



iCalender for iCampus users

Software Requirement Specification

2021.04.24.

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

1.1.	1.	Introduction	7
1.1.		Purpose.....	7
1.2.		Scope.....	7
1.3.		Definitions, Acronyms, and abbreviation.....	7
1.4.		References.....	9
1.5.		Overview.....	9
1.2.	2.	Overall Description	9
2.1.		Product Perspective.....	9
2.1.1.		System Interfaces.....	10
2.1.2.		User Interfaces.....	10
2.1.3.		Hardware Interfaces.....	10
2.1.4.		Software Interfaces.....	10
2.1.5.		Communications Interfaces.....	10
2.1.6.		Operations.....	11
2.1.6.1.		System administrator.....	11
2.1.6.2.		User.....	11
2.2.		Product Functions.....	11
2.2.1.		Sign Up.....	11
2.2.2.		Create Custom Calendar.....	12
2.2.3.		Modify Custom Calendar.....	12
2.2.4.		Delete Custom Calendar.....	12
2.2.5.		Group Sharing.....	12
2.2.6.		Invitation management.....	12
2.2.7.		Management Task Alarm.....	13
2.3.		User Characteristics.....	13
2.3.1.		System Administrator.....	13
2.3.2.		User.....	13
2.4.		Constraints.....	14
2.5.		Assumptions and Dependencies.....	14
1.3.	3.	Specific Requirements	14
3.1.		External Interface Requirements.....	14
3.1.1.		User Interfaces.....	14
3.1.2.		Hardware Interfaces.....	22
3.1.3.		Software Interfaces.....	22

3.1.4.	Communication Interfaces	23
3.2.	Functional Requirements	24
3.2.1.	Use Case	24
3.2.2.	Use Case Diagram	27
3.2.3.	Data Flow Diagram	28
3.3.	Performance Requirements	28
3.3.1.	Static numerical requirement	28
3.3.2.	Dynamic numerical requirement	29
3.4.	Logical Database Requirements	29
3.5.	Design Constraints	29
3.5.1.	Standards compliance	30
3.6.	Software System Characteristics	30
3.6.1.	Product Requirements	30
3.6.1.1.	Usability Requirements	30
3.6.1.2.	Performance Requirements	30
3.6.1.3.	Usability Requirements	30
3.6.1.4.	Security Requirements	31
3.6.2.	Organizational Requirements	31
3.6.2.1.	Environmental Requirements	31
3.6.2.2.	Operational Requirement	31
3.6.3.	External Requirements	31
3.6.3.1.	Safety / Security Requirement	32
3.6.3.2.	Regulatory Requirement	32
3.7.	Organizing the Specific Requirements	32
3.7.1.	Context Model	33
3.7.2.	Process Model	33
3.7.3.	Interaction Model	33
3.7.4.	Behavior Model	34
3.7.4.1.	Data Flow Diagram	34
3.7.4.2.	Sequence Diagram	34
3.8.	System Architecture	34
3.9.	System Evolution	35
3.9.1.	Limitation and Assumption	35
3.9.2.	Evolutions of Hardware and Change of User Requirements	35
1.4.	4. Supporting Information	35
4.1.	Software Requirement Specification	35
4.2.	Document History	36

LIST OF FIGURES

[Figure 1] Use case diagram.....	26
[Figure 2] Data flow diagram.....	27
[Figure 3] Context model.....	32
[Figure 4] Overall process model.....	32
[Figure 5] Sequence diagram.....	33
[Figure 6] System architecture of the system.....	34

LIST OF TABLES

[Table 1] Table of acronyms and abbreviations.....	8
[Table 2] Table of terms and definitions.....	8
[Table 3] User interface of input processing using mouse clicks.....	15
[Table 4] User interface of signup.....	16
[Table 5] User interface of select calendar.....	17
[Table 6] User interface of check invitation.....	18
[Table 7] User interface of manage calendar.....	19
[Table 8] User interface of invite team member.....	20
[Table 9] User interface of manage object.....	21
[Table 10] Hardware interface of applicable device for the system.....	22
[Table 11] Software interface of firebase real-time database.....	22
[Table 12] Software interface between iCampus and the developed system.....	24
[Table 13] Communication interface of client and host.....	24
[Table 14] Use case of sign-up.....	25
[Table 15] Use case of log-in/out.....	25
[Table 16] Use case of Calendar.....	26
[Table 17] Use case of sharing calendar.....	26
[Table 18] Document History.....	36

2. Introduction

2.1. Purpose

This document is a Software Requirements Specification (SRS) for adding iCalender (Custom & Sharing Calender for iCampus users) service. iCalender is designed in i-Campus environment by Team 5 of the Introduction to Software Engineering at Sungkyunkwan University. The requirements for this are summarized, analyzed, and the system is designed and implemented based on the contents described.

the purpose of this document is to outline and publish the Requirement Specification for a additional functionality of i-Campus Calender for i-Campus normal users and users who have team project classes. The current i-Campus Calender functionality is uncomfortable to view and not user-friendly because it is not organized efficiently. But, iCalender will provide comfortable enviroment to manage several schedules by customizing calenders.

Many typical calenders 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, iCalender give users a separated custom calenders that can use task-by-task.

2.2. Scope

The iCalender system is functionality to ease the management of schedule, like lectures, assignments, team project, or other personal jobs, and so on. 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. Above all, we hope to provide a comfortable user experience along with the best possible pricing available to the users.

2.3. 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
Client (user device)	A user device/user that connected to server
Server	A computer or computer program which manages access to a centralized resource or service 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

2.4. References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library
<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

- Team 5. “Software Requirement Specification”. SKKU, Last Modified: Apr. 23, 2021.
https://github.com/skkuse/2021spring_41class_team5/tree/main/doc/Requirements.pdf
- Multimedia Service Team. “Software Requirement Specification of Multimedia Contents-aware Intelligent Information Service System”. Kangwon National University. (2007)

2.5. Overview

The remainder of this Software Requirements Specifications Document includes three chapters. The second chapter, Overall Description provides a description of the product perspective, including the several interfaces, the system functionality and system interaction with other systems. This Overall Description also introduces different types of stakeholders and their interaction with the system. Additionally, the chapter also mentions item details, the system constraints, assumptions, and dependencies of the product. The third chapter, Specific Requirements provides the requirements specification in detailed terms, including a description of the different system interfaces and the software system characteristics. A range of specification techniques are used to specify the requirements more precisely for a variety of users. It also shows a lot of use cases and a data dictionary. The fourth chapter, Supporting Information deals with the prioritization of the requirements. It includes a supporting documentation, a timeline of the documentation of this SRS for the application, iCalender. All team5 members contributed equally to the production of iCalender. And, it will be our pleasure that readers of this document enjoy viewing.

3. Overall Description

3.1. Product Perspective

This product is designed for student users who using i-campus to manage several different tasks through seperate calendars. This functionality will give comfortability to users who wants to carry team project more efficiently by using Task object from i-campus. In team project, if users want to synchronize common schedule with their team members, they can get task object from i-campus database like lectures, assignments, or other custom task object,

then by using alarm function to mobile phone, they can synchronize their tasks stage-by-stage. In personal perspective, users can also make custom calendar with to-do lectures, assignments. These don't have to synchronize with each other person, but alarm can be used to get proper control of time-slices.

3.1.1. System Interfaces

This system is basically existed in i-campus. We just add the several functions on it. So, The users' custom calendar information and edited information are stored in the i-campus server that control the basic calendar information.

