Requirements elicitation and specification

(Project B)

by
Hyeokjin Kwon

Version Number	Date	Contributor	Description
Version 1	08.09.2021	Hyeokjin Kwon	The First draft of the Requirements elicitation and specification
Version 2	11.09.2021	Hyeokjin Kwon	The Second draft of the requirements elicitation and specification
Version 3 (Final)	14.09.2021	Hyeokjin Kwon	The final version of requirements elicitation and specification

Table of Contents

Background/Summary

PROJECT A

Project A is now closely tied to the Sydney Metro system.
It aims to create a system that can detect potential slip hazards using current surveillance systems (such as security cameras), allocate staff and notify passengers.
I contributed to this project as a client representative.

PROJECT B

Project B is closely related to the university server and several in-app systems(applications).
Project B's system provides users with a combined calendar and access to multiple in-apps such as maps/GPS. The system can also access currently installed surveillance systems (such as security cameras) to keep track of how the busy food store is.
I contributed to this project as a system developer.

Introduction of Project B SRS

DOCUMENT OVERVIEW

I will call the system created through project B as MQExpert.
MQExpert is an interactive and extensive system that interacts with multiple systems and performs various features. Among the many features of MQExpert, the most important functional requirements can be divided into three.

1. MQExpert enables efficient plan management for users through interaction with university server and in-app calendar. (Details are described in the functional requirement)
2. MQExpert enables users to search for routes through interaction with GPS/MAP (details are described in the functional requirement).
3. MQExpert provides queue waiting time to users through interaction with the "How busy are they" system and security camera server (details are described in the functional requirement)

Also, there are many non-functional requirements that must be followed for a successful system. Among them, the non-functional requirements with the highest priority are as follows.

1. Correctness: The system probability of failure on demand (POFOD) should be low when the OneID user/ visitor search the shortest route and estimated arrival time for the destination
2. Development: When developing a system, proper modelling language and computer language should be used
3. Security: The system shall store the OneID user and the Food business accounts with a safe method

RELEASE OVERVIEW

Release	Date Released	Version comments
Version 1	08.09.2021	The first draft of the requirements elicitation and specification
Version 2	11.09.2021	The Second draft of the requirements elicitation and specification (some requirements are modified/deleted/added from V.1)
Version 3	14.09.2021	The final version of requirements elicitation and specification (no other changes can be committed)

ENVIRONMENT

Operating system	iOS and iPadOS 14.7.1.
	Android 11
Database	Oracle 9i, 10g
	MySQL 8.0
Java	1.3.1 and above
React native	0.65 and above

Application Functional Requirements

FR1. NEW USER CREATION AND LOGIN & DEFAULT USER

New Account registration

Requirements when Food businesses create a new account to use the MQExpert system

R #	Req. Name	Requirement details
1.1	Create account	The system shall allow Food business to create a new account

Logging into the System

Requirements when OneID user log in with their OneID

R#	Req. Name	Requirement details
1.2	Login with OneID	The system shall allow OneID users (Staff, Student) to log in with their existing OneID
1.3	Verify Account detail	The system shall verify users' account detail include roles(Staff, Student) through connection with the university server
1.4	Auto Login	The system shall allow automatic login and bypass the login prompt whenever OneID user start the system

Visitor mode

Requirement when a visitor (who does not have OneID)uses the system

R#	Req. Name	Requirement details
1.5	Visitor mode	The system allows the Visitor mode to the user who did not log in with OneID

FR2. USER ACCESS

OneID user access

Requirements about which service OneID users can access

R #	Req. Name	Requirement details
2.1	OneID user access	The OneID user can access the In-app calendar, university timetable, academic schedule, GPS, Food business menu, and real-time queue

Visitor access

Requirements about which service Visitors can access

R#	Req. Name	Requirement details
2.2	Visitor access	The visitor can access the In-app calendar, GPS, Food business menu, and real-time queue

Food service access

Requirements about which service Food services can access

R#	Req. Name	Requirement details
2.3	Food service access	The food service account can access the menu and queue manage

FR3. VIEW TIMETABLE AND SCHEDULE

OneID user timetable and schedule

Timetable and schedule that OneID user can access

R#	Req. Name	Requirement details
3.1	View timetable	The OneID user can view the timetable from the university server
3.2	View university academic schedule	The OneID user can view the university academic schedule from the university server

3.3	View personal plan-OneID	The OneID user can view the personal plan from the in-app calendar

Visitor timetable and schedule

Timetable and schedule that the visitor can access

R#	Req. Name	Requirement details
3.4	View personal plan-visitor	The visitor can view the personal plan from the in-app calendar

FR4. NAVIGATING

Navigating

Requirements about navigating

R#	Req. Name	Requirement details
4.1	Search route and time-OneID	The OneID user can search/view the shortest route and estimated arrival time for the destination
4.2	Search route and time-Visitor	The visitor can search/view the shortest route and estimated arrival time for the destination
4.3	Calculate route and time	The GPS/Map calculates the shortest route and estimated arrival time with given destination from the system shall and provide the result to the system shall

