Magicool

Keeping food fresh for early morning deliveries

Chung YouBeen
dept. of Information Systems
Hanyang University
Seoul, Republic of Korea
youbeen0308@gmail.com

Lee JunYoung
dept. of Information Systems
Hanyang University
Seoul, Republic of Korea
leewnsdud123@gmail.com

Park SeokWon
dept. of Information Systems
Hanyang University
Seoul, Republic of Korea
xylitolfrog@gmail.com

***Abstract*—Magicool is a mobile application service designed to address the concerns people encounter when using early-Morning deliveries: ensuring the preservation of refrigerated food that being delivered. This platform empowers individuals to make sustainable choices while enjoying the convenience of courier services and keeping their food fresh. magiCool is also designed to address the environmental concerns that emerge during the courier delivery, since waste such as ice packs are generated during courier delivery to maintain the freshness of food. The magiCool service offers a comprehensive solution for users by combining the following key features: It encourages sustainable courier delivery practices by reducing unnecessary packaging, Users can seamlessly select eco-friendly delivery options, track their packages, and minimize the environmental impact of their shipments. With our service, people can utilize AI speakers to receive voice notifications for estimated time of arrival of goods confirmations, and generative AI technology to offer healthier alternatives to the food user order. magiCool is a service committed to promoting sustainability, reducing waste, and enhancing the overall user experience in terms of refrigerator. By utilizing this innovative service, users can actively contribute to a greener planet while enjoying the benefits of fresher food, streamlined courier delivery services, and ease of refrigerator management.**

***Index Terms*—Early-Morning deliveries, Generative AI, Eco-friendly**

TABLE I
ROLE ASSIGNMENT

Roles	Name	Role Description
Development Manager, User	Lee JunYeong	Development Manager is responsible for creating a clear project plan. This involves defining project goals, objectives, scope, and timeline. He ensures that the project development process is being done well, and makes sure that testing and implementation of the product is well deployed. Users play an indirect role in shaping the app's requirements. Their needs and preferences provide the foundation for what features and functionalities the app should have. Developers and designers rely on user feedback and market research to determine the app's scope. Users are often the primary source of feedback during the app's development and after it's released. Developers use this feedback to refine and improve the app. Also, When users encounter issues or bugs in the app, they can report these problems to the development team. This feedback is essential for resolving issues and ensuring the app's stability.
Software Developer, Customer	Park SeokWon	Software Developers are responsible for designing the overall architecture of the mobile app. They decide on the technology stack, frameworks, and the overall structure of the app. Writing code is a primary responsibility. Developers use programming languages to create the app's functionality. They need to ensure that the code is efficient, maintainable, and follows best practices. Also, Software Developers implement the user interface (UI) and user experience (UX) components of the app. This involves creating responsive and visually appealing layouts. In addition, Software Developers are responsible for testing the app thoroughly to identify and fix bugs and issues. Customer identify what they want to implement with the app when it comes to refrigerator management, and further identify and present what functions should be added to make refrigerator management more convenient and efficient. Such preferences of customers' become the main objective for mobile app service.
Product Designer, Document	Chung YouBeen	A Product Designer is responsible for design- ing applications to provide the most efficient interface from the user's point of view. She designed services to meet the needs of customers. She uses Figma, a UI design tool, to produce high-quality results. She works with developers to think about how to reflect users and deliver functions most efficiently to them. Product designer must focus on the usability of a product or service and modify the design of a product to obtain better results. She is responsible for communicating smoothly with other team members and create collaboration. As our project progresses, she writes and organizes all the documentation. (other references, research documentation required for the service)

I. Introduction

Since the development of the first refrigerator, refrigerators have been an integral part of human life. Today, refrigerators have grown in importance to the point where it is almost impossible to find food without them.

The era of ”one household, one appliance” has come to an end, and the era of ”one activity, one appliance” has begun. Instead of using a single appliance for a variety of activities, the trend is to purchase additional appliances for specific purposes. In the past, it was common for households to have only one major home appliance, such as a television and a refrigerator, which are essential for daily life. In recent years, as Korean consumers’ incomes have reached the level of the world’s most advanced economies and the electronics industry has made it possible to introduce low- and mid-priced appliances through technological advancements, more and more consumers, even those who are not wealthy, are purchasing multiple appliances with the same function.

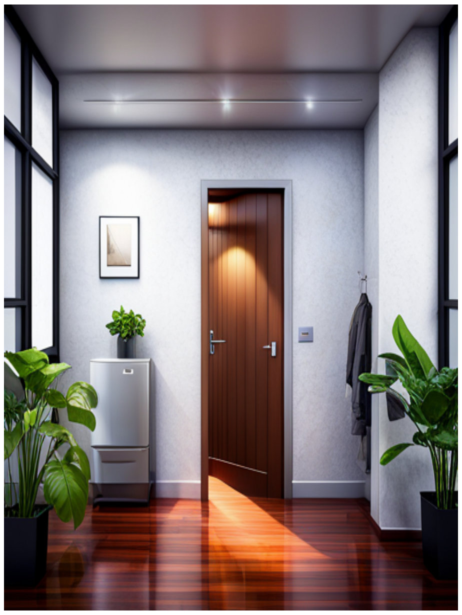


Fig. 1. The built-in refrigerator of the future (near the front door, with the refrigerator door opening into the hallway so that the delivery person can put the package in.)

A. Motivation

The advent of smart home technology has brought about a paradigm shift in the way we interact with our everyday appliances. Refrigerators, once considered a humble and indispensable part of our lives, have now been reimagined as sophisticated devices in the era of LG’s smarthome services. As part of this transformative journey, the development of the Freshkeeper application emerges as a remarkable innovation, one that promises to enhance the way we manage the contents of our refrigerators.

There has been an explosion in the use of fast delivery services for perishables, a concept known as early morning delivery, where customers want to receive fresh goods in the morning. As a result, there are two problems with having the products placed at your doorstep, just like a typical ”home delivery”. The first is the huge amount of packaging and waste generated to maintain freshness. ex. plastic boxes, ice packs, dry ice, gel packs, etc. Secondly, the hallway at the entrance of an apartment building is a common space in the building, and keeping items there for a long time creates problems with fire safety and emergency evacuation routes. In this idea, our team would like to present magiCool mobile application service, which combines a new design of built-in refrigerator hardware with the need for a mini-fridge that can be stored wherever refrigeration is needed, as a way to reduce the waste generated by early morning deliveries.

B. Problem Statement

- Early morning delivery generates a lot of garbage.
  When I look at the trash I collect for recycling, most of it is shipping waste. In particular, there’s a lot of plastic cushioning (aka bubble wrap) used to keep items safe. Recently, there are a lot of insulation materials and Styrofoam boxes. This is due to the delivery of fresh foods such as fruits, meat, and vegetables. According to the Korea Integrated Logistics Association, 8.59 million parcels were delivered to someone’s doorstep on a weekday in 2018. The average number of packages a person receives in a year is 49. With the rapid expansion of the e-commerce market, the number of couriers has increased dramatically. The problem of excessive packaging has become more serious with the proliferation of fresh food delivery, which requires more packaging than regular delivery. The amount of packaging consumed for an order of four to five food items. Frozen products are packed in Styrofoam, regular food in paper boxes, and refrigerated food in insulated bags. Fresh food delivery relies on fast delivery to preserve the freshness of the food. You can place your order the day before and have it on your doorstep the next morning. Because of the fast delivery, people often order small amounts of items on a case-by-case basis. Individually packaged, individually shipped items create more packaging waste. Not only are frozen and refrigerated foods packaged separately, but additional packaging materials such as ice packs are also considerable.
- Leaving personal items in the hallway in front of the front door is dangerous.
  Leaving items in apartment entrance hallways is generally prohibited, and there are several reasons for this. Items can impede evacuation or block emergency escape routes in the event of a fire, and they can be a nuisance to other residents. Additionally, since the entrance hallway is a common area of the building, items can make it difficult to maintain cleanliness, as well as cause visual clutter. As a result, it’s important to follow building rules and avoid leaving items in the entrance hallway for safety and cleanliness reasons.
- Shipping system inconvenience

Fig. 2. Growing waste in shipping

Fig. 3. Eco-friendly shipping bags that are not reclaimed by businesses

Currently, domestic delivery services such as Ecommerce place an order after a user places an order, load the order history, and send an ETA notification to ensure it arrives safely at the customer's address. However, real-time tracking is difficult, so sometimes real-time delivery tracking is inaccurate or delayed. This can create uncertainty for customers about when they'll receive their goods. In the early morning delivery case, if the item is not picked up on time, it may be left unattended until the user returns home.