3.1.2. User Interfaces

An interface is provided through the screen of the desktop computer, and information can be basically input to i-campus homepage. Depending on the preferences entered by the user, the user should be able to be recommended a laptop.

The administrator can access Laptop database. The administrator is provided with an interface through the basic GUI of firebase, and supports registration, deletion, maintenance, and management of laptop information.

3.1.3. Hardware Interfaces

The system is intended for Desktop that can execute chrome internet browser.

3.1.4. Software Interfaces

The system is intended for Internet browser chrome version recommended 90.0.4430.85.

3.1.5. Communications Interfaces

User device and server communicate with HTTP protocol.

3.1.6. Operations

3.1.6.1. System administrator

- Set user-specified menu of calendar
 - ✓ Conditions of stored custom calendar are controlled by System administrator.
 - ✓ Decide which calendar that users want to use.

3.1.6.2. User

- Login
 - ✓ User can login using skku-icampus account, etc.
- Get Calender
 - ✓ User can get specific calendar that they choose in calendar select menu.
- Calendar
 - ✓ User can modify calender's task object information, adding or deleting the objects or setting alarm at specific task objects.
 - ✓ User can see the search history
- Menu operation
 - ✓ Users can check invitation from other team project members and accept or deny invitation.
 - ✓ Users can select calender that they want to view, or edit some conditions.
 - ✓ Users can create new custom calendars by using task object got from icampus system.

3.2. Product Functions

3.2.1. Sign Up

After a user has visited the website, they must register with the system to use it. The login page has a sign up button that redirects you to the page where you register the user. After entering information in the required fields, users can register with the system using the sign up button. Users register using their SKKU account. After registering with the system, users can log in to use the system.

3.2.2. Create Custom Calendar

After logging in, users can create custom calendars. When creating a calendar for the first time, tasks are loaded from I-campus. The imported tasks are divided into class objects and assignment objects. Also, users can create custom tasks. Users can place these tasks anywhere on the calendar. However, tasks with a deadline cannot be deployed beyond the deadline. If

you click the 'Create' button, a custom calendar in which tasks are registered is created. The created calendar can be viewed on the 'My Calendar' page.

3.2.3. Modify Custom Calendar

On the 'My Calendar' page, users can enter the 'Edit Calendar' page by long clicking on the custom calendar. Users can load new tasks from I-campus. In addition, users can create custom tasks. As with creating a calendar, tasks with a deadline cannot be deployed beyond the deadline. When the modification is complete, the user can click the 'Edit' button to save the modified custom calendar.

3.2.4. Delete Custom Calendar

On the 'My Calendar' page, users can delete a custom calendar by clicking and holding it. Deleted calendars cannot be recovered. When a user tries to delete a calendar, it asks the user if he really wants to delete this calendar. Also, it notifies users that deleted calendars cannot be recovered. When the user clicks the 'Delete' button, the custom calendar will be deleted.

3.2.5. Group Sharing

Users can select a calendar from the 'My Calendar' page and share it to a group. Users can invite other users by entering the IDs of other users to share or by creating a calendar sharing link. Invited users can use the shared calendar together.

3.2.6. Invitation management

On the 'Groups' page, users can see their new invitations. Users can check in advance which user's calendar has been invited. Users can accept the invitation by clicking on the 'Accept' button. When you accept the invitation, the invited calendar is added to the 'My Calendar' page. If the user wants to decline the invitation, they can do so by clicking the 'Reject' button.

3.2.7. Management Task Alarm

On the 'My Calendar' page, users can select a calendar to manage their alarms. The user decides what kind of task's alarm is turned on and off. The specified alarm is notified to the user through 'Learning X' at the specified time.

3.3. User Characteristics

3.3.1. System Administrator

System administrators are limited to those who have a good knowledge of the system and a general understanding of the system. It is assumed that the system administrator understands the functionality of I-campus and has sufficient capacity to handle system problems. It is also assumed that a system administrator has completed computer science or similar research, or has completed training to become a system administrator, such as web development or network management, or has equivalent qualifications. In addition, it is assumed that the system administrator has the ability to add/modify new functions to the system to reflect the user's needs. In addition, the system administrator must have the ability to solve problems in the connection with I-campus.

3.3.2. User

In this document, the users commonly referred to are students of SKKU. It is assumed that the user is able to read Korean smoothly, understands its meaning, and is using I-campus, but is feeling uncomfortable with the calendar function of I-campus. In addition, it is assumed that the user needs to share a schedule with others because of a group activity, such as a team project or study. In addition, it is assumed that users can read and understand basic English, and have education and literacy skills to the extent that they have little or no difficulty installing and using applications on their smartphones. It is also assumed that users can freely visit websites using the Internet on their desktop. In general, it is assumed that Koreans aged 18 to 40 are the users.

3.4. Constraints

The system will be designed and implemented based on the contents mentioned in this document. Other details are designed and implemented by selecting the direction preferred by the developer, but the following items are observed.

- Use the technology that has already been widely proven.
- Avoid using technology or software that requires a separate license or pays for royalty. (Exclude this provision if this is the only technology or software that the system must require.)

- Decide in the direction of seeking improvement of overall system performance.
- Decide in a more user-friendly and convenient direction
- Use open source software whenever possible
- Consider the system cost and maintenance cost
- Consider future scalability and availability of the system
- Optimize the source code to prevent waste of system resources
- Consider future maintenance and add sufficient comments when writing the source code
- Websites must comply with Web-standards and Web-content accessibility guidelines.
- Promote web standardization so that any I-campus user can use the service regardless of the user's service environment.
- Icons, pictures, photos, and fonts should be purchased and used in a way that does not violate copyright, or those that are commercially available.
- The linkage with I-campus should be smooth.

3.5. Assumptions and Dependencies

All systems in this document are written with the assumption that the I-campus system is accessible (such as accessing the properties of the source code) and can be given additional functionality. Therefore, if access to the I-campus system is not possible, it can be operated by separating it into a separate web system.

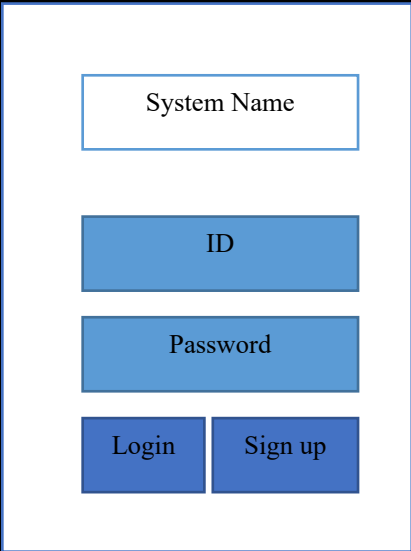
4. Specific Requirements

4.1. External Interface Requirements

4.1.1. User Interfaces

[Table 3] User interface of input processing using mouse clicks

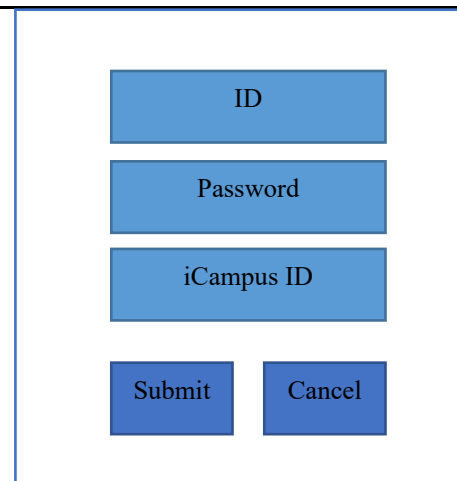
Name	Basic User Interaction Using Mouse Clicks
Purpose/Description	Users transmit their instructions by clicking buttons on the monitor screen
Input source/ Output destination	User/ user device equipped with Window OS
Range/Accuracy/	Range according to the number of buttons on the screen/

