

# BaroPAM User Guide(App/Web)

## Contents

Contents .....	<b>0</b>
1. BaroPAM app.....	<b>1</b>
1.1 Install BaroPAM app.....	1
1.2 Using the BaroPAM app(Menu selection).....	19
1.3 Using the BaroPAM app(Application infomation).....	20
1.4 Using the BaroPAM app(Identity verification).....	28
1.5 Using the BaroPAM app(Server infomation).....	42
1.6 Using the BaroPAM app(Environment settings).....	50
1.7 Using the BaroPAM app(Information sharing).....	68
1.8 BaroPAM App FAQ .....	71
1.9 BaroPAM Integration API .....	72
2. BaroPAM Website.....	<b>113</b>
2.1 Using the BaroPAM Web(Membership registration) .....	113
2.2 Using the BaroPAM Web(Login) .....	118
2.3 Using the BaroPAM Web(Application information management) .....	120
2.4 Using the BaroPAM Web(Server information management) .....	129
2.5 Using the BaroPAM Web(PIN information management).....	140
2.6 Using the BaroPAM Web(Access log management).....	143
2.7 Using the BaroPAM Web(Abort & Logout).....	144
3. About BaroPAM .....	<b>145</b>

## 1. BaroPAM app

Download for BaroPAM app's Android phone (<https://play.google.com/store/apps/details?id=com.baropam>), which is a generator of disposable authentication key to be entered in "Verification code" when logging in to information assets. In Google's "Play Store", the iPhone is available in Apple's "App store", and the installation is the same as that of a normal app.

The BaroPAM app is available for Android 6.0 (Marshmallow) API 23, iOS 13.0 or higher, and does not support landscape mode.

If you cannot download and install the "BaroPAM" app for Android from Google's "Play Store", you must download the apk from the following address and install the "BaroPAM" app manually.

Korean version: [http://nuriapp.com/apk/app\\_baropam.apk](http://nuriapp.com/apk/app_baropam.apk)

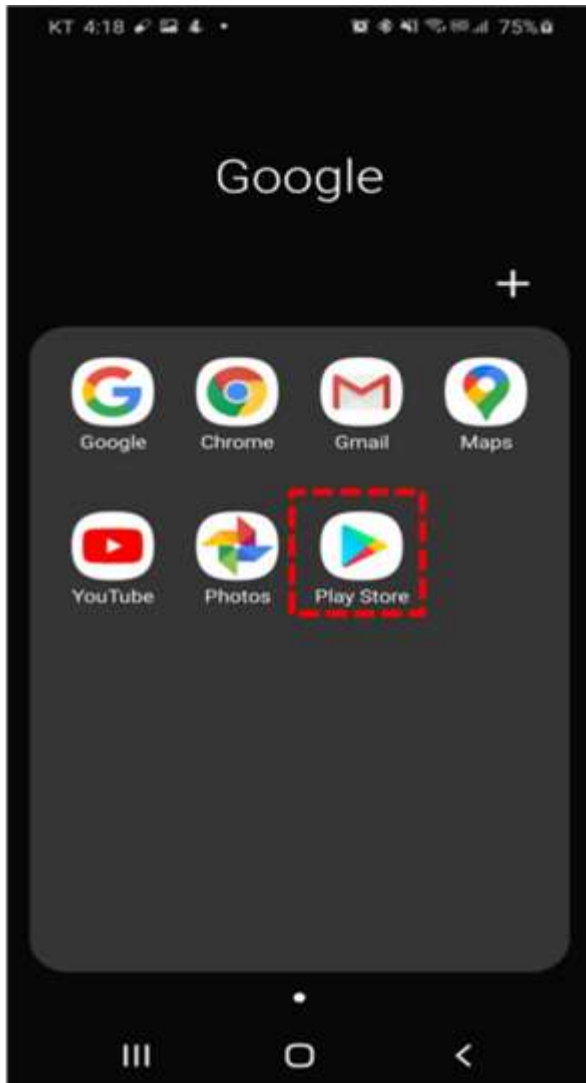
English version: [http://nuriapp.com/apk/app\\_baropams.apk](http://nuriapp.com/apk/app_baropams.apk)

### 1.1 Install BaroPAM app

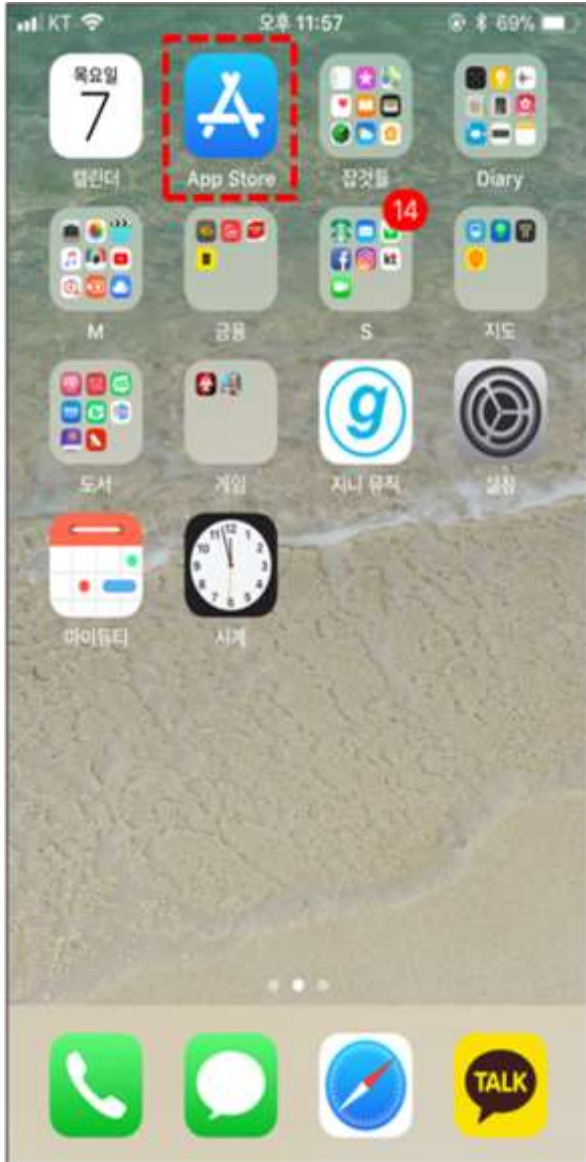
Downloading and installing the "BaroPAM" app proceeds as follows.

1. Click the "Pay Store" or "App Store" icon on your smartphone.

For Android phones)

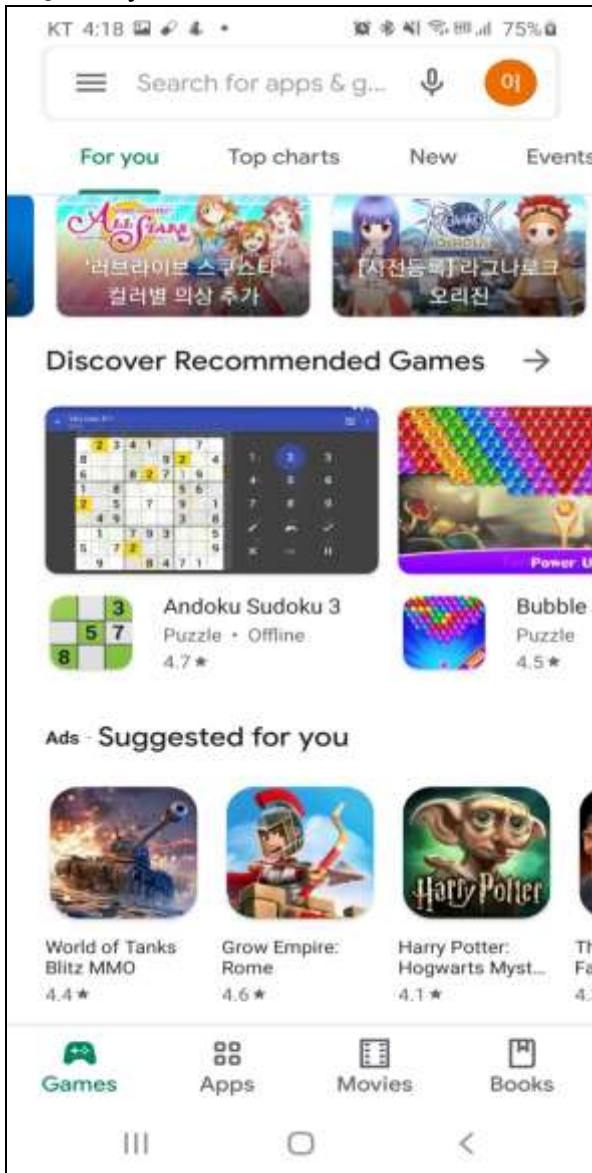


For iPhones)



2. Google "Pay Store" or Apple "App Store" is activated as follows.

Google "Pay Store")

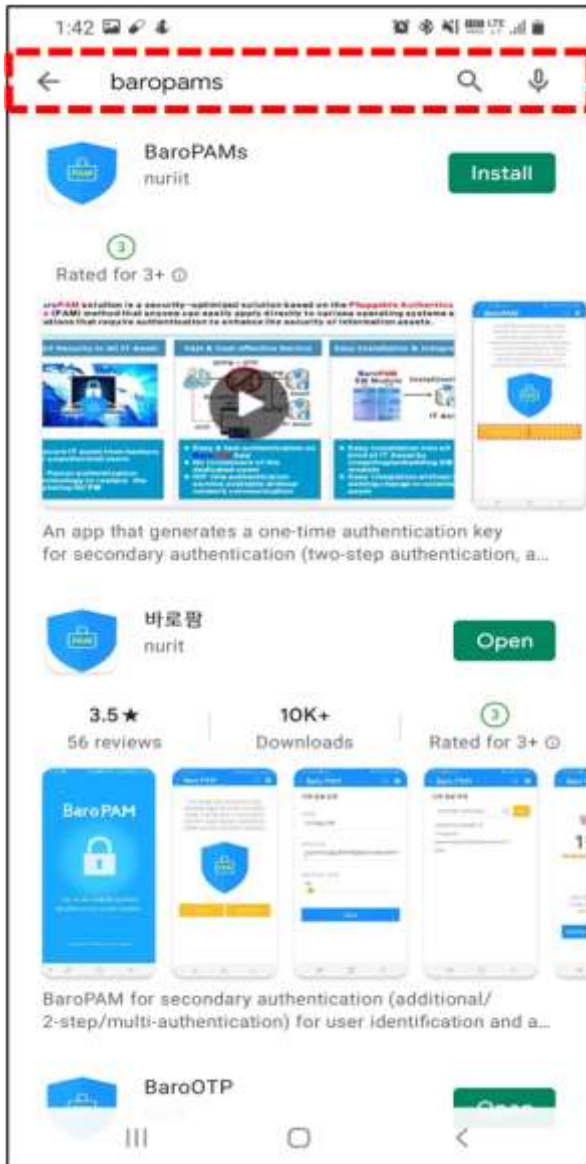


Apple "App Store")



3. Enter "baropams" in the search field at the top and search for "BaroPAMs" app.

Google "Pay Store")

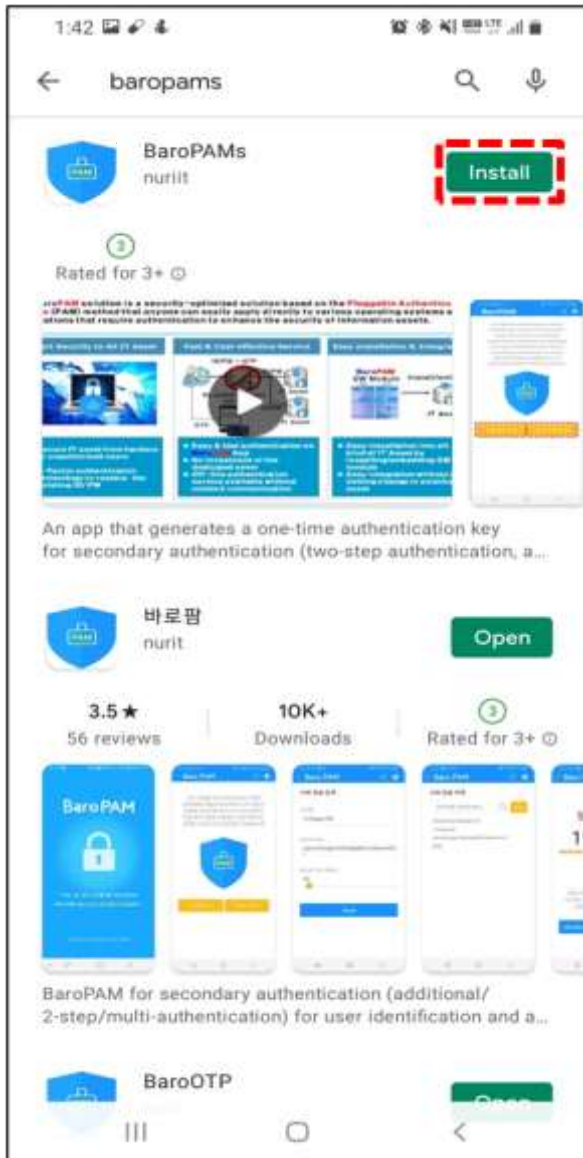


Apple "App Store")



4. After checking the information of the "BaroPAMs" app, click the "Install" or "Receive" button.

Google "Pay Store")



Apple "App Store")

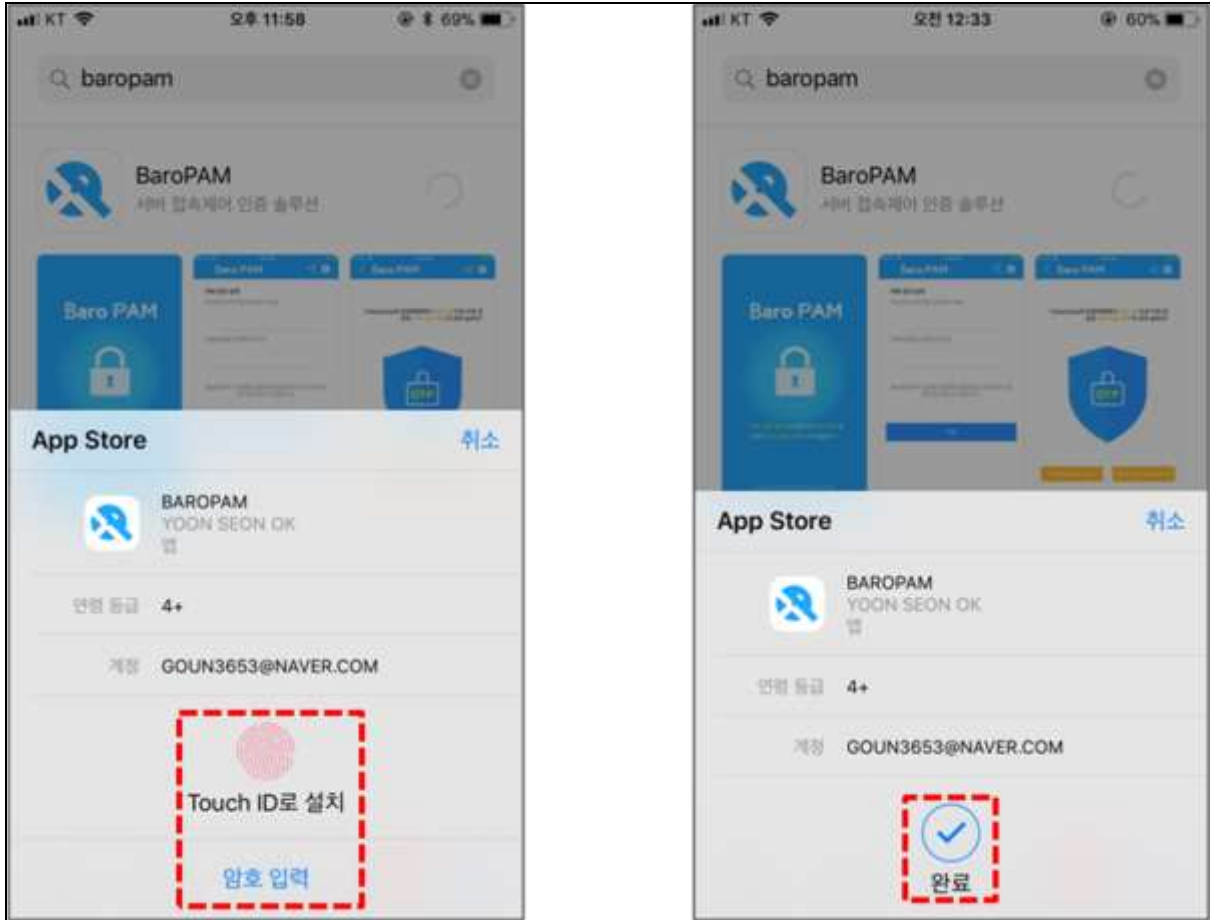


5. Download and install the "BaroPAMs" app as follows.

Google "Pay Store")



In the case of the Apple "App Store", the installation proceeds after completing the following.



6. When the installation of the "BaroPAMs" app is completed normally, the following "BaroPAM, BaroPAMs" icon is added to the smartphone.

Google "Pay Store")



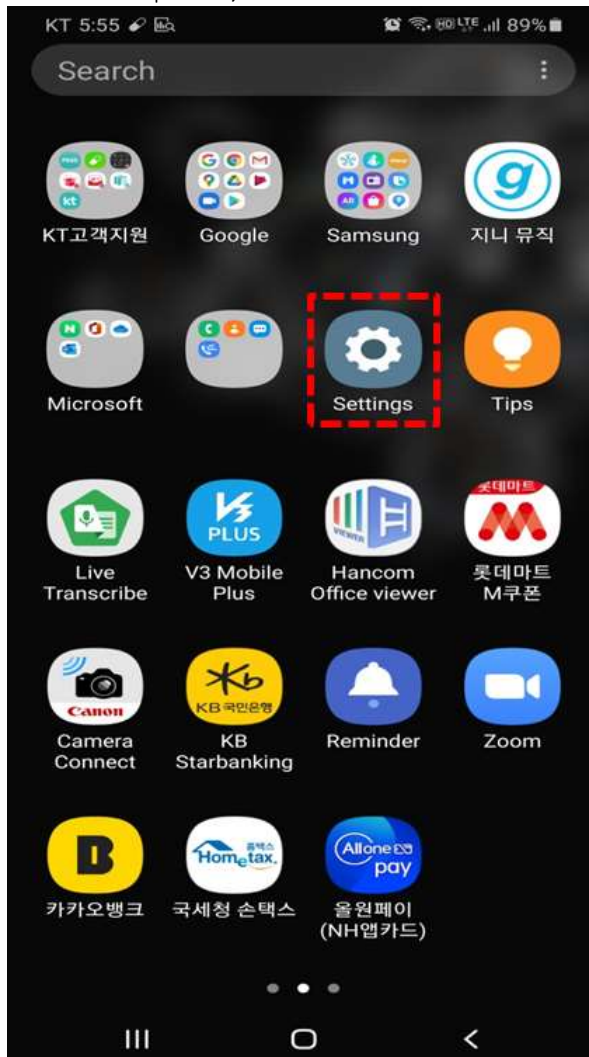
Apple "App Store")



After installing the "BaroPAM" app, you must set up a more convenient **password auto-completion** function from iOS12 (Go to "Settings" -> "Passwords" -> "AutoFill Passwords" of the iPhone and allow "BaroPAM").

7. For Android phones, click the "Settings" icon as follows to set the permissions for the "BaroPAM" app. (In case of iPhone, steps 8~11 do not need to be performed.)

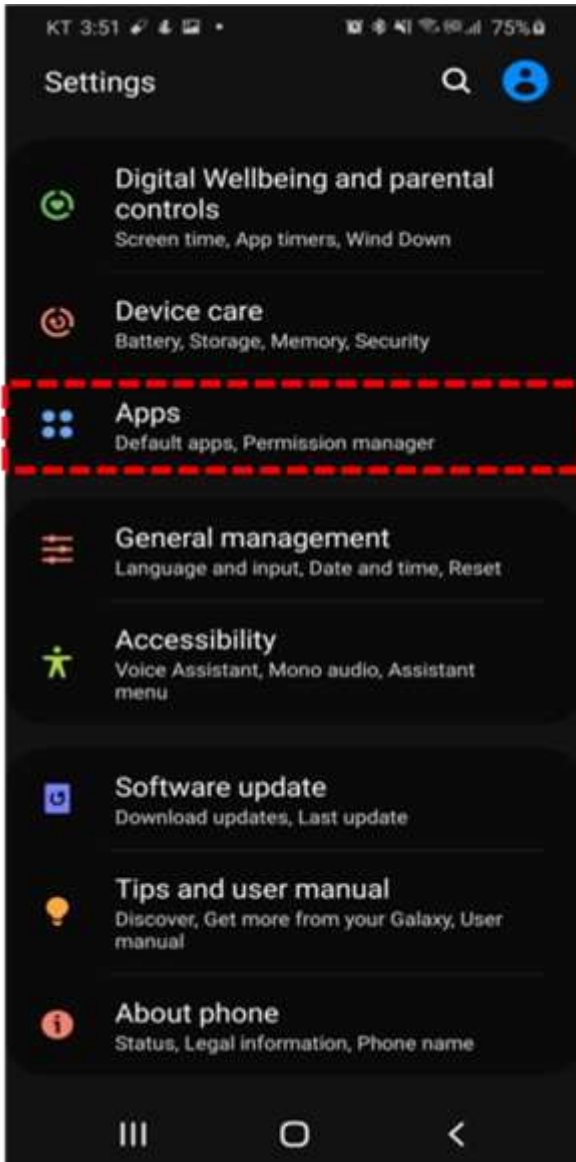
For Android phones)



In the case of an Android phone, a **One-Time Authentication key(OTA key)** is not generated unless the permission is set.

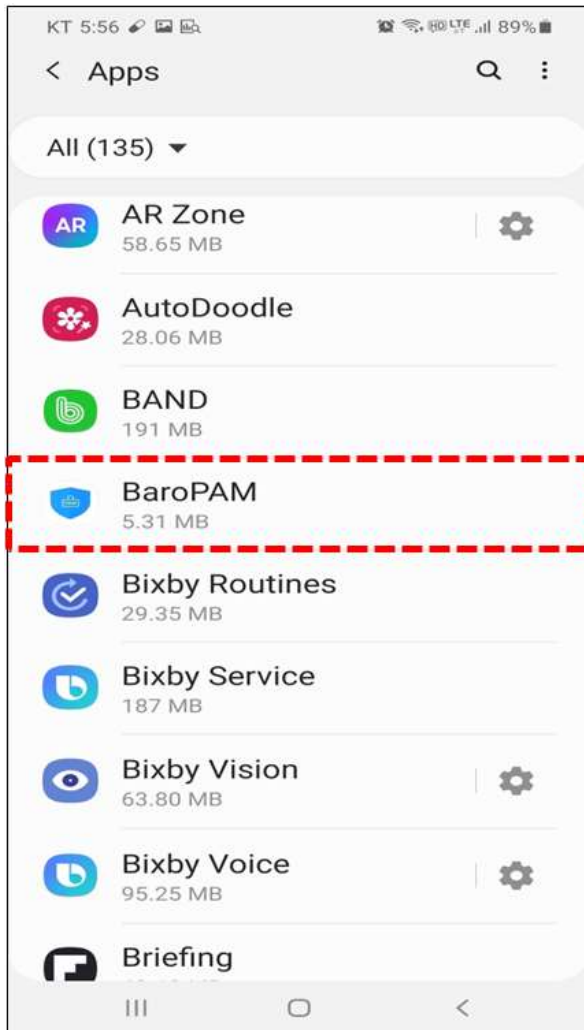
8. Among the icons on the "Settings" screen, click the "Apps" icon as follows.

