Final Report

Team members:
17102047 Jihoon Moon, jihoonmoon@g.seoultech.ac.kr
18102085 Byungchan Lee, 18102085@seoultech.ac.kr
18102087 Yeongju Lee yg2748@seoultech.ac.kr

Contents

Abstract

<itmWiki> is an application that allows communication between the ITM students to share their tacit knowledge like a technology blog. In this report, we propose the reasons why we decide to design this software, and several components for design. There are 56 functional requirements, 11 non-functional requirements based on the category. In the functional requirements, all actions that the user and the administrator can do are defined. Non-functional requirements include background requirements such as security or database management. The scenario indicates how both users, and the administrator use our application. Finally, we design UML diagrams, user interface, database design including use-case diagrams, class diagrams, and sequence diagrams.

I-A. The necessity of the term project

ITM students spend a lot of time searching for relevant content and reading books when studying their major and each time they carry out an assigned assignment or project. What makes us more difficult is the scattered information and the task of making sure the information we are looking for is correct. Our project focuses on this problem. It will help ITM students solve their difficulties and will make students exchange their opinions easily.

I-B. The goal of the term project

Our goal is to contribute to the continued growth of the ITM department. It will lead to fair competition by breaking the harmful effects of information inequality caused by data from individuals to individuals. In addition, this service will promote overall upward leveling of students by improving learning accessibility based on high-quality course-based learning materials. As students' understanding increases every year, we expect that professors also continuously add diversity and depth to lectures.

I-C. Problem description

There are a few departments with a fixed major curriculum, such as the ITM department. In this department, each student can deliver various and high-quality knowledge to junior students by storing the knowledge that each student understands through the professor's classes and additional studies in a specific space to enhance and help other students. However, in the current situation where there is no channel to transmit knowledge, we are trying to design software such as a technology blog that reflects the characteristics of these departments (fixed major curriculum). Based on the ITM department, our software consists of feeds for each grade level’s subjects and feeds for internal and external activities related to the ITM department. Users can obtain or deliver information related to the feed. In addition, they can have a discussion about a specific topic by commenting on the thread, which corresponds to the post.

Our software users divide into 3 main types: First, a user who wants to write by changing tacit knowledge to explicit knowledge. Second, users who want to obtain information about knowledge and subjects they are curious about and in this process, it is divided into users who obtain or give information through discussion and modification. Third, the administrator who manages this software in terms of a policy and the curriculum.

The first user

Suppose there is a 3rd year ITM student taking a software engineering class. The student studied Design Concept through the class, but it was difficult to understand at once due to the characteristics of the ITM department, which is teaching in English, and the relatively simple lecture materials. However, the student was personally interested in it, and in preparation for the exam, he was able to understand it clearly by searching Google, papers, and books, and made the material organized enough to explain it to other students. In order to meet the purpose of our software and to help other ITM students, he uploaded the materials that he had organized to the 3rd year, ‘Software engineering’ feed such as concepts and additional knowledge. Posting specific examination or assignment content is prohibited.

The second user

Suppose there is another 3rd year ITM student taking a software engineering class. Like the first user, she studied Design Concept, but it was difficult to understand only with class and lecture materials. To solve this problem, she finds posts related to Design Concept in our software and uses them as supplementary materials for understanding. Because it is explained by solving the other student's understanding, the second user can learn it with good quality information without spending a lot of time. In addition, she is able to send a direct message, if she does not fully understand or has some ambiguous concepts. There are some cases that can be applied to both the first user and the second user. To explain further, the first user uploaded a post about Design Concept, and the second user got information from the post. The second user can ask a question using comments through the thread because he or she has a question, and the first user can communicate through the thread. In addition, all users using the software have the right to freely modify the contents of the post, so they can correct incorrect or updated information at any time. However, to prevent indiscriminate modification, the modification history is saved along with the real name of the person who modified it. Through this, the information on the corresponding post is constantly updated to deliver the latest information at all times. Lastly, each user can store a post that he/she is interested in as a bookmark and can recommend the post by clicking the “like button” in each post. Each post can be ordered in the number of recommendations on the feed.

Administration

Firstly, there are some accounts for administrators. Suppose there is a staff member who is working at the ITM department. She manages overall works related to the ITM department. In this case, she will get the administration account. Even though it might rarely happen, if the ITM department curriculum is changed, she can update the curriculum tree in our software. To prevent people ,who do not belong to the ITM department, from registering, all people should get an authentication from her (Administrator) when they are trying to register. Lastly, users can report a specific post due to some reasons such as slandering, promotion, or pornography. It will be removed by the administrator based on the number of reports.

Extra

The first user wants to upload a post about Design Concept to the Software engineering feed, but another student has already uploaded a post about Design Concept. However, there are no restrictions on uploading posts for duplicate content because there are different parts or styles depending on the author even if the article is on the same topic.

II. Functional Requirements

Number	Category	Requirement Name
FR-A-01	Account	Register
FR-A-02		Login
FR-A-03		Logout
FR-A-04		Find Password
FR-A-05		Automatic Login
FR-M-01	MyPage	Go to my page
FR-M-02		Go to my post
FR-M-03		Go to my bookmark
FR-M-04		Go to my log
FR-M-05		Go to setting
FR-M-06		Set a notification
FR-M-07		Edit my profile
FR-M-08		Check software version
FR-MP-01	Main Page - Post	Write a post
FR-MP-02		Pop up the regulation
FR-MP-03		Go back to the feed from working
FR-MP-04		Upload a post

FR-MP-05		Store a writing post
FR-MP-06		Store on my post list
FR-MP-07		Update the post contents
FR-MP-08		Store Logs
FR-MP-09		Go to the post
FR-MP-10		Store a post as bookmark
FR-MP-11		Recommend a post
FR-MP-12		Report a post
FR-MP-13		Show posts
FR-MP-14		Show the number of recommendations
FR-MP-15		Filter Post
FR-MC-01	Main Page - Comment	Go to the thread
FR-MC-02		Create a comment
FR-MC-03		Update a comment
FR-MC-04		Report a comment
FR-MC-05		Delete a comment
FR-MC-06		Show a comment
FR-T-01	Tree	Show a tree
FR-T-02		Show a specific subject feed by clicking a node
FR-T-03		Go to a specific feed by clicking a node
FR-S-01	Search	Search a post within all feed
FR-S-02		Search a post within a specific feed

FR-Ad-01	Administration	Login as an administrator with an administration account
FR-Ad-02		Log out as an administrator
FR-Ad-03		Authorize
FR-Ad-04		Create a node in the tree
FR-Ad-05		Update a node in the tree
FR-Ad-06		Delete a node in the tree
FR-Ad-07		Delete a post
FR-N-01	Notification	Make push notification
FR-N-02		Click the notification
FR-N-03		Delete push notification
FR-DM-01	Direct-Message	Go to direct message
FR-DM-02		Read a message
FR-DM-03		Send a message
FR-DM-04		Delete a chat room
FR-DM-05		Search user
FR-DM-06		Send a message via profile

Account Requirements	
FR-A-01	
Function Name	Register
Specification	Sign up as a member (Registration)
Input data	user ID, Password, Name, Phone number, E-mail address, Identification Certification
Output data	Authentication request and email

FR-A-02	
Function Name	Login
Specification	Log in to the software
Input data	Registered student ID and password
Output data	Access token
FR-A-03	
Function Name	Logout from iTMwiki
Specification	Logout from the software
Input data	Access token
Output data	-
FR-A-04	
Function Name	Find password
Specification	Find password
Input data	Private information such as phone number, name, etc
Output data	changed password information
FR-A-05	
Function Name	Automatic Login
Specification	Login to the software automatically
Input data	Registered student ID and password, Automatic Login checkbox status
Output data	Access token
Mypage Requirements	
FR-M-01	
Function Name	Go to my page

Specification	Go to my page section. It consists of several functions such as my post, bookmark, log, setting
Input data	Access token
Output data	My page
FR-M-02	
Function Name	Go to my post
Specification	Go to my posts list. It consists of posts that the user has uploaded
Input data	Access token
Output data	My Post page
FR-M-03	
Function Name	Go to my bookmark
Specification	Go to my bookmark list. It consists of posts that the user has stored
Input data	Access token
Output data	My bookmark page
FR-M-04	
Function Name	Go to my log
Specification	Go to my log list. It consists of logs that the user has done
Input data	Access token
Output data	My log page
FR-M-05	
Function Name	Go to setting
Specification	Go to setting to set application configuration

Input data	Access token
Output data	My setting page
FR-M-06	
Function Name	Set a notification
Specification	Set the notification whether the user gets the notification or not
Input data	checkbox status
Output data	checkbox status
FR-M-07	
Function Name	Edit my profile
Specification	Edit my profile
Input data	Contents that will be edited
Output data	Edited profile
FR-M-08	
Function Name	Check software version
Specification	Check software version
Input data	Version status
Output data	Version status
Mainpage-Post requirement	
FR-MP-01	
Function Name	Write a post
Specification	Write a post on a specific feed
Input data	Access token
Output data	Regulation
FR-MP-02	

Function Name	Pop up the regulation
Specification	Show the regulation for writing and get agreement. Regulation: 1. Anyone can revise this post. 2. Avoid excessive copying and pasting. 3. Reference must be represented. 4. Posting specific contents related to the examination and assignment must be prohibited. 5. Legal action may be taken for posts containing slander, profanity, promotion, or obscene content.
Input data	Regulation
Output data	Agreement
FR-MP-03	
Function Name	Go back to the feed from writing
Specification	If the user disagrees about the regulation, then go back to the feed
Input data	Disagreement
Output data	Feed, Access token
FR-MP-04	
Function Name	Upload a post
Specification	After agreement, the user write contents and upload a post on the feed
Input data	Contents
Output data	Post
FR-MP-05	

Function Name	Store a writing post
Specification	Store a post that the user is writing
Input data	Post
Output data	Post
FR-MP-06	
Function Name	Store on my post list
Specification	After uploading, the post is stored on my post list
Input data	Post
Output data	Post
FR-MP-07	
Function Name	Update the post contents
Specification	Update the post contents
Input data	Access token
Output data	Regulation
FR-MP-08	
Function Name	Store Logs
Specification	When the user do some actions, logs are stored on my log page
Input data	Log content
Output data	Log content
FR-MP-09	
Function Name	Go to the post
Specification	Go to the post when the user clicks my logs, my bookmarks, my notification, search, or my post list

Input data	-
Output data	Specific posts
FR-MP-10	
Function Name	Store a post as bookmark
Specification	Store a post on my bookmark list
Input data	Post
Output data	Post in my bookmark list
FR-MP-11	
Function Name	Recommend a post
Specification	Recommend a post by clicking a button
Input data	Access token
Output data	1 is added to the number of recommendations
FR-MP-12	
Function Name	Report a post
Specification	Report a post by selecting a reason among slandering, promotion, or inappropriate post
Input data	Access token
Output data	Report token
FR-MP-13	
Function Name	Show posts
Specification	Show posts uploaded on the feed
Input data	Access token
Output data	Posts
FR-MP-14	

Function Name	Show the number of recommendations
Specification	Show the number of recommendations of the post
Input data	-
Output data	the number of recommendations of the post
FR-MP-15	
Function Name	Filter Post
Specification	Posts are shown in New or recommendations order
Input data	Conditions
Output data	Ordered Post
Mainpage-Comment Requirement	
FR-MC-01	
Function Name	Go to the thread
Specification	Go to the thread to comment
Input data	Access token
Output data	Thread
FR-MC-02	
Function Name	Create a comment
Specification	Make a comment for asking or discussion on the thread or reply
Input data	Comment
Output data	Post that reflected the comment
FR-MC-03	
Function Name	Update a comment

Specification	Update a comment on the thread
Input data	Comment
Output data	Updated comment
FR-MC-04	
Function Name	Report a comment
Specification	Report a comment by selecting a reason among slandering, promotion, or inappropriate post
Input data	Access token
Output data	Report token
FR-MC-05	
Function Name	Delete a comment
Specification	Delete a comment on the thread
Input data	Access token
Output data	-
FR-MC-06	
Function Name	Show a comment
Specification	Show a comment in the thread
Input data	The contents of comment
Output data	The contents of comment
Tree Requirement	
FR-T-01	
Function Name	Show a tree
Specification	Show a curriculum tree or activity tree at the ITM department
Input data	Access token

Output data	Curriculum or activity tree
FR-T-02	
Function Name	Show a specific subject feed by clicking a node
Specification	Show the specific subject feed or activity feed
Input data	Access token
Output data	Feed
FR-T-03	
Function Name	Go to a specific feed by clicking a node
Specification	Go to a specific subject feed or activity feed
Input data	Access token
Output data	Subject or activity feed
FR-T-04	
Function Name	Delete a post
Specification	Delete a post that has been reported several times with inappropriate reasons
Input data	Reported posts
Output data	-
Search Requirement	
FR-S-01	
Function Name	Search a post within all feed
Specification	Search a post within all feed with keyword in title or contents
Input data	Keyword
Output data	Post list