FR5. SCAN QR CODE

View menu-users

Requirements when OneID user/ visitor scan the QR code to view the menu list

R#	Req. Name	Requirement details
5.1	View Menu with scanning	The OneID user and visitor can choose “view menu list” from the system prompt after scanning the QR code from the store
5.2	View Menu-users	The system shall show the menu list when the user clicks “view menu list”

Check-in process

Requirements when OneID user/ visitor scan the QR code to check-in store

R#	Req. Name	Requirement details
5.3	Check-in with scanning	The OneID user and visitor can choose “check-in the store” from the system prompt after scanning the QR code from the store
5.4	Check-in process	The system shall send check-in data to the MyserviceNSW when the user clicks “check-in the store”

FR6. MANAGE MENU-FOOD BUSINESS

View menu-Food business

Requirements when food business views the menu

R#	Req. Name	Requirement details
6.1	View Menu-Food business	The system shall allow the Food business to view their menu list for checking

Delete Menu

Requirements when food business deletes the menu

R#	Req. Name	Requirement details
6.2	Delete Menu	The system shall allow the Food business to delete the menu from their menu list

Create Menu

Requirements when food business creates the menu

R#	Req. Name	Requirement details
6.3	Create Menu	The system shall allow the Food business to create the new menu list

Update Menu

Requirements when food business updates the menu

R#	Req. Name	Requirement details
6.4	Update Menu	The system shall allow the Food business to update the menu from their menu list

FR7. QUEUE MANAGEMENT

View real-time queue

Requirements when different types of users are viewing the real-time queue

R#	Req. Name	Requirement details
7.1	View real-time queue-Food business	The system shall allow the Food business to view a real-time queue for checking
7.2	View real-time queue-OneID user	The system shall allow the OneID user to view a real-time queue for checking to decide which restaurant they should go
7.3	View real-time queue-Visitor	The system shall allow the visitor to view a real-time queue for checking to decide which restaurant they should go

Update real-time queue

Requirements when the real-time queue is updated

R#	Req. Name	Requirement details
7.4	Change the period setting	The Food business can change the period setting which is related to the term of retrieving new real-time queue data
7.5	Request real-time queue update	The Food business requests update of the real-time queue to the system every fixed period
7.6	Get video footage	The system shall get the video footage from the security camera server every fixed period
7.7	Calculate estimated waiting time	The system shall get the calculated estimated waiting time through the “How busy are they” system

7.8	Update real-time queue	The system shall change the real-time queue with calculated estimated waiting time from the “How busy are they” system

FR8. NOTIFICATION

Notification

Requirements when the system sends a notification to the OneID user

R #	Req. Name	Requirement details
8.1	Send notification	The system shall allow the university server to send notifications about the university events and reminders of the class start to the OneID user

APPLICATION QUALITY (NON-functional) requirements

Rank: High, medium, low by the priority

NFR1. Availability

Definition: The amount or percentage of time that the System is available for use by the users. Availability may be negatively impacted by a variety of events including, but not limited to, user error, hardware failure, external system events, unavailability of support personnel, etc.

NFR1.1
Statement of Requirement:
The OneID user can use possible functions of the system for the given amount of time
Fit Metric/Success Criteria:
The OneID user can use “Login with OneID”, “View timetable”, “View university academic time schedule”, “View personal plan”, “Search route and time”, “Scan QR code” for 24 hours everyday. However, The OneID user can not access to the “real-time queue” of the restaurant when the store is closed(non-opening hours or weekend)
Rank: Medium

NFR1.2
Statement of Requirement:
The visitor can use possible functions of the system for the given amount of time
Fit Metric/Success Criteria:
The OneID user can use “View personal plan”, “Search route and time”, “Scan QR code” for 24 hours everyday. However, The visitor can not access to the “real-time queue” of the restaurant when the store is closed(non-opening hours or weekend)
Rank: Medium

NFR1.3
Statement of Requirement:
The Food business can use possible functions of the system for the given amount of time
Fit Metric/Success Criteria:
The Food business can use “Create an account”, “Manage menu” for 24 hours everyday. However, The Food business can not access to the “Manage queue” when the store is closed(non-opening hours or weekend)
Rank: Medium

NFR2. Compatibility

Definition: The ability of the System under discussion to appropriately interact with others systems in its context.

Statement of Requirement:
The system has high compatibility. Function well when other application are running or used with the different operating system.
Fit Metric/Success Criteria:
The system shall allow the concurrent use with the other systems (“University server”, “In-app calendar”, “GPS/MAP”, “MyserviceNSW”, “How busy are they”, “Security camera server”)
The system shall allow the different OS (IOS, Andriod)
Rank: Medium

NFR3. Correctness

Definition: The allowable maximum number or percentage of errors of commission

Statement of Requirement:
The system probability of failure on demand (POFOD) should be low when the OneID user/ visitor search the shortest route and estimated arrival time for the destination
Fit Metric/Success Criteria:
The system probability of failure on demand (POFOD) should be less than 0.00002 (2 out of 100000 hits) when the OneID user/ visitor search the shortest route and estimated arrival time for the destination
Rank: High

NFR4. Installation Complexity

Definition: The combination of direct or indirect costs of the installation of the System

NFR4.1
Statement of Requirement:
The mobile version of the system can be installed with proper environment with the reasonable price
Fit Metric/Success Criteria:
The mobile version of the system can be installed in android and ios devices for free
Rank: Low

NFR4.2
Statement of Requirement:
The installation of the mobile version of the system needs proper amount of device storage
Fit Metric/Success Criteria:
The installation of the mobile version of the system needs 500MB of device storage
Rank: Medium

NFR5. Scalability

Definition: The ability of the System to fulfill its requirements for increasing numbers of users, transactions, etc.

