



FRIDGE BRIDGE

Software Requirements Specification

Team 2 : [Fridge Bridge]

Date : 2021.10.17

20190078 Dongju Kim

20190501 Juhyun Lee

20190545 Seungjae Lim

20200434 Giyeon Lee

Table of Contents

1. Introduction
2. Overall Description
 - A. Product Perspective
 - B. Product Features
 - C. Operating Environment
 - D. Assumption and Dependencies
3. System Features
 - A. Functional Requirements
 - B. Use Case Diagram & Descriptions
 - C. Sequence Diagrams
4. Preliminary User Manual
5. Non-functional Requirements (Quality Attribute)
6. Acknowledgement

1. Introduction

This Software Requirements Specification document deals with a comprehensive assistance system for refrigerator management, in association with a smart refrigerator equipped with several functions required for ingredients maintenance.

Since 2013, the penetration rate of refrigerators in Korea has exceeded one per household. The biggest reason for the usage of refrigerators is to maintain the freshness of the products stored inside, especially food ingredients. The optimal environment and preservation period required by foods in order to maintain freshness varies depending on the categories, such as vegetables, meat, and fish. Due to the difficulties in maintaining the optimal environment per food ingredient, 16% of food stored in the refrigerator is forgotten and disposed of. Moreover, 6% of food waste generated in households is due to expiration, which implies the necessity of a management system for expiration dates of food ingredients.

In order to solve the problems stated above, we propose a smart assistance system for managing the refrigerator ‘Fridge Bridge’. Fridge Bridge can recommend individual locations of new ingredients, control customized temperature per location, provide a list of stored ingredients and expiration date information, and recommend food ingredient purchases when they are near expiration, which makes it the optimal solution for the refrigerator management. The expected users of the system can be divided into two categories:

1. **Use-by date limited user:** Users who are able to make use of ingredients past expiration date(e.g., Ordinary households).
2. **Sell-by date limited user:** Users who are unable to use food ingredients past expiration date(e.g., Restaurant manager, Ingredients mart manager).

There are a total of 7 use cases: Create New Account, Manage Ingredients, Match Date and Temperature, Manage Cell, Track Expiration Dates, Monitor Real-Time, Send Message. The main goal of the system is to provide an efficient solution for managing the refrigerator and the food ingredients with the use of smart refrigerators. Differentiated information such as use-by, sell-by date information, existing ingredients list, recommended cooking information, etc. are provided according to the two expected user environments, so that users can more efficiently manage ingredients in the refrigerator. In addition, by recommending the appropriate refrigerator cell location and temperature for new ingredients to help maintain the freshness of the stored ingredients, the user can easily manage the ingredients in the complex refrigerator.

2. Overall Description

A. Product Perspective

Our Fridge Bridge is a self-contained product. A refrigerator assistance system in collaboration with an optimized smart refrigerator that can help and manage food ingredients and the refrigerator itself.

B. Product Features

- An ingredient Management system helps efficient consumption of ingredients: The system recommends recipes that can handle the ingredients in the refrigerator by including ingredients that are nearing their expiration date first.
- Automatically update data of ingredients to scan receipts and automatically create a list of newly added ingredients by recognizing characters on receipts.
- Refrigerator Monitoring system facilitates maintenance: system displays the status of the refrigerator to the user and notifies the user if some signals are considered abnormal. When the system gets an abnormal signal, connect to the manufacturer service centre at the user's request. Furthermore, the system keeps track of the period for cleaning the refrigerator and notify the user when the scheduled cleaning date is near.
- The system manages ingredients at an optimal temperature, maximizing the flavour and freshness of the ingredients. The system sets the temperature of the refrigerator cell by cell if possible, and keeps track of the ingredients stored in each cell and checks if the temperature of the cell matches the optimal temperature of the ingredient. When data is updated, the system recommends the cell storage with the optimal temperature of new ingredients based on the temperature of each cell.

C. Operating Environment

- Available on mobile devices at Android platform with API Level no less than 14 (Android 4.0, ICE_CREAM SANDWICH)
- The device is proposed to have at least 40Mb of free space.
- The status analysis system of the refrigerator by connecting to the temperature, humidity sensors in the refrigerator.
- Database of information of Refrigerator ingredient and recipe (SQL+ database)

D. Assumption and Dependencies

- Since the application is based on an Android environment there is a need for a good performance Android phone.
- It is assumed that the refrigerator will be able to set its temperature in reference to the temperature range set by the user.
- It is assumed that the refrigerator should be sufficiently large to set the temperature of the individual cells differently.
- It is assumed that the refrigerator is connected to the network in order to automatically set the appropriate temperature and expiration date of the input ingredient.
- It is assumed that the thermometer that measures the temperature of the interior of the refrigerator should be sufficiently sensitive.