- Notification and Reminder
  Timely delivery notifications and reminders is considerable in user-friendly aspect. Inform upcoming expiration dates have to be available for users.
- User Interface and User Experience
  User-friendly interface and clear visualization of advice is another problem we got.

C. *Objectives*

Our service was designed to target consumers who frequently use early morning delivery services, with the following key features. Let's delve into how these objectives are achieved and the benefits they bring to users.

- Eco-friendly in shipping
  With the magiCool, it allows for refrigeration without the waste that comes with ice packs. This is especially helpful for people who make breakfast with early morning deliveries and for those who receive a fresh food delivery and don't want to move it to the main refrigerator in the house. Later, We will design also including features to minimize power consumption in that built-in mini-fridge.
- Convenience
  magiCool provides a convenient delivery service system. People can intuitively check the scheduled delivery time through the AI speaker. And "What time is the delivery scheduled to arrive today? " in a dashboard format, and the delivery schedule can be viewed in a calendar format. In addition, it provides a note function for items in the refrigerator, a function that recommends healthy substitutes for ordered items with generative AI, and a function to bookmark frequently ordered items or notify you if user have not ordered for a long time.

D. *Research on Related Software*

1) Samsung Family Hub: Samsung Family Hub is a part of Samsung's smart home ecosystem and focuses on enhancing user refrigerator experience. Family Hub offers a refrigerator management application which allows user to monitor and manage the contents of refrigerator in conjunction with the Samsung refrigerator.

- Integration with Smart Refrigerators
  Connect with the Samsung Family Hub refrigerator to check and manage the refrigerator's contents in real-time.
- Ingredient Management
  Track the ingredients inside your refrigerator and receive notifications about expiration dates.
- Recipe Recommendations
  Based on the refrigerator's contents, it offers recipe recommendations to provide cooking ideas.

2) Whirlpool KitchenAid: Whirlpool offers a refrigerator management application that works with Whirlpool and KitchenAid appliances. The application focus on enhancing food preservation and offering convenient features.

- Refrigerator Status Monitoring
  Monitor the temperature and humidity of the refrigerator to maintain freshness.
- Expiration Date Tracking
  Track the expiration dates of food items in the refrigerator and provide alerts.
- Recipe Management
  Search and manage recipes based on the refrigerator's contents.

3) LG ThinQ: LG ThinQ is a broader application that integrates various LG smart devices, including refrigerators.

- Smart Home Management
  LG ThinQ allows you to manage LG smart devices and refrigerators in one place.
- Food Management
  Monitor the contents of your refrigerator and receive notifications about expiration dates.
- Energy Savings
  Optimize and save energy consumption through refrigerator settings.

II. Requirements

A. Signup

Users must create an account, which is a unique ID, to use our service. You can also easily sign up and log in through SNS social login, Kakao, Naver, and Google Login APIs.

B. Mobile Application

The primary requirement is the development of a mobile application specifically designed for the Android OS. The purpose of this application is to reduce disposable items used for early morning deliveries and keep ingredients fresh to provide users richer life.

C. Login/Logout

Login allows users to access their personalized accounts or profiles within the app. It enhances user experience and ensures the security of user data. Logout ends user's current session and prevent unauthorized access of the user account, protecting user privacy.

D. Settings

Setting allows user to customize their preferences when using mobile app to enhance their user experience.

E. Tabs

There must be a 4 navigators at the bottom of the display each of them is named 'Main', 'Devices', 'Report', 'Setting'.

F. Coupang API

Coupang API helps our service implement several functions: notifying delivery date/time estimation, loading personal purchase history, etc.

G. Refrigerator Registration

The application should have the capability to retrieve information related to LG refrigerators from a central server. Alternatively, users should be able to manually register their refrigerators within the application.

H. Adding Items

The core functionality involves the addition of items to be stored within the user's refrigerator. The application should present a predefined list of commonly stored items in refrigerators, including essential details such as item name and expiration date. Additionally, users should be able to manually input items and their associated information if they are not found in the predefined list.

I. Utilizing Generative AI ChatGPT

We added the option to load the user's order history and select healthy substitutions to create a ”Healthy Dishes Generator” function. To do this, we get a KEY value that utilizes Open AI's API. You can also input the items that the user has purchased to create a healthy meal recipe, or you can use it as a function that learns from the user's order history and recommends healthier alternatives.

J. Light Detection System

Photoresisters, a sensor which change its resistance when get light, are utilized to apprehend the volume of meterials inside the refrigerator. Approximate volume measurement is available through the photoresister installed step-by-step, such as the step: 0, 25, 50, 75, 100.

- Sleep Mode
  Sleep mode is a mode that can reduce energy consumption by using this light detection system. If 0 condition of the light detection system lasts for a certain period of time, it can be confirmed as additional energy consumption for cooling is not required anymore because the refrigerator is empty. By activation sleep mode, users can contribute to environmental preservation and also reduce the cost they have to spend.

K. Item information modification

Users must have the option to modify the information of any specific item that they have added to their refrigerator. This editing feature ensures that the data remains accurate and up-to-date.

L. Lock System

The lock system consists of a simple four-digit password which is shared by user and delivery rider. This password will be provided so that drivers can check it with the request section of delivery app, such as the password on the front door of the apartment.

M. Notification

The application includes a notification system that alerts users to the upcoming expiration dates of items stored in their refrigerators. Users can customize their notification preferences, including setting notification intervals, such as receiving alerts one week prior to an item's expiration date.

N. Providing Information

The application serves as an information resource, offering valuable insights and tips to assist users in effectively managing the contents of their refrigerators. This information can cover various aspects of refrigerator management, from food safety guidelines to best practices for organization and storage.

III. Development Environment

A. Choice of software development platform

Our team develops applications for both Mac and Windows operating system environments. We have selected relevant frameworks for cross-platform development. First of all, cross-platform mobile app development refers to the method of developing software on multiple different platforms or operating systems using the same code base. In the early days of cross-platform, people were reluctant to use it because the performance did not keep up with native apps, but recent technologies are approaching the level of native apps and are on their way to being recommended. Some examples include React Native, Flutter, Xamarin, and NativeScript. These frameworks are used to develop applications that run on multiple platforms using one code base. For this, the latest versions of Windows and MacOS are needed.

Windows 11 - Windows 11 is the most recent version, offering a new look and feel to the user interface, performance improvements, and features Microsoft Teams integration and support for Android apps.

MacOS 12 (Monterey) - MacOS 12 offers unified control across Apple devices with Universal Control and enables mirroring of iOS and iPadOS screens to a Mac via AirPlay to Mac. It enhances the user experience by including Focus mode, the Shortcuts app, an improved Safari browser, increased privacy, and teamwork features.

For this project, we created a team workspace to collaborate. The UI/UX design of the application will be done utilizing Figma. For the overall project schedule, execution, process, and meeting log management, we utilized Notion and shared it with team members. To communicate, we used Slack, the collaboration tool. In addition to acting as a messenger, Slack was great for uploading and sharing files.

