iCalendar for iCampus users

Software Design Specification

2021.05.26.

Introduction to Software Engineering 41

TEAM 5 (iCalendar)

Team Leader	Suyoung Min
Team Member	Minseo Kim
Team Member	Chanjong Lee
Team Member	Taewoo Yoo
Team Member	Sumin Ham

CONTENTS

LIST OF TABLES

1. Introduction

This section establishes a plan for system testing to determine if the system is operating as intended to identify and analyze the system for defects after completion of the system. During the development of the 'I-Calendar' system, testing is divided into three stages. The iCalendar project of team 5 is to develop and design a additional functionality used for customizing and sharing calendar with team project members, or friends who using icampus of Sungkyunkwan University. Many typical calendars tend to show all schedules without separating them. This trend is efficient when coordinating as a whole, but not appropriate when managing tasks on a task-by-task basis. So, iCalendar give users a separated custom calendars that can use task-by-task. This Test plan document presents the test for this iCalendar system in many ways. These test-plans are consist of Software Unit Test, and Software Interface Test.

1.1 Purpose

The main purpose of this Test plan document is to provide a description of the technical test for customizing and sharing calendar, iCalendar. This test plan document describes the architecture and design of test for iCalendar. It also provides approach for setting methods to perform several tests. This iCalendars structure and design are based on what is written at Software Design Specification and Software Requirements Specification documents. Based on a previously written document, the purpose of this document is to describe the contents, methods, objectives, etc. of the various tests that will be performed to evaluate our Team 5's iCalendar-system.

1.2 Scope

This section describes what is being tested, which is new to all the functions of a specific product, its existing interfaces, etc. First, to briefly explain the iCalendar system, the iCalendar system is functionality to ease the management of schedule, like lectures, assignments, team project, or other personal jobs, and so on. It is also expected to share official announcements of classes with team members and sync project schedules and processes. This additional functionality can be added on current i-campus and using alarm function in LearningX. This system can provide new way of controlling schedules by using current icampus calendar functionality. Since customized calendar and group sharing are main tasks in I-Calendar, it is crucial for developers to focus on those. The program should make customized calendar objects shown as figure below properly. They should contain informations from the courses of the student according to database. After that, the program should share that calendar with the groups that the student is actually in.

1.3 Definitions, Acronyms, and Abbreviation

The following table explains the acronyms and abbreviations used in this document.

[Table 1] Table of acronyms and abbreviations

Acronyms& Abbreviations	Explanation
SKKU	Sungkyunkwan University
OS	Operating System
GUI	Graphical User Interface
API	Application Programming Interface
UI	User Interface
HTTP	Hypertext Transfer Protocol

The following table defines certain technical terms used in this document.

[Table 2] Table of terms and definitions

Terms	Definitions
User	Someone who uses a system
System administrator	Someone who quantify the calendar.
Back-End	Application part that is not directly accessed by the user, such as the server and database
Front-End	The user interface, also known as the presentation layer of an application
i-campus	The online lecture system of skku

Terms	Definitions
Client (user devic e)	A user device/user that connected to server
Server	A computer or computer program which manages access to a centralized resource or ser vice in a network
Software	The programs and other operating information used by a computer
Network	For connect devices together so that they can share information. In this system, it usually means internet

1.4 References

This document was written in reference by Software Design Specification and Software Requirement Specification written before.

1.5 Overview

There are several essential considerations in setting up software testing. In test plan of Software Design Specification document. The tests are separated by development testing, release testing, user testing. Each of these characteristics like performance, reliability, and security were considered and a test plan was drawn up. And these test will be conducted in two main ways in Software unit test methods, and Software interface test methods.

2. Test method

2.1. Software unit test methods

2.1.1. Pseudocode

- Sign up

```
try{
  Setprofile( User id.Student id);
  if(GetAuthenticationKey() is True)
    return True;
}catch(e){
    alertMessage(″user ID aleady exists!″ or ″invalid iCampus ID!″);
     print(″Failed to signup″);
}
```

- Log in

```
try{
     Loginrequest(Getprofile(user_id));
  if(GetAuthenticationKey() is True)
       return True;
  // and go to the Interface of 'Select Calendar'
}catch(e){
    alertMessage(″wrong ID or password!″);
     print(″Failed to login″);
}
```

- check and accept/reject invitation

```
group_id = getCheckInvitation();
if(CheckInvitation is True){
   Accept(group_id);
}
else if(CheckInvitation is False){
       Refuse(group_id);
}
```