3. System Features

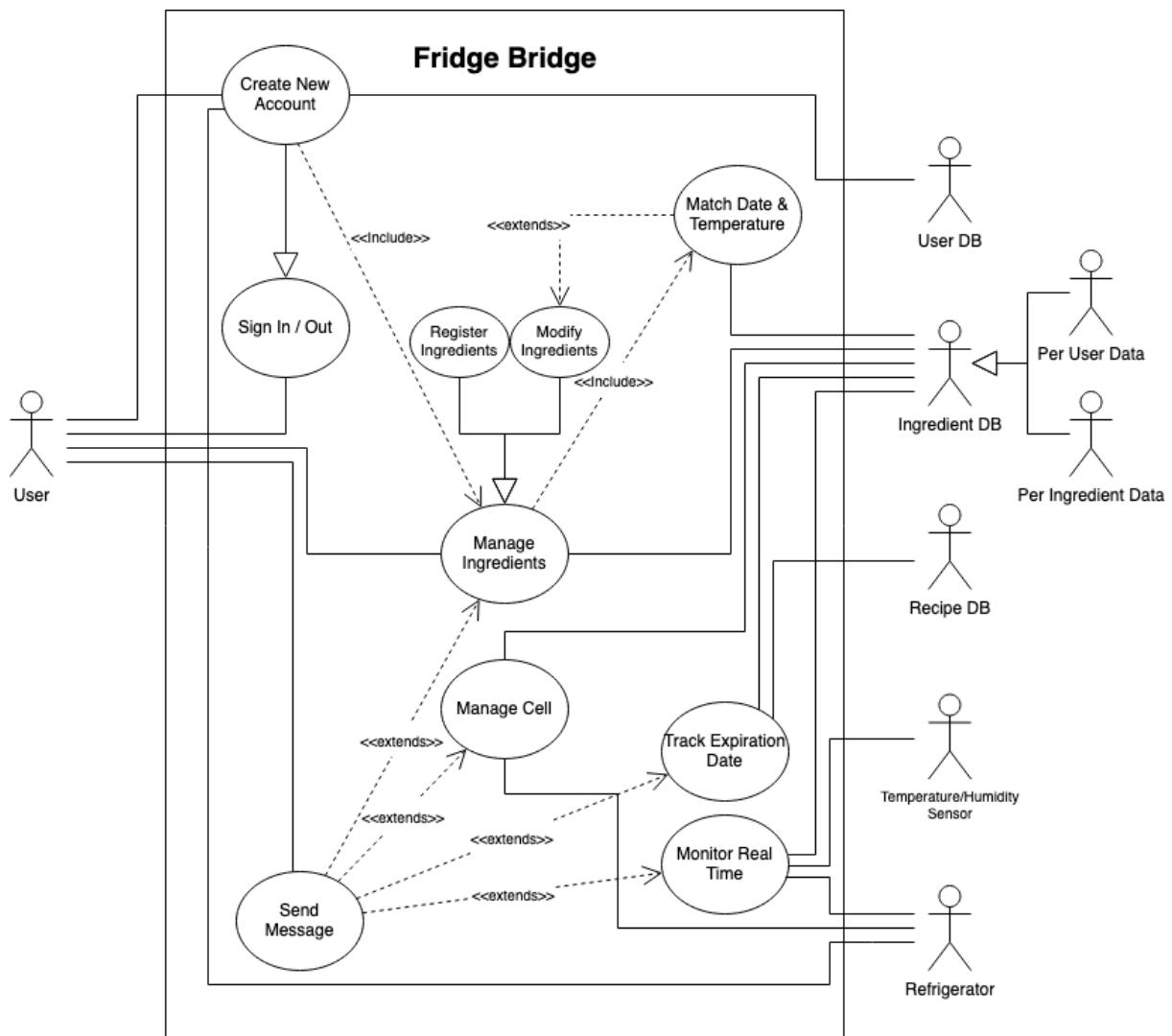
A. Functional Requirements

Hierarchy	Requirement ID	Requirement Description	Actors	Priority
User Management	A_1	<ul style="list-style-type: none">- Enable users to create their own accounts.- Allow access only to one's own refrigerator.- Authenticate the users when creating an account.- Allows users to modify the information in their accounts optionally.- Allows users to sign in and sign out of their accounts.	User Platform User DB	Critical
Ingredients Management	A_2	<ul style="list-style-type: none">- Store and display the ingredients in the user's refrigerator.	User Platform Ingredient DB	Critical

		<ul style="list-style-type: none"> - Enable the users to add, remove, and modify the stored information of the ingredients. - Enables the users to select the date between sell-by date or use-by date 		
Date / Temperature Matching	A_3	<ul style="list-style-type: none"> - Search for the optimal storage temperature and period of the ingredient from ingredient DB. - Enable users to input specific temperatures and dates for each ingredient. - Reject input if user input improper temperature or date. 	Platform Ingredient DB	Critical
Cell Management	B_1	<ul style="list-style-type: none"> - Group the ingredients according to the set temperature and recommend them to the user. - Set the temperature of the refrigerator cell by cell if possible without exceeding the proper temperature of the ingredients. - Keep track of the ingredients stored in each cell and check if the temperature of the cell matches the optimal temperature of the ingredient. - Recommend the cell storage with the optimal temperature of new ingredients based on the temperature of each cell. - Switch information of temperature, ingredients between multiple cells. 	Platform Ingredient DB Refrigerator	Critical
Tracking Expiration Dates	A_4	<ul style="list-style-type: none"> - Keep track of the expiration dates of the stored ingredients, in reference to the type of the 	Platform UserDB RecipeDB	Critical

		<p>ingredient.</p> <ul style="list-style-type: none"> - When the registered deadline is near, notify the user about the expiration. - Recommend recipes that can handle the ingredients in the refrigerator by including ingredients that are nearing their expiration date first. 		
Recipe Data Management	B_2	<ul style="list-style-type: none"> - Allows users to register menus with ingredients and store them in a database. - Allows users to access the database so that they can share the menu. 	Platform UserDB Ingredient DB Recipe DB	Critical
Real-time Monitoring	C_1	<ul style="list-style-type: none"> - Receive the status of the refrigerator by connecting to the (temperature, humidity) sensors in the refrigerator. - Display the status of the refrigerator to the user and notify the user if some signals are considered abnormal. 	Platform Sensors Refrigerator Ingredient DB	Critical

B. Use Case Diagrams & Descriptions



Use case name	Create New Account
Related Requirements	A_1
Goal in Context	Users can check their own ingredients. If the user is not registered to the system, they cannot access the storage status or settings of ingredients in the refrigerator.
Preconditions	Users should be connected to a network.
Successful End Condition	When the user makes a new account, user information is stored in the User DB successfully. If sign-in is attempted, it should successfully be approved only if information matching the user DB exists, and the sign-in information is stored in the user's connected device. When attempting to sign out, the user's sign-in information is successfully removed.

