Java for Mobile Development

WeDiscuz - Real-time Video and Voice Communication for Academia

Zootopia:
Linpeng Lyu (linpengl)
Yilei Chu (ychu1)
Jialu Chen (Jialuc)

User Requirement Document

Version 2.0

	Prepared By	Inspected/Reviewed By	Approved By
Name:	Zootopia Team		
Title:	Software Design Team		
Signature:			
Date:	05/02/2016		

Version History

Version No.	Date	Changed By:	Changes Made:
1.0	03-28-2016	Zootopia Team	Initial requirement analysis of the project
2.0	05-02-2016	Zootopia Team	Refine some requirement and delete some use case

User Requirements Document

Table of Contents

Introduction

The object of this User Requirement Document is to clarify the quality and quantity aspects of the proposed project - WeDiscuz. The URD is to clearly communicate our understanding and requirement of the software to the professor and teaching assistants. This document contains less technical languages and is easy for users to read and understand.

Document Scope

The URD describes all functions provided by WeDiscuz in detail including what users can do with the app, what's the sequence of events triggered by users and when and how the users interact with external system. The URD also describes the operation and development environment for the Android application.

Intended Audience

The primary grader:

Professor and teaching assistants.

The developers:

All zootopia team members.

References

1. User Requirements Document - v2.0 - FreshBurger, Inc. Purchase Order Management Module of Centralized Inventory and Procurement Management System