- create manage calendar

```
CreateCustomCalendar(){
       GetICampusSchedule();

  //Get Create, Modify task request
   while(Managing Calendar){
          if(Create task)
            AddTask(calendar_id, task_id);
         if(Delete task)
            DeleteTask(calendar_id, task_id);
   }
   ShowSchedule();
}
```

- create manage task object

```
setTaskInfo(){
        //get information about task from users
        newCustomTask = MakeCustomTask(task_id, task_name, date, time, place, is_alarm);
        SetTask(newCustomTask);
}


//Add custom task to calendar
AddTask(calendar_id, newCustomTask.task_id);
```

```
if( Task.is_alarm is True){
        try{
                SetAlarm(Task.is_alarm, Task.date, Task.time);
        }catch(error){
                alertMessage(″Failed to set task alarm!″);
        }
}
```

- task alarm management

2.2 Software interface test methods

2.2.1. Firebase

All the tests will be held on Firebase platform.

- Connection test

Integration of the two components(I-Calendar and I-Campus) will be tested. The pseudocode is as below.

```
try{
    Send_Ping(From ICampus to ICalendar)
}catch(e){
        print("Failed to send ping");
}
try{
    Send_Ping(From ICalendar to ICampus)
}catch(e){
        print("Failed to send ping");
}
```

- Data Mapping test

When a user enters data at the front-end side, which is then mapped to correct tables in the database server.

If data gets an update from the user end with the same thing, fewer modifications on data need to be updated to the database and well shown in the UI.

- ACID Properties validation test

Atomicity: Simply gives that a transaction is either “PASS” or “FAIL” element.

Consistency: If any type of transaction exists at the user end, that must be conceived into a valid and legal state in the database.

Isolation: When multiple transactions execute at a single unit of time, then the state of the database must be far all over the transaction period.

- data integrity test

Either if changes happen in data or any record update in the database by any of the CRUD properties (Create, Retrieve, Update, and Delete), the latest information displays on each and every User due to its common interface. Therefore, the database testing cases will be designed in an easy way that validates the consistency of data sets at every display of the server.

- Stress test

Stress testing will be used to test the system behavior under extreme conditions and is carried out till the system failure.

- Durability test

Interruption in the performance of I-Calendar during its life span will be provided as a test case, and it will check the endurance capacity of an application under consideration.

3. Software Unit Test

3.1 Signup

3.1.1 iCalendar account signup test case

3.1.1 iCalendar account signup test case

Test case object : To check correct signup function when positive input inserted

[Table 3] iCalendar account signup test case

Step No.	Test Inputs : iCalendar signup info	Expected Results : Interface of ‘Login’ shows up
1	Type in ID, password the user wants to use and a valid iCampus ID into the form	

2	Click ‘Submit’ button	The screen changes into ‘Login’ page

3.1.2 iCalendar account signup test case - Negative

Test case object : To check correct signup function when negative input inserted

[Table 4] iCalendar account signup test case - Negative

Step No.	Test Inputs : Wrong iCalendar signup info	Expected Results : Message alerting the user of wrong info input
1	Type in an already existing ID or invalid iCampus ID with password into the form	
2	Click ‘Submit’ button	An alert message such as ‘user ID already exists!’ or ‘invalid iCampus ID!’ pops up

3.2 Login

3.2.1 iCalendar account login test case

Test case object : To check correct login function when positive input inserted

[Table 5] iCalendar account login test case

Step No.	Test Inputs : iCalendar login info	Expected Results : Interface of ‘Select Calendar’ shows up
1	Type in valid ID with password into the form	
2	Click ‘Login’ button	The screen changes into ‘Select Calendar’ page

3.2.2 iCalendar account login test case - Negative

Test case object : To check correct login function when negative input inserted

[Table 6] iCalendar account login test case - Negative

Step No.	Test Inputs : Wrong iCalendar login info	Expected Results : Login failed message
1	Type in wrong ID or password into the form	
2	Click 'Login' button	An alert message indicating 'wrong ID or password!' pops up

3.3 Check and accept/reject invitation

3.3.1 iCalendar check and accept invitation test case

Test case object : To check correct check and accept invitation function when positive input inserted

