FIESTA - “MC of the party”

*- make mood & play mini-games -

1st Kang JunGyu
Dept. of Information Systems
Hanyang Univ.
Seoul, Republic of Korea
jgpk226@gmail.com

2nd Kim Jihoon
Dept. of Information Systems
Hanyang Univ.
Seoul, Republic of Korea
jhkimlego@gmail.com

3rd Park Jongsu
Dept. of Information Systems
Hanyang Univ.
Seoul, Republic of Korea
qwt629@gmail.com

***Abstract*—Our team is focusing on providing an application service for making up the mood of parties that are hosted at homes and shared spaces. We plan developing lighting & appliance control, multiplayer games and space management functions, available to all participants for an entertaining party. We came up with the idea “Fiesta” as we thought of a solution that could release the burden of party hosts and make the party more memorable. Party hosts have many factors to consider such as the lighting of the space, temperature, contents of the party, music and many others. To reduce these tasks focused on the host, our team developed three main functions.**

First, “Fiesta” has the function to register and control various appliances at any place. For example, lighting devices can be adjusted in color and brightness; other appliances can be adjusted accordingly, such as temperature control of air conditioners. Moreover, we use generative AI to suggest and change the color and brightness of lights according to user inputs.

Another function is providing multiplayer mini-games that can be enjoyed with mobile devices of participants. Playing the game involves one game host and other participants. The host can see the questions along with answers of the games, answering orders of participants and moving on to the next games. Participants can view questions of the games and are also provided with an answering button that checks order based on the speed of pressing the button.

Finally, space owners can control their spaces by using a unique code generated by the application. These codes can be distributed to other participants which allows them to gain access to all the appliances and games. Importantly, the code can be refreshed which prohibits access from outdated codes.

In conclusion, our team hopes to provide appliance control along with a game function that will reduce the tasks required by party hosts, resulting in a more seamless party and help all participants leave with a memorable party experience.

***Index Terms*—MC, Manager, Lighting, Appliance, Game**

I. Role Assignment

Name	Roles	Task description and etc.
Kang JunGyu	Development Manager, Software developer (Back-end)	Development Manager is responsible for managing the overall project. He oversees the functions that are required, the UI/UX configuration, the flawless operation of the prototype, and whether the project proceeds as planned. In addition, he constantly monitors the team member’s progress and whether they are communicating effectively.
		Software developer (Back-end) is in charge of creating databases, web services, and APIs used by front-end developers. For this position, an understanding of the core database and its features is essential. Every logic as well as integrating with the front-end is developed by him. He uses two tracks of servers, which are the main server and the AI server. The main server interacts with the database, manages user authentication, and handles most server-side logic by Java Spring. The AI Server uses AI codes, open AI, and other libraries for games by Python Flask. He also writes HTTP request codes and WebSocket connection codes on Front-end using React-Native for this project.
Kim Jihoon	Product Designer, Software developer (AI)	A Product Designer must focus on the usability of a product or service and modify the design of a product to obtain better results. He is responsible for communicating smoothly with other team members and create collaboration. A software developer (AI) is responsible for any requirements of the project in terms of data and artificial intelligence. This role selects and refine data into adequate forms in order to make the artificial intelligence model. This role uses machine learning for implementing artificial intelligence to requirements of the software, must run and test model and make a statistical analysis.
Park Jongsu	Software developer (Front-end), UI/UX Designer	Software Developer(Front-end) is responsible for the creation and implementation of the user interface of web applications. They use technologies like HTML, CSS, and JavaScript to build visually appealing and responsive websites. Their duties involve writing and optimizing code, ensuring compatibility across various devices and browsers, and collaborating with back-end developers. They play a crucial role in optimizing website performance, conducting testing and debugging, and staying updated with the latest front-end development trends and practices. UI/UX Designers focus on the visual and interactive aspects of software applications, designing the user interface and ensuring aesthetics and functionality. They conduct user research to create wireframes and prototypes, collaborate with development teams for accurate implementation, and conduct

		usability testing. Their role is to create a user-centric, visually appealing, and accessible experience.

II. Introduction

A. Motivation and Term Definition

In 'MT', Many college students go these days, the 'MC' seems to have a hard time controlling the participants to host the party and run party games. If MC was someone other than a human, everyone could join the party instead of being divided into MC and participants. Also, a lot of game ideas beyond human limits could be generated. This human replacement MC idea could be applied to similar events and meetings like house parties, trending in some nations or periods like disease epidemic.

For a fun MT and eventful house party, not only party games but also Setting the party mood is one of the most important things. To do so, the home appliances in the room could conduct some special roles : the party mode. Setting the party mode with smart-home appliances like LG and control them in the app. Control lightning of all the smart-home appliances to various colors suitable for the party concepts. Control the atmosphere, temperature, and homebrew in the best condition.