Statement of Requirement:
The system shall allow proper amount of concurrent user(CCU)
Fit Metric/Success Criteria:
The system shall allow at least 5000 concurrent user(CCU)
Rank: Medium

NFR6. Development:

Definition: The environment requirement when developing the system

Statement of Requirement:
When developing a system, proper modelling language and computer language should be used
Fit Metric/Success Criteria:
When developing a system, UML 2.x should be used as the modelling language, and Java, react native should be used as the development language
Rank: High

NFR7. Responsiveness:

NFR7.1
Statement of Requirement:
The system shall respond to view timetable and university academic plan from a OneID user within a reasonable time
Fit Metric/Success Criteria:
The system shall respond to view timetable and university academic plan from a OneID user within 90 seconds. If retrieving data is taking more than 90 seconds, retrieving timetable and university academic plan gets automatically canceled and sends a notification to the OneID user.
Rank: Medium

NFR7.2
Statement of Requirement:
The system shall respond to view personal plan from a OneID user and visitor within a reasonable time
Fit Metric/Success Criteria:
The system shall respond to view personal plans from a OneID user and visitor within 60 seconds. If retrieving data is taking more than 60 seconds, retrieving personal plan gets automatically canceled and sends a notification to the OneID user.
Rank: Medium

NFR7.3
Statement of Requirement:

The system shall respond to search route and time from a OneID user and visitor within a reasonable time
Fit Metric/Success Criteria:
The system shall respond to search route and time from a OneID user and visitor within 60 seconds. If retrieving data is taking more than 60 seconds, retrieving route and time gets automatically canceled and sends a notification to the OneID user
Rank: Medium

NFR7.4
Statement of Requirement:
The system shall respond to scan QR code within a reasonable time
Fit Metric/Success Criteria:
The system shall respond to scan QR codes within 80 seconds. If retrieving data is taking more than 80 seconds, retrieving the menu or check-in is canceled and sends a notification to the user
Rank: Medium

NFR8. Security

Definition: The requirements of the System with respect to access control and/or other context-specific security rules and or regulations.

Statement of Requirement:
The system shall store the OneID user and the Food business accounts with a safe method
Fit Metric/Success Criteria:
The system shall hash the users' account password through SHA-256 and salt it to store in the database
Rank: High

NFR9. Performance

Definition: A measure of user expectations of System response times and storage cost

NFR9.1
Statement of Requirement:
The notification of the event from the university server use a proper amount of memory
Fit Metric/Success Criteria:
The notification of the event from the university server use only memory within 20KB
Rank: Low

NFR9.2
Statement of Requirement:
The system shall send a notification to the OneID user before starting the class
Fit Metric/Success Criteria:
The system shall send a notification to the OneID user 10 minutes before the class start (error range $\pm$ 2 minutes)
Rank: Low

GUI SPECIFICATIONS

The following is a list of unique GUI requirements to include the design, layout and usability.

R#	Req. Name	Requirement details
1	Name	The name of the system(MQExpert) should be on the top of the screen with bold, Hiragino Kaku Gothic Std, 36px.
2	Calendar	The personal calendar section should be under the system name and have one row of dates of the selected month.
3	Map	The map section should be under the calendar section and have a small map inside have the "search route" key under it.
4	Home Bar	The home bar should be placed on the bottom of the screen and should include 4 different buttons(home, search, queue, login)

SECURITY REQUIREMENTS

The following table contains the different kinds of user groups.

