Software Convergence Design 2 – Service part

An Jaesung, Seo Harin, Daniel

1. Requirements analysis
2. Page initial design
   – initial design of page architecture and write specification sheet
3. Page final design
4. Page development result
   – class diagram of page ui and source code, development result
5. The minutes of meetings

1. Requirements analysis

This is requirement analysis of Application service part. It is divided with function analysis and user interface analysis. The function analysis focused on the function to be provided by the service, and the UI analysis focused on how to provide function to users through the ui layout.

Requirement name	**Live chat**	**Class-function**
Definition	Implementaion of offline live-chat service	
Details	1. Real-time live-chat service implementation in offline environment - Live chat cannot be joined in multiple rooms at the same time 2. You can open a room directly from the main page or from a text-chat you are currently participating in. - Room title and password can be set - Invite users at the time of opening or after opening 3. Provides text-chat and real-time video call functions - Text chat function can be used as it is - Video call function is available and camera and microphone on/off function are provided 4. Provides screen sharing function - Share the sharer's current screen on the application - Cannot use multiple people at the same time in chat	
Related	Live-chat	

Requirement name	**Text chat**	**Class-function**
Definition	Implementaion of offline text-chat service	
Details	1. Providing chat service through connection between devices in the region in an offline environment 2. Provides text chat and image, other files transfer functions - Preview provided in chat box when sending images 3. Users open chat rooms themselves - Room title and password can be set - The device of the user who opened the room acts as a server for the room - When opening a new chat-room, you can check and invite users who can connect in the area - Invite users within the region even after open room 4. Enter the opened room - Search and access the title of the chat room you want to join - Alternatively, you can check the list of available rooms in the current area.	
Related	Main-page text chat	

Requirement name	Sub-function	Class-function
Definition	serve sub-function	
Details	1. Available as an add-on to text chat - available in text chat 2. Provides an opening screen and a participation screen for each function 3. voting function - Set the voting subject and options, voting deadline, and voting method (anonymous, duplicate) on the voting open screen 4. mini game - Ghost leg, roulette - When performing the function, the result is shared with chat participants 5. team matching system - Provides automatic team matching function by number of chat participants - Random matching, matching detailed criteria (equal grades, minimization of overlapping majors, etc.) - Participants can directly recruit a team	
Related	Text-chat, live-chat	

Requirement name	Personal info settings	Class-function
Definition	Personal information & details setting	
Details	1. Personal profile - Enter name, contact number, E-mail address, major information and additional information - You can set whether to make your profile public or not 2. Display the user's connection status (connected, disconnected, rested, etc.) - Automatic application or direct status setting 3. User blocking function - Register device information when adding to the block list - Block information provision, reject invitation	
Related	Main-page	

Requirement name	Interface settings	Class-function
Definition	Personal interface settings	
Details	1. Personal screen settings for applications are possible 2. theme settings - Can be selected in the personal information setting window - Set the desired color of the interface, including light/dark mode 3. Interface layout settings for live chat - Select between tile/horizontal/vertical scrolling for the camera screen layout setting - Select text chat box from top/bottom, side, and minimized	
Related	Page overall	

Requirement name	Main page	Class-interface
Definition	Main page configuration	
Details	1. personal profile area - Set name and current status display area - Includes a detailed profile setting button, a blacklist button, a live chat open button, and a button to open the currently participating live chat window 2. chat list - You can see the list of chats you are currently participating in, and display the chat room name, last chat history - Search bar arrangement, searchable by title of the chat room you are participating in - Chat rooms displayed in the chat box are displayed in dark color. - Open a new chat room, including a chat room join button - Open a new chat room: Displays a list of nearby users that are detected after entering the room name and password - Join chat room: Enter room name and password 3. chat box - Join the chat by displaying the chat history of the chat room selected from the chat list - When sending a chat, the chat participant's name, profile image, chat transmission time, and chat contents are displayed in one block - The chat input box is located at the bottom of the chat box and includes a file transfer button	
Related	Main-page	

Requirement name	Live-chat	Class-interface
Definition	Live-chat page configuration	
Details	1. camera screen area - Displays the participant's camera screen, and when the camera is off, replaces it with a profile image 2. text box - Same as chat box on main page 3. Screen sharing, detailed functions - Divide the camera screen area and place the corresponding shared screen. 4. Detailed function button - Screen sharing, camera, microphone, additional function on/off button included	
Related	Live-chat page	

Requirement name	Sub-function page	Class-interface
Definition	interface for sub function	
Details	1. Buttons for using detailed functions are located at the top of text chat 2. Voting open window: Title, voting items, date deadline, voting type setting (anonymous, duplicate) 3. Voting window: You can check the currently opened voting list and participate in voting after selecting - List the currently opened voting list on the left list - Display the selected votes on the right area 4. Mini-game window: When opening, after setting the title and item, it works as a game start button - A separate window to display the game result is configured, and the chatting participants can check it immediately 5. Team composition: Select a configuration method when opening (random, standard) - Random composition: After setting the number of people per team, users in the chat room are randomly matched - Standard setting: Standard setting using profile information (age, gender, major, etc.) or A method in which a specific user sets the recruitment criteria and other users participate	
Related	sub function in Text-chat area	

Requirement name	Profile Setting window	Class-interface
Definition	interface for profile setting	
Details	1. Access with a button in the personal profile area of the main page 2. Personal information can be checked and edited - Name, grade, department, phone number, e-mail address, etc. can be set - Profile image can be set - Select whether to disclose information 3. Theme - Dark mode on/off function is provided - When a desired color is selected, the interface is changed to the corresponding color theme 4. Block users list - manage block user to block invite from specific user	
Related	Profile setting window	

Requirement name	Detected user list window	Class-interface
Definition	Datected user list for invite, check	
Details	1. Display a list of connectable users for user invitation 2. When opening a chat room or pressing the Invite button in an already opened room, a new window opens 3. Detected users are displayed by name and you can select the users you want to invite	
Related	Text-chat, live-chat invite function	