FR-S-02	
Function Name	Search a post within a specific feed
Specification	Search a post within a specific feed with keyword in title or contents
Input data	Keyword
Output data	Post list
Administration Requirement	
FR-Ad-01	
Function Name	Login as an administrator with an administration account
Specification	Login as the administrator.. Staff (without professor) belonging to the ITM department can be the administrator
Input data	Administrator ID and password
Output data	Access token
FR-Ad-02	
Function Name	Log out as an administrator
Specification	Log out as an administrator
Input data	Access token
Output data	-
FR-Ad-03	
Function Name	Authorize
Specification	Authorize the user (students or professors) who wants to join the iTMwiki
Input data	User information

Output data	Access token for the user and send an email to the user
FR-Ad-04	
Function Name	Create a node in the tree
Specification	Create the node in the tree
Input data	Access token
Output data	Create the node
FR-Ad-05	
Function Name	Update a node in the tree
Specification	Update the node in the tree
Input data	Access token
Output data	
FR-Ad-06	
Function Name	Delete a node in the tree
Specification	Delete the node in the tree
Input data	Access token
Output data	Delete the node
Notification Requirement	
FR-N-01	
Function Name	Make push notification
Specification	Show the notification to the user's phone
Input data	Reported page, post, or direct message
Output data	-
FR-N-02	
Function Name	Click the notification

Specification	Go to the corresponded page, post or direct message
Input data	Access token
Output data	corresponded page, post or direct message
FR-N-03	
Function Name	Delete push notification
Specification	Delete the notification to the user’s phone
Input data	Notification
Output data	-
Direct-Message Requirement	
FR-DM-01	
Function Name	Go to direct message
Specification	It shows a list of other users that have communicated so far
Input data	Access token
Output data	Direct message page
FR-DM-02	
Function Name	Read a message
Specification	Read a direct message
Input data	Chat contents
Output data	Chat contents
FR-DM-03	
Function Name	Send a message
Specification	Send a direct message
Input data	Access token

Output data	Chat room
FR-DM-04	
Function Name	Delete a chat room
Specification	Delete a chat room
Input data	Access token
Output data	-
FR-DM-05	
Function Name	Search user
Specification	Search username for DM to send a direct message
Input data	Access token
Output data	Chat room
FR-DM-06	
Function Name	Send a message via profile
Specification	Send a message via profile
Input data	Access token
Output data	Chat room

III. Nonfunctional Requirements

Number	Category	Requirement Name	
NFR-1	Product Requirement	Usability Requirement	
NFR-2		Portability Requirement	
NFR-3		Reliabilty Requirement	
NFR-4		Efficiency Requirements	Performance Requirement
NFR-5			Space Requirement
NFR-6	Organizational Requirement	Implementation Requirement	
NFR-7		Database Management	
NFR-8	External Requirement	Interoperability Requirement	
NFR-9		Ethical Requirement	
NFR-10		Legislative	Privacy Requirement
NFR-11			Safety Requirement

Product Requirement	
NFR-P-01	
Requirement Name	**Usability Requirement**
Specification	- Korean and English will be provided as the application default language. - A Wi-fi connection or mobile data connection are needed.

	- The interface should be easy to use for those who use this application for the first time. - Function will be available 24/7.
NFR-P-02	
Requirement Name	**Portability Requirement**
Specification	- It can be operated on IOS, Android and the web. - Whenever it is updated, automatic update is supported
NFR-P-03	
Requirement Name	**Reliability Requirement**
Specification	- Each time an error occurs, an error is reported to the administrator, and the log is used to return it to the state it was before the error occurred.
Product Requirement - Efficiency Requirements	
NFR-PE-04	
Requirement Name	**Performance Requirement**
Specification	-Since we pursue concurrency execution, quick responsiveness and quick processing time should be guaranteed in less than a second.
NFR-PE-05	
Requirement Name	**Space Requirement**

Specification	-Because the application uses a method of storing logs, a lot of memory is used. In order to overcome this, the log after 3 years will be automatically deleted.
Organizational requirements	
NFR-O-06	
Requirement Name	**Implementation Requirement**
Specification	- Using Kotlin, Swift and Node.js to develop applications for Android, IOS ,and Web. -The server will be distributed through AWS EC2.
NFR-O-07	
Requirement Name	**Database Requirement**
Specification	-The user information, log, post are managed in the DB - to make it accessible from anywhere to DB, AWS RDS is used. -There should not be redundancies and omissions.
External requirements	
NFR-E-08	
Requirement Name	**Interoperability Requirement**
Specification	-The software can be run on the web and smartphone at the same time

NFR-E-09	
Requirement Name	**Ethical Requirement**
Specification	-Do not slander and recommend discussion -When writing a post, simple copy and paste is restricted. -Reference should be shown.
NFR-E-10	
Requirement Name	**Privacy Requirement**
Specification	-The personal information can't be used for another purpose. -The personal information could be used under the consent of the user.
NFR-E-11	
Requirement Name	**Safety Requirement**
Specification	-User's information should not be violated by a malicious user. -To guarantee a user a private account, there is only access through login with a student number or professor number. -To prevent loss of information due to errors, all changes are saved as a log.

IV. Scenario

SC-01

Registration	
Actors	User, Administrator
Initial condition	- Mobile devices such as smartphone should be accessible to the Internet network - Applications should be downloaded on a user's mobile device such as smartphones or tablets - User has no account
Final condition	- User get student or professor account
Exceptional case	- The requests are rejected by the administrator - uploaded image inaccurate Additional documents are asked - In the administrator's case, the account is created by the developer in advance. They don't need registration.
Process	1. In login page, click the register button 2. Register page is opened 3. Put the all the required information (name, student ID/Professor ID, phone number, password, email) 4. User is asked to take a picture of his student id card / certificate of studentship / teacher system certificate 5. User clicks the 'sign up' button 6. App returns Welcome and Waiting message in form of pop-up 7. The request is verified under 24 hours 8. The account will be created and added to the database 9. An email of welcome will be sent

SC-02

Log in

Actors	User
Initial condition	- User's account is verified - Devices should be accessible to the network.
Final condition	1. The application should be executed 2. User should have an app account
Exceptional case	- If the ID / password is incorrect, 'try again: please check ID / password' message pops up - In these situations, the process returns to the first step again - If the user's registration information fails to transfer to the server, 'login failed: fill again' message pops up. - If the network is unstable, 're-run app' message pops up.
Process	1. User enters the log-in page 2. User inputs his id and password 3. User can choose whether to use the 'automatic log-in' function or not. 4. User clicks the 'Log in' button 5. Server confirms that the account is approved then the app returns the Tree page 6. If the user didn't check the 'automatic log-in', the automatic log in function can't be configured. So, user have to take 1~5 steps to execute the app again 7. After user takes this log in process at first with choosing 'automatic log-in', user doesn't need to 1~5 step of Login until user will log out

SC-03

Log out

Actors	User
Initial condition	- User is logged in the application - Mobile devices should be accessible to the network
Final condition	- User is logged out
Exceptional case	- In these situations, the process returns to the first step again - If the network is unstable, 'try again: network is unstable' message pops up - If the user clicks the 'cancel' button of the logout popup message - If the user selects automatic login, it is not usually logged out, but the token expires after a week so is logged out.
Process	1. User clicks the 'logout' button at the top of the Mypage 2. 'Want to log out?' message pops up, with a 'confirm' and 'cancel' button 3. User clicks the 'confirm' button 3. Log out process is completed, and the app returns the initial login page

SC-04

Find password	
Actors	User
Initial condition	- Mobile devices should be accessible to the network
Final condition	- User get the temporary password via their mail
Exceptional case	- If the network is unstable, 'try again: network is unstable' message pops up - If the matched email address cannot be found, pop up the warning message "we cannot find the email address. Please re-enter.
Process	1. User clicks the 'find password' button in the login page. 2. User enter the student ID and email address according to the form 3. User clicks the 'confirm' button 4. Password-related email is sent 5. The app returns the initial login page

SC-05

Automatic Login	
Actors	User
Initial condition	User's account is verified Devices should be accessible to the network.
Final condition	1. The application should be executed
Exceptional case	- If the network is unstable, 're-run app' message pops up and the process returns to the first step again

	- If the user id and password is incorrect, user can't enter the app and 'ID or Password is incorrect' message is popped up
Process	1. User enter the user_id and password 2. Automatic logins checkbox is checked 3. After the checkbox is checked, whenever the user enters the app, he can enter the app without login process

SC-06

Authorize	
Actors	Administrator
Initial condition	User has been registered and has not received approval Devices should be accessible to the network
Final condition	1. User can enter the app
Exceptional case	- If the network is unstable, 're-run app' message pops up - If the user is not our school member, the admin clicks the disapproval button
Process	1. In the certification page, administrator check the identification of users 2. If the user is our school member, the admin clicks the approval button.

SC-07

Set Notifications	
Actors	User
Initial condition	- User is in Mypage - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- User can maintain the updated notification setting
Exceptional case	- If the network is unstable, ‘try again: network is unstable‘ message pops up and Mypage is shown again
Process	1. User clicks ‘Setting’ ’textview in the Mypage 2. User change the checkbox status either receiving notification or not 3. The database is updated 4. The app returns the Mypage

SC-08

Edit Profile	
Actors	User
Initial condition	- User’s characteristics have changed - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- Edited personal information is stored in database - Whenever user enter the My page, user can see the edited my information
Exceptional case	- In these situations, the process returns to the first

	step again ➔ If the network is unstable, 'try again: network is unstable' message pops up ➔ If the user clicks the 'cancel' button
Process	1. User clicks the 'edit' button of the My Info field at the Mypage 2. User modify information that he wants to update - User can modify their biography - User can change their profile photo 3. When finished, the user clicks on 'Save' button 4. The database is updated 5. User's 'updated My info' in 'MyPage' is shown

SC-09

Write Post	
Actors	User
Initial condition	- User is in the specific subject's feed page - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- User can upload and see posts in the feed
Exceptional case	- If the network is unstable, 'try again: network is unstable' message pops up and curriculum flow chart page is shown again - In the regulation pop-up, if the user clicks the disagree button, the feed is shown.
Process	1. User clicks the create post button in the form of an oval, a regulation message,which is 'Anyone can edit what you wrote.', 'Avoid excessive copying and

	pasting.', 'Specifying the test and assignments is prohibited.', 'We recommend you leave a reference.', and 'Posts containing slander, abuse and obscene content may be subject to legal action.', is popped up. 2. User clicks the agree button. 3. User enters the title and contents. In this step, the user can make bold and italic text, add a cancel line, appendices, and code. 4. Click the save button. 5. The post is uploaded to the DB 6. The new post-related log is created 7. User's uploaded post is shown

SC-10

Update Post	
Actors	User
Initial condition	- User is in post - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- The contents added by the users are reflected in the post
Exceptional case	- If the user clicks the 'cancel' button, changes are not saved and the post before changes is shown - If the network is unstable, 'try again: network is unstable' message pops up
Process	1. If User clicks the part he wants to change in the post,

	the option bar appears. And users click the update button in the bar. 2. User revises the contents 3. User clicks the save button 4. Changes are saved and log about changes is saved 5. Return the changed post

SC-11

Bookmark	
Actors	User
Initial condition	- User is in the post - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- User can see the bookmarked posts in ‘My bookmark’ in My page
Exceptional case	- If the network is unstable, “try again: network is unstable‘ message pops up
Process	1. Press the ‘bookmark’ button in the option bar 2. The post is stored in the My bookmark list

SC-12

Recommend Post	
Actors	User
Initial condition	- User is in the post - User is logged in the application - Mobile devices should be accessible to the network

Final condition	- In the subject's feed, the number of the recommendation is added by 1.
Exceptional case	- If the network is unstable, 'try again: network is unstable' message pops up
Process	1. Press the like button in the option bar in the post 2. The variable which counts the number of 'like' is increased by 1 3. Changed the number of like is printed out below that post in the feed page

SC-13

Report Post	
Actors	User
Initial condition	- User is in post - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- The report is delivered to administrator
Exceptional case	- If the network is unstable, 'try again: network is unstable' message pops up
Process	1. User clicks the 'report' button in the option bar in the post that has other purpose such as commercial or not appropriate for subject classification 2. The report is delivered to the administrator