[Table 7] iCalendar check and accept invitation test case

Step No.	Test Inputs : Clicking accept button on the screen	Expected Results : Invitation accepted message, the newly shared calendar becomes accessible
1	In 'Select Calendar' page, click 'Check invitations' button	A window presenting a list of invitations toward the user pops up.
2	Click 'Accept' button for the invitation the user would like to accept	The selected invitation disappears from the invitations list and the screen changes into 'Select Calendar' page, the user can find out the newly added calendar on the list. From the view of the inviter, the inviter can see the invitee's iCalendar ID at 'Manage Calendar' page of this shared calendar

3.3.2 iCalendar check and reject test case

Test case object : To check correct check and reject invitation function when positive input inserted

[Table 8] iCalendar check and reject test case

Step No.	Test Inputs : Clicking reject button on the screen	Expected Results : The rejected invitation disappears from the list
1	In ‘Select Calendar’ page, click ‘Check invitations’ button	A window presenting a list of invitations toward the user pops up.
2	Click ‘Reject button for the invitation the user would like to reject	The selected invitation disappears from the invitations list and nothing else happens

3.4 Create and manage calendar

3.4.1 iCalendar create and manage calendar test case

Test case object : To check correct create and manage calendar function when positive input inserted

CASE 1 – Create new calendar / edit existing calendar

[Table 9] Create new calendar / edit existing calendar

Step No.	Test Inputs : iCalendar information needed for managing custom calendar	Expected Results : Input info reflected to calendar list/the calendar
1 (create)	Click ‘New Calendar’ button / click one out of the calendars on the list	The screen changes into ‘Manage Calendar’ page
2 (create, edit)	Type in name and description about the calendar and click ‘save’ button	The screen changes into ‘Select Calendar’ page and from the calendars list, the user can find out newly added calendar / modified calendar name

CASE 2 – Browse on(through) existing calendar(s)

[Table 10] Browse on(through) existing calendar(s)

Step No.	Test Inputs : iCalendar information needed for managing custom calendar	Expected Results : Input info reflected to calendar list/the calendar
1	In ‘Select Calendar’ menu : Click ‘triangle buttons’ on both flanks of the calendar list on the screen In ‘Manage Calendar’ menu : Click one of two ‘triangle buttons’ on top of the current calendar	In ‘Select Calendar’ menu : Calendars being shown on the screen slides to either left or right, and another 3(or less number of) calendars appear In ‘Manage Calendar’ menu : If the current page of the calendar was ‘May 2021’, this turns into ‘April 2021’ or ‘June 2021’ according to the clicked button

CASE 3 – Delete existing calendar

[Table 11] Delete existing calendar

Step No.	Test Inputs : iCalendar information needed for managing custom calendar	Expected Results : Input info reflected to calendar list/the calendar
1	Click a calendar that the user wants to delete	The screen changes into ‘Manage Calendar’ page
2	Click ‘Delete’ button on the screen	A message(along with ‘Yes’, ‘No’ buttons) asking whether the user really wants to delete the calendar pops up.
3	Click ‘Yes’ button	The screen changes into ‘Select Calendar’ page and from the calendars list, the user can find out that the calendar no more exists

CASE 4 – Share existing calendar

[Table 12] Share existing calendar

Step No.	Test Inputs : iCalendar information needed for managing custom calendar	Expected Results : Input info reflected to calendar list/the calendar
1	Click a calendar that the user wants to share	The screen changes into ‘Manage Calendar’ page
2	Click ‘+’ button on the right side of member list of the calendar	The screen changes into ‘Invite Team Member’ page
3	Search invitee’s iCalendar ID with keyword, select the invitee from the list, type in message and click ‘Send’ button	A message saying ‘Invitation successfully sent to your friend!’ pops up, then the screen changes into ‘Manage Calendar’ page

3.4.2 iCalendar create and manage calendar test case – Negative

Test case object : To check correct create and manage calendar function when negative input inserted

CASE 1 – Create new calendar / edit existing calendar

[Table 13] Create new calendar / edit existing calendar

Step No.	Test Inputs : Wrong iCalendar information needed for managing custom calendar	Expected Results : Message alerting the user of wrong info input
1 (create)	Click 'New Calendar' button / click one out of the calendars on the list	The screen changes into 'Manage Calendar' page
2 (create, edit)	Type in already existing name of calendar or leave the description about the calendar empty then click 'save' button	An alert message indicating 'already existing calendar name!' or 'description must be filled out!' pops up

CASE 2 – Browse on(through) existing calendar(s) : N/A as inputs required for browsing on a calendar are only mouse clicks

CASE 3 – Delete existing calendar : N/A as inputs required for deleting a calendar are only mouse clicks