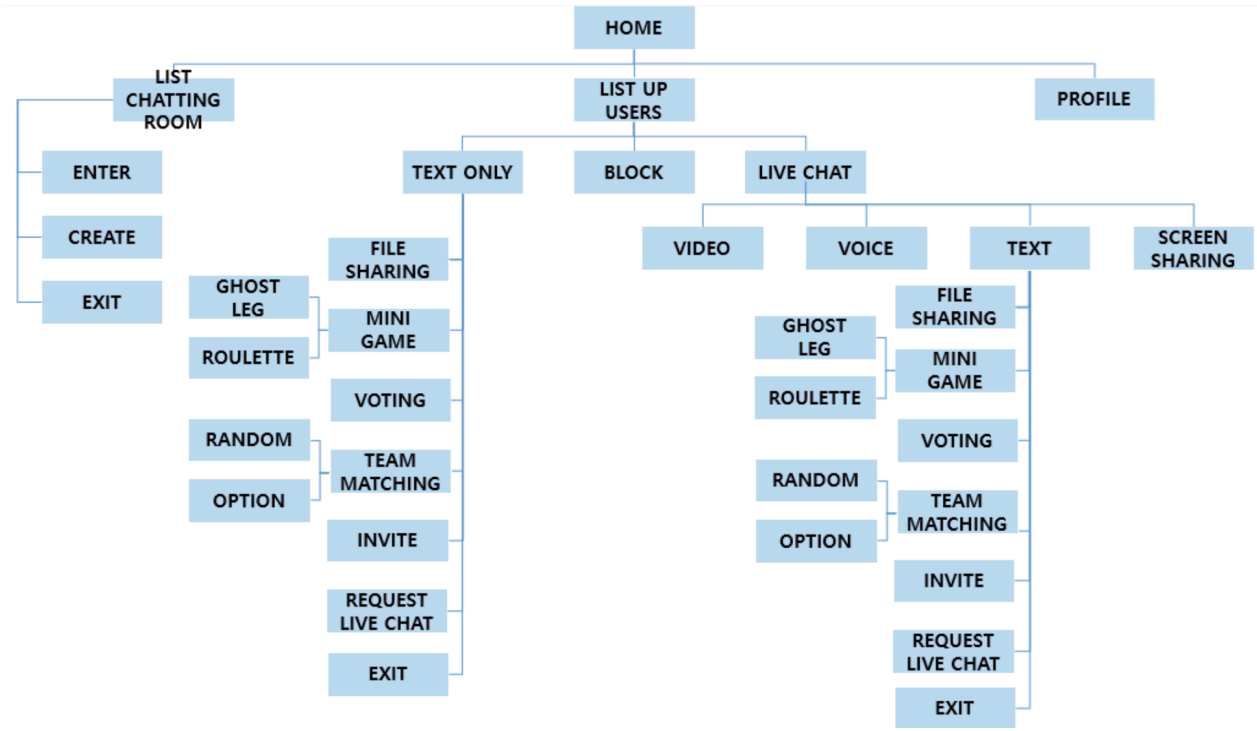
2. Page initial design

According to the requirements analysis, the structure design for the functions and screen composition to be included in the user interface (front-page) was carried out, and write a specification sheet.

Offline chatting - service(front) part spec sheet

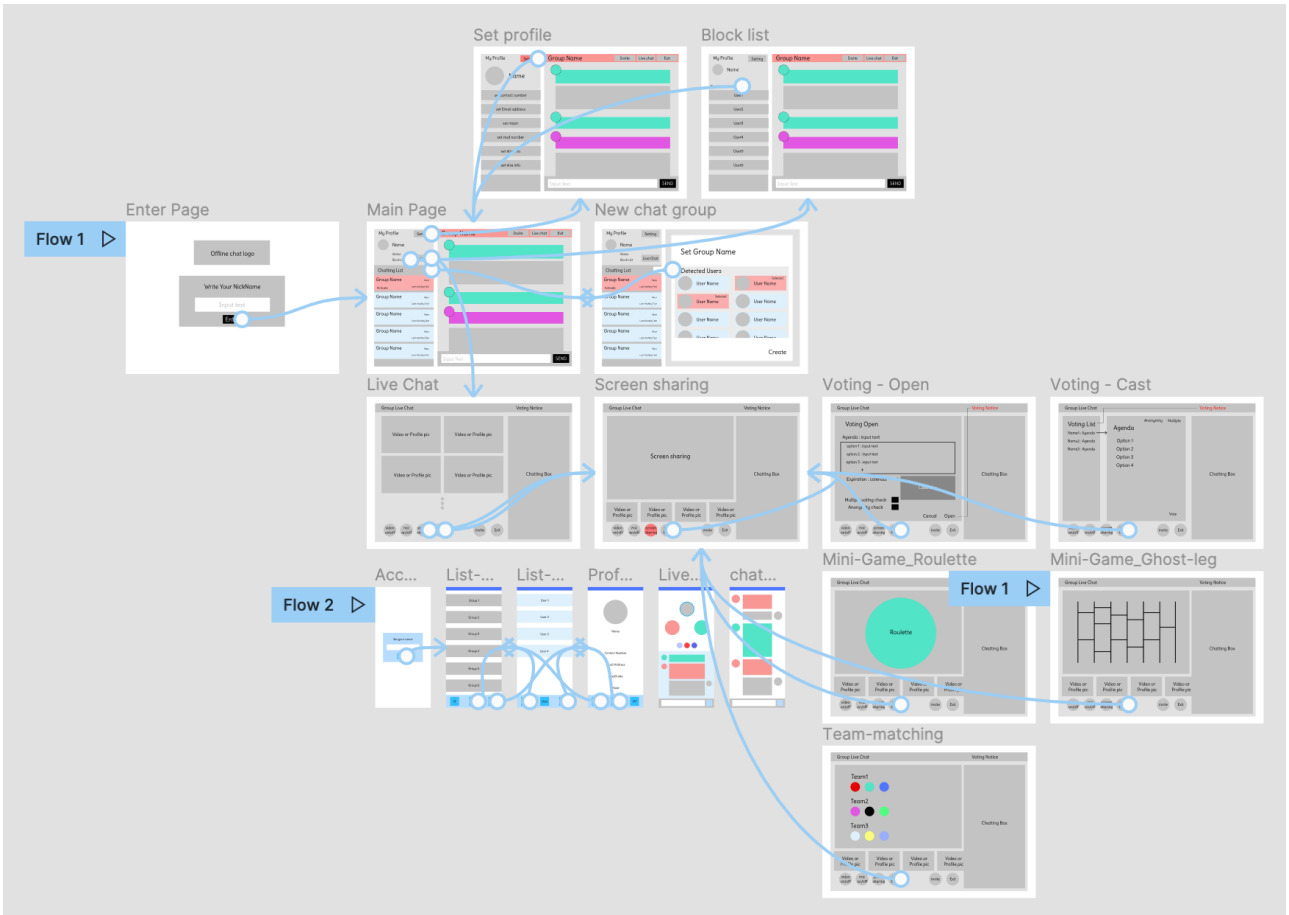
Page	Main-work	Sub-work	Detail(item)	Reference	Memo
Access page	Setting name		Name	English, Korean, number is possible / 4~12 char	Page for accessing the application. Setting the name that is exposed to others. Because of no account system, need no setting nickname whenever access the app.
Profile page	View info		Name(required)	English, Korean, number is possible / 4~12 char	Page for profile info. Name value is required but the others are not. Users can set whether to disclosure or not. (more items can be added)
	Change info		Contact number		
			Major		
			Age/Grade	Number	
			E-mail address		
			Privacy setting	Setting whether disclosure info, true/flase	
List-user page	List-up users	check the other's info(profile)	User list & profile	list of detected users	App find someone who can connect with in same area. And list up to show to users. If users block specific user, set 'block' to cut-off invitation or DM.
		Request text-chat(open text-chat)	chat room address	to decide link previous chat or open new chat room	
		Request live-chat(open live-chat)			
		Block	Block info	Whether to block specific user	
List-chatroom page	List-up chatroom	Enter	Chat group list	list of group I joined	List-up chat room I joined. You can create text-chat room or live-chat room (alone in new room) and invite others on user-list.
		Create			
		Exit			
Text-chat page	chat (text)	Send text	chat record	to show info	Text chat 1:1 or with many users. Just text chat, and send files. You can open live chat that others can enter. "Exit" close the chat page and record is deleted
		Send file(img preview)	File transfer	Text format	
		Invite	Using user-list		
		Request live-chat		can open live-chat with participants	
		Exit			
Live-chat page	Live chat room (meeting room)	Send text	chat record	text format	Create chat room and invite users on user-list-up. Users can use sub-function on chat room(sharing, vote, mini-game, team-matching). when users enter the room, mic-off, video-off. More than one presenter, Screen sharing, cannot exist at the same time.
		Send file(img preview)	File transfer		
		Voice chat	Mic		
		Video chat	Camera		
		Screen sharing	Screen window	Device screen streaming	
		Sub function	Link to sub-func page	voting, team-matching, mini-game	
		Invite	Using user-list		
		Exit			
Vote page	Vote		Agenda, option		Sub-function of chatroom.
			Anonymous/write-in	true/false	
			deadline	Date format	
			Multiple vote	true/false	
Team-matching page	Matching	Random matching	number of each team	Number	Sub-function of chatroom.
			List of users joined in room		
		Based on option	Option		
Mini-game	Ghost-leg		option set		Sub-function of chatroom.
	Roulette		option set		