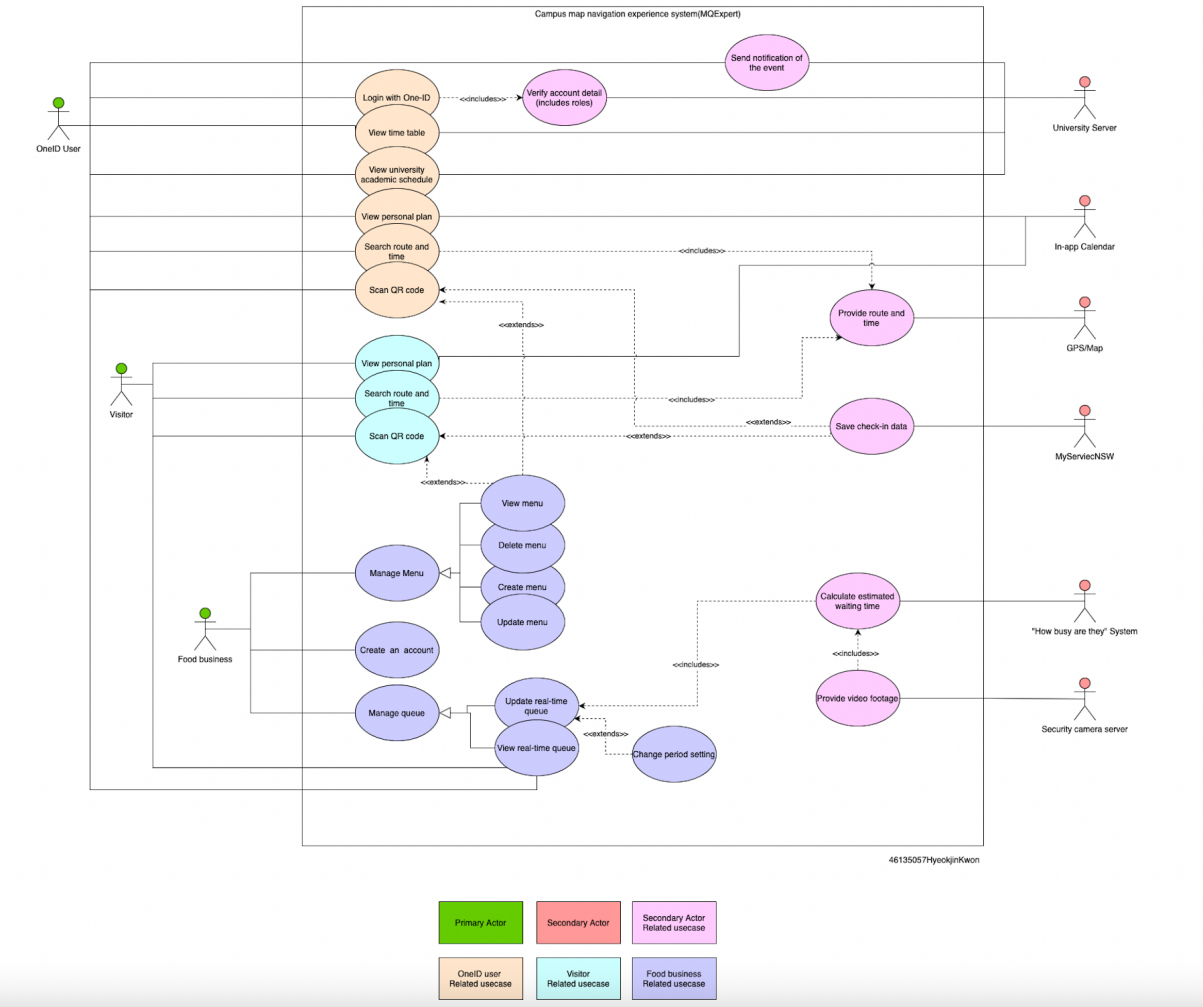
OneID User	➢ Has privileges to access the university server with the OneID and password ➢ Has privileges to access the in-app calendar ➢ Has privileges to access the GPS/Map to search the route for the destination ➢ Has privileges to access the camera to take the QR code ➢ Has privileges to access the food business real-time queue ➢ Has read only privilege to the in-app calendar and university schedule.
Visitor	➢ Is similar to OneID user but has less access to the system. ➢ Has privileges to access the in-app calendar ➢ Has privileges to access the GPS/Map to search the route for the destination ➢ Has privileges to access the camera to take the QR code ➢ Has privileges to access the food business real-time queue ➢ Has read only privilege to the in-app calendar.
Food Business	➢ Anyone managing food businesses on campus ➢ Has limited access to the system ➢ Can create an account ➢ Has privileges to manage the menu list ➢ Has privileges to manage the queue ➢ Has privileges to update real-time period

Requirement Change Management

The following is the change configuration log for this document:

Date	Submitted By	Change type	Change ID	Change Details	Status
27/08/2021	Hyeokjin Kwon	Creation	RCM_1	FR1.5, 2.2, 3.3, 3.4, 4.2, 7.3 were created since a new type of user is added(visitor)	Approved
28/08/2021	Hyeokjin Kwon	Modification	RCM_2	FR 5.1, 5.3 were modified since a new type of user is added(visitor)	Approved
07/09/2021	Hyeokjin Kwon	Deletion	RCM_3	Some non-feasible requirements were deleted after the “feasibility of specification” meeting on 04/09/2021 (e.g. the real-time top-down list by selling amount for food businesses)	Approved
10/09/2021	Hyeokjin Kwon	Creation	RCM_4	NFR5 was created since the scalability need to be specified	Approved
10/09/2021	Hyeokjin Kwon	Creation	RCM_5	FR8 was created since the client representative wanted to add a notification function to the system	Approved
13/09/2021	Hyeokjin Kwon	Modification	RCM_6	NFR5 was modified from 500 concurrent users to 5000 concurrent users considering the university students population	Approved
13/09/2021	Hyeokjin Kwon	Creation	RCM_7	NFR7.4 was created since the response time of the QR scanning was not specified	Approved