CASE 4 – Share existing calendar

[Table 14] Share existing calendar

Step No.	Test Inputs : Wrong iCalendar information needed for managing custom calendar	Expected Results : Message alerting the user of wrong info input
1	Click a calendar that the user wants to share	The screen changes into 'Manage Calendar' page
2	Click '+' button on the right side of member list of the calendar	The screen changes into 'Invite Team Member' page
3	Search for non-existing iCalendar ID / select no invitee from 'User List' / leave 'Message' form empty and click 'Send' button.	An alert message indicating 'Non-existing iCalendar ID!' or 'Select an invitee from the user list!' or 'Message must be filled out!' pops up

3.5 Create and manage task object

3.5.1 iCalendar create and manage task object test case

Test case object : To check correct create and manage task object function when positive input inserted

CASE 1 – Create new task object

[Table 15] Create new task object

Step No.	Test Inputs : iCalendar information needed for managing task object	Expected Results : Input info reflected to the calendar/task object
1	In 'Manage Calendar' page, click a date from the calendar to which the user wants to add a new task object then click 'Object' button	The screen changes into 'Manage Object' page
2	In 'Manage Object' page, click one to choose type of the task among lecture/assignment/ custom. Then click one from lecture/assignment list or type in description then click 'Save' button	The screen changes into 'Manage Calendar' page, and the user can see the newly added object from the calendar

CASE 2 – Edit existing task object

[Table 16] Edit existing task object

Step No.	Test Inputs : iCalendar information needed for managing task object	Expected Results : Input info reflected to the calendar/task object
1	In 'Manage Calendar' page, click a task object from the calendar that the user wants to edit then click 'Object' button	The screen changes into 'Manage Object' page
2	In 'Manage Object' page, click one to choose type of the task among lecture/assignment/ custom. Then click one from lecture/assignment list or type in description then click 'Save' button	The screen changes into 'Manage Calendar' page, and the user can see from the calendar that the object is modified

CASE 3 – Delete existing task object

[Table 17] Delete existing task object

Step No.	Test Inputs : iCalendar information needed for managing task object	Expected Results : Input info reflected to the calendar/task object
1	In ‘Manage Calendar’ page, click a task object from the calendar that user wants to delete then click ‘Delete’ button	A message(along with ‘Yes’, ‘No’ buttons) asking whether the user really wants to delete the task object pops up.
2	Click ‘Yes’ button	The screen changes into ‘Manage Calendar’ page and from the calendar, the user can find out that the task object no more exists

3.5.2 iCalendar create and manage task object test case – Negative

Test case object : To check correct create and manage task object function when negative input inserted

CASE 1 – Create new task object

[Table 18] Create new task object

Step No.	Test Inputs : Wrong iCalendar information needed for managing task object	Expected Results : Message alerting the user of wrong info input
1	In ‘Manage Calendar’ page, click a date from the calendar to which the user wants to add a new task object then click ‘Object’ button	The screen changes into ‘Manage Object’ page
2	In ‘Manage Object’ page, choose no type of the task / nothing from lecture or assignment list / choose ‘custom’ and type nothing into the description form Then click ‘Save’ button	An alert message indicating ‘Choose type of your object!’ / ‘Specify your lecture or assignment object!’ / ‘Description must be filled out!’ pops up

CASE 2 – Edit existing task object

[Table 19] Edit existing task object

Step No.	Test Inputs : Wrong iCalendar information needed for managing task object	Expected Results : Message alerting the user of wrong info input
1	In ‘Manage Calendar’ page, click a task object from the calendar that the user wants to edit	The screen changes into ‘Manage Object’ page

	then click ‘Object’ button	
2	In ‘Manage Object’ page, choose no type of the task / nothing from lecture or assignment list / choose ‘custom’ and type nothing into the description form Then click ‘Save’ button	An alert message indicating ‘Choose type of your object!’ / ‘Specify your lecture or assignment object!’ / ‘Description must be filled out!’ pops up

CASE 3 – Delete existing task object : N/A as inputs required for deleting a task object are only mouse clicks

3.6 Task alarm management

3.6.1 iCalendar set alarm for task object test case

Test case object : To check correct set alarm for task object function when positive input inserted

[Table 20] To check correct set alarm

Step No.	Test Inputs : iCalendar task alarm info	Expected Results : Alarm for a certain task object alerted from the user’s smartphone through ‘Learning X’ application
1	In ‘Manage Object’ page, choose among options like ‘before n minutes / n hours / n days from the due date and time’ or choose specific date and time the user wants to get alarmed	
2	Click ‘Save’ button	An alert message ‘Alarm for the task successfully set up!’ pops up. And the user should be alerted about the task object from the user’s smartphone through ‘Learning X’ application