- MT : Membership Training in Korea - A party lasts roughly two days and is considered a free-form training session, where students spend time socializing with peers in the same academic majors or clubs, often at a remote location.
- MC : Master of Ceremonies - the official host of the ceremonies, parties, events, or similar performances.

B. Problem Statements

There are a number of issues that arise when hosting parties and games. Running out of game ideas, not enough people to run the games, the wrong atmosphere for the party, etc. We will provide a service that will make it simpler for party hosts to manage their appliances during the party.

1) Why House Parties are Trending: Home parties have gained popularity in South Korea, especially among the younger generation, for several reasons

- Influence of Social Media: The convenience of social media platforms makes it easy to document and share special moments. House parties offer a great opportunity to capture and share experiences, driving more people to host them.
- Cost-Effective: House parties are a more budget-friendly option compared to expensive dining or clubbing. Due to recent inflation, the percentage of people dining out has decreased significantly. Therefore, sharing simple food and drinks with friends is economical.
- Comfort and Intimacy: House parties provide a comfortable and intimate setting to spend time with friends and family, fostering meaningful interactions and conversations.
- Variety of Activities: House parties offer a range of activities beyond food and drinks, including board games, karaoke, movie nights, dancing, and quizzes.
- Cultural Influence: There is a growing interest in Western culture in South Korea, and the concept of house parties from the West may have contributed to their popularity.

2) Role of the Party Host: As a party host, you need to be a mood creator, and you want to keep things hyped preventing participants from getting bored. Most of all, you want to make sure your guests have a good time.

There are a number of issues that arise when hosting parties and games. Running out of game ideas, not enough people to run the games, the wrong atmosphere for the party and others. The main role of a house party host is to help guests have a good time and keep the party running smoothly, but they can't fully enjoy the party with other things to worry about.

C. Solution

We will provide a service that will make it simpler for party hosts to manage their appliances during the party. It allows for customization rather than a one-size-fits-all party atmosphere. These days, there are appliances for home parties, but we wanted a way to integrate with smart home systems and devices to make it easy to control multiple devices from one app.

We want to provide a game that requires no preparation and is accessible to everyone. We don't want to differentiate between MCs and game players, but rather provide a time for everyone to have fun during the party.

D. Research on Any Relative Software

III. Requirement Analysis

A. Logging In

Logging in is required for users to have access to functions of the application. Without a login, it is impossible to use any of the functions of the app. If the application is not logged in, users can create their own account or login using an account they have created before.

B. Creating Account

Users can sign up by setting a user id and password of their preference. Signing in requires input of the given information in the sign up process such as userid, password and name.

C. Space Management

"Space" refers to any type of place where parties can be held and added to the software. Space management mainly focuses on space creation and space connection. Space creation means making a new space so that it can be used to link with electronics and made available for others to join. Space connection means gaining access to spaces via the code provided by owners. Each space has a unique code which can be shared with others and is required to access the functions provided. The code can be generated by creating a space in the application and request code generation on the settings tab. Also, Space management is separated into 2

types of classes;owners and users.

Space owner: Can generate random authentication codes for individual spaces. Also, it is possible to refresh the code which terminates the validity of the code for previous users, forcing them to lose access of the space.

Space user: A space user can gain access to spaces with the code provided by the owner. By having the code, the user also has access to the lighting and appliances pre-registered by the Space owner.

D. Status Dashboard

Users can view the status of the space upon logging into the software. It is the first page that a user will encounter upon connecting to a space. The dashboard acts as the main page of the application. It includes the name of the space, a simple announcement bar, data on space temperature and air quality. The name of the space is provided by owners, space temperature and air quality information comes from data on the related appliance. Also, it contains a music player where all users with access can add desired songs to a shared playlist. It is connected to whatever music output device that is present at the space, which means controlling the music player will have an effect on the sound of the space for everyone.

E. Lighting Control

Provides controlling any hardware with lighting function. Users can change the brightness, color and effects of the lights using the software. It is not just for lighting devices but also for any kind of appliance that has a lighting function such as a NUGU candle or LG Mood-Up refrigerator Also, users can input any type of situation desired in natural language and the lighting will automatically change accordingly using AI. The input can be made by a input box using a keyboard or by requesting it in speech to an AI speaker. To help users that cannot make a decision for which lighting to use, there will be some preset requests available to execute.

F. Appliance Control

Provides control of any appliances that are synced with the software, such as changing the space temperature using an air conditioner, checking remaining beer amount on Homebrew, setting the power of a air purifier. It will be functioning as a shared setting, which means any user can adjust the appliance. This will help make the environment of the party adequete for party instantly. Also, space owners can lock certain appliances, which results in prohibiting users from access to the locked appliance. Locking option can be activated by toggling a lock icon inside the appliance settings

G. Party Games

Users can play games from a list of games provided. There are 2 types of users in the process of the game, operator and participant.