System Overview

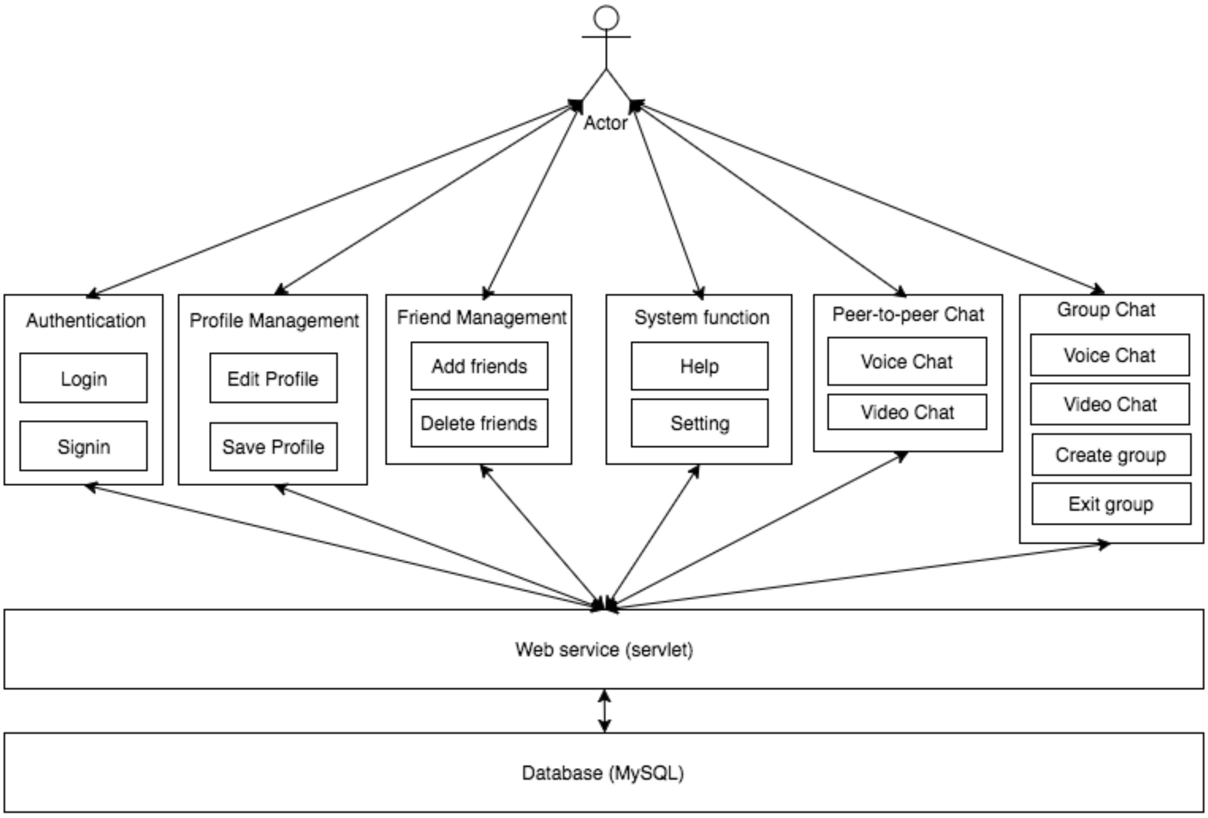

Figure 1. Overview of System

Purpose

This app aims to provide students and teachers at school a convenient way to communicate with each other and break the temporal and spatial limitation of discussion. The mobile-based app not only provide users with voice and video chat function, but also the whiteboard sharing function. When we try to explain some problems or theories to others, we always need to draw some pictures and show them to people. This function would definitely make it easier for users to share idea and thoughts.

Scope

The project WeDiscuz is an independent project and provide users with functions to manage friends, conduct voice or video chat with on or more people through smart phone whose operating system is android.

This app store user information and other data in the backend MySQL database. Each operation that need to use data would interact with the database. Furthermore, the app supports concurrent connection to multiple users and all operations should be done under synchronized mechanism.

General Requirements

The project is to provide user with a convenient and efficient way to discuss with others online. We provide voice and video chat functions to users and users can feel free to communicate with friends. Users can either start a peer-to-peer chat or room chat with friends. Furthermore, the application provides embedded calendar to remind people when an appointment is coming. Basic sign in and login functions are also provided.

System in Context

Students at university always need to discuss some academic problems with each other. In order to eliminate the spatial limitation, we intent to provide this great application in mobile to users and allow them to chat with each other easily. Furthermore, users can share whiteboard with others and make it much easier to communicate their idea to classmates.

User Characteristics

Student and teachers at school who always need to conduct group meeting and academic discussion with others. They may want to break the temporal and spatial limitation and talk with others at anytime and anywhere. They are well-educated and have strong ability to learn new things.

Constraints

A. Consistent login screens by users.
B. Authentication required.
C. The application can run on background and notify users when some request comes.
D. Easily used navigation bar.

System-Wide Requirements

Actor

The system only has one type of user and it might be student or teacher, or other kind of person that need to share idea with others. No actor division needed in this app.

Event

The app is focus on providing a real-time voice and video communication platform to users and some other efficient tools to help user share idea with others. There are several different events in this app: 1) add friends; 2) delete friends; 3) accept friends; 4) request peer-to-peer voice chat; 5) request peer-to-peer video chat; 6) request group voice chat; 7) request group video chat; 8) share whiteboard; 9) create event and set alarm

Detailed Requirements

Functional Requirements

User Requirements Model – Use Case Overview

Use Case No.	Use Case ID	Use Case Name	Priority
1	USER_SIG	User Sign in	High
2	USER_LOG	User Login	High
3	EDIT_PRO	Edit Profile	Medium
4	ADD_FRI	Add Friends	High
5	DEL_FRI	Delete Friends	Medium
6	PEER_VOC_CH	Peer Voice Chat	High
7	PEER_VID_CH	Peer Video Chat	High
8	GRO_VOC_CH	Group Voice Chat	High
9	GRO_VID_CH	Group Voice Chat	High
10	RE_VOC_CHAT	Restart Group Voice Chat	Medium
11	RE_VID_CHAT	Restart Group Video Chat	Medium
12	VIEW_PRO	View friends' profile	Medium
13	SE_FRIEND	Search friend by email	High
14	SET_UP_GROUP	Setup chatroom	High
15	HELP	Help	Low
16	SYS_SET	System setting	Lows
17	LOG_OUT	Log out	Medium

