



SKKUnite

Software Requirements Specification
Introduction to Software Engineering
2023

Team 14

Team Leader	Dean Hunt
Team Member	Anni Jusslin
Team Member	Melody Jansen
Team Member	Noémie Zanzucchi
Team Member	Shakhzodakho Olimkhujaevan

Contents

1. Introduction.....	4
1.1. Purpose.....	4
1.2. Scope of the application.....	4
2. Overall description.....	5
2.1. Product Perspective.....	5
2.1.1. Market Status.....	5
2.1.2. Overall Structure.....	5
2.2. Product Functions.....	6
2.2.1. User Registration.....	6
2.2.3. Matching algorithm.....	6
2.2.4. Connection Establishment.....	6
2.2.5. Conversation Features.....	6
2.2.6. User Dashboard.....	7
2.2.7. Reporting and Moderation.....	7
2.3. User Classes and Characteristics.....	7
2.3.1. User.....	7
2.3.2. System Manager.....	8
2.4. Operating Environment.....	8
2.4.1. Hardware.....	8
2.4.2. Software.....	8
2.5. Design and Implementation Constraints.....	8
2.6. User Documentation.....	9
2.7. Assumptions and Dependencies.....	9
3. Specific Requirements.....	9
3.1. External Interface Requirements.....	9
3.1.1. User Interfaces.....	9
3.1.2. Hardware Interfaces.....	24
3.1.3. Software Interfaces.....	24
3.1.4. Communication Interfaces.....	24
3.2. Functional Requirements.....	25
3.2.1. Use Case.....	25
3.2.2. Use Case Diagram.....	35
3.2.3. Data Dictionary.....	35
3.2.4. Data Flow Diagram.....	39
3.3. Performance Requirements.....	39
3.3.1. Static Numerical Requirement.....	39
3.3.2. Dynamic Numerical Requirement.....	39
3.4. Logical Database Requirements.....	40

3.5. Design Constraints.....	40
3.6. Programming Convention Compliance.....	40
3.7. Software System Characteristics.....	40
3.7.1. Product Requirements.....	40
3.7.2. External Requirements.....	41
3.8. Organizing the Specific Requirements.....	41
3.8.1. Context Model.....	41
3.8.2. Process Model.....	41
3.8.3. Interaction Model.....	42
3.8.4. Behavior Model.....	42
3.8.4.1. Data Flow Diagram.....	42
3.8.4.2. Sequence Diagram.....	42
3.9. System Evolution.....	43
3.9.1. Assumptions.....	43
3.9.2. Anticipated Changes.....	43
4. Supporting Information.....	44
4.1. Software Requirement Specification.....	44
4.2. Document History.....	44

1. Introduction

1.1. Purpose

The following Software Requirements Specification document describes the SKKUnite application. This application is designed and implemented by Team 14 of the Introduction to

Software Engineering course at Sungkyunkwan University. This team consists solely of international student members. Ultimately, SKKUnite is a platform to connect the international students at SKKU with the Korean students. In doing so, we hope to bridge cultures, foster new friendships, and enhance the global educational experience. SKKUnite is a web application, as this is easily accessible for all students and requires no installation from an app store. In this document, we will provide the Requirement Specification. Chapter 2 of this document will look at the application itself. Here, a detailed overview of what can be expected of the application will be given. Furthermore, the specific requirements for the application will be covered in chapter 3. These requirements are comprehensively explained and organized with a number of models. This chapter also includes the system architecture and evolution. Chapter 4 will provide any supporting information, comprising the Software Requirement Specification itself and the document history. That will conclude the Software Requirements Specification document.

1.2. Scope of the application

SKKUnite is an application meant specifically for students at Sungkyunkwan University. The app is meant for both international students and Korean students. A user should first register their account using their SKKU credentials. To create a safe and reliable environment, only students registered with an SKKU account can register for SKKUnite. At registration, it is noted whether the user is an international or a Korean student at SKKU. Once users have an account, they are able to edit their personal profile to make sure we have a clear image of the user. Using the information gathered from the personal profile and the information about whether the user is an international or Korean student, the user is matched to another user. International students are matched to Korean students and vice versa. Once a match has been established, a random conversation starter is given by the app to initiate a chat. This conversation starter will be in the form of a question or statement. At some point, both users are asked to review their experience with the other user. However they rate their experience with the other experience, that will be taken into account with the matching process. Through this process, the application attempts to carefully establish new connections between people that will suit each other's personal profile.

2. Overall description

2.1. Product Perspective

2.1.1. Market Status

The market status of the SKKUnited platform is somewhat difficult to determine, as it is only available to SKKU students and exchange students. On the other hand, the idea of this app could grow in popularity in the platform market among other universities and also outside South Korea. Other developers could also possibly get ideas from this platform on how to develop their own existing platforms. Other developers may be interested in the fact that such learning platforms can boost students' learning and also their motivation to study. It is precisely these factors that could be a good market status for this platform.

2.1.2. Overall Structure

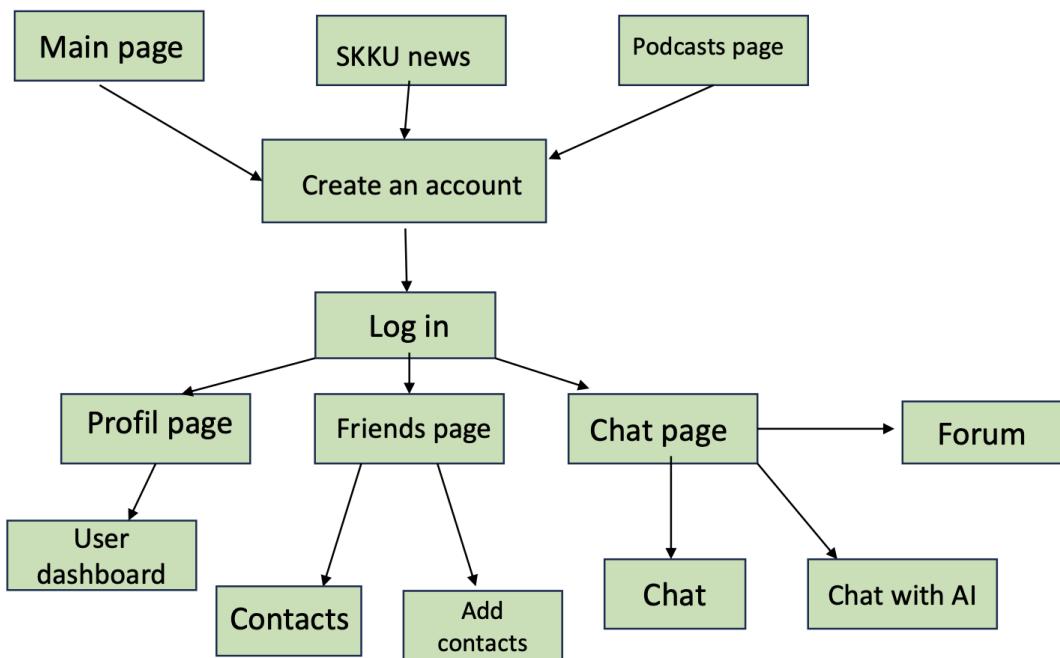


Figure 1: System structure

2.2. Product Functions

2.2.1. User Registration

Users can register with their university email addresses. They need to fill in their first and last name and also create a unique username. The password has to be filled out twice to

make sure that it is correct both times. They have to fill out their student ID so that the application can identify them as international students or Korean students. The user can press the “Create account” button when all the required fields have been filled in. If the user already has an existing account, they can log in under "log in" by entering their username and password.