Name	Basic User Interaction Using Mouse Clicks	
Margin of error	Accuracy according to the accuracy of mouse click from users/ Margin of error of mouse click sensitiveness	
Unit	A click	
Time/ Velocity	Asynchronous user input/ Instant execution of a user instruction	
Relationship with other input/outputs	After receiving all the inputs, the user device transmits the input data to the server for processing the input data and request desired output data	
Format and configuration of screen		<p>1. An activity screen connected to an XML file, mainly composed of TextViews and ImageViews</p> <p>2. A Textview and an ImageView provide a basis for choosing the following input, and several Buttons are ready to receive the inputs from users</p> <p>3. Users are to click a desired button to interact with the system</p> <p>4. From the example image on the left, the user can click 'Sign up' button to create a new account or put in ID, Password and click 'Login' button to login.</p>
Format and configuration of window	N/A	
Data type	Int type value of a button code	
Instruction type	Instruction mapping according to the value of a button code	
Exit message	N/A	

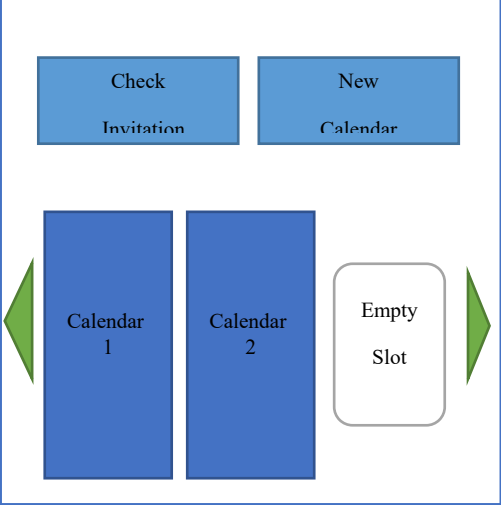
[Table 4] User interface of signup

Name	Signup Interface
Purpose/Description	Users are asked to fill out stuffs like ID, Password, iCampusID to create an account for the system. iCampusID is also needed because the system can relate with iCampus server and get the information about lectures,

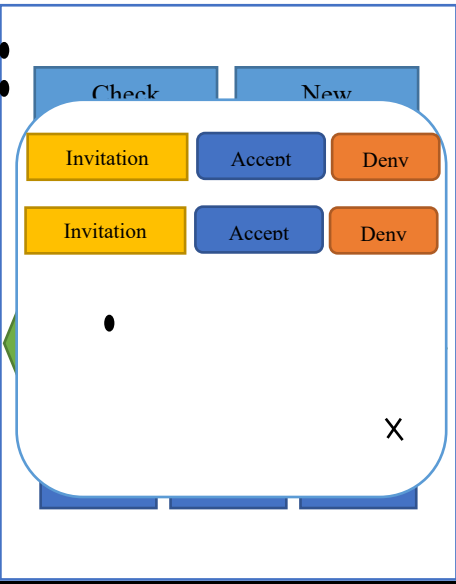
Name	Signup Interface	
	<p>assignment of the user by using the one's iCampusID.</p> <p>When users answer all the questions and press 'Submit' button, the data is transmitted to the server from the user device</p>	
Input source/ Output destination	<p>Host server / Client</p> <p>Client / Host server</p>	
Range/Accuracy/ Margin of error	N/A	
Unit	Screen	
Time/ Velocity	After all the input data were received to the user device/ Communication time between the server and the user device	
Relationship with other input/outputs	ID is used when searching for users to share a calendar with, and iCampusID is used in association with iCampus server when the user is trying to create a task object later	
Format and configuration of screen	<p>1. An activity screen connected to an XML file, mainly composed of TextViews.</p> <p>2. Several TextViews regarding user information with blanks are presented on the screen.</p> <p>3. When the user fills out all the blanks and clicks the button 'Submit', all the account information data is transmitted to the server for creating a new account</p>	
Format and configuration of window	Activity with XML document	
Data type	Text, Widget	
Instruction type	Instruction mapped to the button	
Exit message	N/A	



[Table 5] User interface of select calendar

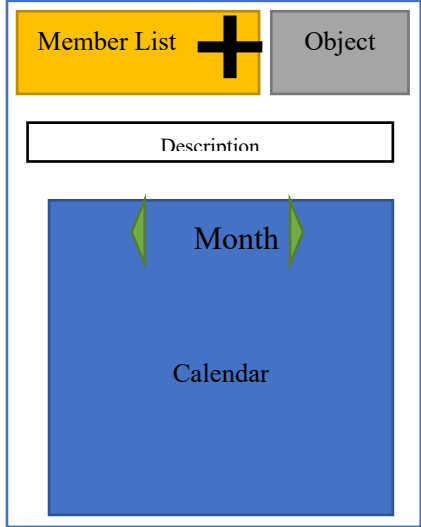
Name	Select Calendar
Purpose/Description	<p>Present a list of calendars created already about which the user can traverse by clicking triangle buttons on flanks of the screen if the user's got more than 3 calendars. Each calendar can be managed after you click the calendar and get over to a 'manage calendar' interface.</p> <p>You can also create a new calendar by clicking 'New Calendar' button on top right of the screen, and to the left there is a button with which you can check out invitations you have received from other users.</p>
Input source/ Output destination	Host server / Client
Range/Accuracy/ Margin of error	Up to 3 calendars created by the user show up on the screen / N/A
Unit	Screen
Time/ Velocity	Asynchronous user input/ Communication time between the server and the user device
Relationship with other input/outputs	'New Calendar' button increases the number of calendars on the list as you can add a new one with that button.
Format and configuration of screen	<p>1. An activity screen connected to an XML file, mainly composed of TextViews and ImageViews shows Textviews and ImageViews that show you some buttons and a list of calendars.</p> <p>2. The button 'Check Invitation' pops up a window where the user can check out invitation messages.</p> <p>3. User can get over to 'manage calendar' interface by either clicking 'New Calendar' button or one of the 'Calendar N' buttons</p> 
Format and configuration of window	Activity with XML document
Data type	Image, Text, Widget
Instruction type	Instruction mapped to the button
Exit message	"Find other laptops!"

