



SMART REFRIGERATOR SYSTEM

Software requirements specification

2022. 04. 30

Introduction to Software Engineering

Team 10

Team Leader	Kwak Junyeong
Team Member	Kim Seyoung
	Kim Seongjun
	Park Suyeon
	Park Woohyeon
	Park Junyeong
	Amirah

Table of Contents

1. Introduction	3
1.1 Purpose	3
1.2 Document Conventions	3
1.3 Intended Audience and Reading Suggestions	3
1.4 Product Scope	3
1.5 References	4
2. Overall Description	5
2.1 Product Perspective	5
2.2 Product Functions	5
2.3 User Classes and Characteristics	6
2.4 Operating Environment	7
2.5 Design and Implementation Constraints	7
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
3. External Interface Requirements	8
3.1 User Interfaces	8
3.2 Hardware Interfaces	21
3.3 Software Interfaces	21
3.4 Communications Interfaces	25
4. System Features	26
4.1 Use case	26
4.2 Database Requirements	35
5. Other Non-functional Requirements	40
5.1 Product requirements	40
5.2 Organizational Requirements	41
5.3 External Requirements	42
6. Appendix A: Glossary	43

1. Introduction

1.1. Purpose

This document specifies requirements for the provision of the 'Smart Refrigerator System'. As the pandemic situation has turned eat-outs to risk, the need for eat-at-homes has gradually increased. However, preparing a meal is not easy for many people. Resolving this difficulty is what our system aims to do. By providing services such as 'Ingredient cognition', 'Recipe Recommendation', 'Cooking guidance' and 'Auto Thawing', our system aims to manage the user's diet in a smart way. This service is designed and implemented under the supervision of the professor and TAs of 'Introduction to Software Engineering' class by team 10 at Sungkyunkwan University. Team 10 will be implementing and designing the system referring to this documentation, also being the main reader of this document.

1.2. Document Conventions

This document is divided into 6 chapters excluding glossaries in which each chapter are differentiated by headings in **bold, Title Case Heading**.

Level	Description
1. First Level Heading	Describe the topic and indicate the beginning of a chapter
1.1. Second Level Heading	Breaks a chapter into a few smaller sections
1.1.1. Third level Heading	Breaks a section into smaller sections

1.3. Intended Audience and Reading Suggestions

For developers, this document will be the guide for the implementation of this system. Specification of the system requirement will be as detailed as possible, to prevent errors that occur due to miscommunication. For those who may take part of this project to promote sales of this system, acquainting oneself with the features of the system by reading this document will be of great help when devising a sales plan.

1.4. Product Scope

Camera attached inside the refrigerator will keep track of the information of the ingredients stored inside it. Quantity, expiration date, and the ingredients name will also be included. Visualized details of the contents inside the refrigerator will enable the user to also keep track of the ingredients in a more comfortable fashion. Based on that information, suitable cooking recipes will be recommended along with cooking instructions as a guide. This helps encourage user to eat healthier. Providing recipes using ingredients that is already in the refrigerator helps

reduce the cost of eating out, or shopping for more ingredients, thus increasing the usage of the system.

1.5. References

Hanjin Kim, Seunggi Lee, Won-Tae Kim, 2018. A Smart Refrigerator System based on Internet of Things. ISSN: 1266-7244

<http://www.epnc.co.kr/news/articleView.html?idxno=46539>

2. Overall Description

2.1. Product Perspective

As the pandemic situation has turned eat-outs to risk, the need for eat-at-homes has gradually increased. However, preparing a meal is not easy for many people. Resolving this difficulty is what our system aims to do. If the refrigerator is brand-new and has nothing in it, recommendations of the most popular meals will be shown, encouraging the user to go out shopping. Since the system with this feature not been released before, the software itself is new.

2.1.1. System Interfaces

The status of ingredients that are in the refrigerator will be stored in the user's local data storage using the AWS cloud server, MVC, and MySQL. The information of recipes is also stored in the local storage.

2.1.2. User Interfaces

Application on the mobile phone will show a list of operations that the system can perform such as displaying the status of the ingredients in the refrigerator or showing recommendations based on the ingredients. Users can decide which operations to perform through the application.

User can also communicate with the server via the screen attached to the refrigerator

2.1.3. Hardware Interfaces

This system is intended for both the mobile phones from Apple and Samsung that uses IOS and Android as its operating system

2.1.4. Software Interfaces

This system supports Android 8.0 or higher and IOS version 13.0 or higher.

2.1.5. Communication Interfaces

The devices are to communicate through the server, following the HTTP protocol using JSON format.

2.2. Product Functions

2.2.1. Ingredient Cognition

By using Computer Vision to the system, a camera attached inside the refrigerator will perceive incoming ingredients with details such as its name, type, (expected) expiration date, its storage location, its status, and so on.

2.2.2. Recipe Recommendation

Referring to the data collected from 'ingredient cognition', the system will provide suggestions of several recipes. After the recipe is chosen, the system will provide cooking instructions as a guide.

2.2.3. Cooking Guidance

Cooking guidance is provided through a video that is linked to the cooking recipe. The video can be paused and resumed according to user. It can be viewed on a user's mobile phone or on the screen attached to the refrigerator.

2.2.4. Auto Thawing System

Registered ingredients to be thawed will be automatically put in the thawing sections, the temperature of the section will be set to suitable environment for thawing. After the ingredient is completely thawed, users will be notified for further action.

2.2.5. Account Management

Provides login service to provide a more individually focused and secure system. To sign up to the system, the provision of E-mail address, password, and the ID of the refrigerator is required.

2.2.6.Notification

Messages sent from other features will be shown on the push message such as the notification for the completion of thawing,

2.3. User Classes and Characteristics

2.3.1.System Administrator

System administrators are expected to have sufficient knowledge on software development. They are also expected to be well-informed of the overall structure of the system. It is also assumed that the system administrators have the capability to reflect given requirements to the software with decent effort. Since discussions of both technical, non-technical issues will be made, communication abilities are also required.

2.3.2.User

Since all the profits are to be expected from the users, users' satisfaction is important. Users are expected to be capable of reading in Korean or English. Expertise in software is not necessary however, they are expected to be capable of using smartphones.

2.3.3.Vendor

Vendors are expected to be more informed than the users but less than the system administrators. The capability to well describe the features in it is necessary. This system is planned to be available both online and offline, so knowledge of using both domains is also necessary. Since it is the users that are to be those that use this service the most, satisfaction from vendors may also be desired, but not necessarily.

2.4. Operating Environment

The application for the software will operate on the versions mentioned below.

Android 8.0 or higher

IOS version 13.0 or higher

The embedded system must have at least 2GB RAM, i5-4590/FX 8350 CPU.

The internal camera must perform 640*480 pixels of resolution and 10fps of frame speed.