Game operator: Page consist of the answer of the game, a list to check the answering order, buttons for moving on to the next game and quitting the game for everyone

Game participant: Page consist of the game question and a push button for confirming the answering order, which is in first-come-first-served basis.

Currently there will be 3 games that will be serviced, however it is possible to include various games since the format is shared. The list of serviced games are as following.

- Image game: An AI model merges the face of two celebrities, the participants are asked to guess the celebrities based on the merged image.
- Landmark guessing game: The game receives the coordinates of famous landmarks around the world from OpenAI API. The data is sent to Google map API for a street view photo of the coordinate. Participants are required to guess the landmark from the street view photo.
- Gather up: Game operator first inputs the number of participants. NUGU speaker plays any music for about 10 seconds, stop the music and tell how many people to gather up. Then, participants gather up according to the given number and participants who could not gather up are eliminated. The process will be entirely conducted by the NUGU speaker, and the given number will be calculated by a pre-set algorithm

IV. Development environment

A. Choice of Software

TABLE I
DEVELOPMENT LANGUAGE AND ENVIRONMENT

Tools and Language	Description and Reason for Selection
Spring	Spring is a comprehensive and widely adopted framework for building Java-based enterprise applications. It provides a range of tools and libraries that simplify the development of large-scale, robust, and highly maintainable applications. The Spring framework addresses various aspects of application development, aspect-oriented programming, and transaction management. Spring facilitates the development of enterprise-class applications and microservices by promoting best practices and modularity. It's an essential tool for Java developers looking to streamline application development and improve code quality. The reasons for choosing Spring framework over others is first, has a characteristic that the developer does not depend on particular libraries or container stacks. Also, in the process of database synchronization(when working on a scale of status change), it is fairly simple to annotate transactions. Most of all, it is very popular that has the biggest community and a very stable framework.
React Native	React Native is an open-source UI software framework created by Meta Platform, Inc. It is used to develop applications for Android, Android TV, iOS, macOS, tvOS, Web, Windows and UWP by enabling developers to use the React

	framework along with native platform capabilities. It is also being used to develop virtual reality applications at Oculus. One of the biggest feature of React Native is being able to create native UI for Android and iOS using JavaScript, and create high-quality UI faster than using HTML. React Native communicates with Native Thread over native bridges, optimizing performance unlike web apps. Using a method of communicating with the native without using this web is called a Hybrid App, and there are Xamarin, Native Script, and flutter. In this project, we chose React Native for developing mobile applications in the frontend development environment because it offers a wealth of resources, and it supports native development environments for both Android and iOS, making it a superior choice over other mobile app programming frameworks.
Python	Python is a high-level, versatile, and dynamically-typed programming language known for its simplicity and readability. It has a vast standard library and a thriving ecosystem of third-party packages and frameworks, which makes it suitable for a wide range of applications, including web development, data analysis, machine learning. Python's syntax is clear and expressive, making it an ideal language for both beginners and experienced developers. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python has gained significant popularity in recent years for data science and machine learning thanks to libraries like NumPy, pandas, scikit-learn, and TensorFlow. It is also used in web development with frameworks like Flask. Python's simplicity, versatility, and large community make it one of the most popular programming languages in the world.
PostgreSQL	PostgreSQL, often referred to as "Postgres," is a powerful open-source relational database management system (RDBMS). It's known for its advanced features and extensibility, making it a popular choice for both small and large-scale applications. PostgreSQL offers robust support for structured data, ACID compliance, and features like indexing, full-text search, and geospatial capabilities. It is highly extensible and allows developers to create custom functions and data types. With a strong focus on data integrity and security, PostgreSQL is widely used in a variety of industries, including web applications, data warehousing, and geographic information systems (GIS).

TABLE II
DEVELOPMENT LANGUAGE AND ENVIRONMENT

Name	Develop Environment
Kang Jungyu	macOS Sonoma 14.0 Spring, Flask
Kim Jihoon	macOS Sonoma 14.0 Pytorch, TeXLive
Park Jongsu	macOS Sonoma 14.0 React-Native, Figma

B. Software in Use

1) GitHub: GitHub is an indispensable platform in the world of software development. It serves as a web-based hub that plays a pivotal role in version control and team collaboration, allowing developers to efficiently manage their source code. It is particularly known for its Git-based repository hosting, which enables developers to store, manage, and monitor changes in their codebase. GitHub provides a range of features designed to streamline the development process, including code repositories for version control, pull requests for peer review, and issue tracking for tracking and managing project tasks and bugs. Developers can work on different branches of a project simultaneously and merge their changes seamlessly. This fosters collaboration among team members and ensures the integrity of the codebase. GitHub has become an essential tool for distributed software development, enabling remote teams to work together seamlessly.

