMessage 활성화 여부 확인
- iMessage가 활성화되어 있는 경우 발신 및 수신 안에 본인의 폰번호가 등록되어 있는지 확인
- SMS로 보내기가 활성화되어 있는지 확인
- SMS/MMS 란의 MMS 메시지 활성화 확인
- SMS/MMS란의 차단된 연락처에서 본인의 폰번호가 등록되어 있는지 확인, 등록되어 있으면 제거할 것.
- 맨션 란에서 알림받기 활성화

2. 아이폰의 메시지 앱에서 직접 자신의 iPhone 번호로 새로운 메시지를 전송해 본다.

- 아예 전송이 안되는 경우
- 전송은 되지만 수신이 안되는 경우

위 경우에 안되는 경우

타인이 본인의 폰번호로 메시지를 보내서 정상으로 수신이 되면, 다음 URL을 참고하여 case 별로 확인해 주십시오.

>> If you can't send or receive messages on your iPhone, iPad, or iPod touch
<https://support.apple.com/en-us/HT204065>

>> 'Why is my iPhone not sending messages?': How to troubleshoot iPhone messaging issues
<https://www.businessinsider.com/why-is-my-iphone-not-sending-messages>

>> 문자 메시지 전달 설정하기
<https://support.apple.com/ko-kr/HT208386>

1.9 BaroPAM 연동 API

1. 애플리케이션 로그인

1.1 BaroPAM 로그인 화면 예시)



The image shows a web-based login interface for BaroPAM. At the top, the 'BaroPAM' logo is displayed in blue and orange. Below the logo, there are two input fields: the first is labeled '이메일 주소를 입력해주시요.!!!' (Please enter your email address!!!) and the second is labeled '일회용 인증키를 입력해주시요.!!!' (Please enter your one-time authentication key!!!). Below these fields is a large blue button with the text '로그인' (Login). At the bottom of the interface, a copyright notice reads 'Copyright © 2012 Nurit. All rights reserved.'

1.2 인증키 검증 부분

애플리케이션에 로그인 시 비밀번호란에 입력한 **일회용 인증키**를 검증하는 API는 "**barokey.jar**"로 제공되며, WAS(Web application Server)의 lib 디렉토리 "**barokey.jar**"를 위치시키거나 classpath에 "**barokey.jar**"가 존재하는 디렉토리를 포함해서 설정해 주면 된다.

```
[root] /home/tomcat/webapps/ROOT/WEB-INF/lib > ls -al
합계 4908
drwxr-xr-x 2 root root    4096  5월  8 11:25 .
drwxr-xr-x 5 root root    4096  5월  9 15:12 ..
-rw-r--r-- 1 root root     116  3월 13 2015 .bash_history
-rw-r--r-- 1 root root  26074  6월 20 20:49 barokey.jar
-rw-r--r-- 1 root root  57779  5월 24 2011 commons-fileupload-1.2.1.jar
-rw-r--r-- 1 root root 109043  5월 24 2011 commons-io-1.4.jar
-rw-r--r-- 1 root root  60841  5월 24 2011 commons-logging-1.1.1.jar
-rw-r--r-- 1 root root  26520  5월 24 2011 commons-logging-adapters-1.1.1.jar
-rw-r--r-- 1 root root  56404  5월 24 2011 cos.jar
-rw-r--r-- 1 root root 313898  5월 24 2011 dom4j-1.6.1-goldkeby.jar
-rw-r--r-- 1 root root  19679  6월  4 2014 gcm-server.jar
-rw-r--r-- 1 root root 341207 10월  6 2008 j2ssh-common-0.2.9.jar
-rw-r--r-- 1 root root 355141 10월  6 2008 j2ssh-core-0.2.9.jar
-rw-r--r-- 1 root root 110582 10월  6 2008 j2ssh-dameon-0.2.9.jar
-rw-r--r-- 1 root root 456805  5월  6 2016 j2ssh-maverick-1.5.5.jar
-rw-r--r-- 1 root root 258160 10월 27 2011 jai_codec.jar
-rw-r--r-- 1 root root 1900631 10월 27 2011 jai_core.jar
```

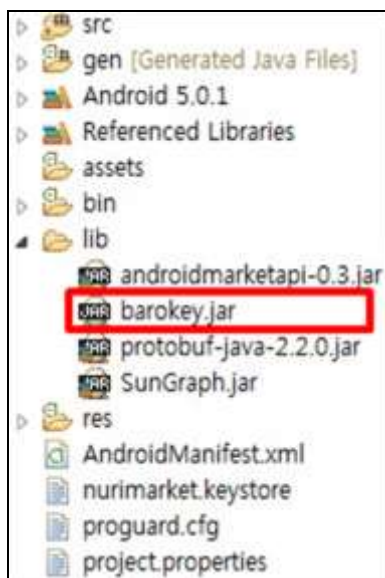
```
-rw-r--r-- 1 root root 464938 3월 6 2012 jimiproclasses-sabisung.jar
-rw-r--r-- 1 root root 23737 6월 4 2014 json-simple-1.1.1.jar
-rw-r--r-- 1 root root 30202 5월 24 2011 json.jar
-rw-r--r-- 1 root root 312603 1월 25 2012 twitter4j-core-2.2.5.jar
```

애플리케이션에 로그인 시 입력한 비밀번호인 **일회용 인증키**를 검증하는 프로그램에 다음과 같은 코드를 삽입하면 된다.

1.3 안드로이드인 경우

1) 인증키 생성기 부분

애플리케이션에 로그인 시 비밀번호란에 입력할 **일회용 인증키**를 생성하는 API는 "barokey.jar"로 제공되며, Eclipse을 사용하는 경우 libs 디렉토리에 "barokey.jar"를 위치시켜야 한다.

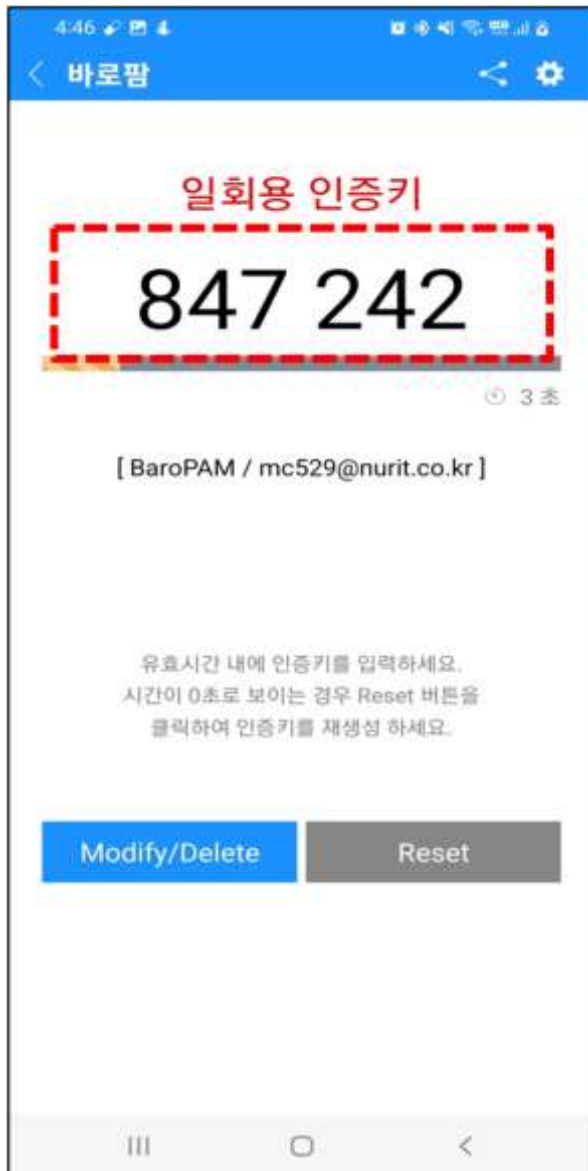


애플리케이션에 로그인 시 입력할 비밀번호인 **일회용 인증키**를 생성하는 프로그램에 다음과 같은 코드를 삽입하면 된다.

```
...
import com.barokey.barokey;
...
String tkey = barokey.generateKEYL(String login_id, String phone_no, String cycle_time);
...
```

파라미터	설명	비고
login_id	로그인 화면의 로그인-ID 항목에 입력한 ID를 설정.	
phone_no	TelephonyManager를 이용해서 앱 내부에서 얻어낸 스마트 폰 번호를 설정.	
cycle_time	개인별로 지정한 일회용 인증키 의 생성 주기(3~60초)를 설정. 만약, 개인별로 지정한 일회용 인증키 의 생성 주기가 다른 경우 일회용 인증키 가 다르게 생성될 수 있다.	

화면 예시)



화면 Layout 예시)

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/bg_body_default"
    android:orientation="vertical">

    <include
        android:id="@+id/inc_header"
        layout="@layout/inc_header"
        android:layout_width="fill_parent"
        android:layout_height="@dimen/head_height" />

    <ScrollView
        android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:layout_marginLeft="@dimen/body_margin_right_default"
        android:layout_marginRight="@dimen/body_margin_right_default"
        android:layout_marginTop="@dimen/head_height">

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/body_frame"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="81dip"
        android:padding="10dp"
        android:text="@string/tv_key_vc"
        android:textColor="@color/text_body_default"
        android:textSize="20dip" />

    <TextView
        android:id="@+id/tv_auth_key"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="150dip"
        android:background="@android:color/transparent"
        android:ems="10"
        android:gravity="center"
        android:imeOptions="actionGo"
        android:inputType="text"
        android:maxLength="8"
        android:nextFocusDown="@+id/btn_login"
        android:singleLine="true"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="@color/text_body_default"
        android:textSize="65dip" />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="1dip"
        android:layout_gravity="center_horizontal"
        android:layout_marginLeft="50dip"
        android:layout_marginRight="50dip"
        android:layout_marginTop="230dip"
        android:background="@color/line_text_under"
        android:visibility="invisible" />

    <com.beardedhen.androidbootstrap.BootstrapProgressBar
        android:id="@+id/progressBar"
        android:layout_width="fill_parent"
        android:layout_height="12dip"

```

```

        android:layout_gravity="center_hor izontal"
        android:layout_marginTop="240dip"
        app:animated="true"
        app:bootstrapBrand="warning"
        app:bootstrapProgress="100"
        app:striped="true" />

```

```

<LinearLayout

```

```

    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="260dip"
    android:orientation="hor izontal">

```

```

    <TextView

```

```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

```

```

    <ImageView

```

```

        android:layout_width="15dip"
        android:layout_height="15dip"
        android:layout_gravity="center_vert ical|right"
        android:background="@drawable/ico_countdown" />

```

```

    <TextView

```

```

        android:id="@+id/tv_remainTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="r ight|center_vert ical"
        android:paddingLeft="10dip"
        android:textColor="@color/text_body_guide"
        android:textSize="17dip" />

```

```

</LinearLayout>

```

```

<TextView

```

```

    android:id="@+id/tv_system_nm"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_hor izontal"
    android:layout_marginTop="315dip"
    android:text=""
    android:textColor="@color/text_body_default"
    android:textSize="18dip" />

```

```

<TextView

```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_hor izontal"
    android:layout_marginTop="380dip"
    android:text="@string/tv_key_msg_1"
    android:textColor="@color/text_body_guide"
    android:textSize="18dip" />

```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="405dip"
    android:text="@string/tv_key_msg_2"
    android:textColor="@color/text_body_guide"
    android:textSize="18dip" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="430dip"
    android:text="@string/tv_key_msg_3"
    android:textColor="@color/text_body_guide"
    android:textSize="18dip" />

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="490dip"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btn_update"
        android:layout_width="fill_parent"
        android:layout_height="@dimen/btn_height_default"
        android:layout_weight="1"
        android:background="@drawable/btn_default_drawable"
        android:text="@string/btn_upd_del"
        android:textColor="@color/white"
        android:textSize="20dip" />

    <TextView
        android:layout_width="6dip"
        android:layout_height="1dip"
        android:layout_gravity="center_horizontal"
        android:background="@android:color/transparent" />

    <Button
        android:id="@+id/btn_reset"
        android:layout_width="fill_parent"
        android:layout_height="@dimen/btn_height_default"
        android:layout_weight="1"
        android:background="@drawable/btn_default_drawable"
        android:enabled="false"
        android:text="@string/btn_reset"
        android:textColor="@color/white"
        android:textSize="20dip" />

</LinearLayout>
```

```

        </FrameLayout>

    </ScrollView>

</FrameLayout>

```

프로그램 예시)

```

package com.baro.key.info;

import android.Manifest;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.Vibrator;
import android.support.v4.app.ActivityCompat;
import android.telephony.TelephonyManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.TextView;

import com.baro.common.base.BaseActivity;
import com.baro.common.base.BaseInterface;
import com.baro.common.setting.IdentifyACT;
import com.baro.common.setting.SettingACT;
import com.baro.common.db.BaroDAO;
import com.baro.common.db.BaroDTO;
import com.baro.common.util.Util;
import com.baro.pam.R;
import com.barokey.barokey;
import com.beardedhen.androidbootstrap.BootstrapProgressBar;

import java.util.Date;

@TargetApi(Build.VERSION_CODES.O)
public class OTPCreateACT extends BaseActivity implements BaseInterface, OnClickListener {
    private Button btn_setting, btn_share, btn_close, btn_reset, btn_update;
    private TextView tv_auth_key, tv_system_nm, tv_remainTime;
    private BootstrapProgressBar progressBar;
    private String intent_reg_dt = "", intent_system_nm = "", intent_login_id = "",
    intent_cycle_time = "";

    private String PhoneNumber = "", SerialNumber = "", AndroidID = "", MacAddr = "";

```

```

private long    createdMillis, remainingSec;

private static final int MESSAGE_REFRESH_REMAINING_SECOND = 101;
private static final int SENDMESSAGE_INTERVAL = 250;

private String[] permission_list = { Manifest.permission.READ_PHONE_STATE,
Manifest.permission.READ_PHONE_NUMBERS,
Manifest.permission.CALL_PHONE,
Manifest.permission.INTERNET,
Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE };

@Override
public void onCreate(Bundle savedInstanceState) {
    try {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.act_otpccreate);

        dao = new BaroDAO(this);
        dto = new BaroDTO();

        dto = dao.selectSET();
        PhoneNumber = dto.getPhone_no();

        if ("".equals(PhoneNumber)) {
            checkPermission();
        }
        drawView();
        getIntentData();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
    }
}

@Override
public void onPause() {
    super.onPause();

    if (null != m_handlerProc) {
        m_handlerProc.removeMessages(MESSAGE_REFRESH_REMAINING_SECOND);
    }
}

@Override
public void onResume() {
    super.onResume();

    if (null != m_handlerProc) {
        m_handlerProc.sendMessageDelayed(MESSAGE_REFRESH_REMAINING_SECOND,
SENDMESSAGE_INTERVAL);
    }
}

@SuppressLint("HardwareIds", "MissingPermission")

```



```

@Override
public void drawView() {
    try {
        vibrate = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);

        findViewById(R.id.body_frame).setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
                imm.hideSoftInputFromWindow(v.getWindowToken(), 0);
            }
        });
        tv_system_nm = (TextView) findViewById(R.id.tv_system_nm);
        tv_system_nm.setOnClickListener(this);

        tv_auth_key = (TextView) findViewById(R.id.tv_auth_key);
        tv_auth_key.setFocusable(true);
        tv_auth_key.setClickable(false);

        progressBar = (BootstrapProgressBar) findViewById(R.id.progressBar);

        tv_remainTime = (TextView) findViewById(R.id.tv_remainTime);

        btn_setting = (Button) findViewById(R.id.btn_setting);
        btn_setting.setOnClickListener(this);

        btn_share = (Button) findViewById(R.id.btn_share);
        btn_share.setOnClickListener(this);