3.6.2 iCalendar set alarm for task object test case – Negative

Test case object : To check correct set alarm for task object function when negative input inserted

[Table 21] To check correct set alarm - Negative

Step No.	Test Inputs : Wrong iCalendar task alarm info	Expected Results : Message alerting the user of wrong info input
1	In 'Manage Object' page, choose a date and time that is beyond the due date and time of the task object(for 'assignment type' object)	
2	Click 'Save' button	An alert message 'Check out the due date and time of your task again!' pops up.

3.7 Logout

3.7.1 iCalendar logout test case

Test case object : To check correct logout function when positive input inserted

[Table 22] To check correct logout function

Step No.	Test Inputs : A click on the logout button	Expected Results : Interface of 'Login' shows up
1	Click 'Logout' button on the screen	The screen changes into 'Login' page

3.7.2 iCalendar logout test case - Negative

N/A as input required for logout function is only a mouse click

4. Software Interface test

4.1. Firebase

4.1.1 Connection test

Test case object: To check external software interfaces of Firebase.

[Table 23] To check correct logout function - Negative

Step No.	Test Input : Ping Command	Expected Results : Ping Status
1	Send Ping Command	Receive Ping Status is true

4.1.2 Data Mapping test

Test case object: To check UI and Database are connected properly.

CASE 1– Calendar & task object data

[Table 24] To check external software interfaces of Firebase.

Step No.	Test Input : Request pages that use DB in iCalendar	Expected Results : Show the appropriate data within the page
1	Move to 'Select calendar' page	Displays a list of calendars that a user can access.
2	Click a specific calendar in the 'Select Calendar' menu page.	The task objects appear on the corresponding date of the calendar.

CASE 2 – Members of shared calendar

[Table 25] Members of shared calendar

Step No.	Test Input	Expected Results
1	Click a specific calendar in the 'Select Calendar' menu page.	The task objects appear on the corresponding date of the calendar.
2	Click 'Shared Users' Button	Shows the names of the team members who are currently sharing the calendar with.

4.1.3 ACID Properties Validation test

4.1.3.1 Atomicity test

Test case object: To verify that it has the same data in both sides.

[Table 26] Atomicity test

Step No.	Test Input : Server disconnection while accepting Team Member Invitation	Expected Results : Invalidates acceptance of invitations
1	Accept invitation in 'Invite Team Member' page	Send invite accept to server.
2	Disconnected during data transfer	
3	Reload 'Invite Team Member' page	The invitation must be seen again without accepted.

4.1.3.2 Consistency test

Test case object: To check the updated and most recent values/status of shared data should appear on all the forms and screens.

CASE 1- Create new calendar / edit existing calendar

[Table 27] Create new calendar / edit existing calendar

Step No.	Test Input : Add / Modify Data Without Errors	Expected Results : Normal result output
1, 2	Same as 3.4.1 iCalendar create and manage calendar test case – CASE 1	
3	Check database	Newly added / changed calendar name is shown normally in the database.

CASE 2 – Delete existing calendar

[Table 28] Delete existing calendar

Step No.	Test Input	Expected Results
1, 2	Same as 3.4.2 iCalendar create and manage calendar test case – CASE 3	
3	Check database	Deleted calendar is not shown in the database.

CASE 3 – Create new task object / Edit existing task object

[Table 29] Create new task object / Edit existing task object

Step No.	Test Input	Expected Results
1, 2	Same as 3.5.1 iCalendar create and manage task object test case – CASE 1, 2	

3	Check database	Newly added / changed task object is shown normally in the database.

CASE 4 – Delete existing task object

[Table 30] Delete existing task object

Step No.	Test Input	Expected Results
1, 2	Same as 3.4.2 iCalendar create and manage calendar test case – CASE 3	
3	Check database	Deleted task object is not shown in the database.

CASE 5 – Accept invitation and add a new member to a shared calendar

[Table 31] Accept invitation and add a new member to a shared calendar

Step No.	Test Input	Expected Results
1	In 'Select Calendar' page, click 'Check invitations' button	A window presenting a list of invitations toward the user pops up.
2	Click 'Accept' button for the invitation the user would like to accept	The selected invitation disappears from the invitations list and the screen changes into 'Select Calendar' page, the user can find out the newly added calendar on the list. From the view of the inviter and other members, they can see the invitee's iCalendar ID at 'Manage Calendar' page of this shared calendar.