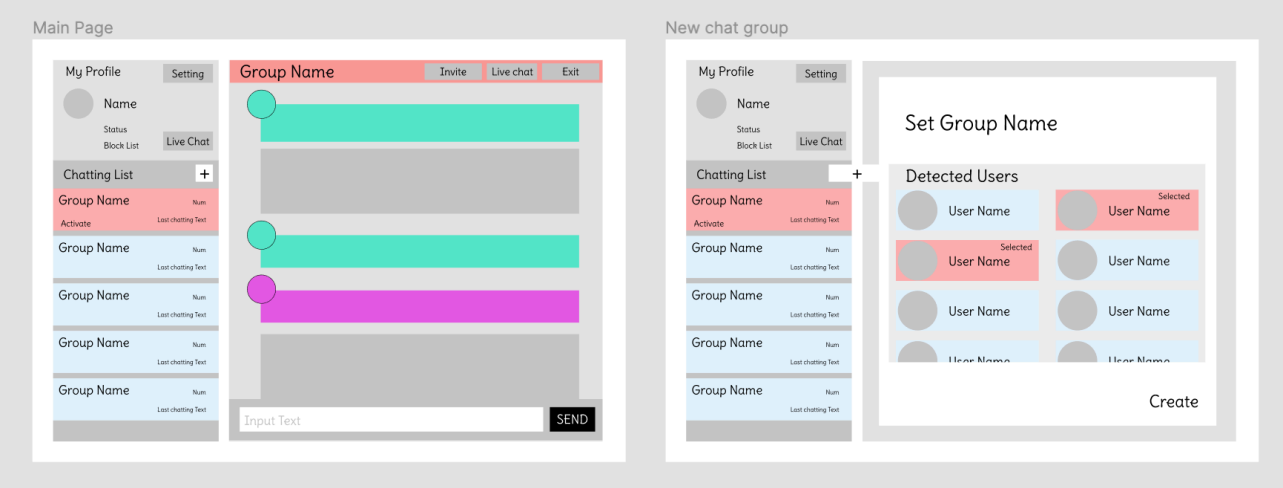
Page architecture



3. Page final design

This is the final page design based on requirements analysis and specification sheet, using Figma. There are separate pages applied to mobile devices and desktop devices, and page for desktop were planned to be developed first. Main-page composed of text-chat and chat-room-list, and live-chat-page where real-time-call can be used were designed. The workflow indicated by each arrow.