Fig. 1. Logo of GitHub

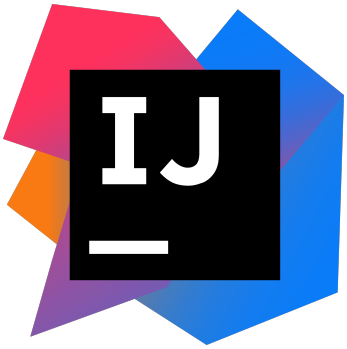

Fig. 2. Logo of IntelliJ IDEA

2) IntelliJ IDEA: IntelliJ IDEA is a powerful integrated development environment (IDE) that is specifically designed for Java developers. It offers a plethora of tools and features to streamline Java software development. The IDE provides intelligent code completion, allowing developers to write code faster and with fewer errors. It also offers extensive code analysis, which can detect potential issues and suggest improvements in real-time. The IDE supports popular Java frameworks, libraries, and build tools, making it a valuable tool for Java developers. Its user-friendly interface and extensive plugin support further enhance the development experience.

Fig. 3. Logo of Visual Studio Code

3) Visual Studio Code: Visual Studio Code, commonly referred to as VS Code, is a highly versatile and extensible code editor developed by Microsoft. It has gained immense popularity in the software development community, not only for its support of multiple programming languages but also for its role in mobile app development, including React Native. VS Code offers syntax highlighting, code navigation, and debugging tools that are indispensable for React Native development. Developers can write and debug code efficiently and take advantage of a vast library of extensions to tailor the editor to their specific needs. It is a lightweight but powerful tool for web and mobile app development, offering a seamless and customizable development experience. It was chosen and being used to compile React-Native codes for our project.

Fig. 4. Logo of Xcode

4) Xcode: Xcode is Apple's official integrated development environment (IDE) specifically designed for macOS and iOS app development. It serves as a comprehensive toolset for developing applications for Apple platforms, including iOS, macOS, watchOS, and tvOS. Xcode includes a code editor with features like code completion, code analysis, and integrated debugging, making it an indispensable tool for writing code for Apple's ecosystem. It also provides a graphical interface builder that simplifies the design of user interfaces for iOS apps. Furthermore, Xcode streamlines the app development process by offering tools for testing, profiling, and app distribution through the App Store. It is a complete package for building, testing, and deploying high-quality applications on Apple devices.

Fig. 5. Logo of PyCharm

5) PyCharm: PyCharm, also developed by JetBrains, is an indispensable integrated development environment (IDE) for Python developers. It offers a set of features designed to streamline Python software development. PyCharm's code editor includes intelligent code completion, which suggests code snippets and identifies potential errors as you type. This feature significantly improves code quality and development speed. The IDE also supports various Python frameworks and libraries, making it easier to work with popular technologies like Django and Flask. It also simplifies virtual environment management, which is crucial for Python project isolation and package management.

Fig. 6. Logo of Jupyter Notebook

6) Jupyter Notebook: Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It's widely used for data analysis, scientific computing, and machine learning. Jupyter Notebooks support various programming languages, including Python.. These notebooks are composed of cells that can contain code, text, or visual outputs, making it a valuable tool for creating reproducible research and data-driven documents. It provides an interactive and user-friendly environment for data exploration, analysis, and sharing. Jupyter Notebook is a preferred among data scientists, researchers, and educators for its flexibility and interactivity.

Fig. 7. Logo of Expo

7) Expo: Expo is a set of tools and services that simplifies the development of React Native applications. It provides a variety of pre-built components and libraries, making it easier for developers to create mobile apps without the need for

extensive native code development. Expo also offers a development client that allows developers to preview their applications on physical devices without the need for complex setup. It's a valuable tool for rapidly prototyping and developing mobile applications and can be especially useful for developers looking for a simplified and accelerated development process.

Fig. 8. Logo of AWS EC2

8) AWS EC2: AWS EC2, part of Amazon Web Services, is a scalable and flexible cloud computing platform that provides virtual servers known as instances. These instances can be customized to run various operating systems and applications, making it a cornerstone for cloud-based software development. Developers can launch, manage, and scale virtual machines based on their needs. AWS EC2 offers various instance types optimized for different workloads, including compute, memory, and storage-optimized instances, enabling developers to choose the right infrastructure for their applications. It plays a critical role in building and deploying web applications, backend services, and other software solutions in the cloud.

Fig. 9. Logo of Miniforge

9) Miniforge: Miniforge is a distribution of the popular Conda package manager, optimized for minimalism and efficiency. Conda is used for managing packages and environments in data science and scientific computing. Miniforge is designed for users who want to create lightweight, customizable environments for specific tasks. It allows you to create isolated and reproducible environments for different Python projects, making it easier to manage dependencies and packages. Miniforge is especially valuable for data scientists and researchers who need to create custom Python environments for their work and prefer a minimalistic approach to package management. It was used in our project to create virtual environments for machine learning codes.