For Android phones)



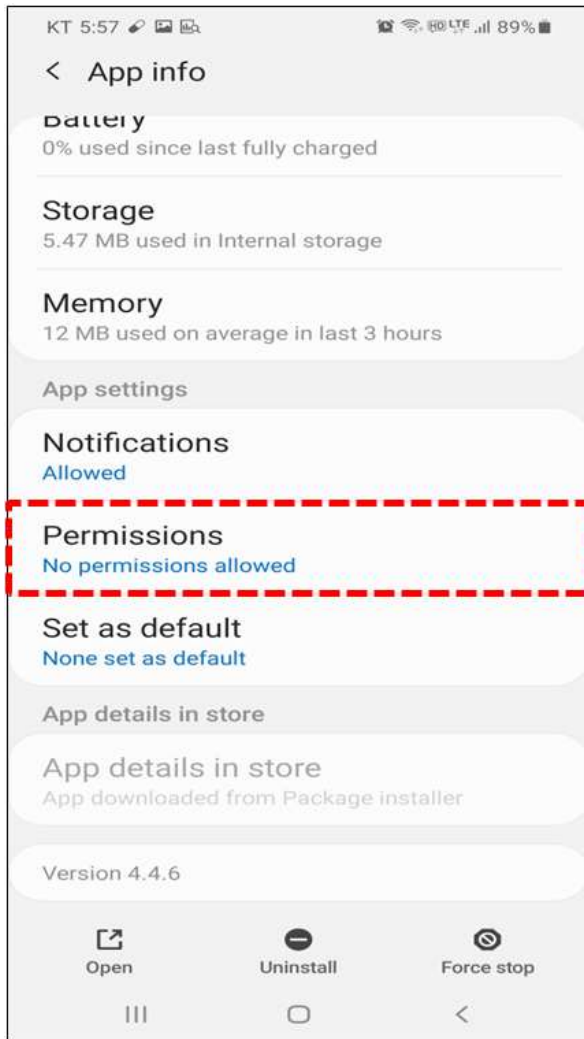
9. On the "Apps" screen, click the "BaroPAM" icon as follows.

For Android phones)



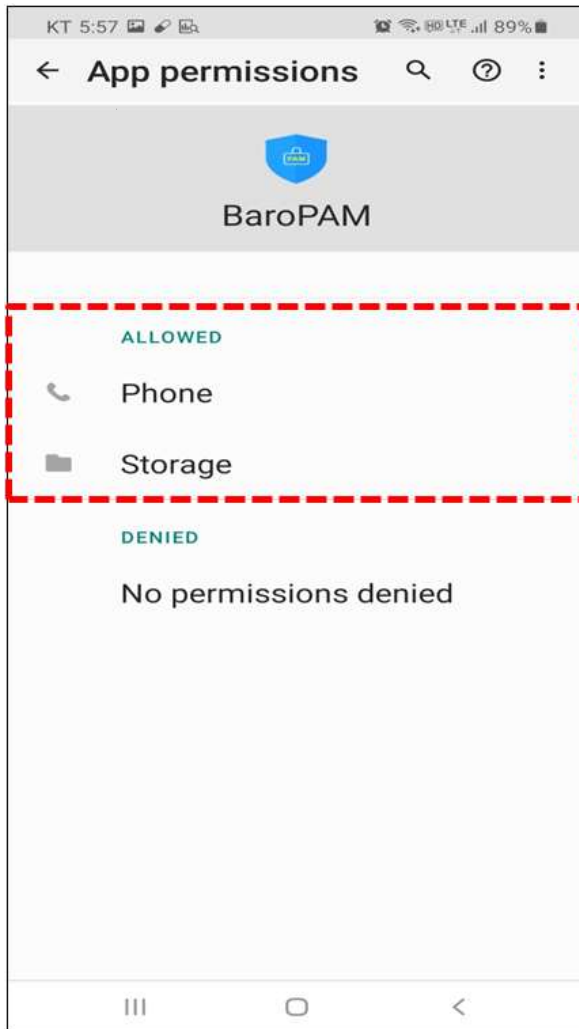
10. If you select "Permissions" on the "App info" screen, a screen to set "App Permissions" appears.

For Android phones)



11. On the "App Permissions" screen, change "App Permissions" from "Storage" and "Phone" from off to on.

For Android phones)



Since the "BaroPAM" app uses the time synchronization method, the time setting of the phone is very important, so the time should be set automatically as follows.

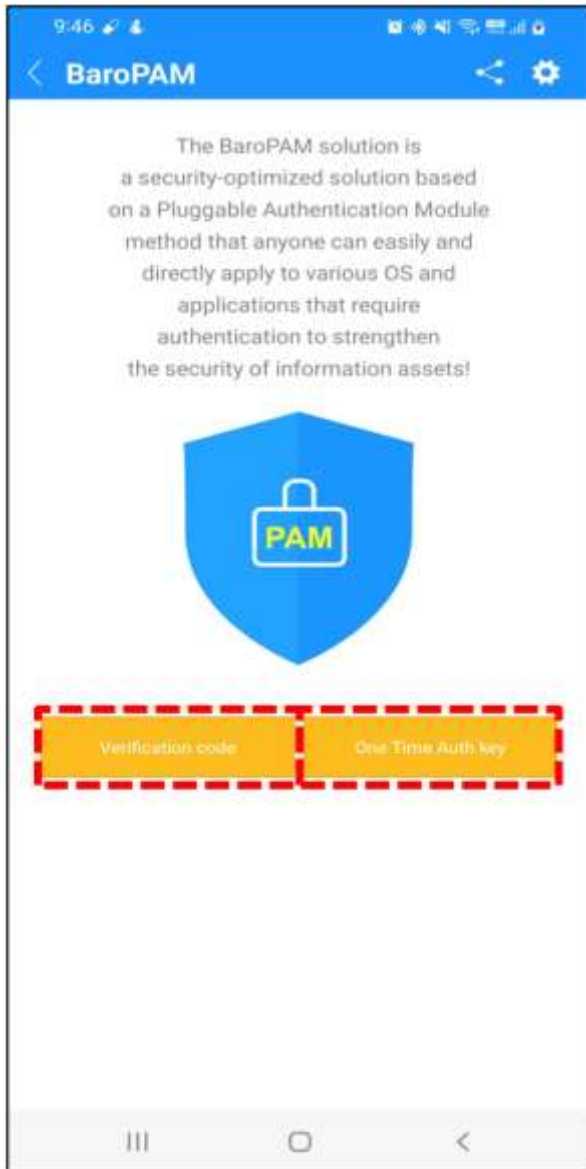
For Android phones: "Settings"-"Date and time"-"Automatic date and time" and "Automatic time zone Auto"-"Allow".

For iPhones: "Settings"-"Date and time"-"Automatic date and time"-"Automatically set"-"Allow".

"BaroPAM" app is largely composed of PIN information, login, server information, login information, environment setting, and information sharing.

## 1.2 Using the BaroPAM app(Menu selection)

1. When the "BaroPAM" app is activated, a screen to select "Verification code" and "One-Time Auth Key" appears as follows.

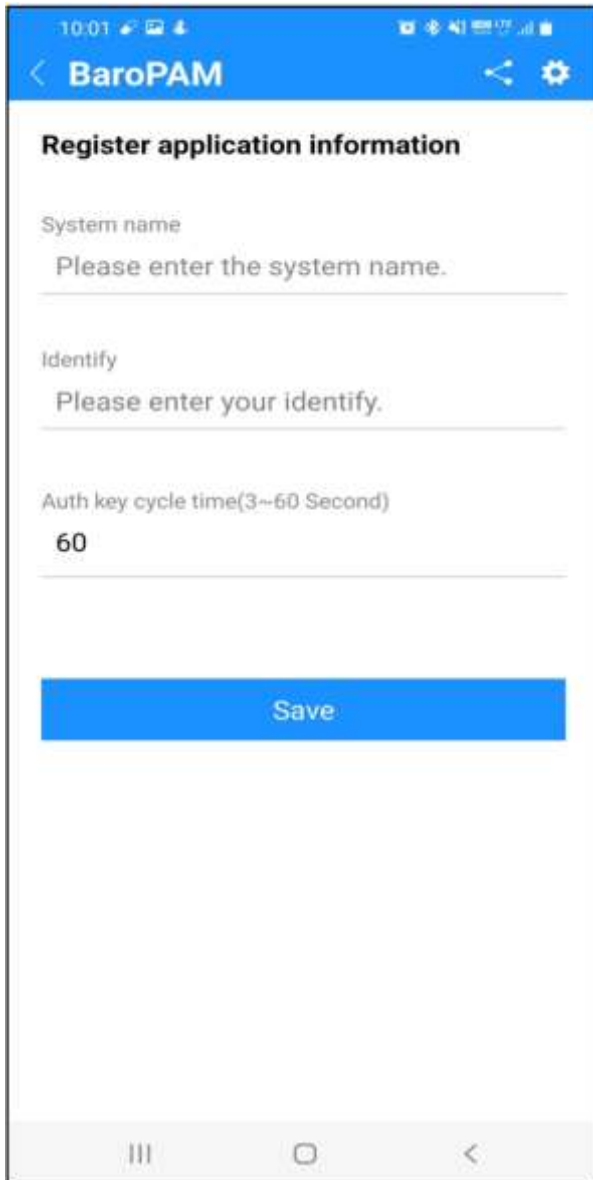


"Verification code" is used when generating a **OTA key** used to access information assets, and "One Time Auth key" is used to replace the password of the application login screen with a **OTA key**.

First, when the "One-Time Auth key" button is clicked, the "List of application information" screen appears when there is application login information, and the "Register application information" screen appears when there is no application login information.

### 1.3 Using the BaroPAM app(Application information)

1. When the "One-Time Auth key" button is clicked on the "Menu selection" screen If the login information of the application exists, the "List of application information" screen appears. If the login information of the application does not exist, the "Register application Information Registration" screen appears.



#### ■System name

The system name to log in is a required field. You can enter at least 1 digit and up to 30 digits.

#### ■Identify

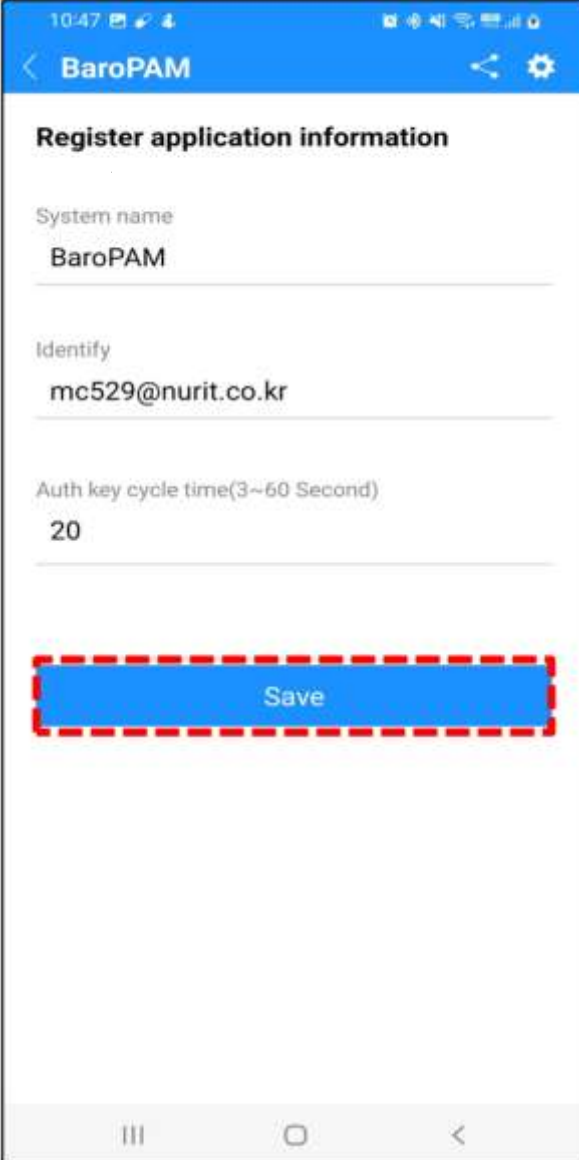
Identify is a required field and you can enter at least 1 digit and up to 50 digits.

**Auth key cycle time(3-60 second)**

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

If the **OTA key** generation cycle (Auth key cycle time) of the **BaroPAM** verification module is different from the **OTA key** generation cycle (Auth key cycle time) specified in the app. Then, there may be a case where you cannot log in because the **OTA key** is different.

2. After entering the login information of the application, click the "Save" button.



When the **"Save"** button is clicked, the first thing to check the validity of the input items is to save the login information of the application.

If the system name is not entered, the message **"Please enter the system name."** appears on the screen.

If identify is not entered, the message **"Please enter your identify."** appears on the screen.

If the auth key cycle time is not entered or is out of range, the message **"Please enter or confirm the auth key cycle."** appears on the screen.

3. If the saving of login information of the application is not completed normally, the "Register application information" screen continues, and when it is completed normally, the "List of application information" screen appears as follows.



If you want to search the login information of the application, enter a search term and click the "Search" button, then the login information of the searched application will appear on the screen.

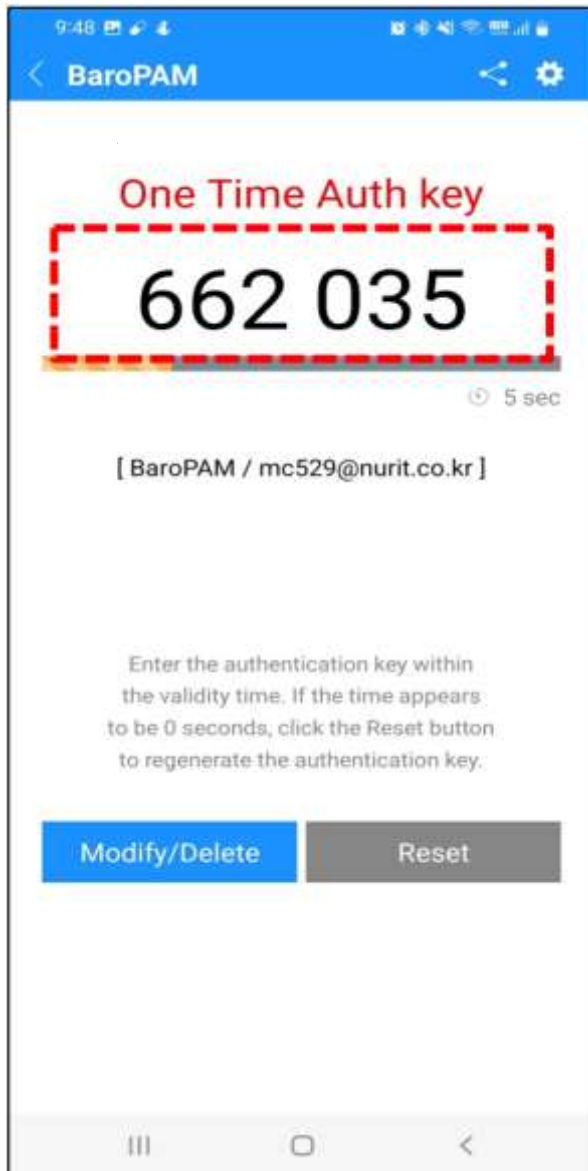
If you need to add new login information for the application, click the "New" button to display the application login information registration screen.

4. If you need to generate a **secondary authentication key(OTA key)**, click the application information item in the "List of application information" as follows.



In the "List of application information", click on the login information item of the application to generate a **secondary authentication key(OTA key)**. It is necessary to apply a separate identification function to prevent unauthorized use of another person's phone number only in the case of an iPhone because it is impossible to obtain an iPhone. The screen for executing the "1.4 Using the BaroPAM app(Identity verification)" is called.

5. Then, a screen for generating a **secondary authentication key(OTA key)** based on the login information of the application appears as follows.



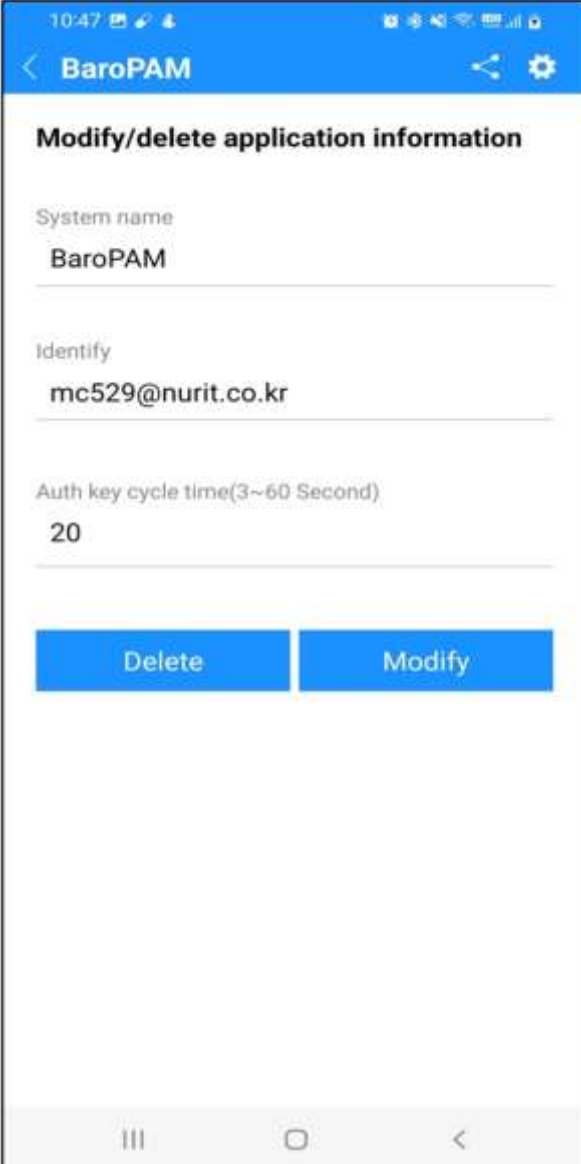
If you need to proceed with the modification/deletion of login information, click the "**Modify/Delete**" button.

During the same authentication key cycle time, the "**Reset**" button is deactivated, and when the authentication key cycle time passes, the "**Reset**" button is activated.

When the activated "**Reset**" button is clicked, a new **secondary authentication key (OTA key)** is generated, and the same secondary **authentication key (OTA key)** is generated during the one-time authentication key generation cycle (Auth key cycle time).

Note) If you click the number "**662 035**", which is the generated **OTA key**, the **OTA key** is regenerated in the same way as clicking the "**Reset**" button.

6. If you need to modify(Modify) or delete(Delete) the login information of the application, click the "Modify/Delete" button and the following screen appears.



The screenshot shows a mobile application interface with a blue header bar containing the text "BaroPAM" and navigation icons. Below the header, the title "Modify/delete application information" is displayed. The form contains three input fields: "System name" with the value "BaroPAM", "Identify" with the value "mc529@nurit.co.kr", and "Auth key cycle time(3~60 Second)" with the value "20". At the bottom of the form, there are two blue buttons labeled "Delete" and "Modify". The bottom of the screen shows the standard Android navigation bar with three icons: a square, a circle, and a triangle.

■System name

The system name to log in is a required field. You can enter at least 1 digit and up to 30 digits.

■Identify

Identify is a required field and you can enter at least 1 digit and up to 50 digits.

■Auth key cycle time(3-60 second)

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

If the **OTA key** generation cycle (Auth key cycle time) of the **BaroPAM** verification module and the **OTA key** generation cycle (Auth key cycle time) specified in the app are different, the application

cannot be logged in because the **OTA key** is different may occur.

Click the "**Delete**" button to delete the login information of the application.

If you click the "**Modify**" button, the first thing to do is to check the validity of the input items and then modify the login information of the application.

If the system name is not entered, the message "**Please enter the system name.**" appears on the screen.

If Identify is not entered, the message "**Please enter your identify.**" appears on the screen.

If the auth key cycle time is not entered or is out of range, the message "**Please enter or confirm the auth key cycle.**" appears on the screen.

If the application's login information modification (Modify) or deletion (Delete) is not completed normally, the "**Modify/Delete application information**" screen continues, and when it is completed normally, the "**List of application information**" screen appears.

## 1.4 Using the BaroPAM app(Identity verification)

### 1. For iPhone

Unlike Android phones, when you select the login information item of the application to generate a **secondary authentication key(OTA key)** on the "List of application information" screen because the device information of the iPhone phone cannot be obtained, a "One-Time Auth key" is generated. There may be cases where it does not move to the screen.

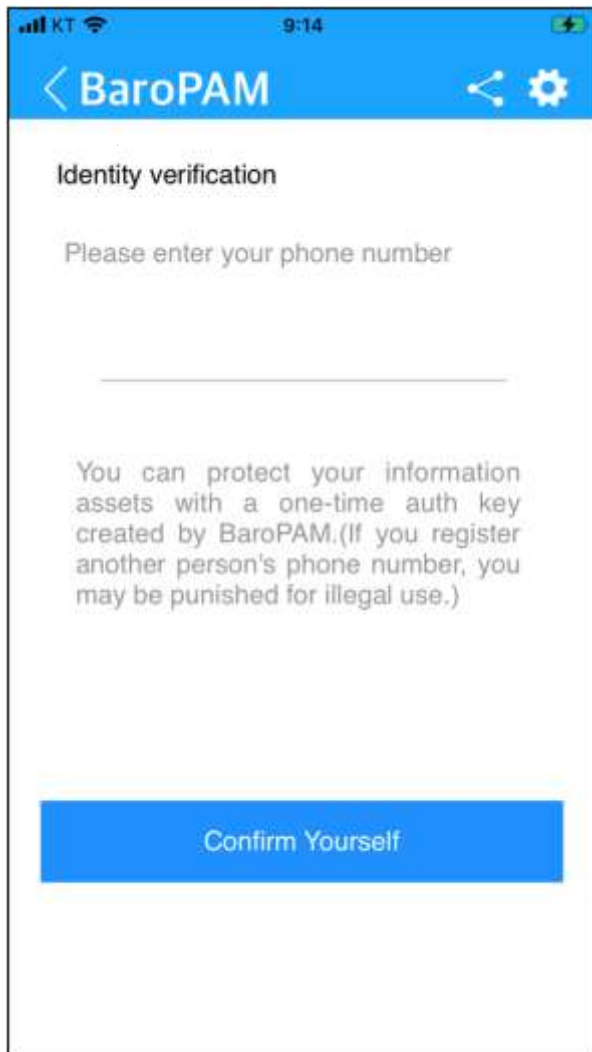
In addition, in order to prevent unauthorized use of other people's phone numbers, it is necessary to apply a separate identity verification function. In the "BaroPAM" app, its own algorithm is applied and the identity verification procedure is executed as follows.

Unlike Android phones, the iPhone cannot obtain device information due to its own security, so it is necessary to apply a separate identity verification function to prevent unauthorized use of other people's phone numbers only in the case of iPhone. The "BaroPAM" app uses its own algorithm. By applying, the identity verification process is carried out as follows.

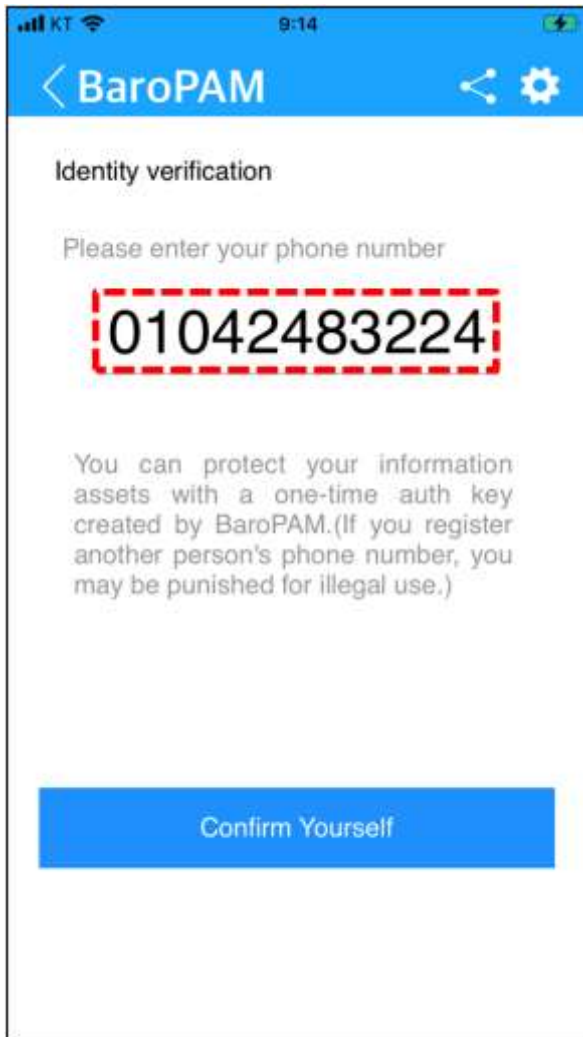
1) In "List of application information", click the application information item to generate a **secondary authentication key(OTA key)**.



2) If you have not performed the identification process, the following screen for identification appears.