Table 1. Overview of Use Cases

User Requirements Model – Use Case Diagram

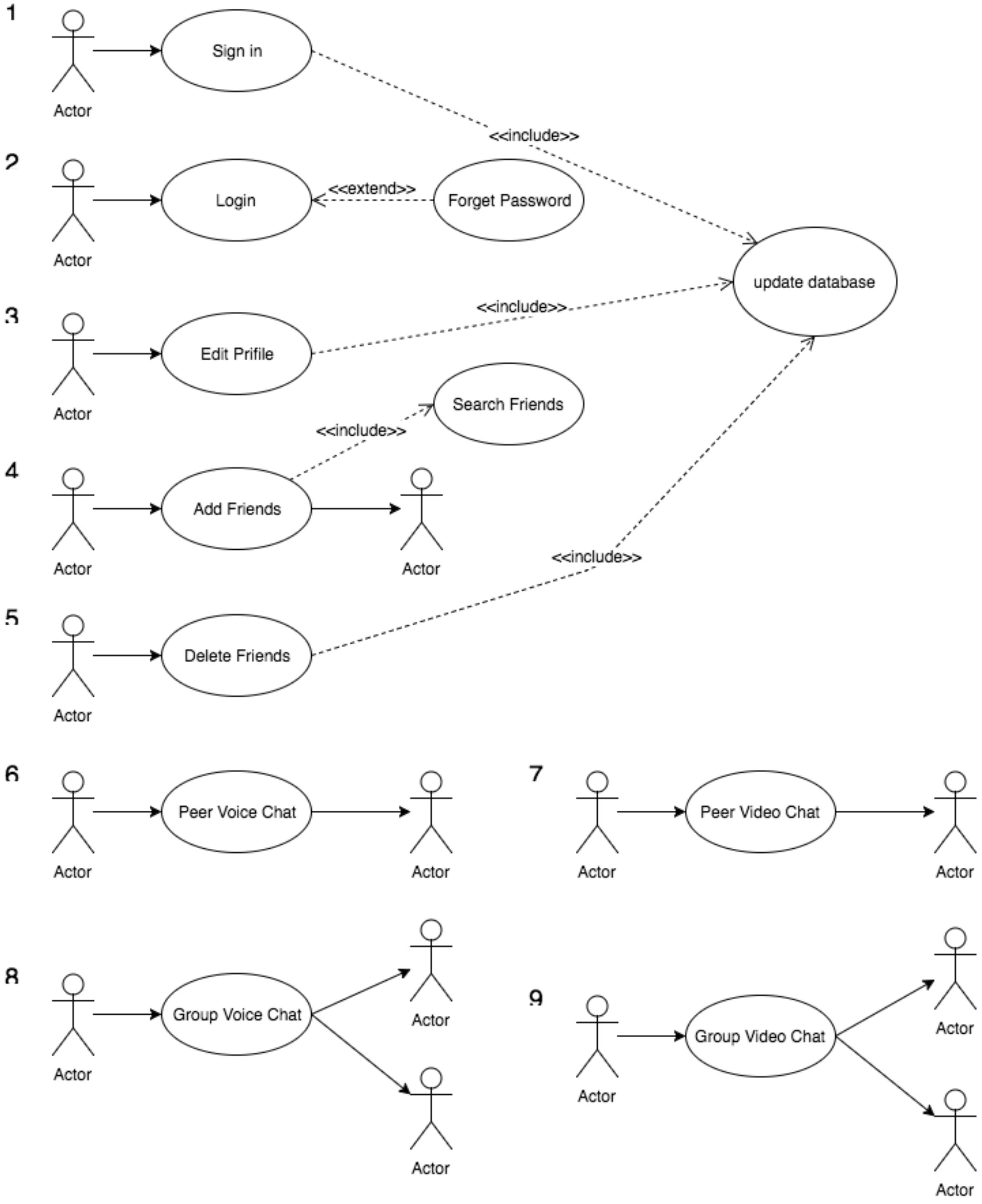

Figure 2. Use Case Diagram - Part 1

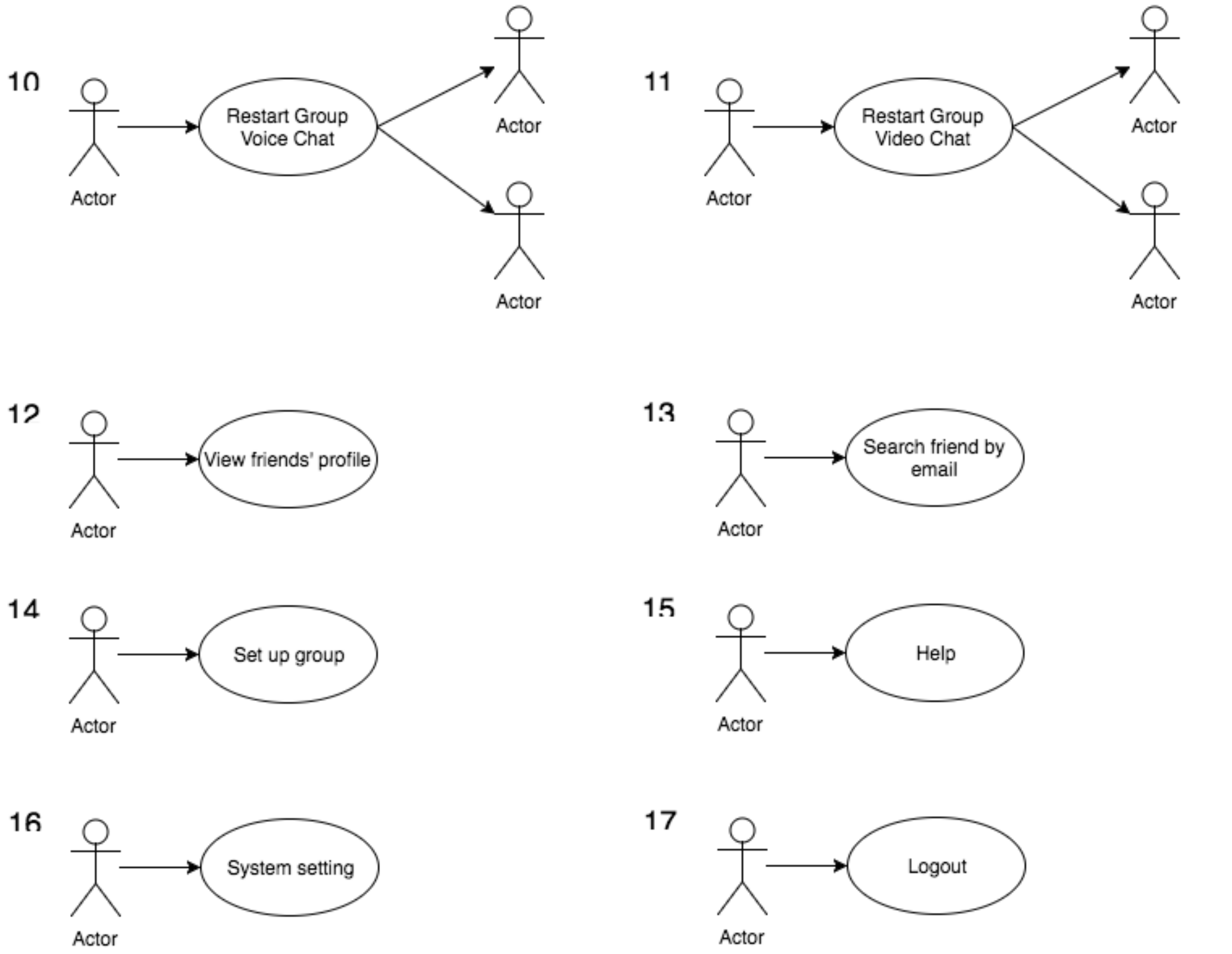

Figure 3. Use Case Diagram - Part 2

User Requirements Model – Detailed Use Case Specifications

Use Case ID: USER_SIG	**Use Case Name:** User Sign In
Primary Actor(s):	New users of the app.
Secondary Actor(s):	N/A
Description:	User enter required information and store the info into database.
Preconditions:	User push the “sign in” button and the sign in dialog displays.
Normal Flow of Event:	1. User enter required information to sign in. 2. User clicks “Ok”.
Postconditions:	After step 2, the login dialog appears.
Frequency of Use:	High
Alternative Flows:	User can clicks “Cancel” button to return to previous login dialog or “Reset” button to clear what has been entered.
Exceptions:	User doesn’t not enter all required field then a message would pop up to remind user to enter all required field. Or user doesn’t enter some field in required format, then a message would pop up to remind user to re-enter the field. Or the email has been used previously.
Assumptions:	The application is running and the user is new to the app.
Issues:	N/A
Source:	
Includes:	N/A
Associated Requirements:	TBD

Table 2. Use Case Specification - USER_SIG: User Sign In

Use Case ID: USER_LOG	**Use Case Name:** User Login
Primary Actor(s):	All authorized users of the app.
Secondary Actor(s):	N/A

