FRIDGE BRIDGE

Software Design Description

Team 2 : [Fridge Bridge]

Date : 2021.11.21

20190078 Dongju Kim

20190501 Juhyun Lee

20190545 Seungjae Lim

20200434 Giyeon Lee

Table of Contents

1. System Overview

A. Requirements

- **Functional Requirement**

R.F.ID	Requirement Description	Dependencies/ Assumptions
R.F.1	Users can create their own accounts. : system shall authenticate the users and create the accounts. It allows users to sign in and sign out of their accounts.	Only user who has own fridge-bridge refrigerator can access by authenticating
R.F.2	Adding Ingredient lists with picture of receipt : add the ingredients type and quantity to the user ingredients list by recognizing a specific word in the receipt picture.	
R.F.3	Match temperature and expiration date : system shall enter the corresponding appropriate temperature and expiration date in the ingredient database to the user ingredients list.	Depend on ingredient DB
R.F.4	Request user to input proper temperature and date : system shall request to enter the information when there is no food ingredient information in the ingredient DB.	Depend on R.F.3 result
R.F.5	Modifying user ingredient lists : system shall add, delete, or modify items to the list.	
R.F.6	Reject or warn improper input for ingredient lists : system shall reject or warning message is sent when the user enters a value improper temperature and expiration date. For example, when the expiration date is registered before the current date, there are times when the appropriate temperature of the ice cream is registered at room temperature of 20 degrees (out of the appropriate temperature range matched to the DB).	Depend on R.F.4 result
R.F.7	Grouping ingredients in user list according to proper temperature	Depend on R.F.4 result

	range : system shall group user ingredients between similar temperature ranges to divide cells.	
R.F.8	Set temperature of the refrigerator cell by cell : system shall set the refrigerator temperature according to the average of the properties of the grouped food ingredients.	Depend on R.F.6 result
R.F.9	Switch informations of temperature, ingredients group between multiple cells	
R.F.10	Track expiration dates of stored ingredients	
R.F.11	Notify the user about the expiration when the registered deadline is near.	Depend on R.F.10, R.F.5 result
R.F.12	Recommend recipes when the registered deadline is near. : system shall recommend the receipt to the user for using ingredients when the expiration date approaches.	Depend on R.F.5 result
R.F.13	Track temperature of refrigerator by connected sensor in cell.	
R.F.14	Send a warning message to the user.	Depend on R.F.6, R.F.11, R.F.12 result

- **Non-functional Requirement**

	R.N.ID	Non-functional requirements
User Interface and Human Factor	**R.N.1**	The main user is every household that has a refrigerator, and the system will manage the refrigerator from time to time.
	R.N.2	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.
	R.N.3	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.

Documentation	**R.N.4**	The general end-user is every household that has a refrigerator.
	R.N.5	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.
	R.N.6	Manuals should include all basic use cases, and some measures were prepared for exceptional circumstances.
Hardware Considerations	**R.N.7**	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).
	R.N.8	The device is proposed to have at least 40Mb of free space.
	R.N.9	The refrigerator should be sufficiently large to set the temperature of the individual cells differently.
	R.N.10	The thermometer that measures the temperature of the interior of the refrigerator should be sufficiently sensitive.
Performance Characteristics	**R.N.11**	Devices should be able to respond to the information transfer within 2 seconds.
	R.N.12	Devices and refrigerators are proposed to use the internet connection at a speed of about 250 Kb/s.
	R.N.13	The refrigerator should be able to set its temperature in reference to the temperature range set by the user error within 3 degrees.
Error Handling and Extreme Conditions	**R.N.14**	The system should check the input for a correct(expected) type and output an error message for improper input.
	R.N.15	The system should notify the user if the internet connection is lost.
	R.N.16	The system should check that information is conveyed accurately.
	R.N.17	The system should record and send logs when either the refrigerator or the device undergoes unexpected errors.
System Interfacing	**R.N.18**	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.
	R.N.19	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.
	R.N.20	Output comes when the refrigerator is in an abnormal state or passes the cleaning period.

Quality Issues	**R.N.21**	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.
	R.N.22	The maximum acceptable downtime per 24 hours is 1 minute and is preferred to be kept as low as possible.
	R.N.23	In case of failure, the system must restart within 15 seconds.
	R.N.24	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.
System Modifications	**R.N.25**	The UI is most likely to be modified to satisfy the design guidelines of the Google Android application.
	R.N.26	The recipe recommends a system of near expirable ingredients that will allow factors of proficiency such as the difficulty and familiarity of cooking.
Physical Environment	**R.N.27**	The application is expected to be hosted at Google Play servers.
	R.N.28	It is necessary for the refrigerator to have a thermometer attached per cell.
	R.N.29	It is necessary for the refrigerator to control a separated and independent thermostat for each cell.
Security Issues	**R.N.30**	The system administrator should have access to any data so that the administrator can take control and modify any unacceptable information.
	R.N.31	The database must be equipped with an adequate level of security system in order to prevent unexpected/malicious access from the external environment.
	R.N.32	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.
	R.N.33	Any kind of physical security is not required at this point.
Resources and Management Issues	**R.N.34**	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.
	R.N.35	The application will be developed in Kotlin; developers should be familiar and be able to fluently develop with Kotlin.
	R.N.36	The tentative intermediate and final deadline for the project is December 8th, 2021 and January 28th, 2022, respectively.
	R.N.37	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.

