

HI-SKIN

1st CHAN MIN KIM
*Dept. Information System
 College of Engineering
 Hanyang University
 Seoul, Korea
 han2cmk@hanyang.ac.kr*

2nd SEOK YOUNG KIM
*Dept. Information System
 College of Engineering
 Hanyang University
 Seoul, Korea
 kim2653seok@gmail.com*

3rd HAE RYUNG CHA
*Dept. Information System
 College of Engineering
 Hanyang University
 Seoul, Korea
 haeryung8@hanyang.ac.kr*

4th YU JIN PARK
*Dept. Information System
 College of Engineering
 Hanyang University
 Seoul, Korea
 jinee22@hanyang.ac.kr*

Abstract—Nowadays, with the increasing importance of skin-care, a growing number of busy modern people are trying skincare at home by purchasing beauty devices instead of going to a dermatologist. For these customers, companies such as LG and FOREO are providing beauty devices and applications that can be linked to them. However, managing at home with a beauty device has several weaknesses in place of dermatology. First, there is a lack of communication. Many companies are trying to provide communication such as management methods, but it is still not enough to replace dermatology. The second is that it ends with care. Skin care requires care tailored to one's own skin type even after care, but there is a lack of action. To develop an application that can compensate for these weaknesses, our team is trying to develop a user-friendly application called HI-SKIN that can work with beauty devices for those who have difficulty taking care of their skin. It is managed by a beauty device registered by a user, and at the same time, it enables communication with users by using AI voice recognition technology. It provides communication functions on various topics such as skin care tips, daily conversations, and management methods according to the user's mood in addition to the existing function of providing guidance on how to use devices. Through these functions, we expect users can enjoy the same service as meeting a doctor in person. And also, through additional challenges program, users can take care of their skin not only at the moment they use the device, but also after care or normally. Depending on the user's skin type, the challenge provides a user-friendly skin care function while allowing the user to perform missions such as moisture soothing pack and drinking more than 1L of water per day. In these days, when the desire to be beautiful continues to increase, concerns about good skin are inevitably growing, and our application HI-SKIN will be a good solution in this situation. [1]

Index Terms—skin care, HI-SKIN, User-Friendly Application, AI, Beauty Device, Communication

TABLE I: Task Distributions for Each Member

Roles	Name	Description
<i>Software Developer (Front-End)</i>	HAE RYUNG CHA	A software developer(Front-End) designs applications using languages such as HTML, CSS and React-Native. Key responsibilities include developing responsive interfaces, implementing interactive features and navigation components, and optimizing code for performance. Essential skills and qualifications include knowledge of responsive web design, mobile-first development, problem-solving abilities, and excellent communication skills.
<i>Software Developer (Back-End)</i>	SEOK YOUNG KIM	Backend is a technology that manages servers or databases, areas that users in web applications do not see. The backend is responsible for managing data or running servers so that users can provide the information they want. In other words, the backend is about dealing with what users at the front-end want to-do. As a result, backend developers engage in various development activities, including system component work, API creation, library creation, and database integration.
<i>Software Developer (Machine Learning)</i>	YU JIN PARK	A Machine Learning Engineer is responsible for designing and developing machine learning systems, implementing appropriate ML algorithms, conducting experiments, and staying updated with the latest developments in the field. They work with data to create models, perform statistical analysis, and train and retain systems to optimize performance. Their goal is to build efficient self-learning applications and contribute to advancements in artificial intelligence.
<i>Project Designer (Documentation)</i>	CHAN MIN KIM	A project designer's primary role is to craft a product with a user-centric approach, demonstrating a deep capacity to empathize with and comprehend the user's journey. This pivotal role involves shaping the fundamental structure of a product or service. The project designer is entrusted with fostering seamless communication among team members and fostering collaboration. The project designer's central focus lies in enhancing the product or service's usability, adapting the product's design as necessary to achieve superior outcomes.

I. INTRODUCTION

A. Motivation

According to a survey conducted by market research firm Embrain Trend Monitor in 2022, significant changes in awareness about skincare have been observed. The survey targeted 1,000 adults aged 19 to 59 and examined their perceptions related to skincare. The results showed a decrease in satisfaction with their skin condition from 41.4% in 2019 to 37.8%. Furthermore, 64.2% of respondents reported increased concerns about their skin. This data suggests that many people have become less satisfied with their skin condition recently, leading them to desire healthier and more vibrant skin.

Several factors contribute to this shift in perception. Firstly, in today's era of social media, platforms like Instagram, YouTube, and Facebook provide easy access to information about skincare and beauty. Influencers and beauty experts share product reviews, skincare tips, and personal experiences, inspiring users to take a greater interest in skincare and develop their skincare routines.

Secondly, there is a rising interest in health and well-being. Healthy skin is perceived as a crucial indicator of overall health and well-being, prompting many individuals to pursue better health through skincare. Skincare is not only seen as a means to enhance one's appearance but also as a way to improve skin health.

As a result of these trends, interest in skincare and beauty devices has been on the rise. According to Embrain Trend Monitor's survey, there is a growing interest in home beauty and skincare. 81.2% of women perceive skincare devices as popular, and products like those from LG Pra.L and Medique have been predicted to become more prevalent. These devices are considered cost-effective alternatives that allow for convenient skincare at home compared to visiting dermatologists. However, this increased interest and device usage come with some skepticism and concerns. Users have doubts about the safety and effectiveness of skincare devices, and there are worries about the potential side effects of performing skincare independently. In fact, a significant percentage of consumers have experienced side effects when using home beauty devices. According to a 2019 survey by the Consumer Education Center, 10% of users reported experiencing side effects. This is often due to a lack of information about potential side effects



Fig. 1: LG Pra.L Care

associated with these devices, highlighting a communication gap between users and skincare devices that differs from traditional dermatology practices.

In conclusion, the changing perceptions of skincare and home beauty are the result of various interacting factors, leading to emerging market trends and consumer demands related to skincare devices. Therefore, we have decided to create a more comfortable, safe, and user-friendly app to accompany skincare devices to address these evolving consumer needs. [2]

B. Problem Statement

- 1 Many people are currently experiencing significant difficulties and discomfort while using skincare beauty devices. The reason for this is that skincare beauty devices often fail to provide users with sufficient information about the potential risks and side effects, unlike procedures conducted by professional skincare experts or dermatologists.
[3]
- 2 Consumers often struggle to establish consistent skincare routines even when using skincare beauty devices. This is primarily because these devices typically provide instructions and information specific to the device itself, often lacking personalized advice based on individual users' skin conditions and needs.
- 3 Consumers consider homecare devices as alternatives to dermatological services, seeking a similar experience even if not entirely equivalent. However, currently available skincare beauty devices often fall short in providing users with comprehensive information compared to dermatologists, making users perceive them as somewhat "clinical." Furthermore, when receiving skincare services at a dermatologist's office, the interaction with the dermatologist adds an element of "fun" or engagement to the experience, which is lacking when using skincare beauty devices. Consequently, many consumers desire to acquire enjoyable or informative aspects akin to visiting a dermatologist while using skincare beauty devices.

C. Research on Related Software

A. LG Pra.L Care

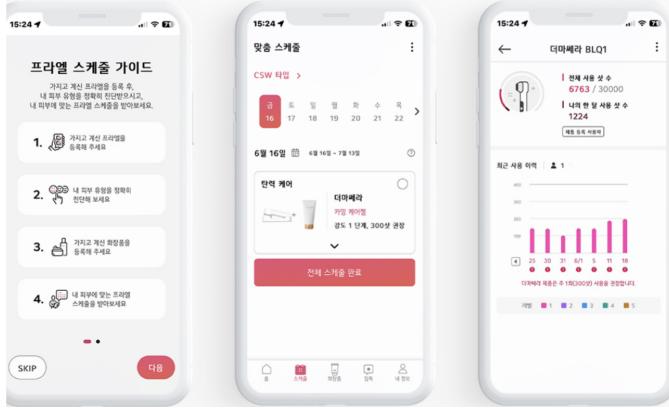


Fig. 2: LG Pra.L Care

LG Pra.L Care is an app created by the South Korean conglomerate LG, designed to enhance the effectiveness of skincare routines for users who own LG Pra.L products. This app offers several key features, including the ability to determine the user's skin type through a 15-question survey and provide recommendations on how to use LG Pra.L products more effectively. Additionally, it offers daily skincare tips based on the weather and air quality and provides a ranking system for skincare products tailored to the user's skin type, assessing their suitability as a percentage match.

B. HWAHAE

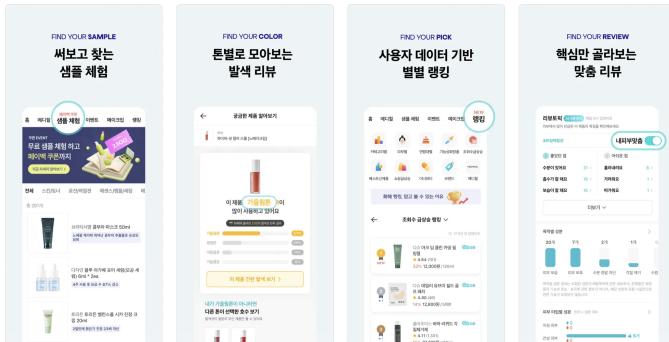


Fig. 3: HWAHAE

HWAHAE is the leading domestic cosmetics app in South Korea, ideal for consumers who are unsure about which skincare or cosmetic products to purchase. This app's primary features include ingredient analysis and user reviews. The review feature mandates users to list both the pros and cons of the products, allowing consumers to gain in-depth insights into the products they intend to buy. Users can also explore popular products by category through the ranking feature,

identifying which cosmetics are currently trending. Furthermore, the "HWAHAE PLUS" section offers beauty-related information, including details about cosmetics and skincare.