TABLE II
DEVELOPMENT LANGUAGE AND ENVIRONMENT

Tools and Language	Reason
Flutter	First announced by Google in 2017, Flutter was originally created to run on the Fuchsia operating system, but it became a cross-platform platform when Google decided to make it compatible with other operating systems. Because it was created by Google, it has excellent performance running on Android first, and it is truly cross-platform as it runs on iOS, Android, Windows app development for macOS, and even web browsers recently, and it is the fastest growing cross-platform. Since Flutter can target iOS, Android, web, and desktop platforms with one codebase, we felt it would simplify development and maintenance, saving us time and money. Another key feature is that Flutter is composed of reusable UI components called widgets, which are used to build every part of the application. These widgets are rich, extensible, and help us build UIs quickly, so we felt it made sense to develop with agile methods to ensure the success of our project.
Node.js	A JavaScript-based software platform that can be used to develop server-side applications. It runs on a variety of platforms including MS Windows, Linux, and macOS. Created with the vision of ”JavaScript everywhere,” the advent of Node.js made it possible to develop both web client and server applications using only the JavaScript language. Our team felt that Node.js was an ideal choice for connecting our front-end and back-end and providing APIs, as we felt that a single language and technology stack could manage the entire stack, which would be very useful in integrating with the front-end.Node.js also has the advantage of having an active developer community and a diverse open source package ecosystem, with tons of libraries and modules readily available through its package manager, npm.

TABLE III
TEAM'S WORK ENVIRONMENT

Name	Environment
Chung YouBeen	Windows 11 Home with 64-bit Operating System, and x64-based Processor, Jupyter Notebook
Park SeokWon	Windows 11 Pro, 64-bit Operating System, x64-based Processor
Lee JunYoung	MacOS Ventura 64-bit

B. Software in use

- Flutter
  Flutter is an open-source UI framework developed by Google, designed to enable rapid development and efficient execution of mobile applications. It utilizes the Dart programming language and allows for the development of apps for both iOS and Android platforms using a single codebase. When applied to our project, Flutter provides a unified development environment that ensures consistency across iOS and Android, reducing development time and costs.
- Ionic
  Ionic is an open-source platform for building mobile and web applications, often used in conjunction with Angular

for developing cross-platform apps. It is based on HTML, CSS, and JavaScript and offers a variety of plugins and tools to simplify app development. Incorporating Ionic into a collaborative project streamlines web app development and facilitates deployment to multiple platforms, enhancing developer-designer collaboration and optimizing the team's development process.

- Apache Cordova
  Apache Cordova is an open-source mobile app development framework that utilizes HTML, CSS, and JavaScript for developing mobile applications. It enables the deployment of apps to various platforms (iOS, Android, etc.) using web technologies. Cordova simplifies code sharing and collaboration among developers with different technology stacks. In a collaborative project, it enhances code sharing and streamlines the development process.
- AWS Lightsail
  AWS Lightsail is a simplified web application hosting and cloud server service provided by Amazon Web Services (AWS). It allows users to host websites and deploy applications without the need for complex configurations. When integrated into a collaborative project, Lightsail provides a hassle-free environment for deploying and scaling applications, ensuring seamless collaboration among team members.
- AWS
  Amazon Web Services (AWS) is a cloud computing and web service platform provided by Amazon, offering a wide range of cloud services to enterprises and developers. These services encompass computing, storage, databases, artificial intelligence, and various other functionalities. Incorporating AWS into collaborative projects empowers teams to efficiently manage infrastructure, share data, and strengthen collaboration among our team.
- Node.js
  Node.js is a JavaScript runtime environment used for developing server-side applications. It is based on an asynchronous event-driven architecture and is particularly useful for creating network applications using JavaScript. In a collaborative project, Node.js facilitates effective communication with databases and ensures rapid response times, enhancing overall team productivity.
- PostgreSQL
  PostgreSQL is a high-performance, secure, and scalable open-source relational database management system (RDBMS). It supports various data types and complex queries, providing reliable data storage and management. When integrated into a collaborative project, PostgreSQL ensures data integrity and enables multiple team members to access and manipulate data.
- EXPO
  EXPO is a framework and toolset that makes it easier to develop React Native apps. Developers can quickly develop and deploy iOS and Android apps using JavaScript and React Native. EXPO expedites development and debugging, ensuring project deadlines are met.
- React Native
  React Native is an open-source framework developed by Facebook that allows for the creation of mobile apps targeting iOS and Android platforms using JavaScript and React. It enables the development of apps for multiple platforms with a single codebase. In a collaborative project, React Native simplifies code sharing and maintenance, allowing for the rapid development of cross-platform apps.
- Firebase
  Firebase is a mobile and web app development platform provided by Google, offering features such as user authentication, databases, hosting, and more. Firebase enables developers to quickly develop and operate apps. In our project, Firebase enhances data sharing and real-time collaboration among our team members.
- Figma
  Figma is a collaborative design tool for creating designs and prototypes, accessible through a web-based application. It allows multiple users to collaborate on design projects simultaneously and supports real-time co-authoring. When used in a collaborative project, Figma streamlines the design collaboration process, enabling designers and developers to share design elements and prototypes and provide feedback in real time.
- Notion
  Notion is a collaboration and productivity tool for performing various tasks, including note-taking, project management, scheduling, and more. Users can create customizable workspaces and efficiently organize multiple tasks. In a collaborative project, Notion facilitates task organization, project tracking, and team communication.
- Github
  Github is a code hosting platform based on Git, used for software development and collaboration. It supports code version control, collaboration, issue tracking, and source code hosting and is widely used in the developer community. In a collaborative project, Github simplifies code management, fosters collaboration among team members, and ensures version control and issue tracking.
- Overleaf
  Overleaf is an online LaTeX editor and collaborative environment that provides tools for writing papers and documents. It is used for creating scientific and technical documents using LaTeX and supports real-time.

C. *Cost Estimation*

We'll need to recreate the space for the built-in fridge near the front door (which is still temporary), so contractually secure that space when people build the building. It'll need to create a fridge slot or built-in space. There will be construction and installation costs for this, and may need professional help. There will also be parts costs to maintain the fridge, and there will be costs when the mini fridge breaks down.

From a software perspective, there may be additional costs for smart home connectivity. You need to consider the cost of setting up software and configuring apps for this connectivity, and the server infrastructure for a smart refrigerator requires web servers, databases, APIs, authentication and security, real-time communication, scalability, and back-end management and maintenance capabilities.

Since we need to get information from the server in real time, we will use AWS, a commercial cloud platform, as an example of a backend server and database that can be integrated for our prototype service. Among them, AWS Lightsail is a simplified platform for use cases such as hosting simple web applications and websites. It simplifies provisioning and management and allows users to quickly launch applications without complex setup. AWS Lightsail uses a simple monthly pricing model, and users can more easily predict their usage and charges. There is also an AWS Free Tier aimed at students and developers, which allows them to use certain AWS services for free within a limited scope.

D. *Task distribution*

we will provide this later at the next phase - design but we've decided it like this (temporarily). It may change later

TABLE IV
TASK DISTRIBUTION

Roles	Name	Description
Backend Developer	Lee JunYeong	A Backend Developer is responsible for the server-side or backend portion of web and mobile applications. They interact with databases, manage the application's logic, and support the actions users perform through the app. Backend Developers handle server and data processing efficiently and maintain security.
Frontend Developer	Park SeokWon	A Frontend Developer is responsible for designing and developing the user interface (UI) of web and mobile applications. They use web technologies like HTML, CSS, and JavaScript to build the visual components of web pages and apps that users interact with, enhancing the user experience.
UI/UX Product Designer	Chung YouBeen	A UI/UX Product Designer is responsible for user experience (UX) and user interface (UI) design. They optimize the design of applications or websites to make them intuitive for users. UX designers understand user requirements, while UI designers implement them visually.