	R.N.38	The development process will be conducted solely by the team members at this point, including system installation and maintenance.

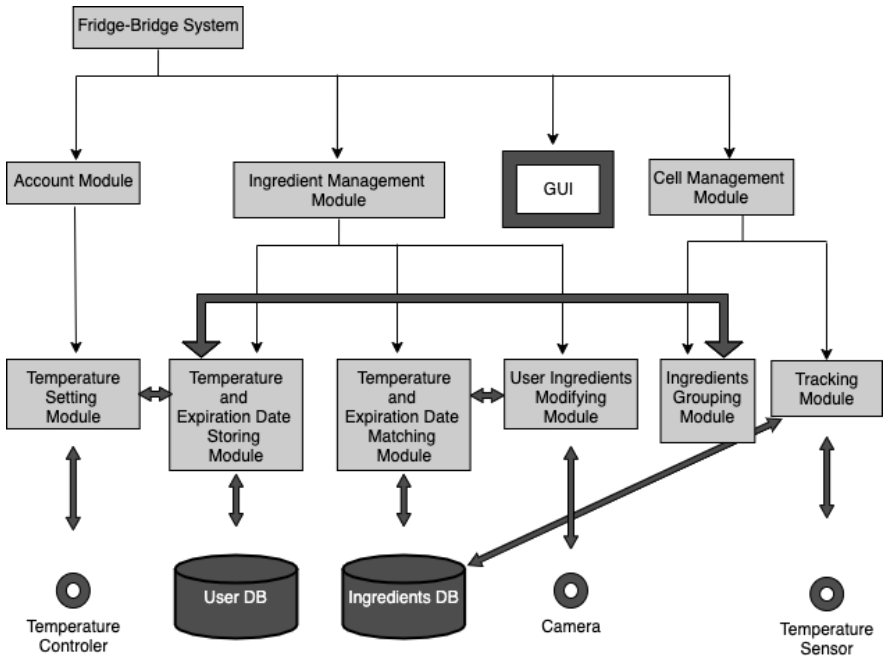
B. Tasks

Task Model		
Task ID	**Task description**	**Related Req(s)**
T.1	Build the development environment.	R.N. : 34, 35, 36, 38
T.2	Developing user interfaces and applications.	R.N. : 2, 3, 7, 8, 9, 10, 11, 24, 25, 30
T.2.1	Building a server that can be linked in real time with a database containing user information, refrigerator settings, and stored ingredients information - *with Firebase*	R.F. : 1, 3, 4, 5, 7, 9 R.N. : 30, 31, 32
T.2.2	Implement new registration and deletion of codes for user authentication.	R.F. : 1
T,2.3	Implement code to list up or edit ingredients stored in the refrigerator.	R.F. : 2, 4, 5, 9 R.N. : 18
T.2.4	Implement code to check and change refrigerator settings.	R.F. : 4, 8, 19
T.2.5	Implement code that sets the appropriate temperature range according to the ingredients and notify a warning message in case of an invalid request.	R.F. : 3, 4, 6, 7, 9, 14 R.N. : 14
T.2.6	Implement code to monitor refrigerator status	R.F. : 10, 13 R.N. : 20
T.2.7	Implement code that periodically tracks the expiration date with the database and notify.	R.F. : 3, 10, 11, 14 R.N. : 19
T.2.8	Implement code that recommends different recipes for each user.	R.F. : 11, 12 R.N. : 19, 26
T.2.9	Make an error handler.	R.F. : 14 R.N. : 15, 16, 17
T.3	Make a vision function to automatically recognize receipts elements.	R.F. : 2 R.N. : 18

T.4	Make a refrigerator whose current status is linked to the server in real time.	R.F. : 8, 13 R.N. : 12, 13, 21, 22, 23, 24, 30
T.5	Make the refrigerator possible to check the temperature information by dividing it for each cell.	R.F. : 8, 13 R.N. : 28
T.5.1	Make the refrigerator possible to set different temperatures for each cell.	R.F. : 8, R.N. : 29
T.6	Create a unique identification number for each refrigerator and link it with user authentication.	R.F. : 1 R.N. : 31
T.7	UI and refrigerator linkage and function check.	R.N. : 21, 22, 23, 24, 25
T.8	Create a user manual.	R.N. : 5, 6
T.9	Sell refrigerator & distributing applications.	R.N. : 27