Failed End Condition	The system could not connect to the User DB, the user request and update user info to DB is failed.	
Primary Actors	The user	
Secondary Actors	User DB	
Trigger	The user tried to sign in/out or create a new account	
Main Flow	Step	Action
	1	The user requests to create a new account
	2	The user requests to sign in
	3	The user requests to sign out
Extension 1	Step	Action
	1-1	When entered refrigerator serial number is matched than approve and user information is saved to User DB
	1-2	If entered refrigerator serial number isn't matched than reject and show notice
Extension 2	Step	Action
	2-1	When entered information matches the data in user DB then approve access
	2-2	If entered information does not match the data in user DB than reject and show notice

Use case name	Manage Ingredients
Related Requirements	A_1
Goal in Context	Users can check and modify the data of ingredients inside the refrigerator. The data should be stored to and fetched from the ingredients DB.
Preconditions	Users should be connected to a network.
Successful End Condition	Modified (i.e., added/edited/removed) data is successfully stored in the DB. Data to be checked by the user is successfully fetched from the DB.
Failed End Condition	The system could not connect to the DB, and the user request is rejected.

Primary Actors	The user	
Secondary Actors	Ingredients DB	
Trigger	The user requests to view or modify(add / edit / remove) the ingredients data.	
Main Flow	Step	Action
	1	The user asks to view the lists of ingredients stored on the DB.
	2	The user asks to modify certain ingredients on the list of ingredients.
	3	The user can choose one of the options to add, edit, or remove certain ingredients.
	4	Only if the request is accepted, user can get access to the ingredient DB.
Extension 1	Step	Action
	3.1	The request is invalid (e.g., negative quantity)
	3.2	The request is rejected; notify the user.

Use case name	Match Date and Temperature
Related Requirements	A_3
Goal in Context	Users can match expiration date and proper temperature range for each ingredient automatically.
Preconditions	Users should register the ingredients at least one normally and connect to the internet for getting information from the ingredient database.
Successful End Condition	All ingredients get a proper expiration date and temperature range.
Failed End Condition	If the system could not connect to DB then the matching is rejected. Ingredients that are not contained in ingredients DB then matching is rejected.
Primary Actors	Ingredient DB
Secondary Actors	None

Trigger	The user registers the ingredients	
Main Flow	Step	Action
	1	User registers several ingredients
	2	check whether the ingredient is in per ingredient database
	3	get proper temperature and date of ingredient
	4	change per user database of temperature and date.
Extension 1	Step	Action
	2.1	If the ingredient's name is not in the ingredient database then request the user to input information about temperature and date.

Use case name	Manage Cell	
Related Requirements	B_1	
Goal in Context	Users can set the temperature for each cell	
Preconditions	Users should be connected to a network.	
Successful End Condition	The user's request for temperature setting is transmitted to the refrigerator	
Failed End Condition	The system cannot deliver the user's request to the refrigerator. Therefore the setting value of the refrigerator cannot be changed.	
Primary Actors	The user	
Secondary Actors	Refrigerator	
Trigger	The user change the setting of refrigerator temperature value	
Main Flow	Step	Action
	1	The user ask to change the temperature setting of a cell
	2	If the user's request is rejected, show a notice with problematic ingredients
Extension 1	Step	Action
	1-1	The request is transmitted to the refrigerator only when the changing value is no problem for all the ingredients stored in the cell.

	1-2	The request is rejected If any ingredient has a problematic setting.
--	-----	--

Use case name	Track Expiration Dates	
Related Requirements	A_1	
Goal in Context	Platform tracks the expiration date of ingredients stored in DB; when the expiration date is near, the platform notifies the user with a recipe using the ingredient stored in the recipe DB	
Preconditions	Users should be connected to a network. Ingredient DB should be nonempty, and there should be at least one recipe stored in the recipe DB using the ingredient near expiration.	
Successful End Condition	Notification is successfully sent to the user about the ingredient near expiration, with the related recipes if they exist.	
Failed End Condition	The system could not connect to the DB, and notification is not sent to the user.	
Primary Actors	Platform	
Secondary Actors	Ingredients DB, Recipe DB, User	
Trigger	The platform recognizes certain ingredients in the DB are near expiration.	
Main Flow	Step	Action
	1	The platform checks the ingredient DB for any imminent expiration dates.
	2	The platform recognizes certain ingredients in the DB are near expiration.
	3	The platform checks for recipes in the recipe DB including the ingredients.
	4	Notify the user about the ingredient near expiration with recipes using the ingredient(s).
Extension 1	Step	Action
	3.1	The recipe DB does not contain recipes including such ingredients.
	3.2	Go to step 4 with empty recipe information.