[Table 6] User interface of check invitation

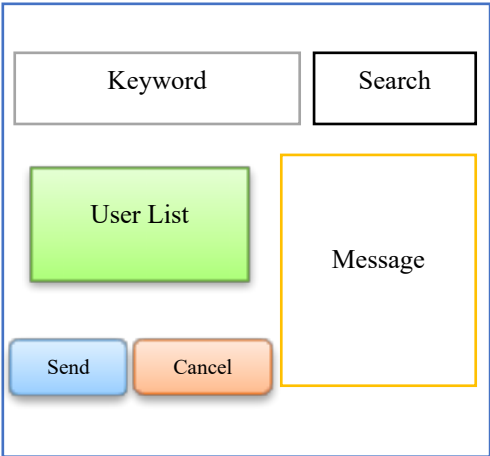
Name	Check Invitation
Purpose/Description	User can check out invitation messages from other users if there are any, then either accept or deny the invitation. By accepting the invitation, the user is granted with a right to edit the calendar proposed by the inviter.
Input source/ Output destination	Host server / Client Client / Host server
Range/Accuracy/ Margin of error	You can see up to 8 invitations on the window at once / N/A
Unit	Pop-up window
Time/ Velocity	Asynchronous user input/ Communication time between the server and the user device
Relationship with other input/outputs	Accepting a calendar increases the number of calendars you can find from the list of 'Select Calendar' interface.
Format and configuration of screen	N/A
Format and configuration of window	 <ol style="list-style-type: none"> 1. A window pops up and you can find linear layout of the list of invitations you have received from other users. 2. If you have more than 8 invitations then you can scroll down to see more of them. 3. The user can either accept or deny on each invitation by clicking corresponding button, to see the invitation object disappears since the process for that is completed. <p>There is a button with the symbol 'x' you can click on when you want to close the pop-up window.</p>
Data type	Text, Image, Widget
Instruction type	Instruction mapped to the button
Exit message	N/A

[Table 7] User interface of manage calendar

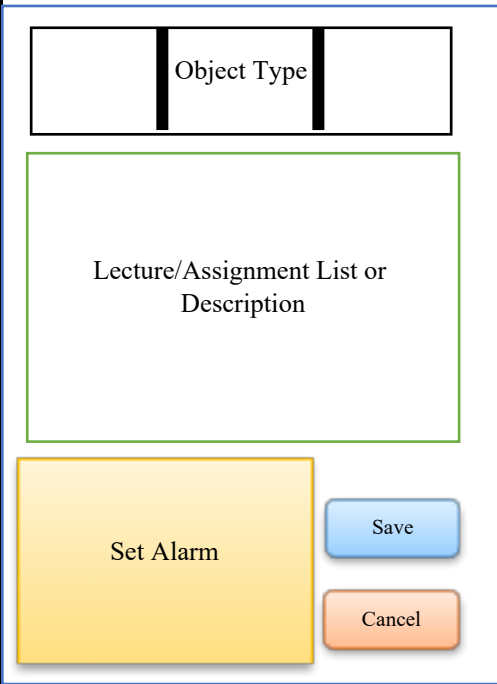
Name	Manage Calendar
Purpose/Description	This interface shows up when the user clicks 'New Calendar' button or one of 'Calendar N's

Name	Manage Calendar
	created already and let the user manage the one's schedule by editing the calendar information. You can find the IDs of team members who are granted with a permission to edit the calendar through Invitations.
Input source/ Output destination	Host server / Client, Client / Host server
Range/Accuracy/ Margin of error	N/A
Unit	Screen
Time/ Velocity	Asynchronous user input/ Communication time between the server and the user device
Relationship with other input/outputs	N/A
Format and configuration of screen	<ol style="list-style-type: none"> 1. The user can traverse through different months of the calendar by clicking two triangle buttons on top of the calendar image. 2. The calendar looks just like a normal calendar we can see in our lives, each day represented as the number of the date. And a grid for each date contains task objects. 3. If you click a number on the calendar and click 'Object' button, this works as adding a task object on that date. 4. If you click a task object in a date grid then click 'Object' button, this works as editing the object information. 5. Description on the contents of a task shows up on 'Description' area when you click the object. 6. New member can be added by clicking '+' symbol on 'Member List' part 
Format and configuration of window	N/A
Data type	Text, Image, Widget
Instruction type	Instruction mapped to the button
Exit message	N/A

[Table 8] User interface of invite team member

Name	Invite Team Member
Purpose/Description	The user can get into this interface by clicking '+' symbol on 'Member List' part of 'Manage Calendar' part, and search for IDs of other users the user wants to add for the member of that calendar.
Input source/ Output destination	Host server / Client Client / Host server
Range/Accuracy/ Margin of error	You can see up to 5 user IDs on 'User List' at once / N/A
Unit	Screen
Time/ Velocity	Asynchronous user input/ Communication time between the server and the user device
Relationship with other input/outputs	N/A
Format and configuration of screen	<p>1. You can fill up 'Keyword' part with some letters contained in the ID of the other user you want to invite and click 'Search' button to see some IDs show up on 'User List'</p> <p>2. You can pick the exact user ID you want to send an invitation to, and comment something on 'Message' part if you want.</p> <p>3. After you click 'Send' button on the bottom, data concerned with your invitation are saved on DB and delivered to the target user when that user is logged-on.</p> 
Format and configuration of window	N/A
Data type	Query, Text, Image, Widget
Instruction type	Instruction mapped to the button
Exit message	N/A"

[Table 9] User interface of manage object

Name	Manage Object
Purpose/Description	The user can get into this interface by clicking 'Object' button from 'Manage Calendar' interface. The user can set the type, contents, alarms for the chosen object
Input source/ Output destination	Host server / Client Client / Host server
Range/Accuracy/ Margin of error	You can see up to 8 user IDs on 'User List' at once / N/A
Unit	Screen
Time/ Velocity	Asynchronous user input/ Communication time between the server and the user device
Relationship with other input/outputs	N/A
Format and configuration of screen	 <ol style="list-style-type: none"> 1. There are 3 types for an object : lecture, assignment, custom for which you can choose on 'Object Type'. If you choose lecture or assignment with a mouse click, a list of items corresponding to your iCampus ID are queried and show up on 'Lecture/Assignment List or Description' part. 2. If you choose custom type instead, 'Lecture/Assignment List or Description' part turns into a blank and some buttons with which you can describe about and set time schedule for the object. 3. After that you can set alarms for the object on 'Set Alarm' part.
Format and configuration of window	N/A
Data type	Query, Text, Image, Widget
Instruction type	Instruction mapped according to the value of a button code
Exit message	N/A

4.1.2. Hardware Interfaces

[Table 10] Hardware interface of applicable device for the system

Name	Applicable device for the system
Purpose/Description	Enable users to take advantage of the service provided by the system/ Window OS Enabled desktop computer and Android OS Enabled Smart Phone. (At least Android 6.0)

4.1.3. Software Interfaces

[Table 11] Software interface of firebase real-time database

Name	Firebase Real-time Database
Purpose/Description	Query input/output for managing multimedia/meta data
Input source/ Output destination	Host server/ User, User/Host server
Range/ Accuracy/ Margin of error	Depends on the performance of the Firebase
Unit	Query
Time/ Velocity	Instant reaction
Relationship with other input/outputs	Related to all inputs/outputs from server
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

[Table 12] Software interface between iCampus and the developed system

Name	iCampus and the developed system
Purpose/Description	To be associated with iCampus and get the information of lectures and assignments related to the user / Query input/output to get such data based on subject code of the lectures
Input source/ Output destination	iCampus server/User, User/ iCampus server
Range/ Accuracy/ Margin of error	Depends on the performance of the iCampus, network environment
Unit	Query
Time/ Velocity	Instant reaction
Relationship with other input/outputs	Related to all inputs/outputs from server
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

4.1.4. Communication Interfaces

[Table 13] Communication interface of client and host

Name	Client and Host
Purpose/Description	<p>1) Each client requests the connection to the host, requesting list of results of user IDs of the system when the user is trying to share a calendar. Host provides a user IDs to the client.</p> <p>2) And if the client chooses a ID from the list and send an invitation to other user, the information is saved into DB and is delivered to that user when the state of the user's account becomes 'logged-on'.</p> <p>3) Host delivers an alarm for an object to the client's mobile phone when it</p>