        ((Button) findViewById(R.id.btn_go_back)).setOnClickListener(this);

        btn_close = (Button) findViewById(R.id.btn_close);
        btn_close.setOnClickListener(this);

        btn_update = (Button) findViewById(R.id.btn_update);
        btn_update.setOnClickListener(this);

        btn_reset = (Button) findViewById(R.id.btn_reset);
        btn_reset.setOnClickListener(this);

        if ("".equals(PhoneNumber)) {
            TelephonyManager getSystemService = (TelephonyManager)
getSystemService(Context.TELEPHONY_SERVICE);
            assert getSystemService != null;
            PhoneNumber = getSystemService.getLine1Number();
        }
    } catch (SecurityException e) {
        e.printStackTrace();
        //Util.MsgToast(OTPCreateACT.this, e.toString() + "Wn" +
getString(R.string.msg_phone_permissions), 0);
        Util.MsgToast(OTPCreateACT.this, getString(R.string.msg_phone_permissions), 0);
        try { Thread.sleep(3000); } catch (Exception ee) {}
    } catch (Exception e) {

```

```

        e.printStackTrace();
        Intent intent = new Intent(OTPCreateACT.this, IdentifyACT.class);
        startActivity(intent);
        finish();
    } finally {
    }
}

public void getIntentData() {
    try {
        Intent intent = getIntent();
        getDefaultIntent(intent);

        if (intent.getStringExtra("reg_dt") != null) {
            intent_reg_dt = intent.getStringExtra("reg_dt").trim();
        }
        if (intent.getStringExtra("system_nm") != null) {
            intent_system_nm = intent.getStringExtra("system_nm");
        }
        if (intent.getStringExtra("login_id") != null) {
            intent_login_id = intent.getStringExtra("login_id").trim();
        }
        if (intent.getStringExtra("cycle_time") != null) {
            intent_cycle_time = intent.getStringExtra("cycle_time").trim();
        }
        if ("".equals(intent_system_nm.trim())) {
            tv_system_nm.setText("[" + intent_login_id + " ]");
        } else if (!"".equals(intent_system_nm) && (!"".equals(intent_login_id))) {
            tv_system_nm.setText("[" + intent_system_nm + " / " + intent_login_id + " ]");
        }
        if (!"".equals(intent_login_id) && !"".equals(PhoneNumber) &&
(!"".equals(intent_cycle_time))) {
            onAuthKey();
        } else {
            finish();
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
    }
}

@Override
public void onClick(View v) {
    try {
        switch (v.getId()) {
            case R.id.btn_setting: // Setting
                Intent intent = new Intent(this, SettingACT.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_DOCUMENT);
                startActivity(intent);
                //finish();
                break;
        }
    }
}

```

```

        case R.id.btn_share:
            intent = new Intent(Intent.ACTION_SEND);
            intent.addCategory(Intent.CATEGORY_DEFAULT);
            intent.putExtra(Intent.EXTRA_TEXT , getString(R.string.app_share));
            intent.putExtra(Intent.EXTRA_TITLE, getString(R.string.app_name ));
            intent.setType("text/plain");
            startActivity(Intent.createChooser(intent, getString(R.string.share_text)));
            //finish();
            break;

        case R.id.btn_go_back: // go back
            finish();
            break;

        case R.id.btn_close: // Close
            moveTaskToBack(true);
            finish();
            android.os.Process.killProcess(android.os.Process.myPid());
            break;

        case R.id.btn_update: // Update
            intent = new Intent(OTPCreateACT.this, OTPUpdateACT.class);
            intent.putExtra("reg_dt" , intent_reg_dt );
            intent.putExtra("system_nm" , intent_system_nm );
            intent.putExtra("login_id" , intent_login_id );
            intent.putExtra("cycle_time", intent_cycle_time);
            startActivity(intent);
            finish();
            break;

        case R.id.btn_reset: // Reset
            if (!"".equals(intent_login_id)      &&      !"".equals(PhoneNumber)      &&
(!"".equals(intent_cycle_time))) {
                onAuthKey();
            } else {
                finish();
            }
            break;
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
}
}

public void onAuthKey() {
    try {
        tv_auth_key.setText("");
        createdMillis = estimateCreatedMillis(intent_cycle_time);

        String otp_org = barokey.generateKEYL(intent_login_id, PhoneNumber, intent_cycle_time);
        String otp_new = otp_org.substring(0, 3) + " " + otp_org.substring(3, 6);
        tv_auth_key.setText(otp_new);
    }
}

```

```

        m_handlerProc.sendMessageDelayed(MESSAGE_REFRESH_REMAINING_SECOND,
SENDMESSAGE_INTERVAL);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
    }
}

private final Handler m_handlerProc = new Handler() {
    @Override
    public void handleMessage(Message message) {
        switch (message.what) {
            case MESSAGE_REFRESH_REMAINING_SECOND:
                try {
                    long cycleMillis = (Long.parseLong(intent_cycle_time) * 1000L);
                    long remainingMillis = estimateRemainingMillis(intent_cycle_time,
createdMillis);
                    long remainingSecond = remainingMillis != 0 ? (remainingMillis / 1000L) :
0;

                    if (0 < remainingMillis) {
                        m_handlerProc.sendMessageDelayed(MESSAGE_REFRESH_REMAINING_SECOND,
SENDMESSAGE_INTERVAL);
                        btn_reset.setEnabled(false);
                    } else {
                        Thread.sleep(500);
                        m_handlerProc.removeMessages(MESSAGE_REFRESH_REMAINING_SECOND);
                        btn_reset.setEnabled(true);
                    }
                    tv_remainTime.setText(remainingSecond + " " + " " +
getString(R.string.remain_time_suffix));

                    if (0 != cycleMillis) {
                        progressBar.setProgress(((int) (((float) remainingMillis / (float)
cycleMillis) * 100.0F)));
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                } finally {
                }
                break;
            }
        }
    }
};

public long estimateCreatedMillis(String cycleSecondString) {
    long remainingMillis = (barokey.getRemainingTime(cycleSecondString) * 1000L) - 200;
    long cycleMillis = (Long.parseLong(cycleSecondString) * 1000L);
    long currentMillis = (new Date()).getTime();
    long elapsedMillis = cycleMillis - remainingMillis;
    long createdMillis = currentMillis - elapsedMillis;
}

```

```

        return createdMillis;
    }

    public long estimateRemainingMillis(String cycleSecondString, long createdTime) {
        long cycleMillis = (Long.parseLong(cycleSecondString) * 1000L);
        long currentMillis = (new Date()).getTime();
        long elapsedMillis = currentMillis - createdTime;

        long remainingMillis = barokey.getRemainingTime(cycleSecondString) * 1000L;
        remainingMillis = cycleMillis > elapsedMillis ? remainingMillis : 0;
        remainingMillis = remainingMillis >= cycleMillis ? 0 : remainingMillis;

        return remainingMillis;
    }

    public void checkPermission() {
        if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M)
            return;

        for(String permission : permission_list) {
            int permssionCheck = checkCallingOrSelfPermission(permission);

            if (permssionCheck == PackageManager.PERMISSION_DENIED) {
                ActivityCompat.requestPermissions(this, permission_list, 0);
            }
        }
    }

    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
        if (requestCode == 0) {
            for(int ii = 0; ii < grantResults.length; ii++) {
                if (grantResults[ii] != PackageManager.PERMISSION_GRANTED) {
                    if (!"android.permission.READ_PHONE_NUMBERS".equals(permissions[ii])) {
                        Util.MsgToast(OTPCreateACT.this, getString(R.string.msg_security_set) + "("
+ permissions[ii] + ")", 0);
                    }
                }
                //finish();
            }
        }
    }
}

```

1.4 아이폰인 경우

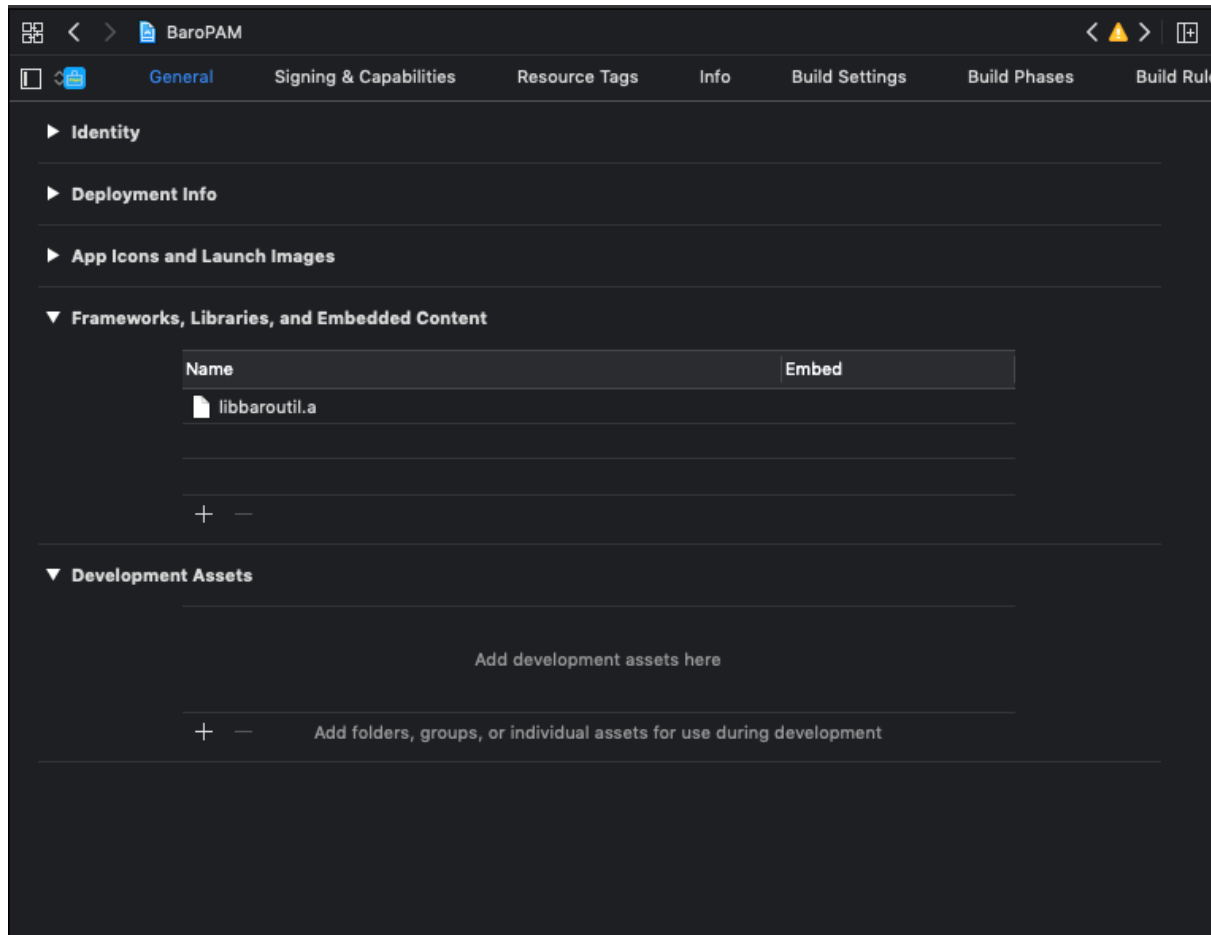
1) 인증키 생성기 부분

애플리케이션에 로그인 시 비밀번호란에 입력할 **일회용 인증키**를 생성하는 API는 "libbaroutil.a"로 제공되며, 이 파일에는 barokey, barocrypt, base64 관련 라이브러리를 포함하는 NSObject Interface용 라이브러리 파일이다.

라이브러리 파일은 두가지 종류로 제공한다. XCode의 iPhone simulator 용과, iPhone용 두가지를 필요에 따라 "libbaroutil.a"로 변경하여 사용한다.

- libbaroutil.a.iphone : iPhone용
- libbaroutil.a.simul : iPhone simulator 용

이 파일은 아래와 같이 XCode의 프로젝트 설정시에 등록하여 사용한다.



BaroKEY 관련된 API는 다음과 같다. 함수는 C 함수 Interface로 되어 있어, 입력값의 자료형은 C 함수 스타일로 표기한다. 사용 예의 소스는 iOS swift 5.0 이상으로 작성된 코드이다.

generateKEYL 함수

애플리케이션에 로그인/인증 시 사용하는 **일회용 인증키**를 생성하는 함수이다.

입력변수	const char *login_id	로그인 화면의 로그인-ID 항목에 입력한 ID를 설정해야 한다.
	const char *phone_no	사용자 휴대폰번호다. 안드로이드용 앱에서와는 달리, iOS에서 사용자의 휴대폰번호를 얻어 오지 않고 서버의 인증모듈에서 사용할 사용자의 휴대폰번호를 앱에서 직접 등록하여 관리하며, 등록된 휴대폰번호를 선택하여 사용한다.
	const char *cycle_time	개인별로 지정한 일회용 인증키 의 생성 주기(3~60초)와 반드시 일치해야 한다. 만약, 개인별로 지정한 일회용 인증키 의 생성 주기가 다른 경우 일회용 인증키 가 다르게 생성될 수 있다.
	const char *key_method	일회용 인증키 의 생성 방식(app1, app256, app384, app512: 앱)으로 "app512"를 설정한다.

리턴 값	일회용 인 증 키	생성한 일회용 인증키 를 반환한다.
------	-----------	----------------------------

swift 5.0 이상에서의 사용예시)

```
private func makeOtpInfo() {
    let loginid = _otp?.LOGIN_ID ?? "mc529@hanmail.net"
    let tel = _otp?.PHONE_NO ?? "01027714076"
    let time = (_otp?.CYCLE_TIME ?? "30")!
    let otpnum = generateKEYL(loginid, tel, time, "app512")
    _otpInfo.text = "[ W(_otp?.SYSTEM_NM ?? "")/W(_otp?.LOGIN_ID ?? "") ]"
    let otpnumStr = String(cString: otpnum!)
    let start = otpnumStr.index(otpnumStr.startIndex, offsetBy: 0)
    let end = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
    let start2 = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
    let end2 = otpnumStr.index(otpnumStr.startIndex, offsetBy: 6)

    _tfOTP.text = otpnumStr[start..

```

화면 예시)



화면 Layout 예시)

Storyboard를 의미한다. 각 파라미터에 대한 의미는 developer.apple.com을 참고한다.