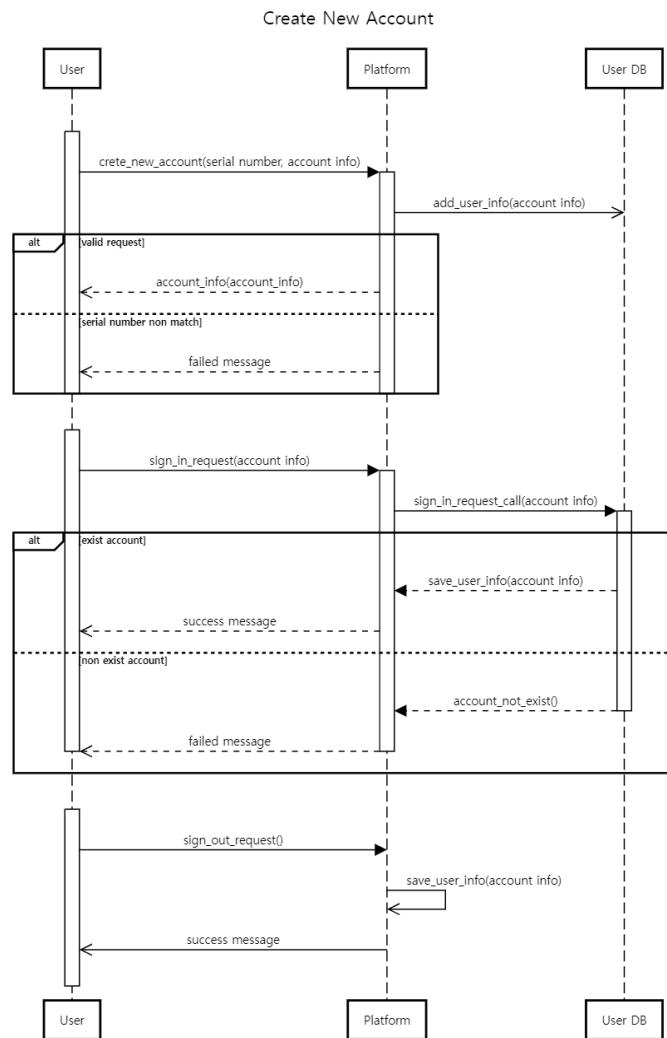
Extension 2	Step	Action
	4.1	The platform cannot connect to the mobile device.
	4.2	Retry step 4 every 1 hour until the expiration date of the ingredient.

Use case name	Monitor Real-Time	
Related Requirements	C_1	
Goal in Context	Display the status of the refrigerator to the user and notify the user if some signals are considered abnormal. (e.g., temperature not being maintained)	
Preconditions	The platform could connect to the sensor and ingredient DB, and ingredient DB has information about the valid state of the cell for each ingredient.	
Successful End Condition	The platform checks the time to clean the refrigerator, detecting the abnormal states of each cell to send a message.	
Failed End Condition	The platform could not connect to the sensor, so it failed to detect the abnormal state.	
Primary Actors	Platform	
Secondary Actors	Ingredient DB, Temperature/Humidity Sensor	
Trigger	Clock of platform	
Main Flow	Step	Action
	1	Platform check clean date to ingredient DB.
	2	Only if the date is over the cleaning cycle, send a message to clean.
	3	Get the cell's state such as temperature and humidity from the sensor.
	4	Compare with Ingredient's appropriate state from ingredient DB.
Extension 1	Step	Action
	4.1	Only if the cell's state is not valid, send an error message and contents to the platform.

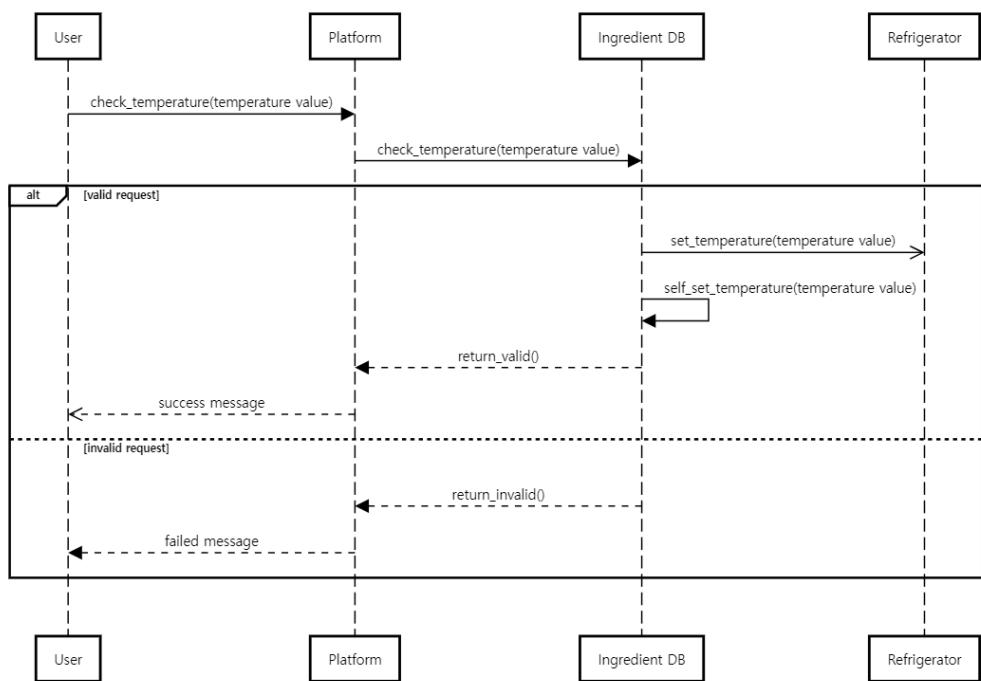
Use case name	Send Message	
Related Requirements	A_2, A_4, B_1, B_2, C_1	
Goal in Context	The platform successfully displays the message and contents such as calls from different use cases.	
Preconditions	The platform receives a send_msg() call and the call is classified as an error with information or contents that have information to display.	
Successful End Condition	The platform successfully displays to the user about contents to be conveyed.	
Failed End Condition	The platform could not connect to the user, so it failed to display information.	
Primary Actors	Platform	
Secondary Actors	User	
Trigger	Platform get send_msg function from different FR.	
Main Flow	Step	Action
	1	The platform receives a send_msg() call.
	2	The platform classifies error code.
	3	Only if the call has an error code, does the platform display error information.
	4	The platform classifies FR by state and content data.
Extension 1	Step	Action
	5.1	Only if the call is from the use case ‘Manage Ingredient’, the platform displays data of the ingredient to the user.
Extension 2	Step	Action
	5.2	Only if the call from the use case ‘Manage Cell’, the platform displays data of the cell’s state to the user.
Extension 3	Step	Action
	5.3	Only if the call from the use case ‘Track Expiration Date’, the platform displays expirable ingredients and their recipe to the user.