IV. SPECIFICATIONS

Magicool provides users with a variety of features to make early morning delivery convenient and fresh while using their LG refrigerators. The magicool is a mini fridge that can be placed near the front door for early morning delivery users. As the number of consumers using early morning delivery has exploded, the amount of waste generated has become a social problem. Therefore, we want to present a mini portable refrigerator that provides refrigeration by using our service, and appeal to the fact that it can be stored anywhere at the moment when refrigeration is needed. When users stored in the database access the application, they are provided with the following key features.
(The app screen hasn't been configured yet, so we couldn't attach it. Therefore, Our team put the overall structure in the form of a graphical diagram. And we will explain the main functions and update it with images as soon as the screen configuration is completed.)

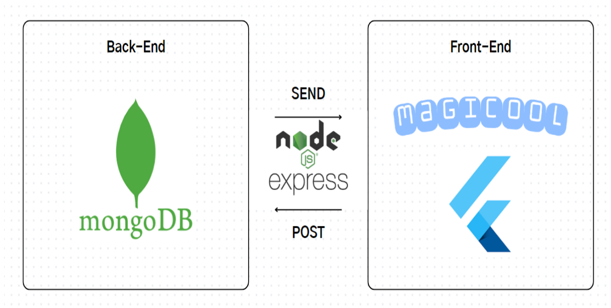
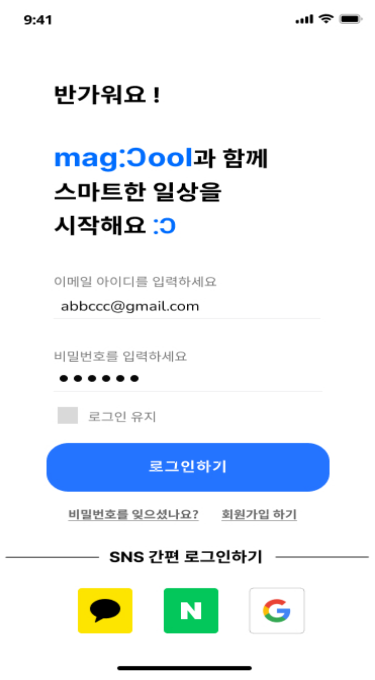


Fig. 4. Application Framework

A. *login and sign-up*

This page requires users to sign up to create an account and then sign in to use this service. The login page is used for user authentication. Because the application handles the user's shipping order information, estimated time of arrival, etc. user's information, all user's information associated with the shipping service must be stored based on a unique email user account.


9:41

반가워요 !

mag:Oool과 함께
스마트한 일상을
시작해요 :Ↄ

이메일 아이디를 입력하세요
abbccc@gmail.com

비밀번호를 입력하세요
••••••

로그인 유지

로그인하기

비밀번호를 잊으셨나요? 회원가입 하기

SNS 간편 로그인하기

Fig. 5. Login Page

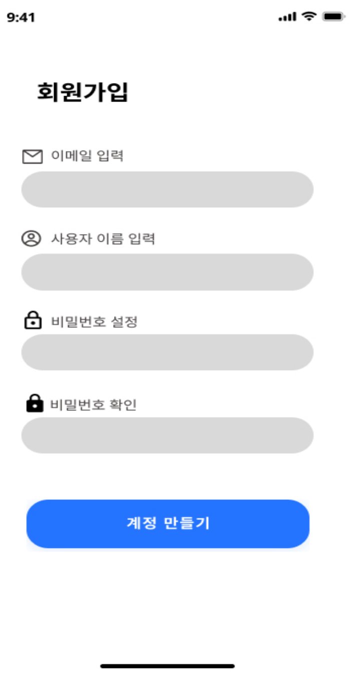


Fig. 6. Signup Page

B. Main Page

The main page will provide a dashboard of frequently requested items at a glance, and we'll also be adding a quick launch feature for the NUGU AI speaker.

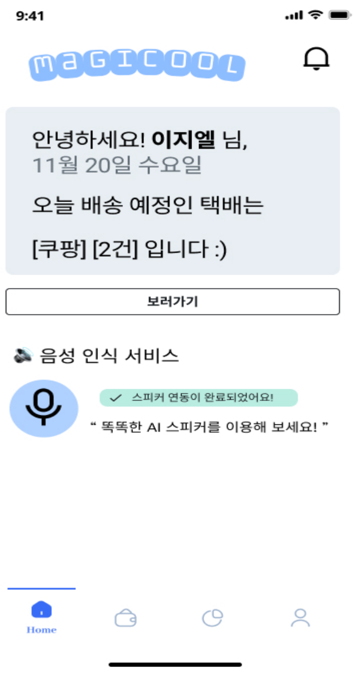


Fig. 7. Main tab

C. Healthy Dishes Generator

Utilizing ChatGPT, a generative AI, we plan to add an option to load the user's order history and select healthy ingredients that can be substituted with healthy ingredients to create a "Healthy Dishes Generator" feature.

ex) If you ordered sugar $\rightarrow$ How about using stevia instead next time? Or Butter $\rightarrow$ vegetable oil This is a feature that allows user to 'input' purchases list and either 1) create a healthy meal recipe or 2) learn from user's past orders and create healthier alternatives.

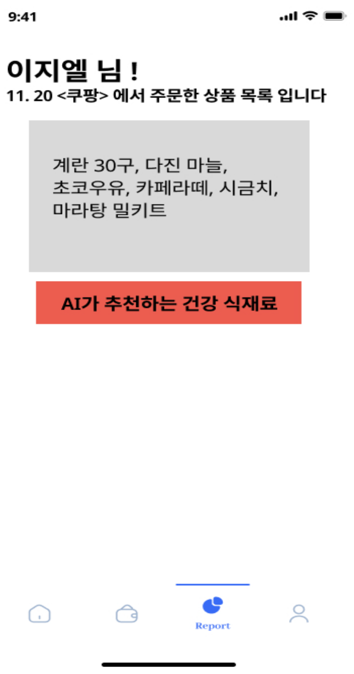

9:41
이지엘 님 !
11. 20 <쿠팡> 에서 주문한 상품 목록 입니다
계란 30구, 다진 마늘,
초코우유, 카페라떼, 시금치,
마라탕 밀키트
AI가 추천하는 건강 식재료
Report

Fig. 8. Report Tab

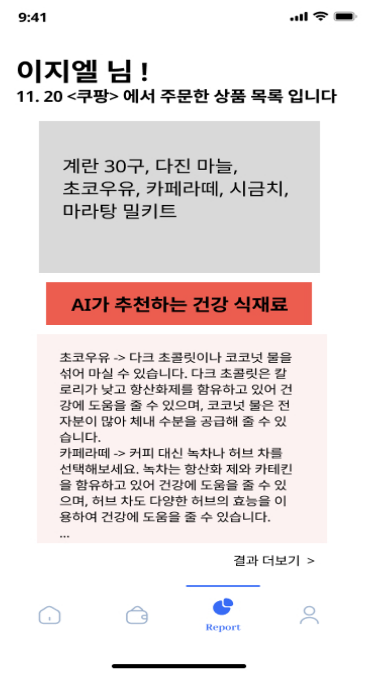
D. Mini Fridge (Freshkeeper) to Main Fridge (LG model) Integration

We can utilize mobile APIs and sensors to check what's inside the mini fridge.Real time check: Check the items inside using the weight sensor. Notification: If the weight sensor is 'true' for a certain period of time send a notification (like checking the mailbox) Sleep mode: Customize the mini fridge to run when no deliveries are scheduled, etc.