2.2.2. User Profile

Users can create and edit profiles. Editing your profile is done through Profile page. You can edit your name, password, courses, profile picture and major. The courses and major can only be edited once a month. The year of study will be updated automatically. Profiles include information such as name, profile picture, major, interests, country, courses you have and year of study.

2.2.3. Matching algorithm

The algorithm takes into account people's majors, year of study and courses so that students have at least one course in common. This matching process occurs once a week so students have time to contact their study partners and start a conversation. If they have not started a conversation within a week, the match will be deleted.

2.2.4. Connection Establishment

The algorithm calculates the similarities between users by course, year and major to match users with each other. The matched users can be seen on the personal contacts page and the user can choose whether or not to add a particular user. They can start chatting only with a connected match.

2.2.5. Conversation Features

SKKUnite includes a real-time chat function with real people and also with AI. Users can write the message to the real-time chat and send it by pressing the “send” button. Users can't see whether the other person has read the message or not. AI chat has pre-generated questions that the user can ask the AI. The user can choose whether to use them or not. AI can logically answer questions mainly related to the content of the application and the world of learning. For other topics, the quality of answers may vary. Users will always get

a notification when a new message is received whether it is from a real person or AI. Also a notification will appear on a user's phone screen.

In the discussion forum, students can create their own discussions, for example to ask for help on a general issue related to the university, courses, etc. To create a discussion, click on the "create topic" button in the bottom corner. Students can view and participate in discussions by clicking on the "open" button next to the forum topic.

2.2.6. User Dashboard

The dashboard allows the user to see a summary of everything. They can see their username and profile picture. Users can also see an overview of the ongoing conversations in which the user is involved and the current connections (up to a maximum of 5 are shown here on the dashboard). In addition to these, the dashboard shows if the user has any new unanswered messages.

2.2.7. Reporting and Moderation

A user can report another user who appears to be inappropriate in chat by pressing the "report user" button in the top left corner.

2.3. User Classes and Characteristics

2.3.1. User

Users of this system are limited to students at SKKU only. It is assumed a user is capable of reading and speaking either basic Korean or basic English. It is also assumed that a user is interested in connecting with students from a different culture through this platform. Basic knowledge of using a web platform must be assumed as well. The user has access to all the functions of the platform after logging in. The user has access to the main page and podcasts even when not logged in.

2.3.2. System Manager

The system manager can manage the system and its code. The manager can make fixes and updates to the user interface. The system manager can also remove inappropriate users from the system. The manager also receives feedback from moderators and monitors the technical performance of the system and detects possible flaws. System manager must have the

necessary skills, for example through education, to be able to manage the different parts and functions of the software correctly. They must have enough knowledge of the whole system so they can manage it. They must also be able to deal with potential error situations so that they can be resolved.

2.4. Operating Environment

2.4.1. Hardware

The system is best suitable for web browsers, especially Google Chrome. The system needs at least 1.5GB RAM and a single processor with 1.0GHz.

2.4.2. Software

This system can be used with a web browser e.g. Google Chrome, Internet Explorer, Mozilla Firefox etc. Google Chrome is the best option for this system.

2.5. Design and Implementation Constraints

Limitations related to software design may arise from device operation limitations. Screen size can affect the flexibility of the software design i.e. screen can look a bit different compared to mobile phone screen. There also might be some technical constraints between different web browsers. Naturally, a poor internet connection can also cause the page to load slowly and some elements to not work.

The following software and programming languages have been used for this software: Figma, MongoDB, Python, Javascript, HTML and CSS. If one of these is missing or the team does not have the skills, it will be difficult to implement this platform.

It is required to have good skills in English because this system's main language is going to be English. This system is aimed for Korean and international students, so English is the most optimal language for both the system and its development.

2.6. User Documentation

User manual is necessary for end users of this system. It should contain a basic installation guide for users which would tell how to install this software and how to use different features. The instructions would include both pictures and text.

2.7. Assumptions and Dependencies

This system is designed for web browser, so all the functions will work best with using web browser. The best browser for this platform is Google Chrome. It will work with Internet Explorer and Mozilla Firefox, but Google Chrome guarantees the best performance.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

Table 1: Welcome Interface

Name	Welcome
Purpose/Description	Users transmit their instructions by clicking the corresponding button
Input source/Output destination	User/User device
Range/Accuracy/Margin of error	Range according to the number of buttons on the screen/ Accuracy according to the precision of interaction with on-screen elements/ Margin of error of precision of mouse and keyboard input
Unit	A mouse click
Time/Velocity	Asynchronous user input/Instant execution of a user instruction
Relationship with other input/outputs	Once all inputs have been received, the user's device forwards the input data to the server for processing and calls for the necessary output data.
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none">1. A screen predominantly featuring text and images.2. The user interface includes multiple buttons prepared to capture user inputs.3. Users are expected to select the appropriate button to engage with the system.

	
Data type	Screen
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 2: Register Interface

Name	Register/Log in
Purpose/Description	Users can log in or register
Input source/Output destination	User/User device and server
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> Two columns of empty text boxes to be used for filling out user information necessary for log in or registration. Text boxes in the left column are for log-in, text boxes in the right column are for registration. At the bottom of each column is a button. Underneath the left column is a 'Log in' button. Underneath the right column is a 'Register' button, which sends the information entered in the text boxes to the server After successful log-in or registration, users are redirected to the Welcome interface

	
Data type	Query
Instruction type	Instruction mapping according to the value of a button code
Exit message	'Log-in successful!' 'Registration successful!'

Table 3: User Profile Interface

Name	User Profile
Purpose/Description	After log-in/registration, users may view their 'User Profile' page by clicking the 'Profile' button. Users can also go on to edit their profile by clicking the 'Edit profile' button.
Input source/Output destination	User/server
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> Users can view the display of their profile page with some personal information and a picture. Users can go to the 'Edit profile' page by pressing the 'Edit profile' button. Users can leave the profile page by scrolling up to see the menu and clicking the button for any desired page to enter.

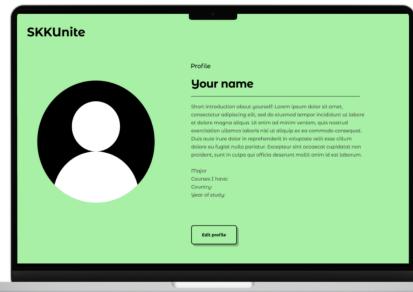
	
Data type	Text, Image
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 4: Contacts List Interface

Name	Contacts List
Purpose/Description	Users can view their list of contacts and ongoing chats by entering through the ‘Contacts List’ button in the menu.
Input source/Output destination	Server/Client
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> Two column layout with the left column displaying a list of all contacts and the right column displaying a list of all ongoing chats By clicking on the user’s profile picture or on the user’s name, the user can enter the corresponding profile page A user can delete a chat by clicking the red “Delete chat” button A user can open the chat by clicking the green “Open chat” button. A user can leave the contacts list page by scrolling up to see the menu and clicking the button for any desired page to enter.