Extension 4	Step	Action
	5.4	Only if the call from the use case ‘Monitor real-time’, the platform displays data of the clean or temperature, humidity to the user.

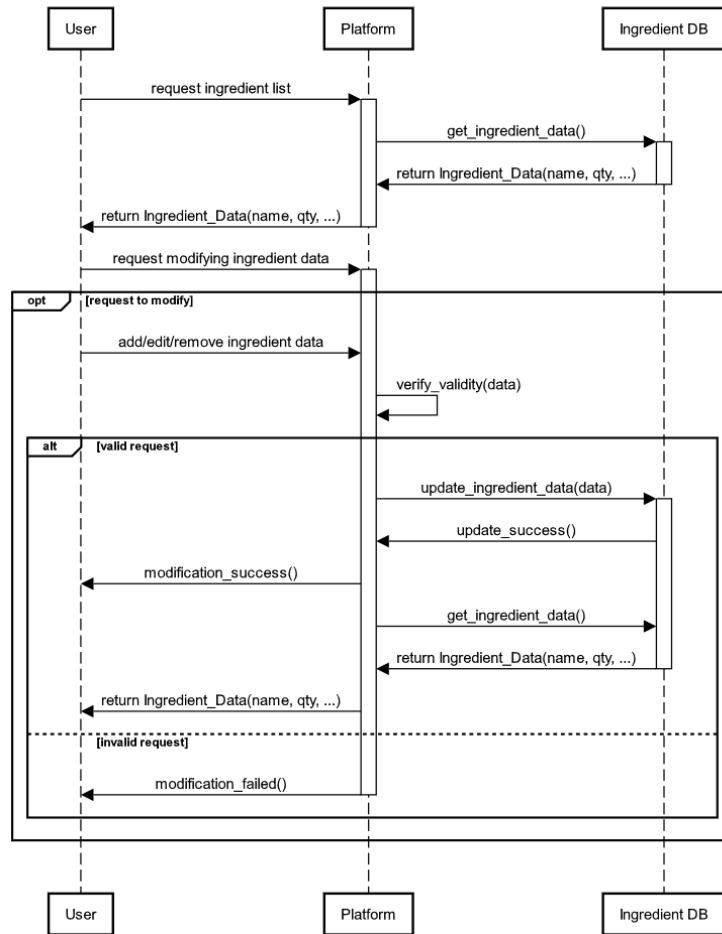
C. Sequence Diagrams

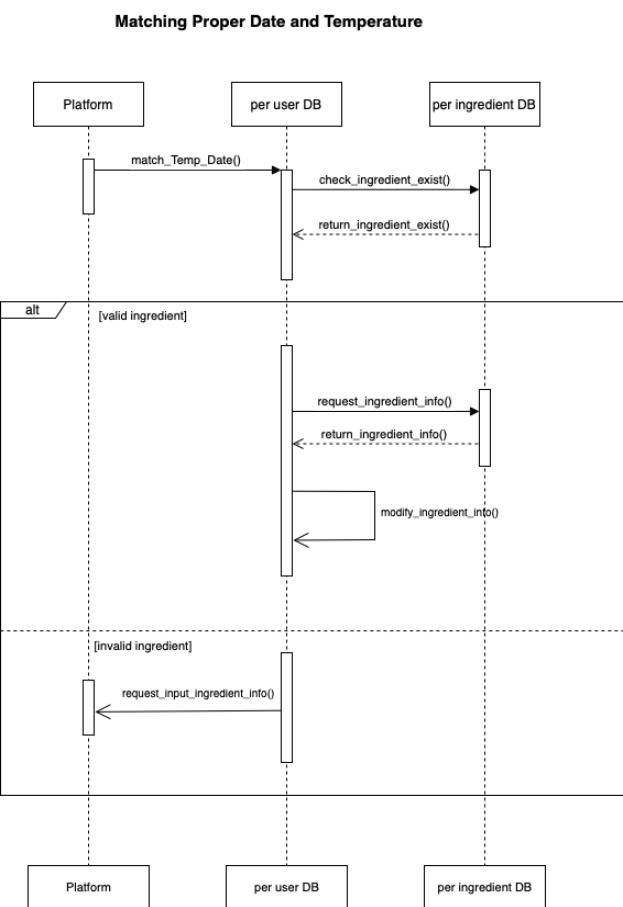
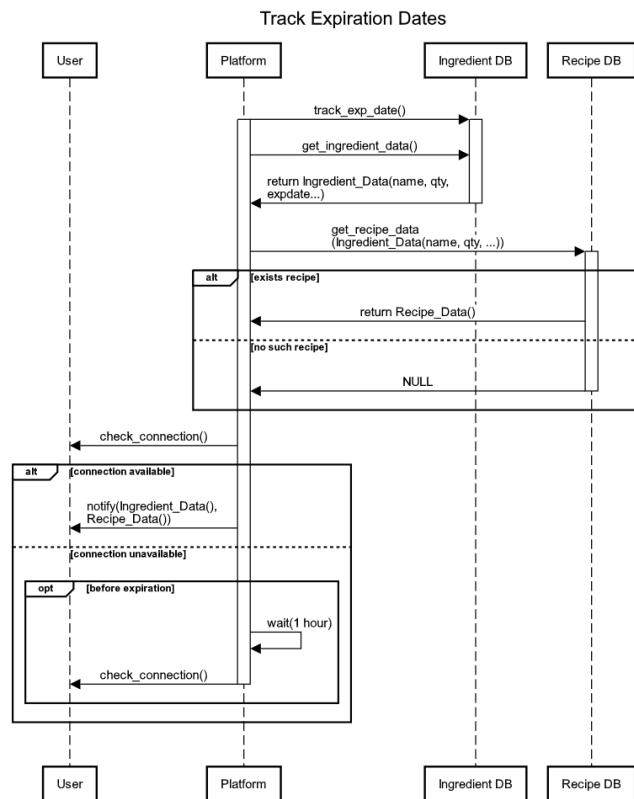