```
<!--Create View Controller-->
<scene sceneID="xJv-bd-Ejb">
  <objects>
    <viewController storyboardIdentifier="CreateOTP" id="BPh-TI-Gd5"
      customClass="OTPCreateViewController" customModule="BaroPAM" customModuleProvider="target"
      sceneMemberID="viewController">
      <layoutGuides>
        <viewControllerLayoutGuide type="top" id="TF9-Et-51n"/>
        <viewControllerLayoutGuide type="bottom" id="rXs-zr-mnc"/>
      </layoutGuides>
      <view key="view" contentMode="scaleToFill" id="DbI-ks-whW">
        <rect key="frame" x="0.0" y="0.0" width="375" height="812"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
        <subviews>
          <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="scaleToFill"
            fixedFrame="YES" text="일회용 인증키" textAlignment="center"
            translatesAutoresizingMaskIntoConstraints="NO" id="00T-0a-9fL">
            <rect key="frame" x="0.0" y="125" width="375" height="40"/>
            <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
            <color key="textColor" white="0.0" alpha="1" colorSpace="calibratedWhite"/>
            <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
              family="SpoqaHanSans" pointSize="17"/>
            <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
          </textView>
          <textField opaque="NO" clipsSubviews="YES" contentMode="scaleToFill" fixedFrame="YES"
```



```

contentHorizontalAlignment="left"      contentVerticalAlignment="center"      text="12345678"
textAlignment="center"      minimumFontSize="17"      translatesAutoresizingMaskIntoConstraints="NO"
id="y9V-i0-Xec">
  <rect key="frame" x="19" y="204" width="336" height="52"/>
  <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
  <color key="backgroundColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
  <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="50"/>
  <textInputTraits key="textInputTraits"/>
</textField>
<button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect"
lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="wn5-JQ-qp2">
  <rect key="frame" x="23" y="683" width="160" height="43"/>
  <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxX="YES"
flexibleMinY="YES"/>
  <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
  <state key="normal" title="Update/Delete">
    <color key="titleColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
  </state>
  <connections>
    <action selector="onEdit:" destination="BPh-Tl-Gd5" eventType="touchUpInside"
id="Lq0-gt-fdh"/>
    <action selector="onOk:" destination="BYZ-38-t0r" eventType="touchUpInside"
id="ya1-b8-A5Q"/>
  </connections>
</button>
<button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect"
lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="phw-d7-Zsz">
  <rect key="frame" x="199" y="683" width="160" height="43"/>
  <autoresizingMask key="autoresizingMask" flexibleMinX="YES" widthSizable="YES"
flexibleMinY="YES"/>
  <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
  <state key="normal" title="Reset">
    <color key="titleColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
  </state>
  <connections>
    <action selector="onReset:" destination="BPh-Tl-Gd5" eventType="touchUpInside"
id="2is-dP-y2P"/>
  </connections>
</button>
<view contentMode="scaleToFill" fixedFrame="YES"
translatesAutoresizingMaskIntoConstraints="NO" id="KTy-6U-0mm">
  <rect key="frame" x="0.0" y="0.0" width="375" height="70"/>
  <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
  <subviews>
    <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center"
lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="5ZR-gQ-4P5">
      <rect key="frame" x="283" y="34" width="31" height="31"/>

```

```

        <autoresizingMask key="autoresizingMask" flexibleMinX="YES" flexibleMaxY="YES"/>
        <inset key="imageEdgeInsets" minX="3" minY="3" maxX="3" maxY="3"/>
        <state key="normal" image="btn_share.png"/>
        <connections>
            <action selector="onShare:" destination="BPh-TI-Gd5" eventType="touchUpInside"
id="rVd-IW-j3A"/>
        </connections>
    </button>
    <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center"
lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="uD6-U2-2w3">
        <rect key="frame" x="322" y="34" width="33" height="32"/>
        <autoresizingMask key="autoresizingMask" flexibleMinX="YES" flexibleMaxY="YES"/>
        <inset key="imageEdgeInsets" minX="3" minY="3" maxX="3" maxY="3"/>
        <state key="normal" image="btn_setting.png"/>
        <connections>
            <action selector="onSetting:" destination="BPh-TI-Gd5"
eventType="touchUpInside" id="Qhc-bj-CHe"/>
        </connections>
    </button>
    <imageView userInteractionEnabled="NO" contentMode="scaleAspectFit"
horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
image="btn_prev.png" translatesAutoresizingMaskIntoConstraints="NO" id="cZQ-Jb-luv">
        <rect key="frame" x="19" y="35" width="31" height="31"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" heightSizable="YES"/>
    </imageView>
    <imageView userInteractionEnabled="NO" contentMode="scaleAspectFit"
horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
image="logo_barootp.png" translatesAutoresizingMaskIntoConstraints="NO" id="vWu-o6-6az">
        <rect key="frame" x="115" y="38" width="145" height="25"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
    </imageView>
</subviews>
    <color key="backgroundColor" red="0.10588235294117647" green="0.56470588235294117"
blue="1" alpha="1" colorSpace="calibratedRGB"/>
</view>
    <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="scaleToFill"
fixedFrame="YES" editable="NO" text="유효시간 내에 인증키를 입력하세요. 시간을 초과한 경우 Reset
버튼을 클릭하여 인증키를 재생성 하세요." textAlignment="natural" selectable="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="s4z-fe-3rj">
        <rect key="frame" x="49" y="585" width="276" height="112"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMinY="YES"/>
        <color key="textColor" red="0.33333333333333331" green="0.33333333333333331"
blue="0.33333333333333331" alpha="1" colorSpace="calibratedRGB"/>
        <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
        <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
    </textView>
    <progressView opaque="NO" contentMode="scaleToFill" verticalHuggingPriority="750"
fixedFrame="YES" progress="0.5" translatesAutoresizingMaskIntoConstraints="NO" id="eFk-qb-ugh">
        <rect key="frame" x="52" y="274" width="270" height="2"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES"/>
    </progressView>

```

```

        <imageView                userInteractionEnabled="NO"                contentMode="scaleToFill"
horizontalHuggingPriority="251"                verticalHuggingPriority="251"                fixedFrame="YES"
image="ico_countdown.png" translatesAutoresizingMaskIntoConstraints="NO" id="UC7-dN-216">
        <rect key="frame" x="250" y="284" width="15" height="15"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
    </imageView>
    <label                opaque="NO"                userInteractionEnabled="NO"                contentMode="left"
horizontalHuggingPriority="251"                verticalHuggingPriority="251"                fixedFrame="YES"                text="0"
textAlignment="natural"                lineBreakMode="tailTruncation"                baselineAdjustment="alignBaselines"
adjustsFontSizeToFit="NO" translatesAutoresizingMaskIntoConstraints="NO" id="c11-3a-nD8">
        <rect key="frame" x="270" y="281" width="52" height="21"/>
        <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
        <fontDescription                key="fontDescription"                name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
        <color                key="textColor"                red="0.3333333333333331"                green="0.3333333333333331"
blue="0.3333333333333331" alpha="1" colorSpace="calibratedRGB"/>
        <nil key="highlightedColor"/>
    </label>
    <label                opaque="NO"                userInteractionEnabled="NO"                contentMode="left"
horizontalHuggingPriority="251"                verticalHuggingPriority="251"                fixedFrame="YES"
text="[emplus/david.kscho@empluses.com]"                textAlignment="center"                lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines"                adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="FZ0-er-yGs">
        <rect key="frame" x="23" y="318" width="332" height="30"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
        <fontDescription                key="fontDescription"                name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
        <nil key="highlightedColor"/>
    </label>
</subviews>
<color key="backgroundColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
</view>
<connections>
    <outlet property="_backView" destination="cZQ-Jb-luv" id="hti-Le-Rra"/>
    <outlet property="_btnReset" destination="phw-d7-Zsz" id="hVD-Q9-8Xq"/>
    <outlet property="_btnUpdate" destination="wn5-JQ-qp2" id="o6G-9e-g0S"/>
    <outlet property="_otpInfo" destination="FZ0-er-yGs" id="d1r-2i-KX2"/>
    <outlet property="_progress" destination="eFk-qb-ugh" id="csW-nT-cyw"/>
    <outlet property="_remainTime" destination="c11-3a-nD8" id="b6H-g5-IXA"/>
    <outlet property="_tfOTP" destination="y9V-i0-Xec" id="loX-6A-goi"/>
</connections>
</viewController>
<placeholder placeholderIdentifier="IBFirstResponder" id="GRs-8z-hxZ" userLabel="First
Responder" sceneMemberID="firstResponder"/>
</objects>
<point key="canvasLocation" x="2948" y="440"/>
</scene>

```

프로그램 예시)

```

import UIKit

class OTPCreateViewController : UIViewController {

```

```

@IBOutlet weak var _progress: UIProgressView!
@IBOutlet weak var _remainTime: UILabel!
@IBOutlet weak var _backView: UIImageView!
@IBOutlet weak var _otpInfo: UILabel!
@IBOutlet weak var _tfOTP: UITextField!
@IBOutlet weak var _btnUpdate: UIButton!
@IBOutlet weak var _btnReset: UIButton!

@IBAction func onClose(_ sender: Any) {
    exit(0)
}

var _timer: Timer?
var _otp: OTPEntity? = nil

override func viewDidLoad() {
    super.viewDidLoad()
    //changeBackground()
    initControls()
    makeTappedView()
    makeOtpInfo()
}

override func viewWillAppear(_ animated: Bool) {
    super.viewWillAppear(animated)
    if (_otp?.IS_DELETE == 1) {
        _otp?.IS_DELETE = 0
        dismiss(animated: false, completion: nil)
    }
}

override func viewDidAppear(_ animated: Bool) {
    super.viewDidAppear(animated)
}

private func initControls() {
    _btnUpdate.backgroundColor = UIColorFromHex(rgbValue: 0x1B90FF)
    _btnReset.backgroundColor = UIColorFromHex(rgbValue: 0x1B90FF)

    super.modalPresentationStyle = .fullScreen
}

private func changeBackground() {
    // MAIN View Background Change
    let background = UIImageView(frame: UIScreen.main.bounds)
    background.image = UIImage(named: "bg_sub.png")
    self.view.insertSubview(background, at: 0)
}

private func makeTappedView() {
    let tap = UITapGestureRecognizer(target: self, action:
#selector(OTPCreateViewController.backTapped))
    _backView.isUserInteractionEnabled = true
    _backView.addGestureRecognizer(tap)
}

```

```

}

private func makeOtpInfo() {
    let loginid = _otp?.LOGIN_ID ?? "mc529@hanmail.net"
    let tel = _otp?.PHONE_NO ?? "01027714076"
    let time = (_otp?.CYCLE_TIME ?? "30")!
    let otpnum = generateKEYL(loginid, tel, time, "app512")
    _otpInfo.text = "[ W(_otp?.SYSTEM_NM ?? "")/W(_otp?.LOGIN_ID ?? "") ]"
    let otpnumStr = String(cString: otpnum!)
    let start = otpnumStr.index(otpnumStr.startIndex, offsetBy: 0)
    let end = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
    let start2 = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
    let end2 = otpnumStr.index(otpnumStr.startIndex, offsetBy: 6)

    _tfOTP.text = otpnumStr[start..

```

```

        print("————> W(otp.REG_DT), W(otp.LOGIN_ID), W(otp.SYSTEM_NM), W(otp.CYCLE_TIME)")
    }

    @IBAction func onSetting(_ sender: Any) {
        switchScreen("Settings")
    }

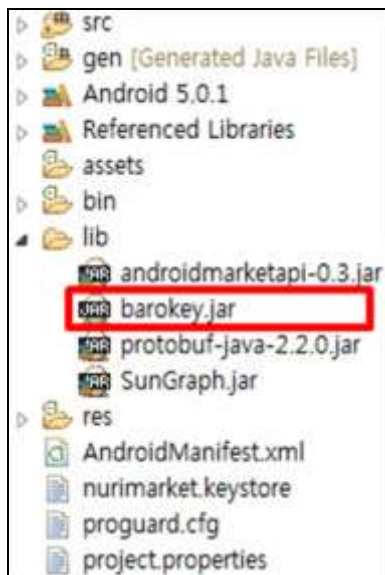
    @IBAction func onShare(_ sender: Any) {
    }
}

```

2. 서버 로그인(Secure key 이용)

2.1 안드로이드 인 경우

"BaroPAM" 앱에서 서버 로그인 시 "Verification code"에 입력할 **일회용 인증키**를 생성하는 API는 "barokey.jar"으로 제공되며, Eclipse을 사용하는 경우 libs 디렉토리에 "barokey.jar"를 위치시켜야 한다.



"Verification code"인 **2차 인증키(일회용 인증키)**를 생성하는 프로그램에 다음과 같은 코드를 삽입하면 된다.

```

...
import com.barokey.barokey;
...
String auth_key = barokey.generateKEYP(secure_key, cycle_time);
...

```

파라미터	설명	비고
secure_key	벤더에 요청하여 제공 받은 키로 서버별, 계정별로 부여할 수 있으며, 반드시 BaroPAM 서버 환경 설정 시 지정한 키를 설정.	
cycle_time	BaroPAM 서버 환경설정 시 지정한 일회용 인증키 의 생성 주기(3~60초)를 설정. 만약, BaroPAM 서버 환경설정 값과 다른 경우 "Verification code"인 일회용 인증키 가 다르게 생성될 수 있다.	

화면 예시)



화면 Layout 예시)

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/bg_body_default"
    android:orientation="vertical">

    <include
        android:id="@+id/inc_header"
        layout="@layout/inc_header"
        android:layout_width="fill_parent"
        android:layout_height="@dimen/head_height" />

    <ScrollView
```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="@dimen/body_margin_right_default"
        android:layout_marginRight="@dimen/body_margin_right_default"
        android:layout_marginTop="@dimen/head_height">

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/body_frame"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="81dip"
        android:padding="10dp"
        android:text="@string/tv_key_vc"
        android:textColor="@color/text_body_default"
        android:textSize="20dip" />

    <TextView
        android:id="@+id/tv_auth_key"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="150dip"
        android:background="@android:color/transparent"
        android:ems="10"
        android:gravity="center"
        android:imeOptions="actionGo"
        android:inputType="text"
        android:maxLength="8"
        android:nextFocusDown="@+id/btn_login"
        android:singleLine="true"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="@color/text_body_default"
        android:textSize="65dip" />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="1dip"
        android:layout_gravity="center_horizontal"
        android:layout_marginLeft="50dip"
        android:layout_marginRight="50dip"
        android:layout_marginTop="230dip"
        android:background="@color/line_text_under"
        android:visibility="invisible" />

    <com.bear dedhen.androidbootstrap.BootstrapProgressBar
        android:id="@+id/progressBar"
        android:layout_width="fill_parent"

```



```

        android:layout_height="12dip"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="240dip"
        app:animated="true"
        app:bootstrapBrand="warning"
        app:bootstrapProgress="100"
        app:striped="true" />

```

```

<LinearLayout

```

```

    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="260dip"
    android:orientation="horizontal">

```

```

    <TextView

```

```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

```

```

    <ImageView

```

```

        android:layout_width="15dip"
        android:layout_height="15dip"
        android:layout_gravity="center_vertical|right"
        android:background="@drawable/ico_countdown" />

```

```

    <TextView

```

```

        android:id="@+id/tv_remainTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="right|center_vertical"
        android:paddingLeft="10dip"
        android:textColor="@color/text_body_guide"
        android:textSize="17dip" />

```

```

</LinearLayout>

```

```

<TextView

```

```

    android:id="@+id/tv_server_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="315dip"
    android:text=""
    android:textColor="@color/text_body_default"
    android:textSize="18dip" />

```

```

<TextView

```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="380dip"
    android:text="@string/tv_key_msg_1"
    android:textColor="@color/text_body_guide"