Description:	Requests user name and password and authenticates user into the app.
Preconditions:	User login display dialog is available and waiting for user input.
Normal Flow of Event:	1. User enters user ID in ID field textbox and then tabs or clicks into password field and enters password. 2. User clicks "Ok".
Postconditions:	After step 2, the main dialog of the app appears.
Frequency of Use:	High
Alternative Flows:	User can click "Cancel" within the login dialog display to cancel or clear a misspelled ID or password.
Exceptions:	User does not enter data in the ID and password dialog box and is given a message that all fields need to contain information or has entered an invalid ID or password.
Assumptions:	The application is up and running, user has obtained a valid ID and password from the system administrator.
Issues:	User never has no id or password.
Source:	
Includes:	N/A
Associated Requirements:	TBD

Table 3. Use Case Specification - USER_LOG: User Login

Use Case ID: EDIT_PRO	**Use Case Name:** Edit Profile
Primary Actor(s):	All authorized users of the app
Secondary Actor(s):	N/A
Description:	User edit current personal information and save it into database.
Preconditions:	User has login to the app and click "Edit" button on personal profile page.
Normal Flow of Event:	1. User re-enters in the fields that they want to edit. 2. User clicks "Save".
Postconditions:	After step 2, a message pops up to remind user the profile has been saved successfully. The current page remains be the profile page.
Frequency of Use:	Medium

Alternative Flows:	User can click “Cancel” within the profile dialog display to cancel edit action.
Exceptions:	User leaves some required fields empty, then a message pops up to remind the user that all required field should not be empty. Or user doesn’t enter information in valid format and is given a message that the field must be in the format of “*******”.
Assumptions:	User has login to the application and click the “Edit” button on the personal profile page.
Issues:	N/A
Source:	
Includes:	N/A
Associated Requirements:	TBD

Table 4. Use Case Specification - EDIT_PRO: Edit Profile

Use Case ID: ADD_FRI	**Use Case Name:** Add Friends
Primary Actor(s):	All authorized users of the app
Secondary Actor(s):	N/A
Description:	User searches for a user or choose a non-friend user in chat room. Then send friend request to the user and waiting for response
Preconditions:	User has login to the app.
Normal Flow of Event:	1. User search for another user or find a non-friend user in a chat group 2. User clicks user’s item and select “Friend request” button.
Postconditions:	After step 2, a message pops up to remind the user the message has been sent successfully and he/she should wait for response.
Frequency of Use:	High
Alternative Flows:	N/A
Exceptions:	The search result contains no friend item, then the user was given a message that “The user you just search for didn’t exist”.
Assumptions:	User has login to the application.
Issues:	N/A
Source:	

Includes:	N/A
Associated Requirements:	TBD

Table 5. Use Case Specification - ADD_FRI: Add Friends

Use Case ID: Delete_FRI	**Use Case Name:** Delete Friends
Primary Actor(s):	All authorized users of the app
Secondary Actor(s):	N/A
Description:	User selects a friends from the friend list and delete the friends.
Preconditions:	User has login to the app and the user exist in the friends list.
Normal Flow of Event:	1. User select a user from friend list. 2. User clicks user's item and select "Delete friend" button.
Postconditions:	After step 2, a message pops up to remind the user the friend has been deleted successfully.
Frequency of Use:	Medium
Alternative Flows:	N/A
Exceptions:	N/A
Assumptions:	User has login to the application.
Issues:	N/A
Source:	
Includes:	N/A
Associated Requirements:	TBD

Table 6. Use Case Specification - Delete_FRI: Delete Friends

Use Case ID: PEER_VOC_CH	**Use Case Name:** Peer Voice Chat
Primary Actor(s):	All authorized users of the app who has become friends with each other.
Secondary Actor(s):	N/A
Description:	User send a voice chat request to one of the friends and make a voice

	call with the friends.
Preconditions:	User has login to the app and has add at list one friend.
Normal Flow of Event:	1. User select one of friends from friend list. 2. User click on "Voice call" button from popup menu and waiting for response. 3. Another user accepts the voice call request and begin voice chat. 4. User ends up the voice chat.
Postconditions:	After step 4, the voice chat page disappears and the app returns to the main page.
Frequency of Use:	High
Alternative Flows:	During request, user interrupts the request.
Exceptions:	After step 2, another user doesn't acceptt the voice chat request. Then the app return to previous page.
Assumptions:	User has login to the application and has pending message in the message box.
Issues:	The communication may be interrupt by signal problem.
Source:	
Includes:	SHA_BOARD: Share Whiteboard
Associated Requirements:	TBD