SC-14

Filter Post

Actors	User
Initial condition	- User has preferences over one or multiple criterias - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- The posts met the filter's criteria is displayed on the feed
Exceptional case	- If the network is unstable, "try again: network is unstable' message pops up
Process	1. User clicks on the spinner for filtering in the top of specific feed 2. User can set up each filter criteria option on the spinner 3. When finished, the system only displays posts meeting the filter

SC-15

Write Comment	
Actors	User
Initial condition	- User is in the post - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- User's comment is shown below the thread
Exceptional case	- If the network is unstable, when click the 'thread for comment' option, "try again: network is unstable' message pops up
Process	1. User clicks the post he wants to comment and clicks

	the comment icon in the option bar. 2. Thread for comment is opened 3. If user wants to write comment, user clicks on the input box, writes his comment and then clicks the save button

SC-16

Update Comment	
Actors	User
Initial condition	- User is in the post and thread for comment is opened - User is logged in the application - The comment is written by himself - Mobile devices should be accessible to the network
Final condition	- User's comment is changed and revised comment is shown in the thread
Exceptional case	- If the network is unstable, "try again: network is unstable' message pops up - If user clicks the 'cancel' button, original comment is shown in the thread
Process	1. User clicks update button in his comment 2. User revises the comment and click save button 3. Changes are saved in the DB 4. Changed comment in the thread is represented

SC-17

Delete Comment

Actors	User
Initial condition	- User is in the post and thread for comment is opened - The comment is written by himself - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- User's comment is deleted in the thread
Exceptional case	- If the network is unstable, "try again: network is unstable' message pops up - If user clicks the 'cancel' button, original comment is shown in the thread
Process	1. User clicks delete button in his comment 2. Warning message 'Are you sure you want to delete it?' is popped up 3. The data related to the comment is deleted in the DB 4. The thread is shown without the comment

SC-18

Report Comment	
Actors	User
Initial condition	- User is in thread - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- The report is delivered to administrator
Exceptional case	- If the network is unstable, 'try again: network is unstable' message pops up

Process	1. User clicks the 'report' button below the comment which has the inappropriate contents such as commercials in the thread. 2. The report is delivered to the administrator

SC-19

Search Post	
Actors	User
Initial condition	- User enters the main page - User is logged in the application - Mobile devices should be accessible to the network
Final condition	- Posts, containing the keyword, are shown on the board
Exceptional case	- If the network is unstable, "try again: network is unstable' message pops up
Process	1. User enters the keyword in the search box, and clicks the 'search' button 2. Posts, containing the keyword, are shown on the feed. - If you search on the Main page, users can see the feed regardless of the category. - If you search in a particular feed, it shows the feed in a specific subject feed.

SC-20

User Certification	
Actors	Administrator
Initial condition	- Each user's request is waiting for identification
Final condition	- Each user's account is activated
Exceptional case	- For some reasons such as faint or expired documents , the request can be rejected. - In this case, the user has to re-upload the document like student id card or certificate of studentship or teacher system certificate once more. The request is sent to each user's email with the list of the required documents.
Process	1. The pictures of student id card or certificate of studentship or teacher system certificate are uploaded by the user and are checked by the administrator 2. The administrator certifies the user and the account is added in database 3. If the requests are certified, a new account will be activated

SC-21

Update Node	
Actors	Administrator
Initial condition	- Admin enters the main page - Admin is logged in the application - Mobile devices should be accessible to the network

Final condition	- New Node is updated in the tree
Exceptional case	- If the network is unstable, ‘try again: network is unstable‘ message pops up
Process	1. In the Main page, the administrator clicks the upload button. 2. New node is popped up and the administrator writes the title of the node in the tree. 3. The tree, with new node, is shown in the Main page

SC-22

Delete Node	
Actors	Administrator
Initial condition	- Admin enters the main page - Admin is logged in the application - Mobile devices should be accessible to the network
Final condition	- Node that wants to delete is deleted in the tree
Exceptional case	- If the network is unstable, ‘try again: network is unstable‘ message pops up
Process	1. In the Main page, the administrator selects the node in the tree to delete and clicks the delete button. 2. The tree, without deleted node, is shown in the Main page

SC-23

Delete Post	
Actors	Administrator

Initial condition	- Admin receives the report about inappropriate post - Admin enters the feed related to the post - Admin is logged in the application - Mobile devices should be accessible to the network
Final condition	- Post that wants to delete is deleted in the feed
Exceptional case	- If the network is unstable, 'try again: network is unstable' message pops up
Process	1. In the feed that contains the post , the administrator clicks the delete button. 2. The administrator selects the node in the tree to delete and right clicks on the post. 3. Admin chooses the delete the post button 4. The feed, without deleted post, is shown

SC-24

Push Notification	
Actors	User
Initial condition	- Devices should be accessible to the network - The app is activating under the background
Final condition	- Push notification is made on the top bar of the smartphone.
Exceptional case	- If the network is unstable, "try again: network is unstable' message pops up
Process	1. Push notification is created in the following situations - New comment is written in their post - My post is changed by other users 2. User clicks the notification 3. Page is changed to the post page

SC-25

Send Direct Message	
Actors	User
Initial condition	- User is logged in the application - Mobile devices should be accessible to the network - User knows the receiver's name or student id
Final condition	- Receiver can read the message in the created chat room with sender

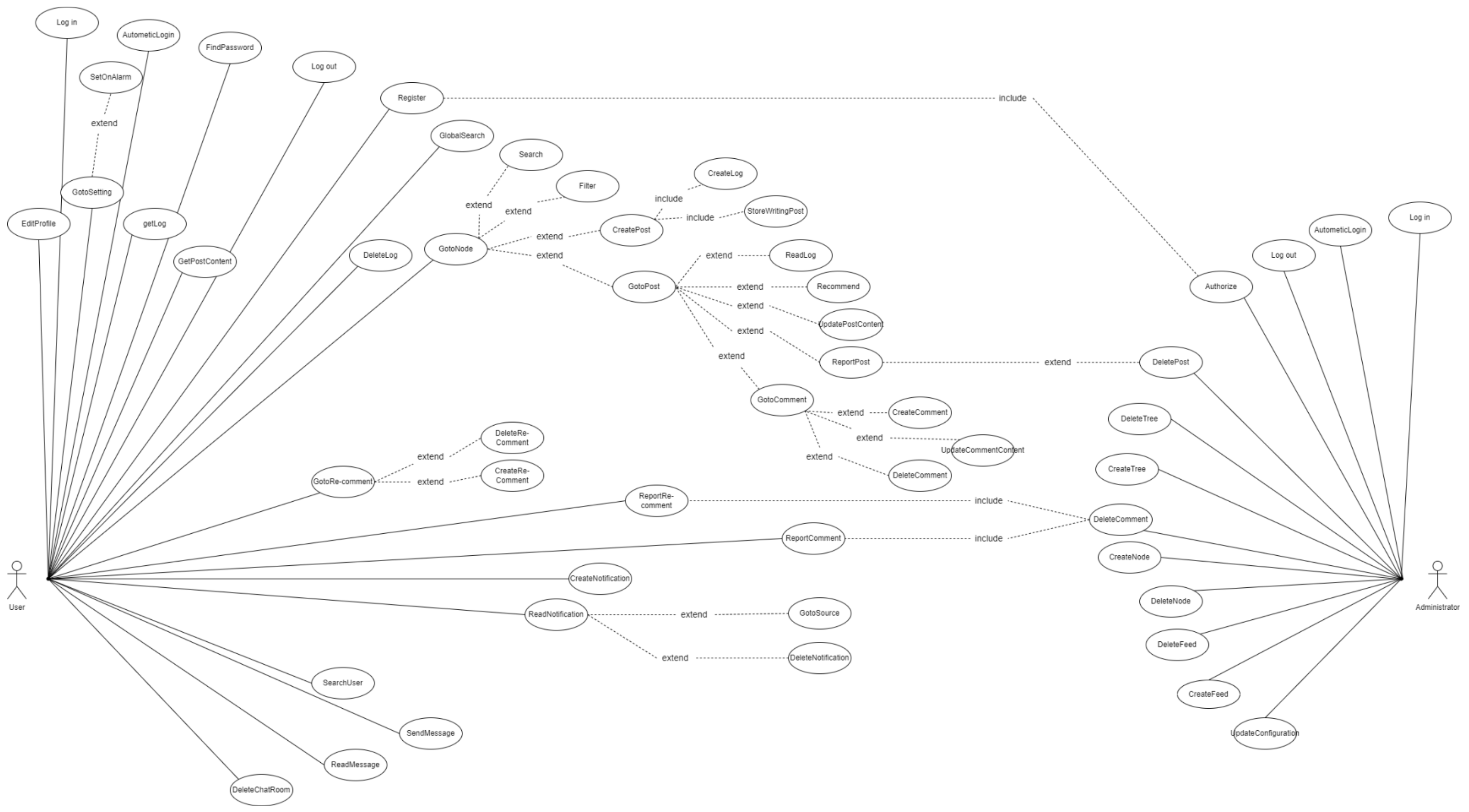
Exceptional case	- If the network is unstable, “try again: network is unstable‘ message pops up - If a user searches the wrong user’s name or id, it returns nothing.
Process	1. By clicking the user’s icon or clicks the message icon in the Direct message page 2. User clicks the receiver from the entire user in this step, users can search the receiver. 3. Chat room is created with the receiver. 4. Enter the message and click the send button 5. The message the user sent remains in the chat room

SC-26

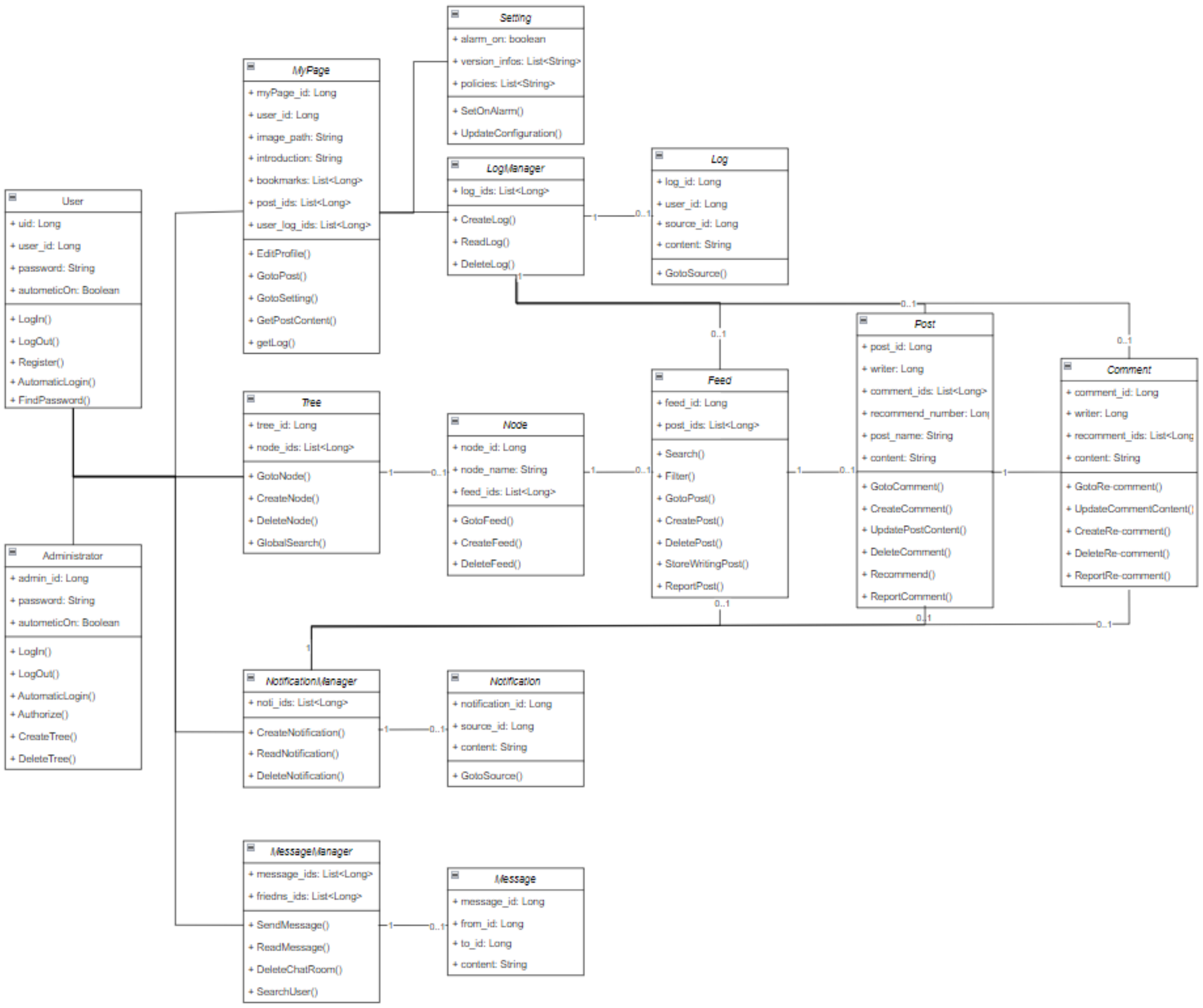
Delete Chat Room	
Actors	User
Initial condition	- Devices should be accessible to the network - There is a chat room that shared a specific user
Final condition	- Chat room that is deleted is removed in the list in the DM page
Exceptional case	- If the network is unstable, “try again: network is unstable‘ message pops up
Process	4. User clicks strongly on the chat room in the list 5. Option windows appears 6. Click the delete option 7. Chat room list is shown without that chat room