```

```
        android:textSize="18dip" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="405dip"
        android:text="@string/tv_key_msg_2"
        android:textColor="@color/text_body_guide"
        android:textSize="18dip" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="430dip"
        android:text="@string/tv_key_msg_3"
        android:textColor="@color/text_body_guide"
        android:textSize="18dip" />

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="490dip"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btn_Modify"
            android:layout_width="fill_parent"
            android:layout_height="@dimen/btn_height_default"
            android:layout_weight="1"
            android:background="@drawable/btn_default_drawable"
            android:text="@string/btn_upd_del"
            android:textColor="@color/white"
            android:textSize="20dip" />

        <TextView
            android:layout_width="6dip"
            android:layout_height="1dip"
            android:layout_gravity="center_horizontal"
            android:background="@android:color/transparent" />

        <Button
            android:id="@+id/btn_reset"
            android:layout_width="fill_parent"
            android:layout_height="@dimen/btn_height_default"
            android:layout_weight="1"
            android:background="@drawable/btn_default_drawable"
            android:enabled="false"
            android:text="@string/btn_reset"
            android:textColor="@color/white"
            android:textSize="20dip" />
```

```

        </LinearLayout>

        </FrameLayout>

    </ScrollView>

</FrameLayout>

```

프로그램 예시)

```

package com.baro.pam.info;

import com.baro.common.base.BaseActivity;
import com.baro.common.base.BaseInterface;
import com.baro.common.setting.SettingACT;
import com.baro.common.db.BaroDAO;
import com.baro.common.db.BaroDTO;
import com.baro.common.util.Util;
import com.baro.pam.R;

import com.barokey.barokey;
import com.beardedhen.androidbootstrap.BootstrapProgressBar;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Message;
import android.os.Vibrator;
import android.os.Handler;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.TextView;

import java.util.Date;

public class PAMCreateACT extends BaseActivity implements BaseInterface, OnClickListener {
    private Button btn_setting, btn_share, btn_close, btn_update, btn_reset;
    private TextView tv_auth_key, tv_server_name, tv_remainTime;
    private BootstrapProgressBar progressBar;
    private String intent_reg_dt = "", intent_server_name = "", intent_secure_key = "",
    intent_cycle_time = "", app_code = "kr";
    Intent intent;

    private long createdMillis, remainingSec;

    private static final int MESSAGE_REFRESH_REMAINING_SECOND = 101;
    private static final int SENDMESSAGE_INTERVAL = 250;

    @Override
    public void onCreate(Bundle savedInstanceState) {

```

```

        try {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.act_pamcreate);

            BaroDAO dao = new BaroDAO(this);
            BaroDTO dto = new BaroDTO();

            dto = dao.selectSET();
            app_code = dto.getApp_code();

            if (app_code.length() != 2) {
                app_code = "kr";
            }
            drawView();
            getIntentData();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
        }
    }

    @Override
    public void onPause() {
        super.onPause();

        if (null != m_handlerProc) {
            m_handlerProc.removeMessages(MESSAGE_REFRESH_REMAINING_SECOND);
        }
    }

    @Override
    public void onResume() {
        super.onResume();

        if (null != m_handlerProc) {
            m_handlerProc.sendMessageDelayed(MESSAGE_REFRESH_REMAINING_SECOND,
SENDMESSAGE_INTERVAL);
        }
    }

    @Override
    public void drawView() {
        try {
            vibe = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);

            findViewById(R.id.body_frame).setOnClickListener(new OnClickListener() {
                public void onClick(View v) {
                    InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
                    imm.hideSoftInputFromWindow(v.getWindowToken(), 0);
                }
            });
            tv_server_name = (TextView) findViewById(R.id.tv_server_name);

```

```

        tv_server_name.setOnClickListener(this);

        tv_auth_key = (TextView) findViewById(R.id.tv_auth_key);
        tv_auth_key.setFocusable(true);
        tv_auth_key.setClickable(false);

        progressBar = (BootstrapProgressBar) findViewById(R.id.progressBar);

        tv_remainTime = (TextView) findViewById(R.id.tv_remainTime);

        btn_setting = (Button) findViewById(R.id.btn_setting);
        btn_setting.setOnClickListener(this);

        btn_share = (Button) findViewById(R.id.btn_share);
        btn_share.setOnClickListener(this);

        ((Button) findViewById(R.id.btn_go_back)).setOnClickListener(this);

        btn_close = (Button) findViewById(R.id.btn_close);
        btn_close.setOnClickListener(this);

        btn_update = (Button) findViewById(R.id.btn_update);
        btn_update.setOnClickListener(this);

        btn_reset = (Button) findViewById(R.id.btn_reset);
        btn_reset.setOnClickListener(this);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
    }
}

public void getIntentData() {
    try {
        Intent intent = getIntent();
        getDefaultIntent(intent);

        if (intent.getStringExtra("reg_dt") != null) {
            intent_reg_dt = intent.getStringExtra("reg_dt").trim();
        }
        if (intent.getStringExtra("server_name") != null) {
            intent_server_name = intent.getStringExtra("server_name").trim();
            tv_server_name.setText("[ " + intent_server_name + " ]");
        } else {
            tv_server_name.setText("");
        }
        if (intent.getStringExtra("secure_key") != null) {
            intent_secure_key = intent.getStringExtra("secure_key").trim();
        }
        if (intent.getStringExtra("cycle_time") != null) {
            intent_cycle_time = intent.getStringExtra("cycle_time").trim();
        }
        if (!"".equals(intent_secure_key) && (!"".equals(intent_cycle_time))) {

```

```

        onAuthKey();
    } else {
        finish();
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
}
}

@Override
public void onClick(View v) {
    try {
        switch (v.getId()) {
            case R.id.btn_setting: // Setting
                Intent intent = new Intent(this, SettingACT.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_DOCUMENT);
                startActivity(intent);
                //finish();
                break;

            case R.id.btn_share:
                intent = new Intent(Intent.ACTION_SEND);
                intent.addCategory(Intent.CATEGORY_DEFAULT);
                intent.putExtra(Intent.EXTRA_TEXT , getString(R.string.app_share));
                intent.putExtra(Intent.EXTRA_TITLE, getString(R.string.app_name ));
                intent.setType("text/plain");
                startActivity(Intent.createChooser(intent,
getString(R.string.share_text)));
                //finish();
                break;

            case R.id.btn_go_back: // go back
                finish();
                break;

            case R.id.btn_close: // Close
                moveTaskToBack(true);
                finish();
                android.os.Process.killProcess(android.os.Process.myPid());
                break;

            case R.id.btn_update: // Update/Delete
                intent = new Intent(PAMCreateACT.this, PAMUpdateACT.class);
                intent.putExtra("reg_dt" , intent_reg_dt );
                intent.putExtra("server_name", intent_server_name);
                intent.putExtra("secure_key" , intent_secure_key );
                intent.putExtra("cycle_time" , intent_cycle_time );
                startActivity(intent);
                finish();
                break;

            case R.id.btn_reset: // Reset

```

```

        if (!"".equals(intent_secure_key) && (!"".equals(intent_cycle_time)))
    {
        onAuthKey();
    } else {
        finish();
    }
    break;
}
} catch (Exception e) {
    e.printStackTrace();
} finally {
}
}

public void onAuthKey() {
    String otp_org = "", otp_new = "";

    try {
        tv_auth_key.setText("");
        createdMillis = estimateCreatedMillis(intent_cycle_time);

        otp_org = barokey.generateKEYP(intent_secure_key, intent_cycle_time);
        otp_new = otp_org.substring(0, 3) + " " + otp_org.substring(3, 6);
        tv_auth_key.setText(otp_new);

        m_handlerProc.sendMessageDelayed(MESSAGE_REFRESH_REMAINING_SECOND,
SENDMESSAGE_INTERVAL);
    } catch (Exception e) {
        e.printStackTrace();
        Util.MsgToast(PAMCreateACT.this, getString(R.string.err_secure_key), 0);

        try { Thread.sleep(3000); } catch (Exception ee) {}
    } finally {
    }
}

private final Handler m_handlerProc = new Handler() {
    @Override
    public void handleMessage(Message message) {
        switch (message.what) {
            case MESSAGE_REFRESH_REMAINING_SECOND: {
                try {
                    long    cycleMillis    = (Long.parseLong(intent_cycle_time) *
1000L);

                    long    remainingMillis    =
estimateRemainingMillis(intent_cycle_time, createdMillis);
                    long    remainingSecond = remainingMillis != 0 ? (remainingMillis /
1000L) : 0;

                    if (0 < remainingMillis) {

m_handlerProc.sendMessageDelayed(MESSAGE_REFRESH_REMAINING_SECOND,
SENDMESSAGE_INTERVAL);

```

```

        btn_reset.setEnabled(false);
    } else {
        Thread.sleep(500);
    }
    m_handlerProc.removeMessages(MESSAGE_REFRESH_REMAINING_SECOND);
    btn_reset.setEnabled(true);
    tv_remainTime.setText(remainingSecond + " " +
    getString(R.string.remain_time_suffix));

    if (0 != cycleMillis) {
        progressBar.setProgress((int) (((float) remainingMillis /
(float) cycleMillis) * 100.0F));
    }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
    }
    }
    break;
}
}
};

public long estimateCreatedMillis(String cycleSecondString) {
    long remainingMillis = (barokey.getRemainingTime(cycleSecondString) * 1000L) -
200;
    long cycleMillis = (Long.parseLong(cycleSecondString) * 1000L);
    long currentMillis = (new Date()).getTime();
    long elapsedMillis = cycleMillis - remainingMillis;
    long createdMillis = currentMillis - elapsedMillis;

    return createdMillis;
}

public long estimateRemainingMillis(String cycleSecondString, long createdTime) {
    long cycleMillis = (Long.parseLong(cycleSecondString) * 1000L);
    long currentMillis = (new Date()).getTime();
    long elapsedMillis = currentMillis - createdTime;

    long remainingMillis = barokey.getRemainingTime(cycleSecondString) * 1000L;
    remainingMillis = cycleMillis > elapsedMillis ? remainingMillis : 0;
    remainingMillis = remainingMillis >= cycleMillis ? 0 : remainingMillis;

    return remainingMillis;
}
}
}

```

2.2 아이폰인 경우

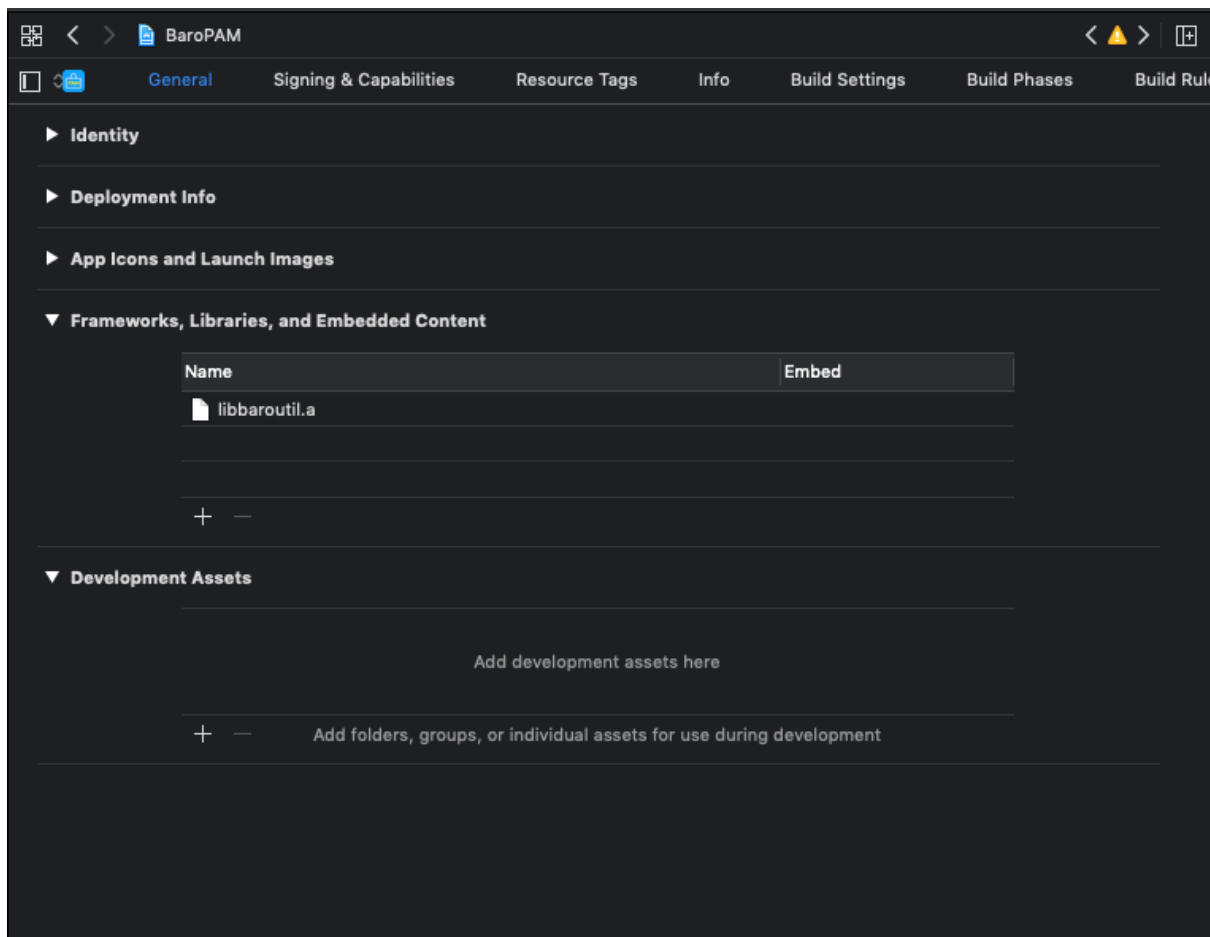
"BaroPAM" 앱에서 서버 로그인 시 "Verification code"에 입력할 일회용 인증키를 생성하는 API는 "libbaroutil.a"으로 제공되며, 이 파일에는 barokey, barocrypt, base64 관련 라이브러리를 포함하는

NSObject Interface용 라이브러리 파일이다.

라이브러리 파일은 두가지 종류로 제공한다. XCode의 iPhone simulator 용과, iPhone용 두가지를 필요에 따라 "libbaroutil.a"로 변경하여 사용한다.

- libbaroutil.a.iphone : iPhone용
- libbaroutil.a.simul : iPhone simulator 용

이 파일은 아래와 같이 XCode의 프로젝트 설정시에 등록하여 사용한다.



BaroKEY 관련된 API는 다음과 같다. 함수는 C 함수 Interface로 되어 있어, 입력값의 자료형은 C 함수 스타일로 표기한다. 사용 예의 소스는 iOS swift 5.0 이상으로 작성된 코드이다.

generateKEYP 함수

서버 로그인 시 사용하는 **일회용 인증키**는 생성하는 함수이다.

입력변수	const char *secure_key	벤더에 요청하여 제공받은 키로 서버별, 계정별로 부여할 수 있으며, 반드시 BaroPAM 서버 환경 설정 시 지정한 키를 설정한다.
	const char *cycle_time	BaroPAM 서버 환경설정 시 지정한 일회용 인증키 의 생성 주기(3~60초)와 반드시 일치해야 한다. 만약, 서버별로 지정한 일회용 인증키 의 생성 주기가 다른 경우 일회용 인증키 가 다르게 생성 될 수 있다.
	const char *key_method	일회용 인증키 의 생성 방식(app1, app256, app384, app512: 앱)으로 "app512"를 설정한다.