Use Case Diagram



Use Case Description

System name: MQExpert

Use Case	View university academic schedule	
Goal	To check university academic schedule	
Preconditions	The OneID user has already completed login and verification	
Success End Condition	MQExpert displays the university academic schedule	
Failed End Condition	University academic schedule is not displayed	
Primary Actors; **Secondary Actors**	OneID user University Server	
Trigger	The OneID user wants access to university academic schedule	
Description / Main Success Scenario	**Step**	**Action**
	1	MQExpert provides the OneID user the option to view university academic schedule
	2	The OneID user selects the 'View university academic schedule' option
	3	The OneID user selects the type of academic schedule they want to retrieve ('Staff''/ 'Student')
	4	MQExpert sends a request to access the university schedule to the university server
	5	The university server checks the identification and role of the OneID user and grant access to the academic schedule
	6	MQExpert displays the university academic schedule to the OneID user
Alternative Flows	**Step**	**Branching Action**
	3.a	The OneID user selects the 'Staff'
	3.a1	MQExpert sends a request to access the university schedule for 'Staff'
	3.b	The OneID user selects the 'Student'

	3.b1	MQExpert sends a request to access the university schedule for ‘Student’
	4.a	The connection between MQExpert and the university server is unstable
	4.a1	MQExpert displays an error message and prompts the OneID user to check the connection
	4.a2	Go back to step1
	5.a	The OneID user’s identification is expired
	5.a1	MQExpert displays an error message and prompts the OneID user to login again to verify the account
	5.b	The OneID user can’t access to the given type of academic schedule (‘Staff’, ‘Student’)
	5.b1	MQExpert displays an error message
	5.b2	Go back to step3
	6.a	The connection between MQExpert and the university server is unstable
	6.a1	MQExpert displays an error message and prompts the OneID user to check the connection
	6.a2	Go back to step1

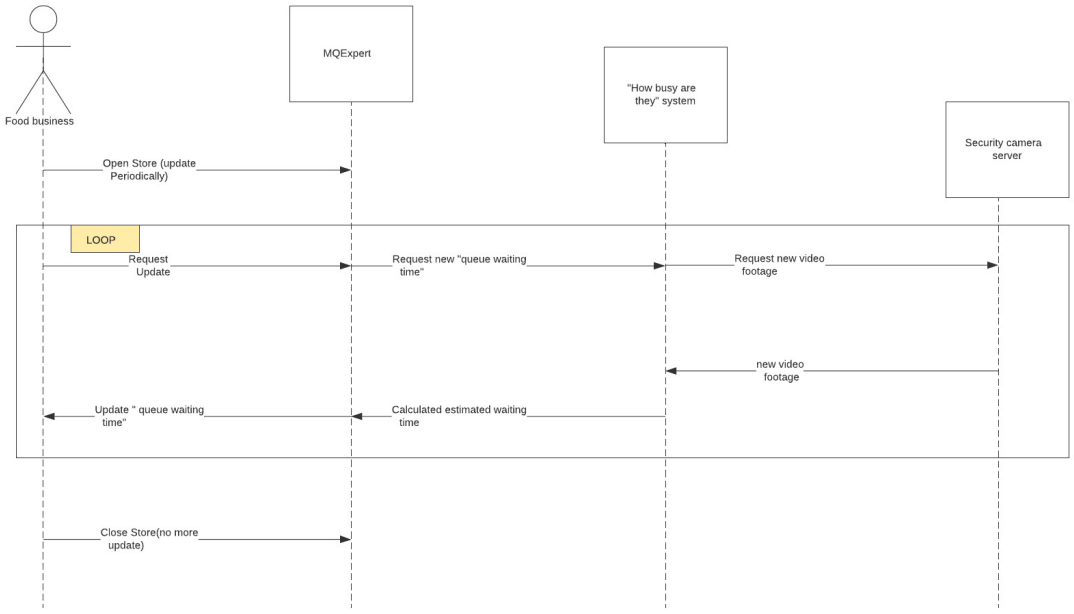
Use Case	Search route and time	
Goal	To get the shortest route and estimated arrival time	
Preconditions	MQExpert has access to user device GPS	
Success End Condition	MQExpert displays the shortest route and estimated arrival time for the given destination	
Failed End Condition	The shortest route and estimated arrival time for the given destination are not displayed	
Primary Actors; **Secondary Actors**	OneID user/ Visitor GPS	
Trigger	The OneID user/ Visitor wants to search the route and time for destination	
Description / Main Success Scenario	**Step**	**Action**
	1	MQExpert prompts the OneID user/ Visitor to enter the destination
	2	The OneID user/ Visitor enters the destination and press ‘directions’
	3	MQExpert transfers given information to the GPS/Map system
	4	The GPS/ Map system calculates and sends the shortest route and estimated arrival time to MQExpert
	5	MQExpert displays the shortest route and estimated arrival time to OneID user/ Visitor
Alternative Flows	**Step**	**Branching Action**
	2.a	The OneID user/ Visitor enters an invalid destination (non-existent location)
	2.a1	MQExpert displays an error message and prompts the user to enter new destination
	2.a2	Go back to step2
	3.a	The connection between MQExpert and the GPS/ Map system is unstable