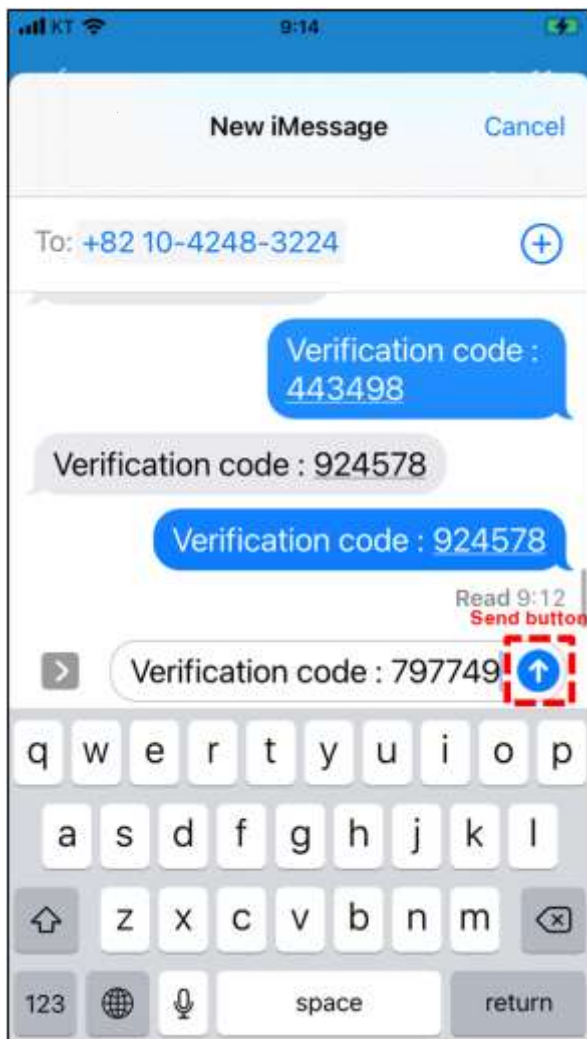
3) Enter the iPhone's phone number as a number and click the "Confirm Yourself" button.



If you click the "Confirm Yourself" button, you first check the validity of the input items and then proceed with the verification work for your phone number.

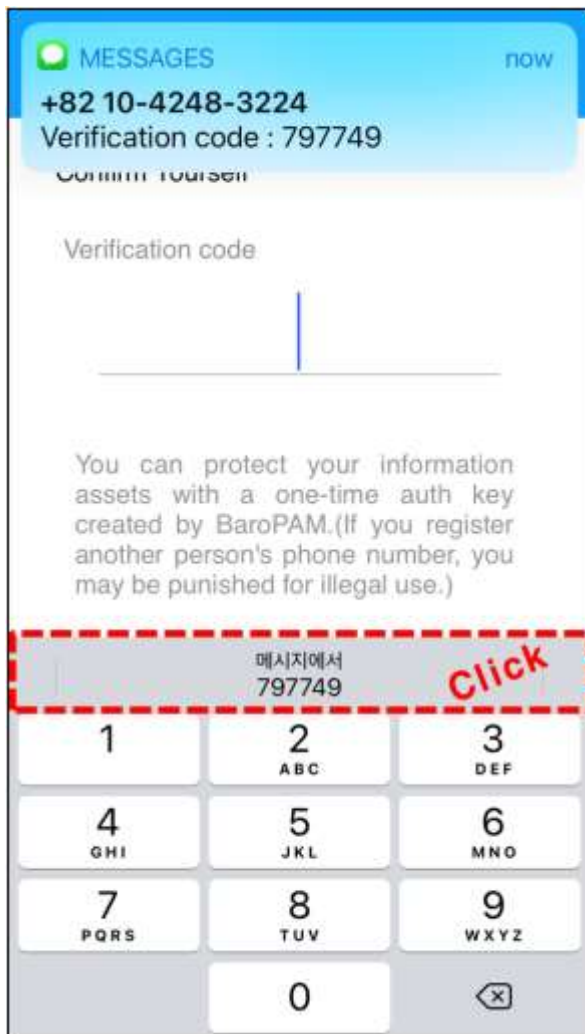
If the phone number is less than 8 digits, the message "Please enter more than 8 digits of phone number." appears on the screen.

4) When it is completed normally, the **OTA key** is generated by an internal algorithm to check whether it is the user's phone number, and then click the "Send SMS" button on the next screen to transmit the generated **OTA key** to the input phone number.



If you enter someone else's phone number, you may be punished for illegal use.

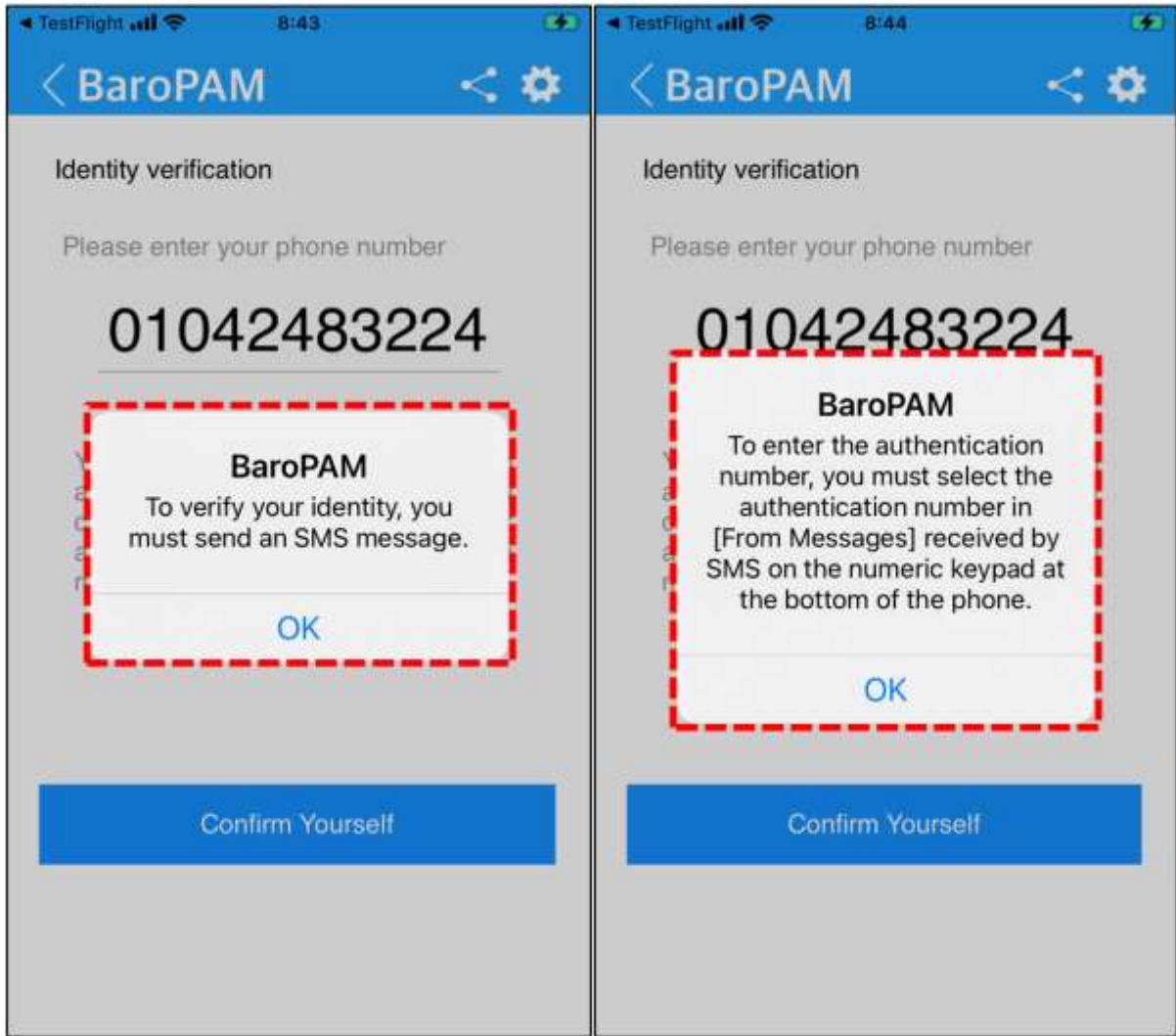
5) Then, the **OTA key** sent by **SMS** is displayed on the keyboard at the bottom as follows.



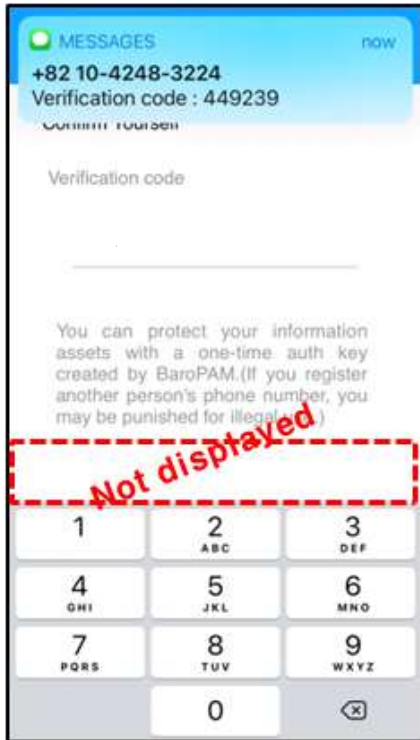
If you click the **OTA key** displayed on the keyboard, the internal **OTA key** verification module proceeds to store the phone number after verification.

If the **OTA key** is not displayed on the keyboard like the screen above, be sure to set up a more convenient password auto-completion function from iOS12 (Go to iPhone's "**Settings**" -> "**Passwords**" -> "**AutoFill Passwords**" -> "**Allow**") Confirmation should be done.

If you enter **OTA key** directly without sending **SMS**, "To verify your identity, you must send an **SMS message.**" or "To enter the authentication number, you must select the authentication number in [From Message] received by **SMS** on the numeric keypad at the bottom of the phone." appears on the screen.

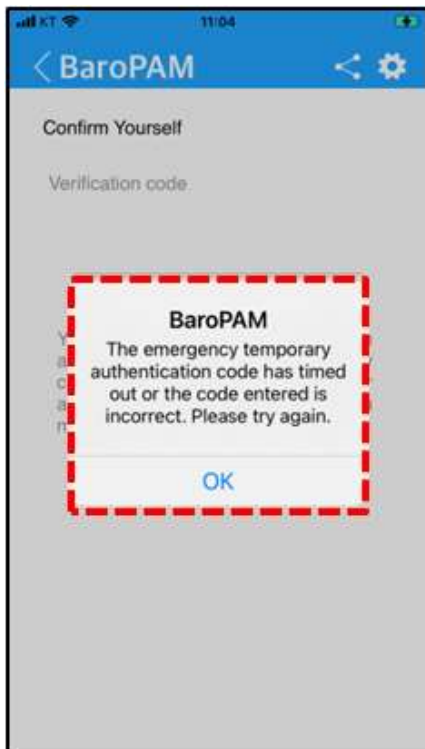


Note) If the **OTA key** sent by **SMS** is received but it is not displayed on the keyboard or the **OTA key** sent by **SMS** is not received.

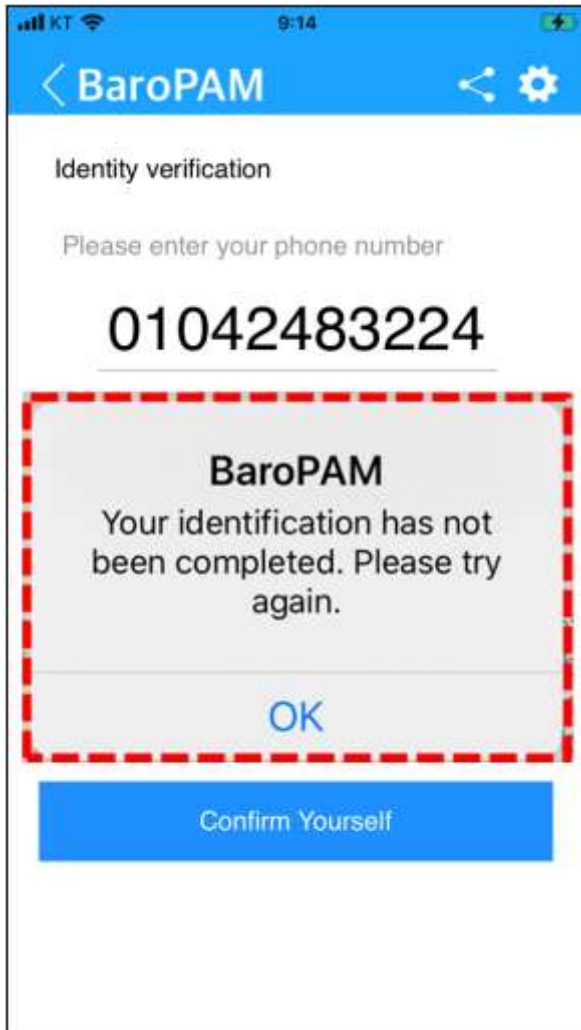


If the **OTA key** is not displayed on the keyboard like the screen above, be sure to set up a more convenient password auto-completion function from iOS12 (Go to iPhone's "**Settings**" -> "**Passwords**" -> "**AutoFill Passwords**" -> "**Allow**") Confirmation should be done.

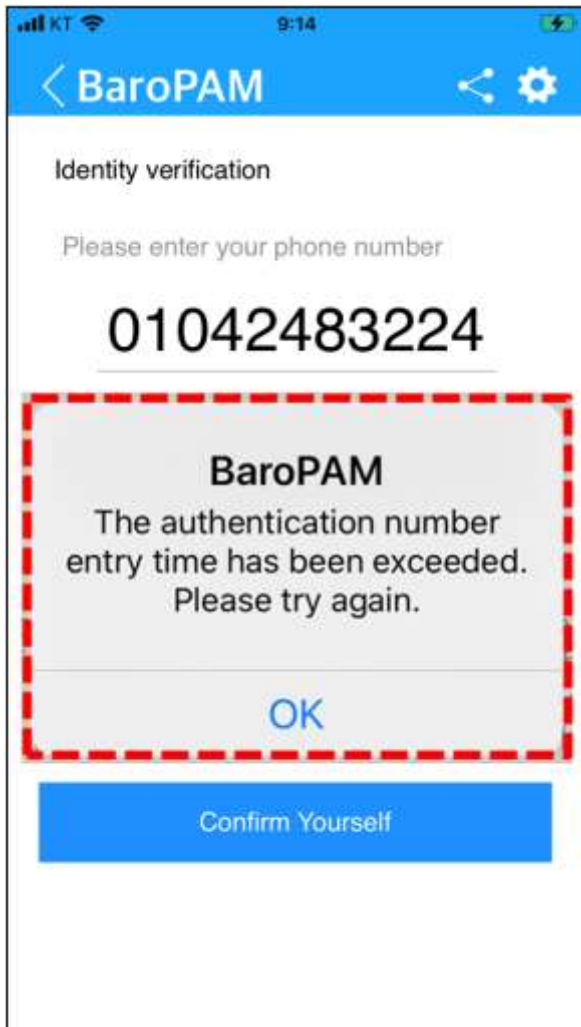
If the urgently issued temporary verification code verification fails, the following message appears on the screen, and urgently issued temporary verification code verification must be performed again.



6) If the identity verification process fails, the following message appears on the screen, and the identity verification process must be performed again.



7) If the **OTA key** input time **exceeds 10 seconds** during the identity verification process, the following message appears on the screen, and the identity verification process must be performed again.

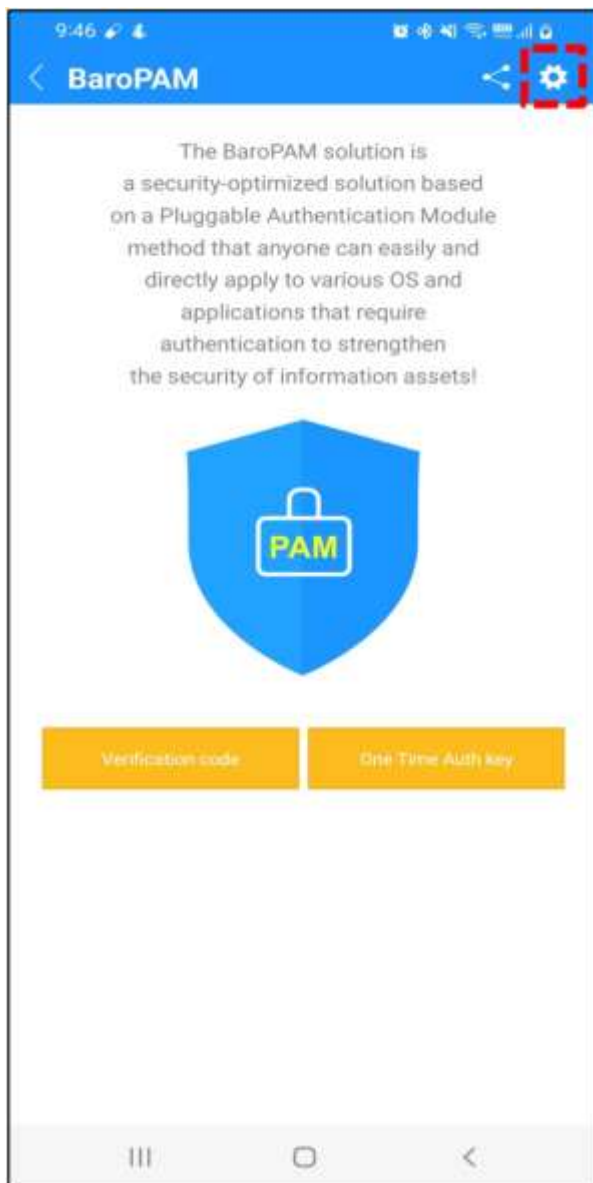


## 2. For Android phone

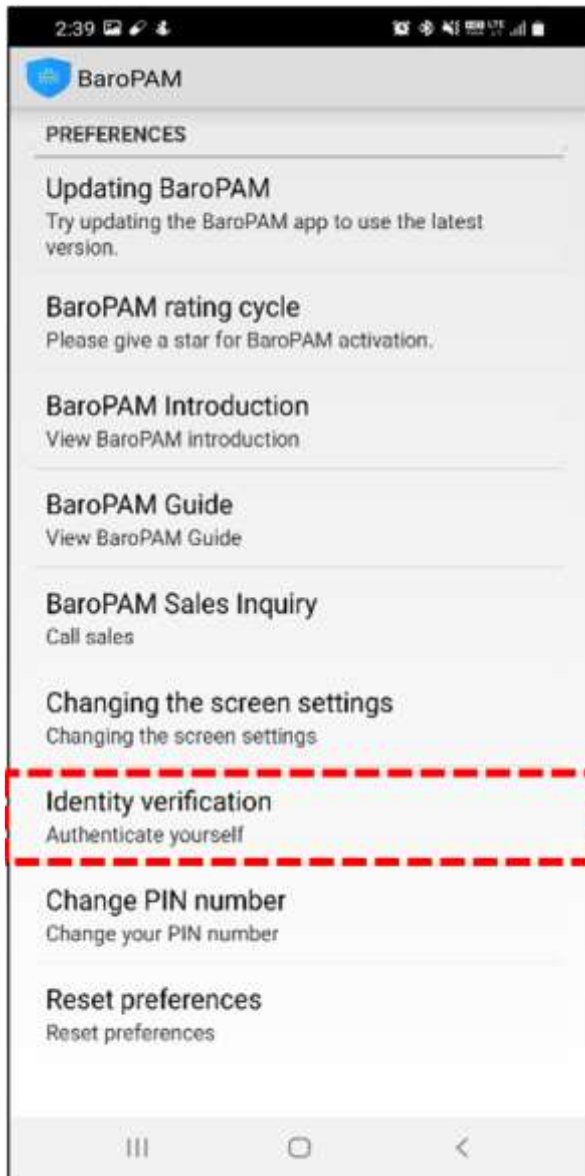
Unlike Android phones, when you select the login information item of the application to generate a **secondary authentication key(OTA key)** on the "List of application information" screen because the device information of the iPhone phone cannot be obtained, a "One-Time Auth key" is generated. There may be cases where it does not move to the screen.

In addition, it is necessary to apply a separate identity verification function in order to prevent unauthorized use of another person's phone number. The "BaroPAM" app applies its own algorithm and performs the identity verification procedure as follows.

1) To configure the "BaroPAM" app, click the "Preferences" icon as follows.



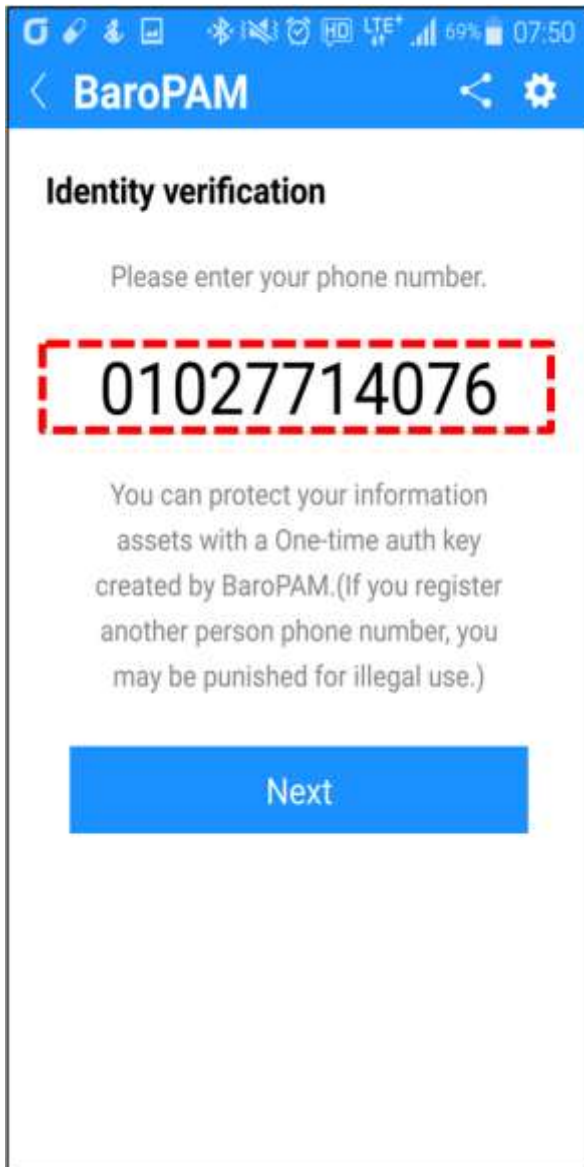
2) Then the "Preferences" screen appears as follows.



3) If you click the "Identity verification" item, the following "Identity verification" screen appears.



4) Enter your phone number as a number and click the "Next" button.

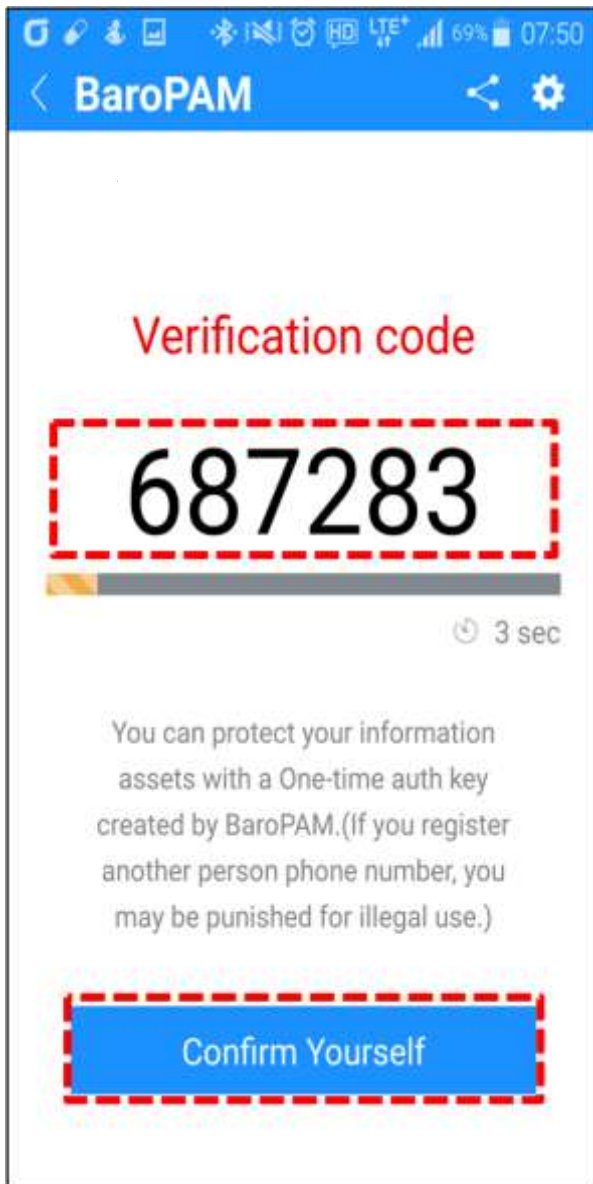


Clicking the "Next" button first checks the validity of the input items, and then delivers the **verification code** generated by the internal algorithm to the "Identity Verification" screen to proceed with the identity verification process.