리턴 값	일회용인증 키	생성한 일회용 인증키 를 반환한다.
------	------------	----------------------------

Swift 5.0 이상에서의 사용예)

```
private func makePamInfo() {
    let secureKey = _pam?.SECURE_KEY
    let time      = (_pam?.CYCLE_TIME ?? "30")!
    let otpnum    = generateKEYP(secureKey, time, "app512")

    _pamInfo.text = "[ W(_pam?.SERVER_NM ?? "")/W(_pam?.HOSTNAME ?? "") ]"
    let otpnumStr = String(cString: otpnum!)
    let start    = otpnumStr.index(otpnumStr.startIndex, offsetBy: 0)
    let end      = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
    let start2   = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
    let end2     = otpnumStr.index(otpnumStr.startIndex, offsetBy: 6)

    _tfOTP.text = otpnumStr[start..

```

화면 예시)



화면 Layout 예시)

Storyboard를 의미한다. 각 파라미터에 대한 의미는 developer.apple.com을 참고한다.

```
<!--Create View Controller-->
<scene sceneID="sVy-j4-Rtq">
  <objects>
    <viewController storyboardIdentifier="CreatePAM" id="z9w-lb-ka0"
      customClass="PAMCreateViewController" customModule="BaroPAM" customModuleProvider="target"
      sceneMemberID="viewController">
      <layoutGuides>
        <viewControllerLayoutGuide type="top" id="03y-ma-Ukk"/>
        <viewControllerLayoutGuide type="bottom" id="WKP-o0-5Tt"/>
      </layoutGuides>
      <view key="view" contentMode="scaleToFill" id="ZXo-9u-j0g">
        <rect key="frame" x="0.0" y="0.0" width="375" height="812"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
        <subviews>
          <view contentMode="scaleToFill" fixedFrame="YES"
            translatesAutoresizingMaskIntoConstraints="NO" id="VzL-Hy-Qth">
            <rect key="frame" x="0.0" y="0.0" width="375" height="70"/>
            <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
            <subviews>
              <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
                contentHorizontalAlignment="center" contentVerticalAlignment="center"
                lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="PTg-Bp-4P9">
                <rect key="frame" x="283" y="35" width="31" height="31"/>
                <autoresizingMask key="autoresizingMask" flexibleMinX="YES" flexibleMaxY="YES"/>
                <inset key="imageEdgeInsets" minX="3" minY="3" maxX="3" maxY="3"/>
                <state key="normal" image="btn_share.png"/>
              </subviews>
            </view>
          </subviews>
        </view>
      </viewController>
    </objects>
  </scene>
```

```

        <action selector="onShare:" destination="VZn-Wm-75J" eventType="touchUpInside"
id="2aI-2K-VMV"/>
        <action selector="onShare:" destination="z9w-lb-ka0" eventType="touchUpInside"
id="Hvz-hY-Yno"/>
    </connections>
</button>
<button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center"
lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="eey-kg-kh0">
    <rect key="frame" x="322" y="34" width="33" height="32"/>
    <autoresizingMask key="autoresizingMask" flexibleMinX="YES" flexibleMaxY="YES"/>
    <inset key="imageEdgeInsets" minX="3" minY="3" maxX="3" maxY="3"/>
    <state key="normal" image="btn_setting.png"/>
    <connections>
        <action selector="onSetting:" destination="z9w-lb-ka0" eventType="touchUpInside"
id="UHA-w0-txo"/>
        <action selector="onSetting:" destination="VZn-Wm-75J" eventType="touchUpInside"
id="qN9-C1-NKq"/>
    </connections>
</button>
<imageView contentMode="scaleAspectFit" horizontalHuggingPriority="251"
verticalHuggingPriority="251" fixedFrame="YES" image="btn_prev.png"
translatesAutoresizingMaskIntoConstraints="NO" id="FyA-Yo-4LD">
    <rect key="frame" x="19" y="35" width="31" height="31"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" heightSizable="YES"/>
</imageView>
<imageView userInteractionEnabled="NO" contentMode="scaleAspectFit"
horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
image="logo_barootp.png" translatesAutoresizingMaskIntoConstraints="NO" id="AWz-gv-3pq">
    <rect key="frame" x="115" y="38" width="145" height="25"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
</imageView>
</subviews>
<color key="backgroundColor" red="0.1058823529" green="0.56470588239999997" blue="1"
alpha="1" colorSpace="calibratedRGB"/>
</view>
<textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="scaleToFill"
fixedFrame="YES" text="TITLE MESSAGE" textAlignment="center"
translatesAutoresizingMaskIntoConstraints="NO" id="6aI-WL-0dk">
    <rect key="frame" x="0.0" y="125" width="375" height="40"/>
    <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
    <color key="textColor" white="0.0" alpha="1" colorSpace="calibratedWhite"/>
    <fontDescription key="fontDescription" name="SpogaHanSans-Regular"
family="SpogaHanSans" pointSize="17"/>
    <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
</textView>
<textField opaque="NO" clipsSubviews="YES" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="left" contentVerticalAlignment="center" text="12345678"
textAlignment="center" minimumFontSize="17" translatesAutoresizingMaskIntoConstraints="NO"
id="N6N-XM-OBH">
    <rect key="frame" x="19" y="204" width="336" height="52"/>
    <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
    <color key="backgroundColor" white="1" alpha="1" colorSpace="calibratedWhite"/>

```

```

        <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="50"/>
        <textInputTraits key="textInputTraits"/>
    </textField>
    <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect"
lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="jvU-oe-4pE">
        <rect key="frame" x="23" y="683" width="160" height="43"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxX="YES"
flexibleMinY="YES"/>
        <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
        <state key="normal" title="Modify/Delete">
            <color key="titleColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
        </state>
        <connections>
            <action selector="onEdit:" destination="BPh-TI-Gd5" eventType="touchUpInside"
id="0YZ-mm-tgY"/>
            <action selector="onEdit:" destination="z9w-lb-ka0" eventType="touchUpInside"
id="5bi-1P-cae"/>
            <action selector="onOk:" destination="BYZ-38-t0r" eventType="touchUpInside" id="K5T-
jU-wz5"/>
        </connections>
    </button>
    <button opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
contentHorizontalAlignment="center" contentVerticalAlignment="center" buttonType="roundedRect"
lineBreakMode="middleTruncation" translatesAutoresizingMaskIntoConstraints="NO" id="OMa-U8-E40">
        <rect key="frame" x="199" y="683" width="160" height="43"/>
        <autoresizingMask key="autoresizingMask" flexibleMinX="YES" widthSizable="YES"
flexibleMinY="YES"/>
        <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
        <state key="normal" title="Reset">
            <color key="titleColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
        </state>
        <connections>
            <action selector="onReset:" destination="BPh-TI-Gd5" eventType="touchUpInside"
id="5pN-uP-iTd"/>
            <action selector="onReset:" destination="z9w-lb-ka0" eventType="touchUpInside"
id="bFb-Wq-VEE"/>
        </connections>
    </button>
    <textView clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="scaleToFill"
fixedFrame="YES" editable="NO" text="USER DEFINED MESSAGE." textAlignment="natural"
selectable="NO" translatesAutoresizingMaskIntoConstraints="NO" id="tvu-My-d27">
        <rect key="frame" x="48" y="585" width="276" height="112"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMinY="YES"/>
        <color key="textColor" red="0.33333333333333331" green="0.33333333333333331"
blue="0.33333333333333331" alpha="1" colorSpace="calibratedRGB"/>
        <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
        <textInputTraits key="textInputTraits" autocapitalizationType="sentences"/>
    </textView>

```

```

    <progressView opaque="NO" contentMode="scaleToFill" verticalHuggingPriority="750"
fixedFrame="YES" progress="0.5" translatesAutoresizingMaskIntoConstraints="NO" id="OPH-FY-y7I">
    <rect key="frame" x="52" y="274" width="270" height="2"/>
    <autoresizingMask key="autoresizingMask" widthSizable="YES"/>
</progressView>
    <imageView userInteractionEnabled="NO" contentMode="scaleToFill"
horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
image="ico_countdown.png" translatesAutoresizingMaskIntoConstraints="NO" id="NCT-hK-zKs">
    <rect key="frame" x="250" y="284" width="15" height="15"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
</imageView>
    <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES" text="0"
textAlignment="natural" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines"
adjustsFontSizeToFit="NO" translatesAutoresizingMaskIntoConstraints="NO" id="csq-gk-Jba">
    <rect key="frame" x="270" y="281" width="52" height="21"/>
    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
    <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
    <color key="textColor" red="0.3333333333333331" green="0.3333333333333331"
blue="0.3333333333333331" alpha="1" colorSpace="calibratedRGB"/>
    <nil key="highlightedColor"/>
</label>
    <label opaque="NO" userInteractionEnabled="NO" contentMode="left"
horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
text="[david.kscho@empluses.com]" textAlignment="center" lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="jeL-QQ-baB">
    <rect key="frame" x="23" y="318" width="332" height="30"/>
    <autoresizingMask key="autoresizingMask" widthSizable="YES" flexibleMaxY="YES"/>
    <fontDescription key="fontDescription" name="SpoqaHanSans-Regular"
family="SpoqaHanSans" pointSize="17"/>
    <nil key="highlightedColor"/>
</label>
</subviews>
    <color key="backgroundColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
</view>
<connections>
    <outlet property="_backView" destination="FyA-Yo-4LD" id="bIF-3v-gYz"/>
    <outlet property="_btnReset" destination="OMa-U8-E40" id="cIb-IR-Fiy"/>
    <outlet property="_btnModify" destination="jvU-oe-4pE" id="6Ic-hF-FDd"/>
    <outlet property="_pamInfo" destination="jeL-QQ-baB" id="sz4-CU-y0h"/>
    <outlet property="_progress" destination="OPH-FY-y7I" id="5Uc-Je-MGC"/>
    <outlet property="_remainTime" destination="csq-gk-Jba" id="AX4-sQ-kSL"/>
    <outlet property="_tfOTP" destination="N6N-XM-OBH" id="OG3-OE-Faf"/>
</connections>
</viewController>
<placeholder placeholderIdentifier="IBFirstResponder" id="5TR-YP-Zsu" userLabel="First
Responder" sceneMemberID="firstResponder"/>
</objects>
<point key="canvasLocation" x="2948" y="-243.10344827586209"/>
</scene>

```

프로그램 예시)

```

import UIKit

class PAMCreateViewController: UIViewController {
    @IBOutlet weak var _backView: UIImageView!
    @IBOutlet weak var _progress: UIProgressView!
    @IBOutlet weak var _remainTime: UILabel!
    @IBOutlet weak var _pamInfo: UILabel!
    @IBOutlet weak var _tfOTP: UITextField!
    @IBOutlet weak var _btnModify: UIButton!
    @IBOutlet weak var _btnReset: UIButton!

    var _timer: Timer?
    var _pam: PAMEntity? = nil

    override func viewDidLoad() {
        super.viewDidLoad()
        initControls()
        makeTappedView()
        makePamInfo()
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        if (_pam?.IS_DELETE == 1) {
            _pam?.IS_DELETE = 0
            dismiss(animated: false, completion: nil)
        }
    }

    override func viewDidAppear(_ animated: Bool) {
        super.viewDidAppear(animated)
    }

    private func initControls() {
        _btnModify.backgroundColor = UIColorFromHex(rgbValue: 0x1B90FF)
        _btnReset.backgroundColor = UIColorFromHex(rgbValue: 0x1B90FF)
    }

    private func makeTappedView() {
        let tap = UITapGestureRecognizer(target: self, action:
#selector(PAMCreateViewController.backTapped))
        _backView.isUserInteractionEnabled = true
        _backView.addGestureRecognizer(tap)
    }

    @objc func backTapped(tapGestureRecognizer: UITapGestureRecognizer) {
        dismiss(animated: false, completion: nil) //
    }

    private func makePamInfo() {

```

```

let secureKey = _pam?.SECURE_KEY
let time      = (_pam?.CYCLE_TIME ?? "30")!
let otpnum    = generateKEYP(secureKey, time, "0", "app512")
//print( "generateTOTPP ==> W(String(describing: secureKey)), W(String(cString:
otpnum!)) " )

_pamInfo.text = "[ W(_pam?.SERVER_NM ?? "")/W(_pam?.HOSTNAME ?? "") ]"

let otpnumStr = String(cString: otpnum!)
let start     = otpnumStr.index(otpnumStr.startIndex, offsetBy: 0)
let end       = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
let start2    = otpnumStr.index(otpnumStr.startIndex, offsetBy: 3)
let end2      = otpnumStr.index(otpnumStr.startIndex, offsetBy: 6)

_tfOTP.text   = otpnumStr[start..

```



```
@IBAction func onSetting(_ sender: Any) {  
    switchScreen("Settings")  
}  
}
```

2. BaroPAM 웹 사이트

정보자산에 로그인 시 Verification code에 입력할 **일회용 인증키**의 생성기인 "BaroPAM" 앱(안드로이드 폰 용과 아이폰 용)에 문제가 발생 시 서비스의 중단이 발생하지 않도록 BaroPAM 웹 사이트(www.baropam.com)를 통해서도 서비스를 제공한다.

"BaroPAM" 웹 화면은 크게 회원가입, 로그인, 서버 정보 관리, 애플리케이션 정보 관리, PIN 정보 관리, 접속로그 관리, Abort, Logout 등으로 구성되어 있다.

2.1 BaroPAM 웹 사용(회원가입)

1. 스마트 폰에서 "웹 브라우저" 앱의 아이콘을 클릭하여 앱을 활성화한 후 앱 상단의 "검색어 또는 URL 입력"란에 BaroPAM의 웹 사이트인 www.baropam.com URL을 입력한다.

(안드로이드 폰인 경우)



아이폰인 경우)



2. BaroPAM 웹 사이트가 활성화되면 다음과 같은 "BaroPAM"의 로그인 화면이 나타난다.



"BaroPAM" 웹 사이트(www.baropam.com)를 처음 사용하는 경우는 "회원가입" 버튼을, 이미 사용한 경우는 휴대폰 번호 11자리와 PIN 번호 8자리를 입력한 후 "로그인" 버튼을 클릭한다.

3. 회원가입이 되어 있지 않은 경우("BaroPAM" 웹 사이트를 처음 사용하는 경우) "회원가입" 버튼을 클릭하면 다음과 같이 회원 가입하는 화면이 나타난다.

PIN번호는 스마트 폰 분실 시 타인이 인증키 정보 유출을 방지하기 위해서 사용한다.

4. 사용하는 휴대전화 11자리 숫자와 PIN 번호 8자리 숫자를 입력한 후 "Save" 버튼을 클릭한다.

"Save" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 회원가입 정보를 저장하는 작업을 진행한다.

휴대폰 번호가 11자리가 아닌 경우 "**휴대번호를 입력해주세요.**"라는 메시지가 화면에 나타난다. 마찬가지로, PIN 번호(PIN Number)가 8자리가 아닌 경우 "**PIN 번호를 입력하세요.**"라는 메시지가 화면에 나타난다.