V. UML

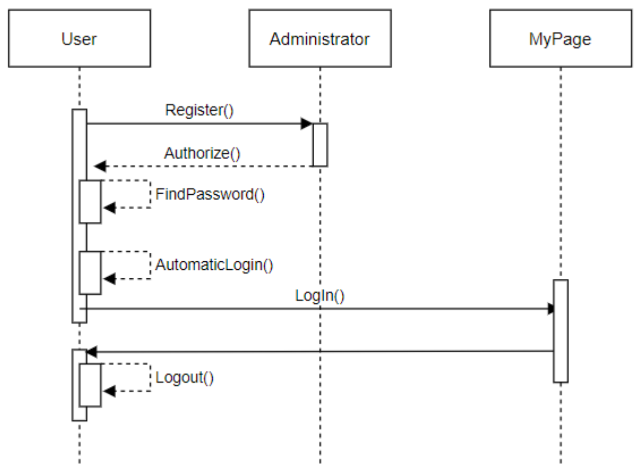
V-A. Use case Diagram

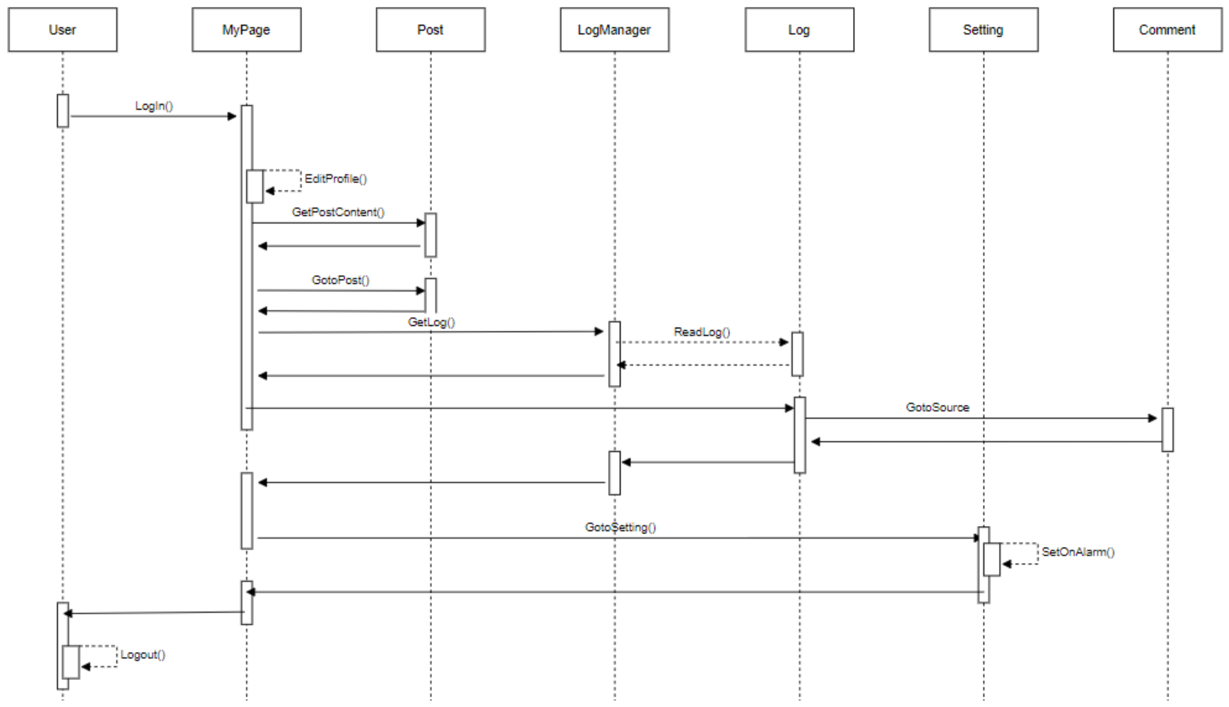


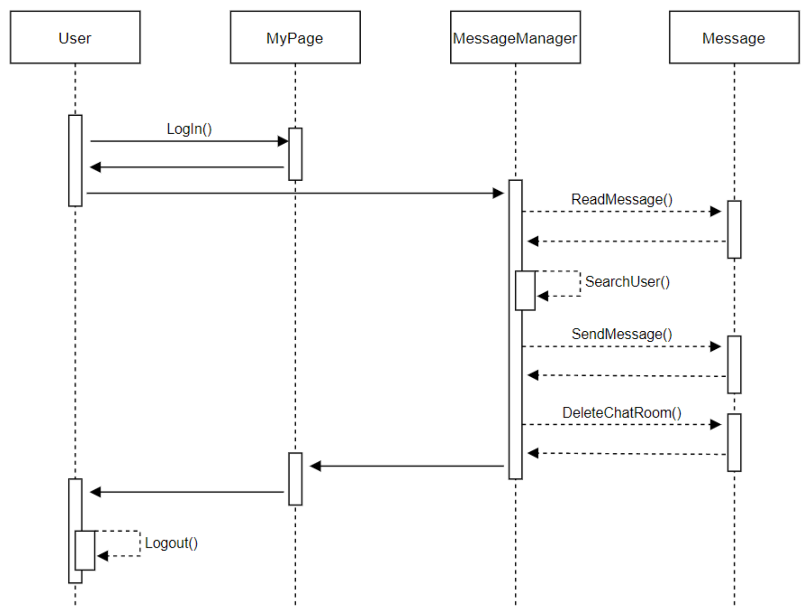
V-B. Class Diagram

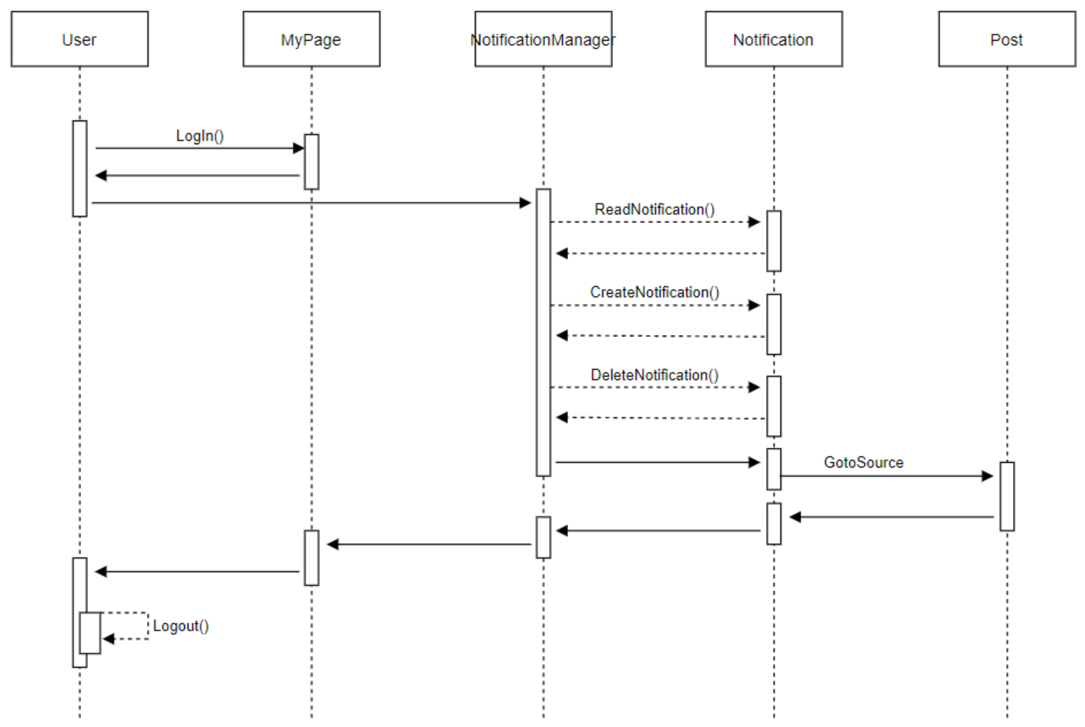


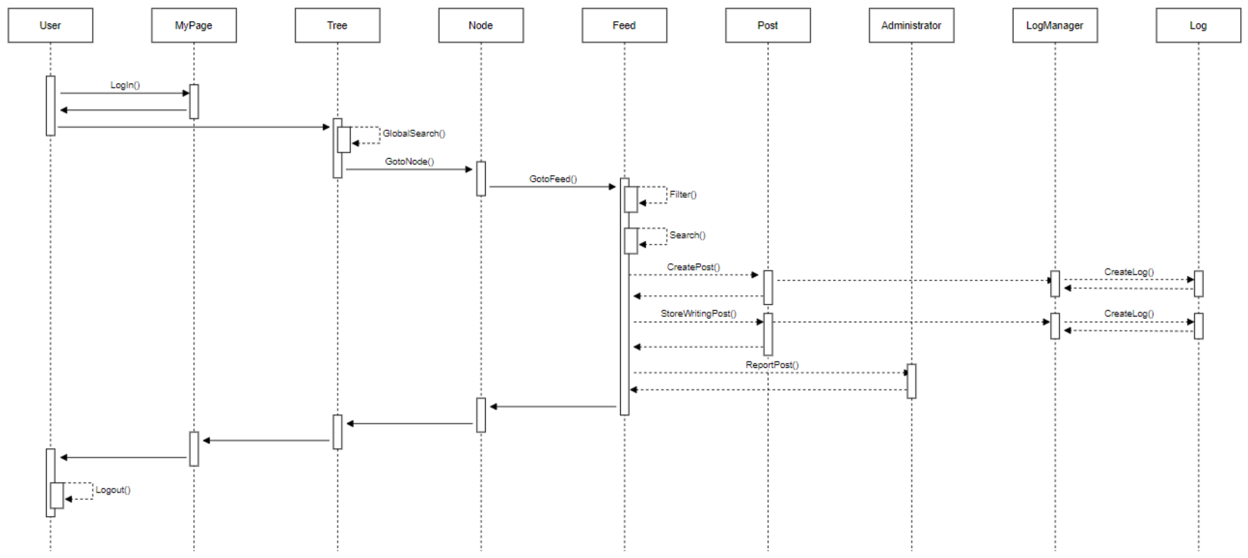
V-C. Sequential Diagram

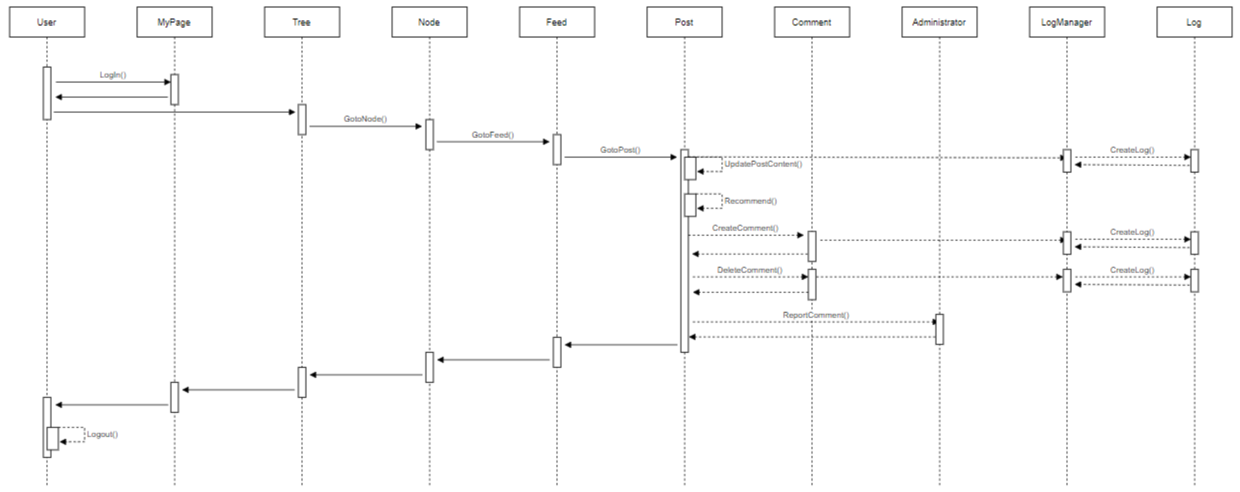


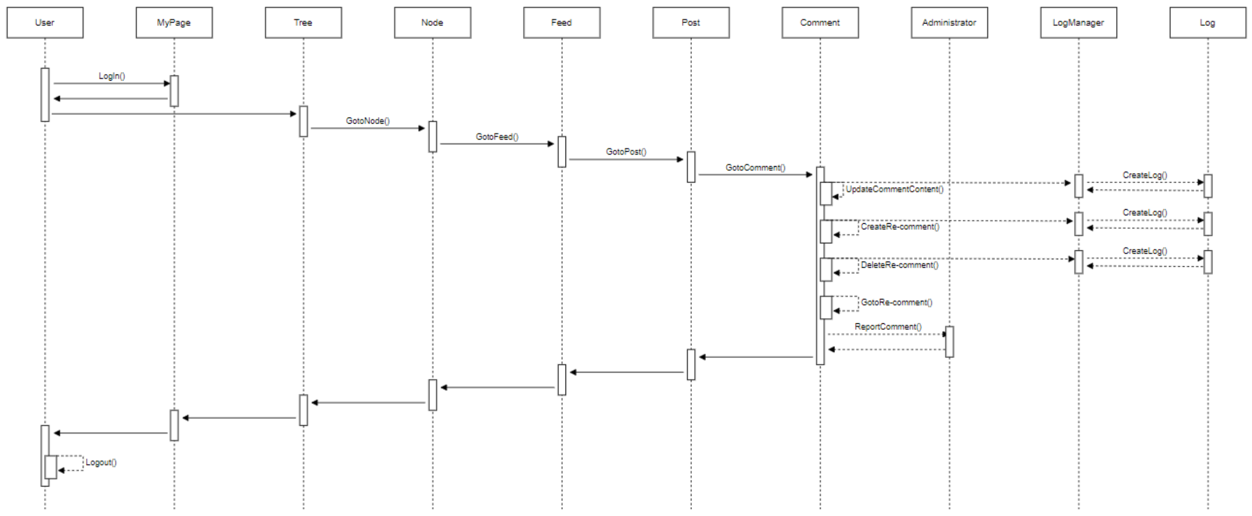


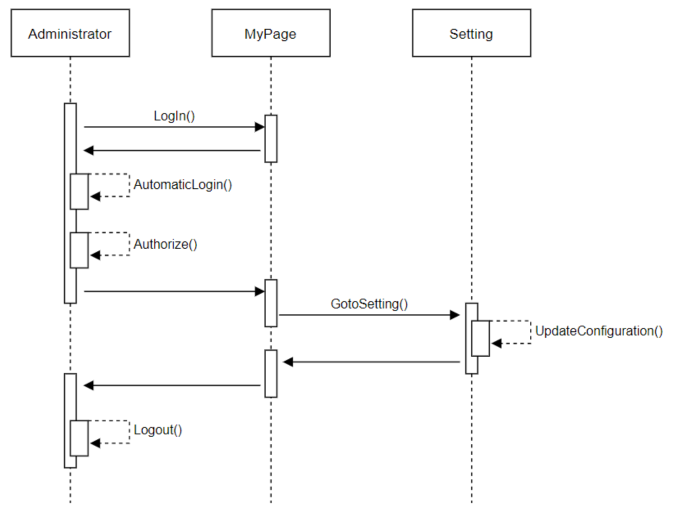

User
MyPage
MessageManager
Message
LogIn()
ReadMessage()
SearchUser()
SendMessage()
DeleteChatRoom()
Logout()

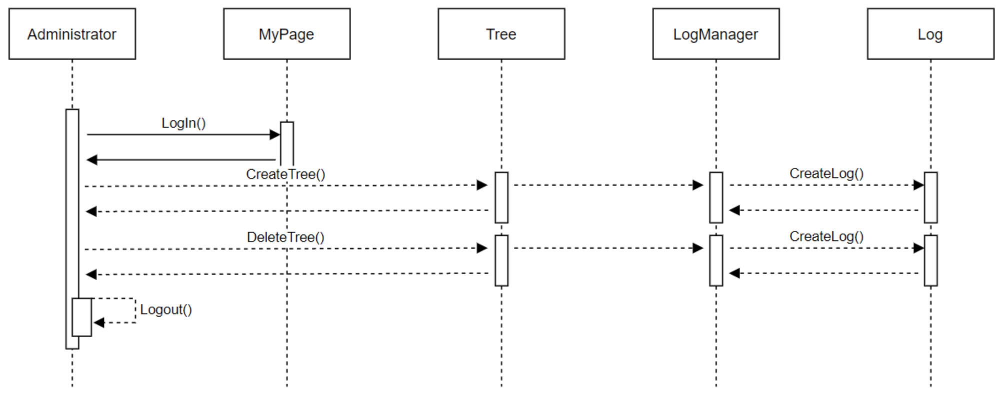

User
MyPage
NotificationManager
Notification
Post
LogIn()
ReadNotification()
CreateNotification()
DeleteNotification()
GotoSource
Logout()

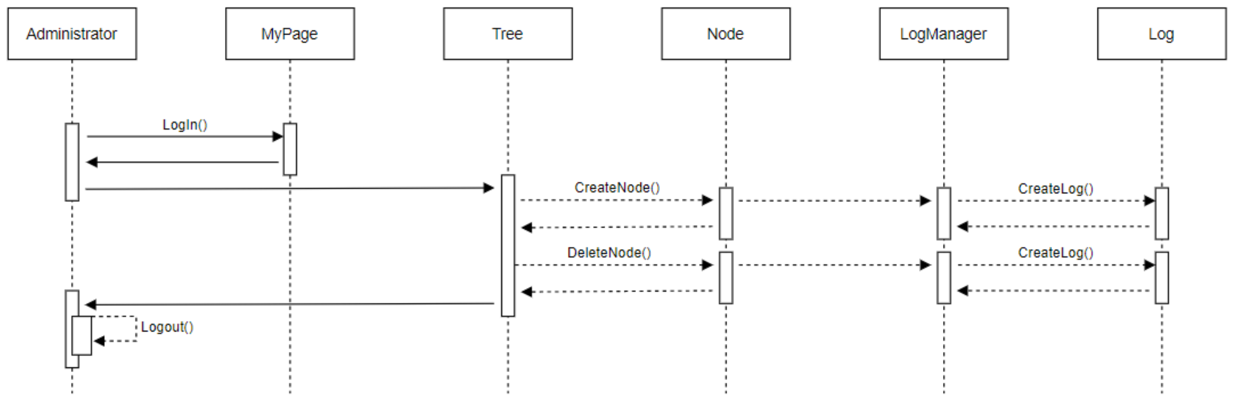

User
MyPage
Tree
Node
Feed
Post
Administrator
LogManager
Log
LogIn()
GlobalSearch()
GotoNode()
GotoFeed()
Filter()
Search()
CreatePost()
CreateLog()
StoreWritingPost()
CreateLog()
ReportPost()
Logout()

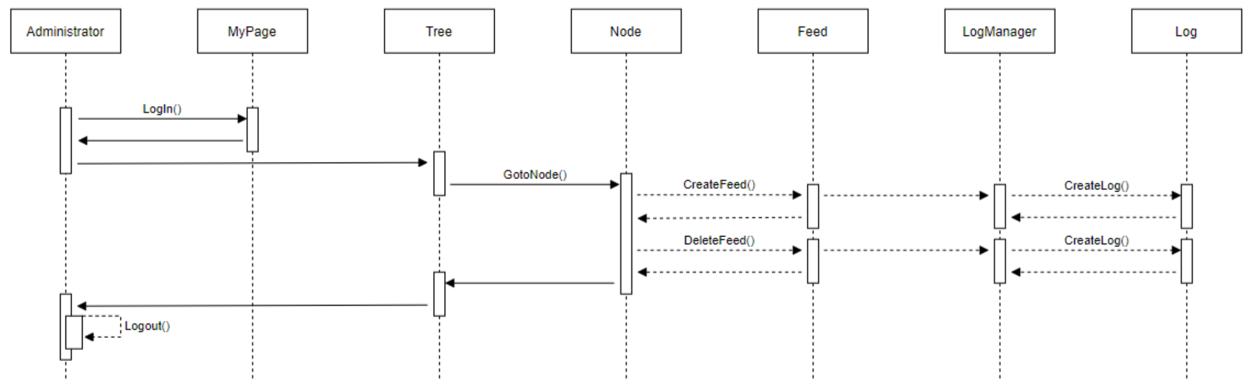

User
MyPage
Tree
Node
Feed
Post
Comment
Administrator
LogManager
Log
LogIn()
GotoNode()
GotoFeed()
GotoPost()
UpdatePostContent()
CreateLog()
Recommend()
CreateComment()
CreateLog()
DeleteComment()
CreateLog()
ReportComment()
Logout()

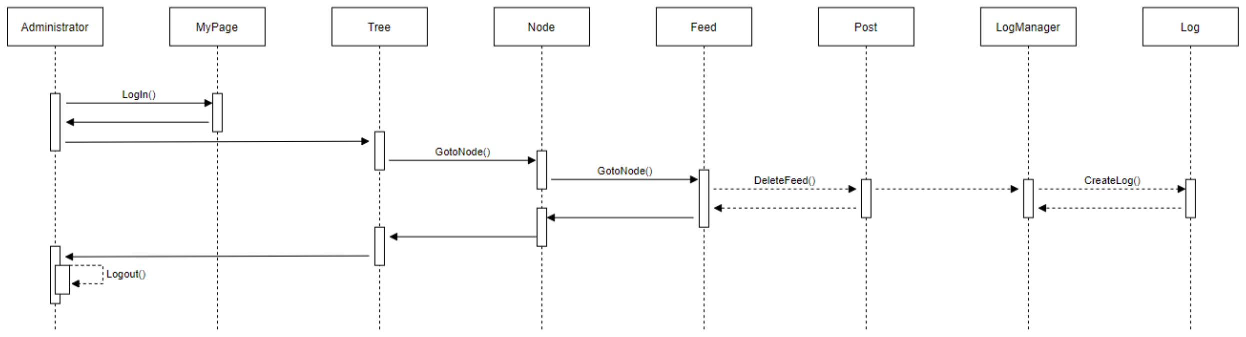

User
MyPage
Tree
Node
Feed
Post
Comment
Administrator
LogManager
Log
LogIn()
GotoNode()
GotoFeed()
GotoPost()
GotoComment()
UpdateCommentContent()
CreateLog()
CreateRe-comment()
CreateLog()
DeleteRe-comment()