	
Data type	Image, Text, Widget
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 5: Chat Interface

Name	Chat
Purpose/Description	Users can enter a chat by starting one after receiving a friend recommendation or enter through the friend list page.
Input source/Output destination	Client/Server
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. TextView of name of recipient and messages sent from both user and recipient. 2. Users can see the profile of the recipient by clicking the profile picture or name of recipient. 3. Users can send messages by typing into the text box and clicking the 'Send' button. 4. Users can send audio messages by clicking on the round button with a microphone icon, recording audio, clicking on the same icon again, and then clicking the 'Send button'. If audio recording should not be sent, then the user should press Backspace key on

	<p>keyboard.</p> <ol style="list-style-type: none"> 5. Users can report a user by clicking the red ‘Report user’ button. 6. Users can exit chat by scrolling up to see the menu and clicking the button for any desired page to enter. 
Data type	Image, Text, Widget
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 6: Chat With AI Interface

Name	Chat With AI
Purpose/Description	Users can enter the Chat With AI page by clicking the ‘Chat with AI’ button in the menu. Users can chat with AI to ask questions about SKKUnite web application functionality and general school life.
Input source/Output destination	Server/Client Client/Server
Range/Accuracy/Margin of error	Range according to user input queries Accuracy based on correctness of AI response Margin of error regarding likelihood of incorrect responses
Unit	Screen
Time/Velocity	Asynchronous user input/Communication time of a user’s query
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. Users can view messages sent by AI and send messages to the AI by typing in the text box and clicking the ‘Send’

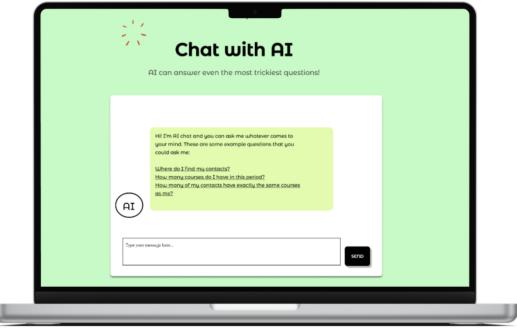
	<p>button.</p> <ol style="list-style-type: none"> 2. Users can leave the chat with AI page by scrolling up to see the menu and clicking a button for any desired page to enter. 
Data type	Text, Widget, Query
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 7: Recommendation Interface

Name	Recommendation
Purpose/Description	Users can enter recommendations by clicking 'Friend recommendations' button. Users can view and add recommended users whom the algorithm has recognized as possible friends for the user.
Input source/Output destination	Server/Client
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. Display of two columns with users recommended by system algorithm as potential friends for a user. 2. Users can view user profiles of recommended users by clicking the profile picture or name of recommended user. 3. Users can add recommended users by clicking the green '+ADD' button.

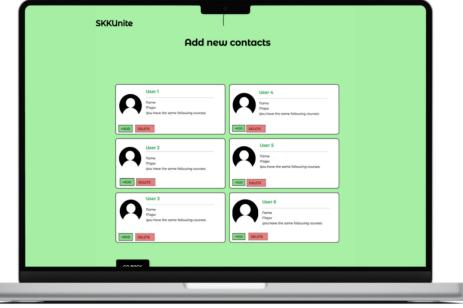
	<p>4. Users can delete recommended users from recommended users list by clicking the red ‘DELETE’ button.</p> <p>5. Users can leave the recommended users page by scrolling up to see the menu and clicking a button for any desired page to enter.</p>
	
Data type	Image, Text, Widget
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 8: User Preference Survey Interface

Name	User Preference Survey
Purpose/Description	Users will be prompted with the user preference survey after a conversation. Users may choose to skip the survey. Survey is used to adapt user preference in the system algorithm for recommending users as potential friends.
Input source/Output destination	Client/Server Server/Client
Range/Accuracy/Margin of error	Range according to survey questions Accuracy based on clarity and relevance of AI responses Margin of error regarding potential for misinterpretation of user responses
Unit	Screen
Time/Velocity	Asynchronous user input/Communication time of a user response
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	1. Users are prompted with survey to

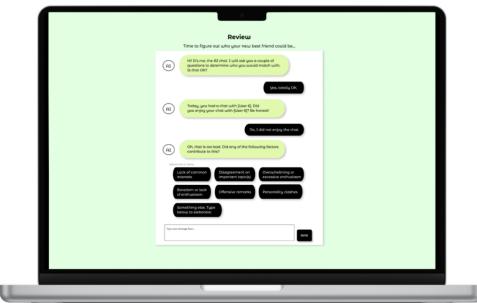
	<p>review a conversation after a conversation. Users may choose to skip or partake in the survey by replying to the AI with one of fixed response options displayed as buttons: ‘Yes, totally OK.’ or ‘Not right now.’</p> <ol style="list-style-type: none"> 2. Users may reply to AI questions by clicking any of the buttons containing response options or by typing into the text box and clicking the ‘Send’ button. 3. Users may end chat anytime by scrolling up to see the menu and clicking a button for any desired page to enter. Survey responses will be saved in server and used for system algorithm. 
Data type	Text, Widget
Instruction type	Instruction mapping according to the value of a button code
Exit message	‘Thank you. Your response will be used to update friend recommendations.’

Table 9: Podcast Interface

Name	Podcast
Purpose/Description	Users can access the Podcast Page by clicking the ‘Podcast’ button in the menu. Users can use the Podcast Page to browse and listen to various podcasts related to SKKUnite, school life, and other relevant topics.
Input source/Output destination	Client/Server
Range/Accuracy/Margin of error	Range according to number of podcasts Accuracy based on loading podcast content Margin of error based on upload errors
Unit	Screen
Time/Velocity	Podcast loading time/Speed at which podcast episodes can be played

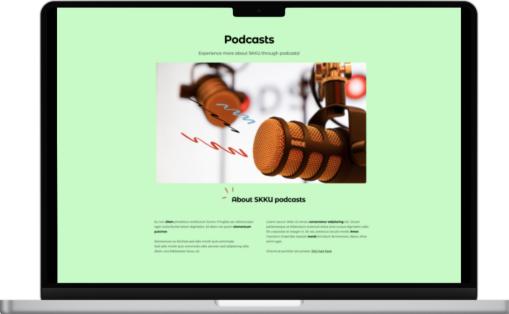
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. Users can view podcasts with information about SKKUnite and other school related items. 2. Users can exit the Podcast Page by clicking a button in the menu for any desired page to enter. 
Data type	Text, Image, Widget, Audio
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 10: About SKKUnite Interface

Name	About SKKUnite
Purpose/Description	Users can view basic information about SKKUnite
Input source/Output destination	Server/Client
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. Users can view information about SKKUnite. 2. Users can exit the About SKKUnite page by clicking a button in the menu for any desired page to enter.

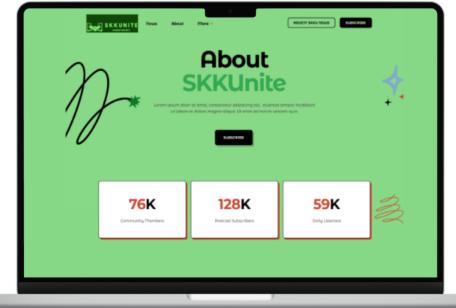
	
Data type	Image, Text
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 11: About SKKUnite Team Interface

Name	About SKKUnite Team
Purpose/Description	Users can enter the About SKKUnite Team page by clicking the 'About SKKUnite Team' button in the menu. Users can view information about the SKKUnite Team.
Input source/Output destination	Server/Client
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> Two columns list the names and profile pictures of the members of the SKKUnite Team. Profiles of SKKUnite Team members can be viewed by users by clicking the name or profile picture. Users can add SKKUnite Team members to their friend list by clicking the green '+ADD' button. Users can exit the About SKKUnite page by clicking a button in the menu for any desired page to enter.