Manage Cell

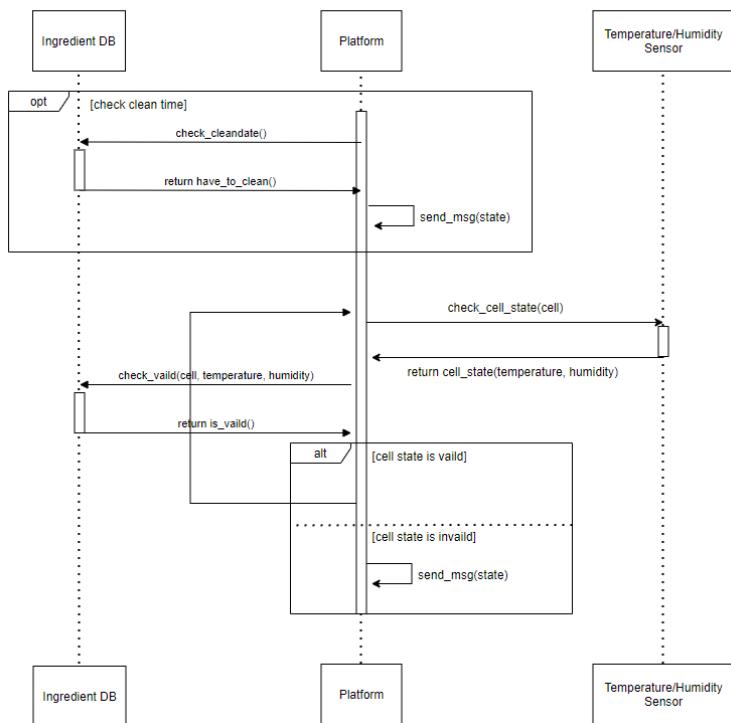


Managing Ingredients

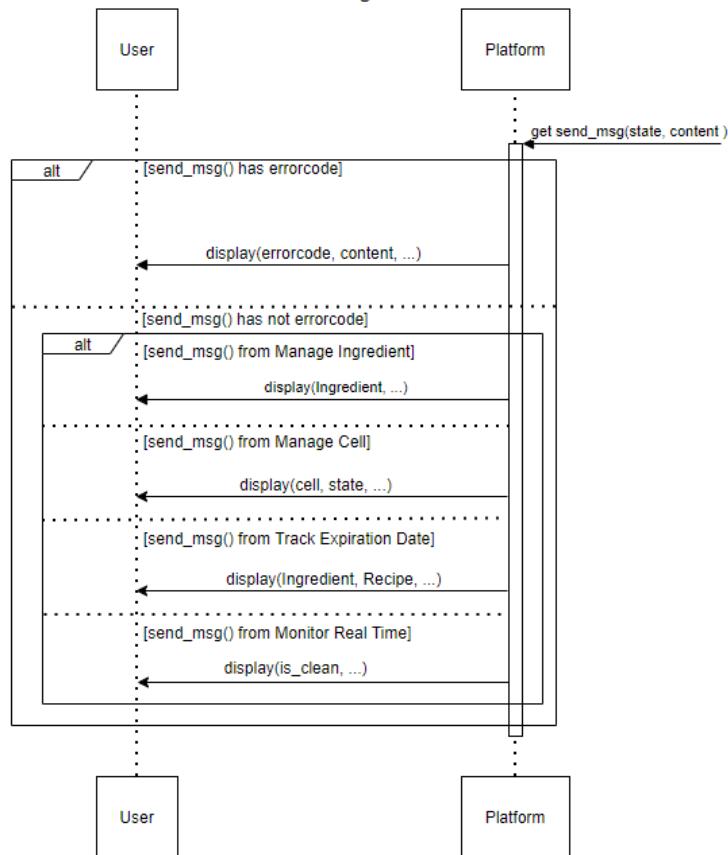




Moniter Real Time



Send Message



4. Preliminary User Manual

A. Target User

- **Use-by date limited user:** Users who are able to make use of ingredients past expiration date(e.g., Ordinary households). When buying food ingredients, they inadvertently buy the ingredients that already exist, or they forget the necessary ingredients and do not buy them. In addition, there are cases where some food ingredients are forgotten until the expiration date is passed.
- **Sell-by date limited user:** Users who are unable to use food ingredients past expiration date(e.g., Restaurant manager, Ingredients mart manager). They have no choice but to sell products that are nearing their expiration date at a discount in order to remove them from the refrigerator quickly. If not, ingredients that are no longer usable after the expiration date cannot be sold and should be disposed of.

B. UI Design Direction

- The design should signify all the ingredients stored by a group, and be able to identify the ingredients stored by individual users.
- The ingredients owned by each individual can be easily identified via text descriptions and photographs(optional).
- The design should include factors that can intuitively convey whether the ingredients each individual has, are properly stored.
- The information should be displayed in the form of an intuitive hexagonal graph that signifies whether the proper temperature is kept and shows the remaining days until the expiration date.

C. Expected UI

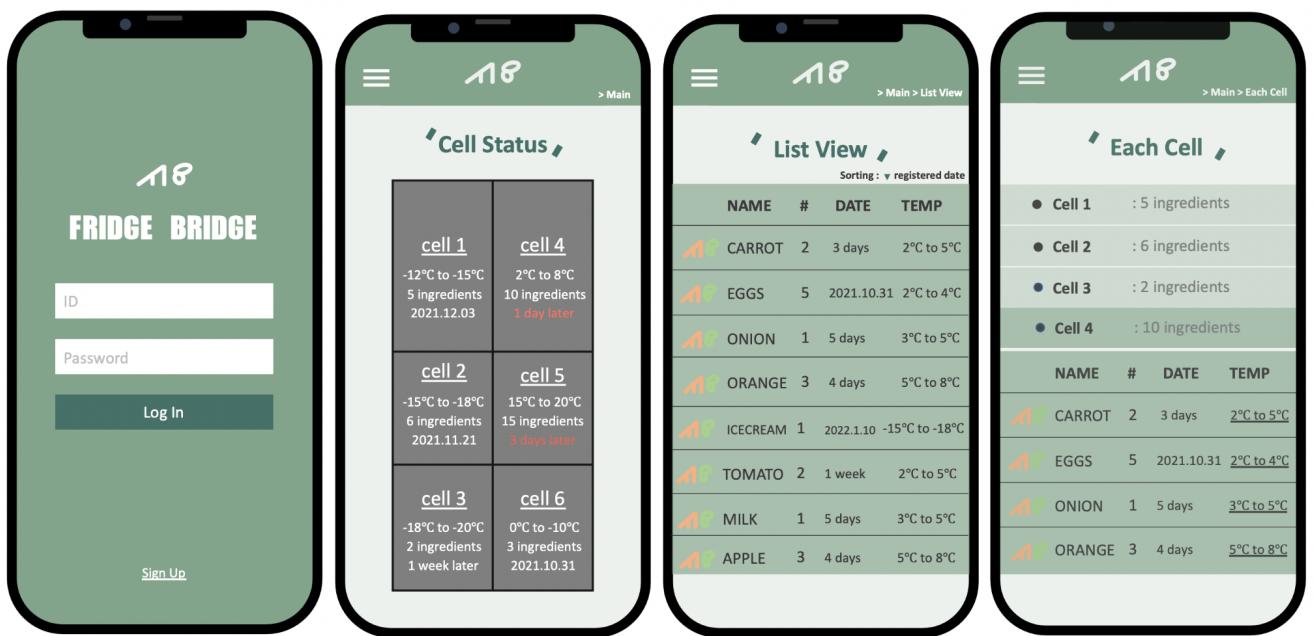
1. Expected menu