2.5. Design and Implementation Constraints

All the implementations and designs will be made with reference to this document. Minor decisions can be made by the personnel who oversees the part, but the items that are mentioned below must be followed.

- Only the Technologies that have widely been authorized should be used.
- Use open-source software whenever possible
- Consideration of the system cost and the maintenance cost should be done.
- Attaching annotations to codes is highly encouraged
- Development should be done under the environment mentioned in this document.

2.6. User Documentation

User manuals will be provided as the user purchases a refrigerator that has this system. This manual will include the way of installing an application that will provide all the features mentioned, and how to use it. It also has explanations related to the refrigerator. The manual will include an explanation of how to use the refrigerator, the roles of each section, explanations of each feature, so on and so forth...

2.7. Assumptions and Dependencies

The system explained in this document is assumed to be implemented on the Android or IOS device and open source. The detailed constraints are mentioned on 2.3 and 2.4.

3. External Interface Requirements

3.1. User Interfaces

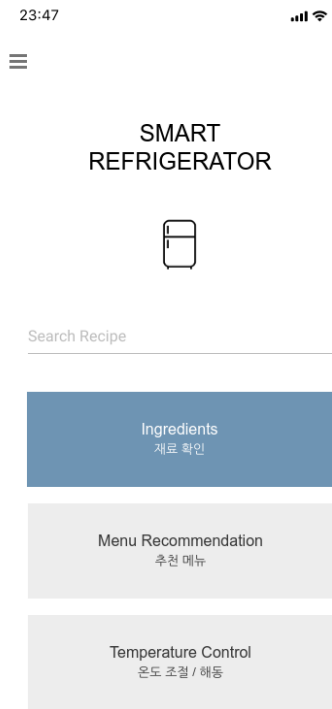
[Table 1] App start interface

Name	Flutter app – start & Login - interface	
Purpose/Description	User chooses whether to login or register before using this program -> for Refrigerator information Once login is complete keep login until expiration date	
Input source/ Output destination	User -> Server	
Time/ Velocity	N/A	
Relationship with other input/outputs	N/A	
Format and configuration		<ol style="list-style-type: none"> Registered user <ul style="list-style-type: none"> type id and password Click Sign in button Unregistered user <ul style="list-style-type: none"> Click Sign up button
Data type	Button, text_field, Image, query, json	
Instruction type	Instruction mapped to the button like onclick at button variable	
Exit Message	<p>“Login success”</p> <p>“Move to Register”</p>	

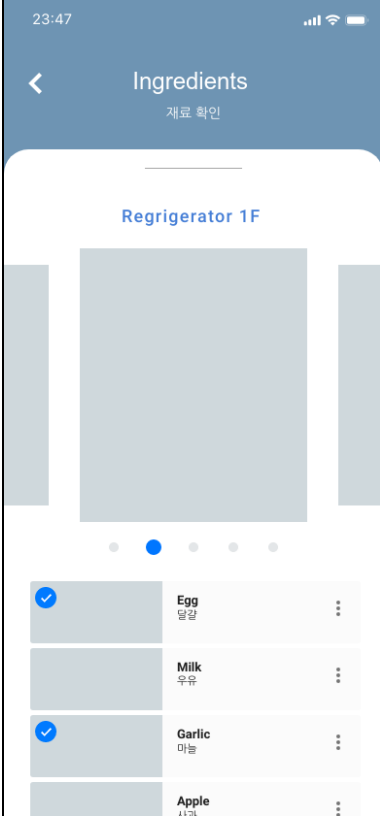
[Table 2] Register Interface

Name	Register Interface
Purpose/Description	To use the service of the system, users must be registered and logged in in which they can register easily by filling out the registration form
Input source/ Output destination	User -> Server
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration	 <ol style="list-style-type: none"> 1. fill out all registration form 2. click sign up 3. server sends email verification email verification will be the form of URL (using time stamp and user id hashing) 4. by clicking the URL, user ID will be activated 5. back to login page
Data type	String -> Json, button, text_field, query
Instruction type	Instruction mapped to the button
Exit message	“Register success, check your email”

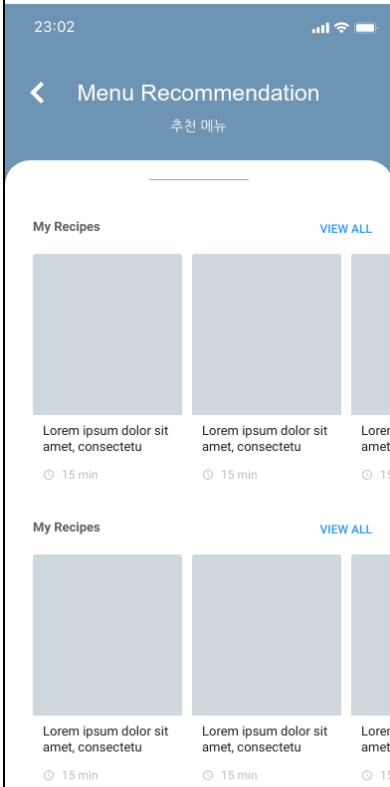
[Table 3] Main Interface

Name	Main Interface
Purpose/Description	User can choose one of many operations to be performed by the system.
Input source/ Output destination	User -> user device
Time/ Velocity	N/A
Relationship with other input/outputs	Through Search Recipe bar, system displays recommendation interface based on users' search. When clicking Menu recommendation, system displays recommendation interface based on user's preferences and ingredients in the refrigerator.
Format and configuration	<div>  </div> <ol style="list-style-type: none"> To search for a recipe <ul style="list-style-type: none"> Type in search bar Press Enter -> Menu Recommendation interface (Based on your search) To check ingredients <ul style="list-style-type: none"> click ingredients button To get menu recommendation <ul style="list-style-type: none"> click menu recommendation To thaw <ul style="list-style-type: none"> click temperature control Using left upper side button user can go setting page
Data type	Button, text_field,
Instruction type	Instruction mapping to button
Exit message	“Starting xxxxx” (check ingredients, menu recommendation etc..)