	3.a1	MQExpert displays an error message and prompts the user to check the connection
	3.a2	Go back to step1

Use Case	Update real-time queue	
Goal	To update real-time queue	
Preconditions	MQExpert has access to Security camera server and “How busy are they” system	
Success End Condition	MQExpert updates real-time queue for Food business	
Failed End Condition	Real-time queue for Food business is not updated	
Primary Actors; **Secondary Actors**	Food business Security camera server, “How busy are they” system	
Trigger	Food business wants to update real-time queue periodically	
Description / Main Success Scenario	**Step**	**Action**
	1	Food business requests update of real-time queue to MQExpert
	2	MQExpert requests a new “queue waiting time” to “How busy are they” system
	3	“How busy are they” system requests new video footage to the security camera server
	4	Security camera server provides video footage of food and retail outlet to “How busy are they” system
	5	“How busy are they” system calculates estimated waiting time using given video footage
	6	“How busy are they” system sends a calculated waiting time to MQExpert
	7	MQExpert updates food business’ queue information
Alternative Flows	**Step**	**Branching Action**

	2.a	The connection between MQExpert and the “How busy are they” system is unstable
	2.a1	MQExpert displays an error message and prompts the user to check the connection
	2.a2	Go back to step1
	5.a	The calculated value from the “How busy are they” system is invalid, e.g.) estimated waiting time: -30 minutes
	5.a1	Check the algorithm of the “How busy are they” system
	6.a	The connection between MQExpert and the “How busy are they” system is unstable
	6.a1	MQExpert displays an error message and prompts the user to check the connection
	6.a2	Go back to step1

Interaction Diagram



Prioritisation explanation

First can select the three most important requirements, reflecting the functional requirements of MQExpert and the opinions of client representatives. The three are "View university academic schedule", "Search route and time", and "Update real-time queue". All three of these are the functions that the client considers most important and these are relatively more complicated than other functions and are closely connected with the secondary actor. "View university academic schedule" is important because it can be executed when OneID is successfully linked to the university server. "Search route and time" is important because it must be able to successfully access the user's phone's GPS/MAP to send and retrieve data.
There are three reasons why "Update real-time queue" is the most important:

1.Other functions are easily found in many applications, but this function was developed exclusively for this application.

2. This function is a very complex function that applies complicated technologies like augmented reality.

3. The client was an expert on the security system, and in the user specification and subsequent meetings, the client representative said this is the function they valued the most.

Requirement elicitations that I used

Interview and survey were used to elicit the requirements.
Since MQExpert is a new system that combines various functions, it is difficult to find a system with exactly the same function. Therefore, after having a client representative use an application with similar features called "Lost Campus(Figure.1)" got feedback on it to find out what to improve and what requirements MQExpert needs.

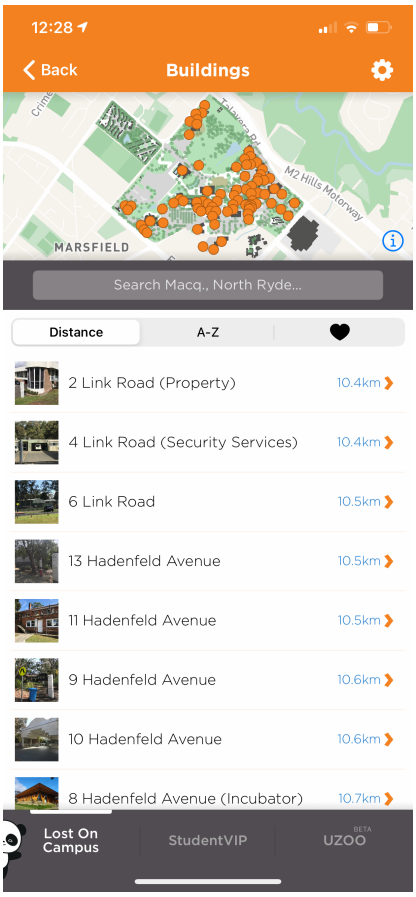


Fig.1) Lost Campus

1. Survey

The survey was used to figure out which functions the client representative considered useful. Functions with high satisfaction by the client representative will be implemented similarly, and functions with low satisfaction will be deleted or be improved.
In Questionnaires, all functions' importance is measured in interval scales. Therefore, the developer can promptly identify the functions that the client considers necessary or relatively insignificant. As shown in Table 1 and Table 2, It is possible to change how the system should be implemented before starting development by identifying clients' needs.

Table 1) Example of the questionnaires used for requirement elicitation

"How efficient is it to find the route for the destination by clicking the site on the map?"				
Not Useful At All			Extremely Useful	
1	2	3	4	5

Table 2) Example of the questionnaires used for requirement elicitation

"How efficient is it to click the Restaurant on the map to see closing hours and additional information?"				
Not Useful At All			Extremely Useful	
1	2	3	4	5

2. Interview

The interview was used to figure out which new functions the client representative considered needed. For a successful interview, I conducted a structured interview with predefined open-ended questions. Furthermore, I asked additional questions when I can't understand what the client representative need.
Table 3-5 is the examples of the questions used for the interview.