Name	Client and Host
	comes to the date/time in accordance with the date/time set up for the object.
Input source/ Output destination	Client/Host server
Unit	packet
Time/ Velocity	At least 10Mbps
Relationship with other input/outputs	Related to all inputs/outputs from server
Format and configuration of screen	N/A
Format and configuration of window	N/A
Data type	Query
Instruction type	Query statement
Exit message	N/A

4.2. Functional Requirements

4.2.1. Use Case

[Table 14] Use case of sign-up

Use case name	Register
Actor	Unregistered user
Description	Process when unregistered user tries to sign up to I-campus.
Normal Course	1. Unregistered user tries to log in to I-campus. 2. If the user is not registered, user clicks sign-up button and system redirect user to sign-up page. 3. User enters log-in information and sign up.
Precondition	User is not registered
Postcondition	
Assumptions	

[Table 15] Use case of log-in/out

Use case name	Register
Actor	registered user
Description	Log-in: Process when registered user tries to log-in to I-campus. Log-out: Process when registered user tries to log-out from I-campus.
Normal Course	<Log-in> 1. Registered user tries to log in to I-campus. 2. The user is redirected to log in page. 3. The user enters ID and Password. 4. If correct, user access to I-campus. <Log-out> 1. User clicks the log-out button. 2. System closes the session for the user.
Precondition	<Log-in> The user is registered to I-campus. <Log-out> The user had logged-in.
Postcondition	<Log-in> The user is connected to I-campus by network. <Log-out> All the changes that the user made while he/she logged in are saved.
Assumptions	The user is a student of SKKU.

[Table 16] Use case of Calander

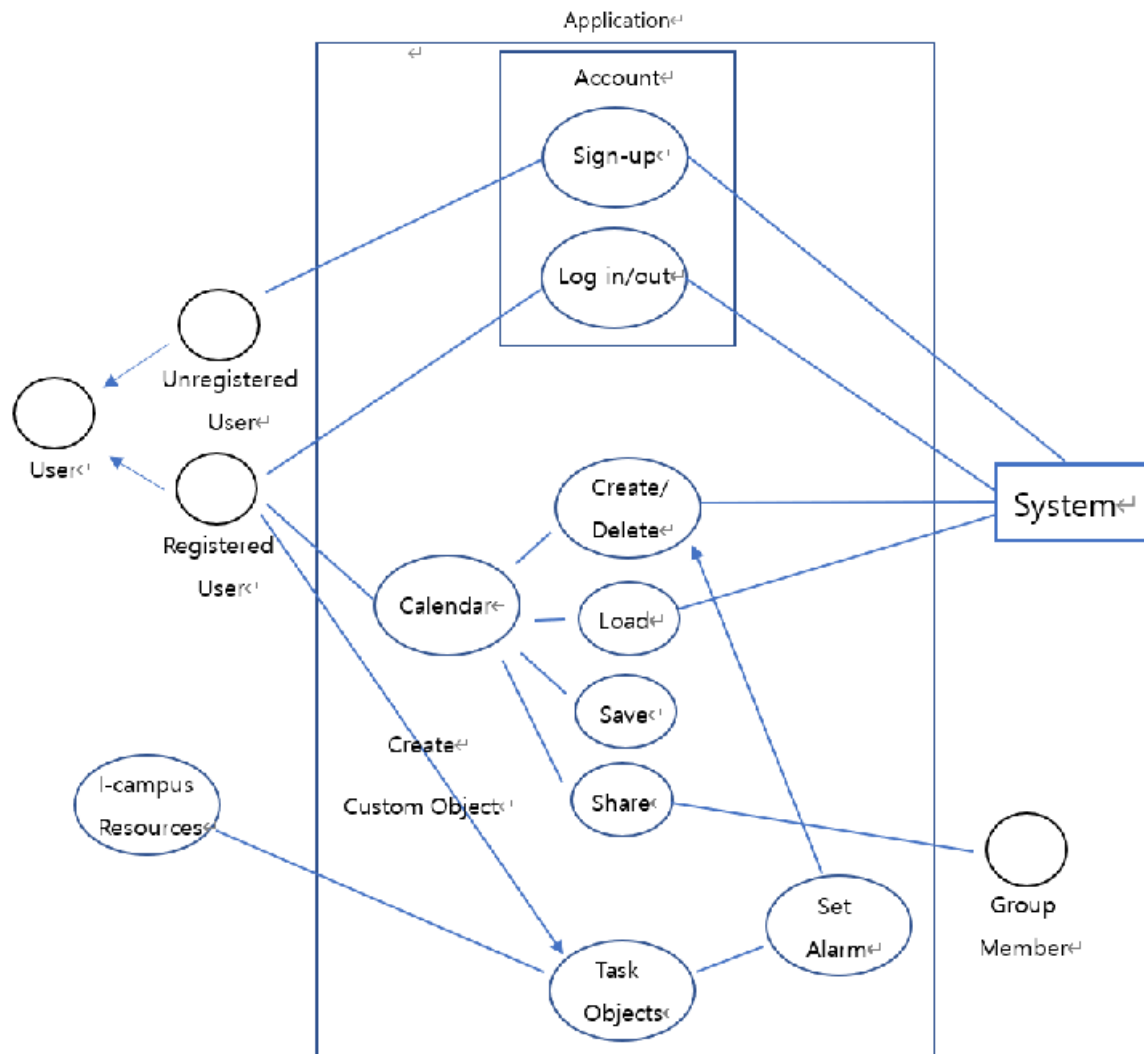
Use case name	Register
Actor	Registered user
Description	Process when registered user creates customized calendar.
Normal Course	1. Unregistered user tries to create/edit his/her own calendar. 2. The user clicks calendar button. 3. System calls new page for calendar. 4. In the page, there are an empty calendar, course list, custom objects, load calendar menu, save calendar menu. 5. In the object list, there is list of courses. 5.1. In the courses, there are clickable&draggable class and assignment objects. 6. User drag&drop the objects in to the calendar or drag some objects out of the calendar to delete the objects. 7. When user wants to just add or delete some objects in existing calendar, the user clicks load calendar button, and the system loads the wanted calendar to the page. 8. When user wants to add some object that are not in course list, he/she clicks custom object button. Then the system calls new page that user can input information of the object. Then system add the object into the current calendar, and the user can drag&drop the new constructed object.

	9. When finished creating/editing calendar, user can save the calendar by clicking save calendar button.
Precondition	The user has done class registration.
Postcondition	User can modify alarm in the way that Object made by user can generate alarm.
Assumptions	The user is a student of SKKU.