회원가입 정보가 정상적으로 완료되지 못하면 "**회원가입**" 화면이 지속되며, 정상적으로 완료되면 "**로그인**" 화면이 나타난다.

"Cancel" 버튼을 클릭하면 현재 화면을 종료하고 "**로그인**" 화면이 나타난다.

2.2 BaroPAM 웹 사용(로그인)

1. BaroPAM 웹 사이트가 활성화되면 다음과 같은 "BaroPAM"의 로그인 화면이 나타난다.



"BaroPAM" 웹 사이트(www.baropam.com)를 처음 사용하는 경우는 "회원가입" 버튼을, 이미 사용한 경우는 휴대폰 번호 11자리와 PIN 번호 8자리를 입력한 후 "로그인" 버튼을 클릭한다.

2. 로그인 화면에서 설정한 11자리 휴대번호와 8자리 PIN 번호(PIN Number)를 입력한 다음 "로그인" 버튼을 클릭한다.

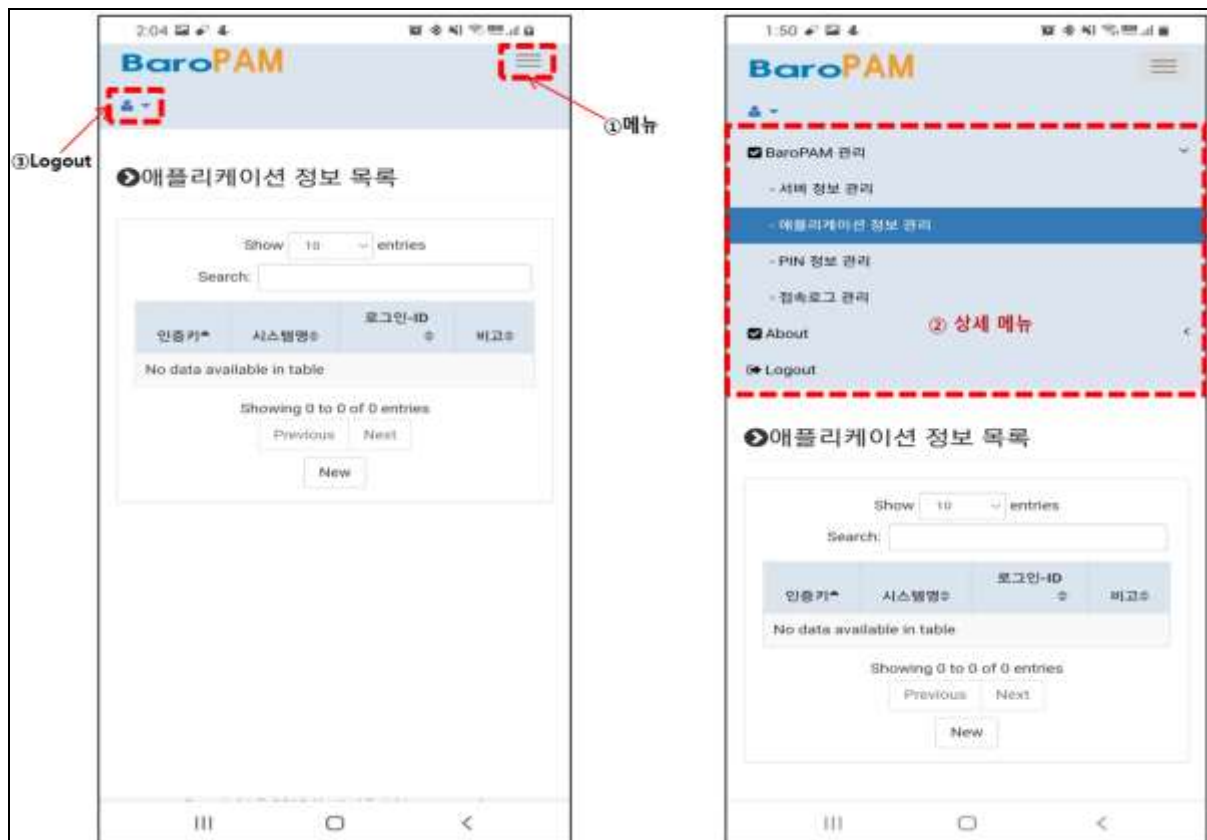


"로그인" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인하는 작업을 진행한다.

만약, 휴대폰 번호가 11자리가 아닌 경우 "**휴대번호를 입력해주시요.**"라는 메시지가 화면에 나타난다. 마찬가지로, PIN 번호(PIN Number)가 8자리가 아닌 경우 "**PIN 번호를 입력하세요.**"라는 메시지가 화면에 나타난다.

입력한 휴대폰 번호가 틀린 경우 "**휴대폰 번호를 확인해주세요.**", PIN 번호(PIN Number)가 틀린 경우 "**PIN 번호를 확인해주세요.**"라는 메시지가 화면에 나타난다.

3. 로그인이 확인되면, 다음과 같이 "애플리케이션 정보 관리" 화면이 나타난다.



"애플리케이션 정보 관리" 화면은 애플리케이션의 로그인 시 비밀번호를 대체하기 위하여 "일회용 인증키"를 생성하기 위한 정보를 관리한다.

"① 메뉴" 버튼을 클릭하면 우측 화면처럼 선택할 수 있는 "② 상세 메뉴"가 나타나며, "③ Logout" 버튼을 클릭하면 "BaroPAM" 웹 사이트(www.baropam.com) 화면을 종료하고 "로그인" 화면이 나타난다.

2.3 BaroPAM 웹 사용(애플리케이션정보관리)

1. 오른쪽 상단에 있는 "메뉴" 버튼을 클릭한 후 상세 메뉴에서 "애플리케이션 정보 관리" 메뉴를 클릭하면 다음과 같이 "애플리케이션 정보 목록" 화면이 나타난다.



2. "애플리케이션 정보 목록" 화면에서 "New" 버튼을 클릭하면 다음과 같이 애플리케이션의 로그인에 대한 정보를 등록하는 화면이 나타난다.

■시스템명(System name)

로그인할 시스템명(System name)은 필수 입력 항목으로 최소 1자리 이상 최대 60자리까지 입력할 수 있다.

■로그인-ID(Login-ID)

로그인-ID(Login-ID)는 필수 입력 항목으로 최소 1자리 이상 최대 50자리까지 입력할 수 있다.

■생성주기(Auth key cycle time, 3~60 second)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, BaroPAM 검증모듈의 **일회용 인증키**의 생성주기(Auth key cycle time)와 웹에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

3. 애플리케이션의 로그인 정보를 입력한 후 "Save" 버튼을 클릭한다.

시스템명	BaroPAM
로그인 아이디	mc529@nurit.co.kr
생성주기(초)	20

List Save

"Save" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 애플리케이션의 로그인 정보를 저장하는 작업을 진행한다.

시스템명(System name)이 입력하지 않은 경우 **"시스템명을 입력하세요."**라는 메시지가 화면에 나타난다.

로그인-ID(Login-ID)가 입력하지 않은 경우 **"로그인-ID를 입력하세요."**라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 **"인증키 생성주기를 입력 또는 확인해주세요."**라는 메시지가 화면에 나타난다.

"List" 버튼을 클릭하면 현재 화면을 종료하고 **"애플리케이션 정보 목록"** 화면이 나타난다.

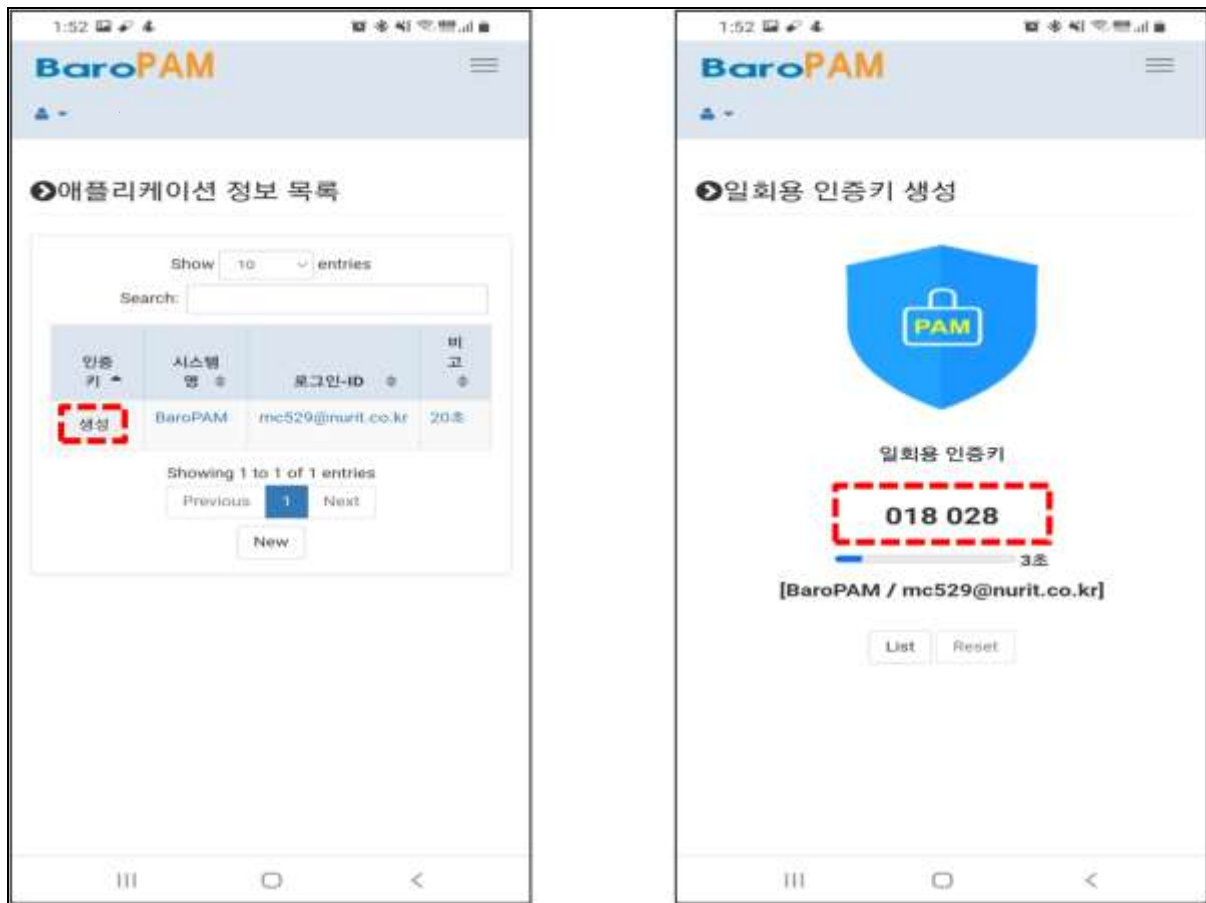
4. 애플리케이션의 로그인에 대한 정보 저장이 정상적으로 완료되지 못하면 "애플리케이션 정보 등록" 화면이 지속되며, 정상적으로 완료되면 다음과 같이 "애플리케이션 정보 목록" 화면이 나타난다.



애플리케이션의 로그인 정보를 검색하고 싶은 경우 검색어를 입력한 다음 "Search" 버튼을 클릭하면 검색된 애플리케이션의 로그인 정보가 화면에 나타난다.

애플리케이션의 로그인 정보를 신규로 추가해야 할 경우 "New" 버튼을 클릭하면 애플리케이션의 로그인 정보를 등록하는 화면이 나타난다.

5. 애플리케이션 정보 목록에서 "생성" 버튼을 클릭하면 새로운 **2차 인증키(일회용 인증키)**가 생성되는데, **일회용 인증키**의 생성주기(Auth key cycle time) 동안은 동일한 **2차 인증키(일회용 인증키)**가 발생하며, 새로 생성한 **일회용 인증키**를 다음과 같이 "일회용 인증키 생성" 화면 상에 나타난다.



"List" 버튼을 클릭하면 현재 화면을 종료하고 "애플리케이션 정보 목록" 화면이 나타난다.

동일한 인증키 생성주기(Auth key cycle time) 동안 "Reset" 버튼은 비활성화되며, 인증키 생성주기(Auth key cycle time)가 지나면 "Reset" 버튼은 활성화시킨다.

활성화된 "Reset" 버튼을 클릭하면 새로운 **일회용 인증키**가 "일회용 인증키 생성" 화면 상에 생성된다.

6. **2차 인증키(일회용 인증키)**를 생성 및 애플리케이션 정보 변경/삭제해야 하는 경우 애플리케이션 정보 목록에서 애플리케이션의 로그인에 대한 정보 항목을 다음과 같이 클릭한다.



7. 그러면 다음과 같이 애플리케이션의 로그인에 대한 정보를 바탕으로 **2차 인증키(일회용 인증키)**를 생성 및 변경/삭제하는 화면이 나타난다.

The screenshot shows a mobile application interface for BaroPAM. At the top, there's a header with the BaroPAM logo and a menu icon. Below the header, the title is '애플리케이션 정보 수정/삭제' (Application Information Modification/Deletion). The main content area contains a form with the following fields:

- 시스템명 (System Name): BaroPAM
- 로그인 아이디 (Login ID): mc529@nurit.co.kr
- 생성주기(초) (Key Generation Cycle (sec)): 20
- 등록일시 (Registration Time): 2022-04-30 13:52:13.952001

Below the form, there are four buttons: List, Delete, Modify, and One Time Auth key.

■시스템명(System name)

로그인할 시스템명(System name)은 필수 입력 항목으로 최소 1자리 이상 최대 60자리까지 입력할 수 있다.

■로그인-id(Login-ID)

로그인-ID(Login-ID)는 필수 입력 항목으로 최소 1자리 이상 최대 50자리까지 입력할 수 있다.

■생성주기(Auth key cycle time, 3~60 second)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, BaroPAM 검증모듈의 **일회용 인증키**의 생성주기(Auth key cycle time)와 웹에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

시스템명(System name)이 입력하지 않은 경우 "시스템명을 입력하세요."라는 메시지가 화면에 나타난다.

로그인-ID(Login-ID)가 입력하지 않은 경우 "로그인-ID를 입력하세요."라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 "**인증키 생성주기를 입력 또는 확인**해주시시오."

애플리케이션의 로그인 정보 수정(Modify) 또는 삭제>Delete)가 정상적으로 완료되지 못하면 "**애플리케이션 정보 수정/삭제**" 화면이 지속되며, 정상적으로 완료되면 "**애플리케이션 정보 목록**" 화면이 나타난다.



"Modify" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 애플리케이션의 로그인에 대한 정보를 수정하는 작업을 진행한다.

"Delete" 버튼을 클릭하면 "**애플리케이션 정보를 삭제 하시겠습니까?**"라는 Message box가 나타나며, "**확인**" 버튼을 클릭하면 해당 애플리케이션의 로그인에 대한 정보를 삭제하는 작업을 진행되며, "**취소**" 버튼을 클릭하면 삭제 작업이 취소된다.

"List" 버튼을 클릭하면 현재 화면을 종료하고 "**애플리케이션 정보 목록**" 화면이 나타난다.

"One Time Auth key" 버튼을 클릭하면 새로운 **2차 인증키(일회용 인증키)**가 생성되는데, **일회용 인증키**의 생성주기(Auth key cycle time) 동안은 동일한 **2차 인증키(일회용 인증키)**가 발생하며, 새로 생성한 **일회용 인증키**를 다음과 같이 "**일회용 인증키 생성**" 화면 상에 나타난다.