If the phone number is less than 8 digits, the message "Please enter more than 8 digits of phone number." appears on the screen.

If you enter someone else's phone number, you may be punished for illegal use.

5) After checking the received **verification code**, click the "Confirm Yourself" button.

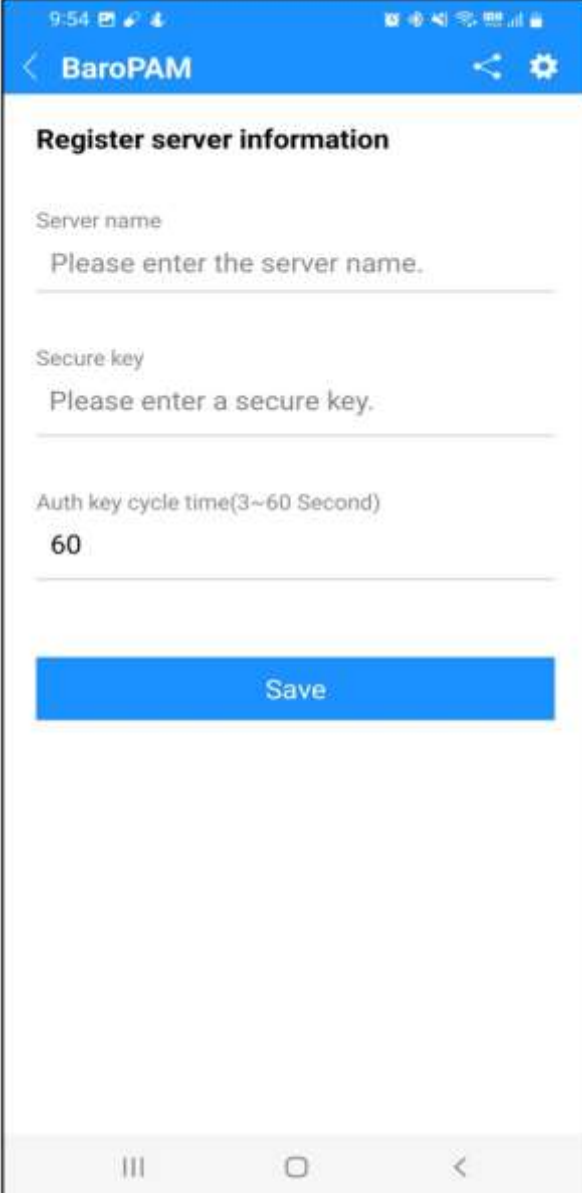


After checking the **verification code**, if you click the "Confirm Yourself" button, the **verification code** is verified by an internal algorithm and then the phone number is stored.

If the identity verification process is successful, it moves to the "Menu selection" screen, but if it fails, it moves to the "Identity verification" screen.

## 1.5 Using the BaroPAM app(Server information)

1. If the information on the information asset is not set (when using the "BaroPAM, BaroPAMs" app for the first time), the screen to register the information on the information asset appears as follows.



### ■Server name

The server name on which BaroPAM will be operated is a required field and can be entered at least 1 digit or up to 30 digits.

### ■Secure key

Secure key given by information asset or account is a required input item, and you must enter the

one requested by the vendor.

If you enter a random "Secure key" that has not been granted by the vendor, an incorrect OTA key is given and you may not be able to log in to the information asset.

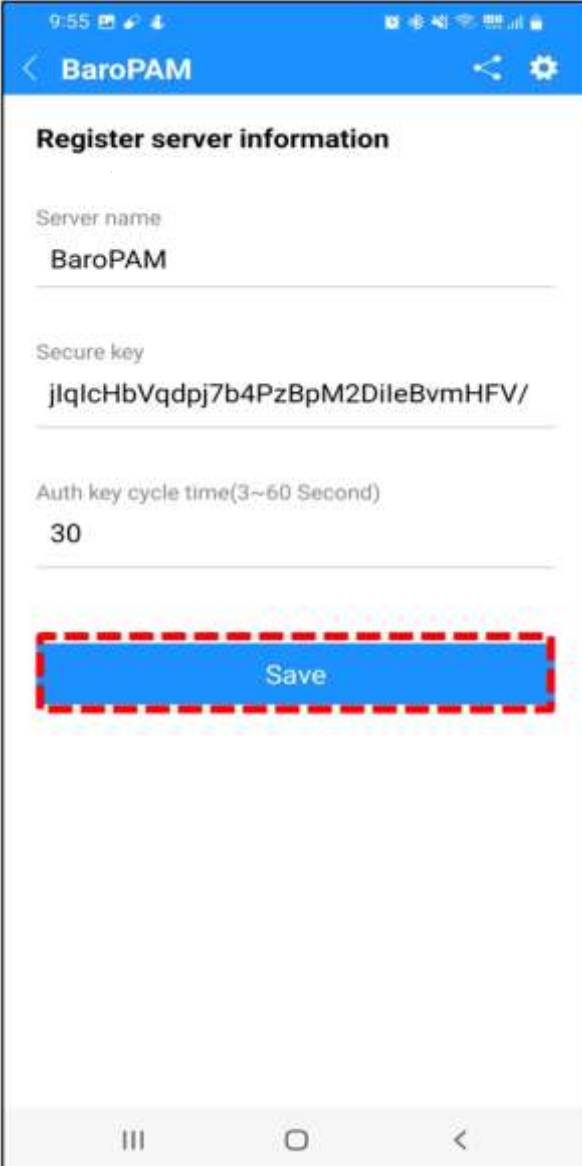
If the secure key set in the information asset and the secure key designated by the app are different, the OTA key is different and you may not be able to login.

#### ■Auth key cycle time(3-60 second)

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

If the OTA key generation cycle (Auth key cycle time) set in the information asset and the OTA key generation cycle (Auth key cycle time) specified in the app are different, the OTA key is different. So there may be cases where you cannot log in.

2. After entering information about the information asset, click the "Save" button.



Clicking the **"Save"** button first checks the validity of the input items and then proceeds to save the information on the information asset.

If the information asset name (Server name) is not entered, the message **"Please enter the server name."** appears on the screen.

If a secure key is not entered, the message **"Please enter a secure key"** appears on the screen.

If the auth key cycle time is not entered or is out of range, the message **"Please enter or confirm the auth key cycle"** appears on the screen.

3. If the information on the information asset is not successfully saved, the "Register server information" screen continues, and when completed normally, the "List of server information" screen appears as follows.



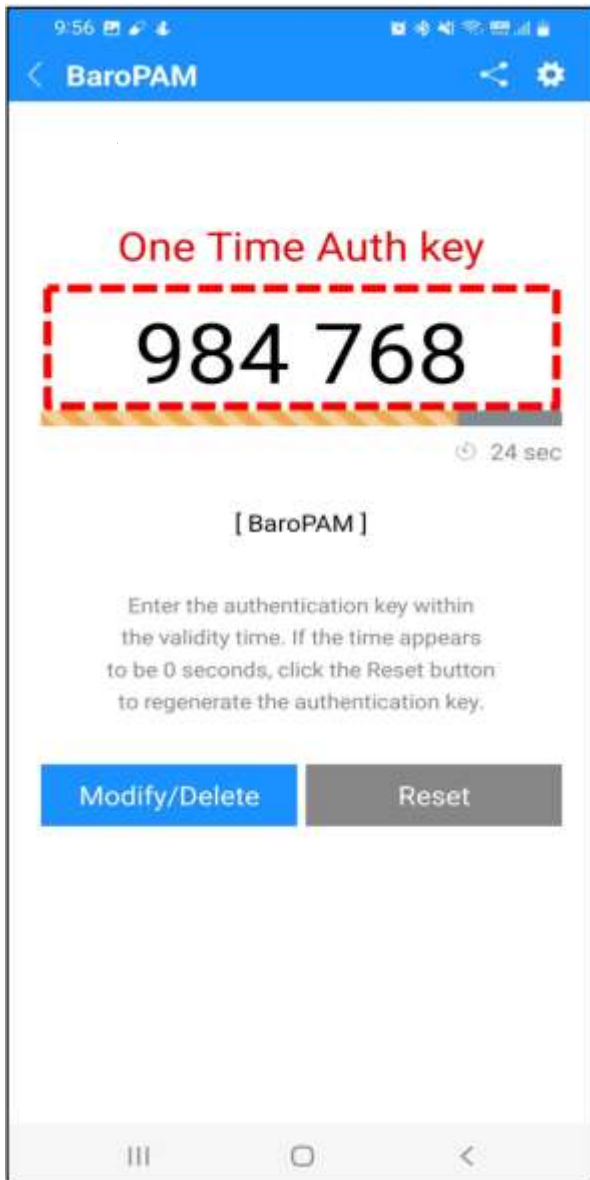
If you want to search for information on an information asset, enter a search term and click the "Search" button to display the information on the searched information asset on the screen.

If you need to add new information about information assets, click the "New" button and a screen to register information about information assets appears.

4. If you need to generate a **secondary authentication key(OTA key)**, click the information item for the information asset in the server information list as follows.



5. Then, a screen for creating a **secondary authentication key (OTA key)** based on the information on the information asset appears as follows.



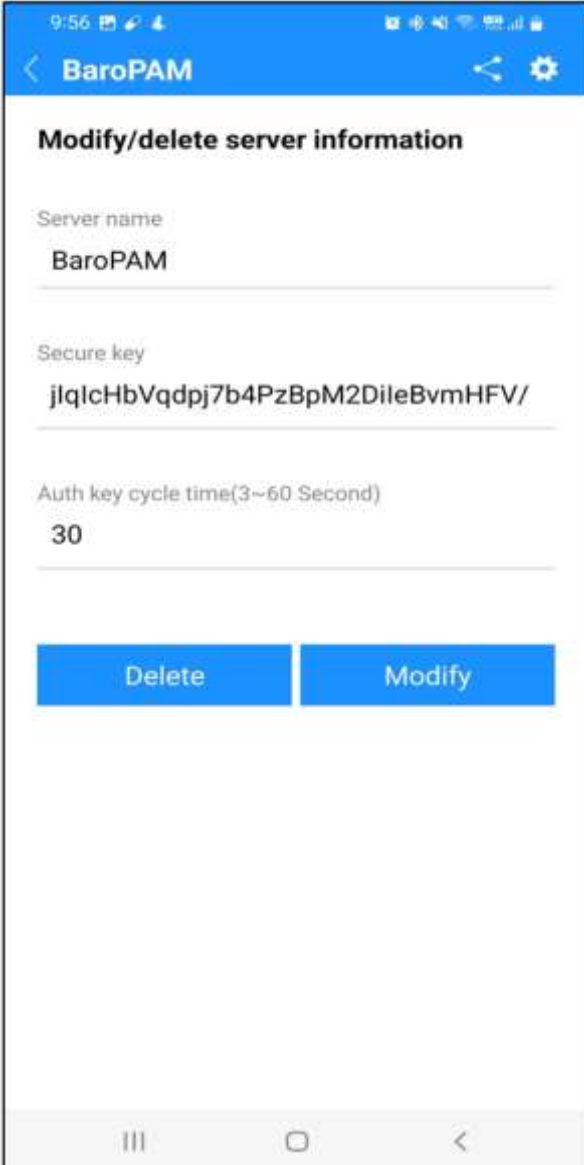
If you need to modify/delete information on information assets, click the "**Modify/Delete**" button.

During the same authentication key cycle time, the "**Reset**" button is deactivated, and when the authentication key cycle time passes, the "**Reset**" button is activated.

When the activated "**Reset**" button is clicked, a new **secondary authentication key (OTA key)** is generated, and the same **secondary authentication key (OTA key)** is generated during the one-time authentication key generation cycle (Auth key cycle time).

Note) If you click the number "**984 768**", which is the generated **OTA key**, the **OTA key** is regenerated in the same way as clicking the "**Reset**" button.

6. If you need to modify or delete information on an information asset, click the "Modify/Delete" button and the following screen will appear.



The screenshot shows a mobile application interface for managing server information. The title bar is blue with a back arrow, the text 'BaroPAM', and a settings gear icon. The main content area is white and titled 'Modify/delete server information'. It contains three text input fields. The first is labeled 'Server name' and contains the text 'BaroPAM'. The second is labeled 'Secure key' and contains a long alphanumeric string 'jIqIcHbVqdpj7b4PzBpM2DileBvmHFV/'. The third is labeled 'Auth key cycle time(3~60 Second)' and contains the number '30'. Below the input fields are two blue buttons: 'Delete' on the left and 'Modify' on the right. The bottom of the screen shows a standard Android navigation bar with three icons: a square, a circle, and a triangle.

■Server name

The server name on which BaroPAM will be operated is a required field and can be entered at least 1 digit or up to 30 digits.

■Secure key

Secure key given by information asset or account is a required input item, and you must enter the one requested by the vendor.

If you enter a random "Secure key" that has not been granted by the vendor, an incorrect OTA key is given and you may not be able to log in to the information asset.

If the secure key set in the information asset and the secure key designated by the app are different, the OTA key is different and you may not be able to login.

**Auth key cycle time(3-60 second)**

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

If the **OTA key** generation cycle (Auth key cycle time) set in the information asset and the **OTA key** generation cycle (Auth key cycle time) specified in the app are different, the **OTA key** is different. So there may be cases where you cannot log in.

Click the **"Delete"** button to delete the information on the information asset.

Clicking the **"Modify"** button first checks the validity of the input items and then proceeds to modify the information on the information asset.

If the information asset name (Server name) is not entered, the message **"Please enter the server name."** appears on the screen.

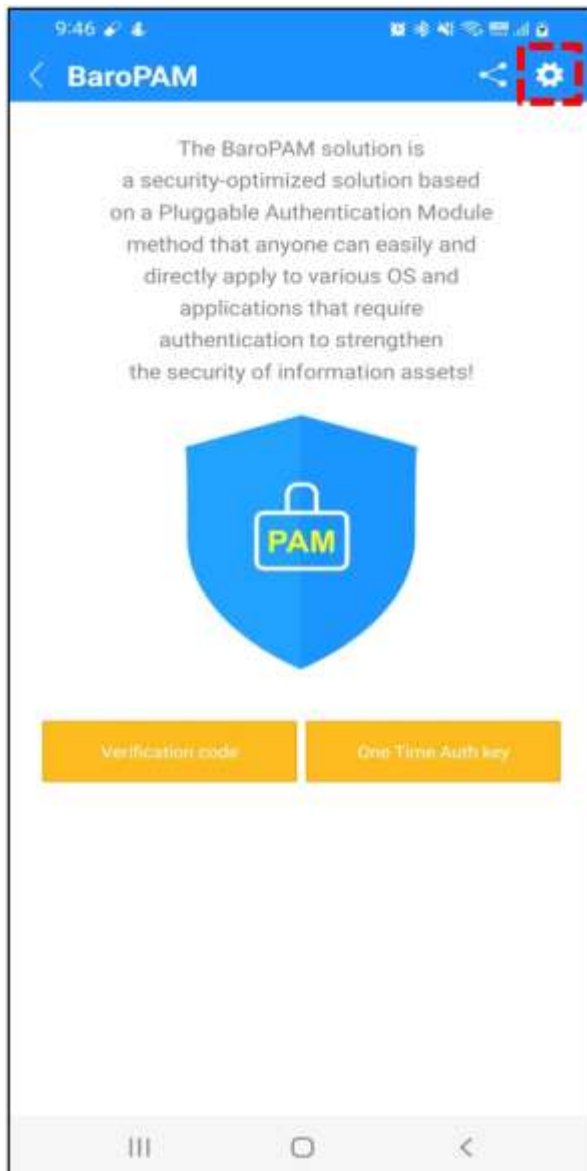
If a secure key is not entered, the message **"Please enter a secure key."** appears on the screen.

If the auth key cycle time is not entered or is out of range, the message **"Please enter or confirm the auth key cycle."** appears on the screen.

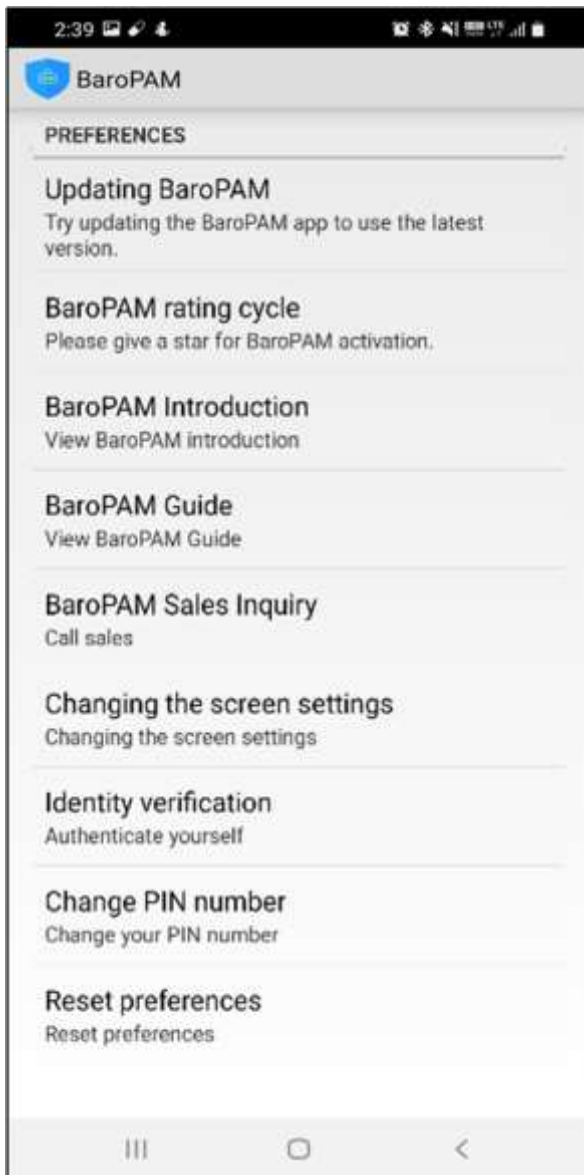
If the modification or deletion of information on the information asset is not completed normally, the **"Modify/Delete server information"** screen continues, and when completed normally, the **"List of server information"** screen appears.

## 1.6 Using the BaroPAM app(Environment settings)

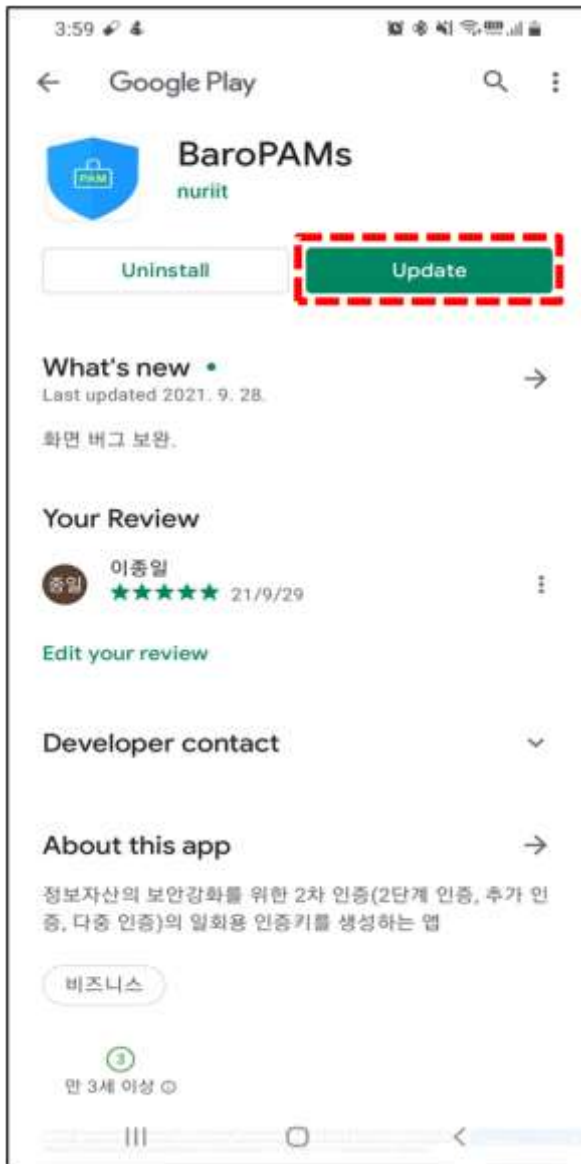
1. To configure the "BaroPAM" app, click the "Preferences" icon as follows.



2. Then the "Preferences" screen appears as follows.

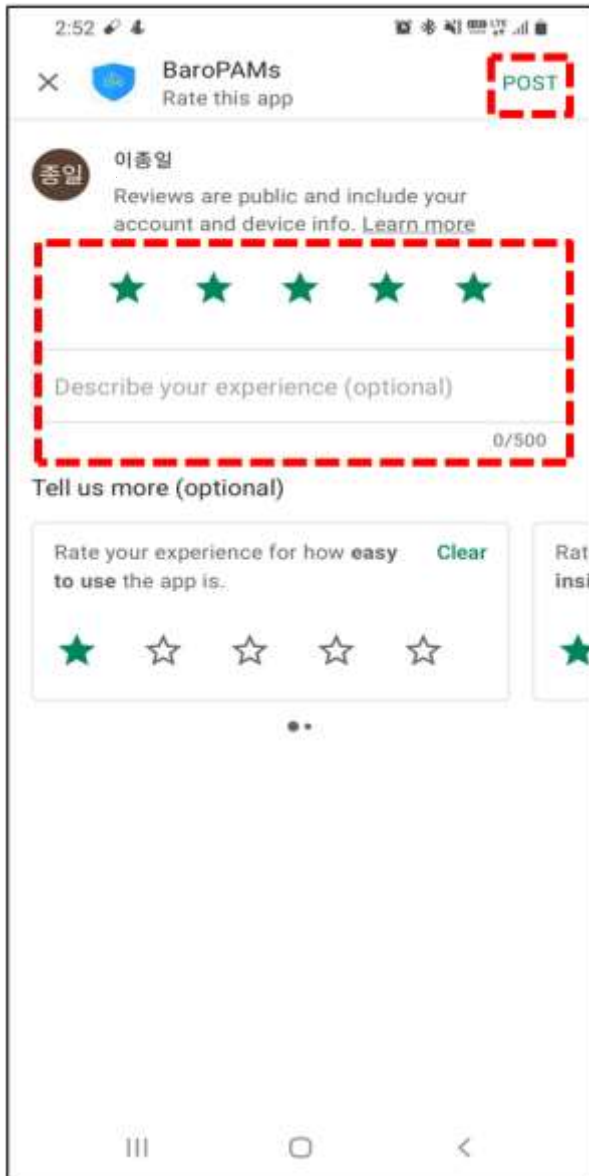


3. Click the "Update BaroPAM" item, and the following "Google Play Store" screen appears.



Click the "Update" button to update the "BaroPAM" app. If there is no "Update" button, it means that there is no information to update.

4. Click the "BaroPAM rating cycle" item and the "Google Play Store" rating cycle screen will appear.



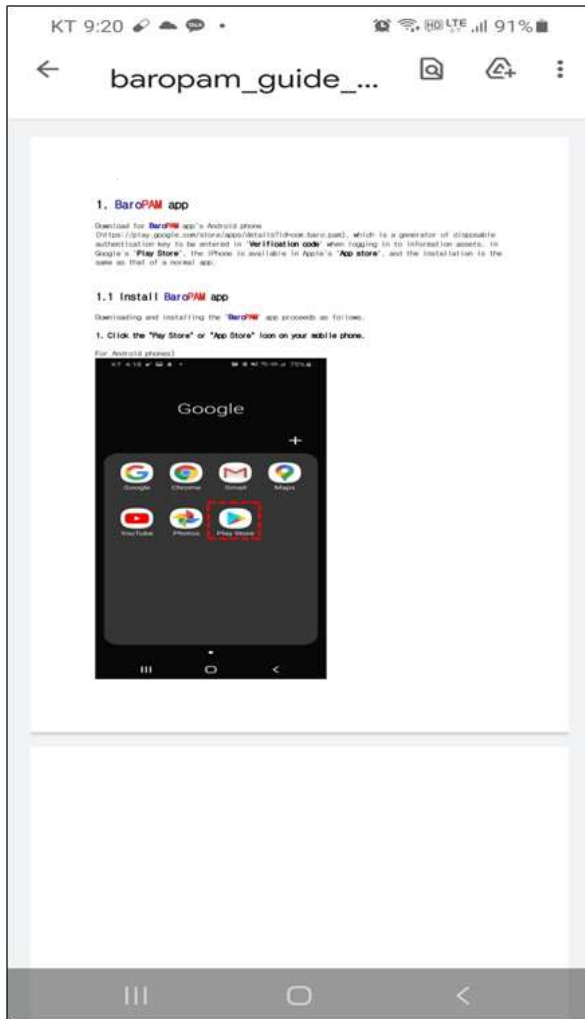
To activate the "BaroPAM" app, fill out and submit a star rating and comment to help activate the app.