2. System Design

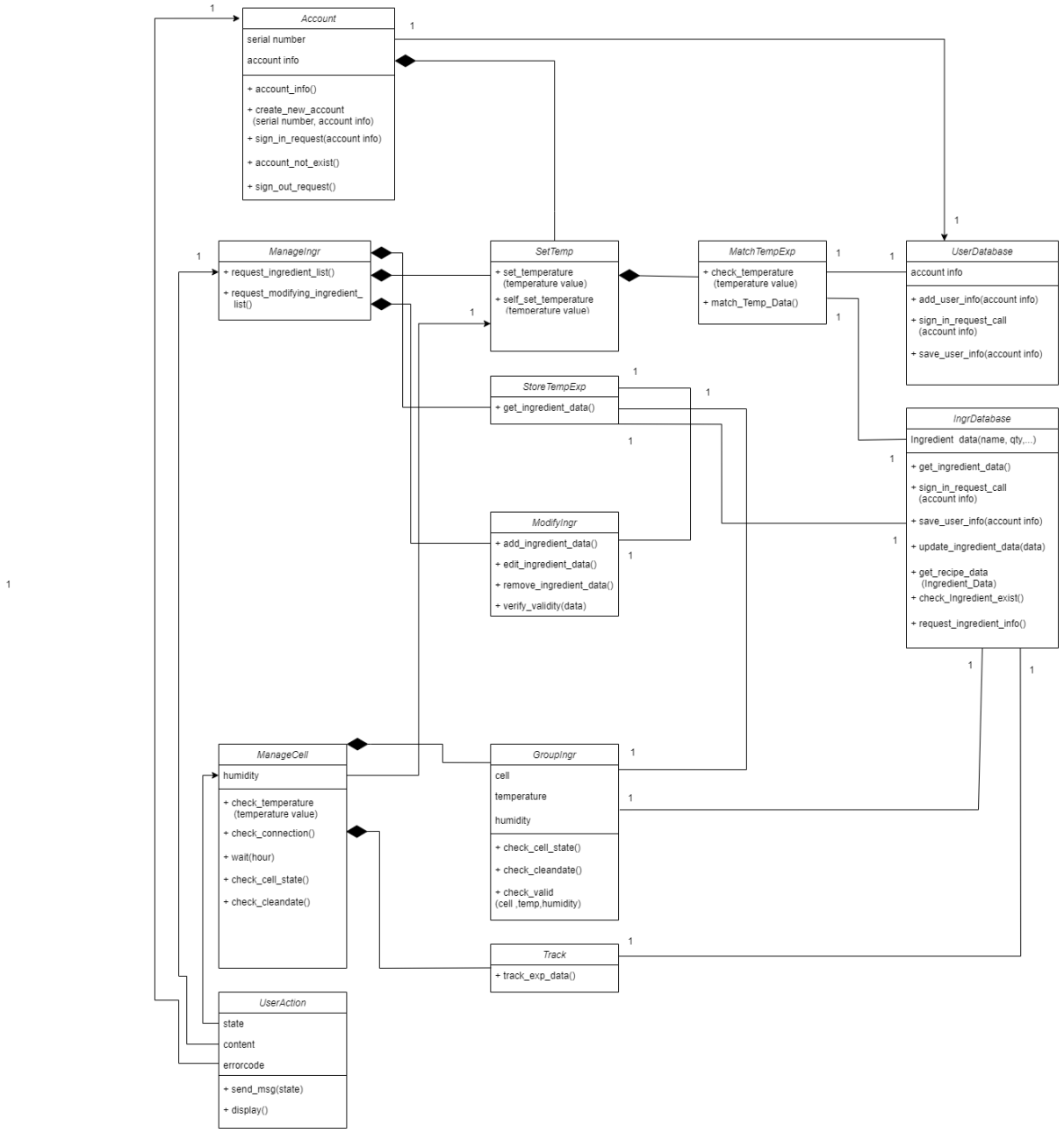
A. System Architecture



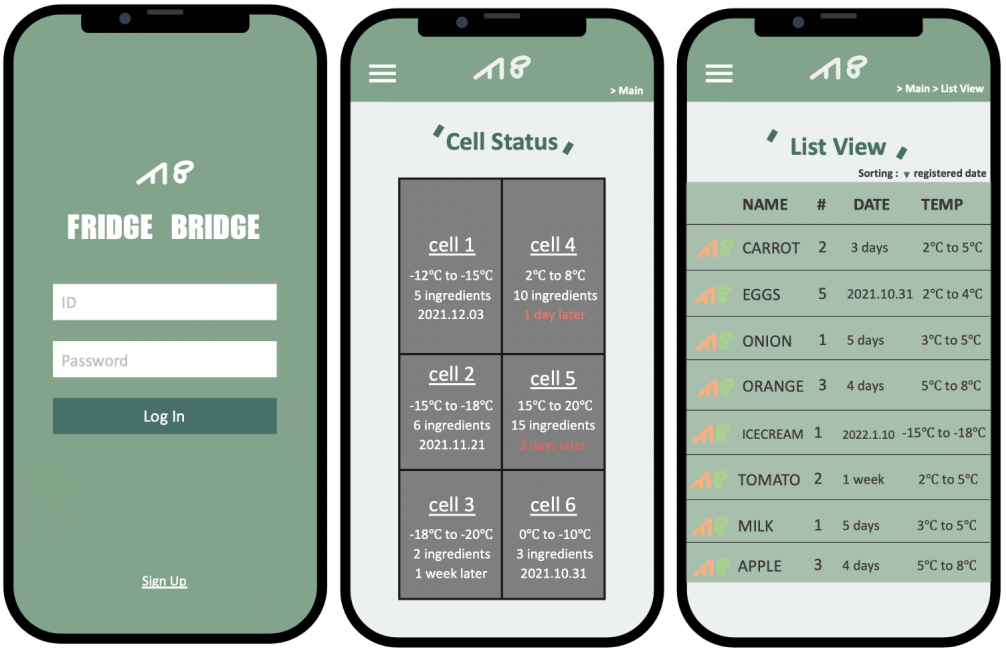
B. Class Diagram

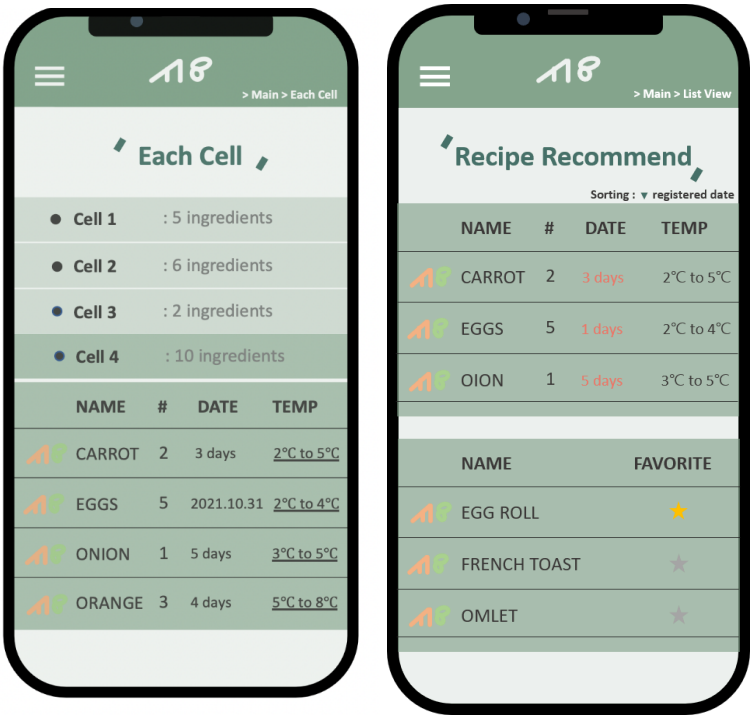
Class Description		
Class Name	**Related Module**	**Class Description**
Account	Account Module	Class for managing and modifying account information
ManageIngr	Ingredient Management Module	Class to proceed ingredients information to store to DB, modify or match expiration date
ManageCell	Cell Management Module	Class to proceed user ingredients data to be grouped and be tracked of
SetTemp	Temperature Setting Module	Class to pass the temperature information to the temperature controller of the refrigerator
StoreTempExp	Temperature and Expiration Date	Class to proceed ingredients data of optimal temperature and expiration date to the User database and pass the information to the ingredients grouping module

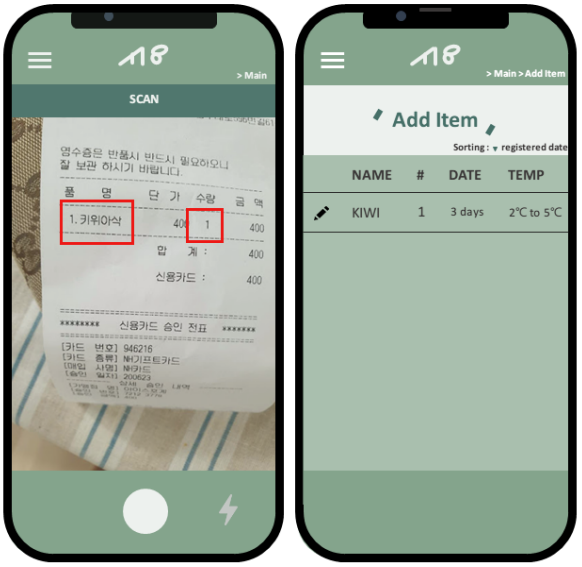
	Storing Module	
MatchTempExp	Temperature and Expiration Date Matching Module	Class to match ingredients data of optimal temperature expiration date with that of data stored in Ingredients database
ModifyIngr	User Ingredients Modifying Module	Class to modify the ingredients data to be stored in the User database
GroupIngr	Ingredients Grouping Module	Class for grouping ingredient informations according to their optimal temperature
Track	Tracking Module	Class to track expiration dates of ingredients stored in database and the temperature between the set value and actual value from sensor