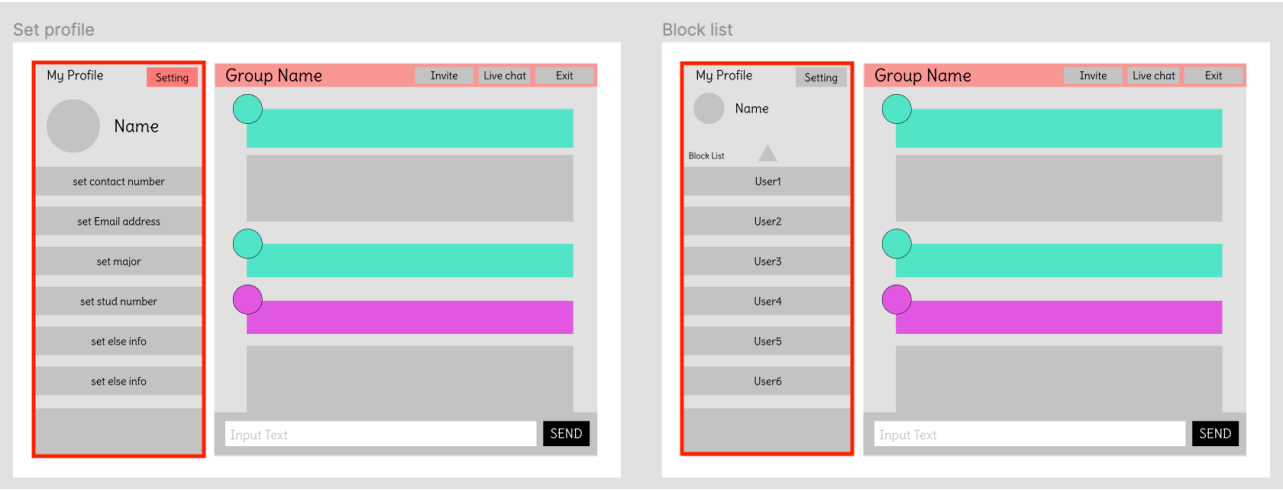




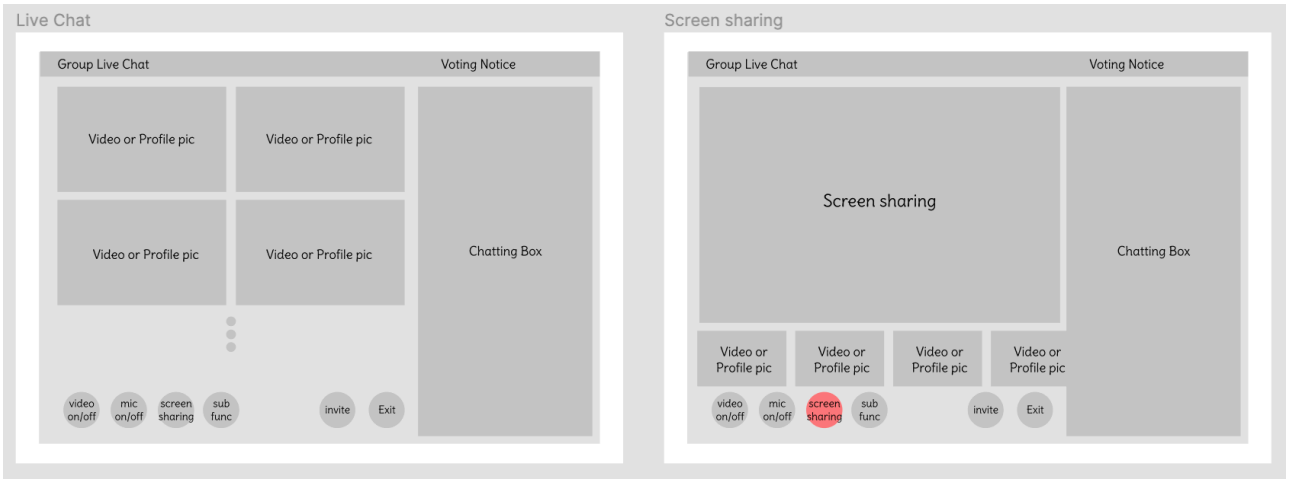
Main-page is divided into left/right parts. The profile area is at the top of left part, the list of chat-rooms you are currently participating in are located at the bottom. The Chat-box for chatting history of currently selected chat room is displayed in the right part.

The profile area of the left part displays the user name, photo, and current status, and includes a profile setting button, a live chat button, and a block list. The chat-room-list shows the chat rooms you are currently participating in and if it is selected, it is changed in a darken color. Include a button to open a new chat room in bottom of that part. If you press the button, a window for create a room is displayed including a list of users who can connect. After selecting the users you want to invite to the chat room, open the room. This window is also used for the invitation function in an already opened chat room.

In the right area, the chat-box of the selected room is displayed. At the top of the chat-area, it includes the room name, an invite button, a live chat button, and a button to leave the room. In the conversation log, each log includes user's photo and name, conversation details, and input time are displayed. A text-input-box is placed at the bottom of the dialog window.

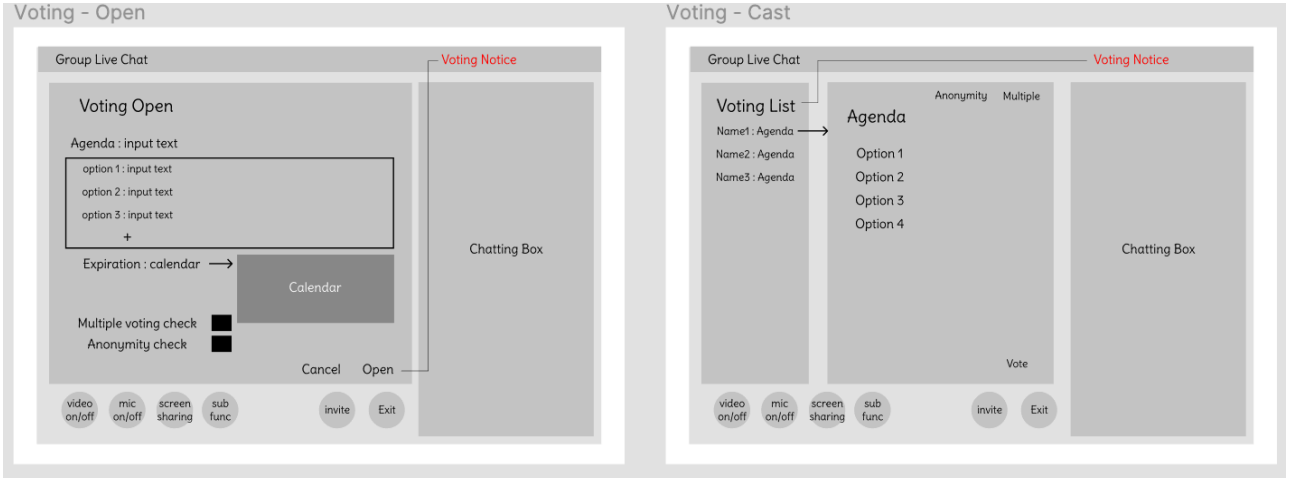


It is the operation ui of the profile setting button and the block list button in the profile area. When the button is pressed, a new window appears and button color is changed to indicate current window is being displayed. profile contents and block list and be edited in each window. if you press the button again, the changes are saved and the window disappears.