C. Glowpick

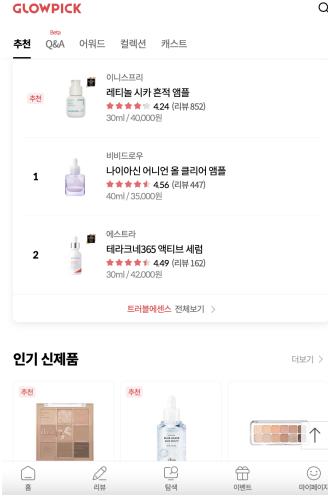


Fig. 4: Glowpick

"Glowpick" is an app that lives up to its slogan of "Finding good cosmetics is a good habit" by offering users cosmetic rankings based on honest product reviews from real consumers. This app not only includes products launched domestically but also registers and sells products from various sources, including roadshops, drugstores, department stores, and even products not officially released in South Korea.

Furthermore, Glowpick provides category-specific rankings for products available in each offline purchasing channel, such as Olive Young, Watsons, LOHBs, and Aritaum. It also offers a wealth of information, including diverse sale details, makeup buying tips, and a review search feature that helps consumers discover great cosmetics through user reviews. Additionally, it provides comprehensive information about the ingredients contained in each cosmetic product.

D. FOREO For You

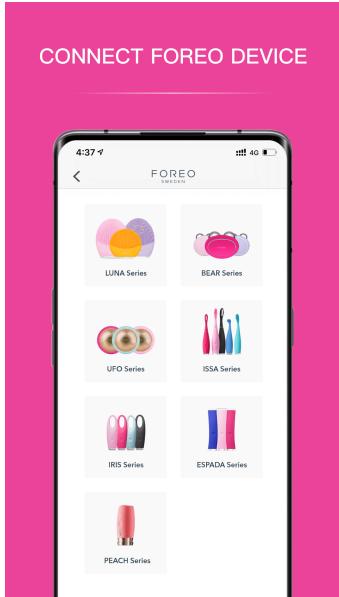


Fig. 5: FOREO For You

FOREO For You is an app that works in conjunction with FOREO's skincare devices, offering skin analysis and personalized skincare guidance. Additionally, this app monitors skin conditions and assists users in using the devices effectively.

E. Clarisonic Mia Smart

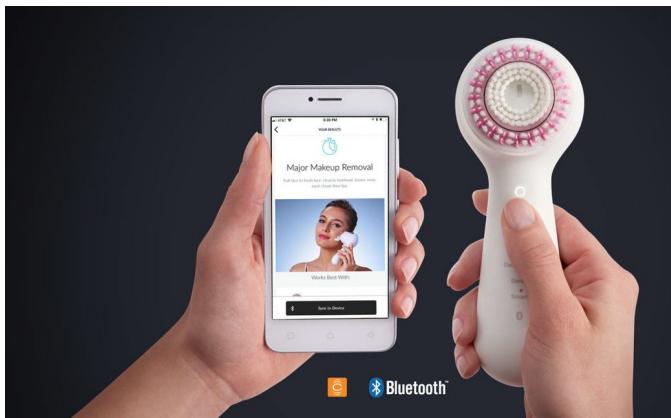


Fig. 6: Clarisonic Mia Smart

Clarisonic Mia Smart is an application that connects to Clarisonic's skincare devices, providing skin analysis and customized skincare routines to users through videos and photos demonstrating how to use the skincare device effectively.

II. REQUIREMENT ANALYSIS

A. Create an Account

If users are new to the app, they can create an account by clicking the "Sign Up" button. After clicking the button, users will be asked to answer a few questions to provide their information and create the account. The following items are the information that is needed for creating an account:

- Name
- Gender
- Nickname
- ID (email)
- Password

After entering all the information, user will be taken to the login page. In addition, user can edit their information on "My Page".

B. Login

Signing in is a crucial step for all members to gain entry to the application, acting as the pathway to harness the complete spectrum of features offered by HI-SKIN. The process of logging in involves inputting the ID and password that were initially provided during the registration phase. This mandatory login procedure constitutes a fundamental element of the HI-SKIN platform, granting users access to a tailored skincare experience and empowering them to proficiently oversee their skin. The authentication mechanism, employing an ID and password, plays a pivotal role in upholding account security and ensuring the protection of confidential information.

C. Skin Type Diagnosis

The app offers a "Skin Type Diagnosis" feature that helps users determine their skin type. Users answer a series of questions about their skin through a survey format. The app then automatically provides information about the user's skin type based on their responses.

D. Cosmetic Recommendations

Users can receive recommendations for cosmetics suitable for their skin type. The app uses data scraped from Glowpick to provide users with the top 3 cosmetics for each skin type, enabling them to engage in skincare beyond the use of beauty devices.

E. Device Registration

Users can register their skincare devices within the app by clicking the "Device Registration" button.

F. Interactive Voice AI Communication

The app enables two-way communication with skincare devices via voice recognition. This interaction allows for daily conversations, addressing concerns, and providing personalized skincare product recommendations based on user preferences.

G. Community

The app offers a community feature, allowing users to communicate with each other. Users can share their skincare concerns and effective skincare tips through the community, making interaction easier with comments and like buttons.

H. AI Interactive Skincare Challenges

The app offers daily skincare challenges tailored to the user's skin type. Completing these challenges can help users improve their skincare habits and routines.

I. Today's Skin Status

"Today's Skin Status" allows users to track their daily skin condition and skincare routines through the app. All recorded data is stored and can be used to generate monthly statistics about skin conditions and skincare practices.

J. My Page

Users can check their skin type diagnosis results, registered beauty devices, and skincare journal for the past week on their My Page. Additionally, there is a feature that allows users to redo the skin type diagnosis.

III. DEVELOPMENT ENVIRONMENT

A. Choices of Software Development Platform

a. Development Platform

1. Windows [4] [5]

Windows operating system is popular among both users and developers for various reasons. For users, it provides a familiar and user-friendly interface, making everyday computing tasks straightforward. Additionally, a wide range of software and games are predominantly supported and optimized for Windows, allowing users to access diverse applications seamlessly.

Developers benefit from Windows through its rich development tools and integrated development environments. Robust tools like Visual Studio support various programming languages and frameworks, facilitating apps and web development and enterprise solution building. Windows provides an environment for developing applications for different platforms and is optimized for game development and graphic

work.

Moreover, Windows exhibits excellent compatibility with various hardware and software, enabling developers to work across different environments more efficiently. This feature is particularly crucial in business and enterprise environments, where many companies adopt Windows to develop and utilize enterprise-level software. Therefore, Windows is acknowledged as a powerful operating system that caters to a broad spectrum of tasks and requirements for both users and developers.

2. MacOS [6]

MacOS is a favored operating system among web and app developers for several reasons. Its Unix-based foundation provides a powerful command-line interface, making it conducive for development tasks. The terminal offers a robust environment for running scripts, installing packages, and executing various developer tools, enhancing efficiency in development workflows. Developers appreciate MacOS for its compatibility with a wide array of programming languages and frameworks. Xcode, the integrated development environment exclusive to MacOS, stands out for creating applications across Apple's ecosystem, including MacOS, iOS, WatchOS, and tvOS. The development environment, combined with the availability of software development kits (SDKs), facilitates the creation of high-quality, native applications.

Moreover, MacOS is highly regarded for its stability and security, essential factors for developers handling sensitive data and applications. The system's stability ensures a reliable platform for coding and testing, while its security features offer a protective environment for sensitive development projects.

For web developers, MacOS supports a variety of web development tools, including popular editors like Visual Studio Code, Sublime Text, and Atom. The operating system's compatibility with web technologies, such as HTML, CSS, and JavaScript, along with its Unix core, provides an ideal environment for web development projects.

Lastly, the integration of hardware and software in Apple products often enhances the development experience. The seamless connection between Apple devices and the ability to test applications on various Apple products contributes to the appeal of MacOS among app developers aiming to create high-quality, well-integrated software for Apple users. Overall, MacOS is a preferred operating system for web and app developers due to its strong development tools, Unix-based

environment, and seamless integration with Apple's hardware and software ecosystem.

3. Android [7]

The Android operating system stands as one of the most prevalent mobile OS platforms in use today. Rooted in the Linux kernel, it is the brainchild of Google, tailored primarily for smartphones and tablets. Its open-source architecture has spurred unprecedented growth, making it the swiftly burgeoning choice among users and developers. This open nature allows for easy customization and integration of advanced functionalities, aligning with the dynamic needs of mobile technology. With over 1.5 billion applications and games downloaded monthly from the Google Play Store, the Android OS is revered for its robust development framework. Users and software developers leverage its power to create a diverse array of applications for a broad spectrum of devices. To support seamless software development, Android furnishes the Android Software Development Kit (SDK), employing the Java programming language. This comprehensive kit comprises a debugger, libraries, a handset emulator using QEMU (Quick Emulator), comprehensive documentation, sample code, and tutorials.

4. iOS [8]

iOS, as the exclusive operating system developed by Apple for its suite of mobile devices, including the iPhone, iPad, and iPod Touch, is renowned for its developer-friendly environment and end-user experience. From a development perspective, the iOS ecosystem offers a robust framework, utilizing languages like Swift and Objective-C within the iOS Software Development Kit (SDK) to create innovative applications.