UserAction	GUI	Class to interact between the user and the modules of the system
UserDatabase	User DB	Class for managing storage of the user database
IngrDatabase	Ingredients DB	Class for managing storage of the ingredients database


Account
serial number
account info
+ account_info()
+ create_new_account (serial number, account info)
+ sign_in_request(account info)
+ account_not_exist()
+ sign_out_request()
ManageIngr
+ request_ingredient_list()
+ request_modifying_ingredient_ list()
SetTemp
+ set_temperature (temperature value)
+ self_set_temperature (temperature value)
MatchTempExp
+ check_temperature (temperature value)
+ match_Temp_Data()
UserDatabase
account info
+ add_user_info(account info)
+ sign_in_request_call (account info)
+ save_user_info(account info)
StoreTempExp
+ get_ingredient_data()
IngrDatabase
Ingredient data(name, qty,...)
+ get_ingredient_data()
+ sign_in_request_call (account info)
+ save_user_info(account info)
+ update_ingredient_data(data)
+ get_recipe_data (Ingredient_Data)
+ check_Ingredient_exist()
+ request_ingredient_info()
ModifyIngr
+ add_ingredient_data()
+ edit_ingredient_data()
+ remove_ingredient_data()
+ verify_validity(data)
ManageCell
humidity
+ check_temperature (temperature value)
+ check_connection()
+ wait(hour)
+ check_cell_state()
+ check_cleandate()
GroupIngr
cell
temperature
humidity
+ check_cell_state()
+ check_cleandate()
+ check_valid (cell ,temp,humidity)
Track
+ track_exp_data()
UserAction
state
content
errorcode
+ send_msg(state)
+ display()
1