5. Click the "BaroPAM Introduction" item and the following "BaroPAM Solution Introduction" screen will appear.



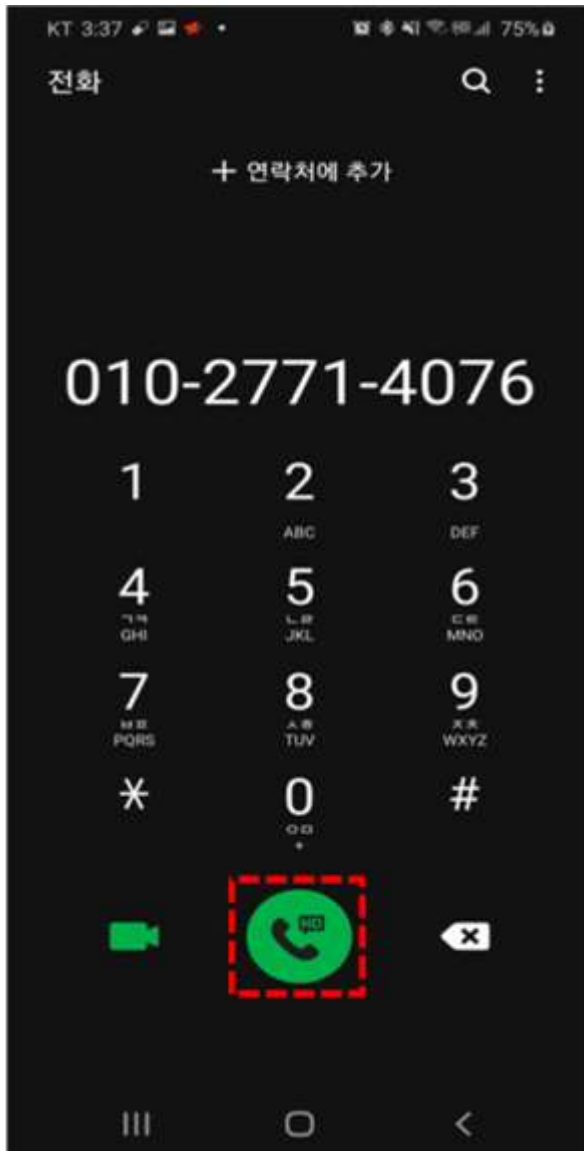
The "BaroPAM Solution Introduction" is provided in the form of a pdf file (baropam\_intro\_en.pdf).

6. Click the "BaroPAM Guide" item, and the following "BaroPAM App User Guide" screen appears.



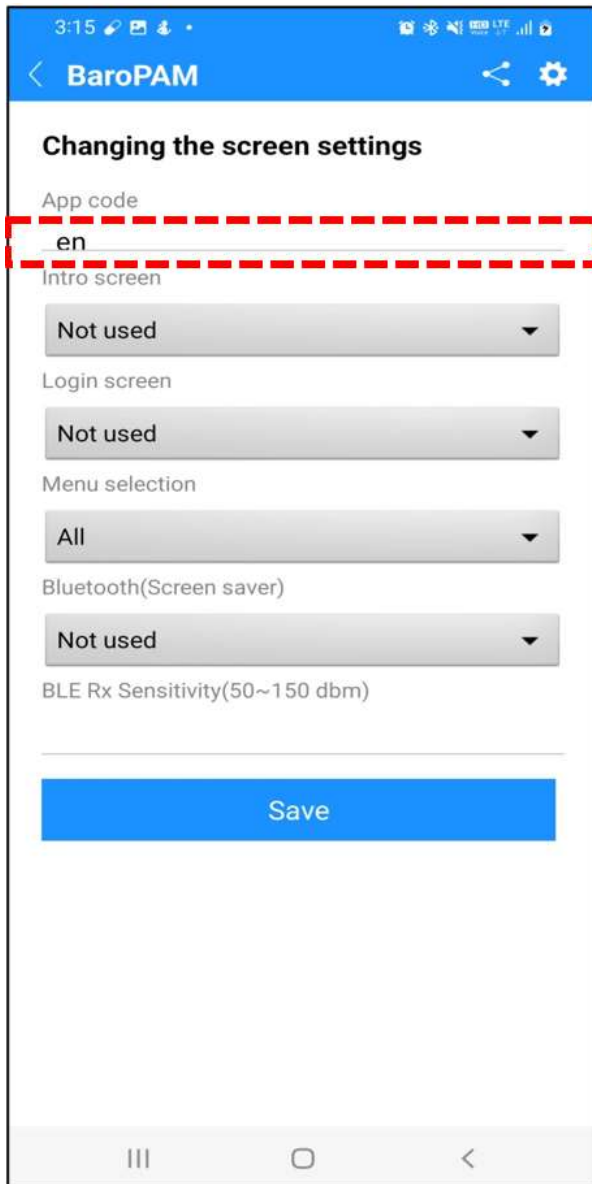
The "BaroPAM App User Guide" is provided in the form of a pdf file (baropam\_guide\_en.pdf).

7. Click the "BaroPAM Sales Inquiry" item and the following "Call" screen will appear.



You can inquire about sales and general information about "BaroPAM" by clicking the "Call" button.

8. Click the "Change the screen settings" item and the following "Change the screen settings" screen appears.



#### ■ App code

Enter the app code given by the vendor. If not entered, the Korean mode "en" is designated by default, and the "BaroPAM" app is switched according to the entered app code.

예) App code

Korean mode: kr  
 English mode: en  
 China mode: cn  
 Japan mode: jp

#### ■ Intro screen

Select whether to use the intro screen. Select "Used" to use the intro screen or "not used" to not use the intro screen.

When using the intro screen When running the "BaroPAM" app, when the "BaroPAM" app is activated, the intro screen of "BaroPAM" is activated for 2 seconds and continues as follows.



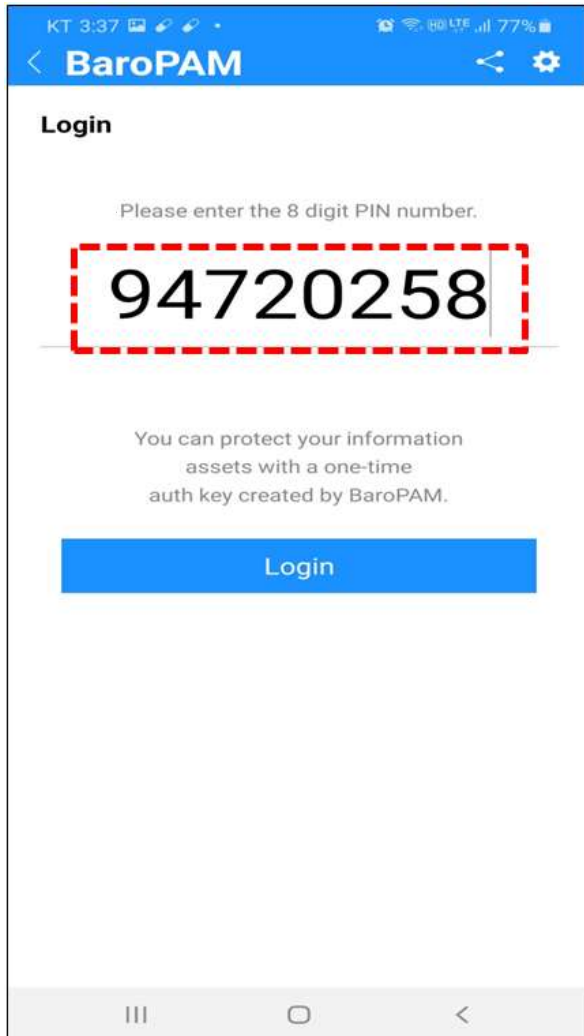
#### ■ Login screen

Select whether to use the login screen. If you do not use the login screen, select "Not used". When using a PIN number, select "PIN number is used". When using fingerprint or face recognition, select "Biometric auth is used".

When using biometric authentication such as fingerprint or face recognition as the login screen, register fingerprint and face recognition first in "Phone Settings → Biometrics and security → Fingerprints and Face Recognition".

When running the "BaroPAM" app, you can log in to the "BaroPAM" app through fingerprint or face recognition.

When using a PIN number as the login screen, the "Login" screen appears as follows.



Clicking the "Login" button first checks the validity of the input items and then proceeds to check if the PIN number is correct.

If the PIN number is not 8 digits, the message "Please enter 8 digits of the PIN number" appears on the screen.

If it is different from the set PIN number, the message "The PIN number is wrong. Please check again and enter it." appears on the screen.

If the PIN number fails 5 times in a row, the message "Access denied." appears on the screen and the app is terminated after 3 seconds.

If the intro screen is used when running the "BaroPAM" app, the intro screen first lasts for 2 seconds and then moves to the login screen. If the login screen is used, the login screen proceeds normally and then the menu selection screen moves.

#### ☐ Menu selection

Select the menu to use. Using the screen for selecting "Menu", "All" when using both server and application login information, and when selecting, When using only server information, select "Verification code", and when using only application login information, select "One Time auth key".

### ■Bluetooth(Screen saver)

Windows and Open source OS (HamonikR OS, Gooroom OS, TMaxOS) set a screen saver for information protection on the desktop PC. Select "**Used**" to use the screen saver lock/unlock function, and select "**not used**" to not use the screen saver lock/unlock function.

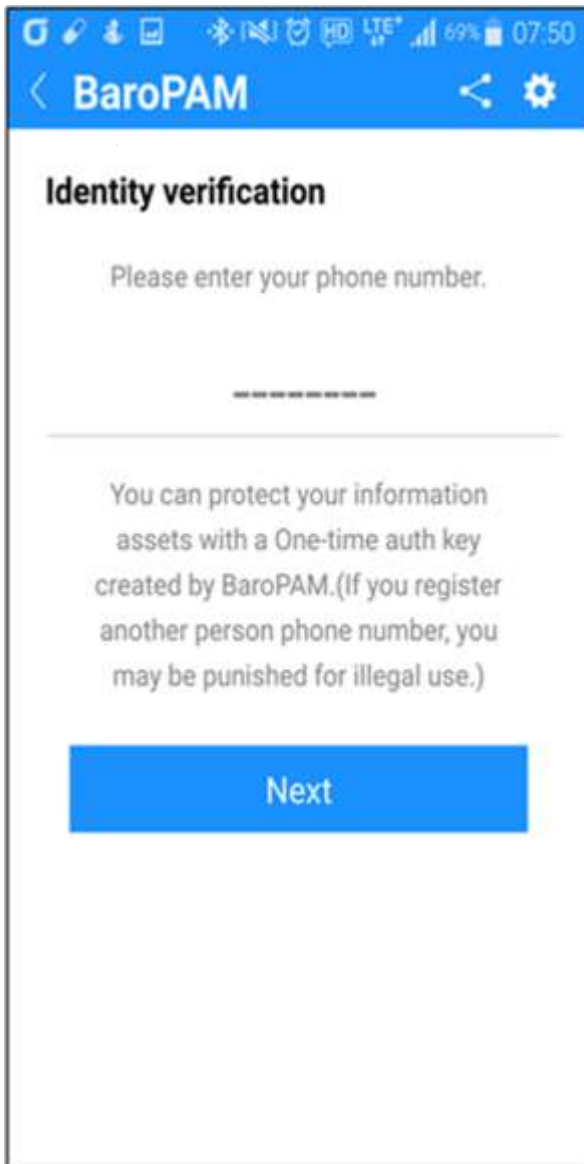
Clicking the "**Save**" button first checks the validity of the input items and then proceeds to save the screen setting information.

If the app code is entered and it is not two digits, a message "**Please enter the app code.**" appears on the screen.

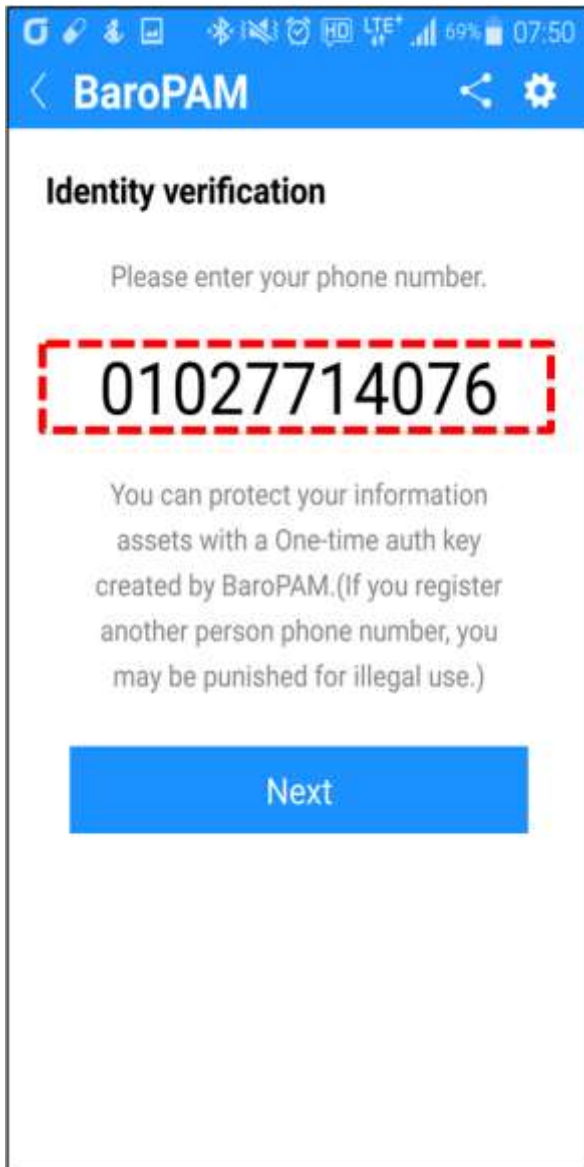
If the number of BLE connections is not entered or out of range, the message "**Enter or check the number of BLE connections.**" appears on the screen.

If the saving operation is not completed normally, the "**Changing the screen settings**" screen is continued. If the intro screen is used normally, the "**Intro**" screen appears, and if not, the "**Menu selection**" screen appears.

9. If you click the "Identity verification" item, the following "Identity verification" screen appears.



1) Enter the iPhone's phone number as a number and click the "Next" button.

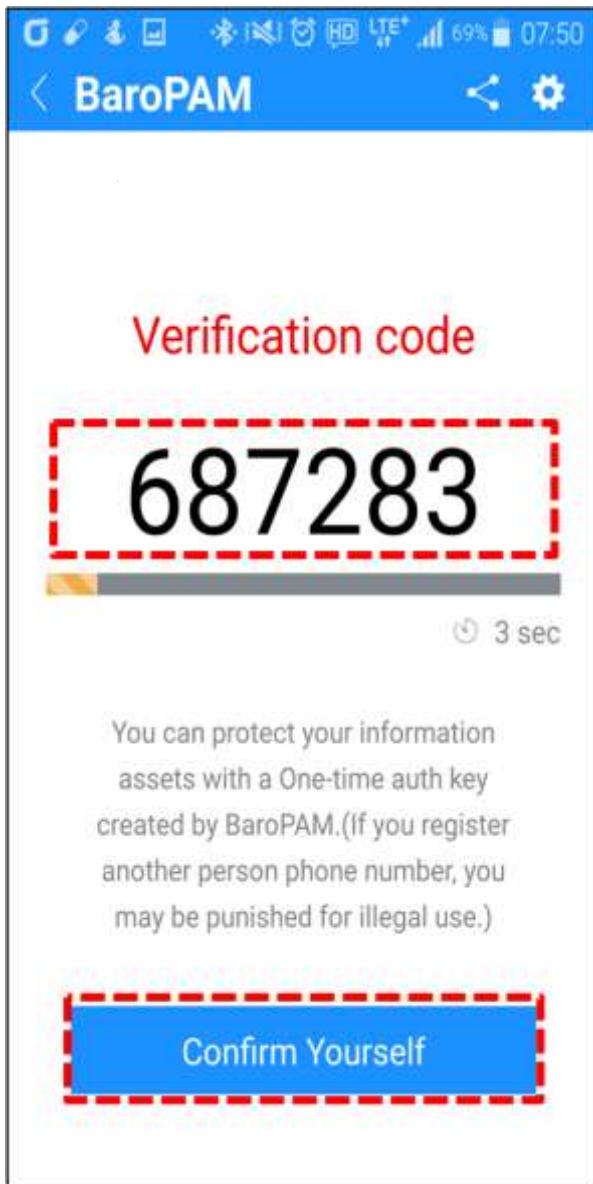


Clicking the "Next" button first checks the validity of the input items, and then delivers the **verification code** generated by the internal algorithm to the "Identity Verification" screen to proceed with the identity verification process.

If the phone number is less than 8 digits, the message "Please enter more than 8 digits of phone number." appears on the screen.

If you enter someone else's phone number, you may be punished for illegal use.

2) After checking the received **verification code**, click the "Confirm Yourself" button.

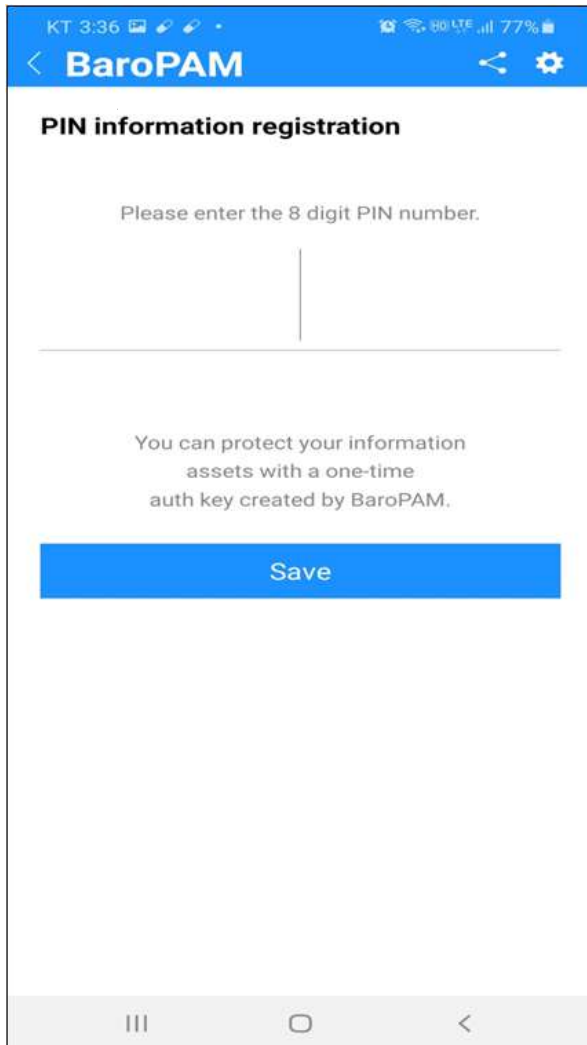


After checking the **verification code**, if you click the "Confirm Yourself" button, the **verification code** is verified by an internal algorithm and then the phone number is stored.

If the identity verification process is successful, it moves to the "Menu selection" screen, but if it fails, it moves to the "Identity verification" screen.

10. If you click the "Change PIN number" item, the following "Change PIN number or PIN number registration" screen appears when you select to use "Login screen" in "Changing the screen settings".

When a PIN number is not registered)

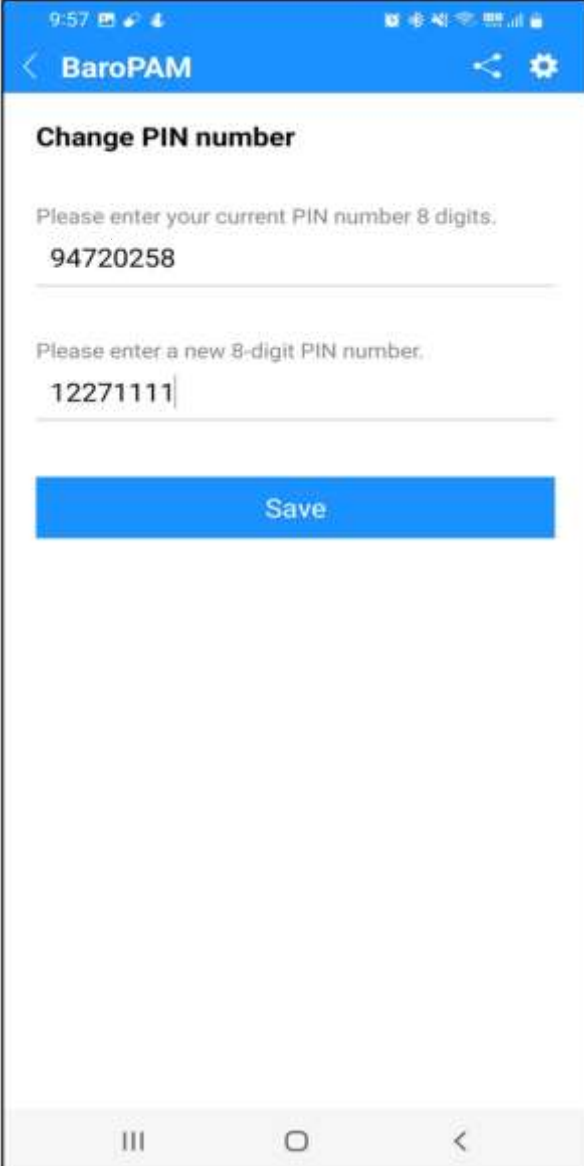


Clicking the "Save" button first checks the validity of the input items and then proceeds to save the PIN number.

If the PIN number is not 8 digits, the message "Please enter 8 digits of the PIN number." appears on the screen.

If the PIN number is not successfully saved, the "PIN number registration" screen continues. If the intro screen is normally used, the "Intro" screen appears, and if not, the "Menu selection" screen appears.

When a PIN number is registered)



■ Enter your current PIN number 8 digits.

Enter the currently set PIN number.

■ Enter a new 8-digit PIN number.

Enter the new PIN number to be set.

Clicking the "Save" button first checks the validity of the input items and then proceeds to save the new PIN number.

If the PIN number is not 8 digits, the message "Please enter 8 digits of the PIN number." appears on the screen.

If the currently set PIN number is different from the newly entered PIN number, the message "The current PIN number is incorrect. Please check again and enter it." appears on the screen.

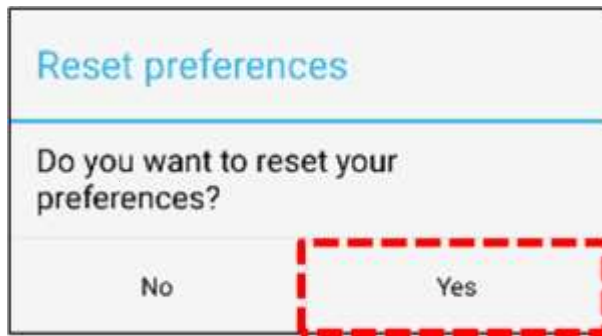
If the currently set PIN number and the new PIN number are the same, the message "The current PIN

number and the new PIN number are the same. Please check again and enter." appears on the screen.

If the new PIN number is not 8 digits, the message "Please enter 8 digits for the new PIN number." appears on the screen.

If saving of the new PIN number is not completed normally, the "Change PIN number" screen continues. If the intro screen is used normally, the "Intro" screen appears, and if not, the "Menu selection" screen appears.