Table 3)

"Do you think QR code is convenient? If you think so, what functions could be used with QR code scanning"
Client representative's answer

Table 4)

"How many user types do you want to have in the system?"
Client representative's answer

Table 5)

"What additional features do you want to have in the system?"
Client representative's answer

Evaluation of teammates

What I liked the most about the client representative was that she kept asking for creative features that I cannot think of. This means she has done a lot of researches and thinking. For example, one of the requirements that she wanted was really creative and cannot be found in the assignment specification. That was the notification from the university server to the system to let the user know they have a class within 10 mins. Furthermore, she always tried to get her work done on time and was a really good communicator. Since I am not born in Australia when I have some problems in understanding syntaxes or her intention she always tries to rephrase and take it slow again for better communication and teamwork.

However, her worst part was that she is too creative sometimes. Some of her requirements were unrealistic, hard to implement, too complicate, off-topic, or costly, which are tough jobs for the development team. For example, she wanted the system to have a feature that sorts each restaurant's order as the real-time top-down list by selling amount. However, I(developer) thought that is not the essential function for customers and the system. Furthermore, that feature is tough to implement and unrealistic since the food business needs to communicate with the order database in real-time. The problem is that the system does not contain an order system and database to track and store the order data.

A plan for further requirements analysis

While continuing developing, the JAD method and prototyping can be used for further requirement elicitation.
The JAD involves clients in the system development stage. JAD does not interview clients one by one but gathers clients together and holds meetings similar to brainstorming to identify new feature ideas and existing problems. Through the JAD process, developers can analyze whether the developing process is on the right track **[1]**. JAD can shorten development time and increase end-user satisfaction.
A prototype is a new type of machine or device that has not yet been built and sold **[1]**. While eliciting requirements through the JAD, the client can have a better understanding of the system by reviewing the prototype.
For example, when iterating prototype development/modification/supplementation, multiple JAD conferences can be held to identify problems with the prototype and add new features.

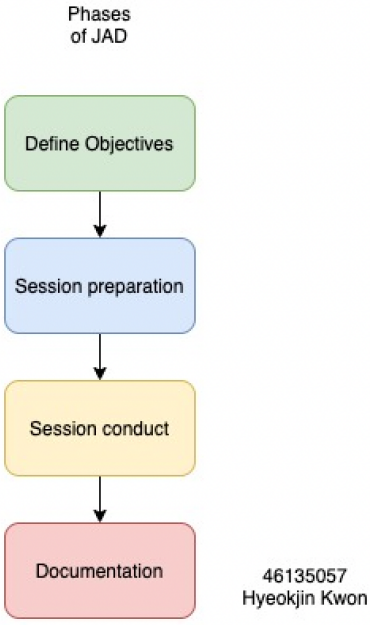


Fig.2) Phases of JAD Source) Adopted from **[2]**

As shown in Figure 2, the developer or JAD manager can conduct the session(discussion/ Brainstorming) when they have the new prototype to test. Depending on their purpose and function, there are different types of prototypes used as references in JAD. Based on completeness, prototypes can be divided into low-fidelity and high-fidelity [3].
Low-fidelity refers to a paper model that shows low-quality and rough designs and functions, and high-fidelity refers to a high-quality prototype that can show links between pages when clicked the button.
Figure 3 depicts the example of the UI prototype of the MQExpert system. UI prototypes can be created to help customers understand how the system works.