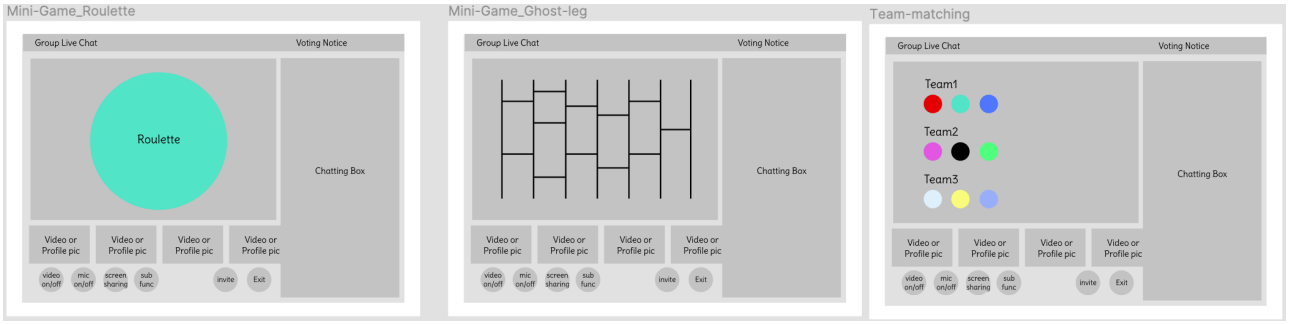


This is the page that appears when you click the live chat button in the profile area. It consists of an area repriseenting the camera screens of users participating in chatting. control button at the bottom, and a text-chat-box on the right side. Main layout is planned like above images and user can directly change the page layout. The camera area is basically displayed in a checkerboard format, and the number of arrangements can be changed according to the number of participants.

If the screen sharing function is used, the area is replaced with the shared screen and the camera is moved to the bottom. the button area includes the mic, camera, screen sharing, sub-function, invite, exit button. User can use voting, mini-game, and team matching system on sub-function. the text chat area on the right side uses same one of main page.



It is a voting screen and has a open screen and a participation screen. The screen is replaced as images. On the open screen, you can set the title, items, deadline, anonymous, and duplicate. When a vote is opened, it inform at the top of the page. In the voting notification, the voting list in progress is displayed on the left, and you can check the agenda and options of the selected vote on the right.

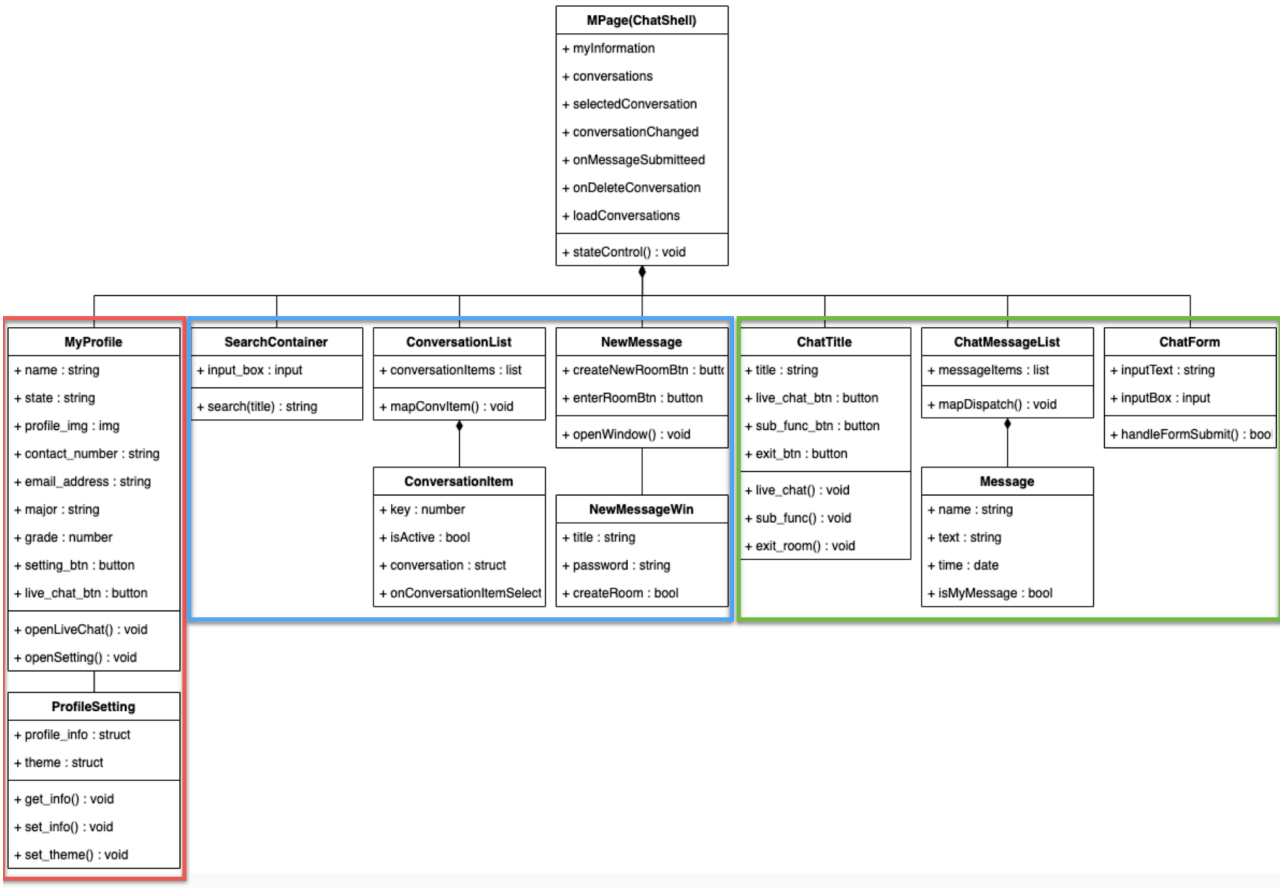


This is the screen for roulette, ladder climbing, and team matching. Each function has a separate screen for open, similar with voting opening screen. When the user uses the sub-function, the screen for open is displayed and the function is performed according to the settings. When there is a result, a notification is displayed at the top, and the participants can immediately check it.