Wake-up notification: When the user wakes up in the morning and opens the main fridge, know that the user is awake and notify the mini fridge that an item has been delivered (send a push message) or through the speaker.

E. Coupang OPEN API Integration

To connect to Coupang OPEN API, we will develop according to the guide, such as searching for products and checking product registration status on the Coupang Developers site. This feature allows that user can see the status of an item, whether it's being prepared, refunded, or placed on order. We're also working on a technology that, when applied to


9:41

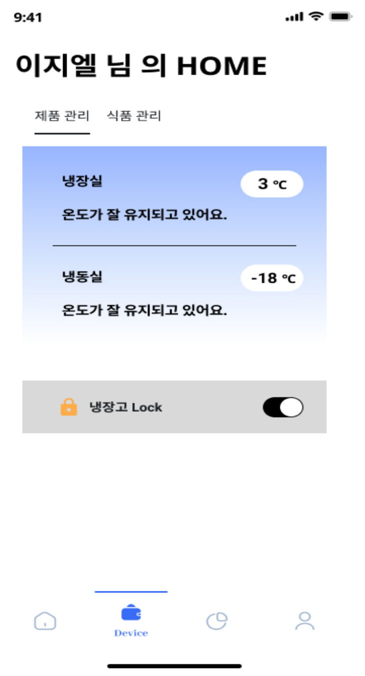
이지엘 님 !

11. 20 <쿠팡> 에서 주문한 상품 목록 입니다

계란 30구, 다진 마늘,
초코우유, 카페라떼, 시금치,
마라탕 밀키트

초코우유 -> 다크 초콜릿이나 코코넛 물을 섞어 마실 수 있습니다. 다크 초콜릿은 칼로리가 낮고 항산화제를 함유하고 있어 건강에 도움을 줄 수 있으며, 코코넛 물은 전자분이 많아 체내 수분을 공급해 줄 수 있습니다.
카페라떼 -> 커피 대신 녹차나 허브 차를 선택해보세요. 녹차는 항산화 제와 카테킨을 함유하고 있어 건강에 도움을 줄 수 있으며, 허브 차도 다양한 허브의 효능을 이용하여 건강에 도움을 줄 수 있습니다.
...

Fig. 9. Report Tab : Healthy alternatives button recommendations by Chat GPT

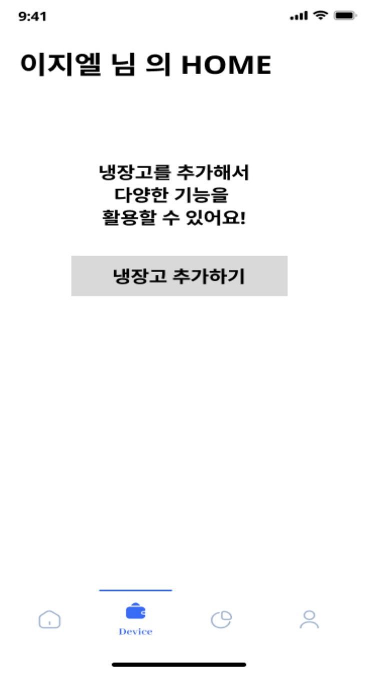

9:41

이지엘 님 의 HOME

냉장실 3 ℃
온도가 잘 유지되고 있어요.

냉동실 -18 ℃
온도가 잘 유지되고 있어요.

Fig. 11. Devices tab-Fridge thermostat and lock features


9:41

이지엘 님 의 HOME

냉장고를 추가해서
다양한 기능을
활용할 수 있어요!

Fig. 10. Devices tab-Adding products

’sleep mode’, will check the estimated arrival time and turn on the refrigerator in time to avoid disturbing you in the wee hours of the morning. If the product is delivered but the weight sensor is false, we are also thinking about checking the image to prevent non-delivery errors.

F. *Utilizing NUGU AI Speaker*

Utilizing the NUGU SDK provided by SK Telecom’s NUGU developers, we will create AI services by connecting the NUGU platform to internet-connected devices and our applications. In our FreshKeeper mobile app, users can check the estimated arrival time based on voice recognition, receive an alert if the door is not closed, and remotely open the door of the mini fridge.

G. *My Page / Settings*

On this page, user can set privacy settings or link LG Refrigerator device here.

V. ARCHITECTURE DESIGN

A. *Overall Architecture*

Our Service is comprised of three big components: Front-end module, Back-end module, and Database module.

- The first module is the front-end module that help users interact with our service. Users interact with out survice via templates and functions of out front-end module. Interacting with our service includes activities such as accessing our back-end server (by HTTP, which refers to Hyper Text Texting Protocol)/database and using NUGU AI speaker/accessing NUGU play server. Our front-end module was developed using Flutter framework(dart).
- The second module is the back-end module. Back-end module help end-users manipulate the database and response to the request sent from front-end module. This back-end module is comprised of several node.js packages downloaded using npm (node package manager), such as ‘express’ package or ‘openai’ package

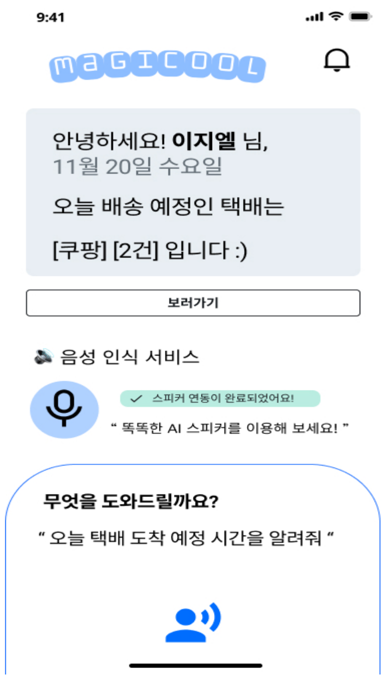


Fig. 12. Activate SKT NUGU AI Speaker

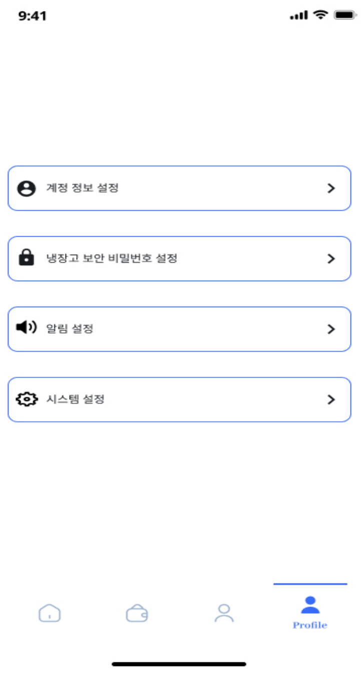


Fig. 13. MY page Tab

for using OpenAI api(Chatgpt). The main role of our back-end module is to send a response to our front-end side after receiving a requests (generated by user activities) from out front-end module. This back-end server is directly connected to our mongodb database so user can manipulate this mongodb database through this back-end module. Out back-end server is running

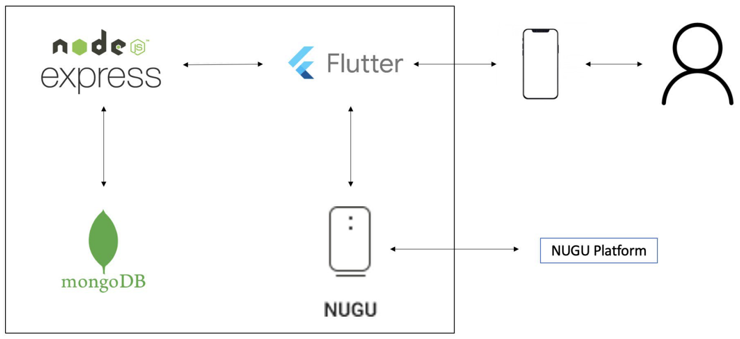


Fig. 14. MY page Tab

on our team's personal laptop and so do our mongodb database. Our back-end module was developed using Express framework (Node.js).