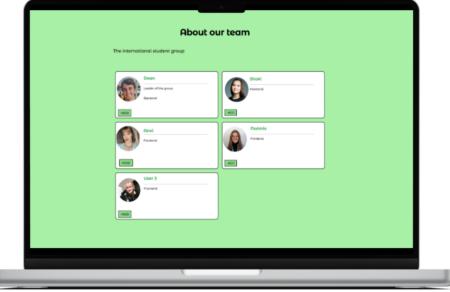
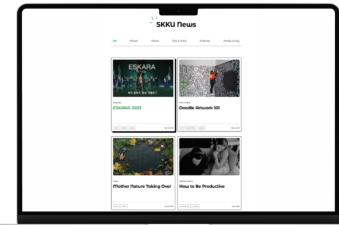
	
Data type	Image, Text, Widget
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 12: SKKU News Interface

Name	SKKU News
Purpose/Description	Users can enter the SKKU News page by clicking the SKKU News button in the menu. Users can view news articles.
Input source/Output destination	Server/Client
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. Users can observe SKKU News article headlines displayed in rows with some image 2. Users can read more about a headline or image by clicking the box in which the headline or image is situated. 3. Users can exit the SKKU News page by scrolling up and clicking a button in the 

	main menu for any desired page to enter.
Data type	Text, Image
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

Table 13: SKKU News Article Interface

Name	SKKU News Article
Purpose/Description	Users can enter the SKKU News Article page by clicking the article in the SKKU News page. Users can view SKKU News Article information.
Input source/Output destination	Server/Client
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. Users can view the headline and image for the SKKU News article and read the article itself. 2. Users can exit the SKKU News Article page by clicking a button in the menu for any desired page to enter. 
Data type	Image, Text
Instruction type	Instruction mapping according to the value of a button code

Exit message	N/A
--------------	-----

Table 14: Contact For Help Interface

Name	Contact For Help
Purpose/Description	Users can enter the Contact For Help page by clicking the ‘Contact’ button in the menu. Users can reach out to SKKUnite for questions.
Input source/Output destination	Client/Server
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> Text boxes for user information (full name, email, phone number, subject of question) are displayed which a user must fill out in order to send contact for help form. Large text box ‘Your message goes here...’ for the question itself. Users must click the ‘Send message’ button below text boxes to send the question to the server. Users can exit the Contact For Help page without sending any message by scrolling up and clicking a button in the menu for any desired page to enter. 
Data type	Text
Instruction type	Instruction mapping according to the value of a button code
Exit message	‘Message has been sent!’

Table 15: Forum Interface

Name	Forum
Purpose/Description	Users can enter the forum by clicking the ‘Forum’ button in the menu. Users can inquire about SKKU-related items on the forum and help other users.
Input source/Output destination	Server/Client Client/Server
Range/Accuracy/Margin of error	N/A
Unit	Screen
Time/Velocity	N/A
Relationship with other input/outputs	N/A
Format and configuration of screen	N/A
Format and configuration of window	<ol style="list-style-type: none"> 1. A display of all forum questions displayed in one column. 2. Users can view more information about the question, observe any potential responses so far and upload a response to a question by clicking the ‘OPEN’ button. 3. Users can pose a question themselves by clicking ‘CREATE TOPIC’ and then typing into the text boxes on the next page. 4. Users can exit the Forum page by scrolling up and clicking a button in the menu for any desired page to enter. 
Data type	Text
Instruction type	Instruction mapping according to the value of a button code
Exit message	N/A

3.1.2. Hardware Interfaces

Table 16: Hardware Interface

Name	Applicable device for the system
Purpose/Description	Enable users to access the web application through web browsers on devices with internet connectivity, supporting a range of input methods, display resolutions, and hardware capabilities. The web application is optimized to run on devices with at least 1.5GB RAM and compatible with any processor of at least 1.0GHz.

3.1.3. Software Interfaces

Table 17: Software interface

Name	MongoDB Database
Purpose/Description	Managing multimedia/meta data through query input/output
Input source/Output destination	Host server/ User, User/Host server, User/User
Range/Accuracy/Margin of error	Depends on the performance of MongoDB
Unit	Query
Time/Velocity	High velocity, can store new data every few minutes
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

3.1.4. Communication Interfaces

Table 18: Communication interface

Name	Client and Host
Purpose/Description	Each client requests the connection to the host, requesting a list of recommendations for friends

	based on a personality analysis by the system algorithm and chat content with other users. Host provides a recommended friends list and chat content to users.
Input source/Output destination	User/Host server
Unit	Query
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

3.2. Functional Requirements

3.2.1. Use Case

<Account Management>

Table 19: Register Use Case

Use case	Register
Actor	Unregistered user
Description	Unregistered user tries to register for SKKUnite
Normal Course	<ol style="list-style-type: none"> 1. If the user is unregistered, then they must register for the service. 2. The user enters the “Log in or register” page. 3. The user enters their own first name in the “First name” field 4. The user enters their own last name in the “Last name” field 5. The user enters a username in the “Username” field. The username will be used later to log in. 6. The user enters their student id in the “Student ID” field. 7. The user enters their own school email in the “Password field” 8. The user fills out a password, in the “password” field.

	<p>9. The user verifies this password by typing it into the “Password again” field.</p> <p>10. The user presses the “Create account button”.</p>
Pre-condition	<p>The user is not yet registered in the system.</p> <p>The user has a valid SKKU email and student ID.</p> <p>The user enters valid information.</p> <p>There is no registered user who is already using the entered username, email address, or student ID.</p>
Post-condition	<p>The following information is added to the system account database, encrypted:</p> <ol style="list-style-type: none"> 1. The first name 2. The last name 3. The username 4. The email address 5. The hash value of the password
Assumptions	N/A

Table 20: Log-in/Log-out Use Case

Use case	Log-in/Log-out
Actor	Registered user
Description	<p>Logging in is the action taken by a registered user attempting to access the system</p> <p>Logging out is the action taken by a user who has previously logged in to exit the system</p>
Normal Course	<p><Log-in></p> <ol style="list-style-type: none"> 1. When a registered user intends to access the system's services. 2. The user provides the ID and password they entered during the registration process. 3. Upon successful verification of the provided information, the user can log in and utilize the system. <p><Log-out></p> <ol style="list-style-type: none"> 1. To exit the system, the user can simply select the log-out button, conveniently located in the menu. 2. Subsequently, when the user reopens the application at a later time, they will need to log in again. 3. In cases where the user closes the application without properly logging out, the system will automatically terminate the session.

Pre-condition	<Log-in> The user has a pre-existing registration in the system. <Log-out> The user is currently signed into the system.
Post-condition	The user should be logged onto/off the system.
Assumptions	N/A

Table 21: Profile Use Case

Use case	Profile
Actor	Registered user
Description	The process of a user viewing either their own profile or another user's profile.
Normal Course	<p><Viewing own profile right after log-in></p> <ol style="list-style-type: none"> 1. The user has just logged in and is brought to their profile face automatically. <p><Viewing own profile while on any random page on the web application></p> <ol style="list-style-type: none"> 1. The user clicks the 'My profile' button in the main menu in the top right corner of the screen. 2. The user is directed to their own profile. <p><Viewing another user's profile></p> <ol style="list-style-type: none"> 1. The user clicks on the profile picture or name of the other user. 2. The user is directed to the other user's profile.
Pre-condition	<p><Viewing own user's profile></p> <p>The user has a pre-existing registration in the system and the user is currently signed into the system.</p> <p><Viewing another user's profile></p> <p>The user has a pre-existing registration in the system and the user is currently signed into the system. The user is not blocked by the other user.</p>
Post-condition	The user can view the profile with the profile picture and an introductory text.
Assumptions	N/A

Table 22: Edit Profile Use Case

Use case	Edit Profile
----------	--------------

Actor	Registered user
Description	This is how individuals personalize their profile picture and introduction, expressing their personality. On the profile page, users have the opportunity to present themselves.
Normal Course	<ol style="list-style-type: none"> 1. To access the profile page, the user taps the profile button located in the main menu at the top of the screen 2. On the profile page, the user should click the 'Edit profile' button to be redirected to the editing page 3. On the editing page a user can change their profile picture and change their introduction. 4. After making changes, the user can save the modified information by tapping the 'save' button.
Pre-condition	The user has a pre-existing registration in the system and the user is currently signed into the system.
Post-condition	Upon clicking the 'save' button, the updated information is transmitted to the server database and subsequently refreshed.
Assumptions	N/A

