

Sleep With Me - COCO

Ko Eunseo

Dept. of Information systems
Hanyang University
Seoul, Korea
koeunseo2@hanyang.ac.kr

Kim Minji

Dept. of Information systems
Hanyang University
Seoul, Korea
99meanz@gmail.com

Moon MyeongKyun

Dept. of Information systems
Hanyang University
Seoul, Korea
11lization@naver.com

Yun Haeun

Dept. of Information systems
Hanyang University
Seoul, Korea
haeun2138@hanyang.ac.kr

Abstract—Modern people habitually watch videos through the media before sleeping. The Corona Blue phenomenon caused by COVID-19 intensified people’s irregular sleep patterns, and more people couldn’t sleep before going to bed and watched videos until late at night. Our team aimed to create an application that automatically provides people with the necessary functions before and after sleep, targeting LG’s mobile monitor StandbyME. Our application not only provides video recommendations and power control so that users can fall asleep, but also provides a function to maintain regular sleep time. Therefore, this helps users maintain their lifestyle. And we maximized the convenience of users by automating the functions of the application.

TABLE I: Role Assignment

Roles	Name	Task description and etc.
User/Customer	Moon Myeongkyun	Find out what kind of sleep patterns people who use electronic devices such as TVs and smartphones usually have before going to bed. Investigate what functions people with these lifestyle patterns can use mainly and whether they can help people with insomnia.
Development Manager	Yun Haeun	The development manager oversees the overall progress of the project. Prepare minutes and manage the schedule about the progress of the project. It also sets future goals for our team and plays a role in monitoring whether each job is performing well.
Software Engineer	Kim Minji	Software developers think about software systems. Investigate the software necessary to implement and think specifically about how to implement it. The best way to connect the application with the speaker-equipped StandbyME is to be found. With feedback from the development manager, continuous updates should be made so that it can be software that users can satisfy.
Designer	Ko Eunseo	The designer produces our team’s logo and icon. ‘Figma’ will be used to design the overall app UI. In the final stage of manufacturing, 3D modeling and data analysis results reports for electronic devices will also be designed together.

I. INTRODUCTION

A. Motivation

Bedrooms are said to be the most important space for modern people, including the MZ generation, at home. However, as COVID-19 spreads, irregular life patterns become commonplace due to social isolation, and so-called ‘Corona Blue’ is becoming a problem. Because of this, many people are staying up all night.

Modern people use electronic devices such as smartphones in their bedrooms before going to bed. It is known that the habit of using electronic devices before going to bed interferes with sleep. However, there are many people who can’t stop even though they know this. This is because there are many people who can’t stop watching their smartphones before going to bed, thinking that it’s better to look at their smartphones than tossing and turning while they can’t sleep.

People use electronic devices before going to bed, but it usually interferes with sleep. But what about electronic devices that help me sleep, rather than interfere with sleep? In fact, many people recognize the need for a good night’s sleep, so they look up videos that help them sleep on YouTube or install and use sleep management apps.

However, smartphones are not suitable for the purpose of sleep management because they have to be held and viewed directly on a small screen. It is also inconvenient to use the smartphone until it gets overheated and then charge it all night again for the next morning.

By improving these points, our team devised COCO, which utilizes the advantages of LG’s mobile TV StandbyME. COCO is an app linked to StandbyME, which helps people get a comfortable sleep and have regular sleep patterns. In this paper, we will look at the main customer base of COCO and the current market status that supports Sleep-tech. Based on this, I will explain the main role of COCO.

B. Intended Audience & Use

Currently, the customer base of StandbyME is mainly composed of MZ generation and newlyweds. In particular, StandbyME is a mobile TV, so user can enjoy video content anywhere in the house. In addition, it is cheaper than general TVs, so both primary and secondary pre-orders have been sold out, especially among MZ generation consumers.

In this way, we looked at the needs of the main customer base of StandbyME which succeeded in business feasibility. As mentioned in 1.1, we focused on 1) StandbyME's property that can be used until sleep because it is mobile TV, 2) the majority of modern people's main TV viewing time is late at night after work, 3) not only those who are currently suffering from insomnia, but also those who want to be guaranteed regular sleep time even if they sleep well. Therefore, we set the user keyword of COCO as 'sleep'. Since COCO is an app that is linked to Stand by me, which is mainly used at home, end-user will be individuals at home.

C. Existing Market

As 'sleep-tech' spreads, various devices and applications for high-quality sleep are being released. By analyzing the advantages and disadvantages of sleep-tech currently provided in the market, we selected the functions to be applied and supplemented in our COCO. Current market trends are as follows.

1) Non-wearable device:

a) *Representative product:* 'Sleep Sense' is a thin and long plate-type device manufactured by Samsung Electronics and Israel's IoT healthcare venture. If user put the device next to the bed, user can analyze the pulse, breathing, and movement that occur during sleep in real time and provide information on smartphone.

b) *Advantages:* The device and smartphone are interlocked, making it convenient for users to use.

c) *Disadvantages:* If non-wearable devices simply provide functions for 'sleep', the number of users will inevitably be small compared to the price.

d) *Application to ideas:* COCO should choose a way to induce sleep without compromising the unique functions of LG Electronics' home appliances.

2) Wearable device:

a) *Representative product:* 'Smart Sleep Headband' is a sleep inducing product provided by Philips. When a user wears the device on his or her head, it helps to relieve sleep and tension through white noise. When linking with the headband application, the user's sleeping time and waking time are identified to manage the living pattern. 'AMO+' is a wearable device for sleep that is worn in the form of a necklace. By applying ultra-fine strength and frequency electromagnetic signals to the human body in a non-contact manner, imbalance in the body is improved to improve sleep quality.

b) *Advantages:* The wearable device is in the form of being worn or attached to a human body. Therefore, sleep can be measured with high accuracy, but poor wearability can interfere with sleep.

c) *Disadvantages:* The wearable device is in the form of being worn or attached to a human body. Therefore, poor wearability can interfere with sleep.

d) *Application to ideas:* COCO we will propose should pursue accuracy and think of a way that does not interfere with human sleep as much as possible.

3) Application:

a) *Representative product:* : 'Calm' is an app that helps meditation and sleep. User can induce sleep by listening to the stories he want (the stories of celebrities, fairy tales, etc.) along with the sounds of nature (rainy sounds, forest sounds, waterfall sounds, etc.). User can set as much time as they want with a timer. It also provides meditation guides for users' mental and health, and music for relaxation and sleep.

b) *Advantages:* The app is convenient to use because it has a lower price barrier than the device. It is convenient to use because it is a non-contact method.

c) *Disadvantages:* Since the app is used in mobile phones, the user must hold the mobile phone himself to watch the video in the application, and the user can feel frustrated with a small screen. In addition, user may feel uncomfortable such as heat generation or battery consumption due to the fact that user charge it all night and use it all day the next day.

d) *Application to ideas:* Our idea is to set the main device as a TV, not a mobile phone. In addition, based on the fact that 'Calm' provides various contents, we intend to provide various sleep-inducing contents in the apps we will provide.

D. Goal

1) *It presents ideas using TV, the most commonly used LG home appliance just before going to bed:* Most people fall asleep watching TV. Focusing on this point, we would like to propose COCO, which can help induce sleep and identify sleep patterns on TV without using wearable devices to prevent disturbance to sleep.

2) *It is intended to identify the user's sleep pattern by identifying the starting and ending points of the sleep time before bedtime:* It provides a UI that allows users to check their sleep patterns through COCO, collect data, and easily check the data so that they can regularly manage and grasp their sleep patterns.

3) *The function is automatically terminated during sleep time so that COCO does not interfere with sleep:* If the TV monitor is operated while users are sleeping, it may interfere with sleep. Therefore, after being counted as the sleep start time, I would like to propose an automatic turn-off mode that can be automatically turned off without the user turning off the TV or setting the time himself.

4) *Taking advantage of TV's advantages, AI technology and IoT are utilized using monitor screens and sounds:* COCO supports personalized video services that can induce sleep and enables language commands through natural language recognition. In addition, when linking with a mobile phone

application through an IoT function, it provides convenience to enable alarm service and mode setting on a mobile phone.

II. REQUIREMENT ANALYSIS

A. Profile

Due to the characteristics of home appliances, one device may have multiple users, so it is possible to create a personal profile that can use their own settings. The information user need to enter when creating a profile is the profile name, age, gender, desired sleep time, and preferred category. The created profile can be checked on the Profile page where all profiles are gathered, and can be freely deleted. In the initial state where no user has created a profile, they are logged in with the default profile. In the case of a multi-person household, user can create a new profile and change the profile.

B. Sleep mode

When the user presses the moon-shaped button, which means sleep mode as a trigger, the sleep mode is activated. When sleep mode is active, a translucent moon-shaped mark appears in the upper left corner so that user can check it. Most of the functions to be described later operate on the premise that sleep mode is activated.

C. Recommend and play videos that are good for sleeping

COCO recommends a list of videos to help user fall asleep by encouraging the user to press the 'sleep video' button. A user can select a category of videos, and when a desired video is selected among them, a playlist related to the video is automatically created. If user does not like the video while watching, user can move to the next video by pressing the 'next video' button as a trigger, and the list of videos the user watched until the end is recorded. By using this history to train application, user can generate a highly accurate list of recommended videos. If the user wants to continue watching the desired video, this function does not work. For example, if the user wants to watch a movie or media transmitted from a platform other than Sleep Custom Videos, the sleep mode will remain and the video recommendation will not work.

D. Exit stand by me screen and record bedtime

From the time the sleep mode is activated, a pop-up notification appears at the top every 20 minutes. The pop-up notification contains the text 'Are you still awake?' and an 'OK' button. The user can dismiss the pop-up notification in two ways before the next pop-up notification appears. The first way is to click the 'OK' button. If the next pop-up notification appears while the pop-up window already exists, it is assumed that the user is sleeping. The time when the user's sleep is

confirmed is recorded as the bedtime, the video played in StandbyME ends, and the screen is turned off.