The closed ecosystem of iOS ensures a more controlled environment for developers, with stringent app submission guidelines and a comprehensive review process, contributing to a high standard of app quality and security within the App Store. The integration of hardware and software is seamless, enabling developers to leverage the full potential of Apple devices and services like Apple Watch, iCloud, and Mac computers, thus providing a unified experience across the Apple ecosystem.

Regular updates to the iOS platform not only introduce new features and enhancements but also address security concerns and bugs, ensuring a stable and secure environment for both developers and end-users. The platform's optimization with the hardware allows developers to create apps

that run smoothly and efficiently, enhancing performance and battery life.

Additionally, the incorporation of Siri, Apple's virtual assistant, allows developers to integrate voice commands and other functionalities into their applications, offering a more interactive user experience. The commitment to user privacy and data security through features such as Touch ID and Face ID underscores the priority Apple places on safeguarding user information.

Overall, the iOS environment provides developers with a combination of reliable tools, a secure and controlled ecosystem, and a high-quality user experience, contributing to its appeal among both developers and end-users.

b. Tools and Language

1. Java

Java is an object-oriented programming language developed by James Gosling of Sun Microsystems and other researchers. It is one of the most commonly used languages in the web application field and is also widely used in software development for mobile devices including Android. Millions of Java applications are in use today as a result of their popularity among developers for over 20 years. It's a fast, secure and reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies. In addition, it is a very suitable language for game development, database processing, big data, and distributed processing. Our team will develop an application, so we will build a backend server using Java.

2. Spring Boot

The Java Spring Framework is a popular enterprise-class open-source framework for creating production-class standalone applications running on Java Virtual Machine (JVM). Java Spring Boot is a tool that helps you develop web applications and microservices faster and more easily using the Spring Framework through three core functions. Automatic configuration, a self-righteous approach to configuration, and the ability to create standalone applications. These features work together to provide tools to help you set up Spring-based applications with minimal configuration and settings. Because of these advantages, our team decided to use the Spring Boot framework to build backend servers in this project.

3. Hibernate [9]

Hibernate is an ORM framework intended to translate between relational databases and the realm of object-oriented development. Hibernate provides a querying interface, using Hibernate Query Language (HQL) or the Hibernate Criteria API. Using hibernate increases productivity because queries can be performed only by method calls without using SQL directly, and it is also excellent in terms of maintenance because it performs the parameters, results, etc. of the DAO related to the table when the table column is changed. Together, Spring and Hibernate are a dynamic duo, capable of simplifying dependency collaboration, reducing coupling, and providing abstractions over persistence operations. Since our team decided to use spring boot for the backend server, we decided to use hibernate, which is well matched with spring boot, to build the database.

4. Jupyter Notebook [10]

Jupyter Notebook is the most widely-used system for interactive literate programming. It was designed to make data analysis easier to document, share, and reproduce. Jupyter originated from IPython and, in addition to Python, it supports a variety of programming languages, such as Julia, R, JavaScript, and C. It also allows the interleaving of not only code and text, but also different kinds of rich media, including image, video, and even interactive widgets combining HTML and JavaScript.

5. Google Colaboratory [11]

Google Colaboratory (also known as Colab) is a cloud service based on Jupyter Notebooks for disseminating machine learning education and research. It provides a runtime fully configured for deep learning and free-of-charge access to a robust GPU. Thus, it can be effectively exploited to accelerate not only deep learning but also other classes of GPU-centric applications.

6. TensorFlow [12]

TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. Its computational model is based on dataflow graphs with mutable state. Graph nodes may be mapped to different machines in a cluster, and within each machine to CPUs, GPUs, and other devices. TensorFlow supports a variety of applications, but it particularly targets training and inference with deep neural networks. It serves as a platform for research and for deploying machine learning systems across many areas.

7. PyTorch

PyTorch is a machine learning library that shows that these two goals are in fact compatible: it was designed from first principles to support an imperative and Pythonic programming style that supports code as a model, makes debugging easy and is consistent with other popular scientific computing libraries, while remaining efficient and supporting hardware accelerators such as GPUs. Our goal with PyTorch is to build a flexible framework to express deep learning algorithms.

8. Docker

Docker was designed in order to simplify the creation, deployment and execution of applications using containers. Containerization allows the user to run applications in a virtual environment by packaging all necessary elements such as files, libraries and other essential components together. Furthermore, containers play a vital role in DevOps processes as an integral part of automated software builds or as part of continuous deployment.

9. Git [13]

Git serves as an effective tool for managing versions, facilitating the seamless integration of changes and updates. However, prior to delving into Git, it's important to grasp the concept of a 'version control system.' Essentially, a version control system captures and monitors modifications made to a file, enabling easy retrieval of any previous iteration when needed. While working on a document, multiple revisions take place, progressing from the initial draft to the ultimate version. Often, files are renamed as 'final,' 'final copy,' 'finalized,' and so on, leading to the replacement of previous versions. This practice can complicate the process of reverting to a specific point in time to understand the alterations made. However, a version control system resolves this challenge. It enables the management of numerous iterations of the same data, facilitating the tracking of changes over time and attributing them to specific contributors. This simplifies the process of reverting to previous or original versions and promptly identifying the responsible individuals for any issues that may arise.

10. JavaScript [14]

JavaScript is the main scripting language for Web browsers, and it is essential to modern Web applications. We present a static program analysis infrastructure that can infer detailed information for JavaScript programs using abstract

interpretation. The analysis is designed to support the full language as defined in the ECMAScript standard, including its peculiar object model and all built-in functions. The analysis results can be used to detect common programming errors – or rather, prove their absence, and for producing type information for program comprehension.

11. React Native [15]

React Native is an open source JavaScript framework for building mobile applications for both iOS and Android devices. It was open-sourced on March 2015 by Facebook, and it's based on the React framework published a few years earlier.

12. Python

Python is a high-level, versatile programming language known for its readability and simplicity, making it an excellent choice for beginners and professionals alike. It supports multiple programming paradigms and offers a vast array of libraries and frameworks, enabling developers to create diverse applications, from web development and data analysis to artificial intelligence and scientific computing. Its clear and concise syntax promotes easy comprehension, fostering rapid development and fostering a robust community that contributes to its continuous evolution and widespread adoption.

13. Bert [16]

The full name of BERT is Bidirectional Encoder Representations from Transformers, which is a language model representation based on self-attention blocks. The main innovation of the model is the pre-training method, which is trained with Masked LM and next sentence prediction to capture the word and sentence level representation respectively. BERT is pre-trained in different language model tasks using existing unmarked corpora. The pre-trained deep bidirectional model with one output layer can reach state-of-the-art performance in many tasks such as question answering and multi-genre natural language inference. The idea is to have a common architecture that fits many tasks and a pre-trained model that reduces the need for labeled data. for a given token, its vector representation is built by summing the corresponding, word, sub-word and position embeddings. These combinations of preprocessing steps make BERT so versatile.

14. ELMo [17]

ELMo is a deep contextualized word representation that models both complex characteristics of word

use (e.g., syntax and semantics) and how these uses vary across linguistic contexts (in order to model polysemy). ELMo extends a standard word embedding model with features produced bidirectionally with character convolutions. The writers in [16] showed that different layers in this deep recurrent model learn different aspects of a given token. Each token converts to a relevant representation using character embeddings. This character embedding representation is then run through a convolutional layer, followed by a max-pool layer. Finally, this representation is passed through a 2-layer highway network [30] before being provided as the input to the LSTM layer. These transformations to the input token allow the model to pick up on character-level features which are helpful for morphologically rich languages such that word-level embeddings could miss and deal with the OOV problem for various NLP tasks. By adding ELMo word embeddings to our model, the model will be able to infer representations for previously unseen words and obtain syntactic information at the morpheme level.

B. Software in use

1. IntelliJ [18]



Fig. 7: IntelliJ

IntelliJ IDEA, JetBrains' flagship Java IDE, provides high-class support and productivity boosts for enterprise, mobile and web development in Java, Scala and Groovy, with all the latest technologies and frameworks supported out of the box. Every aspect of IntelliJ IDEA is specifically designed to maximize developer productivity. Together, powerful static code analysis and ergonomic design make development a productive and enjoyable experience. IntelliJ has powerful recommendations, multiple refactoring and debugging capabilities, and supports quick updates tailored to Java and Spring Boot versions, so we decided to use IntelliJ to develop backend servers.

2. Postman [19]



Fig. 8: Postman

Postman is a downloadable client and web application that was created as a tool to help with the API testing process. It is now a robust platform for API development, with features to support both the building and use of APIs. Postman has tools for documentation, collaboration with teammates or the larger community, and makes it easy to iterate projects and share them. Postman is a helpful interface that lets you view, send, interact with and use API requests. You can easily see if your request worked and what response was returned. Our team will use postman to express Rest API and resolve any inconveniences that may arise when collaborating between the backend and the frontend.

3. Visual Studio Code



Fig. 9: Visual Studio Code

Visual Studio Code (VSCode) is a lightweight, open-source code editor developed by Microsoft, designed for various platforms and primarily used for software development. It supports a wide array of programming languages, providing syntax highlighting, auto-completion, debugging, and language-specific features. Its key strengths include extensibility, allowing users to customize their development environment with numerous extensions. VSCode offers integrated development environment (IDE) features such as code editing, debugging, version control, terminal access, and embedded Git control. It boasts a lightweight installation size, fast startup times, and is supported across Windows, macOS, and Linux operating systems. With a large user community and online support, it fosters collaboration and assistance among users. It is commonly used in web development, application development, data science, and various tech stacks.