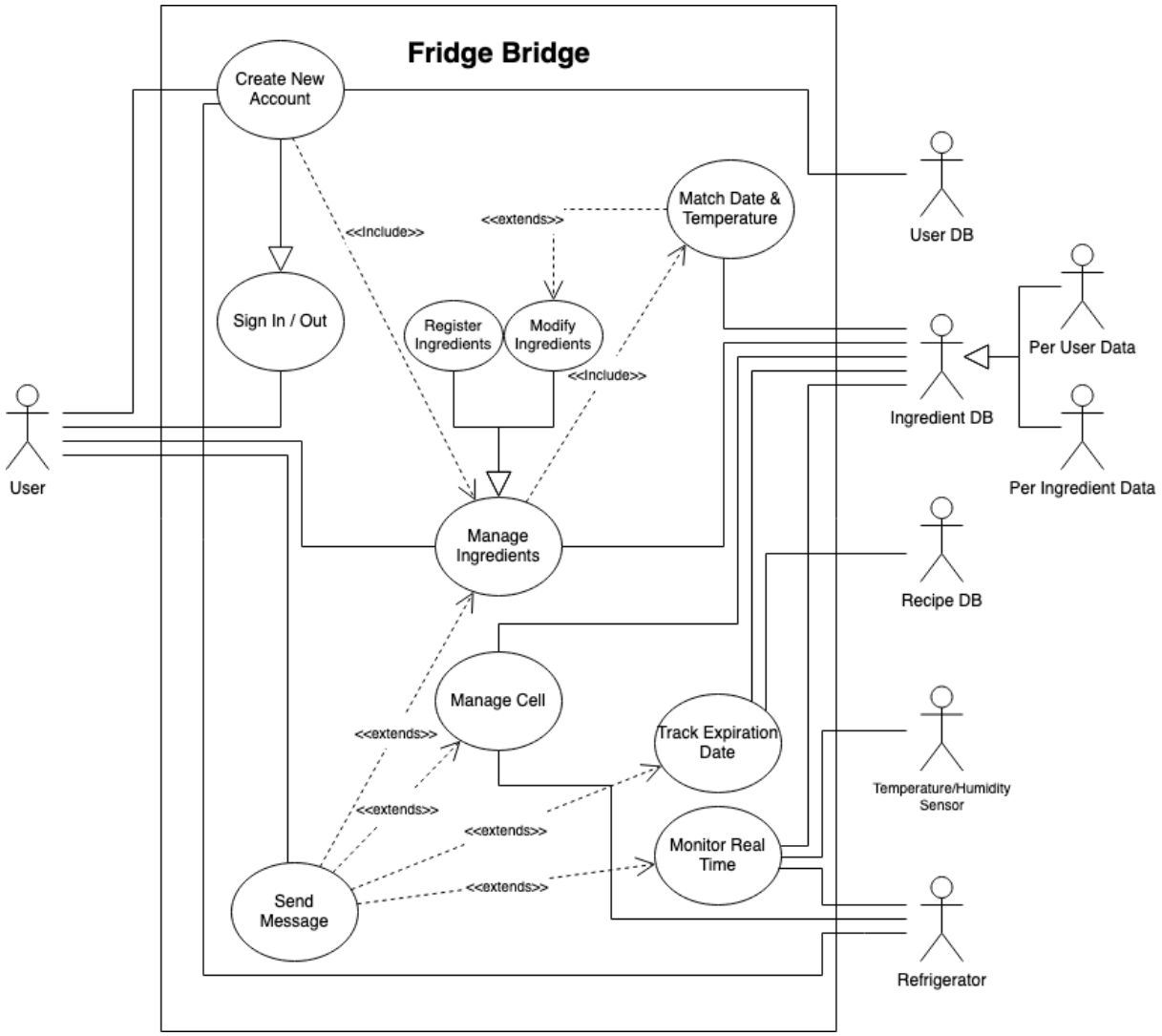
C. User Interface Design (Mockup Design)






> Main
SCAN
영수증은 반품시 반드시 필요하오니
잘 보관 하시기 바랍니다.
품 명
단 가
수량
금 액
1. 키위아삭
400
1
400
합 계 :
400
신용카드 :
400
신용카드 승인 전표
[카드 번호] 946216
[카드 종류] NH기프트카드
> Main > Add Item
Add Item
Sorting : registered date
NAME
#
DATE
TEMP
KIWI
1
3 days
2℃ to 5℃

D. (Refined) User Case Diagram & Description



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.