11. If you click the "Reset preferences" item, the following "Reset preferences" screen appears.



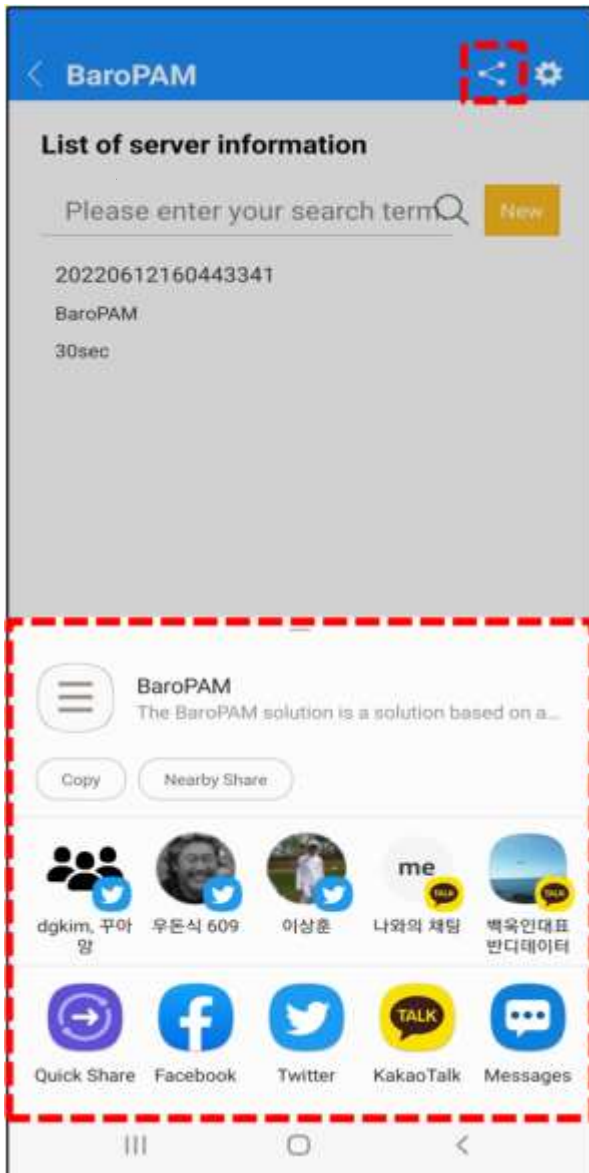
If you click the "Yes" button, the environment information set in the [BaroPAM](#) app will be initialized.

## 1.7 Using the BaroPAM app(Information sharing)

1. To share using SNS of the "BaroPAM" app, click the "Share BaroPAM" icon as follows.



2. Then you will see a screen where you can "Share BaroPAM" as follows.



3. If you want to tweet on Twitter, click the "Tweet" icon and the following screen will appear.



If you click the "tweet" icon, the tweeting process will proceed to the preset Twitter account.

## 1.8 BaroPAM App FAQ

**Message:** In case of iPhone Verification code (Message#n 123456) is not displayed on the keypad when verifying the identity after installing the "BaroPAM" app

**Cause:** The iPhone's "password auto-complete" function is not set.

**Action:** After installing the "BaroPAM" app, be sure to set up a more convenient password auto-completion function from iOS12.

(Go to iPhone's "Settings" -> "Passwords" -> "AutoFill Passwords" -> "Allow")

**Message:** If the date and time of the Android phone or iPhone is different from the current time and the "one-time authentication key" is not correct

**Cause:** It occurs because the date and time of the Android phone or iPhone are not using the time provided by the network.

**Action:** For Android phone, "Settings" -> "General management" -> "Date & Time" -> "Date & Auto Set" and "Time Zone Auto Set" -> "Allow"

For iPhone, "Settings" -> "Date & Time" -> "Automatically set" -> "Allow" of the phone.

**Message:** When a message is sent to a specific type of phone number, it can be sent but not received  
If you can't send a message from your iPhone to your phone number

**Cause:** iPhone Settings Problems Related to Messages

**Action:** Solved after rebooting iPhone

First of all, please check the following to check if messages cannot be sent to your iPhone.

### 1. Settings-in the message

-Check if iMessage is active

-If iMessage is activated, check if your phone number is registered in sending and receiving

-Check if Send by SMS is enabled

-Confirm MMS message activation in SMS/MMS field

-Check if your phone number is registered in the blocked contacts in the SMS/MMS column, and if it is, remove it.

-Enabling notification in the mansion column

### 2. Try sending a new message to your iPhone number directly from the iPhone's Messages app.

-If the transmission does not work at all

-If transmission is possible but not reception

If not in the above case

If someone sends a message to your phone number and receives it normally, check the following URL for each case.

>> 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>

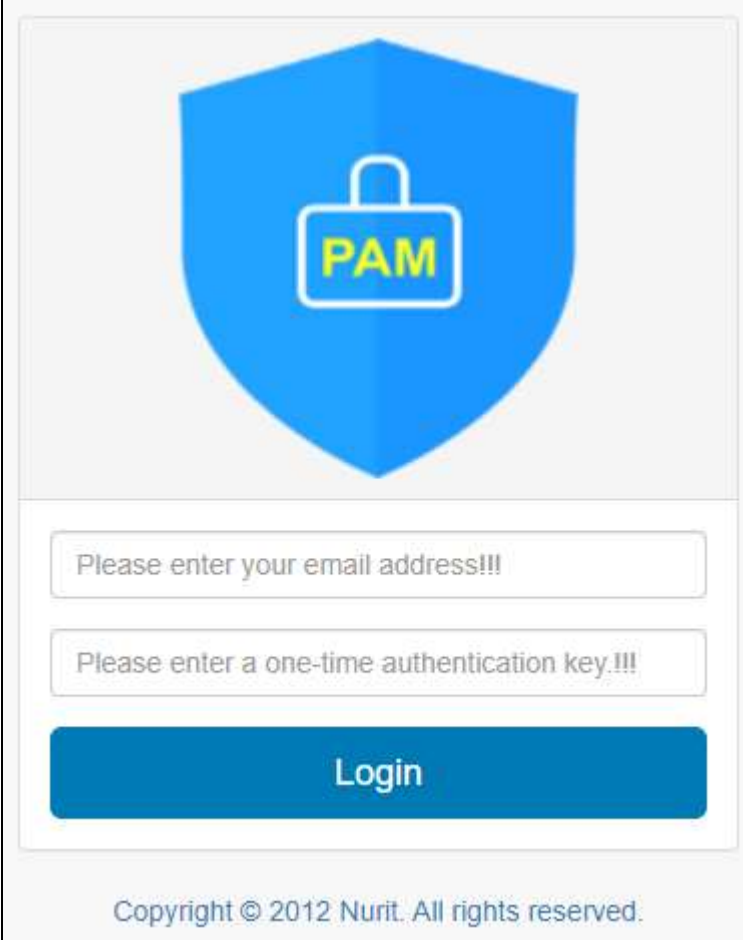
>> Setting up text message forwarding

<https://support.apple.com/ko-kr/HT208386>

## 1.9 BaroPAM Integration API

### 1. Application login

#### 1.1 BaroPAM login screen example)



#### 1.2 Authentication key verification part

The API for verifying the **one-time authentication key** entered in the password field when logging into the application is provided as "**barokey.jar**". You can set the lib directory "**barokey.jar**" of the WAS (Web application Server) or include the directory where "**barokey.jar**" exists in the classpath.

```
[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----- 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

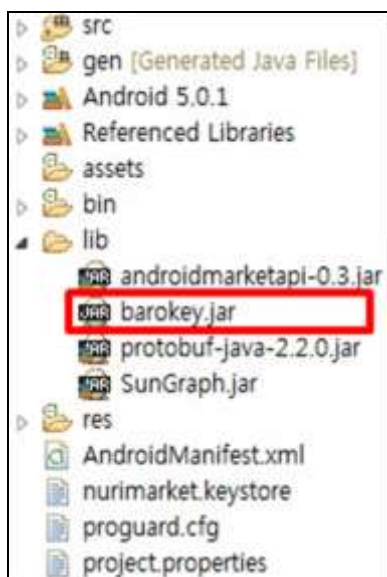
```

The following code can be inserted into the program that verifies the **one-time authentication key**, which is the password entered when logging in to the application.

### 1.3 For Android

#### 1) Authentication key generator part

The API for generating a **one-time authentication key** to be entered in the password field when logging in to the application is provided as "barokey.jar". When using Eclipse, "barokey.jar" must be located in the libs directory.



You can insert the following code in the program that generates a **one-time authentication key**, which is a password to be entered when logging into the application.

```

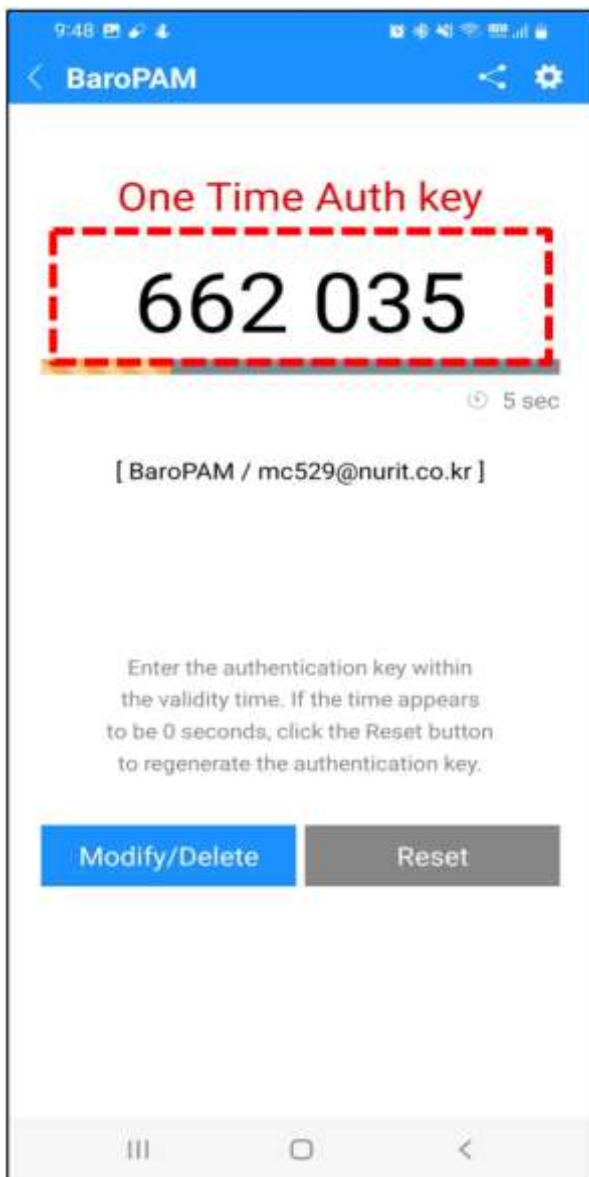
...
import com.barokey.barokey;
...

```

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

Parameter	Remark	Etc
login_id	Set the ID entered in the Login ID field on the login screen.	
phone_no	Set the smartphone number obtained from inside the app using the TelephonyManager method.	
cycle_time	Set the generation cycle (3~60 seconds) of individual <b>one-time authentication key</b> . If the generation cycle of the <b>one-time authentication key</b> designated for each individual is different, the <b>one-time authentication key</b> may be generated differently.	

Screen example)



Screen Layout example)

```
<?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" />

        </FrameLayout>

    </ScrollView>

</FrameLayout>
```

```
<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.baredhen.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_system_nm"
    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_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>

```

#### Program example)

```

package com.baro.otp.info;

import android.Manifest;
import android.annotation.SuppressLint;
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.SettingACT;
import com.baro.common.util.Util;
import com.baro.pam.R;
import com.barokey.barokey;
import com.beardedhen.androidbootstrap.BootstrapProgressBar;

import java.util.Date;

```

```

public class OTPCreateACT extends BaseActivity implements BaseInterface, OnClickListener {
    //public class OTPCreateACT extends AppCompatActivity implements BaseInterface,
    OnClickListener {
        private Button btn_setting, btn_share, btn_close, btn_reset, btn_update;
        private TextView tv_auth_key;
        private TextView tv_remainTime;
        private BootstrapProgressBar progressBar;
        private TextView tv_system_nm;
        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.INTERNET,
        Manifest.permission.ACCESS_WIFI_STATE, Manifest.permission.ACCESS_NETWORK_STATE,
        Manifest.permission.READ_EXTERNAL_STORAGE, Manifest.permission.WRITE_EXTERNAL_STORAGE,
        Manifest.permission.READ_PHONE_STATE, Manifest.permission.CALL_PHONE };

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

        TelephonyManager systemService = (TelephonyManager)
getSystemService(Context.TELEPHONY_SERVICE);
        assert systemService != null;
        PhoneNumber = systemService.getLine1Number();
        PhoneNumber = PhoneNumber.substring(PhoneNumber.length() - 10, PhoneNumber.length());
        PhoneNumber = "0" + PhoneNumber;
    }
}

```

```

        } catch (SecurityException e) {
            e.printStackTrace();
        } 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("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_CLEAR_WHEN_TASK_RESET);
                    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);
        tv_auth_key.setText(barokey.generateKEYL(intent_login_id,
            PhoneNumber ,
            intent_cycle_time));
        mHandlerProc.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 {
                        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;

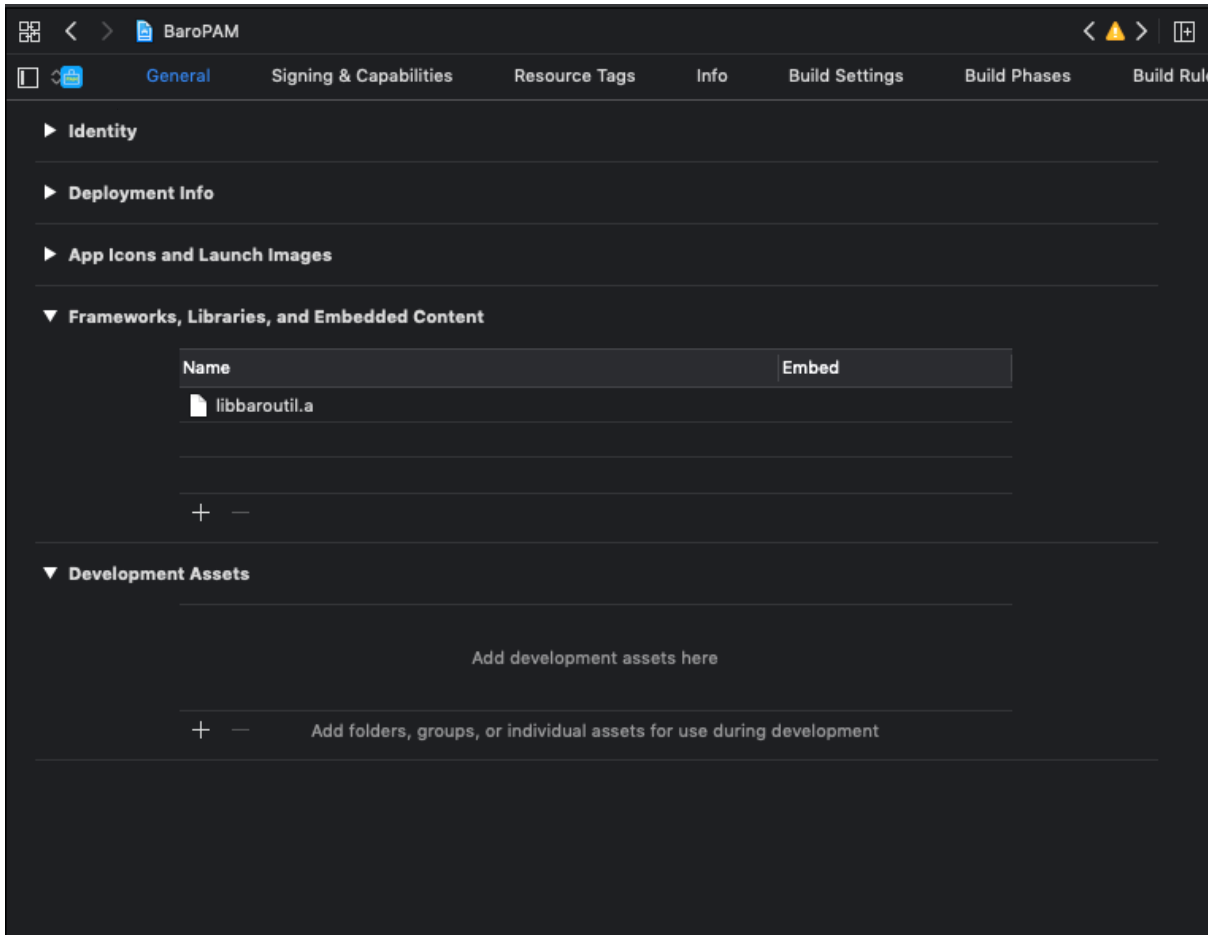
        return createdMillis;
    }
}

```



- libbaroutil.a.simul : for iPhone simulator

This file is registered and used when setting the project of XCode as below.



The APIs related to BaroPAM are as follows. Since the function is composed of the C function interface, the input data type is expressed in the C function style. The source of the usage example is code written in iOS swift 5.0 or later.

**generateKEYL function**

This function creates a **one-time authentication key** used for login/authentication to the application.

Input variable	<b>const char *login_id</b>	The ID entered in the Login ID field of the login screen must be set.
	<b>const char *phone_no</b>	User smartphone number. Unlike in the app for Android, the smartphone number of the user to be used in the authentication module of the server is directly registered and managed in the app without obtaining the user's smartphone number in iOS, and the registered smartphone number is selected and used.
	<b>const char *cycle_time</b>	It must be consistent with the generation cycle (3~60 seconds) of individual <b>one-time authentication key</b> . If the generation cycle of the <b>one-time authentication key</b> designated for each individual is different, the one-time authentication key may be generated differently.

	<b>const char *key_method</b>	"app512" is set as a method of generating a disposable authentication key (app1, app256, app384, app512: app).
Return value	<b>auth key</b>	Returns the generated <b>one-time authentication key</b> .

#### Example of use in swift 5.0 or higher)

```
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..

```

#### Screen example)



### Screen Layout example)

It means Storyboard. Means Storyboard. Refer to developer.apple.com for the meaning of each parameter.

```

<!--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>
        </subviews>
      </view>
    </viewController>
  </objects>
</scene>

```

```

        <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-TI-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-TI-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 재생성하세요." 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=""
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="0.33333333333333331" green="0.33333333333333331"
blue="0.33333333333333331" 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@emplus.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="_tfoot" 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>

```

### Program example)

```
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..

```

```

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

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

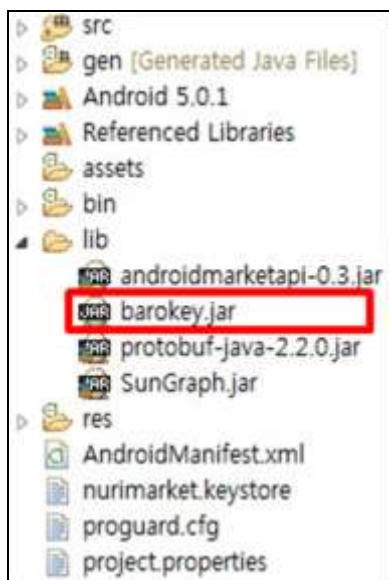
@IBAction func onShare(_ sender: Any) {
}
}

```

## 2. Server login (using Secure key)

### 2.1 For Android

The API that generates a **one-time authentication key** to be entered in the "Verification code" when logging into the server from the **BaroPAM** app is provided as "barokey.jar". When using Eclipse, "barokey.jar" must be located in the libs directory.



The following code can be inserted into a program that generates a **secondary authentication key (one-time authentication key)**, which is a "Verification code".

```

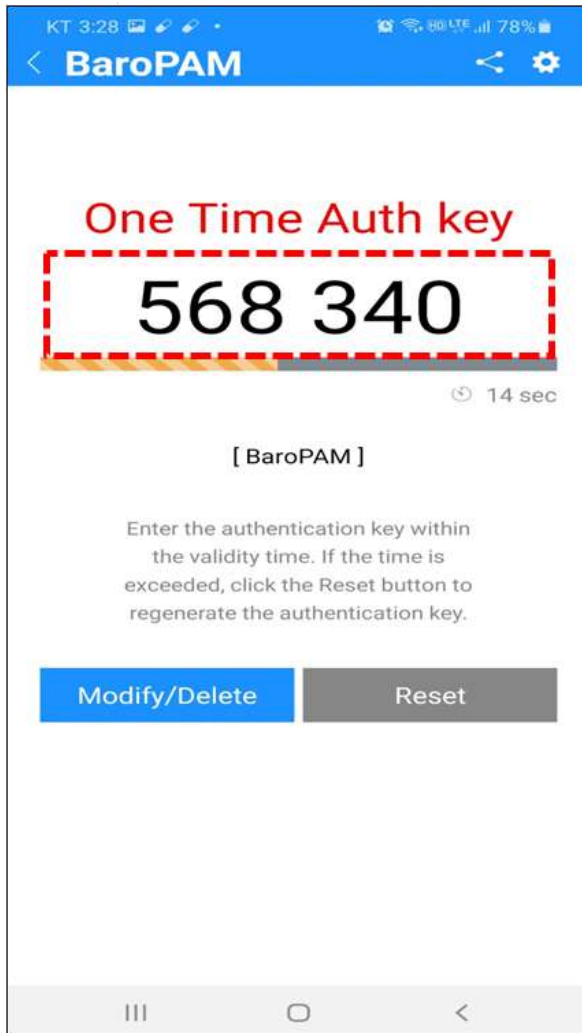
...
import com.barokey.barokey;
...
String auth_key = barokey.generateKEYP(secure_key, cycle_time);
...