Fig. 10. Logo of DataGrip

10) DataGrip: DataGrip, developed by JetBrains, is a specialized database management tool essential in software development. It supports multiple database systems, making it an all-in-one solution for related tasks. DataGrip offers a wide range of features, including SQL code development, schema visualization, and data exploration, simplifying database management through a unified interface for various engines. It enables developers to write and execute SQL queries, analyze data structures, and optimize performance. Intelligent code completion, real-time analysis, and database navigation features expedite development and management.

Fig. 11. Logo of Postman

11) Postman: Postman is a versatile tool for simplifying API testing and working in software development. It serves as an API development and testing platform, enabling developers to create, send, and monitor HTTP requests. Acting as an API client, Postman allows developers to interact with API endpoints, test functionality, and ensure correct operation. It supports automation by enabling the creation of request collections and automated testing. Postman offers a user-friendly interface for request building and organization, facilitating collaboration between developers and testers.

Fig. 12. Logo of Google Colab

12) Google Colab: Google Colab is a cloud-based platform for machine learning and data analysis that provides free access to a virtual machine with GPUs. It runs in a web browser and offers a Jupyter Notebook interface, making it a convenient tool for writing and executing code for data science and machine learning projects. Google Colab allows users to run Python code, perform data analysis, and train

machine learning models using GPU acceleration, which can significantly speed up computations. It's a powerful resource for data scientists and machine learning engineers who need a cost-effective way to perform compute-intensive tasks without investing in local hardware.

Fig. 13. Logo of Nugu Developers

13) NUGU Playbuilder: NUGU Playbuilder is a development platform provided by NUGU, an AI assistant developed by SK Telecom. It is specifically designed for creating voice-activated and conversational AI applications and services. NUGU Playbuilder simplifies the process of building voice interfaces and chatbots, allowing developers to design and implement natural language interactions. This platform provides tools for creating voice scripts, defining dialogue flows, and integrating with various data sources and services, making it a valuable asset for developers looking to create AI-powered applications and services.

Fig. 14. Logo of Figma

14) Figma: Figma is a popular web-based design and prototyping tool known for its collaborative and user-friendly features. It empowers designers to create user interfaces, prototypes, and design components with real-time collaboration, ideal for cross-functional teams. It simplifies sharing design assets with developers, providing a platform for design handoff. Developers can inspect design specs, export assets, and collaborate with designers in real-time. Figma's flexibility and cloud-based approach facilitate seamless communication between designers and developers throughout a project's design and development phases.

15) Notion: Notion is a versatile workspace that caters to the diverse needs of software development teams. It combines note-taking, task management, knowledge sharing, and collaborative documentation. Notion offers a flexible and customizable platform for organizing information, managing

Fig. 15. Logo of Notion

projects, and sharing knowledge. Teams can structure their work using pages, databases, and boards, with support for various data types like text, tables, files, and multimedia content. Notion excels in project management, enabling task lists, reminders, and collaboration on project boards. Notion's seamless collaboration features concurrent work on the same page, making it a valuable tool for collaboration.

Fig. 16. Logo of Slack

16) Slack: Slack is a highly popular communication platform designed to enhance collaboration and communication within software development teams. It offers real-time messaging, file sharing, and integration with various software tools. Slack simplifies team communication by providing chat channels for different topics or teams, direct messaging, and file sharing capabilities. Developers can create channels dedicated to specific projects or discussions, making it easy to organize and find conversations. One of Slack's key strengths is its extensive library of integrations with third-party tools. It allows teams to connect other software tools, like GitHub and Google Drive, directly to their Slack workspace, streamlining workflows and enhancing productivity. Slack also supports notification customization, ensuring that team members receive updates and alerts relevant to their roles and responsibilities. It plays a crucial role in fostering efficient communication, project discussion, and workflow optimization within software development teams.

Fig. 17. Logo of Overleaf

17) Overleaf: Overleaf is a web-based platform designed for collaborative scientific writing and document preparation, especially for LaTeX documents. It simplifies the process of creating complex documents with mathematical equations, tables, and references. Overleaf offers real-time collaboration features, enabling multiple authors to work on the same document simultaneously. It provides a rich text editor with LaTeX support and a compilation engine, allowing authors to see real-time previews of their documents. It's a valuable tool for researchers and academics who need to create and collaborate on scholarly documents and reports. Also, it has a feature where it can be synced with GitHub and committed to a repository.

Fig. 18. Logo of TeXLive

18) TeXLive: TeX Live is a comprehensive distribution of the TeX typesetting system, which is widely used for document preparation in fields such as scientific and technical publishing, academia, and literature. It is cross-platform, supporting Windows, macOS, and Linux, and provides a vast collection of packages and fonts to address a wide range of document types and formats. TeX Live is renowned for its high-quality typesetting and is often used to create documents with complex layouts, mathematical notation, and bibliographies. It is the go-to choice for authors and researchers who require precise control over the visual appearance of their documents.