Table 7. Use Case Specification - PEER_VOC_CH: Peer Voice Chat

Use Case ID: PEER_VID_CH	**Use Case Name:** Peer Video Chat
Primary Actor(s):	All authorized users of the app who has become friends with each other.
Secondary Actor(s):	N/A
Description:	User send a video chat request to one of the friends and make a video call with the friends.
Preconditions:	User has login to the app and has add at list one friend.
Normal Flow of Event:	1. User select one of friends from friend list. 2. User click on "Video call" button from popup menu and waiting for response. 3. Another user accepts the video call request and begin video chat. 4. User ends up the video chat.

Postconditions:	After step 4, the video chat page disappear and the app returns to the main page.
Frequency of Use:	High
Alternative Flows:	During request, user interrupts the request.
Exceptions:	After step 2, another user doesn't accept the video chat request. Then the app return to previous page.
Assumptions:	User has login to the application and has at least one friend.
Issues:	The communication may be interrupt by signal problem.
Source:	
Includes:	SHA_BOARD: Share Whiteboard
Associated Requirements:	TBD

Table 8. Use Case Specification - PEER_VOC_CH:Peer Video Chat

Use Case ID: GRO_VOC_CH	**Use Case Name:** Group Voice Chat
Primary Actor(s):	All authorized users of the app who has become friends with the chat room host.
Secondary Actor(s):	N/A
Description:	User create a chat room, send voice chat requests to some of the friends and make a voice call with all selected friends.
Preconditions:	User has login to the app and has add at list two friend.
Normal Flow of Event:	1. User click the "Create Chat Room" button and create a chat room. 2. User click on "Add member" button in the chat room page and select some of friends to start a group voice chat. 3. Other users accept the requests and start group chat. 4. The room host ends up the video chat or other user exits the room.
Postconditions:	After step 4, the video chat page disappear and the app returns to the main page.
Frequency of Use:	High
Alternative Flows:	User click the "Cancel" button and return to previous page. During request, user interrupts the request.
Exceptions:	After step 2, no other users accept the voice chat request. Then the app return to previous page.

Assumptions:	User has login to the application and has at least two friends.
Issues:	The communication may be interrupt by signal problem.
Source:	
Includes:	SHA_BOARD: Share Whiteboard
Associated Requirements:	TBD

Table 9. Use Case Specification - PEER_VOC_CH:Peer Video Chat

Use Case ID: GRO_VOC_CH	**Use Case Name:** Group Video Chat
Primary Actor(s):	All authorized users of the app who has become friends with the chat room host.
Secondary Actor(s):	N/A
Description:	User create a chat room, send video chat requests to some of the friends and make a video call with all selected friends.
Preconditions:	User has login to the app and has add at list two friend.
Normal Flow of Event:	1. User click the “Create Chat Room” button and create a chat room. 2. User click on “Add member” button in the chat room page and select some of friends to start a group video chat. 3. Other users accept the requests and start group chat. 4. The room host ends up the video chat or other user exits the room.
Postconditions:	After step 4, the video chat page disappears and the app returns to the main page.
Frequency of Use:	High
Alternative Flows:	User click the “Cancel” button and return to previous page. During request, user interrupts the request.
Exceptions:	After step 2, no other users accept the video chat request. Then the app return to previous page.
Assumptions:	User has login to the application and has at least two friends.
Issues:	The communication may be interrupt by signal problem.
Source:	
Includes:	SHA_BOARD: Share Whiteboard
Associated Requirements:	TBD

Table 10. Use Case Specification - GRO_VOC_CH: Group Video Chat

Use Case ID: RE_VOC_CHAT	**Use Case Name:** Restart Group Voice Chat
Primary Actor(s):	All authorized users of the app who has group voice chat history previously.
Secondary Actor(s):	N/A
Description:	User restarts an existing voice chat room.
Preconditions:	User has login to the app and has group voice chat history.
Normal Flow of Event:	1. User select an existing but ending group voice chat in the group chat list. 2. User click on the item and click “Restart” button to resend request to all members in the chat room. 3. Other users accept the requests and start group chat. 4. The room host ends the voice chat or other user exits the room.
Postconditions:	After step 4, the voice chat page disappears and the app returns to the main page.
Frequency of Use:	Medium
Alternative Flows:	User interrupt the request.
Exceptions:	After step 2, no other users accept the voice chat request for a long time. Then the app return to previous page.
Assumptions:	User has login to the application and has group voice chat history.
Issues:	The communication may be interrupt by signal problem.
Source:	
Includes:	SHA_BOARD: Share Whiteboard
Associated Requirements:	TBD