4. Expo



Fig. 10: Expo

Expo is a tool chain built around React Native that streamlines the creation and distribution of cross-platform software. In addition to a managed build environment, and tools for testing and debugging, Expo offers a variety of tools and services that may be used to develop, build, and publish React Native applications. The ability to build features and functionality into their applications makes it simpler for developers to create apps that can work on both Android and iOS.

5. Figma [20]



Fig. 11: Figma

Figma has succeeded in bringing together a whole suite of design tools to provide an all-in-one solution. Figma covers just about everything you need to create a complex interface, from brainstorming and wireframing to prototyping and sharing assets. In addition to this, Figma goes beyond the design side of building a product and generates CSS, IOS, and Android code for developers to use.

6. Node.js [21]



Fig. 12: Node.js

One of the more interesting developments recently gaining popularity in the server-side JavaScript space is Node.js. It's a framework for developing high-performance, concurrent programs that don't rely on the mainstream multithreading approach but use

asynchronous I/O with an event-driven programming model.

7. GitHub



Fig. 13: GitHub

GitHub acts as a platform supporting projects utilizing Git, functioning as a remote command center for Git operations. It provides a hub for version control and developer collaboration, operating as a cloud-based version control system. Git and GitHub are commonly used interchangeably for modern software development collaborations, yet GitHub's functionalities extend beyond this scope. To begin, GitHub is the preferred choice for open-source software, granting access to diverse tool source codes used within our team. Moreover, GitHub serves as a repository for identifying issues or bugs in open libraries. Additionally, it boasts various collaborative features: Pull requests enable thorough reviews of work in different Git branches before merging, and GitHub actions streamline the implementation of continuous integration and continuous deployment (CI/CD). Within our team, we employ GitHub actions to monitor team progress and aid in collectively addressing and resolving errors.

8. Notion



Fig. 14: Notion

Notion is a Software as a Service (SaaS) application accessed via the web, operating as a wiki platform. A key benefit is its capability to generate articles in MD file format and provide live updates. With recent enhancements, it has evolved into an invaluable resource for overseeing project details and effectively handling meeting minutes.

9. Overleaf



Fig. 15: OverLeaf

Overleaf serves as an online tool supporting cooperative composition and editing of LaTeX documents. It boasts an intuitive interface specifically designed for the creation of scientific and technical materials like research papers, reports, and theses. Through Overleaf, several team members can work together on a document concurrently, ensuring smooth collaboration and effective monitoring of modifications. Moreover, it integrates pre-installed functionalities for handling references, equations, tables, and graphics, making it a favored option among scholars and researchers. The content of this document was produced using Overleaf's IEEE template.

10. Google Drive [22]



Fig. 16: Google Drive

Google Drive provides two distinct functions. Similar to its forerunner Google Docs, Drive offers online office applications alongside cloud storage, featuring sharing and collaboration capabilities. The recent addition to the platform is a file storage system that synchronizes with a local folder installed by the user, allowing storage and access of various file types through a Google account. The authors analyze the primary functions of Google Drive and evaluate it in comparison to competitors. Furthermore, they explore Google Drive's usefulness for library staff and its role as a collaborative tool supporting students' academic endeavors and education. Our team utilized Google Drive for weekly meetings to monitor the progress of our team members.

11. ChatGPT



Fig. 17: ChatGPT

ChatGPT is an AI-driven application that facilitates immediate interactions with an AI. While GPT-3.5 was trained on data until 2021, GPT-4 has been trained on more current information. ChatGPT has transformed generative AI, providing improved functionalities for activities like creating reports, summarizing articles, addressing problems, and even assisting with coding tasks.

12. Zoom [23]



Fig. 18: Zoom

Zoom, a versatile software platform, serves as a hub for video conferencing, virtual gatherings, and interactive webinars. Through its interface, users can engage in live video and audio communication, fostering seamless online interactions and discussions. It finds widespread application across various domains, encompassing corporate discussions, distance learning, and personal interactions. The platform accommodates extensive virtual gatherings, allowing numerous participants to engage in video calls, utilize chat functions, share screens, and more, ensuring prompt real-time interactions and collaborative efforts. Users can access Zoom through the dedicated app installed on their devices or by using web browsers, providing accessibility across various platforms. Throughout the COVID-19 pandemic, Zoom garnered immense traction due to the surge in remote learning and telecommuting. Nevertheless, its utility extends beyond these realms, serving diverse needs such as professional meetings and everyday communication. Our team opted for Zoom to conduct weekly gatherings.

13. Amazon EC2



Amazon EC2

Fig. 19: Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) offers the most comprehensive and in-depth computing platform to best meet the needs of your workload, with over 700 instances and options for the latest processors, storage, networking, operating systems, and purchasing models. AWS is the first major cloud provider to support Intel, AMD, and Arm processors, the only cloud to support an on-demand EC2 Mac instance, and the only cloud to support 400Gbps Ethernet networking. AWS offers the best price/performance for machine learning training, as well as the cheapest rates per inference instance in the cloud. AWS runs more SAP, higher performance computing (HPC), machine learning, and Windows workloads than any other cloud. Our team will distribute the server that we compiled using the spring boot to AWS EC2 to run the server.

14. NUGU PlayBuilder



Fig. 20: NUGU PlayBuilder

NUGU play is a unit of service through the NUGU platform in response to your request, and you can create Play in the Play Builder. Help companies or individuals with good content to provide their services to NUGU users through Play. Create one complete play by combining the User Utterance Model, which understands the user's speech, and the actions that perform functions based on it. Our team uses NUGU speakers for AI communication functions while users use LG Pra.L devices. Using the NUGU playbuilder, three intents are set: skin problem solving, cosmetics recommendation, and daily conversation, and the answer is output in

conjunction with openai on the backend proxy server that connects to the NUGU speaker.

C. Cost Estimation

While creating HI-SKIN, we efficiently developed it with a cost of \$5.5, which was spent on purchasing the OpenAI API key. There were no additional expenses involved.

D. Task Distribution

TABLE II: Team Members and Their Tasks

Tasks	Name
Frontend Developer	HAE RYUNG CHA
Backend Developer	SEOK YOUNG KIM
UI-UX Designer	HAE RYUNG CHA
AI Developer	YU JIN PARK
Product Designer	CHAN MIN KIM

IV. SPECIFICATIONS

A. Start Page

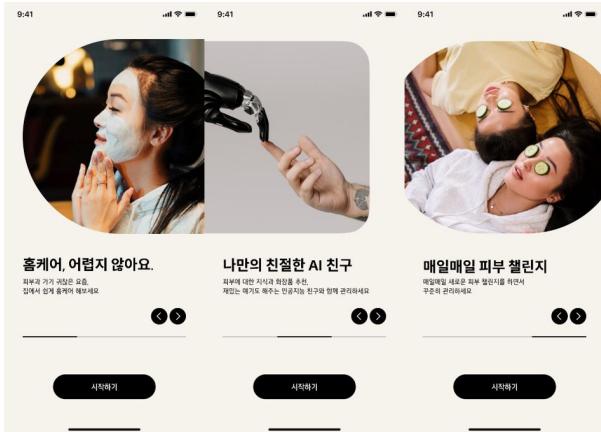


Fig. 21: Start Page

When user download the HI-SKIN app and open it for the first time, user will encounter three sequential screens. In the middle of each screen, a clean image is displayed, accompanied by a brief description of the app below it. There are two buttons below the photo, along with a bar image. You can navigate forward and backward in the display using the button with an arrow. The current page number is indicated by the image below. Upon reaching the final page, pressing the 'Get Started' button with a red border at the bottom instantly transitions the user to the login screen.

B. Log In



Fig. 22: Log In Page

This particular page serves as the platform for all actions associated with user login. Within this interface, users are able to log in using their unique ID and password. Positioned at the screen's top section, there exists a designated area where users can input their ID and password. When the user clicks on that area to enter their ID and password, the keyboard will slide up, covering the area below the login button. It is essential that the ID is entered in the correct format, while the password can contain up to 16 characters. When you enter the username and press 'Next' button, it automatically moves to the password input field. If the user clicks on the 'eye' icon located to the right of the password input field, they can view the password they have entered. Upon accurately entering both the email and password, users can proceed by clicking the login button situated below the password field. Activating this button triggers the transmission of the information in the email and password fields to the backend. The backend then verifies the user's details in the database to ascertain if there's a match and responds accordingly, signaling the success or failure of the login attempt.

When user press 'Complete' button or the login button below, it navigates to the home screen. Additionally, for users without an account, there's an option to create one for login purposes by clicking the 'Sign-Up' button located at the bottom of the screen.

C. Sign Up



Fig. 23: Sign Up Page

Users who do not have an account can create one by selecting the 'Sign Up' button.

- NAME (Up to 20 Characters)
- ID/Email (Correct Email Format)
- Password (Up to 16 Characters)
- Gender (Select Male or Female)

When creating a new account, users can freely click on the 'Log in' button to navigate to the login page.

Successful Sign Up: Upon receiving new information that does not duplicate an existing email as an input value, a notification window will display with the message 'You have successfully signed up,' confirming the successful registration. Once the user successfully signs up as a member, they can return to the login screen using the provided button and log in using the information entered during the sign up process.