<Friends management>

Table 23: Contacts Use Case

Use case	Contacts
Actor	Registered user
Description	The process of a user viewing their list of contacts.
Normal Course	<ol style="list-style-type: none"> 1. The user clicks the 'My contacts' button in the main menu at the top of the screen. 2. The user is directed to their contacts list.
Pre-condition	The user has a pre-existing registration in the system and the user is currently signed into the system.
Post-condition	The user can view their list of contacts.
Assumptions	N/A

Table 24: Add New Contacts Use Case

Use case	Add New Contacts
Actor	Registered user
Description	Users can initiate friend requests towards others, and reciprocally, other users can extend friend requests as well. The recipient has the choice to either accept or decline these requests from their list of received friend requests.
Normal Course	<ol style="list-style-type: none"> 1. A user will receive 10 friend recommendations as decided by the system algorithm. A user can look at the profiles of the recommended users by clicking their profile picture or name. 2. A user can then decide to send a friend request to any of the recommended users. 3. The user can also receive a friend request themselves. 4. The user who got a friend request can accept or refuse a friend request in his/her list of friend requests received. 5. After accepting a friend request, the user will be visible in the contacts list and a chat conversation may be started by either of the users. 6. After declining a friend request, the users will not see each other in the set of recommended friends anymore.
Pre-condition	<p>The user has a pre-existing registration in the system and the user is currently signed into the system.</p> <p><Sending friend request></p> <p>The user sending the friend request must not have sent the same request before and must not have been declined by the user before.</p> <p><Receiving friend request></p> <p>Another user must have sent a friend request to the user.</p>
Post-condition	<p><Accepting/declining a friend request></p> <p>A user will be visible as a contact in the contacts list or not</p>
Assumptions	N/A

Table 25: Delete Contact Use Case

Use case	Delete Contact
Actor	Registered user

Description	If a user wishes to remove a friend from their friend list, they can do so by looking the user up in their contacts list and deleting the user.
Normal Course	<ol style="list-style-type: none"> 1. The user has the ability to look up friends listed in their contacts list. 2. Subsequently, the user can initiate the deletion of a friend by clicking the red "Delete friend" button. 3. The system will inquire whether the user genuinely wishes to remove the friend, providing options for "yes" or "no." 4. Once deleted, the removed user becomes invisible in the friends list.
Pre-condition	The user has a pre-existing registration in the system and the user is currently signed into the system. The user must be friends with the user they wish to delete as a friend.
Post-condition	The removed user is no longer in the user's contact lists and nor is the user in the removed user's contact list.
Assumptions	N/A

<Chat and Forum>

Table 26: Open/Close Use Case

Use case	Open/Close
Actor	Registered user
Description	If a user wishes to start a conversation with a contact, they may open a chat. If a user wishes to continue a conversation with a contact, they may open a chat. If a user wishes to stop a conversation with a contact, they may close the chat.
Normal Course	<p><Open chat to start conversation></p> <ol style="list-style-type: none"> 1. The user finds the contacts list and looks for the contact they wish to start a conversation with. 2. The user has found the contact they wish to start a conversation with and clicks their profile picture or name. 3. The user finds the 'Chat' button on the profile page and clicks it. 4. The user is redirected to the empty chat, types a message into the text box and clicks 'Send'. <p><Open chat to continue conversation></p>

	<ol style="list-style-type: none"> 1. The user finds the contacts list and looks for the contact they wish to start a conversation with in the list of ongoing chats on the right side of the screen. 2. The user has found the contact they wish to continue a conversation with and clicks the green 'Open chat' button. 3. The user is redirected to the ongoing chat conversation. The user types a message into the text box and clicks 'Send'. <p><Close chat></p> <ol style="list-style-type: none"> 1. The user finds the contacts list and looks for the contact they wish to stop a conversation with in the list of ongoing chats on the right side of the screen. 2. The user has found the contact they wish to continue a conversation with and clicks the red 'Delete chat' button.
Pre-condition	<p><Open chat to start conversation></p> <p>The user has a pre-existing registration in the system and the user is currently signed into the system. The user must be friends with the user they wish to start a conversation with.</p> <p><Open chat to continue conversation></p> <p>The user has a pre-existing registration in the system and the user is currently signed into the system. There must be an ongoing conversation between the user they wish to continue a conversation with.</p> <p><Delete chat></p> <p>The user has a pre-existing registration in the system and the user is currently signed into the system. There must be an ongoing conversation between the user they wish to continue a conversation with.</p>
Post-condition	<p><Open chat to start conversation></p> <p>A chat has started between two users.</p> <p><Open chat to continue conversation></p> <p>A chat has continued between two users.</p> <p><Delete chat></p> <p>A chat has ended between two users. A user is redirected to a survey to answer some questions about their opinion on the past chat. The answers are utilized to find better friend recommendations.</p>
Assumptions	N/A

Table 27: Report User Use Case

Use case	Report User
----------	-------------

Actor	Registered user
Description	Users who have displayed behavior that is not in line with the community guidelines can be reported such that they are processed by the system administrator and suspended if the report was appropriate.
Normal Course	<ol style="list-style-type: none"> 1. To report a specific user, visit their profile. 2. On the profile page, locate and click the report button situated at the bottom right. 3. A report page will appear, enabling the user to detail the reasons for reporting. 4. Upon completing the report, click the report button to dispatch a report message to the administrator. 5. The system will reconfirm with the user to ensure their intent to report the specified user. 6. Once the system administrator reviews the reported message and validates the accuracy of the report, the account of the reported user may be suspended.
Pre-condition	The user has a pre-existing registration in the system and the user is currently signed into the system.
Post-condition	The system administrator receives the report.
Assumptions	N/A

Table 29: AI Chat Use Case

Use case	AI Chat
Actor	Registered user
Description	Users can chat with AI to ask questions about SKKUnite web application functionality and general school life.
Normal Course	<ol style="list-style-type: none"> 1. Users can view messages sent by AI and send messages to the AI by typing in the text box and clicking the 'Send' button. 2. Users can leave the chat with AI page by scrolling up to see the menu and clicking a button for any desired page to enter.
Pre-condition	The user has a pre-existing registration in the

	system and the user is currently signed into the system.
Post-condition	The AI chat and the user have a conversation.
Assumptions	N/A

Table 30: Create Topic

Use case	Create Topic
Actor	Registered user
Description	A user can create a topic on the forum to inquire about anything SKKU-related.
Normal Course	<ol style="list-style-type: none"> 1. A user opens the forum by clicking the 'Forum' button in the main menu at the top of the screen. 2. Users can pose a question themselves by clicking 'CREATE TOPIC' and then typing into the text boxes on the next page. 3. A user clicks the 'Post' button to upload the question to the forum for others to view.
Pre-condition	The user has a pre-existing registration in the system and the user is currently signed into the system.
Post-condition	The user has created a new topic on the forum.
Assumptions	N/A

<System Management>

Table 31: ID Verification Use Case

Use case	ID verification
Actor	System Administrator
Description	When a new user tries to sign up to the system, the user should submit his/her student id. Then the system administrator must check it is correct with the other information that has been submitted at registration.
Normal Course	<ol style="list-style-type: none"> 1. A new user provides their student ID and first and last name to the system at registration. 2. The system administrator reviews the student ID and cross-references it with the user's name, while also verifying if

	<p>there is any record of a user with the same information.</p> <ol style="list-style-type: none"> 3. If there are no issues found, the administrator grants approval for the user registration. 4. The system stores the student ID and name for future comparison.
Pre-condition	The new user has completed the registration and has submitted the correct student ID and name.
Post-condition	The user is verified as a student of SKKU and can log into the system.
Assumptions	The user provides correct information at registration.