E. Customized sleep time alarm

The user sets his or her sleeping time through the app in advance. It is based on manually setting an individual's optimal sleep time, but in the setting process, StandbyME recommends optimal sleep time according to gender and age to encourage users to be guaranteed optimal sleep time. When the sleep time is set, the time to wake up from the time when the user's bedtime time is recorded is automatically calculated to make the alarm sound in StandbyME. When the alarm goes off, two buttons appear on standby: a button to end the alarm and a button containing the meaning that the user has woken up. When the alarm end button is pressed, the sleep time alarm function of StandbyME is terminated as it is. When the wake-up button is pressed, the alarm function is terminated, and the wake-up time of the user is recorded to store the sleep time of the user.

F. Sleeping data graphic provided

StandbyME provides a graphic that allows user to see at a glance whether the desired sleep time has been reached on a daily, weekly, monthly, or yearly basis through the user's stored sleep record data. It helps users intuitively know their sleep patterns by marking them green on the day they reach the optimal sleep time they set, red on the day they do not reach, and gray on the day they do not use StandbyME's sleep mode. If the user presses the 'sleep stamp' button, user can see the user's sleep data graphic at any time.

III. DEVELOPMENT ENVIRONMENT

A. Choice of software development platform

Our team will use webOS, a TV software development operating system used by LG Electronics, and Android studio for application development. WebOS will be used to implement an automatic TV off service that will work on StandbyME and will work on linux-based virtual machines. Android studio is used to implement our apps, including video recommendation services using YouTube api. We will demonstrate the behavior of our services and apps at 'Stand by Me' using TV and tablet emulators supported by each operating system. The tools used for planning are figma and AdobeXD (2021), and the languages used for development are JAVA, html, css, javascript (JS).

TABLE II: development language

Tool and language	Reason
JAVA	The biggest feature of JAVA is that it is an independent language of a platform. The program made of java works without any problems if only jvm is installed for the platform. Also, the stability is excellent in two aspects. First, since it is a popular language, there are many references and open sources, and based on this, many large projects have been carried out, so stability is guaranteed in many areas. And secondly, it does not allow pointer variables or memory direct access functions, and it does not allow multiple inheritance, so it is highly stable. JAVA is used as a major language when developing in Android studio.
HTML/CSS /Javascript(JS)	HTML(hypertext markup language) is a markup language for making web pages. Constructive structuring to create web documents is an important role, and because it has images and multiple objects embedded therein, it is easy to create web documents using 'tags'. CSS (cascading style sheets) is a style sheet language that uses html elements to define how they look in various media and adds design elements to structured html documents. Javascript is an object-based script programming language that is mainly used within a web browser. It allows web pages to operate by dynamically changing multiple elements and content. We create an external web app based on webOS using html/css/javascript.

B. Software in uses



1) *Android studio(2020.3.1)*: Android Studio is based on IntelliJ IDEA and is an official integrated development environment (IDE) for developing android apps. The android studio features IntelliJ's code editor and developer tools, and supports flexible gradient-based build systems, fast and functional emulators, integrated environments that can be developed for all android devices, code templates and github integrations, extensive test tools and frameworks, tint tools, C++ and NDK support, and Google cloud platform. These features increase productivity when building the android app on StandbyME



2) *Github*: GitHub is a web service that supports git store hosting. A git is a distributed version management system and instruction to track changes in computer files and coordinate the operations of those files among multiple users. GitHub supports these feathers in the web format so that they can be easily viewed as a graphical interface. GitHub allows colleagues working on a project to share one workspace and set up an environment optimized for collaboration using the functions of git such as merge, commit, branch, etc. We will also share one repository to collaborate efficiently.



3) *webOS IDE/CLI/Emulator*: WebOS is a mobile operating system running on the Linux kernel that started at Palm and is currently being developed by LG Electronics. Since we use webOS 6.0 on our target device StandbyME, we implement some of the functions using webOS. Here, webOS IDE (Integrated Development Environment) is a software for building an application that combines common developer tools into one GUI (Graphical User Interface (GUI). WebOS also provides IDE to support the graphical development environment. We generally write codes in IDE. The webOS CLI (Command-line Interface) is a method in which users and computers interact with commands through terminals or virtual terminals. Using terminals, we use the app and service package and app distribution to create ipk files as cli. Also, we use the are-inspect command to float a console to find out how the app we make works. WebOS Emulator is a tool that indirectly shows how the software we make works on the target device. Several devices equipped with webOS can quickly enter the app menu and multitask without disturbing the screen user is watching through the card view, which is an advantage of webOS. StandbyME (or a device equipped with webOS) also allows user to preview how our app COCO works with the card view function.



4) *Virtual box*: As a computer virtualization program, user can use most of the existing OSs with this program. It offers hardware virtualization VT-x from Intel and AMD-V from AMD. Hardware parts can be implemented virtually. Hard

disks are emulated in a container format called VDI. Currently, many emulators using virtual box are used, and in our project, one of them, webOS TV was used.



5) *Node.js*: Node.js is a software platform used to develop scalable network applications especially on server side. It uses JavaScript as its writing language and has high processing performance through non-block I/O and a single thread event loop. Since it contains a built-in HTTP server library, it can operate on a web server without separate software such as an apartment, which enables more control over the operation of the web server.



6) *MariaDB*: MariaDB Server is one of the most popular open source relational databases. It's made by the original developers of MySQL and guaranteed to stay open source. It is part of most cloud offerings and the default in most Linux distributions. It is built upon the values of performance, stability, and openness, and MariaDB Foundation ensures contributions will be accepted on technical merit.



7) *Figma*: Figma is a UI/GUI production tool and a platform for design. Figma is suitable for efficient prototype work because it is also available on web browsers and free of charge. In addition, there is a 'development tool bar' that conveniently helps develop html and css, so we would like to use the tool as the main design tool. In collaboration, a prototype will be made with figma, and the account holder will share the link to check the art board and work online at the same time.



8) *Adobe XD 2021*: Adobe XD is one of the design tools supported by Adobe that is optimized for UI/GUI production. Before using Figma, we will use XD for prototype work. The

XD has a sample screen that supports the TV screen, so it will be conveniently produced using the tool.



9) *Notion*: Notion is a piece of note taking software and project management software for note-taking, task management, project management, knowledge management, and personal knowledge management using databases and Markdown pages for personal and collaboration work. We used Notion to organize and collaborate on the overall planning and development of COCO in the workspace.

IV. SPECIFICATIONS

A. Main Page - Stand by me

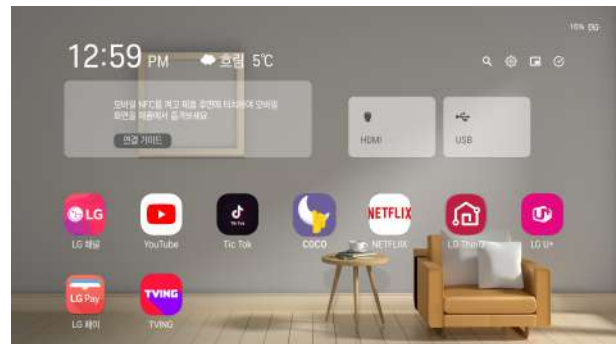


Fig. 1: StandbyME main page

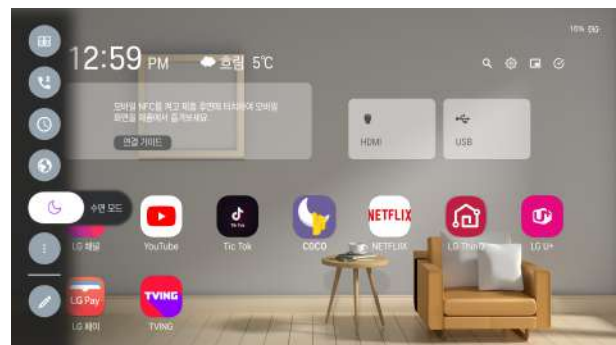


Fig. 2: Open the sidebar

Fig. 1, Fig. 2 are a screen that can be seen when Standby Me is turned on. When the user pushes the left side of the screen to the right, the StandbyME setting window is visible. Currently, buttons such as sound mode and power saving mode are embedded in the StandbyME. Add a button that can turn on and off sleep mode to this button screen.

1) Already logged in to the COCO.



Fig. 3: Click the sleep mode button

When the user turns on the sleep mode through the left button, it is the sleep mode in the name of the last user used. Fig.3 is a screen that user can see when sleep mode button is pressed. User will get a pop-up notification saying, ‘Do you want to start?’. If the name of the user who currently uses the sleep mode and the user profile name to the notification are the same, press the ‘Yes, Start’ button on the left side of the bottom of the pop-up notification to execute the sleep mode. If the name of the user currently using sleep mode and the user applied to the notification are different, user can press the ‘No, I’m a different person’ button on the bottom right of the pop-up notification. By pressing this button, user can go to the Profile page of the COCO and add or change profile.

2) Not logged in to the COCO

Since the user is not logged in to the COCO, the user moves to the Login page of the COCO.

B. Login page - COCO account does not exist.

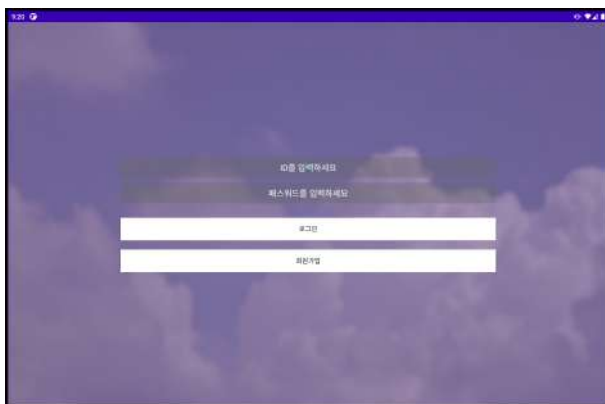


Fig. 4: Login page

Fig. 4 can log in by entering an ID and password. If the user does not have an ID, the user can create an account by pressing the ‘Sign Up’ button.