D. Home Page

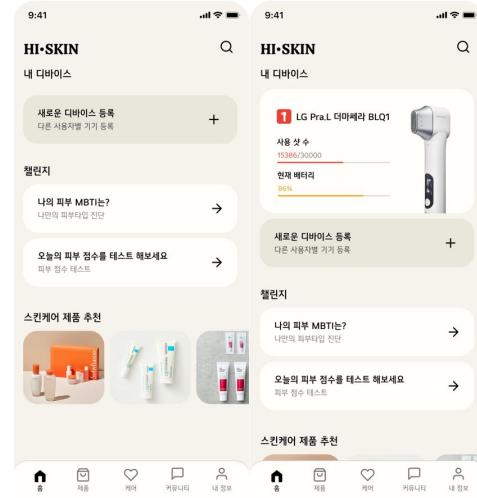


Fig. 24: Home Page

When the device is not registered, there is nothing above the 'Device Registration' button, but when the device is registered, an image of the device appears. User can register a device by clicking the 'Device Registration' button. Pressing the 'What's My Skin MBTI?' button takes you to the skin type test screen. Additionally, clicking the white button navigates to the challenge screen. If you click on the images of recommended cosmetics, it will take you to the cosmetics recommendation page.

E. Device Registration

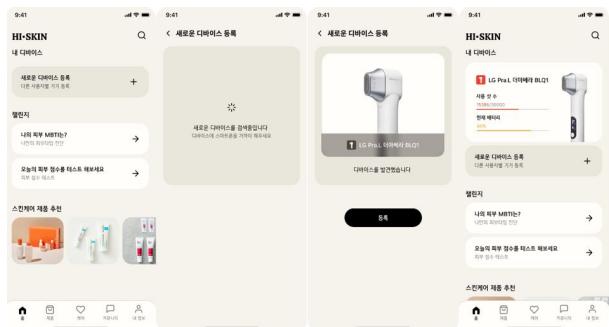


Fig. 25: Device Registration Page

The user can register their owned beauty device on the device registration page. An image appears in the middle of the initial screen, accompanied by a message below it: "Please keep your smartphone close to the device."

When the user brings the smartphone near the device, the device registration process initiates, and the screen displays 'loading' during registration. Upon completion of the registration, the message 'registration complete' appears at the middle of the screen, and after a moment, it automatically transitions to the 'My Device' screen.

On the 'My Device' screen, details about the device registered by the user are shown, and users can return to the home screen by tapping the 'Complete' button located at the screen's bottom.

F. Skin Type Test

Skin type Test is a function that test skin types by selecting answers to skin-related questions and analyzing them. The questions and answers consist of questions selected according to Asian skin characteristics based on the skin type classification method of American dermatologist Leslie Bowman. Questions are largely classified into four categories and each question has two sub-questions. Score according to the user's answer to obtain the sum of the scores of the answers to the two sub-question and give one of the two skin types according to the score. The first question is about dryness and oiliness, and 'D' is given for dryness and 'O' for oiliness. The second question is about sensitivity and resistance, which gives 'S' for sensitivity and 'R' for resistance. The third question is about pigmentation, which gives 'P' for pigmentation and 'N' for non-pigmentation. The fourth question is about wrinkles and tightness, which gives wrinkled skin 'W' and tight skin 'T'. The user would be diagnosed with the final skin type by combining the four diagnosed skin types. It test one of a total of 16 skin types, and information on this is stored in user information, and depending on this skin type, customized skin care solutions such as cosmetics recommendations, challenges, and skin care routine recommendations can be provided.

a. Questions and Answers



Fig. 26: Questions and Answers Page

When the user accesses the skin type test screen, user can view questions for the skin type test. Beneath the question, multiple choices are offered for the user to select the most suitable option for their skin type. Clicking on each answer turns the button a light red color. To proceed to the next question, the user can press the 'Next' button at the bottom of the screen. Users receive points corresponding to their selections: 1 point for choice 1, 2 points for choice 2, 3 points for choice 3, 4 points for choice 4, and 2.5 points for choice 5. By aggregating the scores from answers to two sub-questions, the application test one of the two skin types available based on the user's responses.

b. Skin Type Results



Fig. 27: Skin Type Results Page

The user will be diagnosed with their skin type by selecting the last 4-2 question and pressing the 'Next' button. The HI-SKIN character comes out and guides the user to the skin type and the characteristics of the skin type. The overall features of the skin type are located just below the character. Below that, it guides you on the specific characteristics of the skin type and the characteristics of cosmetics that fit you well. Press the 'Done' button located at the bottom of the screen to move the user back to the main page. Skin types are automatically stored in your information, allowing you to receive customized skin care solutions.

G. Cosmetic Recommendations

Cosmetics Recommendations is a function that allows application to recommend cosmetics that fit the user's skin type. It uses Selenium, a crawling library written in Java on a backend server, to provide cosmetics information obtained from the 'OLIVE YOUNG' site. The best ingredients for each of the 16 skin types are stored in the database, and cosmetics containing good ingredients for each type are searched on the "Olive Young" site to recommend the most popular products. Users will be able to get recommendations for cosmetics that are most optimized for their skin, thereby forming a skincare routine.



Fig. 28: Cosmetic Recommendations Page

The picture above is the screen users see when they enter the cosmetics recommendation section. In a chat format, provide information about the user's skin type, explain the characteristics of the type, and recommend products accordingly.

H. AI Communication

The natural language processing AI model created to continue natural daily conversations with users consists of the following.

a. Data-sets:

The first data used in this model are Korean public corpus, which are Korean Wikipedia, KorQuAD learning/debtset, and Naver movie corpus learning/testset. Pre-processing of this data is necessary to create learning data. To this end, data is preprocessed using morpheme analyzers called KoNLPy, Khaiii, soynlp, and sentencepiece, which are open-source packages.

b. Methodology:

Based on the preprocessed data, a sentence-level embedding model will be learned. The embedding models used here are ELMo and BERT. ELMo is a language model that expresses in probability how natural the word sequence is. ELMo is learned in the process of matching which words will come after the input word sequence. BERT basically uses Self-Attention. BERT masks 15 percent of words when training data enters the input. And it's the way the neural network predicts those hidden words.

c. Evaluation:

Fine tuning is performed to evaluate the embedding results. The previously used data is input and the task of classifying positive/negative polarity is performed. Here, the F1 score will be selected as a major evaluation index to check the performance of the model.

I. Skin Care Challenge



Fig. 29: Skin Care Challenge Page

When a user accesses the challenge page, they will see skincare challenges displayed on the screen. There are questions divided into four areas: skin, moisture, stress, and food management. For each area, there is a checkbox range from 'Very Poor' to 'Very Good.' 'Very Poor' is 5 points, 'Poor' is 10 points, 'Average' is 15 points, 'Good' is 20 points, and 'Very Good' is 25 points. The user can select a challenge they have completed, and upon selection, the respective button turns red. One point is awarded for each selected challenge, contributing to the total score displayed at the top of the screen. The total score ranges from zero to a maximum of hundred. Users can adjust their chosen challenges until midnight the following day. The final challenge score is recorded at midnight and stored in the user database. These scores are

utilized for the current week's skincare assessment and for the past six months' skincare history, accessible on the 'My Page' section.

J. Skin Care

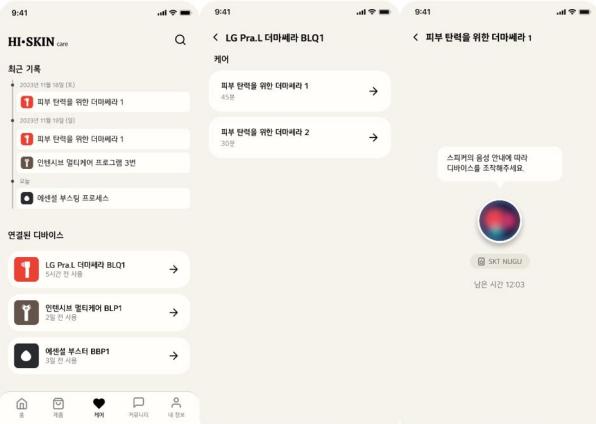


Fig. 30: Skin Care Page

This page provides a voice guide that makes a user feel like the user is getting care from a dermatologist when the user takes care of your skin. By checking the recent records, user can view a list of recent care activities. Below that, there is a list of connected devices, and clicking on each device divides the care activities by device. When selecting each care activity, there is a circle animation in the center that moves according to the voice of the AI, along with displayed connected speaker and remaining time at the bottom.

K. Community

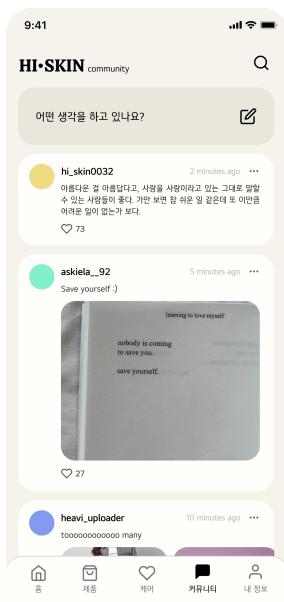


Fig. 31: Community Page

HI-SKIN has a community menu included in the bottom tab. The user can check the entire article by entering the community menu. The article guides useful information such as how to use the device and secret to skin care, and when you click on the article, the information comes up at the bottom of the app. Click the 'To List' button to return to the community screen. When the user presses the heart button next to the article, the article is saved as 'attention article' and can be collected separately.

L. My Page

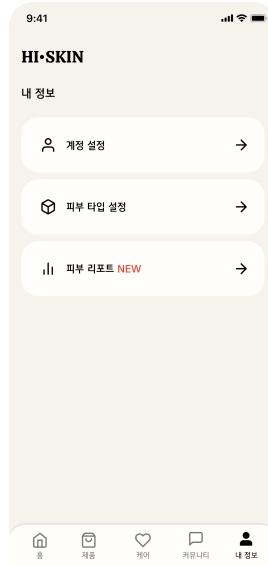


Fig. 32: My Page 1

When the user enters the My Page screen, they can access comprehensive information about their skin.