"List" 버튼을 클릭하면 현재 화면을 종료하고 "애플리케이션 정보 목록" 화면이 나타난다.

동일한 인증키 생성주기(Auth key cycle time) 동안 "Reset" 버튼은 비활성화되며, 인증키 생성주기(Auth key cycle time)가 지나면 "Reset" 버튼은 활성화시킨다.

활성화된 "Reset" 버튼을 클릭하면 새로운 **일회용 인증키**가 "일회용 인증키 생성" 화면 상에 생성된다.

2.4 BaroPAM 웹 사용(서버정보관리)

1. 오른쪽 상단에 있는 "메뉴" 버튼을 클릭한 후 상세 메뉴에서 "서버 정보 관리" 메뉴를 클릭하면 다음과 같이 "서버 정보 목록" 화면이 나타난다.



2. "서버 정보 목록" 화면에서 "New" 버튼을 클릭하면 다음과 같이 서버에 대한 정보를 등록하는 화면이 나타난다.

■서버명(Server name)

BaroPAM이 운영될 서버명(Server name)은 필수 입력 항목으로 최소 1자리 이상 최대 30자리까지 입력할 수 있다.

■Secure key

정보자산별 또는 계정별로 부여되는 Secure key는 필수 입력 항목으로 벤더에 요청하여 부여받은 것을 입력해야 한다.

벤더에서 부여받지 않은 임의의 "Secure key"를 입력하는 경우 잘못된 **일회용 인증키**가 부여되어 정보자산에 로그인 할 수 없는 경우가 발생할 수 있다.

만약, 정보자산에 설정한 Secure key와 앱에서 지정한 Secure key가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

■생성주기(Auth key cycle time, 3~60 second)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, 정보자산에 설정한 **일회용 인증키**의 생성주기(Auth key cycle time)와 앱에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

3. 서버 정보를 입력한 후 "Save" 버튼을 클릭한다.

"Save" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 정보자산에 대한 정보를 저장하는 작업을 진행한다.

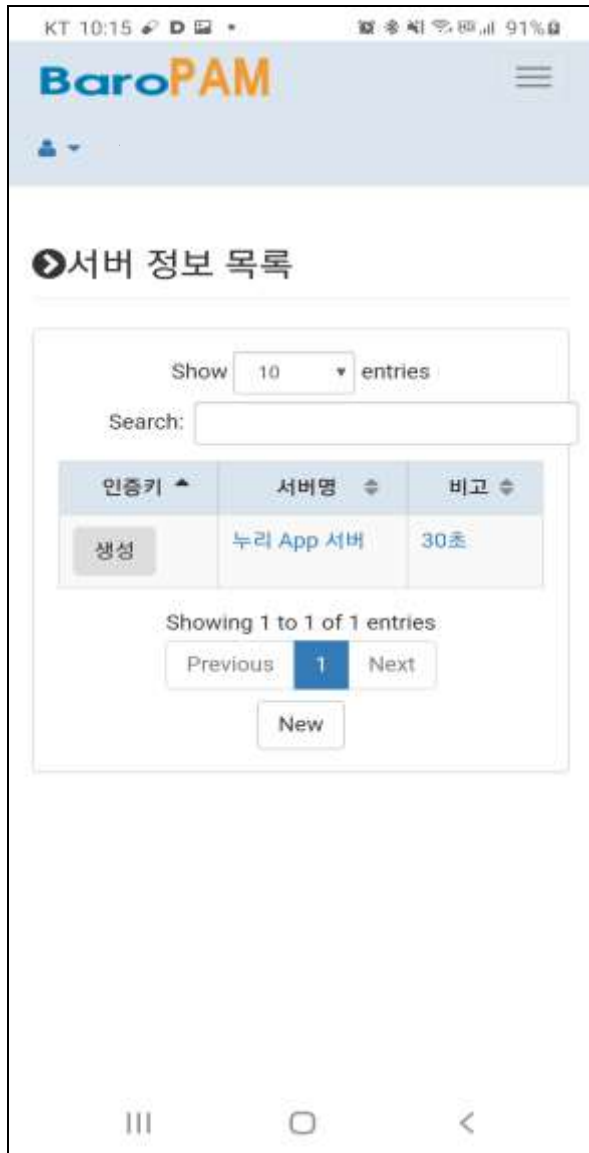
정보자산명(Server name)이 입력하지 않은 경우 "**서버명을 입력하세요.**"라는 메시지가 화면에 나타난다.

Secure key가 입력하지 않은 경우 "**Secure key를 입력하세요.**"라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 "**인증키 생성주기를 입력 또는 확인해주시오.**"라는 메시지가 화면에 나타난다.

"List" 버튼을 클릭하면 현재 화면을 종료하고 "**서버 정보 목록**" 화면이 나타난다.

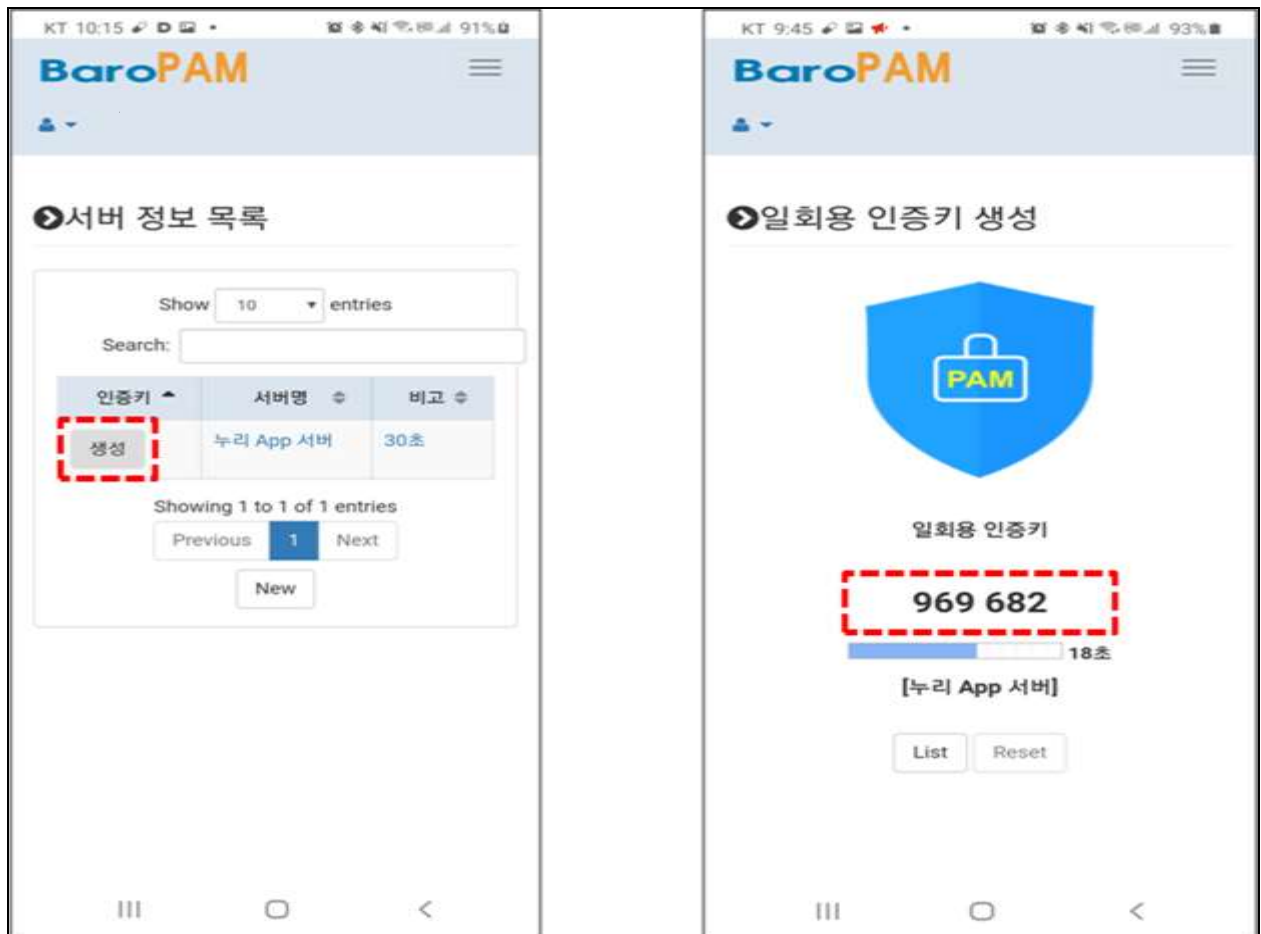
4. 서버에 대한 정보 저장이 정상적으로 완료되지 못하면 "서버 정보 등록" 화면이 지속되며, 정상적으로 완료되면 다음과 같이 "서버 정보 목록" 화면이 나타난다.



서버 정보를 검색하고 싶은 경우 검색어를 입력한 다음 "Search" 버튼을 클릭하면 검색된 서버 정보가 화면에 나타난다.

서버 정보를 신규로 추가해야 할 경우 "New" 버튼을 클릭하면 서버 정보를 등록하는 화면이 나타난다.

5. 서버 정보 목록에서 "생성" 버튼을 클릭하면 새로운 2차 인증키(일회용 인증키)가 생성되는데, 일회용 인증키의 생성주기(Auth key cycle time) 동안은 동일한 2차 인증키(일회용 인증키)가 발생하며, 새로 생성한 일회용 인증키를 다음과 같이 "일회용 인증키 생성" 화면 상에 나타난다.

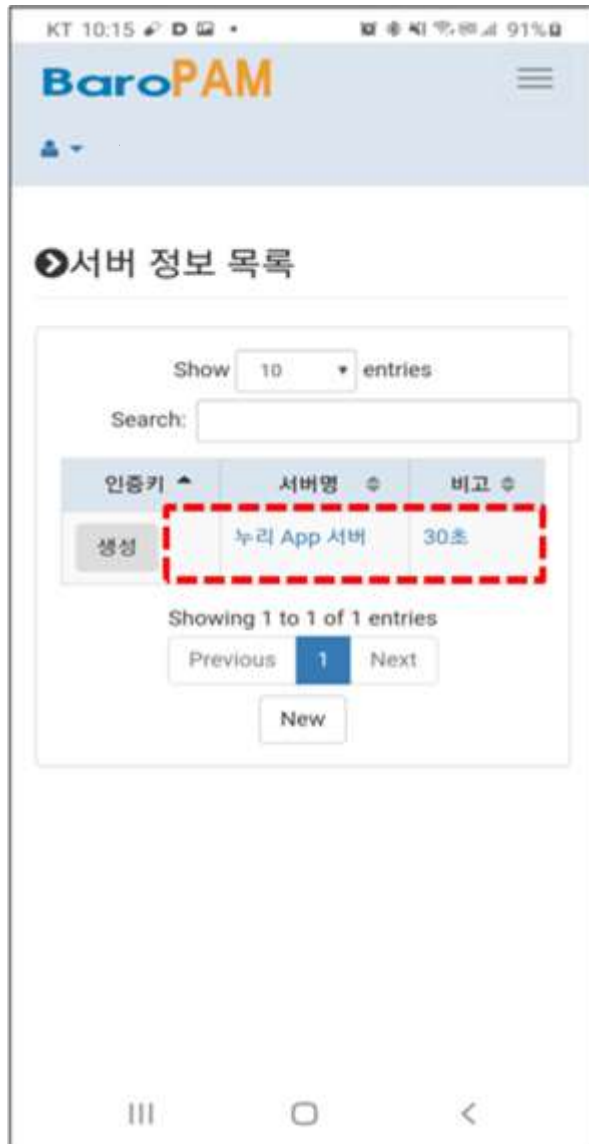


"List" 버튼을 클릭하면 현재 화면을 종료하고 "서버 정보 목록" 화면이 나타난다.

동일한 인증키 생성주기(Auth key cycle time) 동안 "Reset" 버튼은 비활성화되며, 인증키 생성주기(Auth key cycle time)가 지나면 "Reset" 버튼은 활성화시킨다.

활성화된 "Reset" 버튼을 클릭하면 새로운 일회용 인증키가 "일회용 인증키 생성" 화면 상에 생성된다.

6. **2차 인증키(일회용 인증키)**를 생성 및 서버 정보 변경/삭제해야 하는 경우 서버 정보 목록에서 서버에 대한 정보 항목을 다음과 같이 클릭한다.



7. 그러면 다음과 같이 서버에 대한 정보를 바탕으로 **2차 인증키(일회용 인증키)**를 생성 및 변경/삭제하는 화면이 나타난다.

The screenshot shows the BaroPAM mobile application interface. At the top, the status bar indicates the time is 10:16 and the battery level is 91%. The app header displays the BaroPAM logo and a menu icon. The main title is '서버 정보 수정/삭제' (Server Information Modification/Deletion). Below the title is a form with the following fields:

서버명	누리 App 서버
Secure key	jlqlcHbVqdpj7b4PzBpM2DileBv
생성주기(초)	30
등록일시	2020-07-02 09:23:05.298001

Below the form, there are four buttons: List, Delete, Modify, and Verification key.

■서버명(Server name)

BaroPAM이 운영될 정보자산명(Server name)은 필수 입력 항목으로 최소 1자리 이상 최대 30자리까지 입력할 수 있다.

■Secure key

정보자산별 또는 계정별로 부여되는 Secure key는 필수 입력 항목으로 벤더에 요청하여 부여받은 것을 입력해야 한다.

벤더에서 부여받지 않은 임의의 "Secure key"를 입력하는 경우 잘못된 **일회용 인증키**가 부여되어 정보자산에 로그인 할 수 없는 경우가 발생할 수 있다.

만약, 정보자산에 설정한 Secure key와 앱에서 지정한 Secure key가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

■생성주기(Auth key cycle time, 3~60 second)

일회용 인증키의 생성주기(Auth key cycle time)는 필수 입력 항목으로 최소 3초 이상 최대 60초 까지 지정할 수 있다.

만약, 정보자산에 설정한 **일회용 인증키**의 생성주기(Auth key cycle time)와 앱에서 지정한 **일회용 인증키**의 생성주기(Auth key cycle time)가 다른 경우 **일회용 인증키**가 서로 달라서 로그인 할 수 없는 경우가 발생할 수 있다.

"Delete" 버튼을 클릭하면 해당 정보자산에 대한 정보를 삭제하는 작업을 진행한다.