4. Page development result

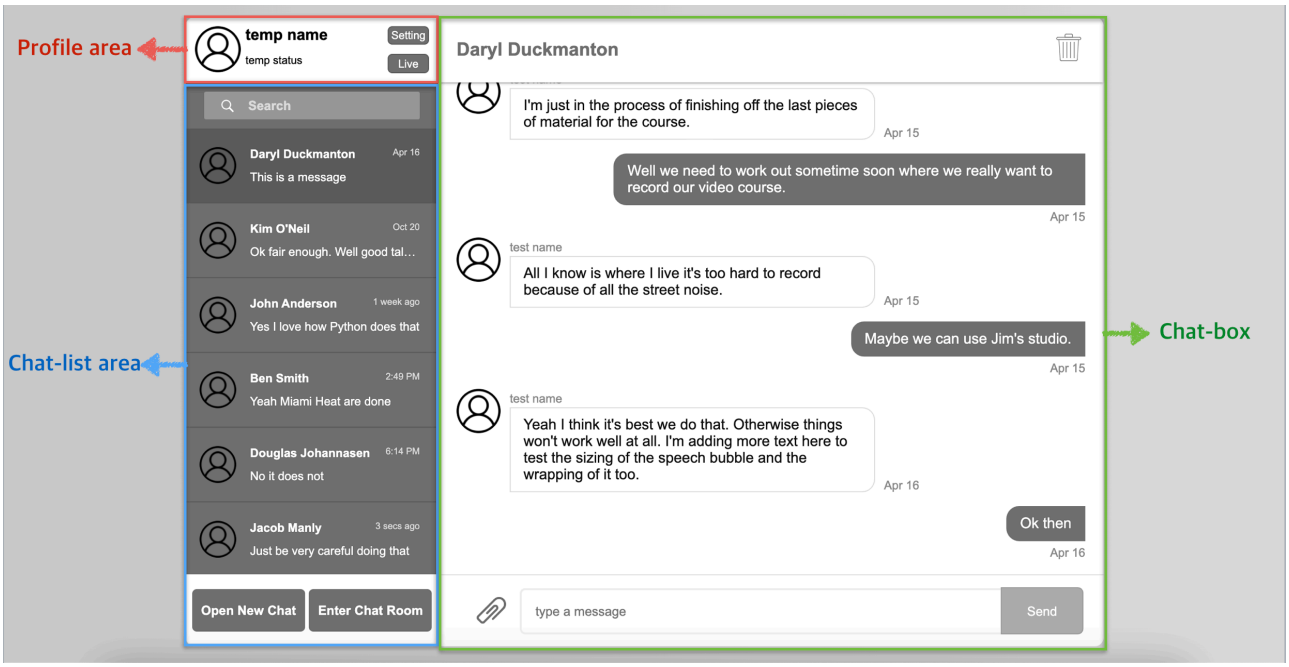
- Class Diagram (Main page)

ChatShell is a panel that contains the entire page, and it is implemented by merging each component on top. The area where each class is grouped ware implemented in the area marked with the same color on the actual page.

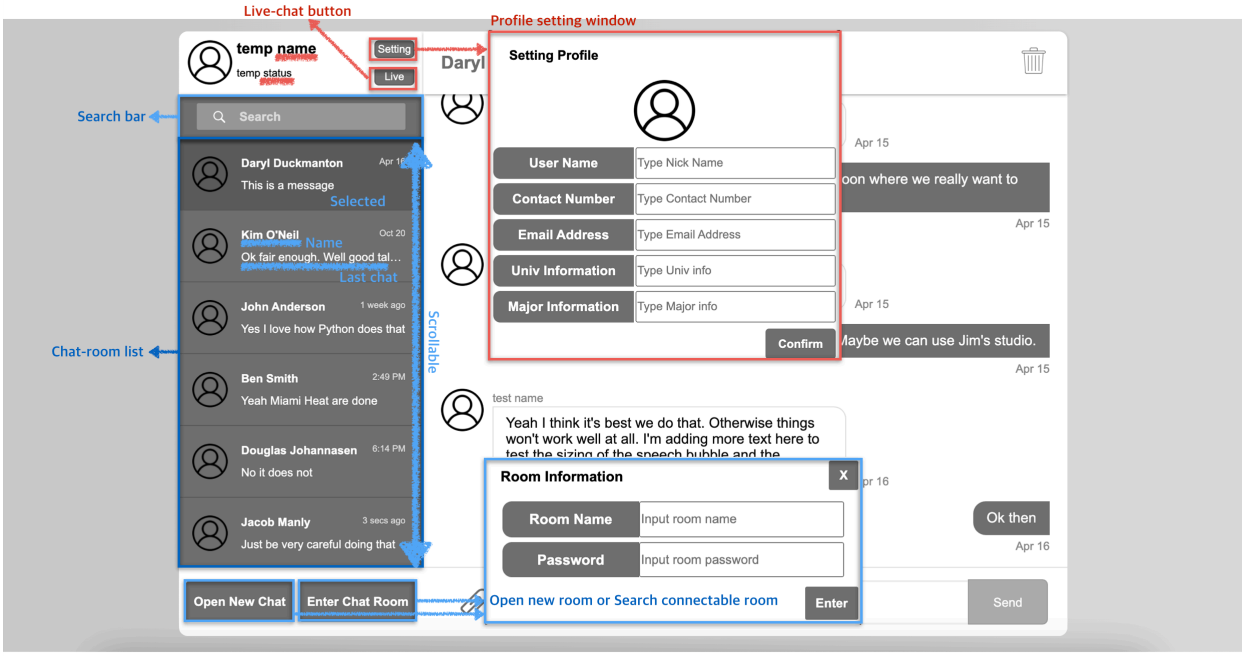


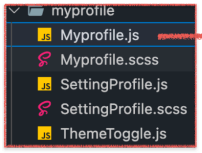
- Page development result (Main page)

Each area of the class diagram is marked with the same color.