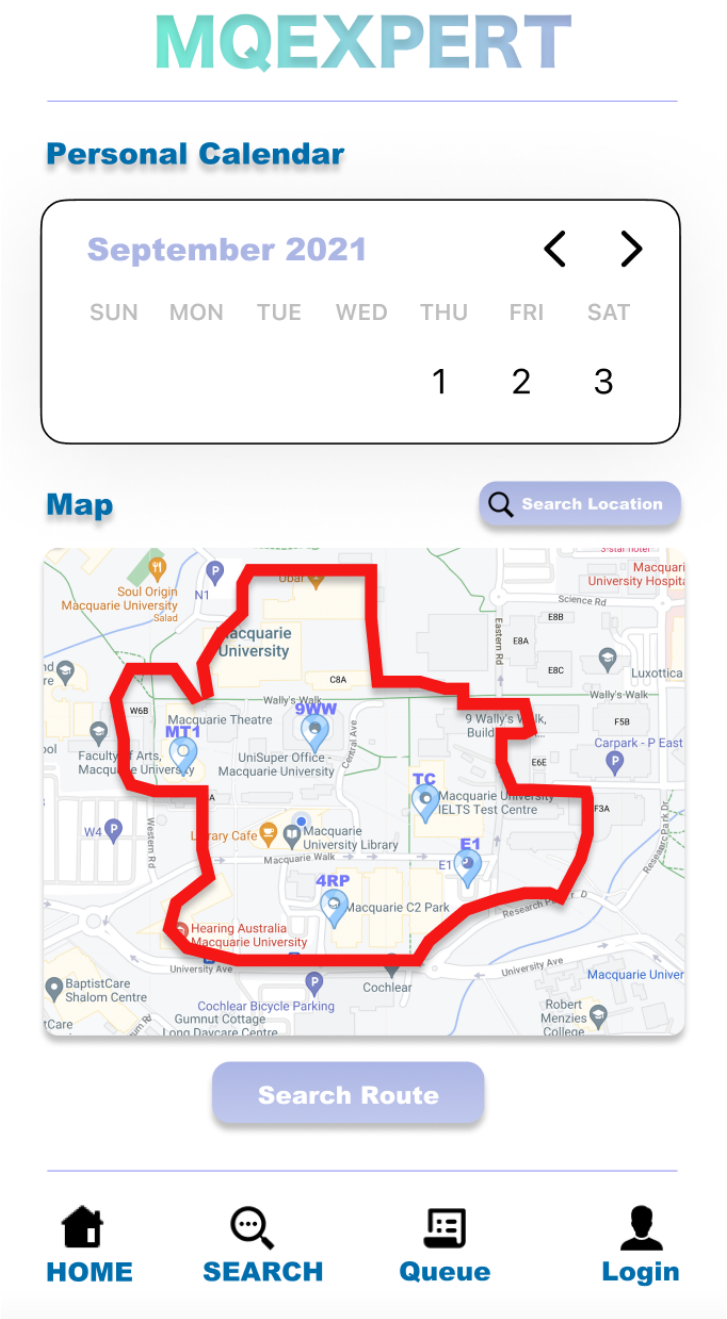


Fig.3) Example of High-Fidelity UI Prototype of MQExpert system made with Figma

Through this Figure 3 UI prototype, clients can check whether the developer understands their requirements properly and designs the system. Furthermore, developers can add functions or change the UI through feedback from clients to adjust the specific feature of the product. In conclusion, through prototype and JAD, insufficient requirements can be supplemented, modified, and deleted.

Me as a client representative for project A

Before the first session(discussion) meeting, I tried to investigate the existing system(Transport NSW) thoroughly. I thought that convenience and safety are the keys to increase user satisfaction with the system, hence I considered deeply which features users really want and propose them to the developer as a client representative. For example, the requirement of the system to classify incident categories from high to low according to the degree of risk provides users with more information about the incident and reduces the probability of exposure to risk. In addition, the requirement to enter into the system whether the staff has completed the First Aid/CPR training is directly related to the arrangement and work of the staff, and this ultimately affects the safety, convenience, and impact of customers. In other words, since the system is related to safety, the requirements that I was most focused on were convenience and the factors indirectly related to the safety of customers.

Furthermore, I was always willing to share ideas when they came up, plan and schedule projects, and lead. I thought that a good client representative would have to irritate the developer and give inspiration, for that reason I gave the developer the right amount of idea and pressure to come up with better systems and features.

However, not everything went effortlessly as planned. The main reason for that is because of the covid-19 circumstances. Since we are not allowed to meet and discuss in person, we had to use a single platform (zoom) to communicate online, which derives a limitation of the resources. In addition, this weakening of human interaction sometimes caused communication failure between client and developer, which resulted in unnecessary time cost.

Log of interactions

DATE	TIME	DURATION	TOPIC	MAIN DECISIONS
24 Aug 2021	6:51 PM	-	Project Choice	Finalised which project we want to do
25 Aug 2021	8:00 PM	30 minutes	General Discussion about A1	Prepare a list of requirements/wants for each project by tomorrow (26th)
26 Aug 2021	8:00 PM	1 hour and 5 minutes	Client Meeting	Agreed upon requirements Discussed SRS structure/document
4 Sep 2021	7:02 PM	30 minutes	Feasibility of Specification	Discussion about the feasibility of the given specification by the client representative
8 Sep 2021	2:08 PM	20 minutes	SRS Draft	Share(show) the 1st draft(version) of the SRS to the client representative
9 September	6:00 PM	30 Minutes	Reflection	Confirmation of elicitation methods and ensured that we were happy with each other (newly-added) requirements
11 September	6:00 PM	40 Minutes	SRS Version 2	Share(show) the 2nd draft(version) of the SRS to the client representative
12 September	5:30 PM	30 Minutes	Reflection	Share opinions for modified, deleted or created requirements
14 September	3:10 PM	60 Minutes	Final Version	Share(show) the final version of the SRS to the client representative

Bibliography

[1] D. S. J Wood, Joint application development, New York: John Wiley & Sons, 1995.

[2] Study.com, [Online]. Available: https://study.com/academy/lesson/joint-application-development-definition-phases-methodology.html. [Accessed 04 09 2021].

[3] James R Rudd,Ken Stern,Scott Isensee, "Low vs. high-fidelity prototyping debate," *Low vs. high-fidelity prototyping debate,* vol. 3, no. 1, p. 76–85, 1996.