B. Directory Organization

Source Code Location: https://github.com/magiCoolSE/

- /magiCool_Frontend:
  android, assets, ios, lib, log, test:
  first base frontend directory with flutter and dart to implement application in android and ios.
- /magiCool_Frontend/android/app/src/main/kotlin/com/magicool/app/res/AndroidManifest.xml
  The AndroidManifest.xml file is an XML file that is generated by default when you create Android studio projects. The AndroidManifest.xml file can describe the functionality and requirements of applications on Android platforms.
- /magiCool_Frontend/android/app/src/main/kotlin/com/magicool/app/MainActivity.kt
  MainActivity.kt is an activity class that is executed when a user clicks on an app. The activity class is mainly used for screen composition. MainActivity inherits AppCompatActivity, which means it inherits the members of the upper class, such as variables and functions.
- /magiCool_Frontend/lib/main.dart
  In a Flutter project, the main.dart file is the entry point for the application. It contains the main() function, which is responsible for starting the application. The main() function runs the code that builds the UI and initializes other components.
- /magiCool_Frontend/lib/core/app_export.dart
  app_export.dart is one of the libraries in the API document of the Dart programming language, which contains frequently used file imports.
- /magiCool_Frontend /lib/presentation
  all of screens designed through figma are located in this directory and loaded by app_routes.

C. Detailed description of Back-end source codes.

- /magiCool_Backend/app.js:

Directory	FIle Names	Modules
/magiCool_Frontend	android assets ios lib test	
/magiCool_Frontend/ android/app/src/main	kotlin/com/magicool/app res AndroidManifest.xml	
/magiCool_Frontend/ android/app/src/main/ kotlin/com/magicool/app	MainActivity.kt	
/magiCool_Frontend/assets	fonts images	
/magiCool_Frontend/lib	core presentation routes theme widgets main.dart	
/magiCool_Frontend/lib/ core	constants utils app_export.dart	
/magiCool_Frontend/lib/ core/utils	date_time_utils.dart image_constant.dart size_utils.dart	
/magiCool_Frontend /lib/presentation	app_navigation_screen container_screen eight-draweritem eleven_screen five_draweritem four_draweritem info_screen iphone_13_pro_max _screen login_one_screen login_screen nine_draweritem sign_up_screen ui_login_screen ui_title_screen x_screen two_screen	
/magiCool_Frontend /lib/routes	app_routes.dart	
/magiCool_Frontend /lib/theme	app_decoration.dart custom_button_style.dart custom_text_style.dart theme_helper.dart	

This file is the main source code for running node.js server using express framework. All of routing and error handling opions are included here.

- /magiCool_Backend/package.json:
  All names of packages being used in this project are written here. Some other configurations can be managed here too such as setting a start command for running a server or changing name of the server.
- /magiCool_Backend/bin/www.js:
  Configurations for starting the server are written here. Configurations such as getting the port and storing in Express or creating http server, error handling, starting the server are included.

Directory	FIle Names	Modules
/magiCool_Backend	bin models node_modules public/stylesheets routes views app.js package-lock.json package.json	
/magiCool_Backend/bin	www	
/magiCool_Backend/ models/	User.js	
/magiCool_Backend/routes	api index.js users.js	
/magiCool_Backend/routes /api	gpt.js register.js	
/magiCool_Backend/views	error.pug index.pug layout.pug	
magiCool_Backend/public/ stylesheets	style.css	

- /magiCool_Backend/models/Users.js:
  This file handles the database schema of Mongodb, and this Users.js shows database schema for user sign up/sign in, the table which is comprised of users' names, email addresses, and passwords.
- /magiCool_Backend/routes/index.js:
  This file is used when routing '/'. When client gets to '/', back-end server response with the text 'Welcome to Express'. Also, this file helps automation of database connection by connecting Mongodb when user gets to '/'.
- /magiCool_Backend/routes/api/gpt.js:
  This file is used when routing '/api/gpt'. When this script is called, it receives a name of unhealthy foods that users put through UI and sends that string to chatgpt, asking chatgpt to recommend better foods in comes of personal health. Asking such request to chatgpt is made by using chatgpt api, with secretly generated chatgpt api key of the server runner. Chatgpt sends a response text of the request to back-end server through chatgpt api and back-end server sends that string to front-end client.
- /magiCool_Backend/routes/api/register.js:
  This file is used when routing '/api/register'. By receiving POST request from the front-end, with json format 'name':'value', 'email':'value', 'password':'value', back-end server adds this data into database connected to itself(Mongodb). After comparing the data to the current database, server rejects the request if the value of 'email' sent is not unique.

VI. Use Case

A. Entry

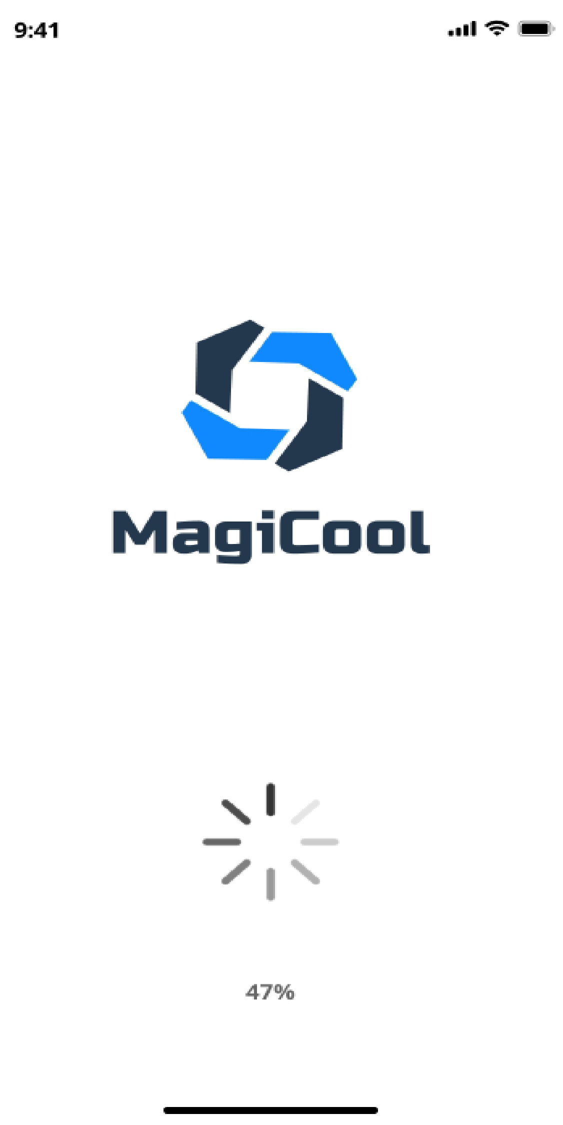


Fig. 15. splash screen

Name	Description
splash screen	splash screen is the first screen that appears when an app is opened on a mobile device. It's also known as a launch screen, startup screen, boot screen, boot skin, or welcome screen.