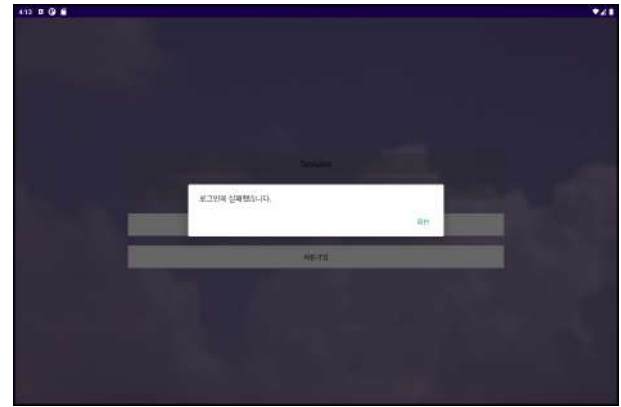


Fig. 5: Login error

Fig. 5 is a screen in which a notification saying ‘Login failed’ appears in the center when an ID and password that do not exist are entered.



Fig. 6: Login error

Fig. 6 is a screen in which a notification saying ‘Please enter all’ appears in the center when not all spaces are entered.

C. Register page

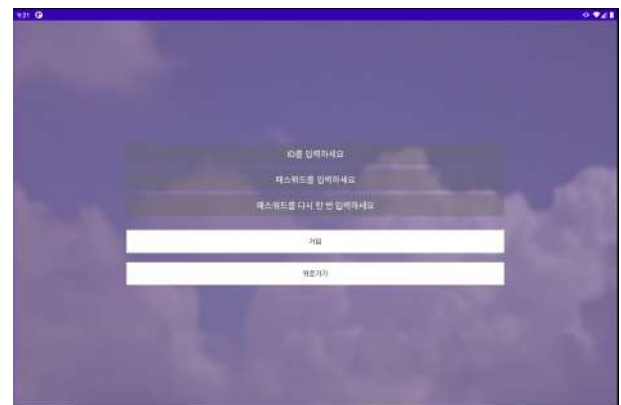


Fig. 7: Sign up page

Fig. 7 is a Register page of the COCO. It appears in the order of ID, password, and password verification.



Fig. 8: Sign up error

Fig. 8 shows a notification of ‘The same ID exists’ when the existing ID and the ID the user intends to subscribe to are the same.

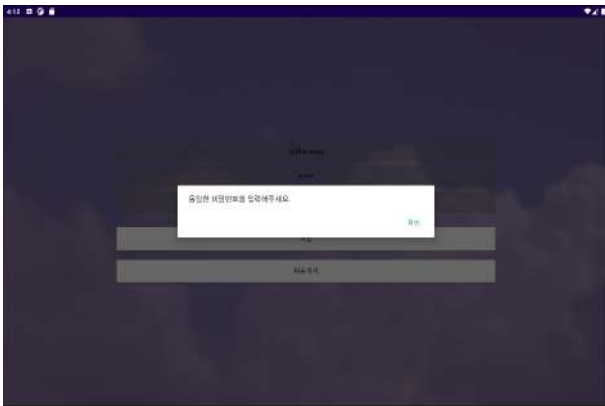


Fig. 9: Sign up error

Fig. 9 shows a notification saying, ‘Please enter the same password’ if the password and ‘Password verification’ that the user wants to sign up for are not the same.

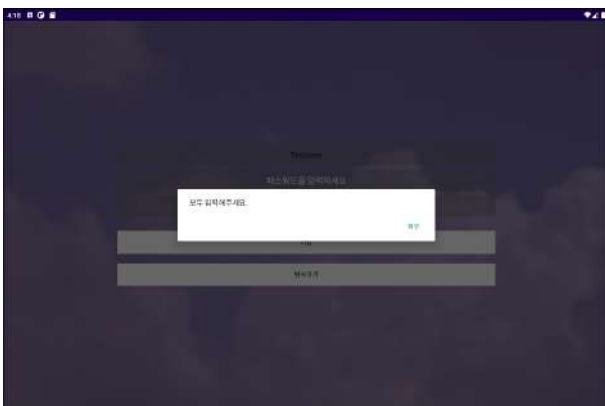


Fig. 10: Sign up error

Fig. 10 is a screen that shows a notification of ‘Please enter all’ when user presses the ‘Register’ button without filling in all the spaces user need to enter.

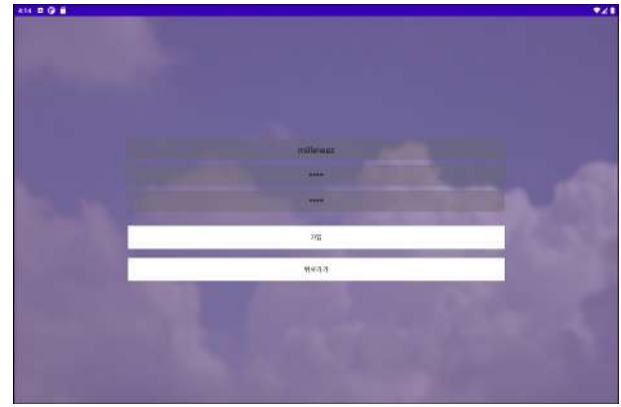


Fig. 11: Correct input

Enter ID, PW, and PW verification correctly and press the join button at the bottom to create an account and return to the Login page. If the user wants to cancel the membership, pressing the return button at the bottom will not create an account and return to the Login page.

D. Main page - Default screen before profile setting.

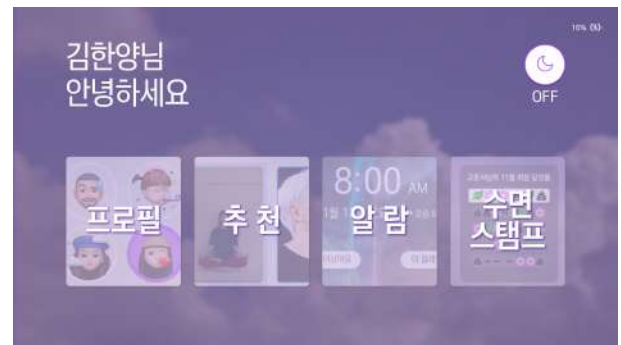


Fig. 12: Main page

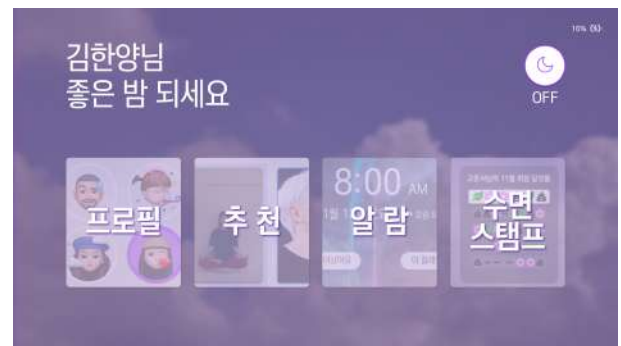


Fig. 13: Main page

Fig. 12 and Fig. 13 are the first screens user can see when he or she sign up and log in for the first time. The default user is ‘Kim Han-yang’ before setting the profile on the Profile page within the application. The phrases shown on the screen will be divided into ‘Kim Han-yang, hello’ before 20 o’clock and

‘Kim Han-yang, have a good night’ after 20 o’clock. At 6 a.m. the following day, the phrase will automatically change to ‘Kim Han-yang, hello’.

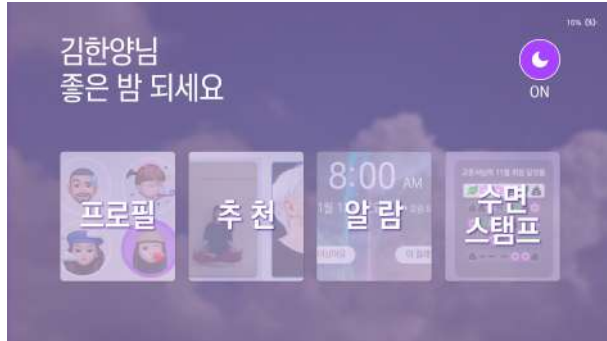


Fig. 14: Main page

The user can turn on and off the sleep mode through the moon-shaped button at the upper right. If it is before the sleep mode is turned on, the moon shape in the upper right is the ‘OFF’ button with a white background. Before the user enters the COCO, turn on the sleep mode in advance on the sidebar of the Main page of StandbyME, or press the sleep mode button on the Main page to reflect it on the Main page and change to the purple ‘ON’ button. Fig. 14 is a screen in which the sleep mode is activated. There are four buttons according to the function on the Main page. A ‘profile’ button that allows user to add or change profiles, a ‘video recommendation’ button that allows user to view videos in the desired category according to that person’s taste, an ‘alarm’ button that is responsible for setting the cycle and time alarm of the sleep mode, and a ‘sleep stamp’ button that allows user to see if the target sleep time is achieved.

E. Profile page

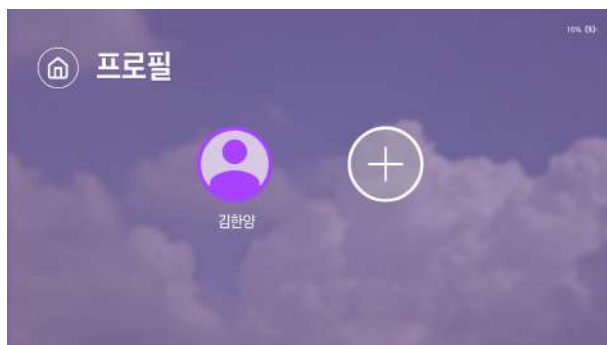


Fig. 15: Profile page

Fig. 15 is a screen user can see when he or she sign up for a membership and enter the Profile page through the Main page after logging in for the first time. There is a default user named ‘Kim Han-yang’, and the user can create a profile through the ‘+’ button located on the right side of the ‘Kim Han-yang’ profile. The set profile turns purple. When the user presses

the ‘home’ icon on the left side of the top, it returns to the Main page.