V. Specification

A. Loading Page

When a application communicates with servers, a delay is inevitable in the process. During the delay, it might show just a blank screen to users. To prevent users from getting a blank page, a loading screen will appear instead. This applies to all functions that require server communication including login, register, space creation, space connection and etc.

This loading page UI is made by 'skeleton' from React Native. While receiving data from the server, components without received data shows that it is loading. In other words, while server is processing and receiving data after a http request has been made by the client. It can make a seamless and user friendly UI by using the loading screen.

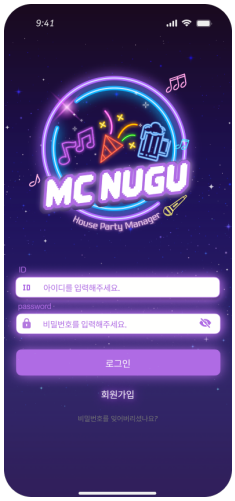

Fig. 19. Login Page

B. Login Page

Logging in is required for users to have access to functions of the application. Without a login, it is impossible to use any of the functions of the app. If the application is not logged in, users can create their own account or login using an account they have created before.

In a back-end point of view, The client receives an ID and password from the user and sends a request to the server when a button is pressed. On the server, it checks if there is 'Member' data in the DB and, if it exists, returns a security token known as JWT. The token expires after a certain time for security reasons and needs to be refreshed. The client stores the JWT and includes it in the header of every request sent.

In a front-end point of view, When a user clicks on the ID input field or the password input field, the input field's border is emphasized, and a keyboard appears. The screen moves up when the keyboard appears to ensure that it does not cover the screen. If the user enters their ID and then presses the next button on the keyboard, it automatically transits to the password input field. The password displayed on the screen is encrypted while it is being entered. Both the ID and password must be entered for the login button to become active. When the login button is pressed, it changes to a loading screen while the login process is executed on the server. After the login process is completed, the screen transitions to a page with a list of spaces. If the login fails, an Alert message is displayed, and the user remains on the login screen.

C. Register Page

Users can sign up by setting a user id and password of their preference. Signing in requires input of the given information in the sign up process such as user id, password and name. There are boxes for input of the ID, name, nickname, password and password check. ID is the one that should be input in the login process, name refers to real name, nickname will be the name that will be shown to other users, password is also used

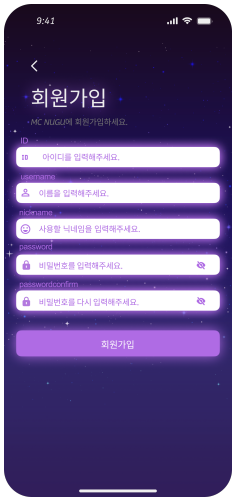

Fig. 20. Login Page

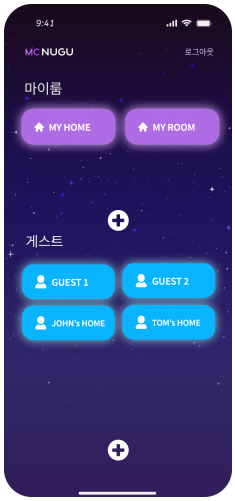

Fig. 21. Space List Page

in the login process and password check is used to double check if the user made their password as intended.

Initially, on the client side, there are character limits and a check to ensure that the password and password check fields match. Once the client-side validation is successful, the client sends these fields to the server because client-side checks can be easily bypassed. On the server side, there is a secondary validation for each field to check for errors. If there are no errors, the data is saved to the Member database.

For the UI which is similar to the login process, clicking on an input field highlights it, brings up the keyboard, and ensures that the keyboard doesn't cover the screen. Even if the user doesn't directly click on an input field, pressing the next button will move to the next input field. The password and password confirm fields are displayed in an encrypted format. If any of the fields (name, nickname, ID, password, password confirm) are left empty or if the password and password confirm do not match, the sign-up button remains inactive. Only when all input fields are correctly filled does the sign-up button become active. Upon successful registration, the screen transits to the login page. If registration fails, an alert message is displayed, and the user remains on the registration screen.

D. Space List Page

"Space" refers to any type of place where parties can be held and these spaces can be added to the application. Page with the list of spaces is where users can outlook the spaces that are registered to the app. It is divided into two parts; first is 'My Home ', which is the list of spaces that are owned by the user and registered to the application. Second is 'Guest' which has a list of spaces that the users are as guests by inputting the entrance code of spaces. Also there are buttons to activate pop-ups that allow users to add a new space or a pop-up for guest users to input the space code they have received.