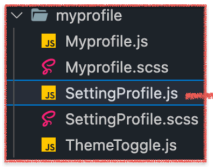
- Profile area and chat-room-list





Profile area

```
const Myprofile = ({ myInformation, state}) => {
    const [windowState, showWindow] = useState(false);
    const toggleWindow = () => {
        showWindow(true);
    };
    return (
        <div className="my-profile">
            <img src={profileImg} alt="profileImg" />
            <div className="my-name">temp name</div>
            <button id="btn-setting" onClick={toggleWindow}>Setting</button>
            {windowState ? <Modal showWindow={showWindow} /> : null}
            <div className="my-state">temp status</div>
            <button className="btn-livechat">Live</button>
        </div>
    );
}
```

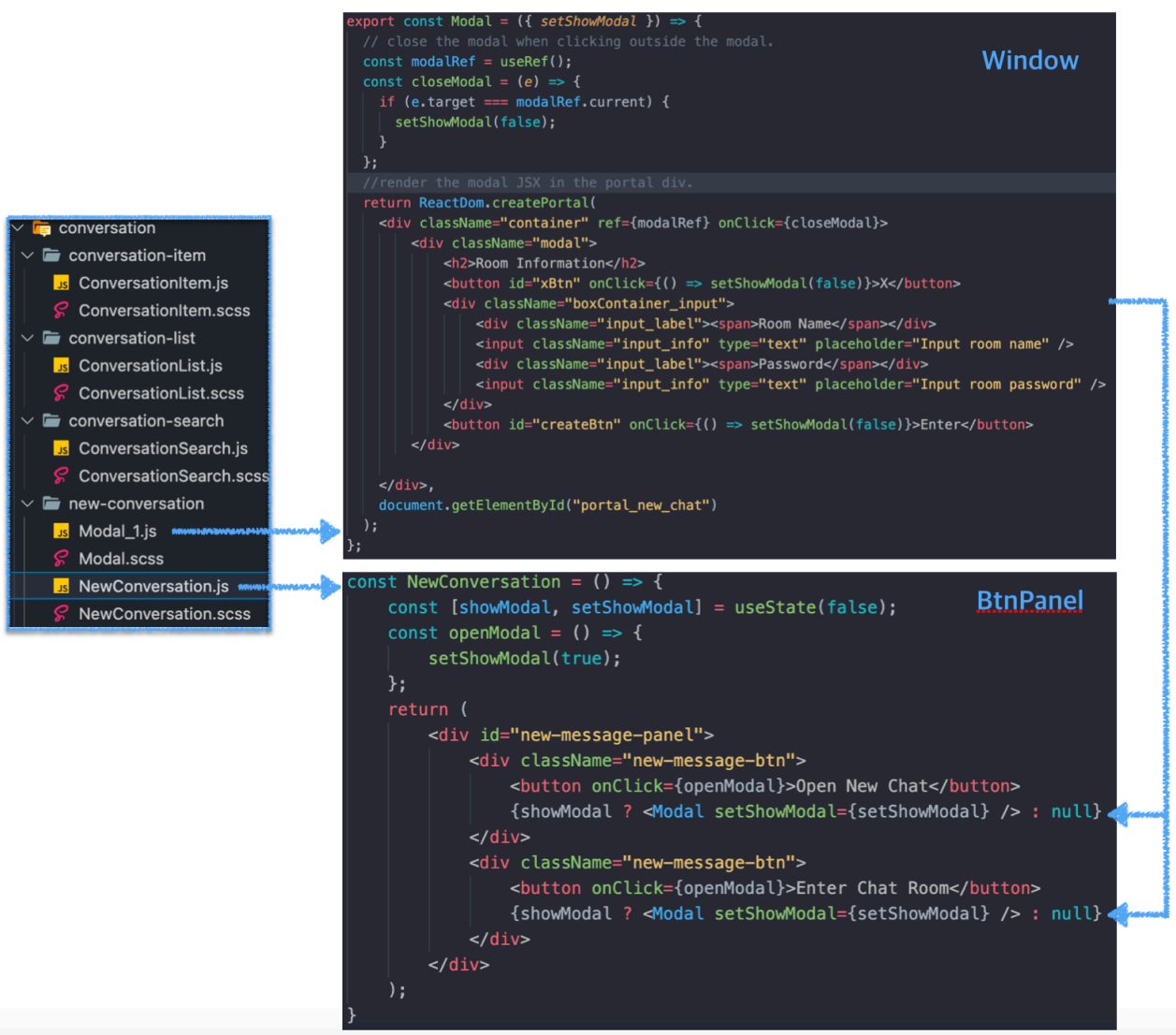


Setting window

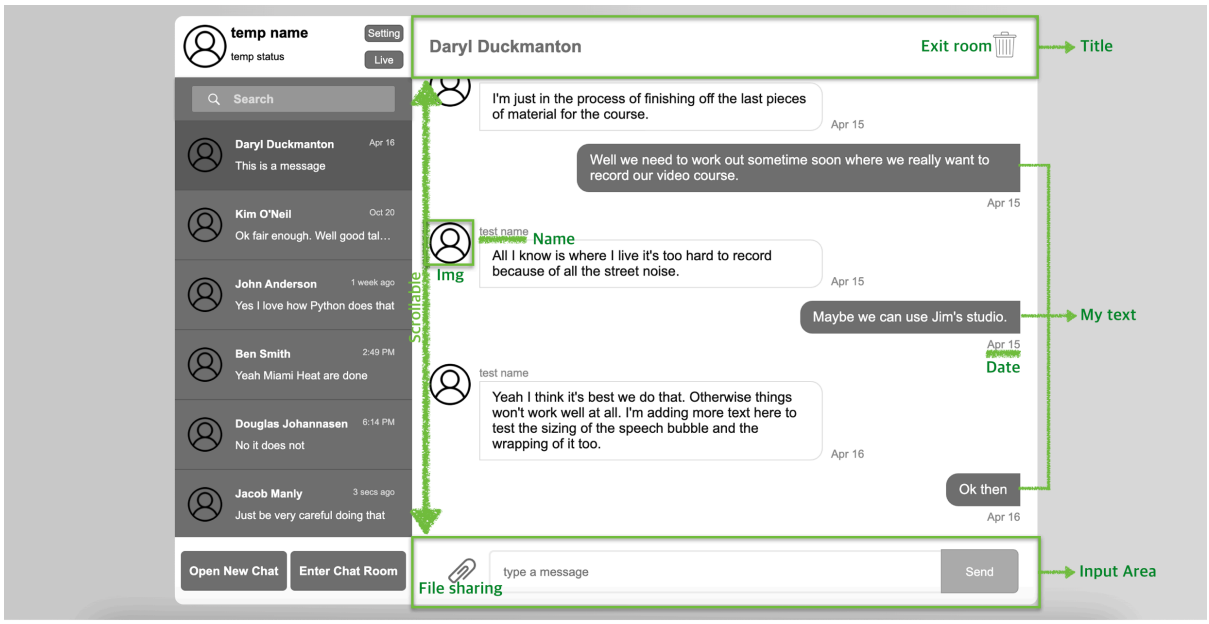
```
export const Modal = ({ showWindow }) => {
  // close the modal when clicking outside the modal.
  const modalRef = useRef();
  const closeModal = (e) => {
    if (e.target === modalRef.current) {
        showWindow(false);
    }
  };
  const [ThemeMode, toggleTheme] = useTheme();

  //render the modal JSX in the portal div.
  return ReactDom.createPortal(
    <div className="container_setting" ref={modalRef} onClick={closeModal}>
        <div className="modal_setting">