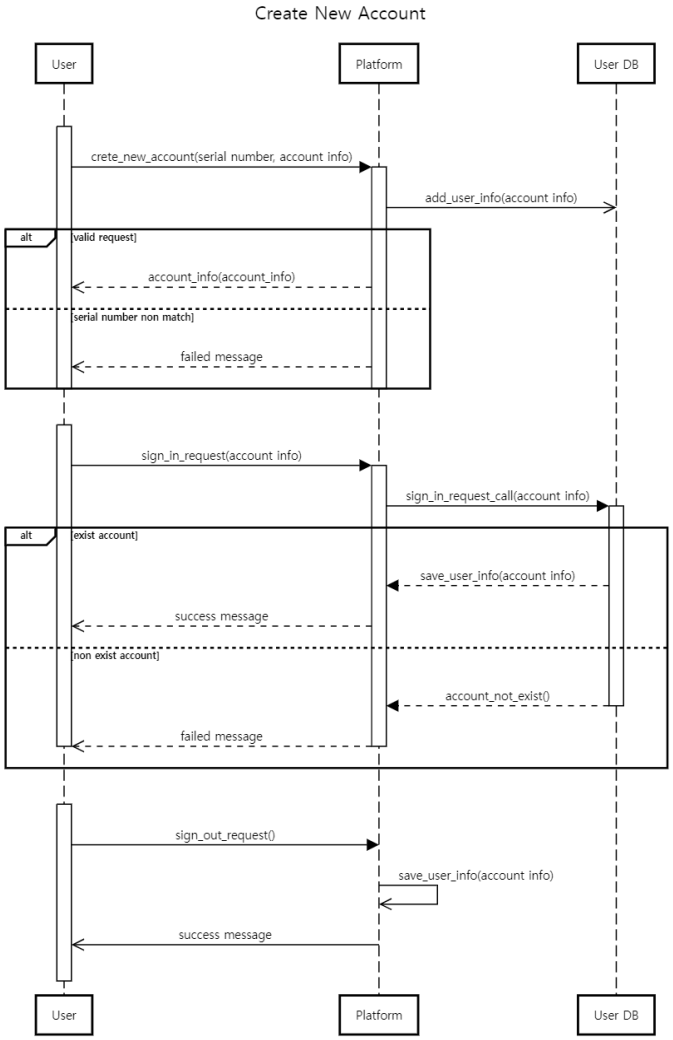
Use case name	**Monitor Real-Time**	
Related Requirements	C_1	
Goal in Context	Display the status of the refrigerator to the user and notify the user if some signals are considered abnormal. (e.g., temperature not being maintained)	
Preconditions	The platform could connect to the sensor and ingredient 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.

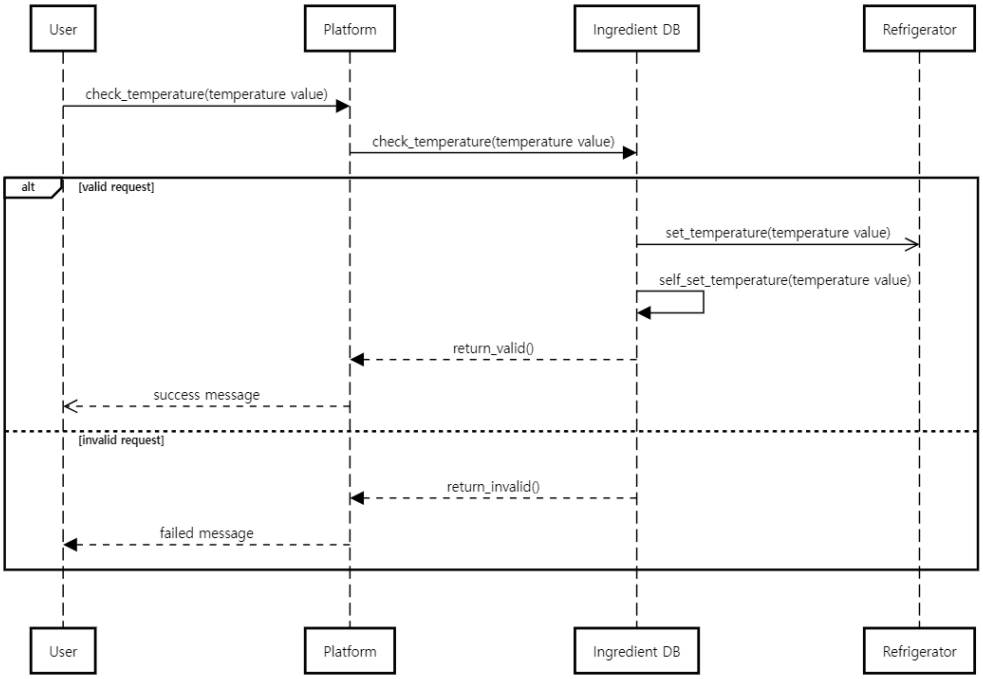
Use case name	**Send Message**	
Related Requirements	A_2, A_4, B_1, B_2, C_1	
Goal in Context	The platform successfully displays the message and contents such as calls from different use cases.	
Preconditions	The platform receives a send_msg() call and the call is classified as an error with information 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.