Table 32: Process Reported Users Use Case

Use case	Process Reported Users
Actor	System Administrator
Description	If a user reports another user, the chat logs and forum contributions of the reported user are forwarded to the system administrator for verification of their behavior.
Normal Course	<ol style="list-style-type: none"> 1. In cases where a user reports another user due to behavior that goes against the community guidelines, any relevant information is shared with the system administrator. 2. The system administrator reviews the data associated with the reported user. 3. If the system administrator verifies the reported user as engaging in behavior that goes against the community guidelines, the user will face suspension from the system.
Pre-condition	The user the system administrator received a report about, must have been reported by a user.
Post-condition	If the system administrator verifies the reported user as engaging in behavior that goes against the community guidelines, the user will face suspension from the system.
Assumptions	N/A

3.2.2. Use Case Diagram

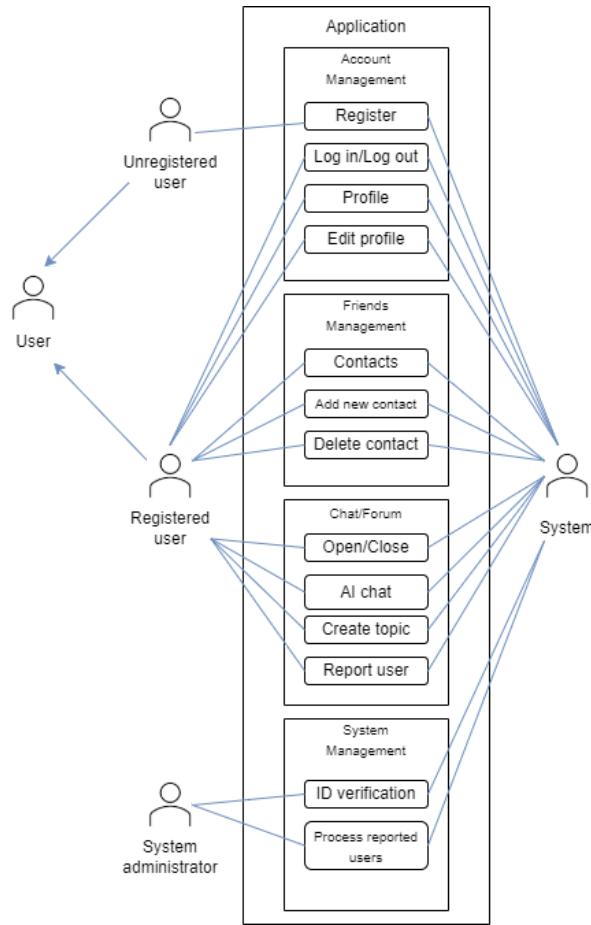


Figure 2: Use case diagram

3.2.3. Data Dictionary

Table 33: User data

Field	Key	Constraint	Description
id	PK	Not Null	User id, same as student id
email		Not Null	User email
first_name		Not Null	User's first name
last_name		Not Null	User's last name
password		Not Null	User password
introduction			User's introduction
major			User's major
courses			User's courses

country			User's country
year_of_study			User's year of study
suspended		Not Null	User being under suspension or not (true or false)
reported		Not Null	User having been reported or not (true or false)

Table 34: User's contact data

Field	Key	Constraint	Description
user_id	PK, FK	Not Null	User id
contact_id	FK	Not Null	User contact id
conversation_ongoing		Not Null	User having an ongoing conversation with contact (true or false)
contact_survey			Survey answers about contact

Table 35: User's chat data

Field	Key	Constraint	Description
user_id	PK, FK	Not Null	user id
conversation		Not Null	Chat content
time		Not Null	Time every message is sent
open		Not Null	Whether a chat conversation is currently ongoing or not (true or false)

Table 36: AI chat data

Field	Key	Constraint	Description
chat_id	PK	Not Null	AI chat id
user_id	FK	Not Null	User's id
time		Not Null	Time every message is sent

open		Not Null	Whether a chat conversation is currently ongoing or has been ended (true or false)
question		Not Null	User's question
response		Not Null	AI chat's response
suggested			Suggestions for user response during chat with AI

Table 37: Forum topics data

Field	Key	Constraint	Description
topic_id	PK	Not Null	Forum topic id
user_id	FK	Not Null	User id from user who created topic
time		Not Null	Time when topic was created
topic		Not Null	Title of topic
content		Not Null	Content of the forum topic

Table 38: Forum topic responses data

Field	Key	Constraint	Description
response_id	PK	Not Null	Forum response id
topic_id	FK	Not Null	Forum topic id
user_id	FK	Not Null	User id from user who posted forum topic response
time		Not Null	Time when forum topic response was posted
content		Not Null	Content of the forum topic response

Table 39: User report data

Field	Key	Constraint	Description
report_id	PK	Not Null	Report id

user_id	FK	Not Null	User id of user who is making the report
reported_user_id	FK	Not Null	User id of user who is being reported
time		Not Null	Time of report submission
report_reason		Not Null	Reason for report as described by reporting user
report_content		Not Null	Relevant item user was reported for (chat, forum topic, forum topic response)
conclusion			Report being accurate or not