Table 11. Use Case Specification - RE_VOC_CHAT: Restart Group Voice Chat

Use Case ID: RE_VID_CHAT	**Use Case Name:** Restart Group Video Chat
Primary Actor(s):	All authorized users of the app who has group video chat history previously.
Secondary Actor(s):	N/A

Description:	User restarts an existing video chat room.
Preconditions:	User has login to the app and has group video chat history.
Normal Flow of Event:	1. User select an existing but ending group video chat in the group chat list. 2. User click on the item and click “Restart” button to resend request to all members in the chat room. 3. Other users accept the requests and start group video chat. 4. The room host ends up the video chat or other user exits the room.
Postconditions:	After step 4, the video chat page disappears and the app returns to the main page.
Frequency of Use:	Medium
Alternative Flows:	User interrupt the request.
Exceptions:	After step 2, no other users accept the video chat request for a long time. Then the app return to previous page.
Assumptions:	User has login to the application and has group video chat history.
Issues:	The communication may be interrupt by signal problem.
Source:	
Includes:	SHA_BOARD: Share Whiteboard
Associated Requirements:	TBD

Table 12. Use Case Specification - RE_VID_CHAT: Restart Group Video Chat

Use Case ID: VIEW_PRO	**Use Case Name:** View friends’ profile
Primary Actor(s):	All authorized users of the app who has at least one friend
Secondary Actor(s):	N/A
Description:	User view the profile of a friend.
Preconditions:	User has login to the app and has at least one friend.
Normal Flow of Event:	1. User select an existing friend 2. User click on the item and click “Profile” button to view the profile.

Postconditions:	After step 2, the main page disappears and the app jump to the profile page.
Frequency of Use:	Medium
Alternative Flows:	NA
Exceptions:	NA
Assumptions:	User has login to the application and has at least one friend.
Issues:	NA
Source:	
Includes:	NA
Associated Requirements:	TBD

Table 13. Use Case Specification - VIEW_PRO: View friends' profile

Use Case ID: SE_FRIEND	**Use Case Name:** Search friend by email
Primary Actor(s):	All authorized users of the app
Secondary Actor(s):	N/A
Description:	Search a friend from database by email
Preconditions:	User has login to the app.
Normal Flow of Event:	1. User input the email of one user. 2. User clicks "Search" button. 3. Result shows on the screen.
Postconditions:	After step 3, the profile of found user shows on the screen
Frequency of Use:	High
Alternative Flows:	No result found and an alert dialog appears.
Exceptions:	NA
Assumptions:	User has login to the application.
Issues:	NA
Source:	
Includes:	NA

Associated Requirements:	TBD

Table 14. Use Case Specification - SE_FRIEND: Search friend by email

Use Case ID: SET_UP_GROUP	**Use Case Name:** Setup chatroom
Primary Actor(s):	All authorized users of the app
Secondary Actor(s):	N/A
Description:	Setup a chatroom
Preconditions:	User has login to the app.
Normal Flow of Event:	1. User click the "+" button on the room page. 2. User inputs room name, description and selects room members. 3. User click "OK" button to complete the set process.
Postconditions:	After step 3, the user enter the chatting page.
Frequency of Use:	High
Alternative Flows:	Room has existed.
Exceptions:	NA
Assumptions:	User has login to the application.
Issues:	NA
Source:	
Includes:	NA
Associated Requirements:	TBD

Table 15. Use Case Specification - SE_FRIEND: Search friend by email

TECHNICAL REQUIREMENTS

The purpose of this part is to obtain agreement regarding the platforms to be used for deploying and developing the system with professor and teaching assistants. This is input to assist technical analysis. Provide some standard to develop the project.

Operational Environment

The application should be deployed on the android system and the backend database is deployed on the Amazon AWS instance. The device must provide Wifi, tough screen, voice device and camera device.

Development Environment

Android Studio 1.5 and Amazon AWS