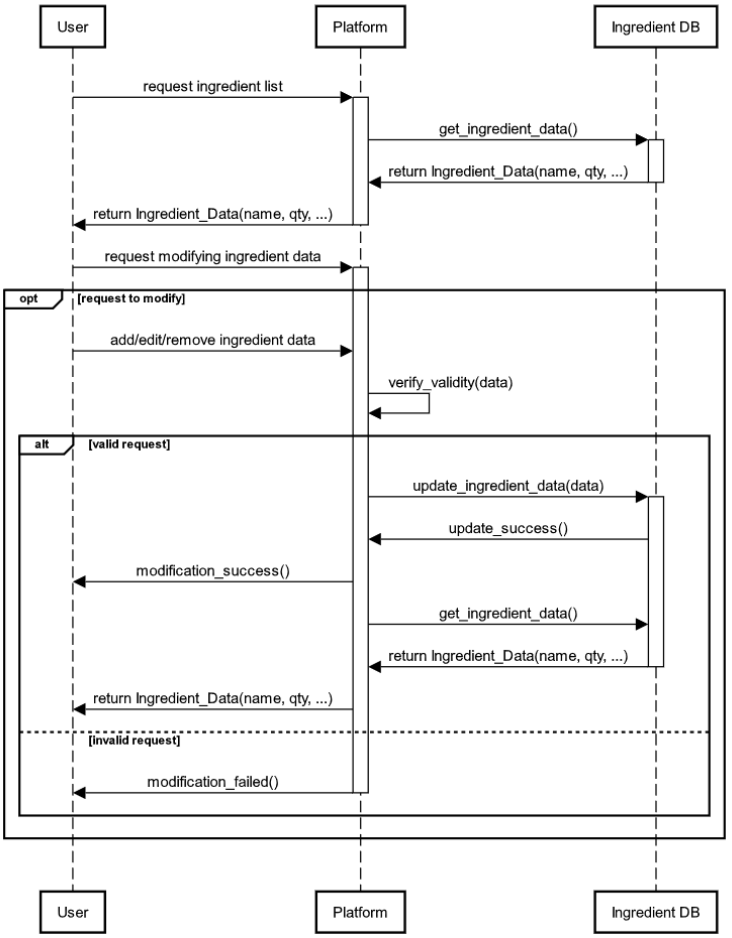
E. (Refined) Sequence Diagram



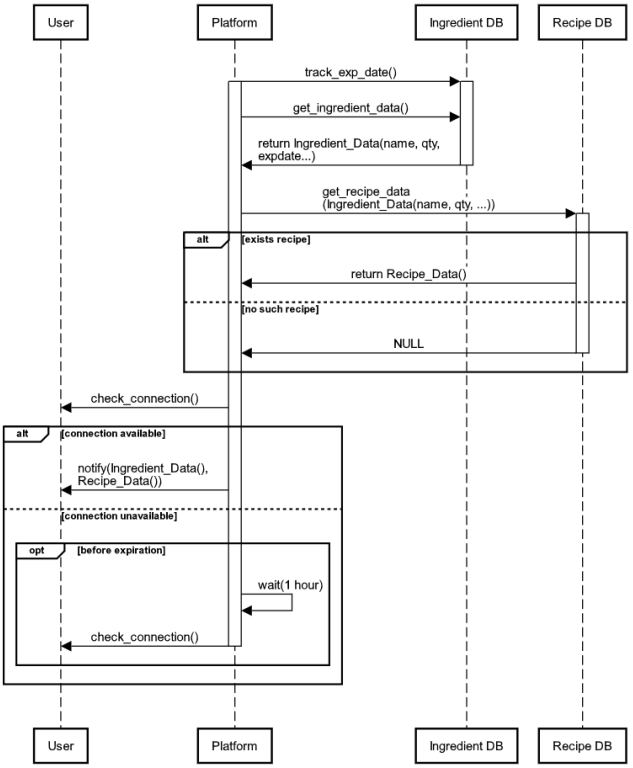
Manage Cell



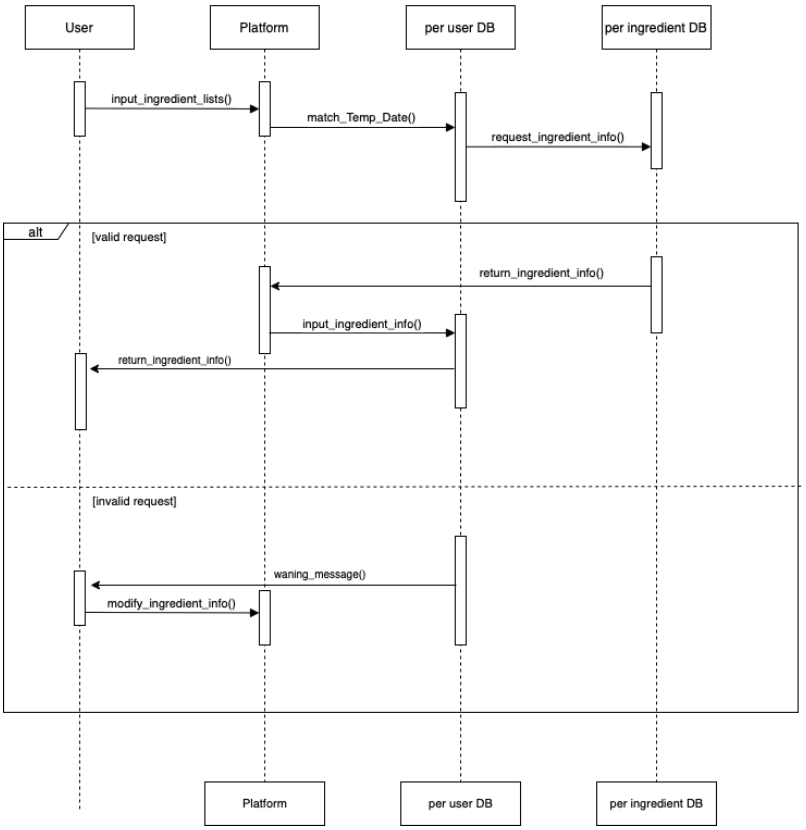
Managing Ingredients



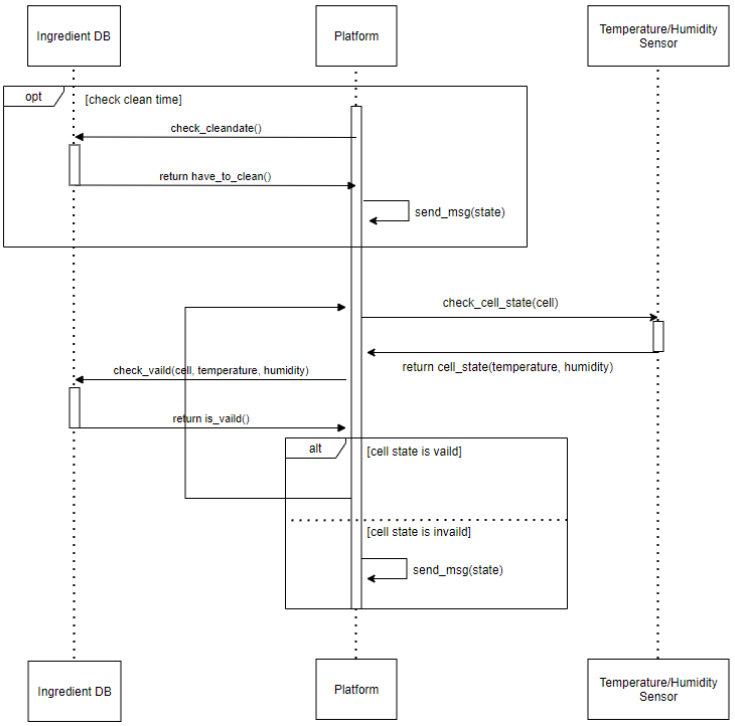
Track Expiration Dates



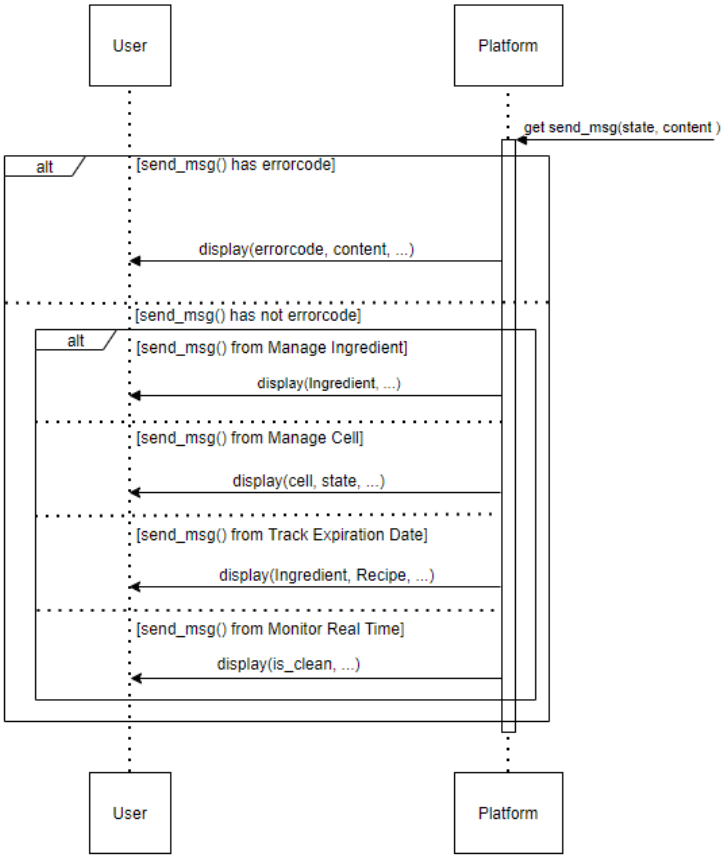
Matching Proper Date and Temperature



Moniter Real Time



Send Message



3. 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
Dongju Kim	1-B) System Overview - Task 3) Acknowledgement
Giyeon Lee	2-B) System Design - Class Diagram 2-C) System Design - User Interface Design
Juhyeon Lee	1-A) System Overview - Requirements 2-A) System Design - System Architecture Edit entire document shape
Seungjae Lim	2-B) System Design - Class Diagram 2-C) System Design - User Interface Design