[Table 4] Ingredient interface

Name	Ingredient interface	
Purpose/Description	Contain information page about ingredients	
Input source/ Output destination	User Device -> Server -> User Device (showing ingredients) User -> User Device (clicking ingredients)	
Time/ Velocity	(Method 1) after entering this screen/ (Method 2) after entering main page (to reduce time to networking) /Dependent on network	
Relationship with other input/outputs	N/A	
Format and configuration		<ol style="list-style-type: none"> 1. App requests information related to ingredients 2. Server send information 3. Application display ingredients information <p>Exception: no network</p> <ul style="list-style-type: none"> • If Clients can't use the network skip 1~3 and display information using local data. • By clicking ingredients (like egg, milk) user can see detail information like Expiration date. • By sliding the screen upwards, screen will display other part of the refrigerator and the list at the bottom will refresh.
Data type	Json, String, List view, query, image	
Instruction type	Instruction (see detail information) mapping on List view's adapter	
Exit message	N/A	

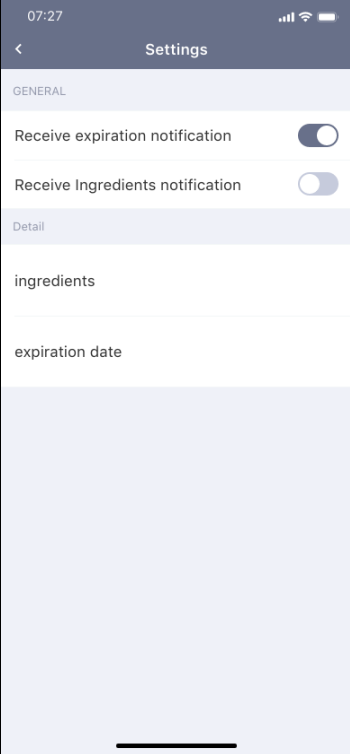
[Table 5] Menu recommendation interface

Name	Menu recommendation interface
Purpose/Description	uses client's preference and refrigerator's ingredients or search text
Input source/ Output destination	User -> User device (Clicking menu) User Device -> Server -> User Device (showing user menu)
Time/ Velocity	after entering this screen/Depends on network
Relationship with other input/outputs	Main menu's output
Format configuration and	<div>  </div> <ol style="list-style-type: none"> 1. App Requests information related to menu 2. Server send information using query (search by text and ingredients or user preference and ingredients) 3. Application gets data and parse data 4. Application display recommendation information <p>Ex1: no network</p> <ul style="list-style-type: none"> • Infinite loading on the bottom side. <p>Ex2: clicking menu</p> <ul style="list-style-type: none"> • By clicking menu (like spaghetti) user can see detail information like recipe, need ingredients and scarce materials using refrigerator's data <p>Upper side showing record of before</p> <p>Downside showing search result</p>
Data type	Json, String, List view, query, image
Instruction type	Instruction (see detail information) mapping on List view's adapter
Exit message	N/A

[Table 6] thawing controller Interface

Name	thawing controller Interface
Purpose/Description	Thawing controller Screen
Input source/ Output destination	Server -> User device (get thawing information) User -> User device -> User refrigerator (thawing control)
Time/ Velocity	after sending query/Dependent on network
Relationship with other input/outputs	N/A
Format configuration and	<div>  <p>The screenshot shows a mobile app interface titled 'Temperature controller'. It displays a list of four thawing parts: THAWING PART1 (돼지고기), THAWING PART2 (소고기), THAWING PART3 (연어장), and THAWING PART4 (연어장). Each part has a status indicator (a person icon) and a remaining time (e.g., 3월 14일 3시 33분). A message '해동중이 아닙니다' (Not thawing) is shown below each part.</p> </div> <ol style="list-style-type: none"> 1. Get Thawing information by server(request) 2. Screen shows thawing information (such as time remaining for each refrigerator compartment) <p>To thaw food</p> <ul style="list-style-type: none"> • Click thawing part • Enter desired time to thaw <p>Then device sends data to refrigerator at a fixed port</p> <p>Then The refrigerator asynchronously checks the port for information</p> <p>➔ The possibility of abuse, attack? -> maybe using authentication</p>
Data type	Json, String, query, image, List view
Instruction type	Instruction mapping on List view's adapter
Exit message	N/A

[Table 7] Setting interface

Name	Setting interface
Purpose/Description	User can change setting like on/off Expiration date reminder, Scarcity reminder of ingredients (target ingredients, number of ingredients)
Input source/ Output destination	User -> User Device -> Server
Time/ Velocity	N/A
Relationship with other input/outputs	For users to customize their notifications
Format and configuration	<div>  <p>1. To block notification</p> <p>Click general's switch -> dark side is enabled, opposite side is disabled</p> <p>Dark side -> enabled down button</p> <p>Opposite side -> disabled down button</p> <p>2. To change preference for expiration date</p> <ul style="list-style-type: none"> • click expiration date button • Then user can enter day:hour <p>3. To change preference for scarcity ingredients</p> <ul style="list-style-type: none"> • click ingredients button • Then user can search ingredients and click it and enter quantity to be notified <p>After getting out of this page, Device send this information to server to notify it</p> </div>
Data type	Json, String, button, switch, query
Instruction type	Instruction (see detail information) mapping on switch and button
Exit message	N/A

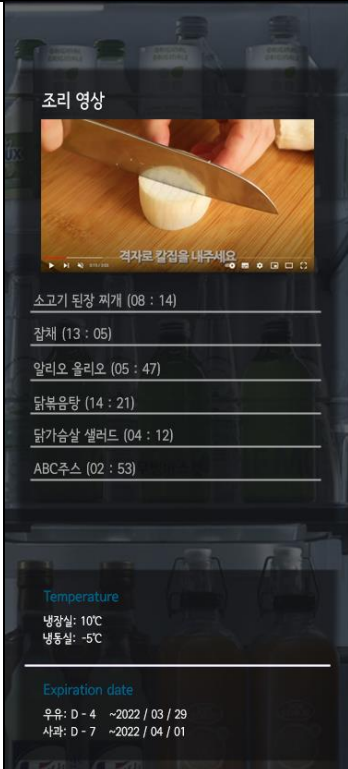
Refrigerator screen

[Table 8] Main Screen Interface

Name	Main Screen Interface
Purpose/Description	User chooses what to do by clicking on the refrigerator screen Additionally, lower side of screen shows the food that is nearing its expiration date except it's already past it
Input source/ Output destination	User -> User Device
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration	<div data-bbox="461 943 815 1720"> </div> <p>The program updates the lower side of screen when food nearly expired -> check database and change food that is nearing it</p> <ol style="list-style-type: none"> To thaw food <ul style="list-style-type: none"> • touch 해동 설정 • move to thawing controller interface To check ingredients <ul style="list-style-type: none"> • touch 유통기한 • move to ingredients interface To change temperature <ul style="list-style-type: none"> • touch 온도 조절 • move temperature controller interface To see recipe <ul style="list-style-type: none"> • touch 조리법 • move menu recommendation interface To control water purifier temperature

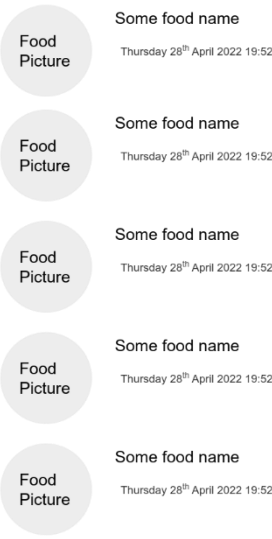
Name	Main Screen Interface
	<ul style="list-style-type: none"> • touch 정수기 • like 온도조절 button • 2 bars occur on both side of button
Data type	Button, text, date
Instruction type	Instruction mapping on button
Exit message	N/A