CreateLog()
GotoRe-comment()
ReportComment()
Logout()


Administrator
MyPage
Setting
LogIn()
AutomaticLogin()
Authorize()
GotoSetting()
UpdateConfiguration()
Logout()


Administrator
MyPage
Tree
LogManager
Log
LogIn()
CreateTree()
CreateLog()
DeleteTree()
CreateLog()
Logout()


Administrator
MyPage
Tree
Node
LogManager
Log
LogIn()
CreateNode()
CreateLog()
DeleteNode()
CreateLog()
Logout()


Administrator
MyPage
Tree
Node
Feed
LogManager
Log
LogIn()
GotoNode()
CreateFeed()
CreateLog()
DeleteFeed()
CreateLog()
Logout()


Administrator
MyPage
Tree
Node
Feed
Post
LogManager
Log
LogIn()
GotoNode()
GotoNode()
DeleteFeed()
CreateLog()
Logout()

V-D. Interface Design

NO.	Function (method)	Parameter	Return	Process
1	LogIn	Long: user_id String: password	Void	Get the string type value of the user_id and password. After successful certification, users can log in and enter into the Tree.
2	LogOut	-	Void	When the user clicks the logout button, logout is returned.
3	Register	String: user_id String: password String: name Int:Phone Number String: Identification	Void	Putting user_id, password, name, Phone Number, and Identification and these are stored to the database
4	Automatic Login	Boolean: isToken	Void	If the user sets the isToken is true, after that, it is logged in automatically. Otherwise, nothing will happen
5	FindPassword	String: user_id Int: Phone number	String: Password	If the user enters the user_id and Phone number, the corresponding Password is returned