[Table 17] Use case of sharing calendar

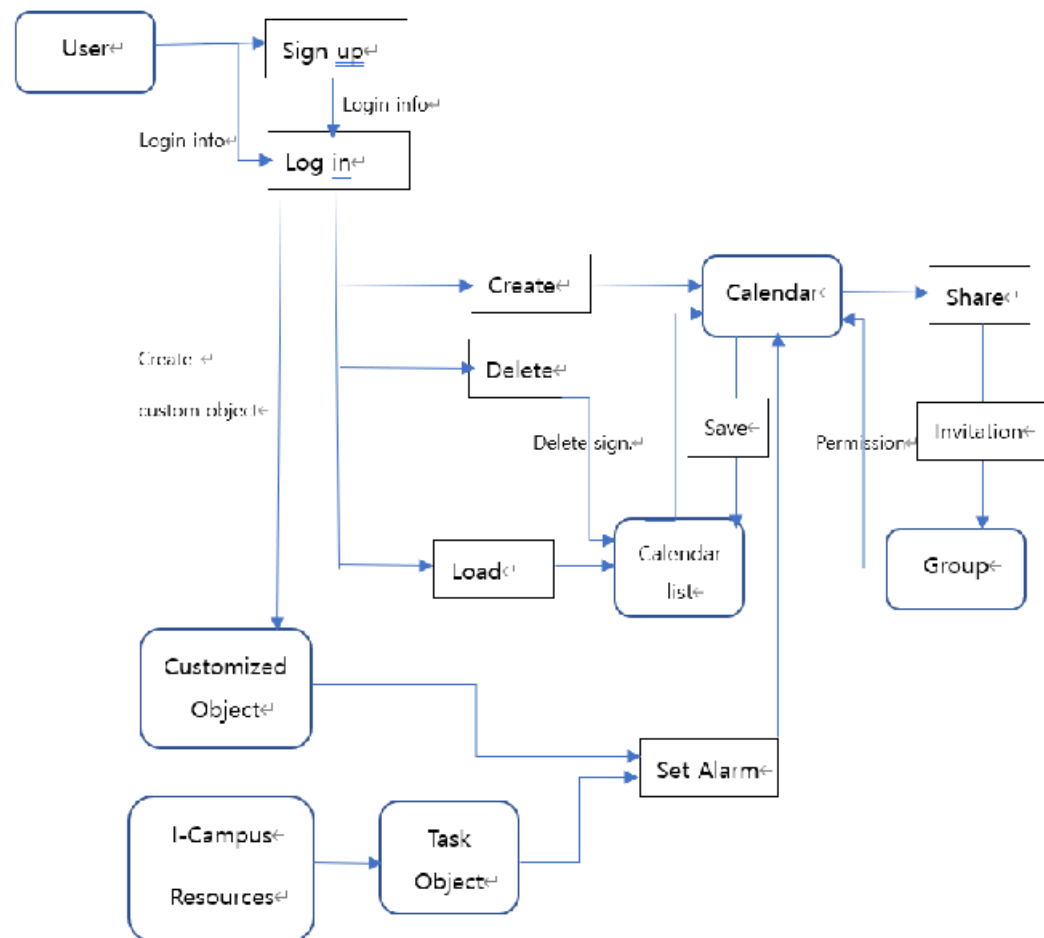
Use case name	Register
Actor	Registered user
Description	Process when registered user tries to share calendar to group.
Normal Course	<ol style="list-style-type: none"> 1. Registered user clicks share button. 2. System calls new page that contains list of groups. 3. User selects the group to share calendar. 4. When group(s) selected, system sends invitation to team members. 5. When the team members accept invitation, calendar is shared. 6. User can check other member's calendar in group menu.
Precondition	The user is in any group.
Postcondition	Permission allowed when user check group member's calendar
Assumptions	The user is a student of SKKU.

3.2.2 Use Case Diagram



[Figure 1] Use case diagram

3.2.3 Data Flow Diagram



[Figure 2] Data flow diagram

3.3 Performance Requirements

The following requirements are based on estimates and may be changed in the final application.

3.3.1 Static numerical requirement

- The system management tool supports only one administrator.

- The system supports only one simultaneous user for each PC. The system is not support multiple connection on the same device. However, user can switch accounts and access after disconnecting.
- The system is should run smoothly on your PC on Windows 7/8/10, Mac OS X 10.6 or more recent version, and to use with Google Chrome or Firefox.

3.3.2 Dynamic numerical requirement

- The system run smoothly for least 200 simultaneous users. And the system can be handles least 10,000 active user account and profile.
- The application should run within 10 seconds.
- Each account must be activated within 5 seconds of linking.
- The personal custom calendar should be displayed within 5 seconds.
- When adding a schedule, the data must be stored on database within 5 seconds, and database updates must be completed within 5 seconds.
- The system should be able to send a calendar sharing request that one user sends to another user within 10 seconds.
- The system must send an exact push notification within 3 seconds of the alarm time specified by the user.

3.4 Logical Database Requirements

The system uses Firebase's database library named 'Firestore' to manage data. It should be able to store information about schedules and alarms set by each user on database. It should be able to store information about groups and group members sharing calendars, and their permissions. These data are retained until the user deletes it directly or leaves the system. Duplicate queries must be processed at high speed, and the constraints and performance of general database systems must be satisfied.

3.5 Design Constraints

The system shall contain only components that can be distributed under the MIT license. The system should be able to access from various PCs and mobile devices with Android operating system, and the administrator must be able to access and manage it through a web browser.

The system must be designed to run on Firebase and use the Firestore database.

3.5.1 Standards compliance

All programs in the system are written in accordance with the JAVA standards, and other matters follow conventional programming techniques. The names of functions and variables in the program use camel case, and underscore notation is applied to databases. System management tool should be access through web browser in accordance with HTML5 standard.

3.6 Software System Characteristics

Software system characteristics are revealed through non-functional requirements. So, this section describes several non-functional requirements of the system. Non-functional requirements are classified Product Requirements, Organization Requirements, External Requirements as follows.

3.6.1 Product Requirements

Product requirements specify or constrain the runtime behavior of the software. Our system should satisfy the following requirements.

3.6.1.1 Usability Requirements

This is one of the most important non-functional requirements of our system. The system should be easy to use by non-technical users and should be organized to minimize user errors. The usage of technical terms or jargons should be minimized, explain it easily if needed. Each user shall be able to use all the system functions instinctively without a learning manual. That means, user interface should be simple, instinctive, and easy to use.

3.6.1.2 Performance Requirements

While communicating with the server, adding and modifying the group calendar is the most time-consuming task in our system. Also, usability degrades as it takes longer time. It must give the result to user 5 seconds or less response time in a Chrome desktop browser.

3.6.1.3 Space Requirements

The system should provide its purposed service at any given time. The system should provide the service while minimizing errors. Its result should be reliable as expected by the user. For

each schedule object, space is needed to store date, content, alarm information, and function (class, assignment, other custom), and for group calendars, space is needed to store permission setting information for each member.

3.6.1.4 Security Requirements

The users should be properly authenticated before using the system. It should make sure that an unauthorized user cannot gain access as a system manager and make system unavailable. Also, an unauthorized user cannot gain access as system user to confidential information such as user's personal Information, ID, and password. In addition, in a group calendar, users who do not obtain read permission cannot access the calendar, and even if they do, users who do not obtain write permission cannot add or delete a new schedule object in the calendar.

3.6.2 Organizational Requirements

These requirements are broad system requirements derived from policies and procedures in the customer's and developer's organizations

3.6.2.1 Environmental Requirements

The system's class objects and assignment Objects, which are used as schedule objects come from and rely on 'i-Campus'. They are already keeping enormous amount of schedule information. In the system, even if a user changes the content of a schedule object personally within this calendar, the class's or assignment's title does not change on the actual i-Campus system.

3.6.2.2 Operational Requirement

Users of the system are probably students who take classes using the i-Campus web site. The system will be developed and operated by we team 5. The system read the data from the DB after the user log-in and shows the result with a user-friendly calendar style graphical UI, as the user has arranged. It shows the result within 5 seconds. The system works as web-application in. PC and work at any-time. Also, the alarm function works linked with LearningX application in smartphone.