B. log-in

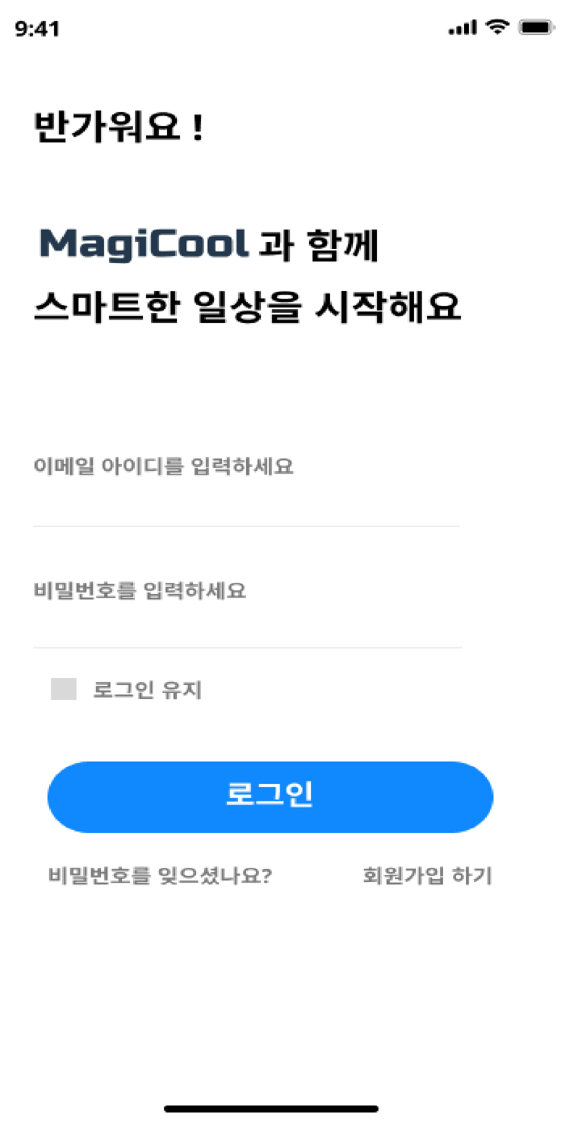
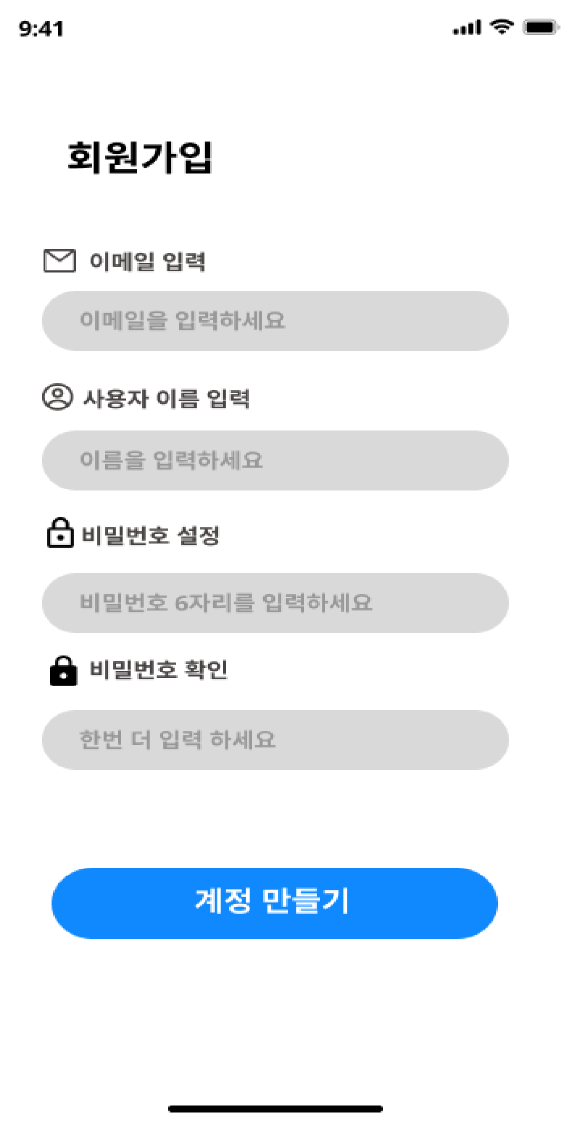


Fig. 16. log-in page

Name	Description
log-in page	A login page is a page that allows a user to access a website or application by entering their credentials. These credentials can include a username, email, or password. Some login pages also allow users to authenticate with a social network login.

C. sign up


9:41

회원가입

이메일 입력

이메일을 입력하세요

사용자 이름 입력

이름을 입력하세요

비밀번호 설정

비밀번호 6자리를 입력하세요

비밀번호 확인

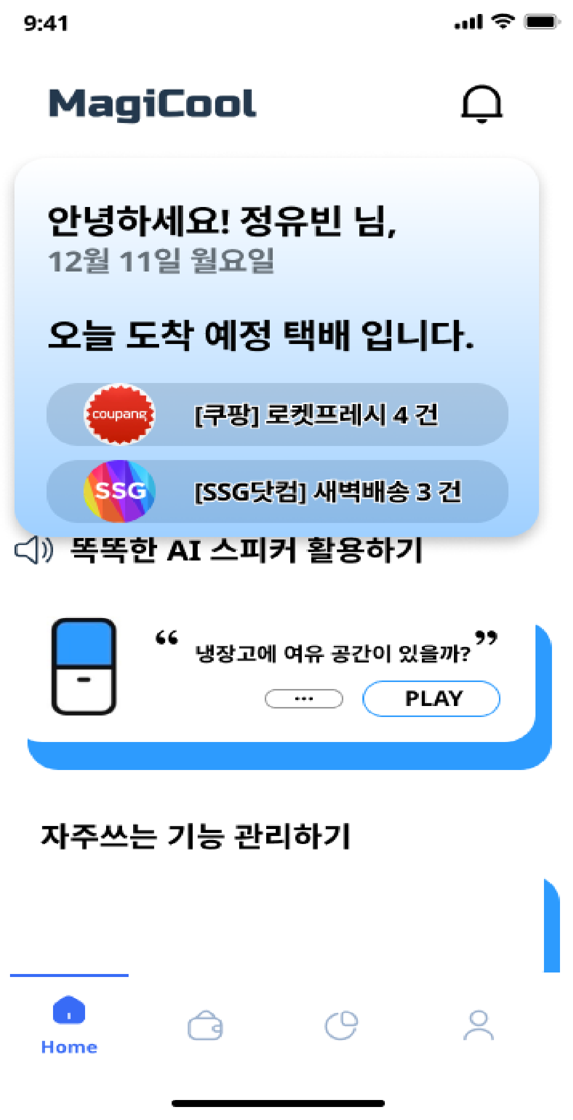
한번 더 입력 하세요

계정 만들기

Fig. 17. log-in page

Name	Description
signup page	A signup page, also known as a registration page, is a web page that allows users to register and access a system. A signup form is a web page, popup, or modal where users enter information to access a website's services.

D. home


9:41

MagiCool

안녕하세요! 정유빈 님,

12월 11일 월요일

오늘 도착 예정 택배 입니다.