[Table 9] Cooking guide Interface

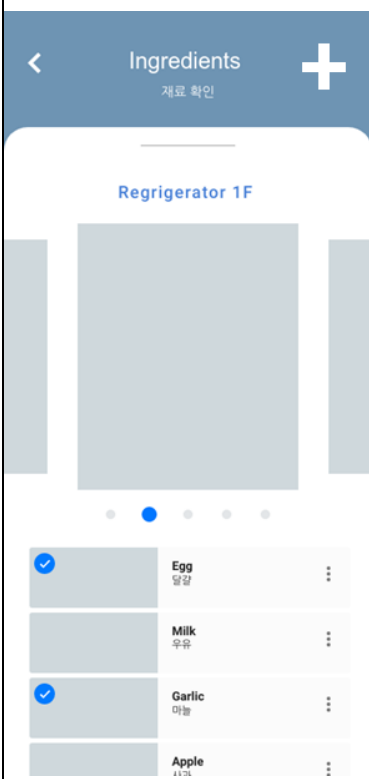
Name	Cooking guide interface
Purpose/Description	User can watch videos or cooking instructions that is stored in the database
Input source/ Output destination	User -> Refrigerator -> Server Server -> Refrigerator -> User
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format configuration and	 <ol style="list-style-type: none"> 1. User touches the '추천 메뉴' in the main menu. 2. User selects a recipe video from the recommended recipe list. 3. Server sends the video or text of selected recipe, and the screen on the refrigerator shows it.

Name	Cooking guide interface
Data type	Json, String, button, video, query
Instruction type	Instruction mapping on the button
Exit message	N/A

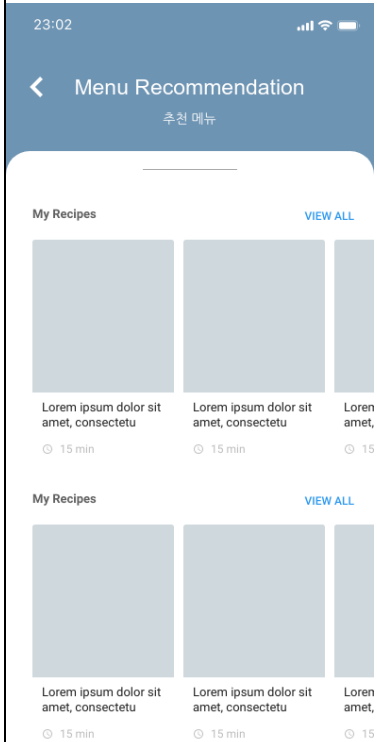
[Table 10] Thawing controller Interface at refrigerator

Name	thawing controller Interface
Purpose/Description	Similar to the one in the application, but slightly different internal behavior
Input source/ Output destination	User -> User refrigerator (thawing control) User refrigerator -> Server (to notify thawing control information)
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration	<div> <p>To thaw food</p> <ul style="list-style-type: none"> • Click thawing part • Enter desired time to thaw • Refrigerator start thawing <p>And sends data to server</p> </div> 
Data type	Json, String, query, image
Instruction type	Instruction mapping on each part
Exit message	N/A

[Table 11] Ingredient interface at refrigerator

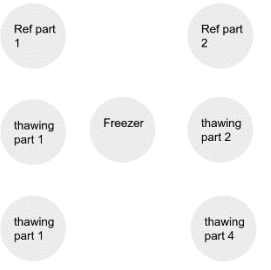
Name	Ingredient interface	
Purpose/Description	Similar to the one in the application, but slightly different internal behavior	
Input source/ Output destination	User -> User refrigerator User refrigerator -> Server (When using CV system or use Ingredients, or use plus button)	
Time/ Velocity	N/A	
Relationship with other input/outputs	N/A	
Format and configuration		<p>displays ingredients information</p> <p>when clicking ingredients (like egg, milk) user can see detailed information like Expiration date</p> <p>User can push data using plus button at upper right side of screen</p> <ul style="list-style-type: none"> • Click button • Then search bar appears • User types on the search bar • Refrigerator sends search data to server • Refrigerator gets data and display it • User chooses appropriate ingredient • type number Quantity of ingredient • choose part put it • then refrigerator send data at server
Data type	Json, String, query, image	
Instruction type	Instruction mapping on each part	
Exit message	N/A	

[Table 12] Menu recommendation interface at refrigerator

Name	Menu recommendation interface
Purpose/Description	Similar to the one in the application, but slightly different internal behavior
Input source/ Output destination	User -> User device (Clicking menu) User refrigerator -> Server -> User refrigerator (showing menu) User refrigerator -> Server -> User Device (Sending notification when choosing menu with not enough ingredients)
Time/ Velocity	after entering this screen/Dependent on network
Relationship with other input/outputs	N/A
Format configuration and	 <ol style="list-style-type: none"> 1. Program Requests information related to menu 2. Server sends information using query (search by text and ingredients or user preference and ingredients) 3. Application gets data and parse data 4. System displays recommendation information <p>Or user can search data using search bar, algorithm is same with upper side</p> <p>Ex1: no network Infinite loading at lower side</p> <p>Ex2: clicking menu</p> <p>By clicking menu (like spaghetti) user can see detail information like recipe, needed ingredients and scarce materials using refrigerator's data and choose this recipe then refrigerator send scarce materials to Server then server send this data to User device</p> <p>Upper side shows record of before lower side shows search result</p>
Data type	Json, String, query, image
Instruction type	Instruction (see detail information) mapping on each part

Name	Menu recommendation interface
Exit message	N/A

[Table 13] Temperature controller Interface

Name	Temperature controller Interface
Purpose/Description	User can change temperature of each part of refrigerator, but can't change if thawing is on (and also have upper bound and lower bound each part ex) freezer, it is between -20 ~ -10)
Input source/ Output destination	User -> User refrigerator
Time/ Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration	<div>  <p>User can change temperature each part of refrigerator but can't change thawing container</p> <p>When thawing is active</p> <ul style="list-style-type: none"> Click button And scroll bar is active left side of button (at left button), or right side of button (at right button), or upper side of button (at center button (Freezer)) <p>Also detail temperature(-21도) will appear at opposite side(right, left, down)</p> <ul style="list-style-type: none"> User can slide it then detail temperature is changed, user can set temperature user want </div>
Data type	Button, scroll, number
Instruction type	Instruction mapping on Button, scroll
Exit message	N/A

3.2. Hardware Interfaces

[Table 14] Hardware interface of applicable device for the system

Name	Applicable device for the system
Purpose/Description	Enable users to take advantage of the service provided by the system/Android OS/IOS Enabled Smartphone. (At least Android 8.0 or IOS 13.0)

[Table 15] Hardware interface of refrigerator for the system

Name	Applicable refrigerator for the system
Purpose/Description	The system is intended for a Smart Refrigerator. The Smart Refrigerator must include an embedded system, and the internal camera. The embedded system must have at least 2GB RAM, i5-4590/FX 8350 CPU. The internal camera must perform 640*480 pixels of resolution and 10fps of frame speed.