- authentication
- main
- main > list_view
- main > each_cell
- main > each_cell > set_temperature

2. GUI

- [Authentication] : Only registered users can access the relevant refrigerator information

- [Main] : Shows the storage status of the entire refrigerator, each cell displays the owner's items, shows the set temperature for each cell, and displays an exclamation mark on a cell with a problem.
- [Main > List_view] : Displays a list of all ingredients. Each ingredient row displays the owner, temperature status, located cell number, and expiration date. They are listed in order of use or sell by date.
- [Main > each_cell] : List of ingredients stored in the cell, owner display, list sorted by use or sell-by date, temperature status display, cell temperature setting button, can only see the ingredients of a specific person.
- [Main > Each_cell > Set_temperature] : The page for setting the temperature of the cell, the appropriate temperature is displayed, and whether there are any problematic issues.



D. Happy Case Scenario

The happy case scenario we think is simple. There are many examples

- When the user goes to buy food, he checks the ingredients stored in the refrigerator and selects and purchases only the necessary items.
- Before arriving at home in hot summer, the temperature of the cell where drinks are collected is lowered so that the user can enjoy a colder drink.

- With the help of the system, members such as single-person households, where management of the shelf life of ingredients is not particularly easy, can minimize food waste.
- Supermarket managers can check expired items and easily arrange them in FIFO order.