6	Authorize	Boolean: isValid	Void	If user's info is valid, administrator inputs whether it is valid or not as a parameter, and then, user can login to this app
7	DeletePost	Long: post_id	Void	Post correspond to the post_id is deleted
8	CreateTree	-	Long: tree_id	In the tree page, new tree is created and it returns the tree_id
9	DeleteTree	Long: tree_id	Void	Delete tree correspond to the tree_id
10	UpdateTree	Long: tree_id	Void	Update changes correspond to tree_id
11	CreateNode	String: node_name List<Long>: feed_ids	Long: node_id	Node is created with node_name and empty feed_ids list. And this Node is added in the Tree while returning node_id
12	DeleteNode	Long: node_id	Void	Node correspond to the node_id is deleted
13	CreateFeed	List<Long>: post_ids	Long: feed_id	Create Feed with empty post_ids list. After feed is created, feed_id is returned
14	UpdateFeed	Long: feed_id	Void	Update feed

				correspond to the feed_id
15	DeleteFeed	Long: feed_id	Void	Feed correspond to the feed_id is deleted
16	UpdateConfiguration	String: version_info	Void	When version and policy information is changed, corresponding
		String: policy		
17	EditProfile	String: image_path	Void	When image_path and introduction that the user wants to update is entered as a parameter, after that, the profile image is updated.
		String: introduction		
18	GetPostContent	Long: post_id	Void	User can go to the post correspond to the post_id
19	getLog	Long: log_id	Void	User can get the log correspond to the log_id
20	SetOnAlarm	Boolean: isAlarm	Void	If user sets the isAlarm is true, alarm can be shown
21	CreateLog	Long: source_id	Long: log_id	Changed content and source_id is entered as a parameter. And then, the log_id is returned.
		String: content		
22	ReadLog	Long: log_id	String: content	LogManager can read the log to see what

				users have changed.
23	DeleteLog	Long: log_id	Void	In order to use memory efficiently, logs that have passed a certain period of time are deleted.
24	GotoSource	Long: source_id	Void	If the user clicks the event that happens such as log and notification, move to the event. It is used as the global method.
25	GotoNode	Long: node_id	Void	If the user click a specific node, move to the feed
26	GlobalSearch	String: word	Void	If the user want to find a specific post related to the keyword, it shows the result within the entire range
27	GotoFeed	Long: feed_id	Void	If the user is out of the post, the user goes to the feed
28	Search	String: word	Void	If the user want to find a specific post related to the keyword, it shows the result within a limited range
29	Filter	String: condition	Void	A condition that the user wants to apply is

				inputted as a parameter, and it's used for filtering the post
30	GotoPost	Long: post_id	Void	When the user click the post, the page is changed to post correspond to the post_id
31	CreatePost	String: post_name	Long: post_id	Create Post with post_name and content. After post is created, post_id is returned
		String: content		
32	StoreWritingPost	String: title	Long: post_id	The title and content of the post that the user was writing are delivered as parameters, and the post_id for temporary store is returned
		String: content		
33	ReportPost	Long: post_id	Void	Post correspond to the post_id is reported
34	GotoComment	Long: comment_id	Void	The user clicks comment to see the content of comment or write the comment
35	CreateComment	String: content	Long: comment_id	Create a comment with an empty re-comment list. After that, comment_id is returned
		List<Long>: re-comment		

36	UpdatePostContent	String: content	Void	Update the content, which is in the post.
37	Recommend	-	Void	When it is executed, recommend_number is added by 1
38	ReportComment	Long: comment_id	Void	If there is inappropriate content in the comment, the user can report it.
39	GotoRe-comment	Long: comment_id	Void	Move to the re-comment in the comment in the thread
40	UpdateCommentContent	List<Long>: comment_ids Long: comment_id String: content	Void	Update contents in the comment on the thread
41	CreateRe-comment	String: content	Long: re-comment_id	Create re-comment with contents. After that, re-comment_id is returned
42	DeleteRe-comment	Long: re-comment_id	Void	Comment correspond to re-comment_id is deleted
43	ReportRe-comment	Long: re-comment_id	Void	Report the re-comment correspond to the re-comment_id
44	DeleteComment	Long: comment_id	Void	Comment correspond to comment_id is deleted
45	CreateNotification	Long: notification_id	Long: notification_id	Notification object is

		String: content		created with notification_id and content. And this Notification is popped up
46	ReadNotification	Long: notification_id	String: content	Show a notification content correspond to notification_id
47	DeleteNotification	Long: notification_id	Void	Notification correspond to notification_id is deleted
48	GotoSource	Long: source_id	Void	When the user click the notification or log, the page is changed to post correspond to the post_id
49	SendMessage	Long: from_id	Long: message_id	Written content is sent from the sender(from_id) to receiver(to_id), and the messege_id is returned.
		Long: to_id		
		String: content		
50	ReadMessage	Long: message_id	String: content	Show a message content correspond to message_id
51	DeleteChatRoom	Long: message_id	Void	Deleted message_id is inputted as a parameter, and it's used for deleting the chat room corresponding to the message_ id
52	SearchUser	String:name	Void	Receiver's name

				is inputted as a parameter, and it’s used for searching the receiver
53	GotoSetting	-	Void	Move to the setting option

V-E. Detailed Design

Class name	User		
Function	Input	Output	Process
Login	Long: user_id String: password	Void	Login as user
Logic	Get the long and string type, user_id and password. If the Id and password exist in the users table in DB , users and administrators can log in and enter into the main menu. (SELECT USER_ID FROM USERS FROM USER_ID = "$USER_ID" AND PASSWORD = "$PASSWORD")		

Class name	User		
Function	Input	Output	Process
LogOut	-	Void	Logout
Logic	When the user and the administrator clicks the logout button, the user is in the logout state. If AutomaticLogin is not checked, the user's access token expires.		

Class name	User		
Function	Input	Output	Process
Register	String: user_id String: password String: name Int: Phone number String: Identification	Void	Register as user

Logic	For the registration, get several string type inputs and int, user_id, password, name, identification, and phone number. The user's registration request is sent to the administrator
	
	SELECT USERID FROM USER;
	
	INSERT INTO USER(USER_ID,NAME,PHONENUMBER,PASSWORD,IDENTIFICATION)		

	VALUES (#{USERID},#{NAME},#{PHONENUMBER},#{PASSWORD},#{IDENTIFICATION})

Class name	User		
Function	Input	Output	Process
AutomaticLogin	Boolean: isToken	Void	Check for using automatic login
Logic	If a user or an administrator has an access token that has not expired, the user will be logged in automatically. If a user or an administrator does not have an access token, the user cannot be logged in automatically.		

Class name	User		
Function	Input	Output	Process
FindPassword	String: user_id Int: Phone number	String: password	Find password, if the user forget it
Logic	Get string and int type input, user_id and phone number. If the Id and phone number exist in the users table in DB, return the password. SELECT PASSWORD FROM USERS WHERE -		

Class name	Administrator		
Function	Input	Output	Process
Authorize	Boolean: isValid	Void	Permission to login

Logic	Get boolean input as a parameter for Login, and check if it is true, succeed to login. Only the administrator can use the Authorize function.

Class name	Administrator		
Function	Input	Output	Process
DeletePost	Long: post_id	Void	Delete a post based on the report content
Logic	Administrators delete the corresponding post object through post_id. And Delete the id of the post from the post_ids of the feed to which the post belongs. All comments belonging to that post are also deleted. DELETE POST_ID FROM Posts WHERE -		

Class name	Administrator		
Function	Input	Output	Process
CreateTree	-	Long: tree_id	Create a tree based on the curriculum
Logic	Administrator creates tree objects. And this data is stored in the DB.		

Class name	Administrator		
Function	Input	Output	Process
DeleteTree	Long: tree_id	Void	Delete a tree if curriculum is finished
Logic	Administrators delete the corresponding tree object through tree_id. And all objects belonging to that tree are also deleted. DELETE TREE_ID FROM TREES WHERE -		

Class name	Administrator

Function	Input	Output	Process
CreateNode	String: node_name List<Long>: post_ids	Long: node_id	Create a node based on the subject of curriculum
Logic	Administrators create the node for curriculum change. Get string and list type, node_name and feed_ids to create nodes and it returns node_id. INSERT INTO NODES (TREE_ID, CREATED_AT) VALUES (${}, ${})		

Class name	Administrator		
Function	Input	Output	Process
DeleteNode	Long: node_id	Void	Delete a node based on the subject of curriculum
Logic	Administrators delete the corresponding node object through node_id. And Delete the node_id from the node_ids of the tree to which the post belongs. All objects belonging to that node are also deleted. DELETE NODE_ID FROM NODES WHERE -		

Class name	Administrator		
Function	Input	Output	Process
CreateFeed	List<Long>: post_ids	Long: feed_id	Create a feed based on the subject
Logic	Administrators create the feed to show the posts. Get list type input, post_ids to create feed and it returns feed_id INSERT INTO FEED (NODE_ID, CREATED_AT) VALUES (${}, ${})		

Class name	Administrator		
Function	Input	Output	Process
DeleteFeed	Long: feed_id	Void	Delete a feed based

			on the subject
Logic	Administrators delete the corresponding feed object through feed_id. And Delete the feed_id from the feed_ids of the node to which the feed belongs. All objects belonging to that feed are also deleted. DELETE FEED_ID FROM FEEDS WHERE -		

Class name	Administrator		
Function	Input	Output	Process
UpdateConfiguration	String: version_info String: policies	Void	Update a configuration based on the version and policies
Logic	Get the two String to configuration, version_infos and policies and add them to each List, Also, the revised items are deleted by the administrator.		

Class name	MyPage		
Function	Input	Output	Process
EditProfile	String: image_path String: introduction	My page	Edit a profile with image and introduction content
Logic	Get string type inputs, image_path and introduction from the user and these datas is stored in the DB. Also, change the UI using these variables.		

Class name	MyPage		
Function	Input	Output	Process
GetPostContent	Long: post_id	Void	Move to the post
Logic	If the user clicks the post, Search the location of the post in the DB and go to that using the id.		

Class name	MyPage		
Function	Input	Output	Process
getLog	Long: log_id	Void	Get a log from the perspective of the user
Logic	If the user clicks the log, Search the log in the DB and return the content of log using the id. SELECT TEXTCONTENTS FROM LOGS WHERE -		

Class name	Setting		
Function	Input	Output	Process
SetOnAlarm	Boolean: isAlarm	Void	Set an alarm
Logic	Get boolean type variable from the user and assign to “isAlarm” variable.		

Class name	LogManager		
Function	Input	Output	Process
CreateLog	Long: source_id	Long: log_id	When the user changes the content, log is generated and is stored
Logic	Use the changes caused by user activities, corresponding object id and user_id to make the Log object. And add the generated log_id to user_log_ids in MyPage object. INSERT INTO LOGS (USER_ID, POST_ID, TEXTCONTENTS, CREATED_AT) VALUES (${}, ${}, ${})		

Class name	LogManager

Function	Input	Output	Process
ReadLog	Long: log_id	String: content	Get a log from the perspective of the logmanager
Logic	Read the log to see what users have changed. SELECT TEXTCONTENTS FROM LOGS WHERE -		

Class name	LogManager		
Function	Input	Output	Process
DeleteLog	Long: log_id	Void	Delete a log
Logic	In order to use memory efficiently, logs that have passed a certain period of time are deleted. DELETE LOG_ID FROM LOGS WHERE -		

Class name	Log		
Function	Input	Output	Process
GotoSource	Long: source_id	Void	Move to a specific source from the log and notification.
Logic	Get long type input, source_id to move a specific source from the log and notification.		

Class name	NotificationManager		
Function	Input	Output	Process
CreateNotification	Long: notification_id String: content	Long: notification_id	Notification object is created with notification_id and content. And this Notification is popped up

Logic	Get long and string type inputs, notification_id and content to create notification SELECT ${} FROM USERS WHERE -

Class name	NotificationManager		
Function	Input	Output	Process
ReadNotification	Long: notification_id	String: content	Show a notification content correspond to notification_id
Logic	Get long type input, notification_id to show a notification SELECT * FROM USERS WHERE -		

Class name	NotificationManager		
Function	Input	Output	Process
DeleteNotification	Long: notification_id	Void	Notification correspond to notification_id is deleted
Logic	Get long type input, notification_id to delete the notification		

Class name	Notification		
Function	Input	Output	Process
GotoSource	Long: source_id	Void	When the user click the notification, the page is changed to post correspond to the post_id
Logic	Get long type input, source_id, user move to a specific activity corresponded to the notification		

Class name	MessageManager		
Function	Input	Output	Process
SendMessage	Long: from_id Long: to_id String: content	Long: message_id	Written content is sent from the sender(from_id) to receiver(to_id), and the messege_id is returned.
Logic	Get long and string type inputs, from_id, to_id, and content to send a message to a specific user. If there is no chat room with a specific user, the chat room is created automatically and messages are sent.		