Fig. 16: Add profile page

When the user presses the ‘+’ button in Fig. 15, it moves to the Add Profile page of Fig. 16. There is a button for selecting a name, gender, and content taste. Up to three tastes can be selected, and the types are ‘animal’, ‘education’, ‘classic’, ‘jazz’, ‘nature’ and ‘daily life’. When the user adds a profile, user can get recommendations for videos that people of that age enjoy by setting the user’s age. After filling in user information, press the ‘complete’ button at the bottom to create a profile. When the user presses the ‘cancel’ button at the bottom, the profile is not generated and then moved to the Fig. 15. In addition, even if the user presses the home icon at the top left, it moves to Fig. 15.

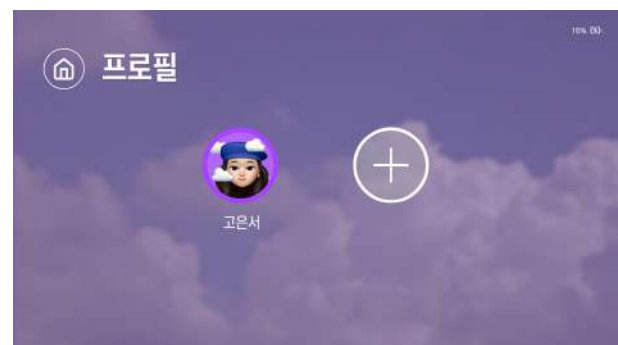


Fig. 17: Profile page

Default user ‘Kim Han-yang’, which is automatically set in the COCO, is replaced by a profile added by the position when a new profile is added. Fig. 17 is a screen that changes the existing ‘Kim Han-yang’ to ‘Ko Eun-seo’. The profile in use is marked with a purple border. Pressing the home icon at the upper left leads to the Main page of the COCO

F. Main page - after profile settings



Fig. 18: Main page

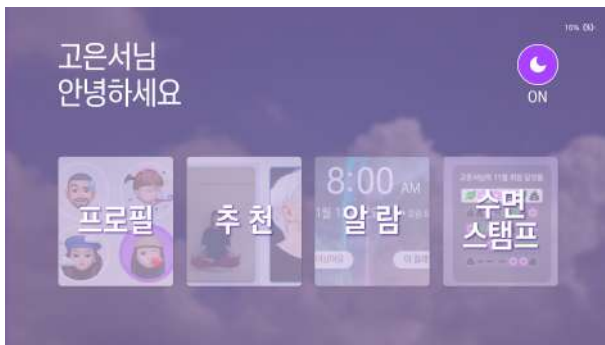


Fig. 19: Main page



Fig. 20: Main page



Fig. 21: Main page

Since the user has changed to 'Ko Eun-seo', the phrase at the top of the screen also changes over time, saying 'Ko Eun-seo, hello' or 'Ko Eun-seo, have a good night'. Fig. 18 and Fig. 19 are the Main pages before 20 o'clock. Fig. 20 and Fig. 21 are the Main pages after 20 o'clock. Other explanations are the same as D.

G. Video recommendation page

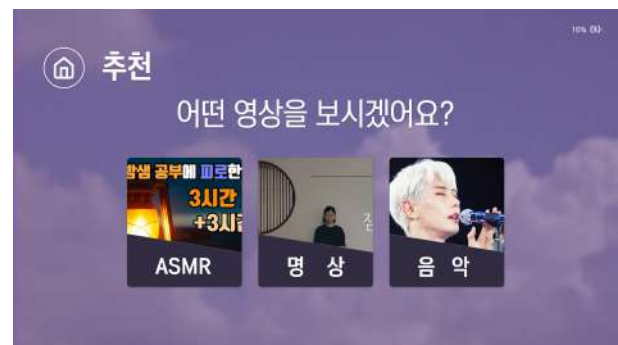


Fig. 22: Content category page

Fig. 22 is a Recommendation page visible when the user presses the 'recommendation' button on the Main page. The categories of recommended content are divided into 'ASMR', 'Meditation' and 'Music'. When the user presses the 'home' icon on the left side of the top, it returns to the Main page. In Fig. 22, when the user presses the 'ASMR', 'Meditation' and 'Music' buttons, it moves to each detailed page. On each detailed page, content may be recommended based on the taste input by the user when setting the profile.

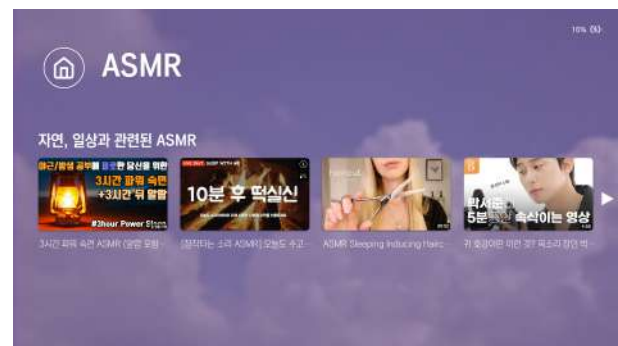


Fig. 23: ASMR page



Fig. 24: Meditation page



Fig. 25: Music page

Fig. 23 is a screen visible when the 'ASMR' button is pressed. ASMR videos related to 'Nature' and 'Daily life', which are pre-set video tastes by the user 'Ko Eun-seo' are listed, and user can choose from them to watch the video, and Fig. 24 is a screen that appears when user presses the 'Meditation' button. Meditation videos related to 'Nature' and 'Daily life', which are pre-set video tastes by the user 'Ko Eun-seo' are listed, and the user can choose from them to watch the video, and Fig. 25 is a screen that the user can see when user presses the 'Music' button. There are music videos related to 'Nature' and 'Daily life', which are pre-set video tastes by the user 'Ko Eun-seo' and the user can choose from among them to watch the video.



Fig. 26: video

Fig. 26 is a video selected by the user. When the user selects the video, the video may be viewed as a full screen.

1) When the sleep mode is not set

If the sleep mode is not set, the video may be continuously viewed until a desired time. If the user wants to stop watching the video, cancel the entire screen, it returns to Fig. 23 or Fig. 24, or Fig .25.

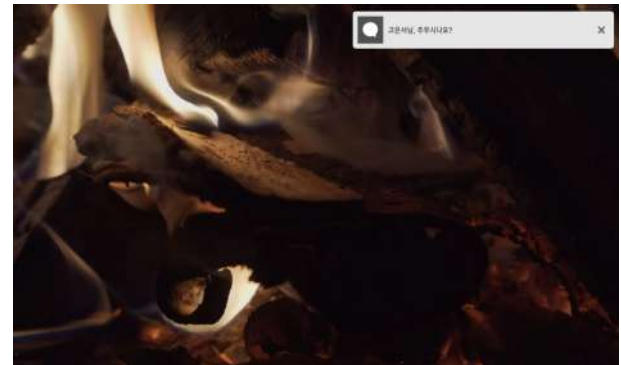


Fig. 27: Notification

2) When the sleep mode is set

a) Fig. 27 is a screen that shows a notification of 'User, are you sleeping?' at the top when the sleep mode is set. If the user is not sleeping or wants to continue watching the video, press the 'X' button to remove the notification. When the notification is removed, it appears again and again repeatedly according to the 'sleep confirmation notification period' set by the user on the Alarm page. The time value of the preset notification cycle is 20 minutes.



Fig. 28: Notification



Fig. 29: Sleeping screen

b) If there is no response to the notification during the notification cycle set by the user, COCO recognizes that the user is sleeping. Fig. 28 is a screen in the upper right corner of the phrase 'User, have a good dream'. And soon it turns into a sleeping screen.

H. Alarm setting page



Fig. 30: Alarm setting page

Fig. 30 is a screen that can be seen by pressing the third Alarm page on the Main page of COCO. From the top left, there are buttons that allow user to select 'target sleep time', 'minimum wake-up time', 'sleep check cycle', and 'alarm repeat cycle' clockwise. 'Target sleep time', 'minimum wake-up time' and 'alarm repeat cycle' are used for the user's morning alarm and become data constituting the Sleep stamp page. The 'sleep check period' is reflected as a notification period to check whether a user in sleep mode is sleeping. After setting all, pressing the 'confirm' button on the bottom right will fix the setting, and when pressing the cancel button on the bottom left, it will return to the Main page. In addition, if the users press the 'home' icon on the left side of the top, it returns to the Main page.



Fig. 31: Alarm setting page

Fig. 31 is a screen that allows the user to set an appropriate sleep time per day through the 'target sleep time' button. It can be set every hour from 6 hours to 8 hours. The default value before setting is 7 hours.



Fig. 32: Alarm setting page

Fig. 32 is a screen that allows the user to set the time to wake up through the 'minimum wake-up time' button. After the standby screen is switched to the sleeping screen, the alarm goes off after the target sleeping time. However, if there is a minimum wake-up time before that, the alarm will go off at the minimum wake-up time. There is no basic alarm time before setting, and the alarm function is activated after the user sets it.



Fig. 33: Alarm setting page

Fig. 33 is a screen that can determine whether to repeat when a morning alarm goes off through the 'alarm repeat

cycle' button. When the user wants to repeat the alarm, the repetition cycle may be set in units of '3 minutes', '10 minutes' and '15 minutes'. The default value before setting is 'none', and when the user sets the repetition time, the alarm repetition function is activated.



Fig. 34: Alarm setting page

Fig. 34 is a screen that allows user to set how many minutes a notification appears to check the user's sleep status when the user turns on the sleep mode through the 'Sleep Check Cycle' button. The default value of the sleep confirmation notification before setting is 20 minutes.

I. Alarm page



Fig. 35: Alarm page

Fig. 35 is a screen when the alarm goes off at a designated time by the user who sets the alarm. The time, date, and day of the week appear on the alarm screen. At the bottom of the screen, there are two buttons from the left, 'I woke up' and 'I want to sleep more'. When the 'I woke up' button is pressed, the alarm is terminated, and COCO records whether the target sleeping time is achieved on the user's sleep stamp(I). If user presses the 'I want to sleep more' button, if the users set the alarm repetition, the alarm goes off again after the set repetition cycle. If the alarm is not set repeatedly, the alarm ends immediately, and the user's target sleeping time on that day is recorded as not achieved.