5. Non-functional Requirements (Quality Attribute)

A. User Interface and Human Factors

- The main user is every household that has a refrigerator, and the system will manage the refrigerator from time to time.
- The user interface of the application should satisfy high readability and be designed aesthetically while being practical so that it can be easily used by anyone and also be attractive to the users.
- Users should learn and be familiar with the connection interface between the application and the refrigerator in order to use the full functionality of the application.

B. Documentation

- The general end-user is every household that has a refrigerator.
- Manuals for the general user should include how to register, login, scan receipts to automatically create a list of updated ingredients, manually input expiration dates and photos for chosen ingredients, check for a recommended recipe, check the status of the refrigerator, and check for the optimal temperature cell of ingredient for easy to organize.
- Manuals should include all basic use cases, and some measures were prepared for exceptional circumstances.

C. Hardware Considerations

- System should be available on mobile devices on the Android platform with an API Level of no less than 14(Android 4.0, ICE_CREAM SANDWICH).
- The device is proposed to have at least 40Mb of free space.
- The refrigerator should be sufficiently large to set the temperature of the individual cells differently.

- The thermometer that measures the temperature of the interior of the refrigerator should be sufficiently sensitive.

D. Performance Characteristics

- Devices should be able to respond to the information transfer within 2 seconds.
- Devices and refrigerators are proposed to use the internet connection at a speed of about 250 Kb/s.
- The refrigerator should be able to set its temperature in reference to the temperature range set by the user error within 3 degrees.

E. Error Handling and Extreme Conditions

- The system should check the input for a correct(expected) type and output an error message for improper input.
- The system should notify the user if the internet connection is lost.
- The system should check that information is conveyed accurately.
- The system should record and send logs when either the refrigerator or the device undergoes unexpected errors.

F. System Interfacing

- Input comes from the user's device to the database when scanning receipts to automatically update data of ingredients, and manually input expiration date and photo of chosen ingredient, and usage of ingredient.
- Output goes to the user's device from the database when there are near expiration ingredients to recommend recipes and suggest an optimal temperature cell.
- Output comes when the refrigerator is in an abnormal state or passes the cleaning period.

G. Quality Issues

- The system should be able to provide 100% reliability. It must work properly 24/7 as any failure may result in the corruption of ingredients.
- The maximum acceptable downtime per 24 hours is 1 minute and is preferred to be kept as low as possible.
- In case of failure, the system must restart within 15 seconds.

- The system consists of two parts: the mobile application and the refrigerator. The mobile application is portable and can be used anywhere. The refrigerator is not portable and is fixed.

H. System Modifications

- The UI is most likely to be modified to satisfy the design guidelines of the Google Android application.
- The recipe recommends a system of near expirable ingredients that will allow factors of proficiency such as the difficulty and familiarity of cooking.

I. Physical Environment

- The application is expected to be hosted at Google Play servers.
- It is necessary for the refrigerator to have a thermometer attached per cell.
- It is necessary for the refrigerator to control a separated and independent thermostat for each cell.

J. Security Issues

- The system administrator should have access to any data so that the administrator can take control and modify any unacceptable information.
- The database must be equipped with an adequate level of security system in order to prevent unexpected/malicious access from the external environment.
- Data should be backed up regularly and automatically. Specifically, a backup should be conducted every week by the administrator. Backed up data should be regularly checked for any faults that may affect the system.
- Any kind of physical security is not required at this point.

K. Resources and Management Issues

- The development will be conducted on Android Studio by the members of the team. The development process is planned to be finished within two months.
- The application will be developed in Kotlin; developers should be familiar and be able to fluently develop with Kotlin.
- The tentative intermediate and final deadline for the project is December 8th, 2021 and January 28th, 2022, respectively.

- The expected budget for the project is KRW 120 Million. KRW 80 Million will be invested for hardware development, and KRW 40 Million will be used for software development personnel.
- The development process will be conducted solely by the team members at this point, including system installation and maintenance.

6. Acknowledgement

For efficient editing, we mainly divided the management part, and if there were any questions, we had a meeting to receive feedback from other team members so that we could proceed responsibly for all parts of the project to each member.

Name	Main Management	Presentation	Edited
Dongju Kim	1) Introduction 3-C) Use Case Description tables and Sequence Diagrams for two use cases(Create new account, Cell management) 4-A,B,D) Preliminary User Manual.	1) Introduction 2) Overall Description 4) Preliminary User Manual.	Functional Requirement Flow of use cases
Giyeon Lee	2-A,B,C) Overall Description 3-B,C) Overall Use Case Diagram, Use Case Description tables and Sequence Diagrams for two use cases(Managing ingredients & Tracking expiration dates) 5) Quality Attribute, the second half	3-B,C) Use Case Descriptions & Sequence Diagrams for the first five use cases	Functional Requirements Overall document revision
Juhyeon Lee	2-A) Product Perspective 3-A,B) Overall Functional Requirement and Use Case Diagrams for one use case (Matching temperature and date) 4-C) GUI Design	3-A) System Features, Functional Requirements 3-B,C) Last two use cases	Overall use cases Functional Requirement

Seungjae Lim	<p>2) Overall Description, the second half (Operating environment and Assumption and dependencies)</p> <p>3-B,C) Use Case Description tables and Sequence Diagrams for two use cases (Monitor real-time, Send messages)</p> <p>5) Non-functional Requirements, overall</p>	<p>5) Non-functional Requirement (with the answers for the questions on the document)</p>	<p>Made 3-A) table for Functional Requirement Edited use case diagram and descriptions table</p>
---------------------	--	---	--