3.3. Software Interfaces

[Table 16] Software interface of MySQL

Name	MySQL Database
Purpose/Description	Query input/output for managing ingredients/recipes data.
Input source/ Output destination	Host server ↔ User Host server ↔ Refrigerator
Range/ Accuracy/ Margin of error	Depends on the performance of MySQL.
Unit	Query

Time/ Velocity	Instant reaction
Relationship with other input/ outputs	Related to all inputs/outputs from server
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

[Table 17] Software interface of ingredients detection with YOLO

Name	Ingredients detection with YOLO
Purpose/ Description	Pre-trained YOLO model detects the class of ingredients from the camera image.
Input source/ Output destination	Camera → Database
Range/ Accuracy/ Margin of error	Depends on the performance of the embedded CPU, Pre-trained YOLO model. Depends on the quality of image from the camera.
Unit	Image

Time/ Velocity	Instant reaction
Relationship with other input/ outputs	Output ingredient information becomes the base of the recipe recommendation system.
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Image, Class
Instruction type	Data handling
Exit message	N/A

[Table 18] Software interface of expiration date detection with Tesseract OCR

Name	Expiration date detection with Tesseract OCR
Purpose/ Description	Tesseract OCR detects the expiration date text in the ingredient label from the camera image.
Input source/ Output destination	Camera → Database
Range/ Accuracy/ Margin of error	Depends on the performance of the embedded CPU, Tesseract OCR. Depends on the quality of image from the camera.
Unit	Image

Time/ Velocity	Instant reaction
Relationship with other input/ outputs	Output expiration date information becomes the base of the recipe recommendation system.
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Image, Text
Instruction type	Data handling
Exit message	N/A

3.4. Communications Interfaces

[Table 19] Communication interface of Thaw system

Name	Thaw system
Purpose/ Description	Users order a refrigerator to thaw a specific room.
Input source/ Output destination	User → Refrigerator
Unit	Packet
Time/ Velocity	At least 10Mbps

Relationship with other input/ outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

[Table 20] Communication interface of Ingredients list

Name	Ingredient list
Purpose/ Description	Users query ingredients list or recommended recipes. Database replies to the query. Database notifies users of the ingredients that are about to expire.
Input source/ Output destination	User ↔ Database
Unit	Packet
Time/ Velocity	At least 10Mbps
Relationship with other input/ outputs	N/A

Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

4. System Features

4.1. Use Case

4.1.1.Account management

[Table 21] Use case of register

Use case name	Register
Actor	Unregistered User
Description	For the system to keep track of the personal storage state and provide all the functions, a new user will have to register in this system.
Normal Course	<ol style="list-style-type: none"> When the user enters the application, the start and login page comes up. The unregistered user selects the register menu to create an account. The user is redirected to the registration page. The user fills out the registration form which consists of <ul style="list-style-type: none"> E-mail address Password Refrigerator ID The server sends a verification e-mail. Clicking the URL in this e-mail, the user will be successfully registered in the system. With the activated account, the registered user can now make use of the smart refrigerator system. The application returns to the log-in page.

Pre-condition	<p>The user is unregistered.</p> <p>The user provides all information in a valid format.</p> <p>Verification through the e-mail is completed.</p>
Post-condition	The application goes back to the initial starting page.
Assumptions	N/A

[Table 22] Use case of log-in/out

Use case name	Log-in/out
Actor	Registered User
Description	The registered user may log-in to or log-out of the system.
Normal Course	<p><Log-in></p> <ol style="list-style-type: none"> The user can enter id and password from the start and login page. If the account information is correct, the user is logged in to the system and the main page will show up. <p><Log-out></p> <ol style="list-style-type: none"> If the user wants to get out of the system or switch to another account, the user may click the log-out button to exit.
Pre-condition	<p><Log-in></p> <p>The user has an account.</p> <p>The user has entered all the information correctly.</p> <p><Log-out></p> <p>The user is logged-in to the system.</p>
Post-condition	N/A
Assumptions	N/A

[Table 23] Use case of user settings

Use case name	User Settings
Actor	Registered User
Description	Each user may have personal preference for the notification settings. The user can set the conditions for expiration reminder and scarcity check.
Normal Course	<ol style="list-style-type: none"> From the main page menu, the user can access the setting page. The user can customize the following conditions: <ul style="list-style-type: none"> Expiration date reminder time to notify Scarcity target ingredient, quantity
Pre-condition	Time to notify expiration date has a system default value.
Post-condition	N/A
Assumptions	N/A

4.1.2. Ingredients management

[Table 24] Use case of add/delete

Use case name	Add/Delete
Actor	Registered User
Description	The user can add/delete ingredients just by putting them into/out of the refrigerator. Further operations are done automatically by the system.
Normal Course	<ol style="list-style-type: none"> The user puts new ingredients into the refrigerator. When the door is closed, the system detects it and takes a photo of the current state. The computer vision system scans the photo and organizes information into a tuple with: <ul style="list-style-type: none"> ingredient id ingredient name user ID img url storage expiration date quantity storage date

	4. The system administrator compares it to the previous state and updates information on the database.
Pre-condition	The door is closed.
Post-condition	The user ingredient data is updated.
Assumptions	N/A

[Table 25] Use case of ingredient check

Use case name	Ingredient Check
Actor	Registered User
Description	The user can check what is currently inside the refrigerator.
Normal Course	<ol style="list-style-type: none"> 1. The user enters the ingredient page. 2. The user can check the current state of the database which consists of the following columns. <ul style="list-style-type: none"> • ingredient name • img url • storage • expiration date • quantity • storage date
Pre-condition	The door is closed.
Post-condition	The user can add or modify the data.
Assumptions	N/A

[Table 26] Use case of ingredient information modification

Use case name	Ingredient Information Modification
Actor	Registered User
Description	The user can modify the registered information.
Normal Course	<ol style="list-style-type: none"> From the ingredients page, the user can add or modify information. The user can add arbitrary ingredients that the system has not processed. <ul style="list-style-type: none"> Simple searching for the ingredient name is done. The user enters other attributes. Storage and quantity are mandatory. The user can modify existing information. The user clicks the confirmation button. The system updates the user ingredient table.
Pre-condition	The current ingredient state has been loaded.
Post-condition	The user ingredient data is updated.
Assumptions	The computer vision system may not recognize all the details correctly.