Class name	MessageManager		
Function	Input	Output	Process
ReadMessage	Long: message_id	String: content	Show a message content correspond to message_id
Logic	Get long type input, message_id to read message from a specific message_id		

Class name	MessageManager		
Function	Input	Output	Process
DeleteChatRoom	Long: message_id	Void	Deleted message_id is inputted as a parameter, and it's used for deleting the chat room corresponding to the message_ id
Logic	Get long type input, message_id to delete chat room for chatting DELETE MESSAGE_ID FROM CHANNELS WHERE -		

Class name	MessageManager		
Function	Input	Output	Process

SearchUser	String:name	Void	Receiver’s name is inputted as a parameter, and it’s used for searching the receiver
Logic	Get string type input, name to search a specific user SELECT * FROM USERS WHERE -		

Class name	Tree		
Function	Input	Output	Process
GotoNode	Long: node_id	Void	Move to the node
Logic	If the user clicks the node, Search the location of node in the DB and go to that using the id.		

Class name	Tree		
Function	Input	Output	Process
GlobalSearch	String: word	Void	Search a keyword within the whole range
Logic	Get string type input, word to search a specific word within the whole range SELECT * FROM POSTS WHERE -		

Class name	Node		
Function	Input	Output	Process
GotoFeed	Long: feed_id	Void	Move to the feed
Logic	If the user clicks the feed, Search the location of the feed in the DB and go to that using the id.		

Class name	Feed		
Function	Input	Output	Process
Search	String: word	Void	Search a keyword within a limited range
Logic	Get string type input, word to search a specific word within a limited range SELECT * FROM POSTS WHERE -		

Class name	Feed		
Function	Input	Output	Process
Filter	String: condition	Void	Show posts in a specific order on the feed
Logic	Get string type input, condition to filter posts		

Class name	Feed		
Function	Input	Output	Process
GotoPost	Long: post_id	Void	Move to the post
Logic	Get long type input, post_id to move a post		

Class name	Feed		
Function	Input	Output	Process
CreatePost	String: post_name String: content	Long: post_id	Create a post
Logic	Get string type inputs, post_name and content to create post		

Class name	Feed		
Function	Input	Output	Process
StoreWriting Post	String: title String: content	Long: post_id	Store a content when the user stop writing
Logic	Get string type inputs, title and content to store content, which the user are writing		

Class name	Feed		
Function	Input	Output	Process
ReportPost	Long: post_id	Void	Report a post for a specific reason
Logic	Get long type input, post_id to report it as inappropriate post		

Class name	Post		
Function	Input	Output	Process
GotoCommen t	Long: comment_id	Void	Move to the comment through the thread
Logic	Get long type input, comment_id to move a specific comment section		

Class name	Post		
Function	Input	Output	Process
CreateComm ent	String: content List<Long>: re-comment	Long: comment_id	Create a comment in the thread

Logic	Get string and list type inputs, content and re-comment to write comment

Class name	Post		
Function	Input	Output	Process
UpdatePostContent	String: content Long: post_id	Void	Update contents in the post corresponded to post_id
Logic	Get string content and post_id to update its contents.		

Class name	Post		
Function	Input	Output	Process
DeleteComment	List<Long>: comment_ids Long: comment_id	Void	Delete the comment id corresponded to id
Logic	Delete the comment_id in the comment_ids list. User can delete only the comment that the user wrote, Administrator can delete all comment.		

Class name	Post		
Function	Input	Output	Process
Recommend	-	Void	When it is executed, recommend_number is added by 1
Logic	If the user click the like button, recommned_number is added by 1		

Class name	Post

Function	Input	Output	Process
ReportComment	Long: comment_id	Void	Report a comment for a specific reason
Logic	Get long type input, comment_id to report comment		

Class name	Comment		
Function	Input	Output	Process
GotoRe-comment	Long: comment_id	Void	Move to the comment in the comment in the thread
Logic	Get long type input, comment_id to move a specific comment section		

Class name	Post		
Function	Input	Output	Process
UpdateCommentContent	List<Long>: comment_ids Long: comment_id String: content	Void	Update contents in the comment on the thread
Logic	Get comment_id and content to update its content.		

Class name	Comment		
Function	Input	Output	Process
CreateRe-comment	String: content	Long: re-comment_id	Create a re-comment, which means comment in the comment

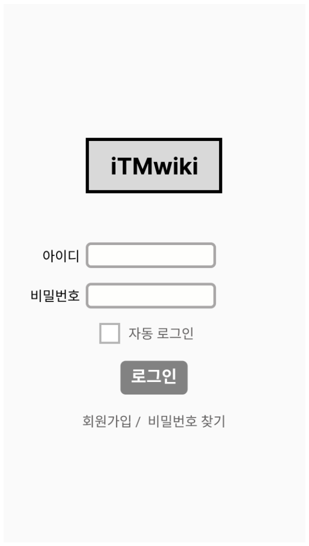
Logic	Get long type input, re-comment_id to create re-comment

Class name	Comment		
Function	Input	Output	Process
DeleteRe-comment	Long: re-comment_id	Void	Delete a re-comment
Logic	Get long type input, re-comment_id to delete re-comment. User can delete only the re-comment that the user wrote, Administrator can delete all re-comment DELETE RE-COMMENT_ID FROM COMMENT WHERE -		

Class name	Comment		
Function	Input	Output	Process
ReportRe-comment	Long: comment_id	Void	Report a re-comment for a specific reason
Logic	Get long type input, comment_id to report comment		

Class name	Comment		
Function	Input	Output	Process
GotoSetting	-	Void	Move to setting option
Logic	Move to the setting option to change setting		

VI. User Interface Design


iTMwiki

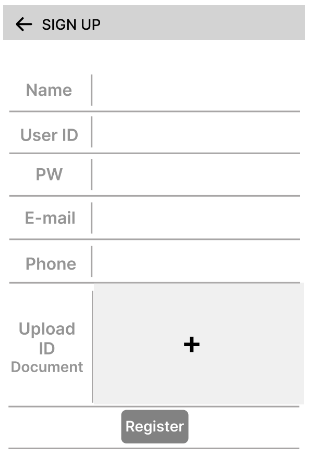
아이디

비밀번호

자동 로그인

로그인

회원가입 / 비밀번호 찾기


SIGN UP

Name

User ID

PW

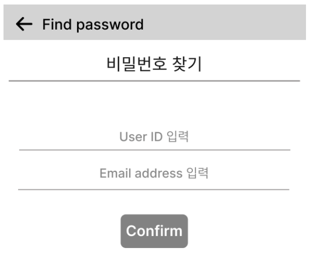
E-mail

Phone

Upload ID Document

+

Register

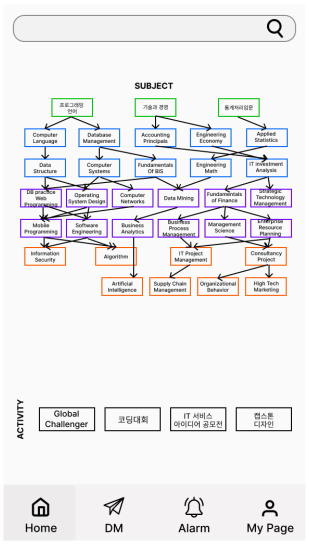

Find password

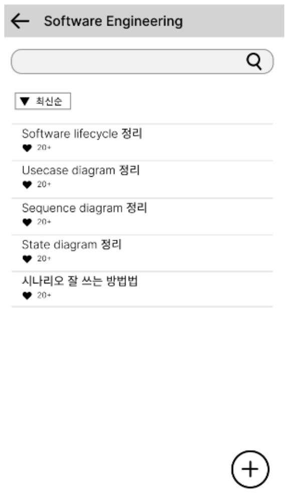
비밀번호 찾기

User ID 입력

Email address 입력

Confirm




Software Engineering

▼ 최신순

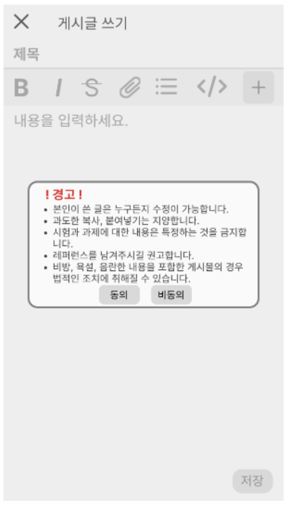
Software lifecycle 정리
♥ 20+

Usecase diagram 정리
♥ 20+

Sequence diagram 정리
♥ 20+

State diagram 정리
♥ 20+

시나리오 잘 쓰는 방법법
♥ 20+


게시글 쓰기

제목

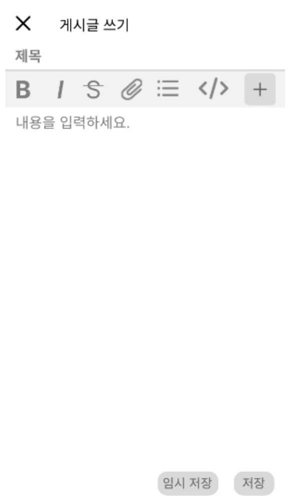
내용을 입력하세요.

! 경고 !

- 본인이 쓴 글은 누구든지 수정이 가능합니다.
- 과도한 복사, 붙여넣기는 지양합니다.
- 시험과 과제에 대한 내용은 특정하는 것을 금지합니다.
- 레퍼런스를 남겨주시길 권고합니다.
- 비방, 욕설, 음란한 내용을 포함한 게시물의 경우 법적인 조치에 취해질 수 있습니다.

동의 비동의

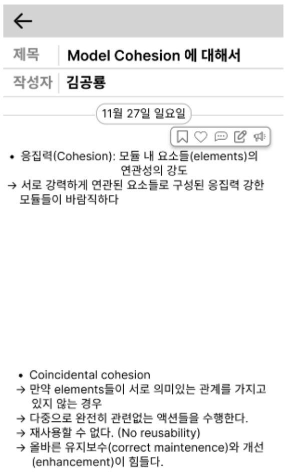
저장


게시글 쓰기

제목

내용을 입력하세요.

임시 저장 저장


제목 **Model Cohesion 에 대해서**

작성자 **김공룡**

11월 27일 일요일

- 응집력(Cohesion): 모듈 내 요소들(elements)의 연관성의 강도

→ 서로 강력하게 연관된 요소들로 구성된 응집력 강한 모듈들이 바람직하다

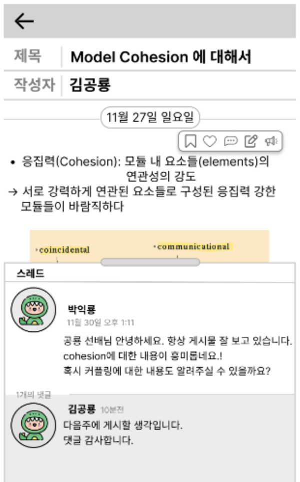
- Coincidental cohesion

→ 만약 elements들이 서로 의미있는 관계를 가지고 있지 않는 경우

→ 다중으로 완전히 관련없는 액션들을 수행한다.

→ 재사용할 수 없다. (No reusability)

→ 올바른 유지보수(correct maintenence)와 개선(enhancement)이 힘들다.


제목 **Model Cohesion 에 대해서**

작성자 **김공룡**

11월 27일 일요일

- 응집력(Cohesion): 모듈 내 요소들(elements)의 연관성의 강도

→ 서로 강력하게 연관된 요소들로 구성된 응집력 강한 모듈들이 바람직하다

◦coincidental ◦communicational

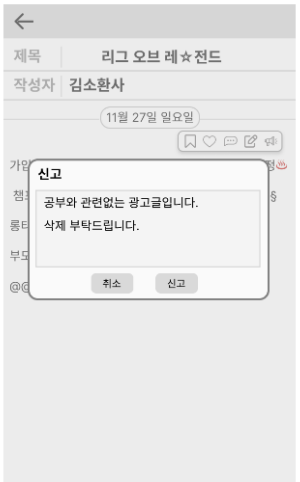
스레드

박익룡
11월 30일 오후 1:11

공룡 선배님 안녕하세요. 항상 게시물 잘 보고 있습니다.
cohesion에 대한 내용이 흥미롭네요.!
혹시 커플링에 대한 내용도 알려주실 수 있을까요?

1개의 댓글

김공룡 10분전
다음주에 게시할 생각입니다.
댓글 감사합니다.


제목 **리그 오브 레☆전드**

작성자 **김소환사**

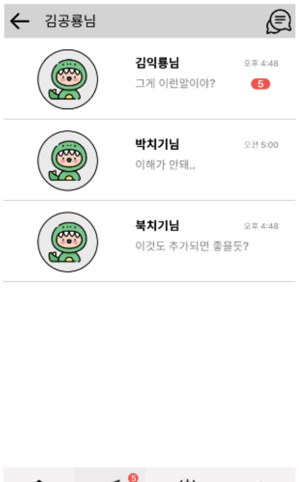
11월 27일 일요일

신고

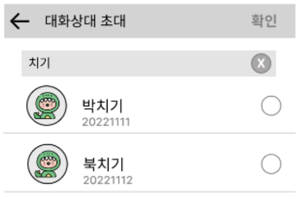
공부와 관련없는 광고글입니다.