```

Parameter	Remark	Etc
secure_key	The key provided by requesting the vendor can be assigned to each server and account. Be sure to set the key specified when setting the <b>BaroPAM</b> server environment.	
cycle_time	Set the generation cycle (3-60 seconds) of the <b>one-time authentication key</b>	

	<p>specified when configuring the BaroPAM server.          If the configuration value is different from the BaroPAM server configuration value, a <b>one-time authentication key</b> that is a "Verification code" may be generated differently.</p>	
--	--	--

Screen example)



Screen Layout example)

```
<?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>

```

### Program example)

```

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.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_Modify, btn_reset;
    private TextView tv_auth_key;
    private TextView tv_server_name;
    private TextView tv_remainTime;
    private BootstrapProgressBar progressBar;
    private String intent_reg_dt = "", intent_server_name = "", intent_hostname = "",
    intent_secure_key = "", intent_cycle_time = "";
    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);

    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_Modify = (Button) findViewById(R.id.btn_Modify);
        btn_Modify.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) {
            String[] a_server_name = intent.getStringExtra("server_name").split("/");
            for(int ii = 0; ii < a_server_name.length; ii++) {
                if (ii == 0) {
                    intent_server_name = a_server_name[ii];
                } else {
                    intent_hostname = a_server_name[ii];
                }
            }
        }
        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_server_name.trim()) &&
            "".equals(intent_hostname.trim())) {
            tv_server_name.setText("");
        } else if ("".equals(intent_server_name.trim())) {

```

```

        tv_server_name.setText("[ " + intent_hostname + " ]");
    } else if ("".equals(intent_hostname.trim())) {
        tv_server_name.setText("[ " + intent_server_name + " ]");
    } else {
        tv_server_name.setText("[ " + intent_server_name + " / " + intent_hostname
+ " ]");
    }
    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_CLEAR_WHEN_TASK_RESET);
                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_Modify: // Modify/Delete
                intent = new Intent(PAMCreateACT.this, PAMModifyACT.class);
                intent.putExtra("reg_dt" , intent_reg_dt );

```

```

        intent.putExtra("server_name", intent_server_name);
        intent.putExtra("hostname" , intent_hostname );
        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() {
    try {
        tv_auth_key.setText("");
        createdMillis = estimateCreatedMillis(intent_cycle_time);
        tv_auth_key.setText(barokey.generateKEYP(intent_secure_key,
intent_cycle_time));
        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,

```

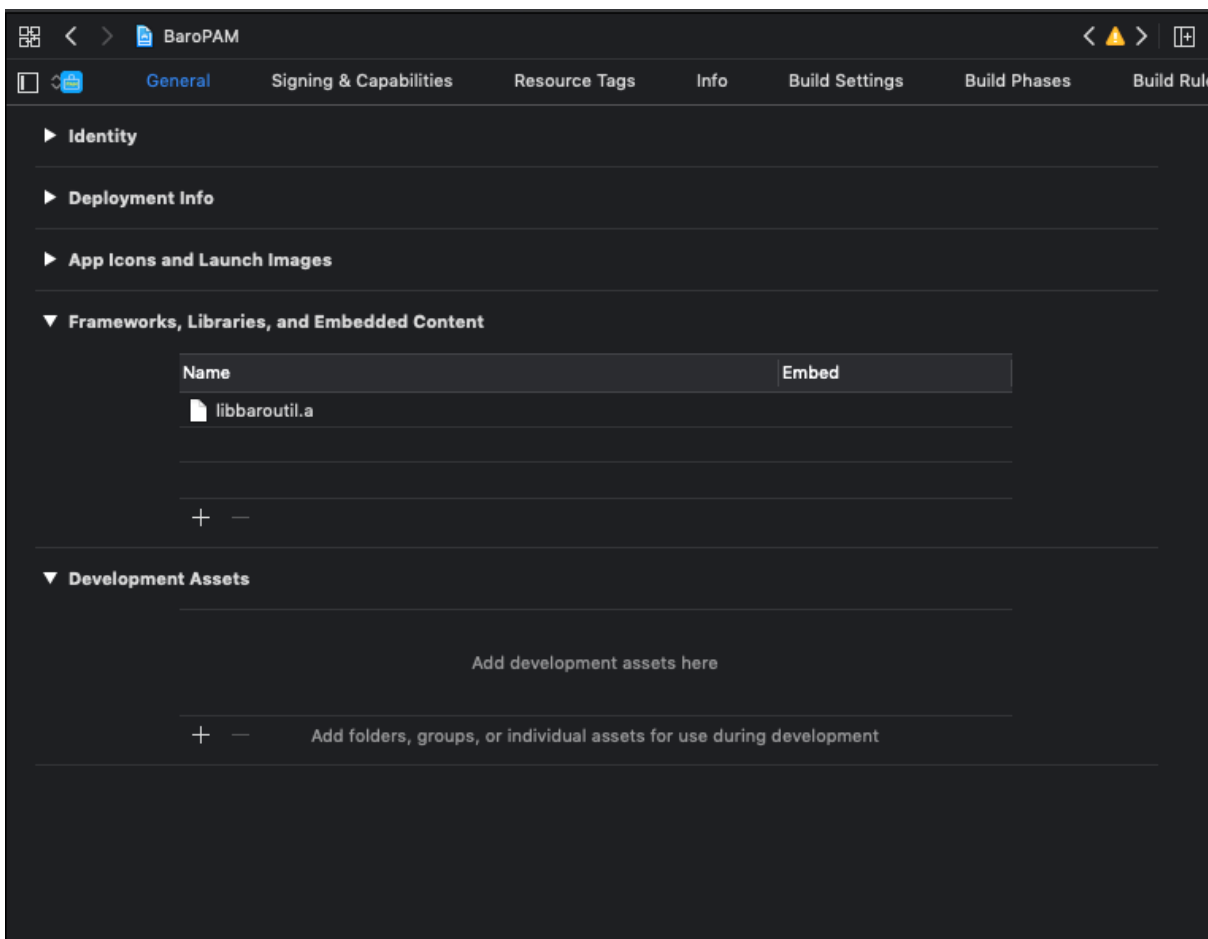


The API that creates a **one-time authentication key** to be entered in the "Verification code" when logging into the server from the BaroPAM app is provided as "libbaroutil.a". This file is a library file for NSObject Interface, which includes barokey, barocrypt, and base64 related libraries.

There are two types of library files. XCode's iPhone simulator and iPhone are both changed to libbaroutil.a if necessary.

- libbaroutil.a.iphone : for iPhone
- libbaroutil.a.simul : for iPhone simulator

This file is registered and used when setting the project of XCode as below.



The APIs related to BaroPAM are as follows. Since the function is composed of the C function interface, the input data type is expressed in the C function style. The source of the usage example is code written in iOS swift 5.0 or later.

**generateKEYP function**

The **one-time authentication key** used to log in to the server is a function that is generated.

Input variable	<code>const char *secure_key</code>	The key provided by requesting the vendor can be assigned to each server and account. Be sure to set the key specified when setting the BaroPAM server environment.
	<code>const char</code>	When configuring the BaroPAM server environment, it must match the <b>one-</b>

	*cycle_time	<b>time authentication key</b> generation cycle (3-60 seconds). If the generation cycle of the <b>one-time authentication key</b> designated for each server is different, the <b>one-time authentication key</b> may be generated differently.
	const char *key_method	"app512" is set as a method of generating a disposable authentication key (app1, app256, app384, app512: app).
Return value	<b>auth key</b>	Returns the generated <b>one-time authentication key</b> .

#### Example of use in Swift 5.0 or higher)

```
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..

```

#### Screen example)



### Screen Layout example)

Means Storyboard. Refer to [developer.apple.com](https://developer.apple.com) for the meaning of each parameter.

```

<!--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-c0-5Tt"/>
      </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="2a1-2K-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="6a1-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="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="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="0.33333333333333331" green="0.33333333333333331"
blue="0.33333333333333331" 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="clb-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>