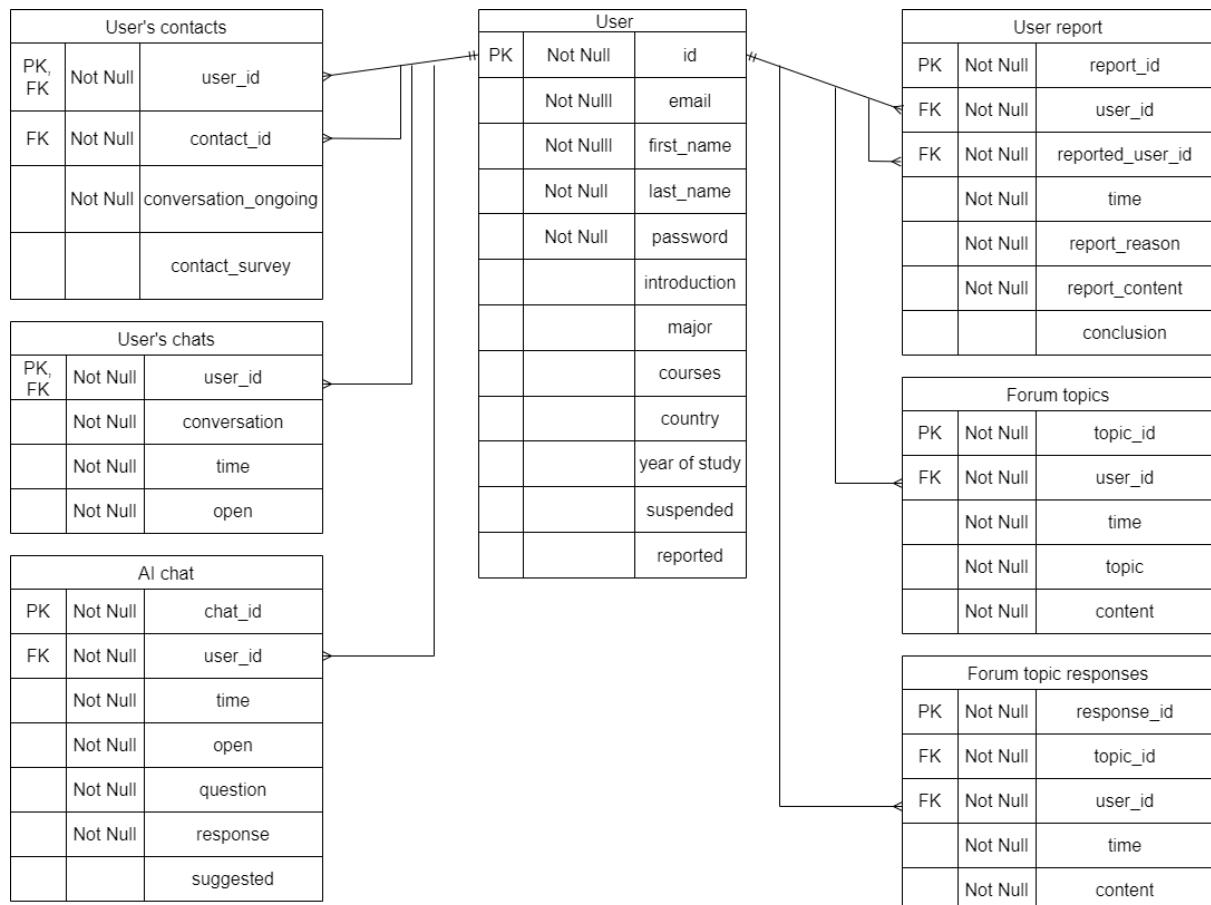


Figure 3: Entity relationship diagram

3.2.4. Data Flow Diagram

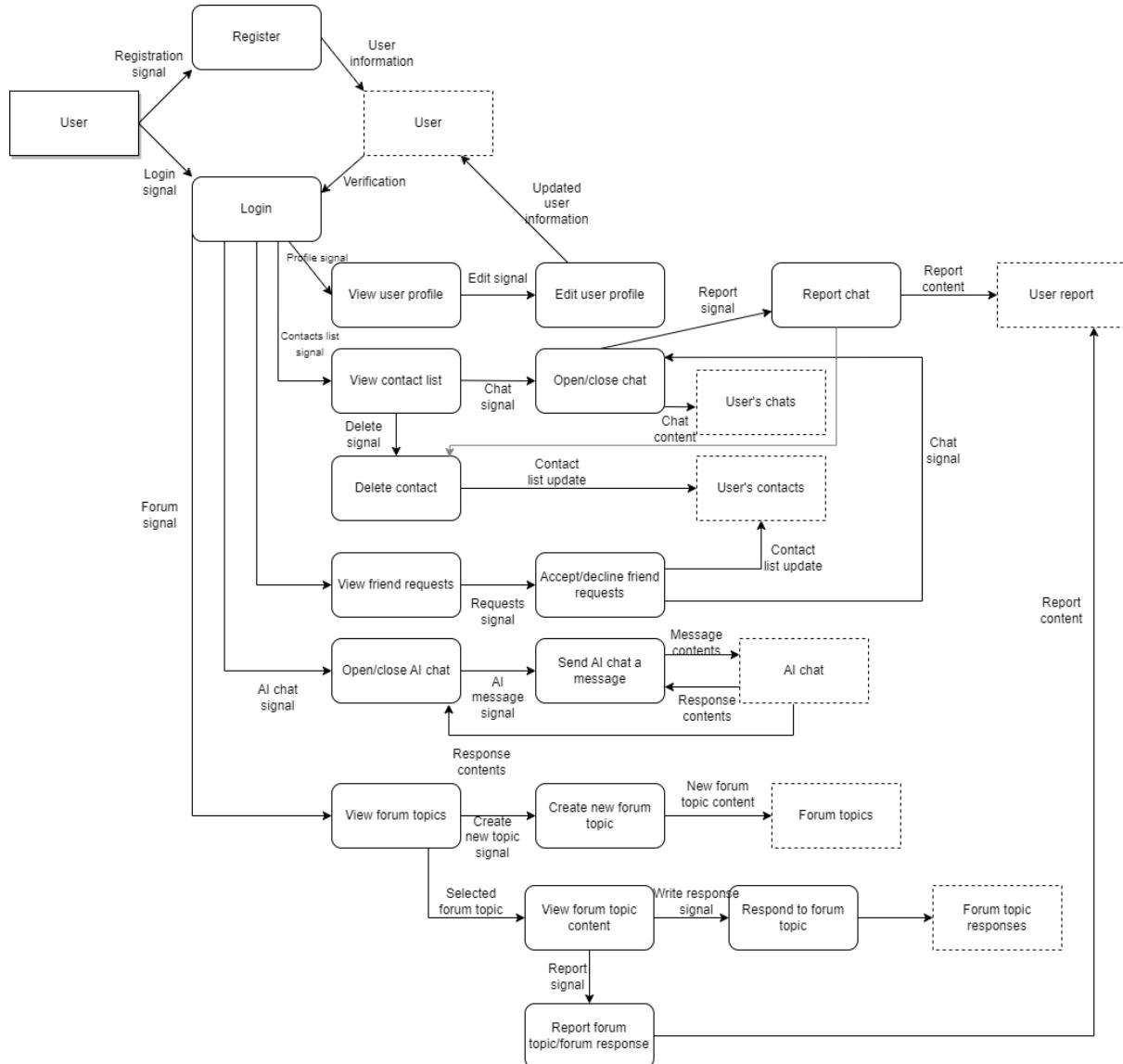


Figure 4: Data flow diagram

3.3. Performance Requirements

3.3.1. Static Numerical Requirement

- The system supports one concurrent user. After disconnecting, another user may log in and access their account.
- The web application is optimized to run on devices with at least 1.5GB RAM and compatible with any processor of at least 1.0GHz.

3.3.2. Dynamic Numerical Requirement

- The system is designed to run optimally with approximately 500 concurrent users.
- The system can handle 10000 members
- All mappings within the web application have a time limit of 10 seconds

3.4. Logical Database Requirements

The system utilizes a database named MongoDB to handle user information. The database stores user details. The database is carefully maintained to ensure efficient performance in processing information.

3.5. Design Constraints

When we didn't have any budget for our app development, it had a substantial impact on our project. We faced limitations in hiring skilled professionals, which slowed down development and forced us to reduce the features and functionality we had initially envisioned. Testing and quality assurance became a significant challenge, and we struggled with marketing and promotion due to the lack of financial resources. Scalability issues, security concerns, and difficulties in maintaining and updating the app also arose. We had to heavily rely on free tools and services, which often came with limitations. To address these challenges, we explored alternative funding sources and prioritized essential features in our development process.

3.6. Programming Convention Compliance

Ensuring programming convention compliance during our app development was essential for maintaining code quality and consistency. While we didn't have a budget, adhering to established coding standards was crucial for the long-term viability of the project. We couldn't afford expensive tools, but open-source linting and formatting tools helped us enforce coding guidelines. Regular code reviews among the team became even more critical to catch deviations from conventions early on. Although we faced budget constraints, our commitment to maintaining clean, consistent code helped us reduce future technical debt and improve collaboration among the development team.

3.7. Software System Characteristics

3.7.1. Product Requirements

The platform should be designed so that it can also be used by people who are not professionals (such as students, for example). It is desirable that users do not need any separate instructions or manuals in order to use the platform smoothly.

The performance of the system should also be as fast as possible. The algorithm that calculates user compatibility should complete the process within 5 seconds.

You must be an official student of Sungkyunkwan University to use this platform. They must have a registered student number to create an account for the application. Users must have their own unique username that no other user has. The password must be at least 8 characters long.

3.7.2. External Requirements

The system must be designed so that basic users cannot access the system in any conditions. The system must be protected so that no outside parties can damage it.