J. Sleep stamp page



Fig. 36: Sleep stamp page

Fig. 36 is a screen that can be seen by pressing the fourth Sleep stamp page on the Main page of the COCO. The icon shows whether the 'target sleep time' set in Fig. 30 has been achieved. If it is achieved at the desired sleep time, it is marked as a planet icon, if it is not achieved, a black hole icon, and when the sleep mode is not activated, it is marked as a blank. On the right, the percentage of sleep targets achieved over a month is shown as a percentage. When user sleep data is accumulated, one's sleep data may be viewed on a yearly and monthly basis through a button on the screen. If the users press the 'home' icon on the left side of the top, it returns to the Main page.

V. ARCHITECTURE AND DESIGN

A. Overall Architecture

The first module is a front end in the form of a native app. We used Android Studio to implement and design application functions so that users can use their services comfortably. Users can easily execute sleep mode through buttons on the main screen of the application. Even after user turns on the sleep mode, user can get recommendations for videos so that user can get help with sleep. In addition, users can create or change profiles, and they can freely adjust the settings of sleep mode and alarm mode for each profile. In addition, the application converts sleep records into images and shows them to users so that users can intuitively check their sleep patterns.

The second module is a webOS front end. We used it to see how some functions actually appear on our device, StandbyME. Pop-up notification was created using the LS2 API module, a service API provided by webOS. Users can see the notifications that appear when they actually turn on sleep mode on StandbyME.

The third and fourth modules configure a backend that can deliver information to the application. For the backend, a database using MariaDB and a server using node.js were used. The database has a total of three tables: a table that manages accounts that have access to apps, a table that manages profiles within accounts, and a table that manages sleep records for

each profile. To read or write data from the database, use a server created using node.js. Node.js has a code written to allow users to respond to POST requests generated while using the application. Each code can read or write data from the table by creating a query statement based on the parameter of the request.

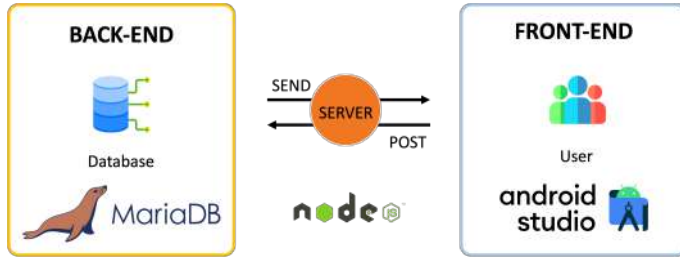


Fig. 37: application Framework.

B. Directory Organization

Our project repository has various files that are used for building our application. Each files directory and usage are explained as follows.

C. Module 1: Android Studio

1. Purpose: We used Android Studio to develop the sleep management application COCO. Android Studio is an integrated development environment (IDE) created by Google based on IntelliJ, and it is easy to develop applications through basic layouts and Java language. In addition, user can implement various functions with SDK tools' libraries and functions, and user can see the actual execution screen through the emulator.

2. Functionality: Android Studio is a development environment that enables the application to communicate with users through UI, communicate with backend servers, and perform major functions of applications. We created a screen that will be shown to the user through front-end work in Android studios and implemented background service functions. In addition, this module can connect the server and DB to the application to get user account data. Other SDK tools provided by Android studios allow user to implement a variety of services.

3. Location of Source Code: SE_SleepWithMe/APP

4. Class components: There are several components in the Android Studio.

1) *MainActivity.java*: MainActivity is the main page of the app that appears after the user logs in. When MainActivity is executed, a ServiceConnection object that can be connected to the sleep mode service is created. On the screen, there are four buttons that allow user to go to another page and a toggle button that allows user to turn on/off sleep mode. Pressing the moon-shaped toggle button at the top where the user can set the sleep mode creates and shows a dialog to check if the

TABLE III: File Directory

Directory	File Name	Module Name
SE_SleepWithMe /App/app/src/main/java/com/example/lullaby/	AccountActivity.java AddAccountActivity.java AlarmActivity.java AlarmSettingActivity.java MainActivity.java MyBroadcastReceiver.java MyDataActivity.java MyService.java SimpleTextAdapter.java SleepActivity.java	Android Studio
SE_SleepWithMe /App/app/src/main/java/com/example/lullaby/data/	AccountData.java GlobalVariable.java Profile.java	Android Studio
SE_SleepWithMe /App/app/src/main/java/com/example/lullaby/login/	LoginActivity.java SignUpActivity.java	Android Studio
SE_SleepWithMe /App/app/src/main/java/com/example/lullaby/network/	LoginNetworkTask.java ProfileNetworkTask.java RequestURLConnection.java SignUpNetworkTask.java	Android Studio
SE_SleepWithMe /App/app/src/main/java/com/example/lullaby/videos/	AsmrFragment1.java AsmrFragment2.java AsmrPagerActivity.java MeditFragment1.java MeditFragment2.java MeditPagerActivity.java MusicFragment1.java MusicFragment2.java MusicPagerActivity.java SelectActivity.java VideoActivity.java	Android Studio
SE_SleepWithMe /App/app/src/main/res/layout/	activity_account.xml activity_add_account.xml activity_alarm.xml activity_alarm_setting.xml activity_asmr_pager.xml activity_login.xml activity_main.xml activity_medit_pager.xml activity_music_pager.xml activity_my_data.xml activity_select.xml activity_signup.xml activity_sleep.xml activity_video.xml fragment_asmr_1.xml fragment_asmr_2.xml fragment_medit1.xml fragment_medit2.xml fragment_music1.xml fragment_music2.xml recyclerview_item.xml sleep_dialog.xml	Android Studio
SE_SleepWithMe /APP/app/src/main/res/values/	arrays.xml colors.xml strings.xml themes.xml	Android Studio

Directory	File Name	Module Name
SE_SleepWithMe /webOS/co- coApp/src/	index.html button.html recom.html off.html	webOS
SE_SleepWithMe /Backend/	coco_db.sql	MariaDB
SE_SleepWithMe /Backend/	coco-server.js	Node.js

user's account is correct. If the button to start sleep mode in Dialog is pressed, the Activity is bound to MyService.

2) *MyService.java*: MyService is a service that runs in the background when sleep mode is turned on. When the bindService function is executed in MainActivity, the onBind function is called and the IBinder object is returned. The returned IBinder serves as an interface that connects the service and the client. The sleepAlert function in the MyService displays a pop-up notification to check sleep status every time the user sets. If the user does not click on the pop-up for 20 minutes, the service is unbound and sleepScreen, the function to switch the activity to the SleepActivity is implemented.

3) *MyBroadcastReceiver.java*: MyBroadcastReceiver serves to process responses to pop-up notifications for sleep mode services. In the onReceive function, when the user presses a button in the pop-up, the intent argument is received and if the action within the intent is 'keep', call the sleepAlert function of MyService. On the other hand, if the action in intent is 'stop', it unbinds MyService.

4) *SleepActivity.java*: In SleepActivity, the activity_sleep.xml screen appears during the user's sleep. And after the user's target bedtime, a function to switch the activity to AlarmActivity is executed.

5) *AlarmActivity.java*: When AlarmActivity is executed, the Ringtone alarm rings repeatedly until the user presses the button. When the user presses the button, the alarm stops, and the true or false value is stored with the date in the achievement variable of the AccountData class. In the center of AlarmActivity, the current date and time are received as currentTimeMillis and displayed.

6) *RequestHttpConnection.class*: It is the class closest to the server in the front end, and it creates and sends POST requests based on the url and content values received as factors. And the information that the server responds to is stored in the form of a string and transferred as a function of the Network Task class.

```
//RequestURLConnection.java
public class RequestURLConnection extends
    Thread {
    public String request(String _url,
        ContentValues _params){

        HttpURLConnection urlConn = null;
        StringBuffer sbParams = new
            StringBuffer();

        if (_params == null)
            sbParams.append("");
        else {
            boolean isAnd = false;
            String key;
            String value;

            for(Map.Entry<String, Object>
                parameter : _params.valueSet()){
                key = parameter.getKey();
                value =
                    parameter.getValue().toString();

                if (isAnd)
                    sbParams.append("&");

                sbParams.append(key).append("=")
                    .append(value);

                if (true)
                    if (_params.size() >= 2)
                        isAnd = true;
            }
        }
        Log.d("asdf","parmas: " +
            sbParams.toString());

        try{
            URL url = new URL(_url + "?" +
                sbParams.toString());
            urlConn = (HttpURLConnection)
                url.openConnection();

            urlConn.setRequestMethod("POST");
            urlConn.setRequestProperty("Accept-
                charset", "UTF-8");
            urlConn.setRequestProperty("Context_
                Type","application/x-www-form-
                urlencoded");

            if (urlConn.getResponseCode() !=
                HttpURLConnection.HTTP_OK) {
                return null;
            }

            BufferedReader reader = new
                BufferedReader(new
                    InputStreamReader(urlConn
                        .getInputStream(), "UTF-8"));

            String line;
            String page = "";

            while ((line = reader.readLine()) !=
                null){
                page += line;
            }
        }
    }
}
```



```

    }

    return page;
} catch (MalformedURLException e) { //
    for URL.
        e.printStackTrace();
} catch (IOException e) { // for
    openConnection().
        e.printStackTrace();
} finally {
    if (urlConn != null)
        urlConn.disconnect();
}

return null;
}
}
}

```

7) *LoginNetworkTask.class/ SignUpNetworkTask.class/ MyDataActivityNetworkTask.class*: It is a medium that connects *RequestHttpConnection* class with other classes and supports asynchronous functions. This class creates a starting point for handing over url and content values to the server. In addition, parsing the string that stores the server's response and storing it in a singleton object can be defined using the *onPostExecute* function.