CASE 6 – Reject invitation

[Table 32] Reject invitation

Step No.	Test Input	Expected Results
1	In 'Select Calendar' page, click 'Check invitations' button	A window presenting a list of invitations toward the user pops up.
2	Click 'Reject' button for the invitation that the user would like to reject	The selected invitation disappears from the invitations list and nothing else happens. From the view of the inviter and

		other members, they also can see that no new member has been invited.

4.1.3.3 Isolation test

Test case object: Check data when caller and callee have different processing times or modify one data at the same time.

[Table 33] Isolation test

Step No.	Test Input : Save duplicate data simultaneously	Expected Results : Save only last the request
1	Two members click a shared calendar at the same time.	The screen changes into ‘Manage Calendar’ page
2	Each two members change the name or description of the calendar and press the ‘Save’ button.	Changes in the first saved member are covered by changes in the later saved member.

4.1.4 Data Integrity test

Test case object: To check the data integrity (Entity integrity, Domain Integrity, etc.) is observed.

CASE 1 – Entity Integrity test

[Table 34] Entity Integrity test

Step No.	Test Input : Save data with invalid data type	Expected Results : Error detection and alert the error
1	Click ‘New Calendar’ button / click one out of the calendars on the list	The screen changes into ‘Manage Calendar’ page
2	Click the ‘Save’ button without typing a name or already existing name of calendar for the calendar.	An alert message indicating ‘Please enter a name for the calendar’ / ‘already existing calendar name!’ pops up and not saved.

CASE 2- Domain Integrity test

[Table 35] Domain Integrity test

Step No.	Test Input	Expected Results
1	In ‘Manage Calendar’ page, click a task object from the calendar that the user wants to edit then click ‘Object’ button	The screen changes into ‘Manage Object’ page
2	In ‘Manage Object’ page, type the time of schedule by alphabet or over 24 o’clock and click ‘Save’ button.	An alert message indicating ‘Please enter the time between 0:00 and 24:00.’ pops up and not saved.

4.2 Notification API

4.2.1 Connection test

Test case object: To check external software interfaces of notification API.

[Table 36] Connection test

Step No.	Test Input : Ping command	Expected Results : Ping Status
1	Send Ping Command	Show Ping Status is true

4.2.2 Runtime error detection

4.2.2.1 Stress test

Test case object : To check when there are infinitely many notifications.

[Table 37] Stress test

Step No.	Test Input : Set many alarms	Expected Results : Error
1	Set many task objects’ alarm time at the same time	An alert message ‘Alarm for the task successfully set up!’ pops up.
2	The time to set the notification time.	The Notification API sends an alarm to the user’s smartphone through ‘Learning X’ application. However, there is a possibility that too many alarms are sent at the same time, causing errors.

4.2.2.2 Durability test

Test case object : To check whether a notification is sent when disconnected or latency from the server

[Table 38] Durability test

Step No.	Test Input : Server disconnection	Expected Results : Waiting
1	Save a notification for a task object.	An alert message 'Alarm for the task successfully set up!' pops up.
2	Disconnect from the server and then the time to set the notification time.	Because the connection to the server is lost, the notification API waits until it is connected, and sends an alarm again when it is connected.

5. Supporting Information

This Software Test Plan documentation was written according to Software Engineering 41 class' form.

5.1 Document History

–

[Table 39] Document History

Date	Version	Description	Writer
2021/05/19	0.1	Style and overview	Suyoung Min
2021/05/22	0.2	Addition of 1	Suyoung Min
2021/05/22	0.3	Addition of 2.1	Sumin Ham
2021/05/22	1.1	Addition of 3.1	Chanjong Lee
2021/05/22	1.2	Addition of 3.2	Taewoo Yoo
2021/05/22	1.3	Addition of 4.1	Minseo Kim
2021/05/23	1.4	Addition of 5	Suyoung Min
2021/05/23	1.5	Addition of 2.2	Sumin Ham
2021/05/24	1.5	Revision of 2	Sumin Ham

Date	Version	Description	Writer
2021/05/24	1.6	Revesion of 3.2	Taewoo Yoo
2021/05/25	1.7	Revision of 3.1	Chanjong Lee
2021/05/26	1.9	Revision of structure	Minseo Kim
2021/05/26	2.0	Revision of 5	Suyoung Min