4.1.3.Recipe Management

[Table 27] Use case of recipe recommendation

Use case name	Recipe Recommendation
Actor	Registered User
Description	The user may request the system to recommend a recipe, based on the current refrigerator state and user preference (search records).
Normal Course	<ol style="list-style-type: none"> The user may request a recommendation on the recommendation page. Through a simple algorithm, the system finds the most appropriate recipe. The followings are considered, multiplied by the particular weights: <ul style="list-style-type: none"> Ingredients currently in the refrigerator Days left to the expiration date of each User preference information and more The user can check the search results in a list display. By clicking each, the user can check the detailed recipe and required ingredients information.

Pre-condition	Information about personal preference (e.g. search records, past meal records, preferred class) is used if established.
Post-condition	The user may start a cooking guide mode. The user choice is stored for future use.
Assumptions	N/A

[Table 28] Use case of recipe search

Use case name	Recipe Search
Actor	Registered User
Description	The user may search for his desired recipe to get help from the guide.
Normal Course	<ol style="list-style-type: none"> 1. The user can search for some recipes with the menu or ingredient keyword. 2. Through searching, the system comes up with the search result. This searching process is based only on the keywords from 1. 3. The result is displayed as a list. 4. By clicking each, the user can check the detailed recipe and required ingredients information.
Pre-condition	N/A
Post-condition	The user may start a cooking guide mode. The search records are stored for future use.
Assumptions	N/A

[Table 29] Use case of cooking guide

Use case name	Cooking Guide
Actor	Registered User
Description	The user can cook with a recipe video guide displayed on the application.
Normal Course	<ol style="list-style-type: none"> 1. As a result of the recommendation or searching process, the user will be provided with the list of dishes. 2. The user may check the details by clicking each dish and select the recipe to start the cooking guide. 3. The application shows the chosen recipe in the chosen mode. 4. The user can cook with ease with the guidance.
Pre-condition	The requested recipe list is provided.
Post-condition	N/A
Assumptions	N/A

4.1.4. Temperature Management

[Table 30] Use case of temperature control

Use case name	Temperature Control
Actor	Registered User
Description	<p>The user may change the temperature of each compartment through the temperature control menu of the application.</p> <p>Options: refrigerator, freezer, and thaw-able compartment</p>
Pre-condition	N/A
Post-condition	The embedded system changes the temperature.
Assumptions	N/A

[Table 31] Use case of thaw reservation

Use case name	Thaw Reservation
Actor	Registered User
Description	The user can reserve thawing operation for each frozen ingredient or thaw-able compartment.
Normal Course	<ol style="list-style-type: none"> 1. The user may check ongoing and completed thawing processes from the thaw reservation page. 2. The user can make a thaw reservation with one of the following options: <ul style="list-style-type: none"> • The target ingredient and desired time for its use • Specific compartment 3. The system automatically adjusts the temperature depending on the database information for the thawing operation.
Pre-condition	The target ingredient should be inside the thaw-able compartment.
Post-condition	The embedded system changes the temperature.
Assumptions	N/A

4.1.5. Notification

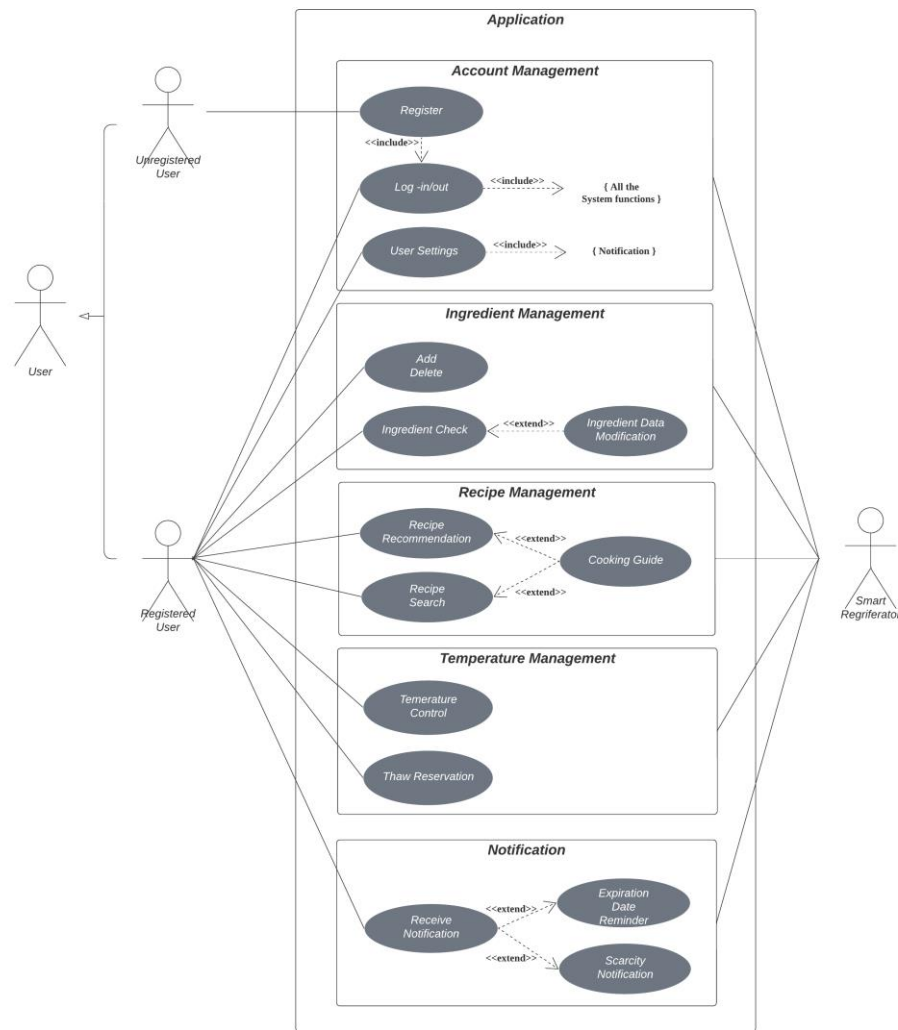
[Table 32] Use case of expiration date reminder

Use case name	Expiration Date Reminder
Actor	Registered User
Description	The system sends a notification when the expiration date is coming up.
Normal Course	<ol style="list-style-type: none"> 1. Using the database information, the system administrator checks if any ingredient is up to the coming-up date. Whether to receive this notification and how close to the expiration date the notification should be sent is determined by the user. 2. If any, the user receives a notification from the application.
Pre-condition	N/A