8) *LoginActivity.java/ SignUpActivity.java/ MyDataActivity.java*: In Android Studio, we enter id and pw into *LoginActivity* so that we can receive user information stored in the table from the server. And if user doesn't have an ID, enter id and pw in *SignUpActivity* to add id to the user table and default profile to the profile of that id. Finally, *MyDataActivity* requests a sleep record for a specific profile to the server, processes the response results, and outputs them on the screen.

5. How and Why we use it: We used Android Studio because it allows simulating apps with a variety of devices such as mobile phones, tablets, and TVs without having own devices. Android Studio has a powerful graphic layout editor function, making it convenient to design user interfaces. In addition, it supports the completeness of the code by supporting various analysis tools that provide refactoring.

D. Module 2: webOS

1. Purpose We used webOS, the OS environment of Standby Me, because we were developing an application exclusively for Standby Me. WebOS is an operating system for Linux-based mobile and TV, and can be easily developed in languages such as html, css, and js. Various functions can be implemented using webOS-only API with good versatility, and user can also see that it works within the device through a TV emulator.

2. Functionality Users can see some of the features of our developed standby me-only application COCO actually work on webOS, which is being used in standby me. In order to communicate with webOS service, we were able

to communicate with the web page by creating a 'webOS Service Bridge()'. The LS2 API module is retrieved and built-in 'com.webos.notification' service API and we used them. I was able to check how the 'Sleep confirmation notification' of the COCO actually looked by replaying some functions of the app created using the service API in 'webOS TV IDE'.

3. Location of Source Code: SE_SleepWithMe/webOS

4. Callback function components: The javascript file was embedded in index.html. I got the webOS API and created a callback function. We created a parameter object and delivered the desired notification phrase so that it could appear in the notification notification. The callback object was continuously declared for 4 seconds each because the retention time of notification notification within webOS was fixed at 5 seconds. The callback cycle was set using the *setInterval()* function and the *clearInterval()* function, and the interval cycle was reset when a certain time was arbitrarily set.

5. How and why we use it: We developed it using webOS, which is used for StandbyME, the target device. We downloaded and used webOS IDE, webOS Emulator, and webOS CLI from webOS TV Development website. API and module were used by referring to the tutorial and reference shown on the webOS docs website.

E. Module 3: MariaDB / Module 4: Node.js

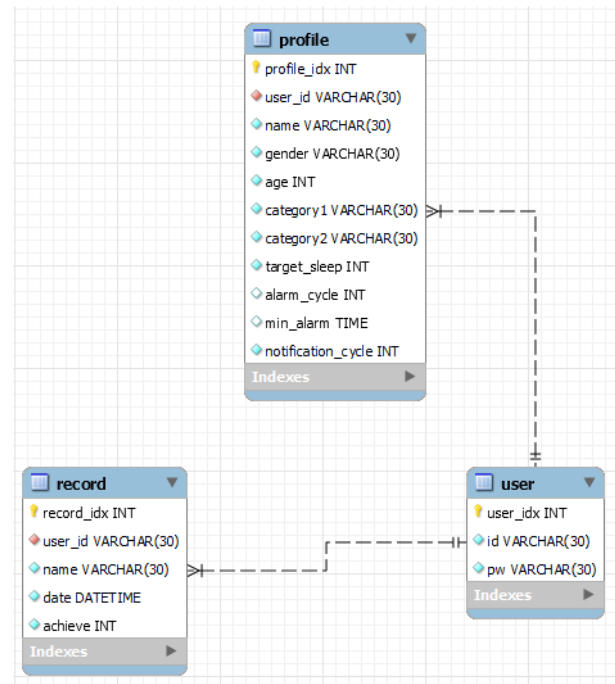


Fig. 38: Database

1. Purpose: The database is necessary to read and write user information required by the application. In this database, various information from user id and pw to sleep

records was stored using sql language. And the server created using Node.js is responsible for connecting modules made of Android Studio and modules made of MariaDB.

2. Functionality: Express.js is one of the web frameworks for Node.js and is designed for application and API development. By using Express.js, we can build servers and process data transmission more conveniently. We can use this to create and change databases, or use sql syntax to control multiple information in tables. Functions that operate differently according to parameters are also implemented by receiving POST requests from the front end.

3. Location of Source Code: SE_SleepWithMe/Backend

4. Class components: There are few methods implemented in this class.

1) *coco-server.js*: The server file contains a code that can process requests sent by Request HttpConnection on the front end. The server recognizes parameters in POST requests such as login, membership registration, and record inquiry, writes sql sentences, and returns the set result value. For normal requests that do not cause errors, the data delivered to the front end is contained in a single string.

2) *coco_db.sql*: The database has a data table that can respond to the server's query statement, and rules about the reference relationship between tables or the format of the data are written. The id in the user table is used as a foreign key in the remaining profile and record tables.

5. How and why we use it: Since storing all data locally becomes an obstacle to app expansion, we decided to build a database outside and get data through the server. We could have created a server using Node.js, but we used Express.js because it was easier and more stable to create a server with Express.js. And to build servers and databases, he joined the Google Cloud Platform(GCP) and purchased virtual instances.

VI. USE CASES

A. Use Case 1: Turn on the application



Fig. 39: StandbyME main page

Fig. 39 is a main screen that appears when the user turns on StandbyME power.



Fig. 40: COCO icon

Fig. 40 is the COCO icon. After installation, the application icon should appear on device.

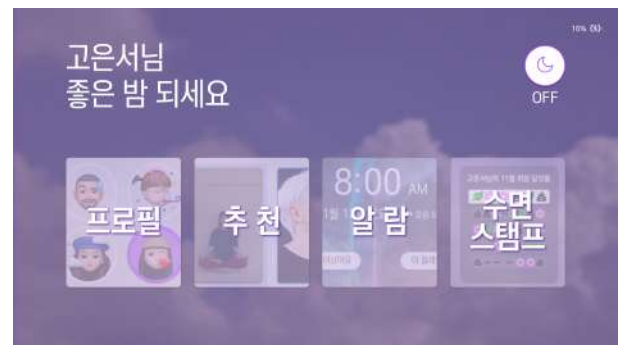


Fig. 41: COCO main page

Fig. 41 is COCO's main page. When the user clicks the moon-shaped COCO button, the app start screen appears like Fig. 41, and COCO can be used.



Fig. 42: COCO home icon

Fig. 42 is home icon in COCO. When the user wants to access the main page from another menu, the user may come to the StandbyME main page through a home button, Fig. 42.

B. Use Case 2: User Registration

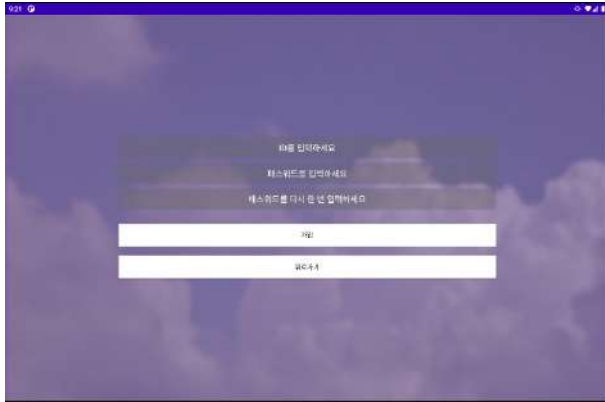


Fig. 43: SignUp page

Users who want to sign up for the COCO can sign up on the Fig. 43 screen by clicking the button at the bottom of the login page.

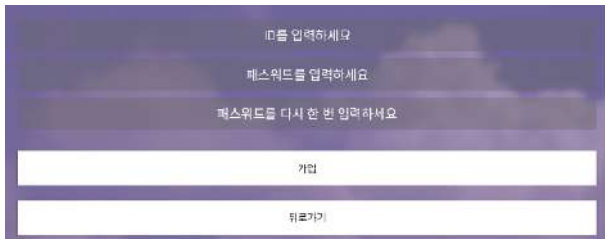


Fig. 44: SignUp Blank

Fig. 44 is the spaces that the user must enter when signing up for membership. Sign in from the main page to the registration page, the user puts input values in each of the ID, password, and password verification columns and presses the bottom complete button to complete the membership registration. If the user presses the back button, the membership can be canceled.

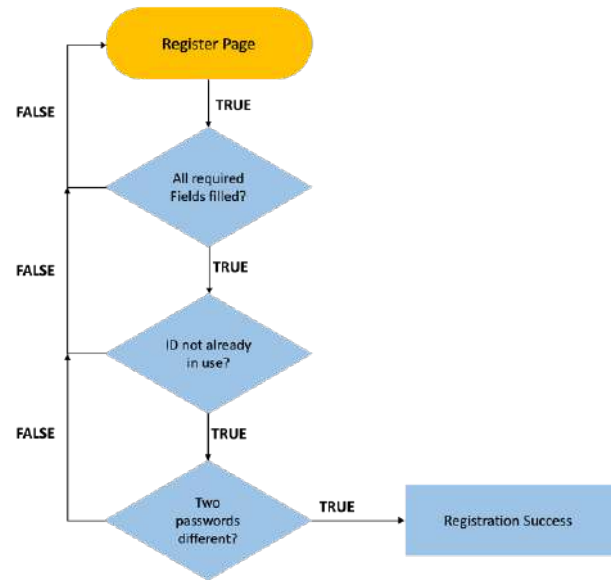


Fig. 45: User Registration Process

C. Use Case 3: Log in

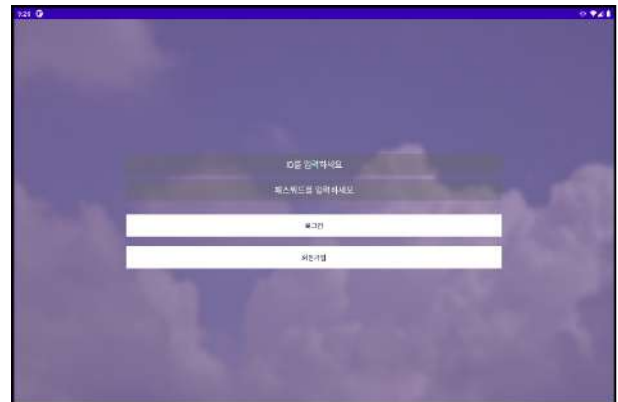


Fig. 46: SignIn page

Users can log in on the Fig. 46.