삭제 부탁드립니다.

취소 신고


김공룡님

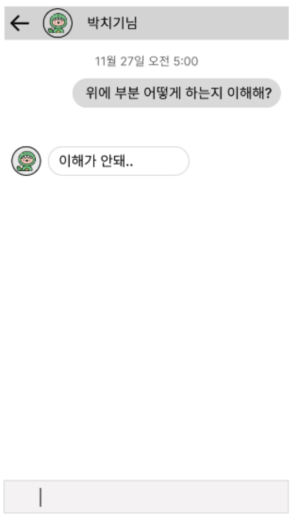
김익룡님	그게 이런말이야?	오후 4:48 5
박치기님	이해가 안돼..	오전 5:00
북치기님	이것도 추가되면 좋을듯?	오후 4:48


대화상대 초대 확인

치기

박치기
20221111

북치기
20221112

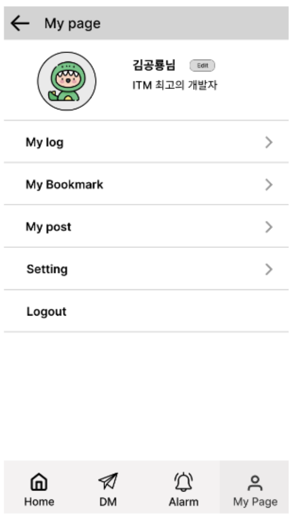


박치기님

11월 27일 오전 5:00

위에 부분 어떻게 하는지 이해해?

이해가 안돼..



My page

김공룡님 Edit

ITM 최고의 개발자

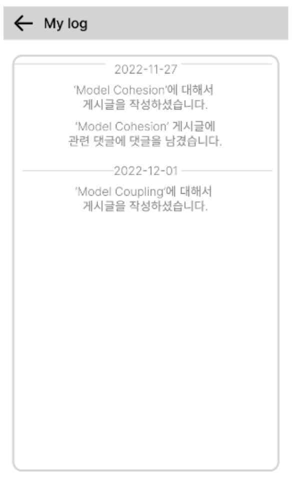
My log

My Bookmark

My post

Setting

Logout



My log

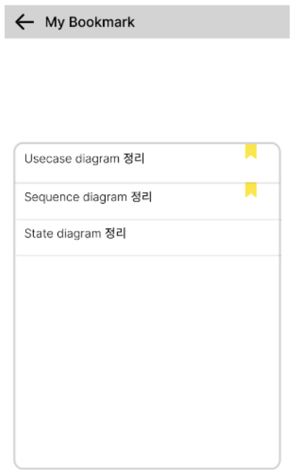
2022-11-27

'Model Cohesion'에 대해서 게시글을 작성하셨습니다.

'Model Cohesion' 게시글에 관련 댓글에 댓글을 남겼습니다.

2022-12-01

'Model Coupling'에 대해서 게시글을 작성하셨습니다.

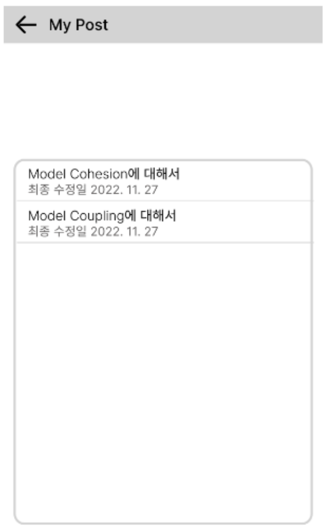


My Bookmark

Usecase diagram 정리

Sequence diagram 정리

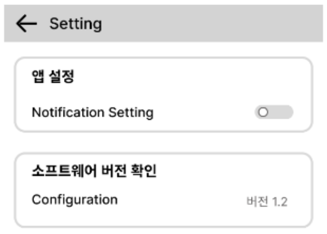
State diagram 정리



My Post

Model Cohesion에 대해서
최종 수정일 2022. 11. 27

Model Coupling에 대해서
최종 수정일 2022. 11. 27



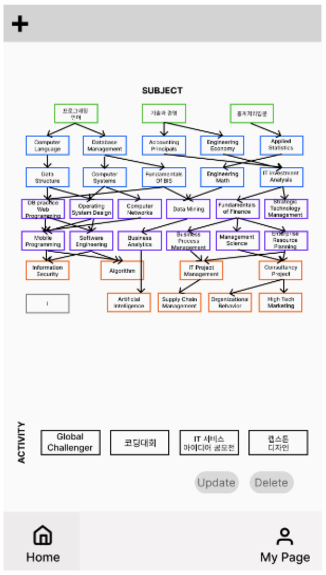
Setting

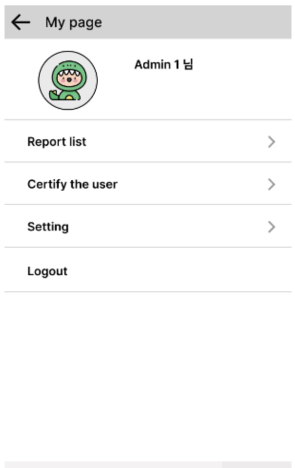
앱 설정

Notification Setting

소프트웨어 버전 확인

Configuration 버전 1.2





My page

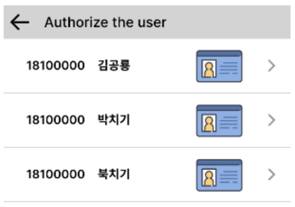
Admin 1 님

Report list

Certify the user

Setting

Logout

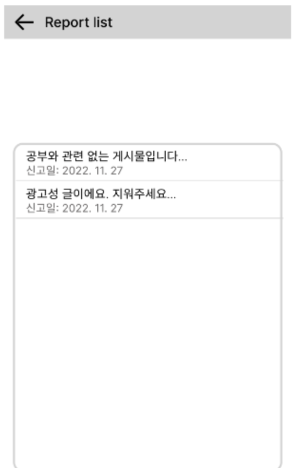


Authorize the user

18100000 김공룡

18100000 박치기

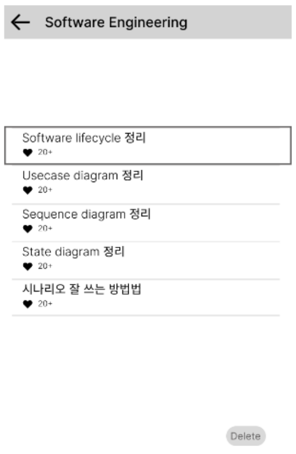
18100000 북치기



Report list

공부와 관련 없는 게시물입니다...
신고일: 2022. 11. 27

광고성 글이에요. 지워주세요...
신고일: 2022. 11. 27



Software Engineering

Software lifecycle 정리
♥ 20+

Usecase diagram 정리
♥ 20+

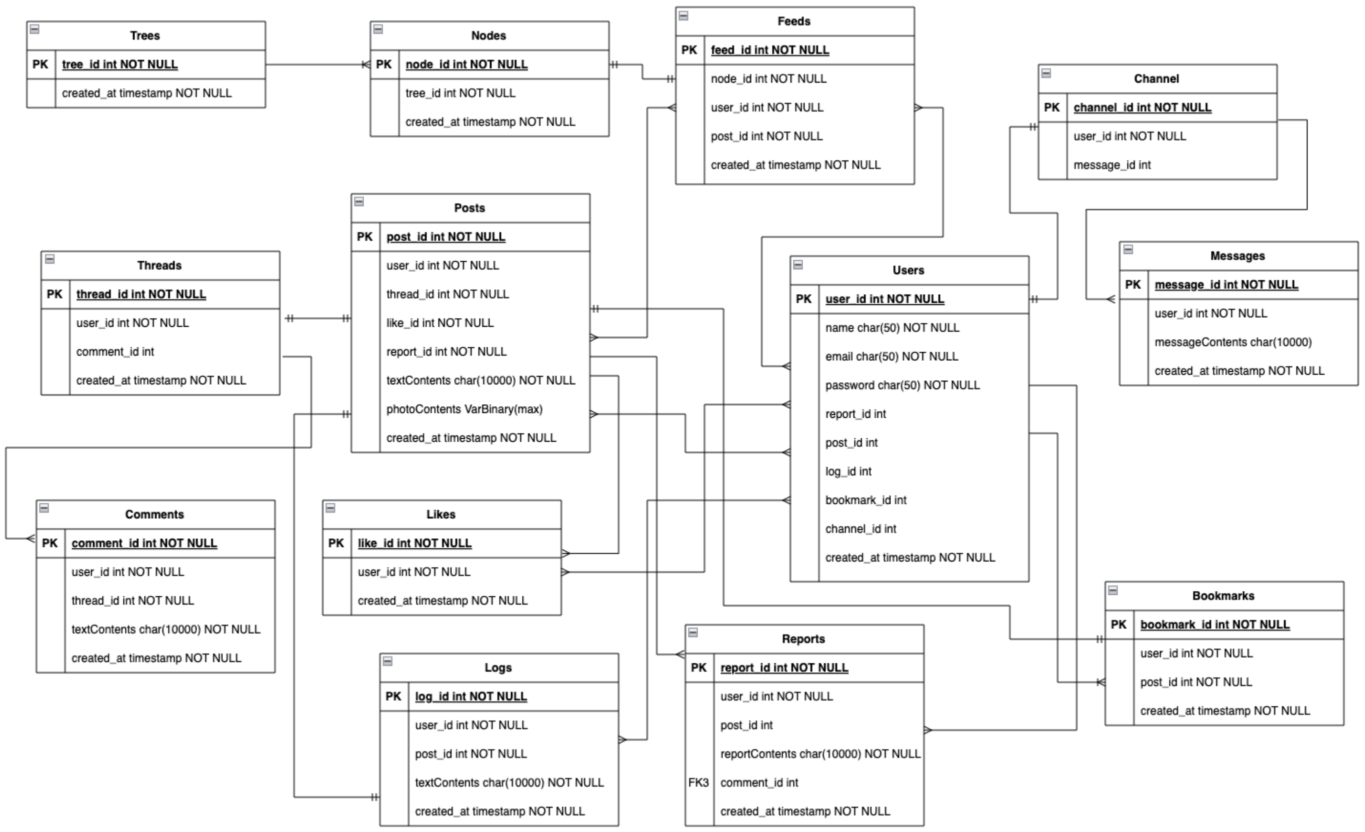
Sequence diagram 정리
♥ 20+

State diagram 정리
♥ 20+

시나리오 잘 쓰는 방법법
♥ 20+

Delete

VII. DataBase Design Diagram


Trees
PK tree_id int NOT NULL
created_at timestamp NOT NULL
Nodes
PK node_id int NOT NULL
tree_id int NOT NULL
created_at timestamp NOT NULL
Feeds
PK feed_id int NOT NULL
node_id int NOT NULL
user_id int NOT NULL
post_id int NOT NULL
created_at timestamp NOT NULL
Channel
PK channel_id int NOT NULL
user_id int NOT NULL
message_id int
Posts
PK post_id int NOT NULL
user_id int NOT NULL
thread_id int NOT NULL
like_id int NOT NULL
report_id int NOT NULL
textContents char(10000) NOT NULL
photoContents VarBinary(max)
created_at timestamp NOT NULL
Threads
PK thread_id int NOT NULL
user_id int NOT NULL
comment_id int
created_at timestamp NOT NULL
Users
PK user_id int NOT NULL
name char(50) NOT NULL
email char(50) NOT NULL
password char(50) NOT NULL
report_id int
post_id int
log_id int
bookmark_id int
channel_id int
created_at timestamp NOT NULL
Messages
PK message_id int NOT NULL
user_id int NOT NULL
messageContents char(10000)
created_at timestamp NOT NULL
Comments
PK comment_id int NOT NULL
user_id int NOT NULL
thread_id int NOT NULL
textContents char(10000) NOT NULL
created_at timestamp NOT NULL
Likes
PK like_id int NOT NULL
user_id int NOT NULL
created_at timestamp NOT NULL
Bookmarks
PK bookmark_id int NOT NULL
user_id int NOT NULL
post_id int NOT NULL
created_at timestamp NOT NULL
Reports
PK report_id int NOT NULL
user_id int NOT NULL
post_id int
reportContents char(10000) NOT NULL
FK3 comment_id int
created_at timestamp NOT NULL
Logs
PK log_id int NOT NULL
user_id int NOT NULL
post_id int NOT NULL
textContents char(10000) NOT NULL
created_at timestamp NOT NULL