Post-condition	N/A
Assumptions	N/A

[Table 33] Use case of scarcity notification

Use case name	Scarcity Notification
Actor	Registered User
Description	The system sends a notification when any designated ingredient is scarce.
Normal Course	<ol style="list-style-type: none"> 1. The system administrator monitors the ingredients registered for the scarcity check. The list of ingredients to be monitored is set by the user. 2. If any, the user receives a notification from the application.
Pre-condition	Scarcity check ingredient is designated by the user.
Post-condition	N/A
Assumptions	N/A



[Figure 1] Use Case Diagram

4.2. Database Requirements

4.2.1.Data Dictionary

[Table 34] Data Dictionary - recipe

Field	Key	Constraint	Type	Description
recipe_id	PK	NOT NULL	int	ID of recipe
recipe_name		NOT NULL	varchar (25)	Name of dish with recipe
recipe_class		NOT NULL	varchar (25)	ex) Korean, Japanese

img_url		NOT NULL	varchar (25)	URL where image of dishes is stored
order_url		NOT NULL	varchar (50)	URL where guide video for cooking is stored

[Table 35] Data Dictionary - ingredient

Field	Key	Constraint	Type	Description
ingredient_id	PK	NOT NULL	int	ID of cooking ingredient
ingredient_name		NOT NULL	varchar (25)	Name of cooking ingredient
img_url		NOT NULL	varchar (25)	URL where image of ingredient is stored
expiration_date			int	Default expiration date by ingredient
quantity_notify		NOT NULL	int	Minimum quantity for notification to users

[Table 36] Data Dictionary - recipe ingredient

Field	Key	Constraint	Type	Description
ingredient_id	FK	NOT NULL	int	ID of cooking ingredient
recipe_id	FK	NOT NULL	int	ID of recipe
quantity		NOT NULL	int	quantity of ingredient that needed in recipe

[Table 37] Data Dictionary - user ingredient

Field	Key	Constraint	Type	Description
ingredient_id	FK	NOT NULL	int	ID of cooking ingredient
user_id	FK	NOT NULL	varchar (25)	User ID
ingredient_name		NOT NULL	varchar (25)	Name of cooking ingredient
img_url		NOT NULL	varchar (25)	URL where image of material is stored
storage		NOT NULL	varchar (25)	Which storage material store ex) refrigerated / frozen / thawed
expiration_date		NOT NULL	date	expiration date of material
quantity		NOT NULL	int	Quantity of material that is stored
storage_date		NOT NULL	date	Start date of storage

[Table 38] Data Dictionary - thawed ingredient

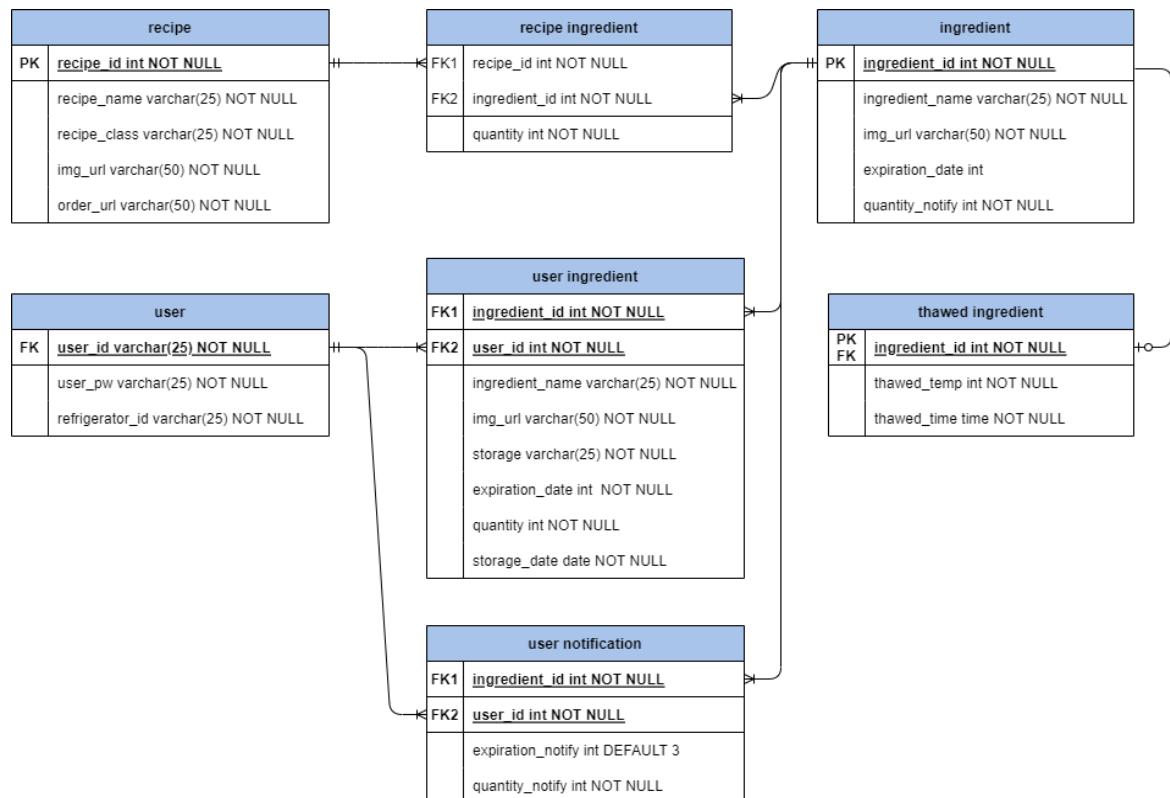
Field	Key	Constraint	Type	Description
ingredient_id	FK	NOT NULL	int	ID of cooking ingredient
thawed_time		NOT NULL	time	Time for thawing ingredient
thawed_temp		NOT NULL	int	Temperature for thawing ingredient

[Table 39] Data Dictionary - user notification

Field	Key	Constraint	Type	Description
ingredient_id	FK	NOT NULL	int	ID of cooking ingredient
user_id	FK	NOT NULL	varchar (25)	User ID
expiration_notify		DEFAULT 3	int	When the system informs the user before the expiration date.
quantity_notify		NOT NULL	int	Minimum quantity for notification to users