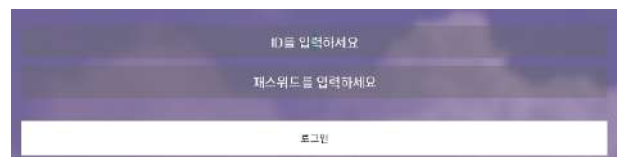


Fig. 47: Login Blank

The user inputs an id in the first ID box, and the user's password in the second password box. If user presses the confirmation button at the bottom, login is executed.

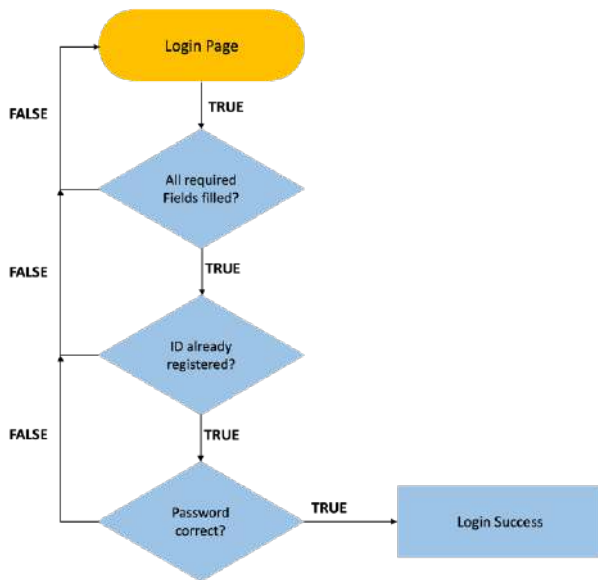


Fig. 48: User Login Process

D. Use Case 4: Sleep mode ON - StandbyME main page

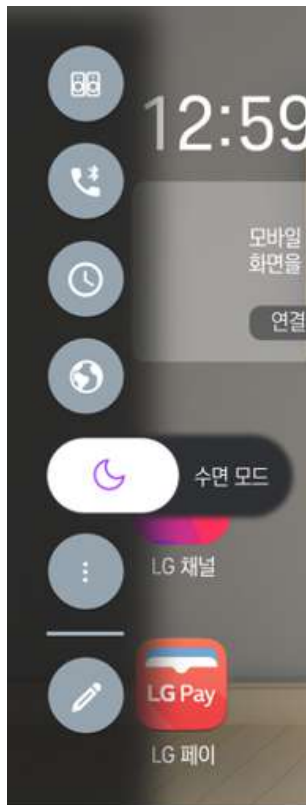


Fig. 49: sleep mode OFF in StandbyME main page

Users can turn on sleep mode outside the app. When the user pushes the left menu to the right with finger on the StandbyME home menu screen, a moon-shaped sleep mode button is generated.

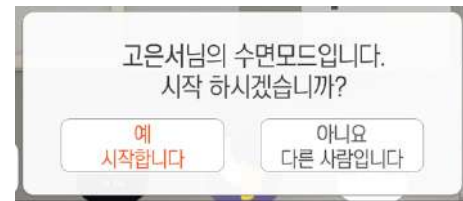


Fig. 50: pop-up notification — turn off the sleep mode

When the user presses the button, a pop-up notification appears to check if the user's profile is correct.

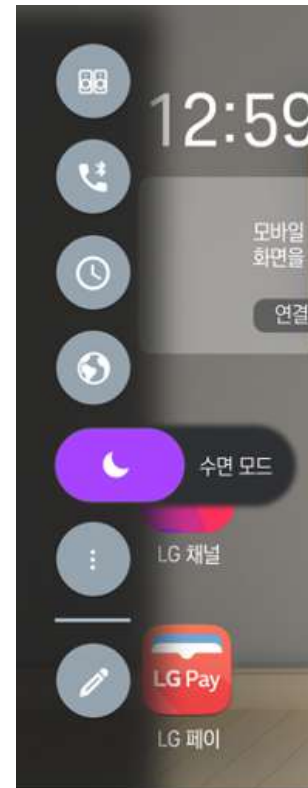


Fig. 51: sleep mode OFF in StandbyME main page

When a user presses the button that the user profile is correct in the pop-up notification, the sleep mode is turned on.

E. Use Case 5: Sleep mode ON - COCO main page



Fig. 52: sleep mode OFF in COCO main page

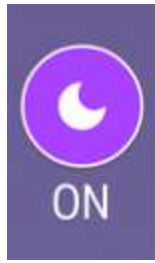


Fig. 53: sleep mode ON in COCO main page

The user may also turn on the sleep mode on the StandbyME main page. When the user clicks the moon-shaped button in the upper right corner, the sleep mode is turned on.



Fig. 54: sleep mode ON

When the sleep mode is turned on, the following notification window appears at the bottom of the COCO.

F. Use Case 6: Sleep check alarm period setting.



Fig. 55: alarm setting button



Fig. 56: alarm setting page

The user enters the alarm setting page and sets a sleep check alarm period setting. Press the alarm button, the third menu on the main page, to see the alarm setting screen.

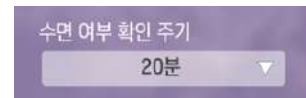


Fig. 57: Sleep check period

The user sets a sleep check alarm period at the lower right.

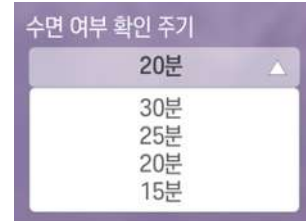


Fig. 58: Sleep check period (open)

When the user presses the triangle button on the right, the user can choose the notification confirmation cycle.

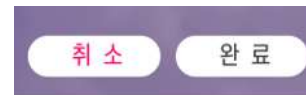


Fig. 59: Done button

If user clicks the desired time among the options and press the completion button at the bottom of the page, the sleep confirmation notification cycle is set.

G. Use Case 7: Sleep check pop-up - not sleeping



Fig. 60: sleep mode pop-up notification

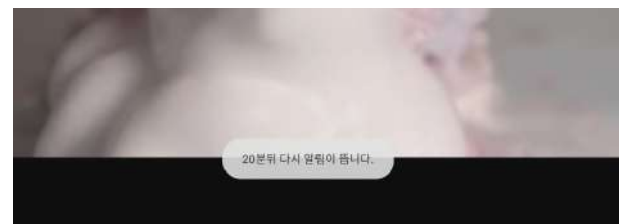


Fig. 61: Press 'No' button

When the user turns on the sleep mode, a sleep confirmation pop-up notification appears in the upper right corner whenever the sleep check cycle set in advance by the user passes. (The pop-up notification appears every 20 minutes before the user sets the sleep check cycle.) If the user wants to continue performing the work currently performed in StandbyME or is not sleeping, user can press the 'No' button in the pop-up

notification. When the user presses the ‘No’ button, the pop-up notification temporarily turns off, and after a certain period of time, a pop-up notification appears to check sleep.

H. Use Case 8: Sleep check pop-up - sleeping



Fig. 62: Screen off pop-up notification

If the user wants to turn off StandbyME or is sleeping in use case 7, the pop-up window may not respond. If the user does not respond to the pop-up window even after the user’s pre-set sleep confirmation notification period has passed, a notification saying, ‘The screen will turn off soon.’ will appear at the bottom of the screen on StandbyME.



Fig. 63: Sleeping screen

After the notification that the screen is turned off appears, it is switched to the sleeping screen.

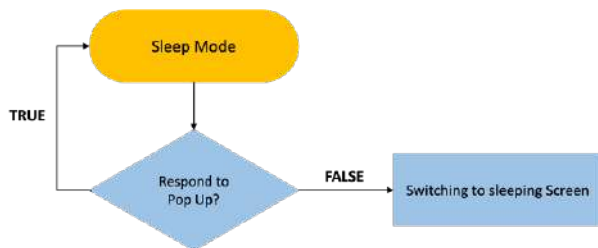


Fig. 64: Sleep mode process

I. Use Case 9: Alarm setting



Fig. 65: Alarm setting button

The user may set the alarm by clicking the ‘Alarm’ button on the COCO main page.



Fig. 66: Alarm setting button

The alarm setting page is as shown in Fig. 65.



Fig. 67: Target sleep time setting



Fig. 68: Alarm setting button



Fig. 69: Alarm setting button

In the alarm setting page, the user may set a ‘target sleep time’, a ‘minimum wake-up time’, and an ‘alarm repetition period’.

J. Use Case 10: Alarm - wake up



Fig. 70: Alarm ringing page

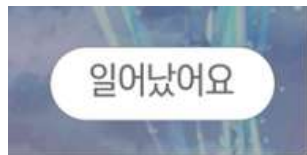


Fig. 71: Alarm ringing page

When the alarm goes off in StandbyME, if the user hears the alarm and wants to end the alarm, the alarm ends by pressing the ‘wake up’ button on the left.

K. Use Case 11: Alarm - sleep more

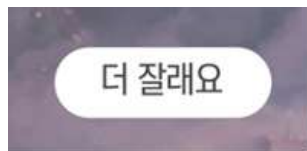


Fig. 72: Sleep more button

When the alarm time set by the user is reached, the alarm goes off on the StandbyME screen. If the user wants to sleep more even after the alarm goes off, press the ‘I want to sleep more’ button on the right. If the user has previously set the alarm iteration cycle to ‘none’, the alarm will no longer sound,

and if the alarm iteration cycle is set, the alarm will sound again after the set time.