Fig. 33: My Page 2

Clicking on the 'Skin Type Settings' button takes you to the skin type test screen, allowing you to retake the skin type test. Below that, clicking on the 'Skin Report' button enables you to review the user's skin challenge records for 1 week, 3 months, and 6 months.

V. ARCHITECTURE DESIGN IMPLEMENTATION

A. Overall Architecture

The overall structure of our service consists of an application and AI Nugu speaker.

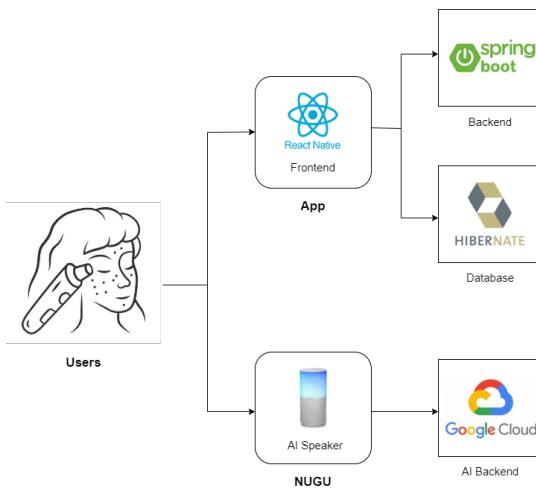


Fig. 34: Application Framework

The first module that makes up the application is the frontend. Our team proceeded with development using React Native and expo. Users can perform skin type tests through the application and receive cosmetic recommendations. You can check the community page where application users can share their own beauty tips, and on the skincare challenge screen, you can check your own challenge score and a graph that analyzes your skin challenge for three months and six months.

The second module that makes up the application is the backend. A spring boot was used to build an application server, and hibernate was used as a database linked to it. The backend serves to perform the tasks requested by the user. Our application performs the functions of signing up and logging in, calculating skin type test results, recommending cosmetics, and calculating challenge scores. A user table, a skin type table, a challenge score table, and a challenge response table exist in the database. When the information is received from the user, the server stores the information in the database.

Another structure of our service is the Nugu speaker. When users take care of their skin using an LG Pra.L device, they can talk to Nugu speakers. The Nugu speaker was developed using the Nugu playbuilder provided by SKT, the AI backend linked to Nugu used Google Cloud Platform, and the language used

node.js. There are a total of three intents in the conversation: solving skin problems, recommending cosmetics, and daily conversations. When the user talks to the speaker, it analyzes which of the three intentions are and responds to the user by outputting the data value received from the backend from the speaker among the various actions accordingly.

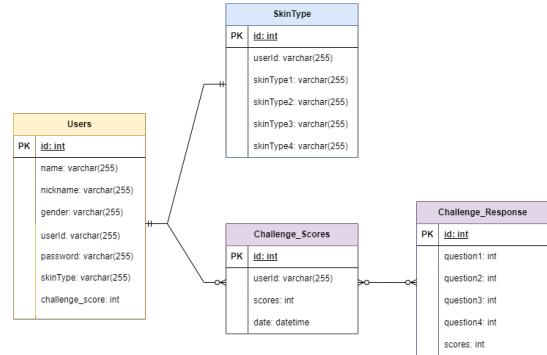


Fig. 35: Database Entity Relationship Diagram

B. Directory Organization

HI-SKIN consists of four GitHub repositories: HISKIN_frontend, HISKIN_backend, HISKIN_AI, and HISKIN_documentation. The HISKIN_frontend repository contains files related to overall design and functions for interacting with the application's users. The HISKIN_backend repository includes files that work with the repository and database. The HISKIN_AI repository contains files related to the AI component of HI-SKIN. Lastly, the HISKIN_documentation repository includes Latex code and a PDF file documenting the project.

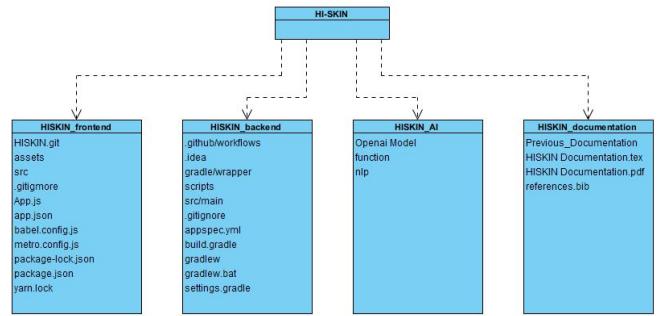


Fig. 36: Repository

C. Module 1: Front-End

- Purpose: We have developed a React Native-based skincare management app for users of LG Pra.L products. React Native allows for cross-platform development and boasts a rich set of libraries. Additionally, we utilized Expo, enabling us to display and modify code in real-time on actual devices. Through this app, we

establish a connection between users and the backend, fetching data from the backend to present it to users in real-time.

2. Functionality: React Native facilitates communication between users and various components, as well as the backend, through the app's UI/UX. The frontend is developed using React, while the backend handles functionalities such as login, sign-up, skin type testing (with options for selecting skin types), skin type results, and challenges (with scores for each category). Throughout this process, the server connects this data with user information and stores it in the database.

3. Location of Source Code: HYU-SE-HISKIN/HISKIN_frontend

Directory	File Name	Modules Used
HYU-SE-HISKIN/ HISKIN_frontend	App.js app.json babel.config.js metro.config.js package-lock.json yarn.lock	HYU-SE-HISKIN/ HISKIN_frontend assets/fonts
	welcome1.png welcome2.png welcome3.png welcomeLayer1 welcomeLayer2 welomeLayer3 bar1.png bar2.png bar3.png AppName_small.svg AppName_large.svg mydevice.svg iconhome.svg iconchallenge.svg iconcare.svg iconcommunity.svg iconmyinfo.svg iconhome_filled.svg iconcare_filled.svg iconcommunity_filled.svg iconnid.svg iconpw.svg IconAccount.svg IconReport.svg IconSetting.svg IconHuman.svg skintyperesult.svg editbutton.svg plusbutton.svg nextbutton.svg morebutton.svg heart.svg communityimage.svg cosmetic1.svg cosmetic2.svg cosmetic3.svg CareMain.svg CareRoutine1.svg CareRoutine2.svg CareRoutine3.svg checkbox_unchecked.svg checkbox_checked.svg	HYU-SE-HISKIN/ HISKIN_frontend src
HYU-SE-HISKIN/ HISKIN_frontend/ assets/images		HYU-SE-HISKIN/ HISKIN_frontend src/components
		HYU-SE-HISKIN/ HISKIN_frontend src/contexts
		HYU-SE-HISKIN/ HISKIN_frontend src/navigations
		HYU-SE-HISKIN/ HISKIN_frontend src/screens
		HYU-SE-HISKIN/ HISKIN_frontend src/utils

4. Class Components:

- App: During the app loading process, all fonts are loaded from assets/fonts. Once this operation is complete, the components necessary to represent the app are rendered on the screen.
- Components: One of the features of React Native is the ability to create components and import them into screens. Building components for frequently used elements such as buttons and images, or for consistently structured components within the app like CommunityBox and ImageLinker, significantly simplifies the development process.
- User: Using ‘useContext’, we can check whether the user has logged in or signed up, confirming whether the username and password information has been sent to the backend. If the information has been successfully transmitted, we can then immediately navigate from the AuthStack to the MainStack.
- AuthStack: Within the stack, there are Welcome, Login, and Signup screens.
- MainStack: Within the stack, there are MainTab as well as DeviceRegistration, Care, Challenge, SkinTypeTest, SkinTypeResult, Cosmetics, and SkinReport screens.
- MainTab: Within the stack, there are Home, Challenge, Care, Community, and Info screens.
- Welcome: It delivers the overall description of the HI-SKIN application in card format. Users can refer to the design, function, and structure of the application as they turn over these cards.
- Login: Users can input their ID and password. When they enter the data, the moment they press the 'Log In' button, the values are sent to the backend. If the backend database contains the information, the login is successful, and the user is directed to the home screen. If the user information is not found, they can navigate to the sign-up screen by clicking on the 'Sign Up' text at the bottom.
- Signup: Users can enter their name, gender, nickname, ID, and password. If they press the 'Sign Up' button, the data is sent to the backend and stored in the database. If there are no issues in this process, the user is immediately directed to the home screen with a successful sign-up message. The user can then log in using this ID and password. By clicking on the login text at the bottom of the sign-up page, they can return to the login screen.
- Home: After a successful login, the user is greeted with the main screen containing an overview of the app's functionalities. Various buttons provide access to device registration, skin type analysis, challenges, and cosmetic recommendations. At the bottom, users can also view community posts. If no devices are registered, users can press the "Device Registration" button to register a device. After registration, the device's appearance is displayed below the app logo. Clicking on the device picture allows users to navigate to the care screen. By clicking on the "Skin Type Test" button, users are directed to the skin type testing page. The "Challenge" button leads to the challenge screen. Clicking on the text "Cosmetic Recommendations" or cosmetic images takes users to the cosmetic recommendations page.
- DeviceRegistration: After the loading page, a message appears saying "The device has been registered," accompanied by an image of the registered device on the screen.
- Care: At the top, users can view a history of the devices they have used. Below that, a list of registered devices is displayed. Selecting a device reveals the care routine for that device, and clicking on the routine allows users to initiate the care process. The central image's animation changes according to the selected care routine, aligning with the chosen care.
- SkinTypeTest: The app displays 8 questions and 4 answer choices from 'utils/skinTypeQuestions.js' on the screen. Upon selecting an answer choice, the choice turns red, and clicking the 'Next' button transitions to the next question. On the final 8th question, clicking the 'Complete' button sends the selected choices to the backend. The app then receives the examination results from the backend, and these results are passed to the 'SkinTypeResult' screen, which is navigated to upon clicking 'Complete.'
- SkinTypeResult: The app displays the conveyed skin type results and corresponding explanations on the screen. Users can use this screen to gain a better understanding of their skin and receive recommendations for cosmetic ingredients suitable for their skin.
- Challenge: Users can measure their daily skin score, divided into four areas, each with checkboxes containing data ranging from 5 to 25 points. Clicking these checkboxes fills the circular line graph at the top in real-time, allowing users to visually check their scores. These scores are transmitted to the backend and recorded daily.
- Cosmetics: When a user completes the skin type test, the results are stored in their user information and utilized for the cosmetic recommendation feature. If there are no skin type results in the backend, it notifies the user of an error. The skin type results are sent to the backend to initiate web scraping using the 'ChromeDriver.' It searches the Olive Young website for key ingredients corresponding to each skin type and provides the front end with the image and name of the first product that appears. The front end presents this data to the user in a chat format to enhance their understanding of the information.
- Community: A screen where users can share simple daily concerns, skin issues, or express themselves through text or attached photos.