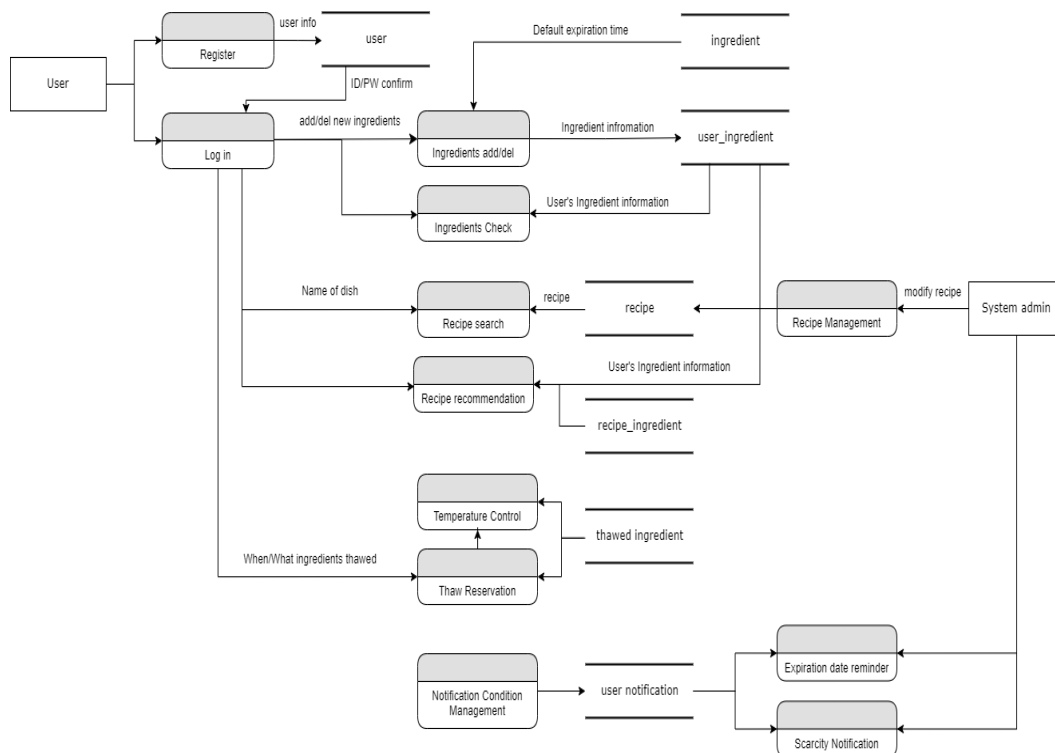
[Table 40] Data Dictionary - user

Field	Key	Constraint	Type	Description
user_id	PK	NOT NULL	varchar (25)	User ID
user_pw		NOT NULL	varchar (25)	Hashed user password
refrigerator_id		NOT NULL	varchar (25)	Refrigerator ID



[Figure 2] ER Diagram

4.2.2.Data Flow Diagram



[Figure 3] Data Flow Diagram

5. Other Nonfunctional Requirements

5.1. Product requirements

5.1.1. Usability

- The system should have a user-friendly interface to make it easier for non-professional/non-technical users to use the LCD screen of refrigerators and mobile applications.
- User interface elements use icons and words together, and the language is provided with both English and Korean. The usage of technical terms should be minimized, and the image/icon should be simple and easy. Each element should be able to identify its function immediately without an additional manual.
- There should be a button on every screen that goes to the previous screen.

5.1.2. Performance

- The provision of all information, such as the recipe recommendation algorithm and expiration date notification, aims to give the results within 5 seconds. Recognizing new food ingredients and storing their information also aims to complete the process within 5 seconds.

5.1.3. Space

- The storage space occupied by mobile applications should not exceed 500MB. The storage of data uses the cloud service.

5.1.4. Dependability

- Recommendation and notification systems should provide services with minimal errors, and the results should be reliable as expected by the user. It does not allow no response to a user's request. The average number of errors in a system response shall not exceed 5 percent.
- If the server failed to return the result within 5 seconds, the system should request a query again for the provided video, image, text file, etc. If the result is not returned after the request is made again, the system considers the network disconnected and should send a network reconnect requirement to the user.
- Ingredients data recognized by the refrigerator should be stored in a cloud service and synchronized with a mobile application.

- Commands from the mobile application should be immediately delivered to the smart refrigerator to perform functions corresponding to the commands.

5.1.5.Security

- The user should be authenticated before using the system. Unauthorized users should not be allowed to use the service.
- The user's password is unknown to the system operator and is used only at login.
- Consent of access rights is required when scanning the inside of a smart refrigerator using a camera. Images taken by the camera should not be leaked to the outside.
- Information such as the list of ingredients and expiration date shall not be leaked to the outside.

5.2. Organizational Requirements

5.2.1.Environmental

- Recipes provided by the system are borrowed from the 'Recipe Ingredient Information' of EPIS Korea Agency of Education, Promotion and Information Service in Food, Agriculture, Forestry and Fisheries).
- The system is supported in Android 8.0 and later versions. iOS is supported in 13.0 and later versions.
- The system is supported on the embedded system of smart refrigerators having at least 2GB RAM, i5-4590/FX 8350 CPU.
- The system provides the same service on both smart refrigerator and smartphone, so the two devices must be synchronized.

5.2.2.Operational

- The user for each smart refrigerator is identified through the ID. User information is managed in the database.
- Information provided in synchronization between the mobile application and the smart refrigerator should be stored in the cloud service.

5.2.3.Development

- Mobile application development should be done through Flutter and Dart. Android app development should be done for Android 8.0 or higher, and iOS app development should be done for iOS version 13.0 or higher.
- Food ingredient classification and categorization should be done through YOLO.
- The expiration date detection should be done through Tesseract OCR.
- Data storage and management should be done through AWS cloud server management, MVC and MySQL.

5.3. External Requirements

5.3.1.Regulatory

- Recipes provided by the system should be able to block the content if requested by the original author under the Copyright Protection Act.
- When using a camera in a smart refrigerator, consent for camera access rights should be obtained in advance. The user may refuse this, but in that case, there may be restrictions when using the service. In addition, leakage of the image to the outside should be prohibited.

5.3.2.Ethical

- It should be prevented from intentionally harming the health of users of this system.

5.3.3.Accounting

- The system should not require additional payments from users in development. Costs used for system development and operation should be in accordance with the provisions of the contract.

5.3.4.Safety/security

- According to the Personal Information Protection Act, all personal information should be encrypted and stored, and be used only within the scope agreed by the user.

6. Appendix A: Glossary

[Table 41] Table of acronyms and abbreviations

Acronyms& Abbreviations	Explanation
RAM	Random Access Memory
AWS	Amazon Web Services
CPU	Central Processing Unit
MVC	Model View Controller
OS	Operating System
JSON	JavaScript Object Notation
API	Application Programming Interface
UI	User Interface
HTTP	Hypertext Transfer Protocol
FPS	Frame Per Second
URL	Uniform Resource Locator
LCD	Liquid Crystal Display

[Table 42] Table of terms and definitions

Terms	Definitions
User	Someone who uses a system
System administrator	All personnel who took part in the development of a system.
Vendor	A personnel who sells things. It will be the refrigerator for this case.
Recipe	The process of cooking food organized as a set of instructions to follow, including the ingredients needed.
Ingredient	The food or substances in category in accordance to the database ingredient table.
Expiration date	Expiration date of each ingredient in the refrigerator.
Quantity	An amount or number of ingredients.
Storage	Where in the refrigerator a certain ingredient is stored.