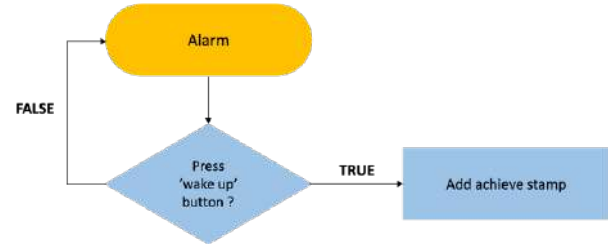


Fig. 73: Alarm process

L. Use Case 12: Sleep mode OFF - StandbyME main page

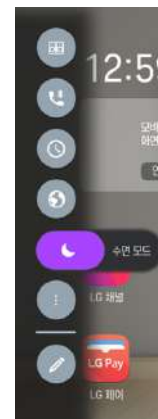


Fig. 74: Sleep mode on in StandbyME main page

The user turn off the sleep mode on the main page of StandbyME. StandbyME home menu screen, user click the moon-shaped button that appears when the user pushes the left menu to the right with his finger.

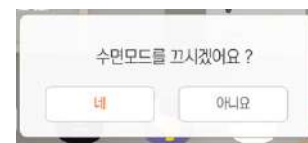


Fig. 75: pop-up notification — turn off the sleep mode

When the user presses the moon-shaped button, a pop-up notification appears to confirm that the user turns off the sleep mode, and when the user presses the ‘Yes’ button in the pop-up notification, the sleep mode is turned off.

M. Use Case 13: Sleep mode off - on the main page of the COCO

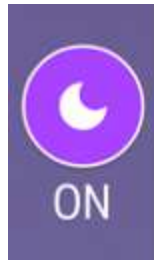


Fig. 76: Sleep mode on in COCO main page



Fig. 77: Sleep mode off in COCO main page

The user may turn off the sleep mode on the main page of the Fig0 COCO application. When the user clicks the moon-shaped button with the sleep mode located in the upper right corner activated, the sleep mode is turned off.

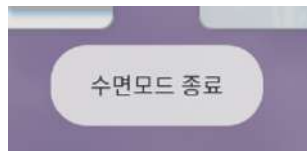


Fig. 78: Sleep mode off

When the user turns off the sleep mode, the following notification window appears at the bottom of the COCO.

N. Use Case 14: Sleep mode off - turned off after alarm

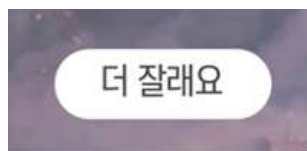


Fig. 79: Sleep more button

When the user wakes up after hearing the alarm and presses the 'wake up' button on the alarm screen, the sleep mode off notification window appears at the bottom and the sleep mode is automatically turned off.

O. Use Case 15: Add a profile



Fig. 80: Profile setting button

When the user clicks the 'Profile' button at the first on the main page, the profile setting page appears.

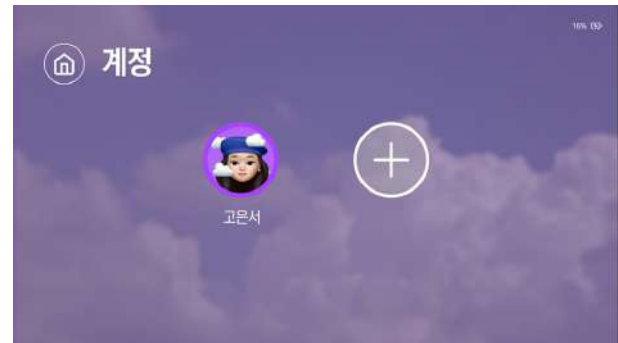


Fig. 81: Profile page

If the user wants to add a profile, press the '+' button on the right.



Fig. 82: Profile adding page

It is a page that appears when a user presses the '+' button. On the corresponding page, the user may set up to three profile names, gender, and categories.



Fig. 83: Name section

First, enter the profile name that the user wants in the upper left section.

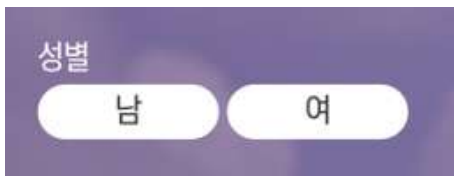


Fig. 84: Gender section

Second, on the lower left section, click the 'Gender' button corresponding to the user.



Fig. 85: Age section

Third, on the lowest left section, the user can choose his age.



Fig. 86: Categories section

Fourth, on the right, click up to three categories of interest to the user.



Fig. 87: Done button

After completing the task, pressing the 'done' button at the bottom of the screen to complete the profile addition.

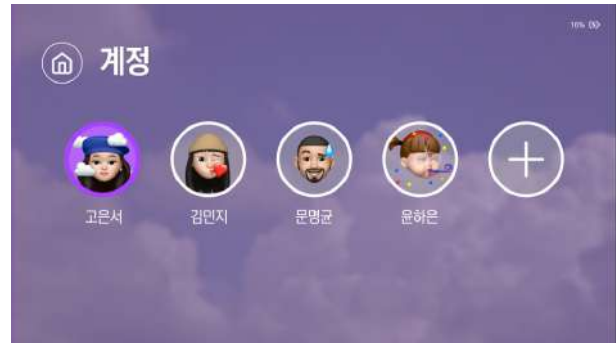


Fig. 88: Profile page - adding complete

P. Use Case 16: Profile change - COCO profile page

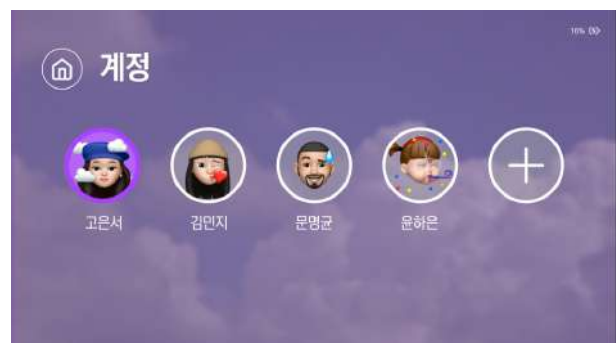


Fig. 89: Before the profile change

If the user wants to use the COCO with an profile other than the existing profile, user can access the profile page that comes out by clicking the 'profile' button at the first on the COCO main page.

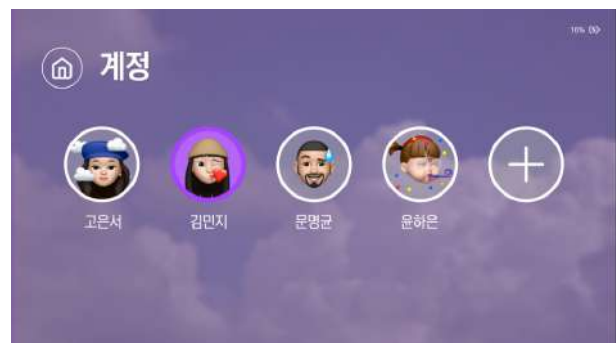


Fig. 90: After the profile change

When the user presses the ‘profile’ button that the user wants to set on the corresponding page, the user of the COCO is set with the corresponding profile, and a purple border is created on the changed profile.

Q. Use Case 17: Change profile

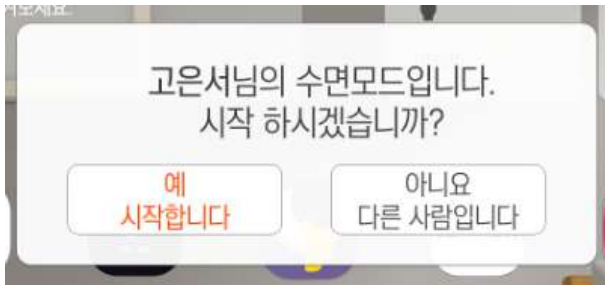


Fig. 91: Profile checking

When the user selects the sleep mode outside the COCO, a pop-up window such as figure 0 appears. When the user presses the ‘No, I’m a different person’ button in the pop-up notification, it goes to the profile setting page. When the user chooses the profile he or she wants to set on the corresponding page, the user of COCO is set on the corresponding profile, and a purple border is formed on the changed profile.

R. Use Case 18: Video Recommendation



Fig. 92: Recommendation button

If the user wants to see a sleep-inducing video, user can press the second button, the ‘recommendation’ button, on the main page of the COCO application.

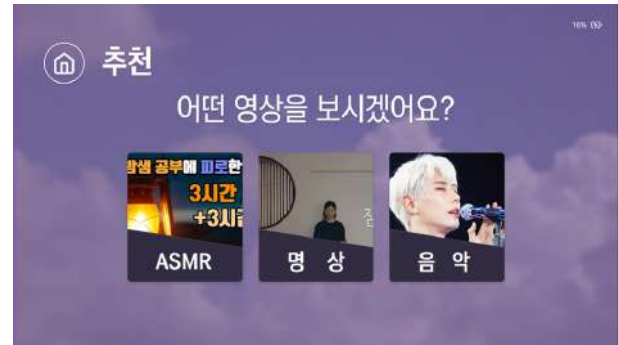


Fig. 93: Recommendation page

The user may select one of three types of videos, ASMR, meditation, and music, from the recommendation page. When the user clicks the desired button among the three video types, the user recommends an video related to the pre-selected preferred category when setting up an account. If user turns the screen to the right, user can check the video more. If the user clicks the video he or she wants to see among the videos recommended by the COCO, the video will be played on the entire screen.

S. Use Case 19: Sleep Stamp



Fig. 94: Sleep Stamp

If the user wants to check his or her sleep goal achievement for a month, click the ‘sleep stamp’ button on the fourth main page of the COCO.



Fig. 95: Sleep data graphic

By selecting the year/month at the top, the user can check the sleep goal achievement value that the user wants to check with the left sleep data graphic.

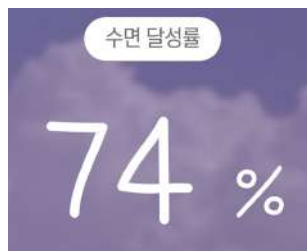


Fig. 96: Data percentage

On the right, the user can see at a glance the achievement of the monthly sleep goal as a percentage.