The JWT, sent in the header, is decoded to extract the user's ID. Using this ID, both the list of spaces owned by the member and the list of spaces the member has joined as a guest are retrieved.

On the screen, the space list is displayed in two columns, with a "My Home" section and a "Guest" section, clearly distinguishing between the spaces the member owns and the spaces they have joined as a guest. Clicking the "Create Space" button opens a pop-up screen for creating a space, and clicking the "Enter Code" button opens a pop-up screen for entering a code to join an existing space. Space owners will see their spacess listed under "My Home," while guests will see records of the spaces they have joined under the "Guest" section. Clicking the "Log Out" button will transition the screen back to the initial login screen.

E. Space Registration Pop-Up

For a space to be ready to use in the application, it needs to be registered firsthand. The space registration pop-up can be accessed via the space list page. By activating the pop-up, space owners can input the name of the space and receive a invitation code for the space. After this process is valid, the new space should be added to the space list page.

On the screen, the user inputs a room name, and a request is sent to the server to store it in the Home database. During this process, a room ID is generated along with a 4-digit random code that will be used for guest access.

A pop-up screen for entering the space title is displayed. While this pop-up is open, the background of the room selection screen becomes opaque, and elements on the room selection screen cannot be interacted until the pop-up is closed. If the user clicks the "Back" button, they return to the space selection screen. If they enter a valid space title and click the "Confirm" button, a space is created, and the screen transitions to the Dashboard Page of the newly created space. If the space title is not entered correctly, a space creation failure alert is displayed, and the pop-up screen remains open.

F. Space Entrance Pop-Up

Space entrance pop-up is the first step for guest users to be able to gain allowance to spaces. This pop-up can be accessed from the space list page. A space user can gain access to spaces with the code provided by the owner. By having the code, the user also has access to the lighting and appliances pre-registered by the Space owner. Upon inserting a valid space code, it will add a new space to the guest section of the space list page. Now users are allowed on the spaces.

On the screen, when a code is entered, a request is sent to the server. The server searches the Home database for a matching entry. It also retrieves the user's ID from the JWT and fetches Member information. Once a match is found, the server stores the relationship between the Home and the Member in the Guest database, marking the user as a guest in that particular space.

The process for entering a space using a code is similar to the space creation pop-up. After entering a valid code and clicking the "Confirm" button, the user enters the corresponding space, and the initial page, the Dashboard Page, is displayed. If the code is not entered correctly, an alert message is shown, prompting the user to re-enter the code, and the pop-up screen remains open.

G. Dashboard (Main Page)

The Dashboard page is the main page that is first shown to users when entering a space. The main function of the dashboard page is showing an overview of the ongoing party. To help this objective, users can control the music playing from an audio device connected to the application, if there is any. Also there is a shared playlist where party participants can add desired songs to the list. In addition, there are information on the status of the space such as room temperature and air quality. These data can be sourced from the appliances that are connected to the space. Finally at the top there is the name of the space, exit space button, space code and a brief announcement bar. Name and code for the space is based on the data from the database and exiting button leads users back to the space list page. For the announcement bar, any user can post up to 25 characters and all users in the space can see the announcement.

It's a screen composed of tabs, and from the user's perspective, it's not ideal to send requests every time you switch tabs, as it can cause delays. Therefore, as soon as you enter this screen, all the data for the space, including lighting and appliance lists, as well as the game list, is fetched all at once. This data is stored in a context for later use.

Announcements are displayed as text animations that move from the left side of the screen to the right side, taking 1 minute. There's a button to return to the room selection screen, allowing users to navigate back. On the screen, the album cover of the music currently playing on the AI speaker is displayed, along with music playback options and playlists.

H. Lightning & Appliance Dashboard

The appliance tab is the second tab of the application, which has functions that controls the appliances that are connected by the owner. The appliances are categorized into two groups: those with lighting and those without lighting. This page can give an overview of all the lighting and non-lighting appliances of the space. The dashboard consists of 2 identical pages for each type of appliance.

In the Lighting tab, only appliances with the "light" variable set to true in the appliance list data are displayed. Users can enter their desired ambiance in an input field, and based on the current number of people in the space and the ambiance input, they receive lighting RGB values and effects recommendations from a generative AI. These recommendations are then transmitted to all the lights using Matter for configuration.

In the appliance tab, appliances without lighting features are able to be controlled. The specific functions vary on what type of appliances are added to the tab. It will be specified on later specifications but one example could be a temperature adjustment for an air conditioner.

All appliance buttons are displayed on the screen in two columns and support simple on/off functionality. Tapping each appliance button takes the user to a detailed control screen for each individual appliance. By using the tabs at the top of the screen, users can switch to a screen for controlling appliances without lighting. "Add Lighting" or "Add Appliance" button allows users to add new appliances. Users can enter the desired lighting in an input field and, by pressing the keyboard's completion button, can obtain suitable lighting effects through OpenAI API and the server.