- Info: There are buttons that allow users to either take the skin type test or access their skin score report. Pressing the "Skin Type Test" button navigates to the skin type test screen, while clicking the "Report" button leads to the Skin Report screen.
 - SkinReport: Clicking the "Report" button allows users to view graphs of their skin scores over one week, three months, and six months. Clicking the "Duration" text and the buttons for one week, three months, or six months displays the corresponding graph in the white box below. Skin score data is retrieved from the backend upon request.
5. Where It's Taken From: The user interface on the frontend provides a direct gateway for users to interact with the HISKIN service. Users input essential information to access the platform. Furthermore, utilizing the data repository, the interface seamlessly showcases personalized content through an array of visual materials.
6. How / Why you Used it: React offers the advantage of easy installation, allowing for the straightforward use of various modules. Additionally, the abundance of examples and a broad community provided ample reference materials for development. Using Expo allowed us to test our application on both Android and iOS seamlessly. The ability to view and modify the screens in real-time during testing further facilitated our development process. Hence, we chose React and Expo for our development.
- D. Module 2: Back-End**
- Purpose: The backend server receives a request from the frontend, performs work in conjunction with the database, and delivers the value back to the frontend. When a user makes a request through an app, the server performs it, and when a value is given, the value is stored in the database. Our team created a backend server using Springboot, a representative framework of Java used by many domestic companies as a development language, and made Amazon EC2 servers operate in a Ubuntu environment. As the database, Hibernate, which is known to be very compatible with spring boots, is used. Build a backend server by storing information necessary for users to use the application in Hibernate.
 - Functionality: The server performs a function of performing a task requested by the user and storing a value corresponding thereto in a database. When a user signs up for membership, information is stored in a database, and when a skin type test is performed, the user's skin type is stored and cosmetics that match it are recommended. It also performs the function of storing scores and time in the database and sending data of the last seven, three, and six months to the frontend whenever the user performs the challenge.
3. Location of Source Code: HYU-SE-HISKIN/HISKIN_backend
- | Directory | File Name | Modules Used |
|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HYU-SE-HISKIN/HISKIN_backend | .github.workflows
scripts
src/main
appspec.yml
build.gradle | |
| HYU-SE-HISKIN/HISKIN_backend/.github/workflows | deploy.yml | |
| HYU-SE-HISKIN/HISKIN_backend/scripts | start.sh
stop.sh | authController.js
authRouter.js
authService.js |
| HYU-SE-HISKIN/HISKIN_backend/src/main | java/hiskin_hiskin_backend/resources | |
| HYU-SE-HISKIN/HISKIN_backend/src/main/resources | application.yml | |
| HYU-SE-HISKIN/HISKIN_backend/src/main/java/hiskin_hiskin_backend | config
controller
domain
dto
repository
service
util
HiskinBackendApplication.java | SpringBoot Application |
| HYU-SE-HISKIN/HISKIN_backend/src/main/java/hiskin_hiskin_backend/config | CorsConfig.java | Bean Configuration
CorsRegistry
WebMvcConfigurer |
| HYU-SE-HISKIN/HISKIN_backend/src/main/java/hiskin_hiskin_backend/controller | ChallengeController.java
CosmeticsController.java
MyPageController.java
SkinTypeController.java
UserController.java | RestController
RequestMapping
GetMapping
PostMapping
Autowired
HttpStatus
MediaType
ResponseEntity
LocalDate
HashMap
List
Map
Collectors |
| HYU-SE-HISKIN/HISKIN_backend/src/main/java/hiskin_hiskin_backend/domain | ChallengeScore.java
User.java | lombok
persistence
Getter
Setter
NoArgsConstructor |
| HYU-SE-HISKIN/HISKIN_backend/src/main/java/hiskin_hiskin_backend/dto | ChallengeResponseDTO.java
ChallengeScoreDTO.java
SkinTypeResponse.java
UserLoginRequest.java
UserRegistrationRequest.java | Getter
Setter
LocalDate |
| HYU-SE-HISKIN/HISKIN_backend/src/main/java/hiskin_hiskin_backend/repository | ChallengeScoreRepository.java
UserRepository.java | LocalDate
List
JpaRepository
Modifying
Query
Param |

HYU-SE-HISKIN/ HISKIN_backend/ src/main/java/ hiskin_hiskin_backend/ service	ChallengeScoreService.java ChallengeService.java CosmeticsCrawlerService.java SkinTypeTestService.java UserService.java	Autowired Service Selenium HashMap Map PageRequest Pageable Sort
HYU-SE-HISKIN/ HISKIN_backend/ src/main/java/ hiskin_hiskin_backend/ util	ChallengeScoreGenerator.java LoggedInUserHolder.java SkinTypeSearchKeywords.java	ArrayList List Random Component

4. Class Components:

- build.gradle: This is a file that builds the dependencies and settings needed for the spring boot to work.
- appspec.yml: this file is a file that Code Deploy will refer to for deployment to AWS EC2. With the AppSpec file, we can set which files in the project are copied to which path in EC2, and we can also automatically launch the server by specifying a script to perform after the deployment process.
- .github/workflows/deploy.yml: When the file is pushed to the GitHub main branch, the github action flow is activated to push it to the AWS S3 bucket and then perform CodeDeploy
- scripts/start.sh: As a script to run the application, only copy and run the JAR file because we have already built it in the GitHub Actions workflow.
- scripts/stop.sh: Script to exit if the application is already up.
- CorsConfig: This class is a code that resolves CORS errors when working with backend servers and frontend servers. All Get, Post, Put, and Delete methods at the address "http://localhost:8081" are allowed to solve CORS-related problems that arise when the backend and frontend are linked.
- ChallengeController: This class is a file related to the skin care challenge API. When a user performs a challenge, it includes an API that calculates the total score of the challenge and an API that responds to the frontend of the last seven challenge scores.
- CosmeticsController: This class is a file related to the cosmetic recommendation API. It includes an API that inquires the skin type stored in the user DB, searches for a search word matching it on the OliveYoung site, and responds to the frontend with the product name and image url ranked first in popularity.
- MyPageController: This class is a file related to the My Page Skin Challenge cumulative data API. When the user inquires about the skin challenge situation of 3 or 6 months, the response is provided to the frontend.
- SkinTypeTestController: This class is a file related to the skin type test API. When a user proceeds with eight skin type tests, the score is analyzed and the result of one of the 16 skin types is provided to the frontend.
- UserController: This class is a file related to user registration and login API. It includes an API that provides a response to the success or failure of the user's registration and an API that provides a response to the success or failure of the login compared to the user DB at the time of login.
- ChallengeScore: This class is a file related to the "Challenge Score" table. The challenge score db column includes userId, score, and date.
- User: This class is a file associated with the "users" table. Columns in user db include name, nickname, gender, userId, password, skin type, and challenge score.
- ChallengeResponseDTO: This class is a file related to storing the responses of the four challenges.
- ChallengeScoreDTO: This class is a file related to the challenge score and date.
- SkinTypeResponse: This class is a file related to user's skin type
- UserLoginResponse: This class is a file related to login process. It includes userID and password
- UserRegistrationRepository: This class is a file related to registration process. It includes name, nickname, gender, userId and password.
- ChallengeScoreRepository: This class is a file that makes the database accessible using JPA in relation to the challenge score.
- UserRepository: This class is a file that makes the database accessible using JPA in relation to the users.
- ChallengeScoreService: This class is a file related to processing the challenge score. It includes a function that stores the challenge score and date in db whenever the user performs the challenge, and a function that extracts the recent seven challenge scores.
- ChallengeService: This class is a file related to how to calculate the challenge score. When the user performs the challenge, it includes a function that calculates the total score by assigning a score according to the answer. In addition, when the challenge is performed, the function of storing the challenge score in the user db is also included.
- CosmeticsCrawlerService: This class is a file related to how to recommend cosmetics. It includes a function of searching for a skin type from the user's db and searching for a search word matching it on the Olive Young site to crawl the top product name and image url in the popularity order.
- SkinTypeTestService: This class is a file related to skin type testing. It includes a function that finally diagnoses the final skin type by changing the answer selected by the user in the skin type test into a score and assigning skin types for a total of four skin

categories.