[쿠팡] 로켓프레시 4 건

[SSG닷컴] 새벽배송 3 건

똑똑한 AI 스피커 활용하기

“ 냉장고에 여유 공간이 있을까? ”

PLAY

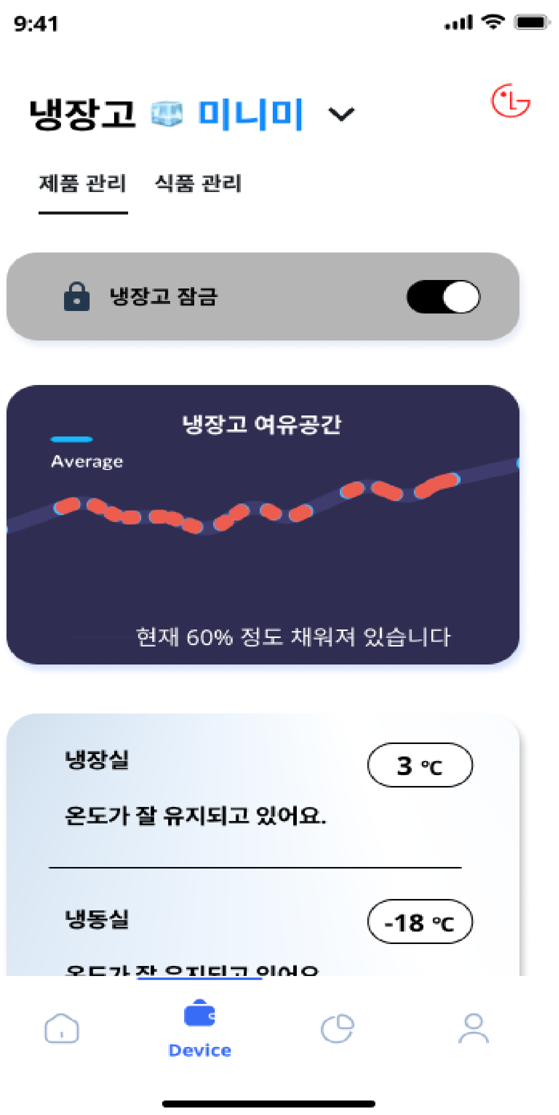
자주쓰는 기능 관리하기

Home

Fig. 18. log-in page

Name	Description
home page	A homepage is the first page of a website and corresponds to the front door or gate of the website. It is a URL or a local file that automatically pops up when your web browser starts up.

E. device


9:41

냉장고 미니미

제품 관리 식품 관리

냉장고 잠금

냉장고 여유공간

Average

현재 60% 정도 채워져 있습니다

냉장실 3 °C

온도가 잘 유지되고 있어요.

냉동실 -18 °C

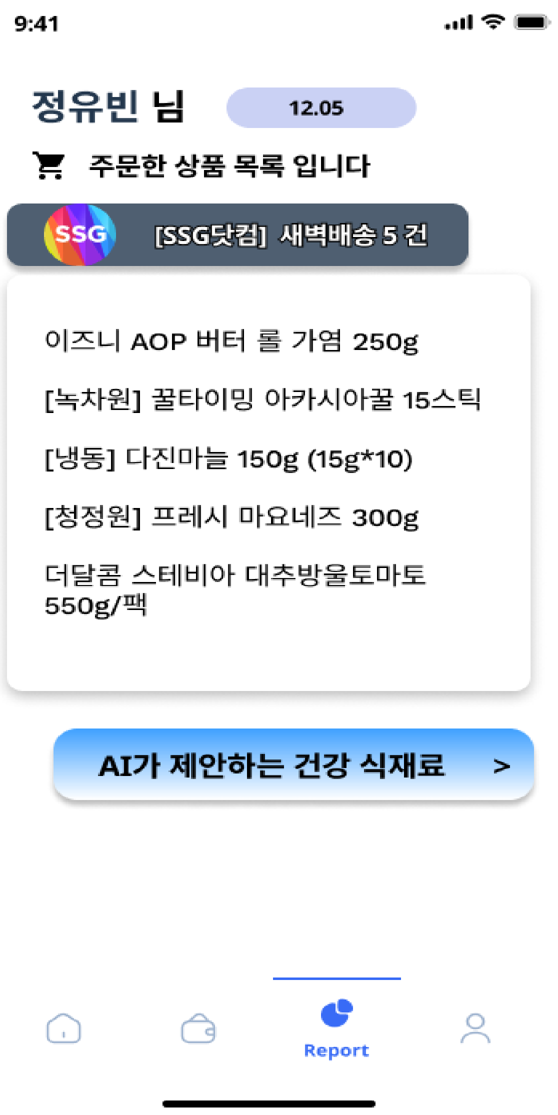
온도가 잘 유지되고 있어요

Device

Fig. 19. log-in page

Name	Description
device page	A device page is a central location for all troubleshooting and repair information for a specific device.

F. report


9:41

정유빈 님 12.05

주문한 상품 목록 입니다

[SSG닷컴] 새벽배송 5 건

이즈니 AOP 버터 롤 가염 250g

[녹차원] 꿀타이밍 아카시아꿀 15스틱

[냉동] 다진마늘 150g (15g*10)

[청정원] 프레시 마요네즈 300g

더달콤 스테비아 대추방울토마토 550g/팩

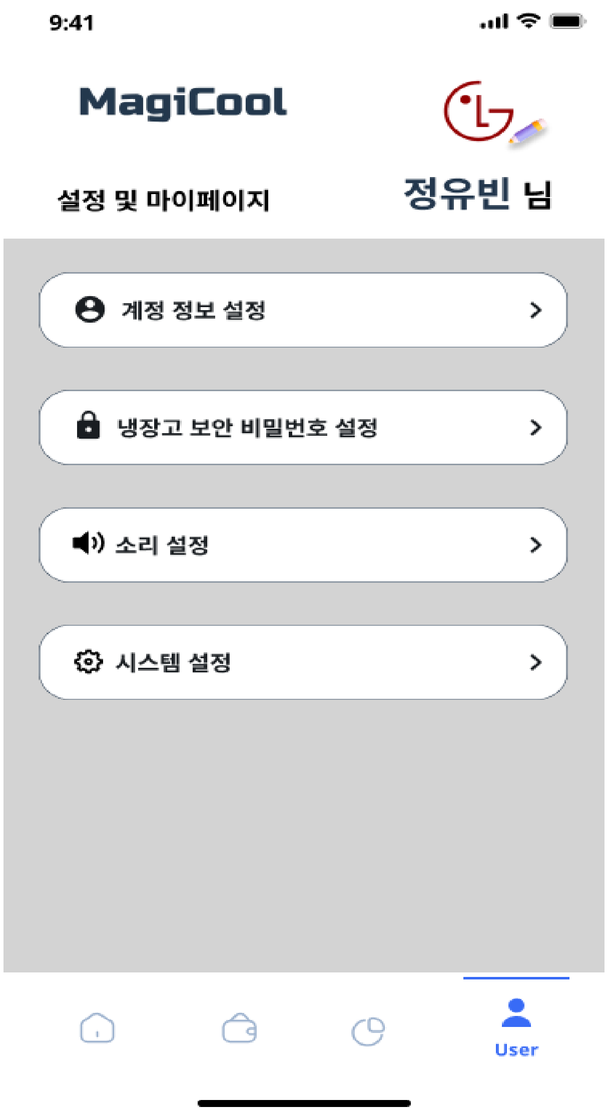
AI가 제안하는 건강 식재료 >

Report

Fig. 20. log-in page

Name	Description
report page	A report page can display the results of reports in their own pages. Reports are divided into report types.

G. *mypage*


9:41

MagiCool

설정 및 마이페이지

정유빈 님

계정 정보 설정

냉장고 보안 비밀번호 설정

소리 설정

시스템 설정

Fig. 21. log-in page

Name	Description
mypage	A cloud-hosted SSO solution that allows users to quickly access their applications. It also reduces routine requests for IT admins through self-service device registration and management.

VII. Discussion

We would like to discuss the difficulties our team had when developing our service first. First, starting the project was the most difficulty our team had. Adding a new team member and redistributing task and roles for developing our service was not easy. Also, our team spent a lot of time choosing and revising a topic of our service and this surely made our team having less time planning/developing and testing our mobile application. Furthermore, since all of our team members knew nothing about software development, studying the knowledge needed when developing the software consumed a lot of time. Lastly, The number of members of our team is only three, which is less than other teams (four on average), so there were more tasks distributed per team members, compared to other teams in the class. These difficulties prevented our team from making a successful output (mobile application service), and our team is surely ashamed by the result our team made. However, there are lessons our team has learned in terms of team project during this project, such as the importance of clear task distribution between team members or getting a sense of group software development. It has been hard and confusing time progressing this team project but it is meaningful that our team has experienced a team project related to service development.