I. Lighting Device & Appliance Registration Pop-Up

In order to control the appliances from the app, it needs to be registered initially. The registration pop-up will receive information about appliances from the owner ultimately sending the data to the server. Information needed for registration are the serial numbers and name of appliance. In an input bar on the pop-up serial numbers and names are entered. The server automatically categorizes the appliance based on the serial number, checking if it has lighting or determining its type.

Tapping the serial number input field brings up the numerical keyboard, while tapping the name input field displays the Korean keyboard. The "Back" button allows users to return to the lighting screen, and after entering the information correctly, clicking the "Confirm" button results in the creation of a new lighting appliance. If the information is not entered correctly, the lighting appliance is not created, and a "Lighting(Appliance) Creation Failure" alert message is displayed.

J. Lighting Control Page

Using a standardized format with Matter, the frontend sends requests to the appliance in a structured format based on the input it receives.

This screen allows you to control detailed lighting effects for appliances with lighting. At the top of the screen, the icon and

name of the selected lighting appliance are displayed. You can control not only the on/off function of the lighting appliance but also check and modify features such as color, brightness, and usage.

K. Appliance Control Page

For all appliances, you have the on/off function. If it's an air conditioner, you can control temperature, fan speed, energy-saving mode, and reservation functions. If it's an air purifier, you can control fan speed, view particulate matter (PM) levels, and fine particulate matter (PM2.5) levels. If it's a steamer, you can select courses and view the timer. For robot vacuum cleaners, you can choose cleaning modes, select strong, medium, or weak power, and enable quiet mode. Each appliance has its specific and detailed control features.

L. Game Dashboard

The game dashboard is the gateway to playing any of the game provided by the application. The main function is to show the list of games available and by pressing on a component of a list, it will activate the game. The details of the games such as name, the logic of the game are all coming from the server, which means that any kind of game can be added anytime on the back-end.

Upon entering the main tab, the data context received is used to display the list of games on the screen. Clicking on a game in the list creates a game room (Game_Room DB), with the first person to click becoming the game's host. Subsequently, they are connected to the host's screen. (Only one game room can be created within a space.) Clicking the "Join" button allows a user to join a game room currently in progress in the space. Subsequently, they are connected to the participant's screen. The game list received from the server is displayed in a single column on the screen.

M. Game Host Page

The game host page is a page for the first person to press a game on the game list. By being the first person pressing, it automatically makes them the host of the game. The host's page consists of all information needed to run the game such as game question, game answer, go to next quiz, end game and a serial list of users based on the speed of pressing the answering button. The host can give user chances to answer according to the list of users and verify if the player has got the answer right. If the answer is correct or nobody got the answer or any situation that regards to move on, the host can press the next game button.

Upon entering the screen, a WebSocket connection is established, and a request with messageType = ENTER is sent. When the "Next Question" button is pressed, the Flask server fetches the question for the selected game, sends it over WebSocket to display it for all participants, and sends a request with messageType = NEXT. When the "Reveal Answer" button is pressed, the answer to the received question is sent over WebSocket to display it for all participants, and a request with messageType = ANSWER is sent. At the bottom of the screen, the nickname of the participant at the top of the first-come, first-served list is displayed. Clicking the "Next" button removes the next person from the list.

The name of the selected game is displayed in the top left corner of the screen, while the "End Game" and "Next Question" buttons are positioned in the top right corner. The game screen may vary depending on the type of game, with images displayed for games that require them and a default image for games that do not. At the bottom of the screen, a list of participants in the game, organized by arrival order, is displayed.

N. Game Participant Page

Game participant page is for users that are actually playing the game. The same page is displayed to all users. The main functions provided are to see what the question or quiz is, and a button that gives you the chance to answer these questions.

The screen for participants is similar to the game host page but lacks the "End Game" and "Next Question" buttons. When participants press a button, they are added to the list in the order in which they respond.

Upon entering the screen, a WebSocket connection is established, and a message with messageType = ENTER is sent. When the game host displays a question, all participants automatically receive a request and see the question. Pressing the "Answer" button adds the participant to the list in the Game_Room DB in a first-come, first-served manner.

O. App Setting Page

App setting page is a page where any kind of settings regarding the app can be modified, such as logging out. The main difference and function of the application's setting is that it has buttons regarding the refreshing of the space code.

When you press the "Code Refresh" button, the room's access code is randomly changed, and at the same time, all guests who were invited to the room are removed.

Pressing the "Code Refresh" button causes all user's room access permissions to expire, and they are redirected to the room selection screen.

VI. ARCHITECTURE DESIGN

VII. USE CASES

VIII. CONCLUSION & DISCUSSION

ACKNOWLEDGMENT