- UserService: This class is a file related to the user. It includes a function of storing the skin type in the user DB when the user proceeds with the skin type test and a function of storing the challenge score in the user DB when performing the challenge.
 - ChallengeScoreGenerator: This Class is a file that generates a challenge score.
 - LoggedInUserHolder: This class is a file that stores userID when a user logs in.
 - SkinTypeSearchKeywords: This class is a file containing search keys to be searched on Olive Young sites for each skin type.
 - src/main/resources/application.yml: This class is a file related to the setting for the spring boot to work properly.
5. Where It's Taken From: The backend server works with the database while exchanging values from the front end. HI-SKIN's backend server is currently running on AWS EC2 because it's a good platform to run the server for a small amount of money.

6. How / Why you used it: The backend server performs tasks that need to happen when the user uses the app. It aims to provide satisfactory services to users while storing and importing values in the database. Among many development languages, servers were built using Springboot, a representative framework of Java known to be widely used by Korean companies.

E. Module 3: AI

1. Purpose: There is a function that helps users manage using home care devices, but it does not give users the feeling of being cared for at the dermatologist even at home. Ending the day and communicating with a friendly AI like a friend during a skincare session that refreshes hard feelings can give users the power to welcome the next day in a good mood. Accordingly, we intend to implement a function that recognizes the user's voice through AI, enables natural conversation, and recommends customized cosmetics according to the user's skin type. To this end, OpenAI's chat completion api will be used to recognize the voice with the user and naturally talk to the user, and the AI function will be connected to SKT's NUGU speaker to output the voice of artificial intelligence.

2. Functionality: AI presents an appropriate response using openai's chat completion api based on user utterance information transmitted from the backend. This can be customized to receive responses from only the desired field by limiting its own system to ask only appropriate questions to users. In addition, even though it is the first question of the user, the advantage is that it is possible to induce an appropriate response to the user's request

by arbitrarily writing the previous conversation.

Furthermore, we learned natural language processing models to understand the processes that enable us to understand, generate, and manipulate human language. NNLM, Word2Vec, and FastText were used as basic word embedding models to represent prediction-based distributed presentation. Furthermore, from a task perspective, the performance of each model was measured by applying the Naive Bayes classification, cnn, and lstm models to the text classification task. Dataset was combined from dailydialog, isear, and emotion-stimulus to create a balanced dataset with 5 labels: joy, sad, anger, fear, and neutral. The texts mainly consist of short messages and dialog utterances. Our dataset consists of written dialogues, messages and short stories. Based on the dataset, the model is applied to the multiclass text classification problem and basically consists of data preprocessing, model training and verification, and testing with different inputs. Accuracy, confusion matrix, and f1 score were used as model training and verification.

3. Location of Source Code: HYU-SE-HISKIN/HISKIN_AI

HYU-SE-HISKIN/ HISKIN_AI/ Openai Model	SE_FineTuning.ipynb SE_LangChain_Basic.ipynb SE_OpenAI_API.ipynb	
HYU-SE-HISKIN/ HISKIN_AI/ Function	index.js package-lock.json package.json	
HYU-SE-HISKIN/ HISKIN_AI/ nlp/models 1.Basic Embedding Model	1-1.NNLM.ipynb 1-2Word2Vec-Skipgram	1-3FastText.ipynb test.txt train.txt
HYU-SE-HISKIN/ HISKIN_AI/ nlp/models 2.CNN	2-1TextCNN-ipynb	
HYU-SE-HISKIN/ HISKIN_AI/ nlp/models 3.RNN	3-1TextRNN.ipynb 3-2TextLSTM.ipynb 3-3Bi-LSTM.ipynb	
HYU-SE-HISKIN/ HISKIN_AI/ nlp/models 4.Attention Mechanism	4-1Seq2Seq.ipynb 4-2Seq2Seq(Attention) 4-3Bi-LSTM(Attention)	
HYU-SE-HISKIN/ HISKIN_AI/ nlp/models 5.Model based on Transformer	3-5-2Bert.ipynb	transformer.ipynb

4. Class Components:

- Basic Embedding: This is a folder that contains word embedding models such as NNLM, Word2Vec, and FastText. These models convert words into vectors and apply activation functions to learn the semantic relationship between words.
- CNN: This is a folder that contains convolutional neural network (CNN)-based models suitable for text classification tasks.

- LSTM: This is a folder that contains the Long Short-Term Memory (LSTM) models which can learn more complex sequence patterns with mitigation of gradient disappearance problems and introduction of gate mechanisms.
 - Fine Tuning: This is a folder that contains naive bayes classification, cnn, and lstm models described in the model folder are applied to the multi-class text classification problem.
 - Fine Tuning Data: This is a folder that contains our datasets which are combined from dailydialog, isear, and emotion-stimulus. It is a balanced dataset with 5 labels: joy, sad, anger, fear, and neutral.
 - Traditional ML: This is a file that includes traditional machine learning with scikit-learn, using Naive Bayes classification, random forest, and linear regression classifiers for emotional text classification problems.
 - Fine Tuning CNN W2V Wiki: This is a file that includes CNN deep network with pretrained 300 dimensional word2vec for emotional text classification problems.
 - Fine Tuning LSTM W2V Wiki: This is a file that includes LSTM, Bi-LSTM deep network with pretrained 300 dimensional word2vec for emotional text classification problems.
 - Openai: This is a folder that contains an openai model to respond to the user's utterance.
 - Function: This is a folder that contains AI backend codes which receives data from the nugu speaker, executes machine learning, and sends the result value as a response.
- [10] B. V. e. a. Pimentem, J.F. (2021) Understanding and improving the quality and reproducibility of jupyter notebooks. [Online]. Available: <https://link.springer.com/article/10.1007/s10664-021-09961-9>
- [11] T. Carneiro, R. V. Medeiros Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance analysis of google colaboratory as a tool for accelerating deep learning applications," *IEEE Access*, vol. 6, pp. 61 677–61 685, 2018.
- [12] M. Abadi. (2016) Tensorflow: learning functions at scale. [Online]. Available: <https://dl.acm.org/doi/10.1145/2951913.2976746>
- [13] hxyxneee. (2021) What is git? - velog. [Online]. Available: <https://velog.io/@hxyxneee/git%EC%9D%B4%EB%9E%80-%EB%AC%B4%EC%97%87%>
- [14] S. H. Jensen, A. Møller, and P. Thiemann, "Type analysis for javascript," in *Static Analysis: 16th International Symposium, SAS 2009, Los Angeles, CA, USA, August 9-11, 2009. Proceedings 16*. Springer, 2009, pp. 238–255.
- [15] J. Warén, "Cross-platform mobile software development with react native," 2016.
- [16] M. Affi and C. Latiri, "Be-blc: Bert-elmo-based deep neural network architecture for english named entity recognition task," *Procedia Computer Science*, vol. 192, pp. 168–181, 2021.
- [17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [18] I. IntelliJ, "the most intelligent java ide," *JetBrains [online]. [cit. 2016-02-23]. Dostupné z: https://www.jetbrains.com/idea/# chooseYourEdition*, 2011.
- [19] R. Hyams and T. Pilko, "'you could use the api!': A crash course in working with the alma apis using postman," *Code4Lib Journal*, no. 54, 2022.
- [20] F. Staiano, *Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop*. Packt Publishing Ltd, 2022.
- [21] S. Tilkov and S. Vinoski, "Node. js: Using javascript to build high-performance network programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.
- [22] T. O. Gallaway and J. Starkey, "Google drive," *The Charleston Advisor*, vol. 14, no. 3, pp. 16–19, 2013.
- [23] L. Kohnke and B. L. Moorhouse, "Facilitating synchronous online language learning through zoom," *Relc Journal*, vol. 53, no. 1, pp. 296–301, 2022.

5. Where It's Taken From:

6. How / Why you used it:

REFERENCES

- [1] L. Zhang, A. Adique, P. Sarkar, V. Shenai, M. Sampath, R. Lai, J. Qi, M. Wang, and M. A. Farage, "The impact of routine skin care on the quality of life," *Cosmetics*, vol. 7, no. 3, p. 59, 2020.
- [2] T.-W. Kim, C.-H. Lee, H.-J. Min, D.-D. Kim, D.-H. Kim, S.-Y. Park, and H.-W. Kim, "Skincare device product design based on factor analysis of korean anthropometric data," *Cosmetics*, vol. 9, no. 2, p. 42, 2022.
- [3] Song, "Rising demand for 'beauty devices' in the united states, even influencing beauty tech," *Cosmetics*, 2022.
- [4] O. C. Novac, M. Novac, C. Gordan, T. Berczes, and G. Bujdosó, "Comparative study of google android, apple ios and microsoft windows phone mobile operating systems," in *2017 14th international conference on engineering of modern electric systems (EMES)*. IEEE, 2017, pp. 154–159.
- [5] W. Stallings, "The windows operating system," *Operating Systems: Internals and Design Principles*, 2005.
- [6] P. Sherry, *Foundation Mac OS X Web Development*. Apress, 2013.
- [7] R. Singh, "An overview of android operating system and its security," *int. journal of Engineering Research and Applications*, vol. 4, no. 2, pp. 519–521, 2014.
- [8] B. Wukkada, R. Nambiar, and A. Nair, "Mobile operating system: Analysis and comparison of android and ios," *International Journal of Computing and Technology*, vol. 2, no. 7, pp. 273–276, 2015.
- [9] P. T. Fisher and B. D. Murphy, *Spring persistence with Hibernate*. Springer, 2010.