3.8. Organizing the Specific Requirements

3.8.1. Context Model

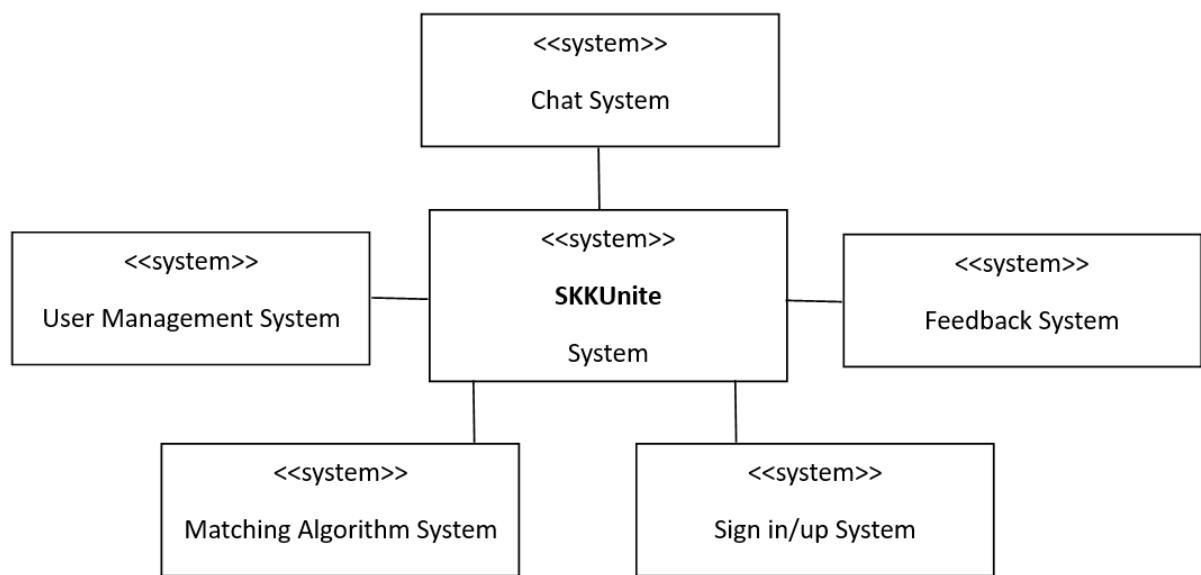


Figure 5: Context model

3.8.2. Process Model

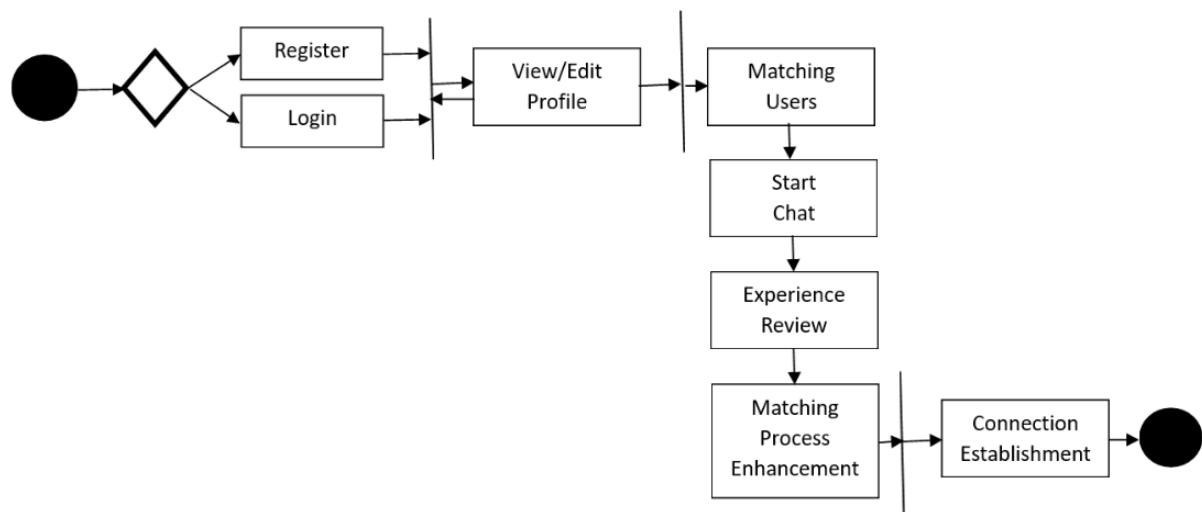


Figure 6: Process model

3.8.3. Interaction Model

See 3.2.2 Use Case Diagram

3.8.4. Behavior Model

3.8.4.1. Data Flow Diagram

See 3.2.4 Data Flow Diagram

3.8.4.2. Sequence Diagram

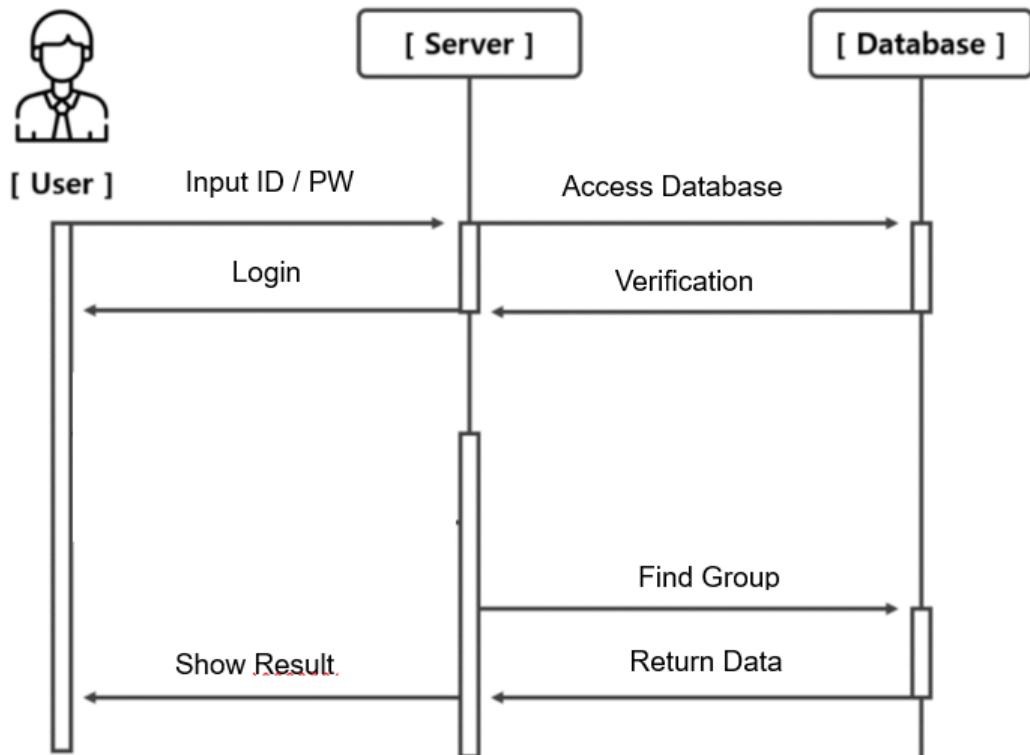


Figure 7: Sequence diagram

3.9. System Evolution

3.9.1. Assumptions

We are assuming that this application will be contained within SKKU, meaning that the amount of users of the app will remain relatively small. This is an important assumption, because our system architecture is not built to scale particularly well, so the functionality of our app relies on the assumption that the number of users will remain relative to the enrollment at SKKU.

3.9.2. Anticipated Changes

As the app is developed, we are not expecting many huge anticipated changes, unless certain university or legal regulations change drastically.

4. Supporting Information

4.1. 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).

4.2. Document History

Table 40: Document history

Version	Date	Description
1.0	2023-11-05	Initial draft of the SRS document