```

## Program example)

```

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, "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 Website

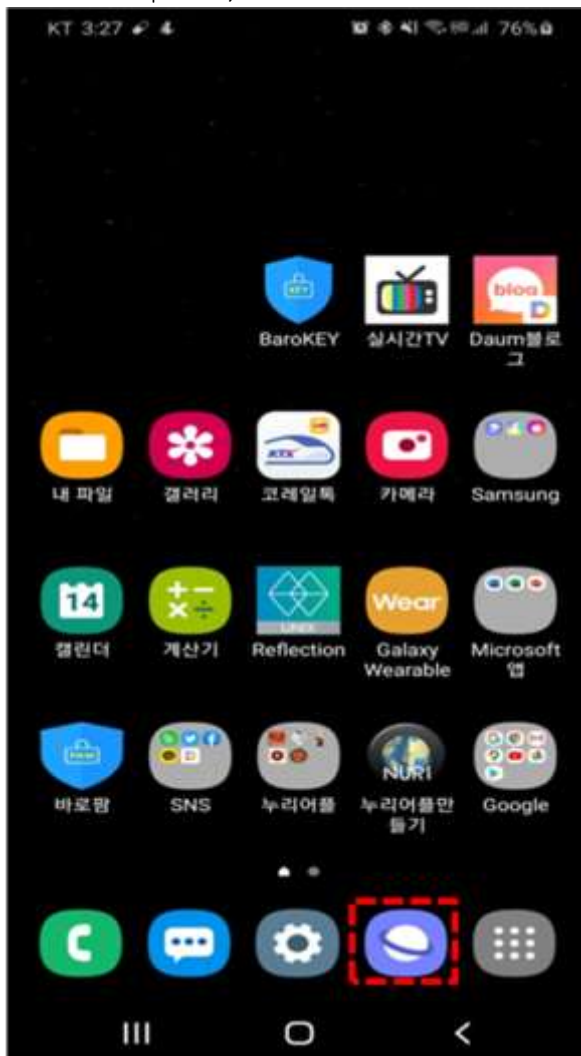
Also, through the BaroPAM website ([www.baropam.com](http://www.baropam.com)), so that the service is not interrupted when there is a problem with the BaroPAM app (for Android phones and iPhones), which is a generator of disposable authentication keys to be entered in the verification code when logging into information assets. Provide services.

The "BaroPAM" web screen is largely composed of membership registration, login, server information management, application information management, PIN information management, Access log management, Abort, and Logout.

### 2.1 Using the BaroPAM Web(Membership registration)

1. Activate the app by clicking the icon of the "web browser" app on your smartphone, and then enter the URL of BaroPAM's website, [www.baropam.com](http://www.baropam.com), in the "Enter search term or URL" field at the top of the app.

For Android phones)



For iPhones)



2. When the BaroPAM website is activated, the following "BaroPAM" login screen appears.



If you are using the "BaroPAM" website ([www.baropam.com](http://www.baropam.com)) for the first time, enter the "회원가입 (Sign Up)" button, or if you have already used it, enter your 11-digit smartphone number and 8-digit PIN number and click the "로그인(Login)" button.

3. If you are not registered as a member (for the first time using the "BaroPAM" website), click the "회원가입(Sign Up)" button and the following screen will appear.



The screenshot shows a mobile application interface for BaroPAM. At the top, the status bar shows 'KT 9:47' and '85%' battery. The app header features the 'BaroPAM' logo and a menu icon. Below the header is a user profile icon. The main content area is titled '회원가입' (Sign Up) with a right-pointing arrow. Underneath, there are two input fields: '휴대전화' (Mobile Phone) and 'PIN번호' (PIN Number). Both fields have placeholder text: '휴대폰 번호를 입력해주세요!!!' and 'PIN번호를 입력해주세요!!!'. At the bottom of the form are two buttons: 'Cancel' and 'Save'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

The PIN number is used to prevent the leakage of authentication key information by others when the smartphone is lost.

4. Enter the 11-digit number and 8-digit PIN number of your smartphone and click the "Save" button.

휴대전화	01027714076
PIN번호	94721234

Buttons: Cancel, Save

Clicking the "Save" button first checks the validity of the input items and then saves the membership information.

If the smartphone number is not 11 digits, the message "휴대번호를 입력해주세요.(Please enter your mobile number)" will appear on the screen. Similarly, if the PIN number is not 8 digits, the message "PIN 번호를 입력해주세요.(Please enter the PIN number)" appears on the screen.

If the member registration information is not completed normally, the "회원가입(Sign Up)" screen will continue, and if it is completed normally, the above "로그인(Login)" screen will appear.

Click the "Cancel" button to exit the current screen and the above "로그인(Login)" screen will appear.

## 2.2 Using the BaroPAM Web(Login)

1. Enter the 11-digit mobile number and 8-digit PIN number you set on the login screen, then click the "로그인(Login)" button.

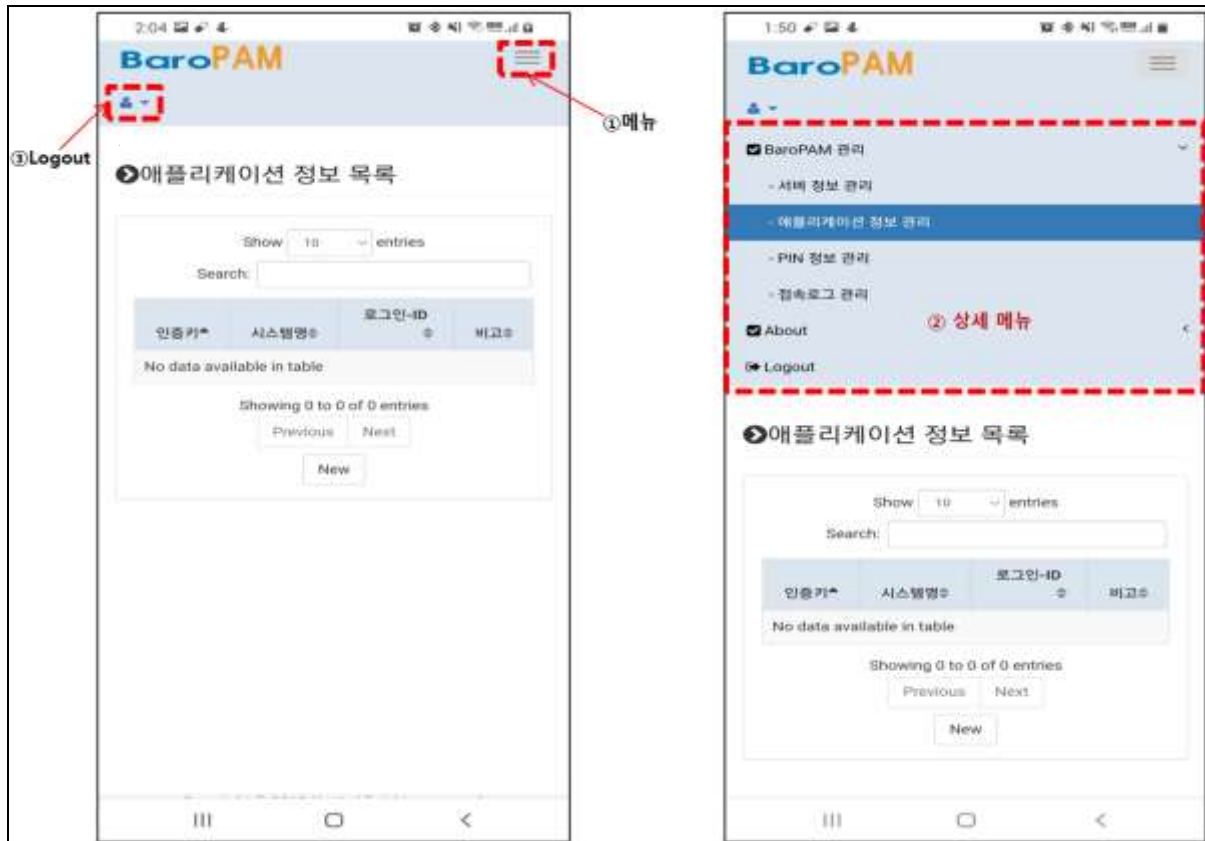


When you click the "로그인(Login)" button, the first step is to check the validity of the input items.

If the smartphone number is not 11 digits, the message "휴대번호를 입력해주시요.(Please enter your mobile number)" appears on the message screen. If the PIN number is not 8 digits, the message "PIN 번호를 입력해주시요.(Please enter the PIN number)" appears on the screen.

If the entered smartphone number is wrong, the message "휴대폰 번호를 확인해주세요.(Please check your smartphone number)" and the "PIN 번호를 확인해주세요.(Please check your PIN number)" message will appear on the screen if the PIN number is incorrect.

2. When login is confirmed, the "애플리케이션 정보 관리(Application information management)" screen appears as follows.



The "애플리케이션 정보 관리(Application information management)" screen manages information for generating a "OTA key" to replace the password when logging in to the application.

Click the "① 메뉴(Menu)" button to display the "② 상세 메뉴(Detailed Menu)" that you can select as shown on the right screen. Click the "③ Logout" button to exit the "BaroPAM" website ([www.baropam.com](http://www.baropam.com)) screen and click "The "Login" screen appears.

## 2.3 Using the BaroPAM Web(Application information management)

1. Click the "메뉴(Menu)" button at the top right and then click the "애플리케이션 정보 관리 (Application information management)" menu in the detailed menu, and the "애플리케이션 정보 목록(List of application information)" screen appears as follows.



2. Click the "New" button on the "애플리케이션 정보 목록(List of application information)" screen to display the screen for registering information on application login as follows.

The screenshot shows a mobile application interface for registering application information. At the top, the BaroPAM logo is visible. Below it, the title '애플리케이션 정보 등록' (Application Information Registration) is displayed. The registration form consists of three input fields: '시스템명' (System Name) with a placeholder '시스템명을 입력해주세요.0~60', '로그인 아이디' (Login ID) with a placeholder '로그인 아이디를 입력해주세요.0~50', and '생성주기(초)' (Creation Cycle in seconds) with a value of '60'. Below the fields are 'List' and 'Save' buttons. The bottom of the screen shows the standard Android navigation bar.

#### ■시스템명(System name)

The system name to log in is a required field. You can enter at least 1 digit and up to 60 digits.

#### ■로그인-ID(Login-ID)

Login-ID is a required field and you can enter at least 1 digit and up to 50 digits.

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

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

It may not be possible if the **OTA key** generation cycle (Auth key cycle time) of the **BaroPAM** verification module and the **OTA key** generation cycle (Auth key cycle time) specified on the web are different. There may be cases where you cannot log in because the **OTA key** is different.

3. After entering the application login information, click the "Save" button.

The screenshot shows a mobile application interface for 'BaroPAM'. At the top, there is a header with the 'BaroPAM' logo and a menu icon. Below the header, the title '애플리케이션 정보 등록' (Application Information Registration) is displayed. The main content area contains three input fields: '시스템명' (System Name) with the value 'BaroPAM', '로그인 아이디' (Login ID) with the value 'mc529@nurit.co.kr', and '생성주기(초)' (Auth Key Cycle Time) with the value '20'. Below these fields are two buttons: 'List' and 'Save'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

When the "Save" button is clicked, the first thing to check the validity of the input items is to save the login information of the application.

If the system name is not entered, the message "시스템명을 입력하십시오.(Please enter the system name)" appears on the screen.

If Login-ID is not entered, the message "로그인-ID를 입력하십시오.(Please enter Login-ID)" appears on the screen.

If the auth key cycle time is not entered or is out of range, the message "인증키 생성주기를 입력 또는 확인하십시오.(Please enter or confirm the auth key cycle)" appears on the screen.

Clicking the "List" button exits the current screen and displays the "애플리케이션 정보 목록(List of application information)" screen.

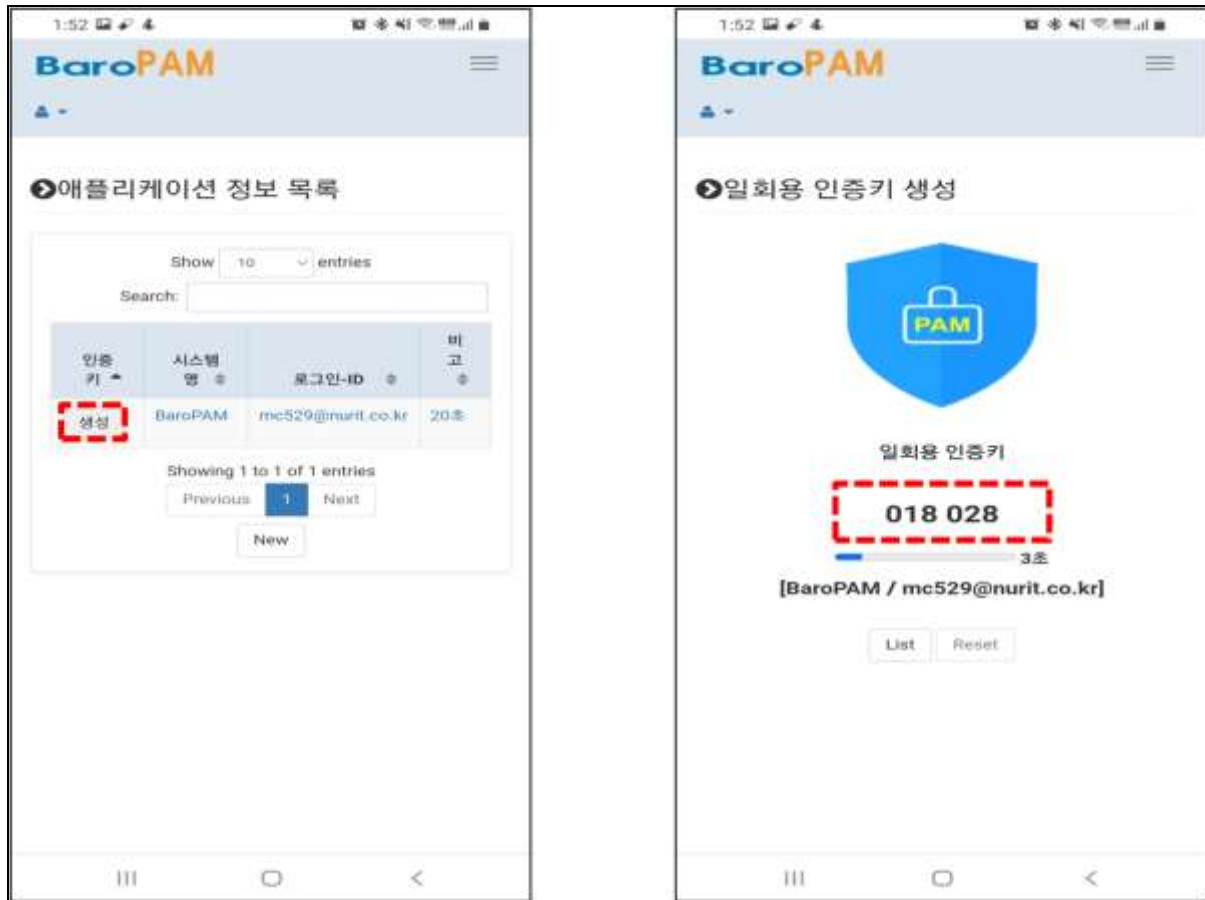
4. If the information saving for the application login is not completed normally, the "애플리케이션 정보 등록(Register application information)" screen continues, and if it is completed normally, the "애플리케이션 정보 목록(List of application information)" screen appears as follows.



If you want to search the login information of the application, enter a search term and click the "Search" button, then the login information of the searched application will appear on the screen.

If you need to add new login information for the application, click the "New" button to display the application login information registration screen.

5. "생성(Generate)" button in the application information list, a new **secondary authentication key(OTA key)** is generated. During the **OTA key** generation cycle (Auth key cycle time), the same **secondary authentication key(OTA key)** occurs, and the newly created **OTA key** appears on the "일회용 인증키 생성(Generate OTA key)" screen as follows.



Clicking the "List" button exits the current screen and displays the "애플리케이션 정보 목록(List of application information List)" screen.

During the same authentication key cycle time, the "Reset" button is deactivated, and when the authentication key generation cycle (Auth key cycle time) passes, the "Reset" button is activated.

When the activated "Reset" button is clicked, a new **OTA key** is created on the "일회용 인증키 생성 (Generate OTA key)" screen.

6. If you need to create a **secondary authentication key(OTA key)** and modify/delete application information, click the application login information item in the application information list as follows.



7. Then, a screen for creating, modify/deleting a **secondary authentication key(OTA key)** based on the application login information appears as follows.

The screenshot shows a mobile application interface for BaroPAM. At the top, the status bar shows the time 1:53 and various icons. The app header displays 'BaroPAM' and a menu icon. The main title is '애플리케이션 정보 수정/삭제' (Application Information Modification/Deletion). Below this is a form with four input fields: '시스템명' (System Name) with 'BaroPAM', '로그인 아이디' (Login ID) with 'mc529@nurit.co.kr', '생성주기(초)' (Generation Cycle) with '20', and '등록일시' (Registration Time) with '2022-04-30 13:52:13.952001'. At the bottom of the form are four buttons: 'List', 'Delete', 'Modify', and 'One Time Auth key'. The bottom navigation bar shows three icons: a list icon, a home icon, and a back icon.

#### ■시스템명(System name)

The system name to log in is a required field. You can enter at least 1 digit and up to 60 digits.

#### ■로그인-id(Login-ID)

Login-ID is a required field. You can enter at least 1 digit and up to 50 digits.

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

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

If the **OTA key** generation cycle (Auth key cycle time) of the Barofarm authentication module is different from the **OTA key** generation cycle (Auth key cycle time) specified on the web, you may not be able to log in. Log in may not be possible because the **OTA key** is different.

If the system name is not entered, the message "시스템명을 입력해주세요.(Please enter the system name)" appears on the screen.

If Login-ID is not entered, the message "로그인-ID를 입력해주시요.(Please enter Login-ID)" appears on the screen.

If the auth key cycle time is not entered or is out of range, the message "인증키 생성주기를 입력 또는 확인해주시요.(Please enter or confirm the auth key cycle)" appears on the screen.

If the application's login information modification(Modify) or deletion>Delete) is not completed normally, the "애플리케이션 정보 수정/삭제(Modify/Delete application information)" screen continues, and when it is completed normally, the "List of application information" screen appears.



When the "Modify" button is clicked, the validity of the input items is checked first, and then the information on application login is corrected.

When you click the "Delete" button, a message box appears, "애플리케이션 정보를 삭제 하시겠습니까?(Are you sure you want to delete the application information?)" If you click the "확인(OK)" button, the operation of deleting the information for the application log-in is performed. If you click the "취소(Cancel)" button, the deletion is performed. The job is canceled.

Clicking the "List" button exits the current screen and displays the "애플리케이션 정보 목록(List of application information List)" screen.

Clicking the "One Time Auth key" button creates a new **secondary authentication key(OTA key)**. During the authentication key cycle time, the same **secondary authentication key(OTA key)** is generated.

The newly created **OTA key** appears on the "일회용 인증키 생성(Generate one-time authentication key)" screen as follows.



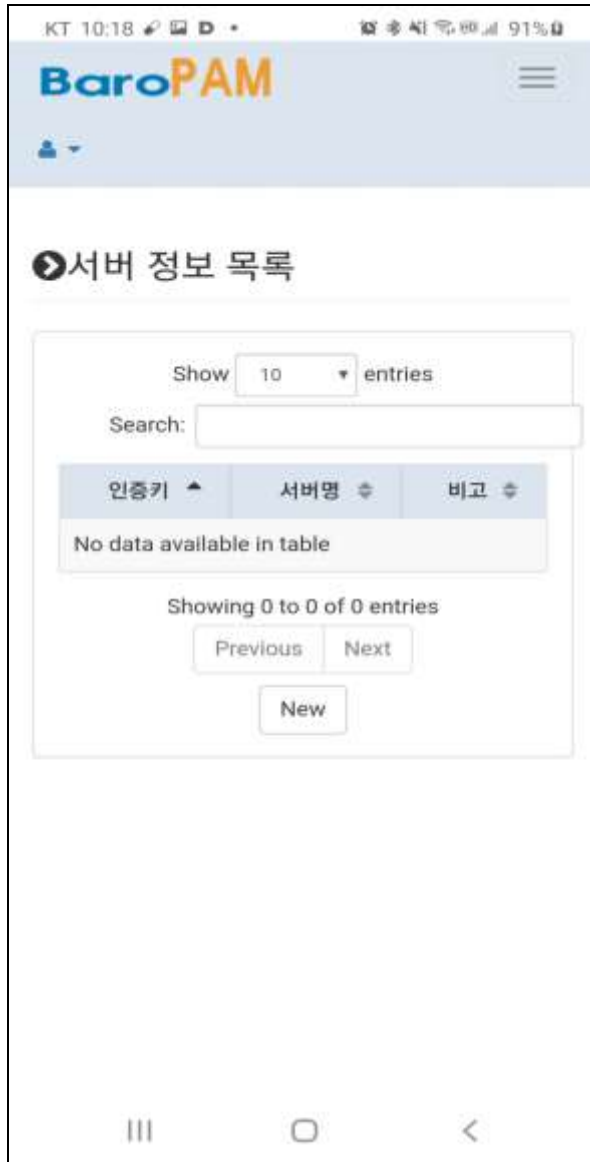
Clicking the "List" button exits the current screen and displays the "애플리케이션 정보 목록(List of application information)" screen.

During the same authentication key cycle time, the "Reset" button is deactivated, and when the authentication key generation cycle (Auth key cycle time) passes, the "Reset" button is activated.

When the activated "Reset" button is clicked, a new **OTA key** is created on the "일회용 인증키 생성 (Generate one-time authentication key)" screen.

## 2.4 Using the BaroPAM Web(Server information management)

1. Click the "메뉴(Menu)" button at the top right and then click the "서버 정보 관리(Server Information Management)" menu in the detailed menu, and the "서버 정보 목록(List of server information)" screen appears as shown below.



2. If you click the "New" button in the "서버 정보 목록(List of server information)" screen, a screen to register information for the server appears as follows.

The screenshot shows the BaroPAM mobile application interface. At the top, the status bar displays 'KT 10:15' and '91%' battery. The app header shows 'BaroPAM' and a menu icon. The main title is '서버 정보 등록' (Server Information Registration). Below the title, there are three input fields:

서버명	서버명을 입력해주세요!!!
Secure key	Secure key를 입력해주세요!!!
생성주기(초)	60

At the bottom of the form, there are two buttons: 'List' and 'Save'.

#### ■Server name

The server name on which BaroPAM will be operated is a required field and can be entered at least 1 digit or up to 30 digits.

#### ■Secure key

Secure key given by information asset or account is a required input item, and you must enter the one requested by the vendor.

If you enter a random "Secure key" that has not been granted by the vendor, an incorrect OTA key is given and you may not be able to log in to the information asset.

If the secure key set in the information asset and the secure key designated by the app are different, the OTA key is different and you may not be able to login.

**Auth key cycle time(3-60 second)**

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

If the **OTA key** generation cycle (Auth key cycle time) set in the information asset and the **OTA key** generation cycle (Auth key cycle time) specified in the app are different, the **OTA key** is different. So there may be cases where you cannot log in.

3. After entering the server information, click the "Save" button.

The screenshot shows the '서버 정보 등록' (Server Information Registration) screen in the BaroPAM application. It features three input fields: '서버명' (Server Name) with the value '누리 App 서버', 'Secure key' with the value 'jllqicHbVqdpj7b4PzBpM2DileBv', and '생성주기(초)' (Generation Cycle) with the value '30'. At the bottom of the form are two buttons: 'List' and 'Save'. The top of the screen shows the BaroPAM logo and a menu icon. The status bar at the top indicates the time is 10:15 and the battery level is 91%.

Clicking the "Save" button first checks the validity of the input items and then proceeds to save the information on the information asset.

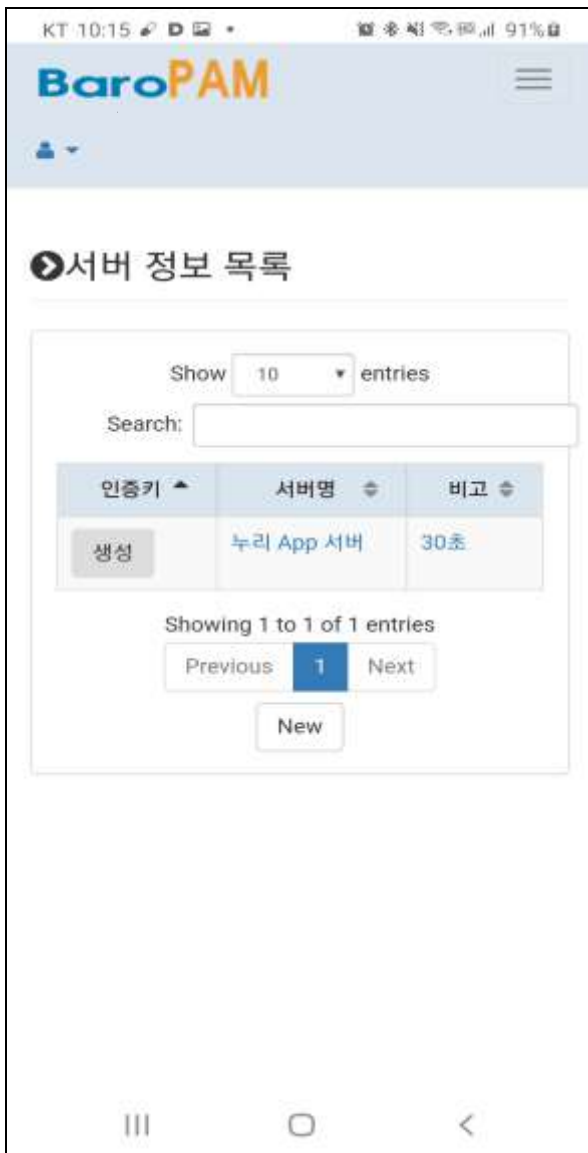
If the information asset name (Server name) is not entered, the message "서버명을 입력해주시오.(Please enter the server name)" will appear on the screen.

If a secure key is not entered, the message "Secure key를 입력해주시오.(Please enter a secure key)" appears on the screen.

If the auth key cycle time is not entered or is out of range, the message "인증키 생성주기를 입력 또는 확인해주시오.(Please enter or confirm the auth key cycle)" appears on the screen.

Clicking the "List" button exits the current screen and displays the "서버 정보 목록(List of server information)" screen.

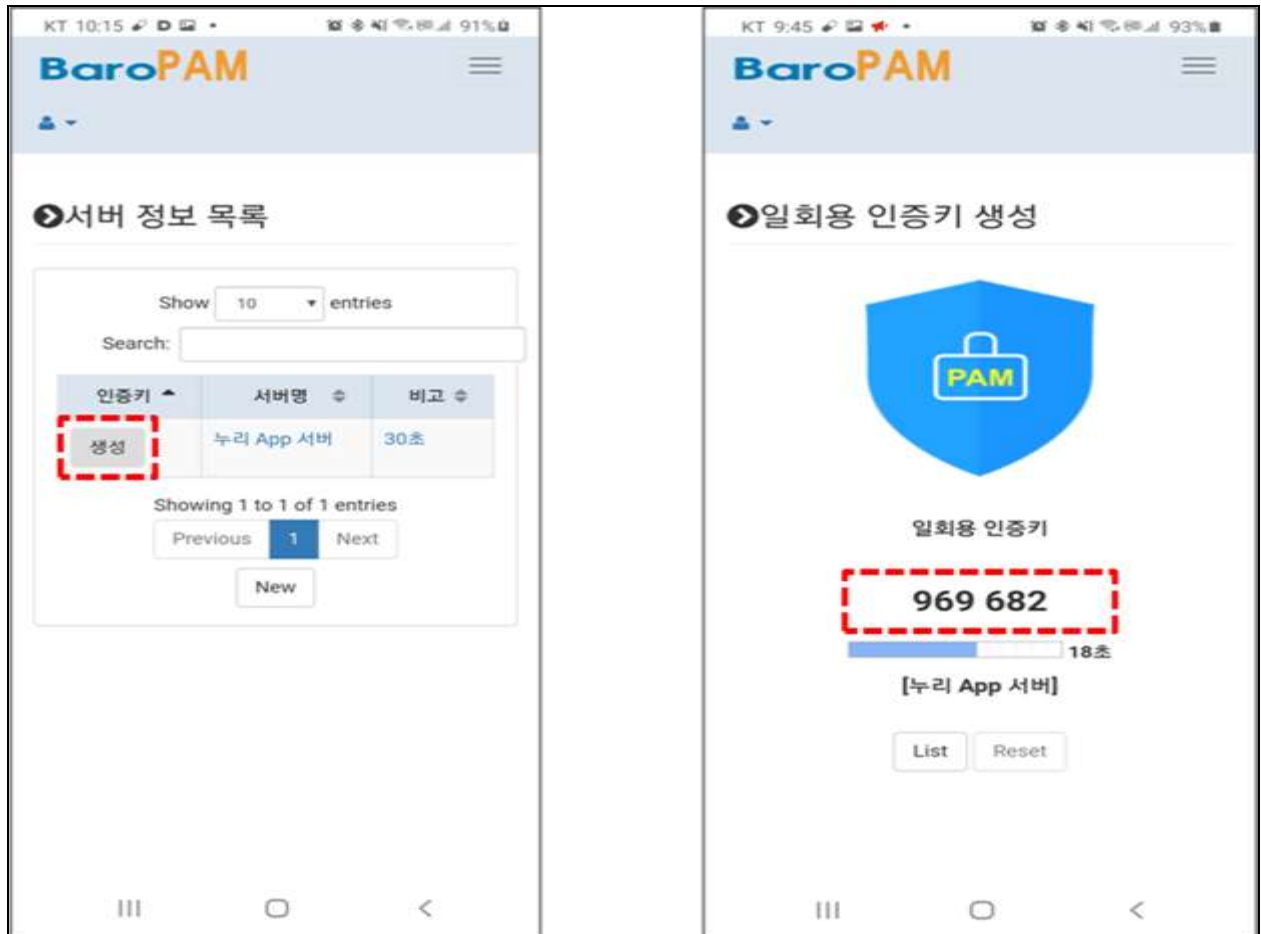
4. If the information storage for the server is not completed normally, the "서버 정보 등록 (Register server information)" screen continues, and if it is completed normally, the "서버 정보 목록(List of server information)" screen appears as follows.



If you want to search for server information, enter a search term and click the "Search" button to display the searched server information on the screen.

If you need to add new server information, click the "New" button and a screen to register server information appears.

5. When you click the "생성(Generate)" button in the server information list, a new **secondary authentication key(OTA key)** is generated. During the authentication key cycle time, the same **secondary authentication key(OTA key)** is generated. Occurs, and the newly created **OTA key** appears on the "일회용 인증키 생성(Generate one-time authentication key)" screen as follows.



Clicking the "List" button exits the current screen and displays the "서버 정보 목록(List of server information)" screen.

During the same authentication key cycle time, the "Reset" button is deactivated, and when the authentication key generation cycle (Auth key cycle time) passes, the "Reset" button is activated.

When the activated "Reset" button is clicked, a new **OTA key** is created on the "일회용 인증키 생성 (Generate one-time authentication key)" screen.

6. If you need to generate a **second authentication key(OTA key)** and modify/delete server information, click the information item for the server in the server information list as follows.



7. Then, the following screen appears to create, modify/delete a **secondary authentication key(OTA key)** based on information about the server.

The screenshot shows the BaroPAM mobile application interface. At the top, the status bar displays 'KT 10:16' and '91%' battery. The app header includes the 'BaroPAM' logo and a menu icon. Below the header, the title is '서버 정보 수정/삭제' (Server Information Modification/Deletion). The main content area contains 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'. The bottom navigation bar shows standard Android icons: a list icon, a home icon, and a back icon.

#### ▣서버명(Server name)

The server name on which **BaroPAM** will be operated is a required field and can be entered at least 1 digit or up to 30 digits.

#### ▣Secure key

Secure key given by information asset or account is a required input item, and you must enter the one requested by the vendor.

If you enter a random "**Secure key**" that has not been granted by the vendor, an incorrect **OTA key** is given and you may not be able to log in to the information asset.

If the secure key set in the information asset and the secure key designated by the app are different, the **OTA key** is different and you may not be able to login.

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

Auth key cycle time is a mandatory input item and can be specified for a minimum of 3 seconds and a maximum of 60 seconds.

If the **OTA key** generation cycle (Auth key cycle time) set in the information asset and the **OTA key** generation cycle (Auth key cycle time) specified in the app are different, the **OTA key** is different. So there may be cases where you cannot log in.

Click the "**Delete**" button to delete the information on the information asset.

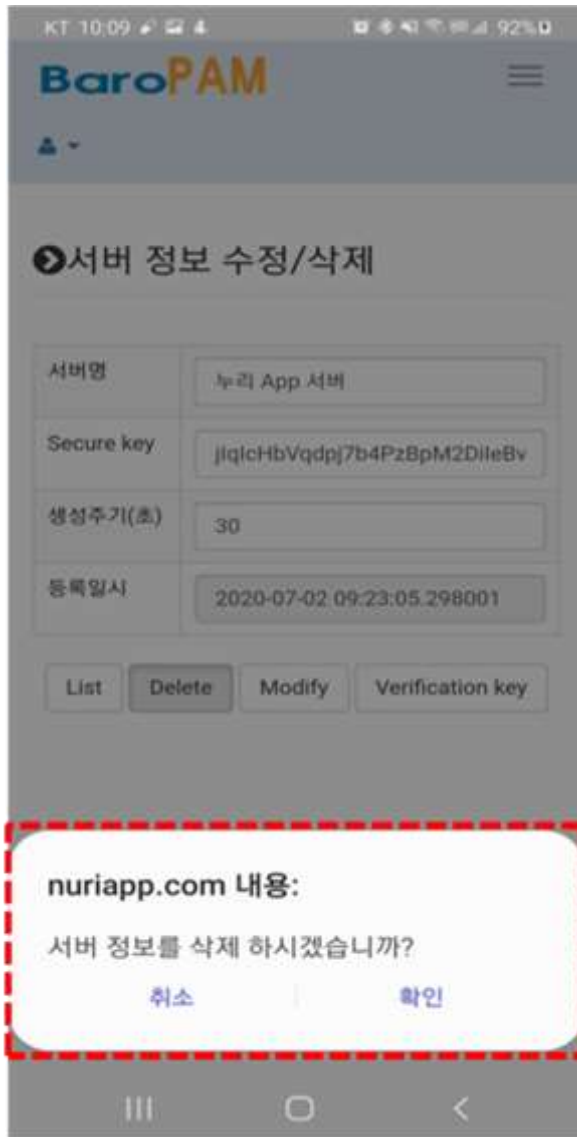
Clicking the "**Modify**" button first checks the validity of the input items and then proceeds to modify the information on the information asset.

If the information asset name (Server name) is not entered, the message "**서버명을 입력해주세요시오.(Please enter the server name)**" will appear on the screen.

If a secure key is not entered, the message "**Secure key를 입력해주세요시오.(Please enter a secure key)**" appears on the screen.

If the auth key cycle time is not entered or is out of range, the message "**인증키 생성주기를 입력 또는 확인해주세요시오.(Please enter or confirm the auth key cycle)**" appears on the screen.

If the modification or deletion of information on the information asset is not completed normally, the "**서버 정보 수정/삭제(Modify/delete server information)**" screen will continue. The "**서버 정보 목록(List of server information)**" screen as shown in the figure appears.



If you click the "Modify" button, the first step is to check the validity of the input items and then modify the information on the server.

Click the "Delete" button to display the message box "서버 정보를 삭제 하시겠습니까?(Are you sure you want to delete the server information?)" Click the "확인(OK)" button to proceed to delete the information for the server. Click the "취소(Cancel)" button to delete it. It is canceled.

Clicking the "List" button exits the current screen and displays the "서버 정보 목록(List of server information)" screen.

Clicking the "One Time Auth key" button creates a new **secondary authentication key(OTA key)**. During the authentication key cycle time, the same **secondary authentication key(OTA key)** is generated. The newly created **OTA key** appears on the "일회용 인증키 생성(Generate one-time authentication key)" screen as follows.



Clicking the "List" button exits the current screen and displays the "서버 정보 목록(List of server information)" screen.

During the same authentication key cycle time, the "Reset" button is deactivated, and when the authentication key generation cycle (Auth key cycle time) passes, the "Reset" button is activated.

When the activated "Reset" button is clicked, a new **OTA key** is created on the "일회용 인증키 생성 (Generate one-time authentication key)" screen.

## 2.5 Using the BaroPAM Web(PIN information management)

1. Click the "메뉴(Menu)" button at the top right and then click the "PIN 정보 관리(Manage PIN Information)" menu in the detailed menu, and the "PIN 정보 변경(Modify PIN Information)" screen appears as follows.

현재 PIN번호	현재 PIN번호를 입력해주시요
새로운 PIN번호	새로운 PIN번호를 입력해주시요
휴대전화	휴대폰 번호를 입력해주시요!
등록일시	2018-12-17 05:29:13.939296
변경일시	2019-08-10 17:25:35.397007

Modify

### ▣현재 PIN 번호(Current PIN number)

Enter 8 digits of the currently set PIN number.

### ▣새로운 PIN 번호(New PIN number)

Enter the 8-digit PIN number to be set.

### ▣휴대전화(Cell Phone)

Enter 11 digits for the number of your smartphone.

현재 PIN번호	94720256
새로운 PIN번호	12271110
휴대전화	01027714076
등록일시	2018-12-17 05:29:13.93929
변경일시	2019-08-10 17:25:35.39700

Modify

If you click the "Modify" button, the first step is to check the validity of the input items and then save the new PIN number.

If the PIN number is not 8 digits, the message "현재 PIN번호를 확인하십시오.(Please confirm the current PIN number)" appears on the screen.

If the currently set PIN number is different from the newly entered PIN number, the message "현재 PIN번호가 틀립니다.(The current PIN number is incorrect)" appears on the screen.

If the currently set PIN number and the new PIN number are the same, the message "현재, 새로운 PIN 번호가 동일합니다.(Currently, the new PIN number is the same)" appears on the screen.

If the new PIN number is not 8 digits, the message "새로운 PIN번호를 확인하십시오.(Please confirm the new PIN number)" appears on the screen.


If the smartphone number is not 11 digits, the message "휴대폰 번호를 확인하십시오.(Please check your smartphone number)" appears on the screen.

If the currently set smartphone number and the entered smartphone number are different, the message "휴대폰 번호가 틀립니다.(The smartphone number is incorrect)" appears on the screen.

If saving of the new PIN number is not completed normally, the "PIN 정보 변경(Modify PIN Information)" screen continues, and when it is completed normally, the "로그인(Login)" screen appears.

## 2.6 Using the BaroPAM Web(Access log management)

1. Click on the "메뉴(Menu)" button on the top right and then click on the "접속로그 관리(Access Log Management)" menu in the detailed menu to display the "접속로그 목록(List of access log)" screen as shown below.

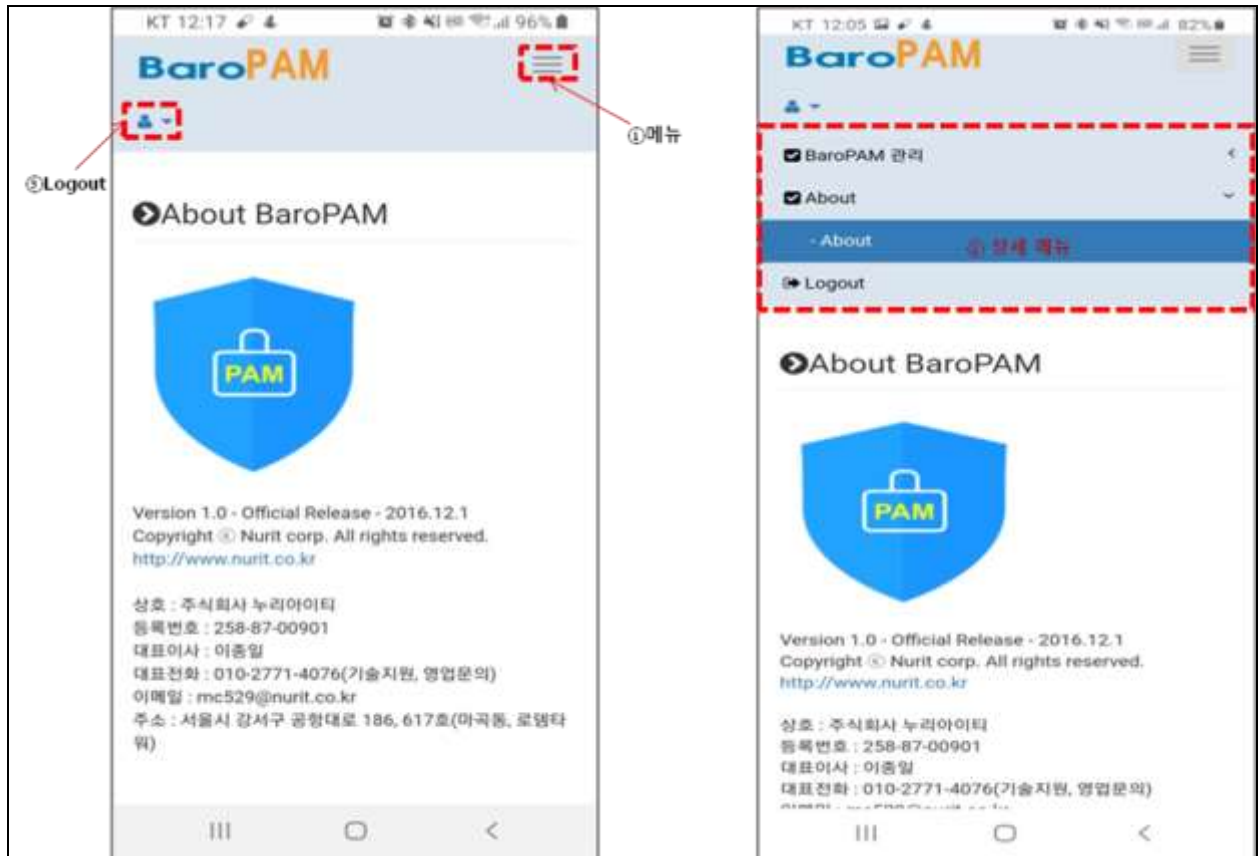


The screenshot shows the BaroPAM web interface for '접속로그 목록' (Access Log List). The interface includes a header with the BaroPAM logo and a menu icon. Below the header, there is a title '접속로그 목록' and a search area with filters for '접속기간' (Access Period) set to '2020-06-16 ~ 2020-06-16', a 'Show 10 entries' dropdown, and a search input field. A table displays the access log entries with columns for '접속일시' (Access Time) and '접속IP' (Access IP). The table contains 8 rows of data.

접속일시	접속IP
20200616071509517001	121.171.109.60
20200616100521502001	211.51.35.93
20200616101538435001	211.51.35.93
20200616101628710001	211.51.35.93
20200616101947838001	211.51.35.93
20200616102301400001	211.51.35.93
20200616102737724001	211.51.35.93
20200616103225300001	220.86.116.28

## 2.7 Using the BaroPAM Web(Abort & Logout)

1. Click the "메뉴(Menu)" button at the top right and then click the "About" menu in the detail menu. The "About" screen appears as follows.



"About" provides an overview of BaroPAM.

Click the "① 메뉴(Menu)" button to display the "② 상세 메뉴(Detailed Menu)" that you can select as shown on the right screen. Click the "③ Logout" button to exit the "BaroPAM" website (www.baropam.com) screen and click. The "Login" screen appears.

### 3. About BaroPAM



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

Company: Nurit Co., Ltd.  
Registration Number: 258-87-00901  
CEO: Jongil Lee  
Tel: +82-2-2665-0119(Technical support, sales inquiry)  
email: mc529@nurit.co.kr  
Address: #913, 15, Magokjungang 2-ro, Gangseo-gu, Seoul (Magok-dong, Magok Techno Tower 2)