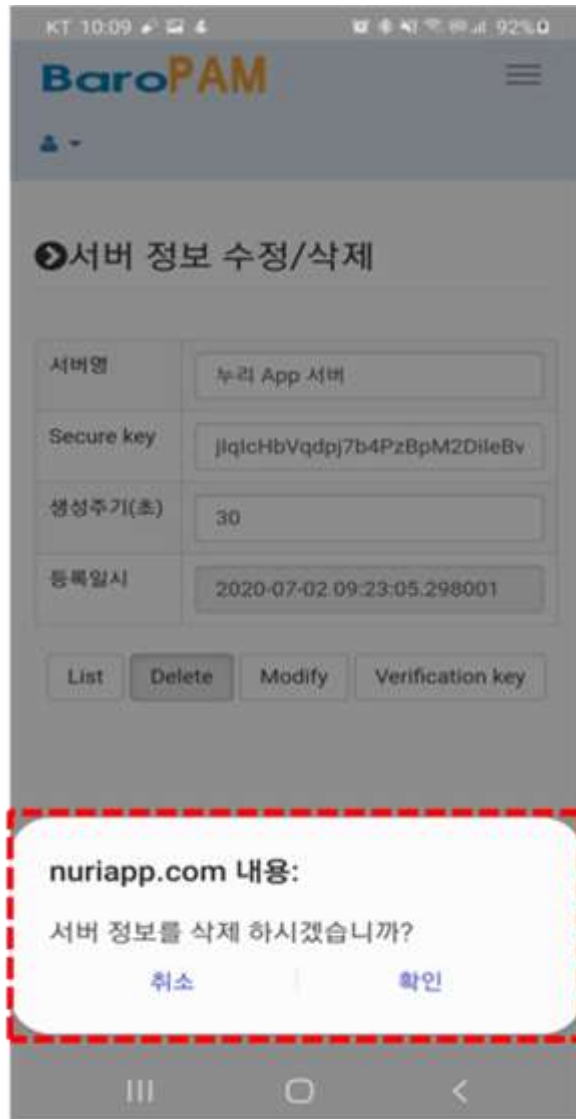
"Modify" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 정보자산에 대한 정보를 수정하는 작업을 진행한다.

정보자산명(Server name)이 입력하지 않은 경우 "서버명을 입력하세요."라는 메시지가 화면에 나타난다.

Secure key가 입력하지 않은 경우 "Secure key를 입력하세요."라는 메시지가 화면에 나타난다.

일회용 인증키의 생성주기(Auth key cycle time)가 입력하지 않거나 범위를 벗어난 경우 "인증키 생성주기를 입력 또는 확인해주시오."라는 메시지가 화면에 나타난다.

정보자산에 대한 정보 수정(Modify) 또는 삭제>Delete)가 정상적으로 완료되지 못하면 "서버 정보 수정/삭제" 화면이 지속되며, 정상적으로 완료되면 "서버 정보 목록" 화면이 나타난다.

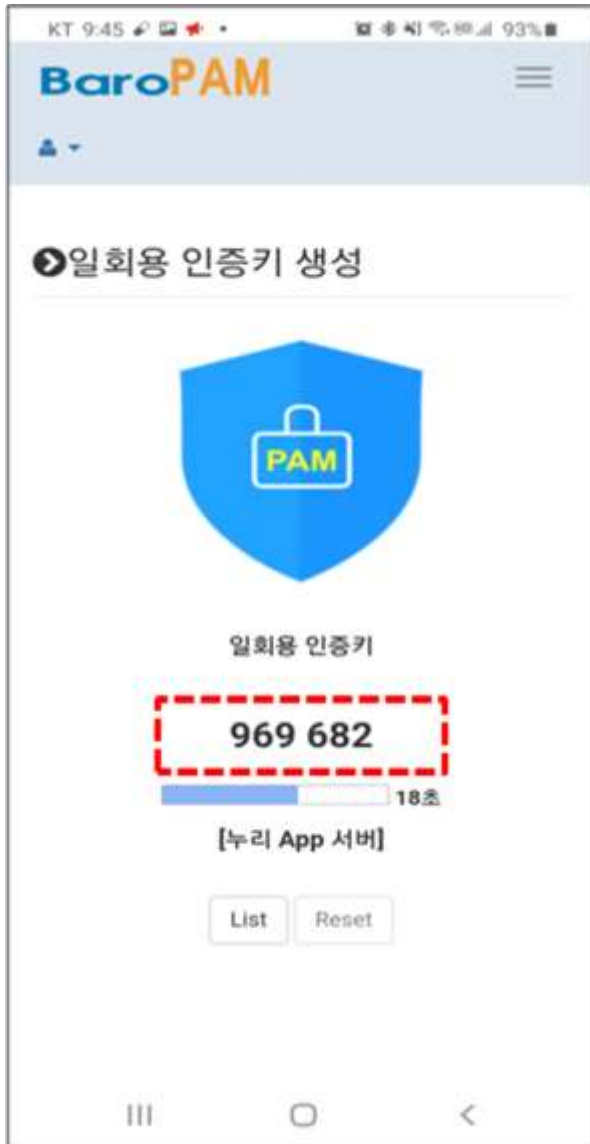


"Modify" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 서버에 대한 정보를 수정하는 작업을 진행한다.

"Delete" 버튼을 클릭하면 "서버 정보를 삭제 하시겠습니까?"라는 Message box가 나타나며, "확인" 버튼을 클릭하면 해당 서버에 대한 정보를 삭제하는 작업을 진행되며, "취소" 버튼을 클릭하면 삭제 작업이 취소된다.

"List" 버튼을 클릭하면 현재 화면을 종료하고 "서버 정보 목록" 화면이 나타난다.

"One Time Auth key" 버튼을 클릭하면 새로운 **2차 인증키(일회용 인증키)**가 생성되는데, **일회용 인증키**의 생성주기(Auth key cycle time) 동안은 동일한 **2차 인증키(일회용 인증키)**가 발생하며, 새로 생성한 **일회용 인증키**를 다음과 같이 "일회용 인증키 생성" 화면 상에 나타난다.



"List" 버튼을 클릭하면 현재 화면을 종료하고 "서버 정보 목록" 화면이 나타난다.

동일한 인증키 생성주기(Auth key cycle time) 동안 "Reset" 버튼은 비활성화되며, 인증키 생성주기(Auth key cycle time)가 지나면 "Reset" 버튼은 활성화시킨다.

활성화된 "Reset" 버튼을 클릭하면 새로운 **일회용 인증키**가 "일회용 인증키 생성" 화면 상에 생성된다.

2.5 BaroPAM 웹 사용(PIN정보관리)

1. 오른쪽 상단에 있는 "메뉴" 버튼을 클릭한 후 상세 메뉴에서 "PIN 정보 관리" 메뉴를 클릭하면 다음과 같이 "PIN 정보 변경" 화면이 나타난다.

PIN 정보 변경	
현재 PIN번호	현재 PIN번호를 입력해주십시오
새로운 PIN번호	새로운 PIN번호를 입력해주십시오
휴대전화	휴대폰 번호를 입력해주십시오
등록일시	2018-12-17 05:29:13.939296
변경일시	2019-08-10 17:25:35.397007
Modify	

■현재 PIN 번호

현재 설정된 PIN 번호(PIN Number) 8자리를 입력한다.

■새로운 PIN 번호

새로운 설정할 PIN 번호(PIN Number) 8자리를 입력한다.

■휴대전화

사용하는 휴대폰 번호를 숫자 11자리를 입력한다.

KT 11:34 80%

BaroPAM

ⓘ PIN 정보 변경

현재 PIN번호	94720256
새로운 PIN번호	12271110
휴대전화	01027714076
등록일시	2018-12-17 05:29:13.939296
변경일시	2019-08-10 17:25:35.397007

Modify

"Modify" 버튼을 클릭하면 제일 먼저 입력 항목들의 유효성을 확인한 후 새로운 PIN 번호(PIN Number)를 저장하는 작업을 진행한다.

만약, PIN 번호(PIN Number)가 8자리가 아닌 경우 "**현재 PIN번호를 확인해주시오.**"라는 메시지가 화면에 나타난다.

현재 설정된 PIN 번호(PIN Number)와 새로 입력한 PIN 번호(PIN Number)가 다른 경우 "**현재 PIN번호가 틀립니다.**"라는 메시지가 화면에 나타난다.

현재 설정된 PIN 번호(PIN Number)와 새로운 PIN 번호(PIN Number)가 동일한 경우 "**현재, 새로운 PIN번호가 동일합니다.**"라는 메시지가 화면에 나타난다.

만약, 새로운 PIN 번호(PIN Number)가 8자리가 아닌 경우 "**새로운 PIN번호를 확인해주시오.**"라는 메시지가 화면에 나타난다.

휴대폰 번호가 11자리가 아닌 경우 "**휴대폰 번호를 확인해주시오.**"라는 메시지가 화면에 나타난다.

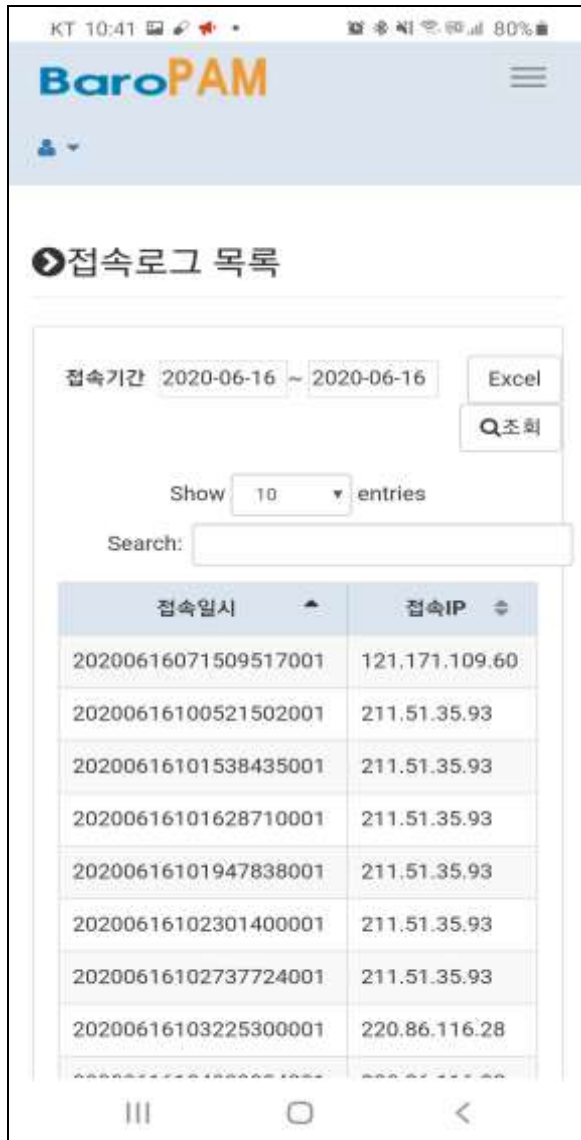
현재 설정된 휴대폰 번호와 입력한 휴대폰 번호가 다른 경우 "**휴대폰 번호가 틀립니다.**"라는 메시지가 화

면에 나타난다.

새로운 PIN 번호(PIN Number) 저장에 정상적으로 완료되지 못하면 "PIN 정보 변경" 화면이 지속되며, 정상적으로 완료되면 "로그인" 화면이 나타난다.

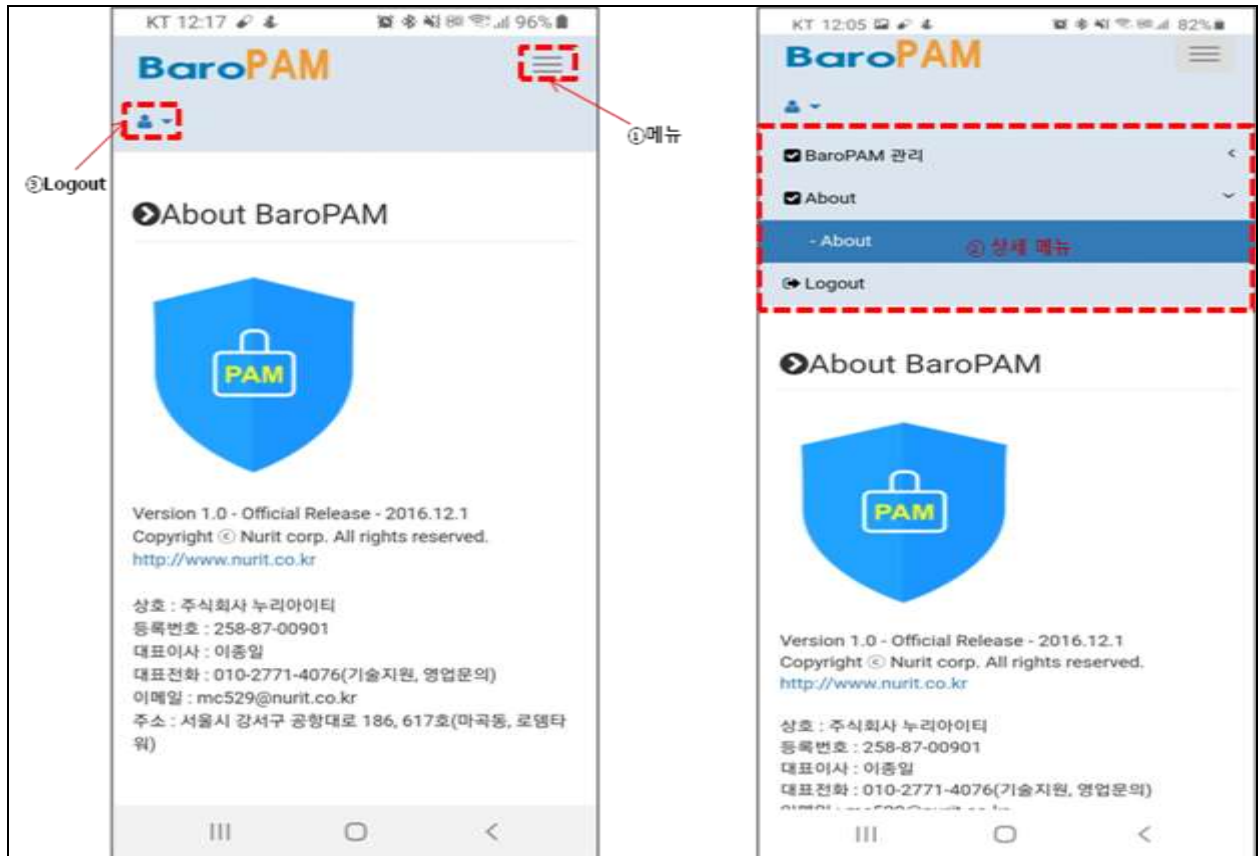
2.6 BaroPAM 웹 사용(접속로그관리)

1. 오른쪽 상단에 있는 "메뉴" 버튼을 클릭한 후 상세 메뉴에서 "접속로그 관리" 메뉴를 클릭하면 다음과 같이 "접속로그 목록" 화면이 나타난다.



2.7 BaroPAM 웹 사용(Abort & Logout)

1. 오른쪽 상단에 있는 "메뉴" 버튼을 클릭한 후 상세 메뉴에서 "About" 메뉴를 클릭하면 다음과 같이 "About" 화면이 나타난다.



"About"는 BaroPAM에 대한 대략적인 정보를 제공한다.

"① 메뉴" 버튼을 클릭하면 우측 화면처럼 선택할 수 있는 "② 상세 메뉴"가 나타나며, "③ Logout" 버튼을 클릭하면 "BaroPAM" 웹 사이트(www.baropam.com) 화면을 종료하고 "로그인" 화면이 나타난다.

3. About BaroPAM



Version 1.0 - Official Release - 2016.12.1
Copyright © Nurit corp. All rights reserved.
<http://www.nurit.co.kr>

제 조 사 : 주식회사 누리아이티
등록번호 : 258-87-00901
대표이사 : 이종일
대표전화 : 02-2665-0119(영업문의/기술지원)
이 메 일 : mc529@nurit.co.kr
주 소 : 서울시 강서구 마곡중앙2로 15, 913호(마곡동, 마곡테크노타워2)