3.6.3 External Requirements

It covers all requirements that are derived from factors external to the system and its development process.

3.6.3.1 Safety / Security Requirement

The system should ensure that any personal information is not accessed by external system. The system should be safe enough to prevent user's data from damaging from external natural disaster.

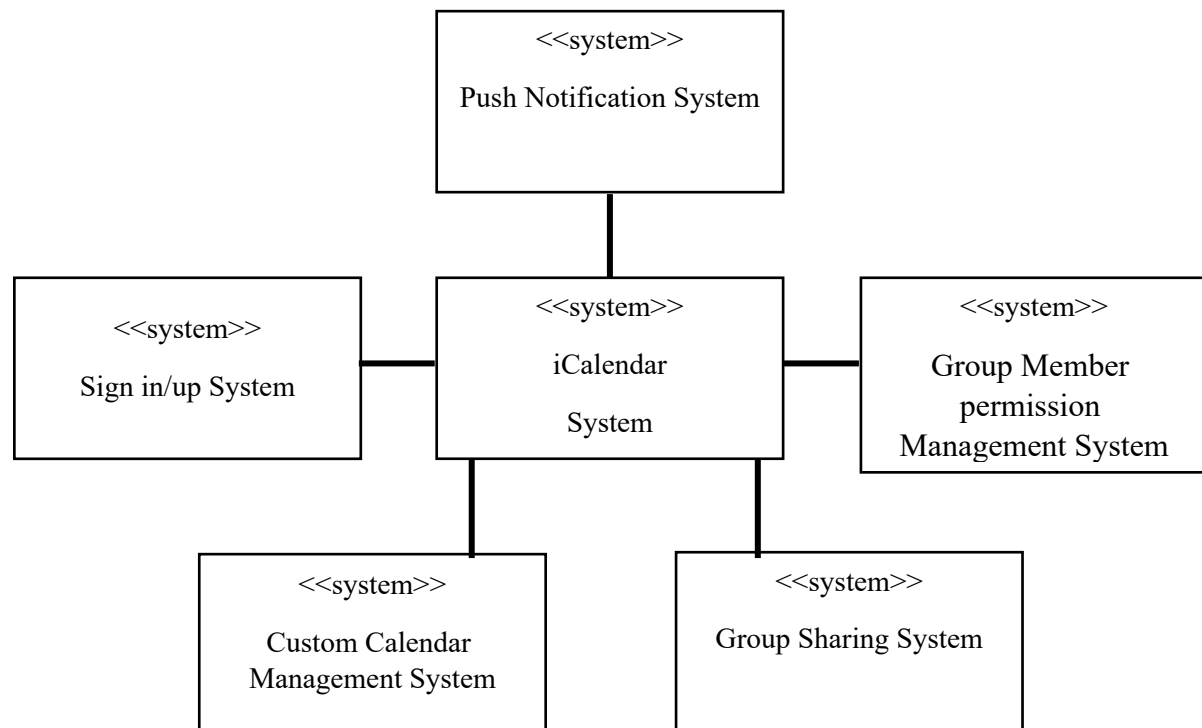
3.6.3.2 Regulatory Requirement

The calendar we are going to use is based on the learning management platform called Canvas from Instruction Corporation. Therefore, permission should be given from Instructure Corp. before commercially used. User's privacy should not be violated according to law. The system should be developed in accordance with a national Privacy standard.

3.7 Organizing the Specific Requirements

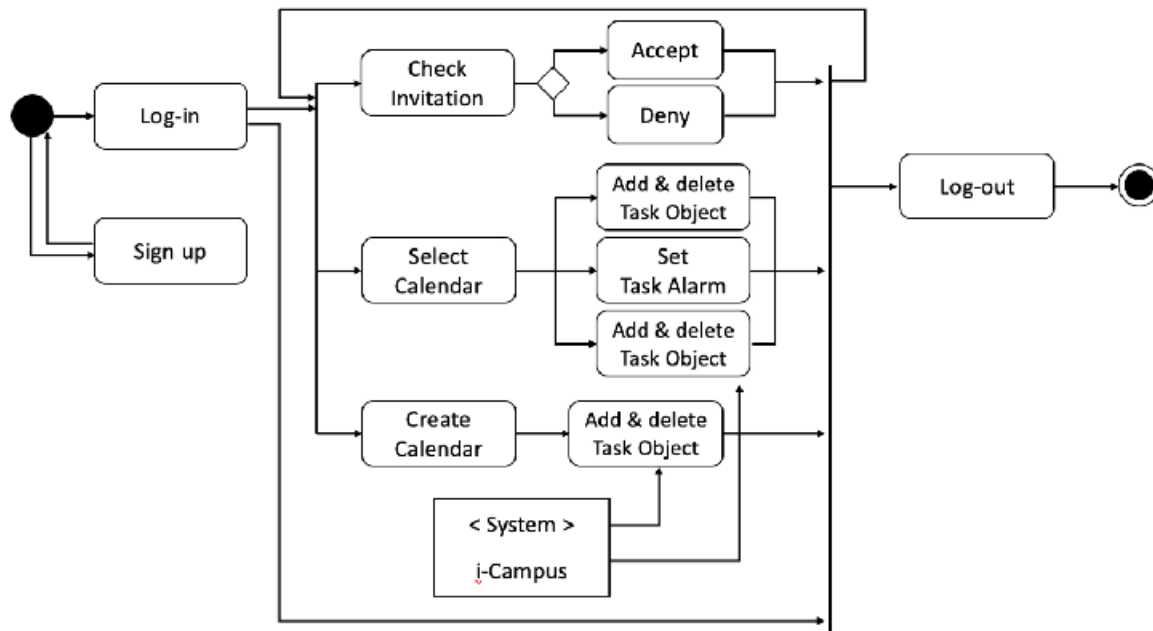
In this section, we describe the system model using graphical notation based on Unified Modeling Language (UML) and tabular form. The system model describes the relationship among the system, sub-systems, components, and surrounding environments, showing more specific requirements.

3.7.1 Context Model



[Figure 3] Context model

3.7.2 Process Model



[Figure 4] Overall process model diagram

3.7.3 Interaction Model

See 3.2.2. Use Case Diagram

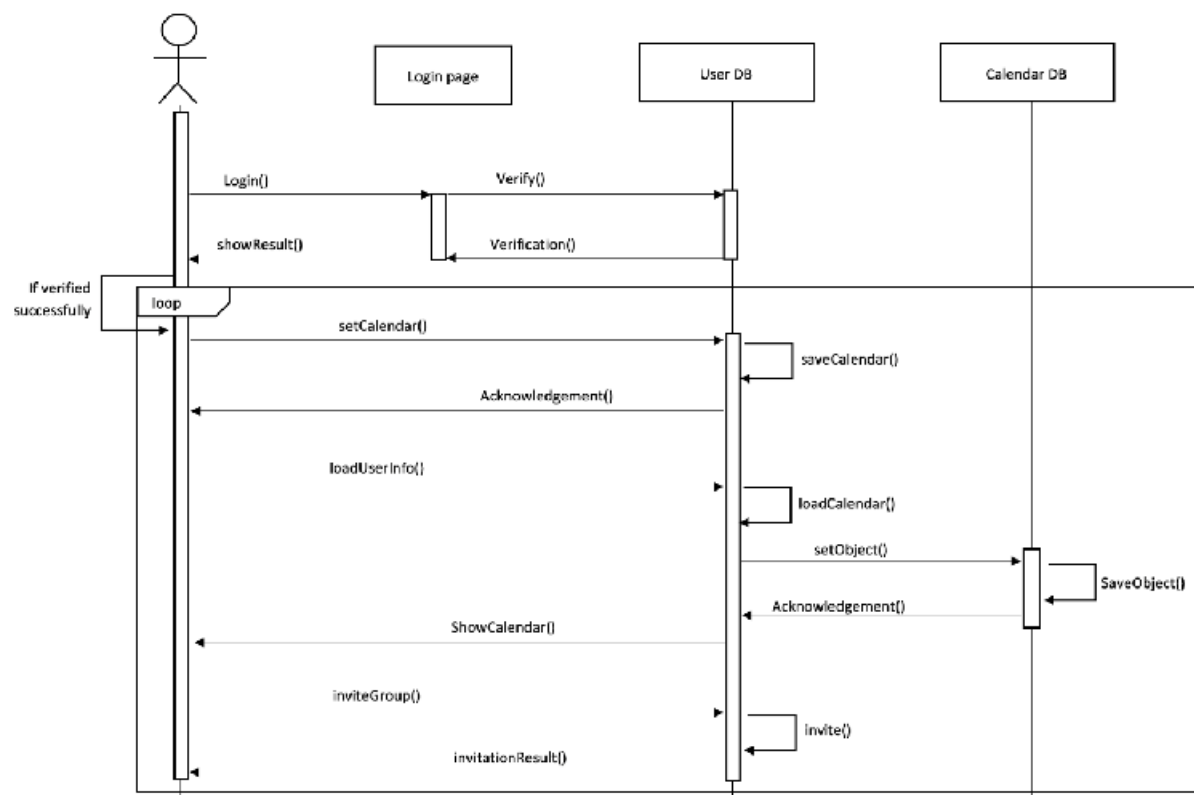
3.7.4 Behavior Model

3.7.4.1 Data Flow Diagram

See 3.2.4. Data Flow Diagram

3.7.4.2 Sequence Diagram

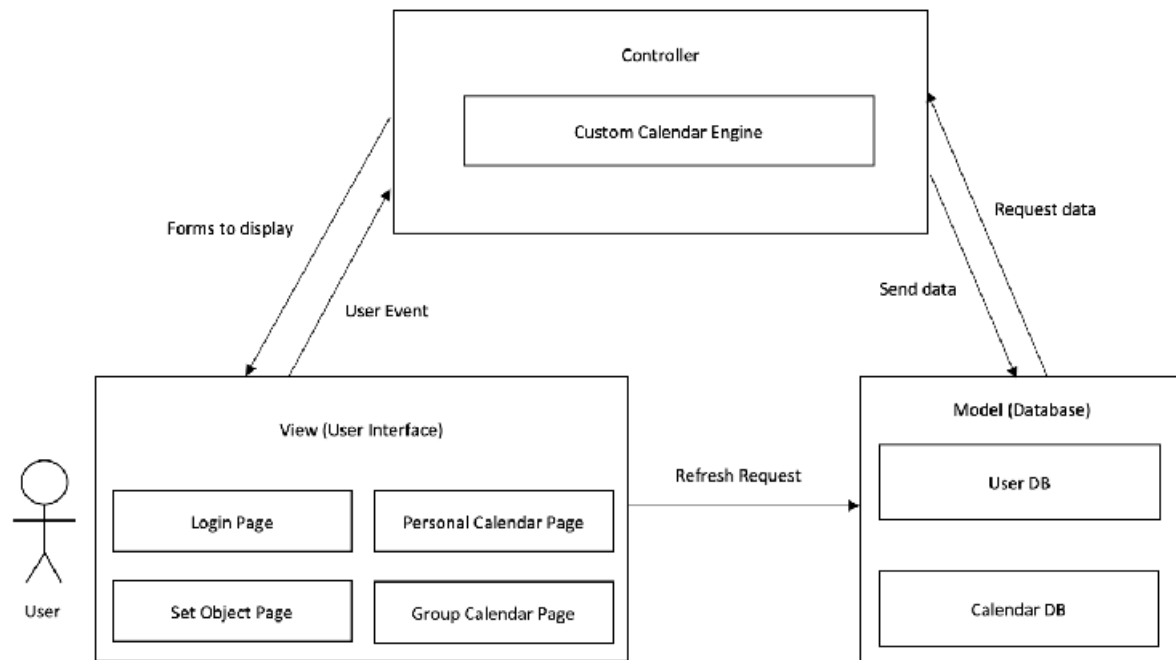
It describes mainly the sequence of processing recommendation which is our main system.



[Figure 5] Sequence diagram

3.8 System Architecture

This Section presents a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. We use MVC pattern as graphical models of the architecture.



[Figure 6] System architecture of the system

3.9 System Evolution

In this section, we describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.

3.9.1 Limitation and Assumption

The system currently assumes that there are only three types of objects in the custom calendar: class, assignment, and other custom objects. Group sharing calendar allows users who receive invite to look the calendar, and users who also receive modify permission can add, modify, or delete the objects. In addition, it is assumed that calendars can be performed on PC websites, and the mobile device can only send an alarm push linked the Learning X application.

3.9.2 Evolutions of Hardware and Change of User Requirements

Now, we assume that objects are classified into only three objects, but we expect that quiz or test objects may be added later. Also, although the current system allows calendar modification only on PCs, it would be better to prepare for work in mobile applications in the future.

Currently, most lectures are online, so classes and assignments are registered in i-Campus. However, if the classes are switched to offline classes later, our system will likely make users more bothered because lectures or assignments will not be automatically uploaded. Therefore, in order to maintain a convenient system, we must be prepared for this change of circumstances. For example, class objects can be registered automatically according to the user's timetable.

5. Supporting Information

5.1. Software Requirement Specification

This software requirements specification was written in accordance with the IEEE Recommendation (IEEE Recommended Practice for Software Requirements Specifications, IEEE-Std-830).

5.2. Document History

[Table 18] Document History

Date	Version	Description	Writer
2021/04/12	0.1	Style and overview	Suyoung Min
2021/04/13	1.0	Addition of 2.3, 2.4, 2.5	Taewoo Yoo
2021/04/17	1.1	Addition of 3.1	Chanjong Lee
2021/04/17	1.2	Addition of 3.7	Minseo Kim
2021/04/17	1.3	Addition of 1.1, 1.2, 1.3	Suyoung Min
2021/04/18	1.4	Addition of 3.3, 3.4, 3.5, 3.6	Minseo Kim
2021/04/18	1.5	Addition of 2.1, 2.2, 2.3	Taewoo Yoo
2021/04/20	1.6	Addition of 3.2	Sumin Ham

Date	Version	Description	Writer
2021/04/21	1.7	Revision of 1.1, 2.2 and style	Suyoung Min
2021/04/22	1.8	Revision of 3.2	Sumin Ham
2021/04/22	1.9	Revision of 2.3	Taewoo Yoo
2021/04/22	1.10	Revision of 3.1	Chanjong Lee
2021/04/23	1.11	Revision of 3	Chanjong Lee
2021/04/24	1.12	Addition of 3.8, 3.9	Minseo Kim