            <h2>Setting Profile</h2>
            <div className="mod_btn_panel">
                <ThemeToggle toggle={toggleTheme} mode={ThemeMode}>
                    DarkMode
                </ThemeToggle>
            </div>
            <img src={profileImg} alt="profileImg" />
            <div className="data_container">
                <div className="tag_container">
                    <span>User Name</span>
                </div>
                <input
                    type="text"
                    placeholder="Type Nick Name" />
```

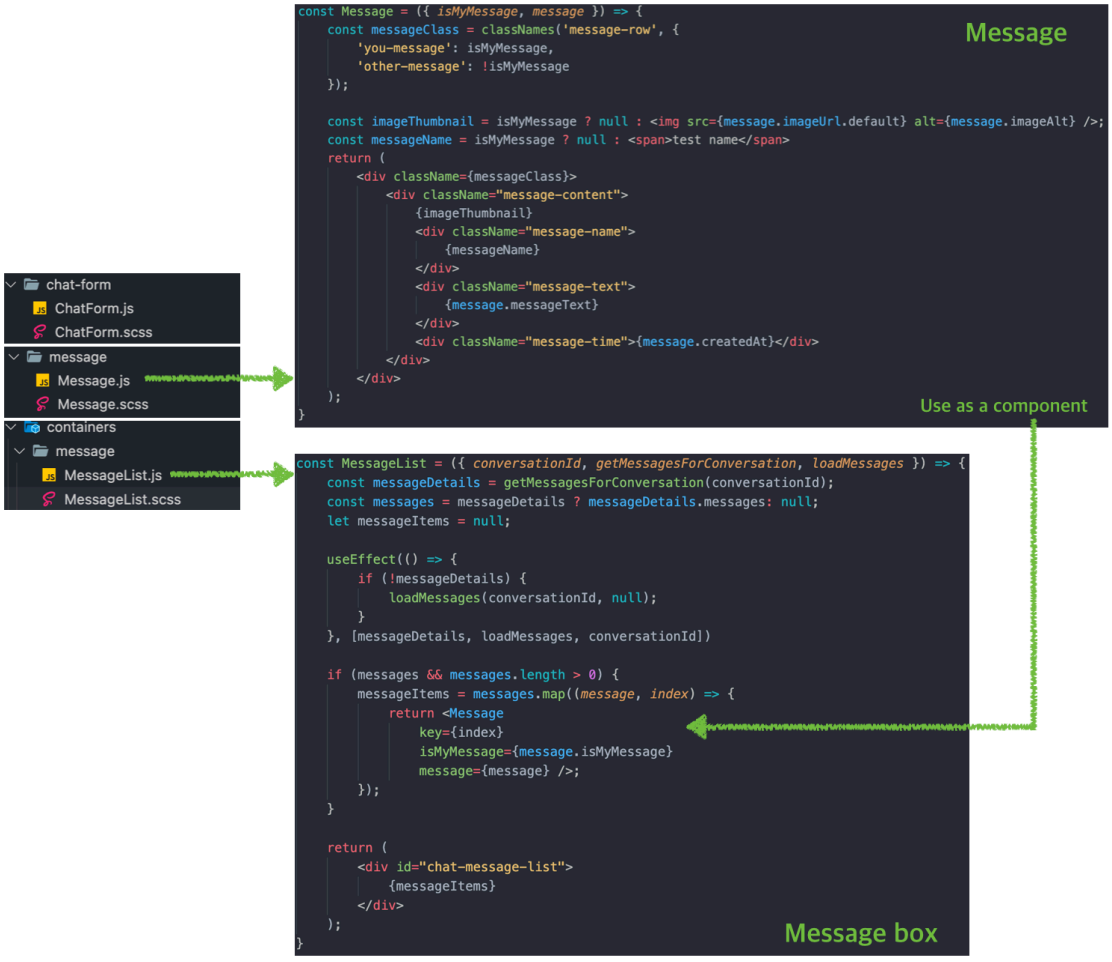
```
    </div>,
    document.getElementById("portal_set_profile")
  );
};
```


conversation
conversation-item
ConversationItem.js
ConversationItem.scss
conversation-list
ConversationList.js
ConversationList.scss
conversation-search
ConversationSearch.js
ConversationSearch.scss
new-conversation
Modal_1.js
Modal.scss
NewConversation.js
NewConversation.scss
ConversationItem
Use as a component
ConversationList


conversation
conversation-item
ConversationItem.js
ConversationItem.scss
conversation-list
ConversationList.js
ConversationList.scss
conversation-search
ConversationSearch.js
ConversationSearch.scss
new-conversation
Modal_1.js
Modal.scss
NewConversation.js
NewConversation.scss
Window
BtnPanel

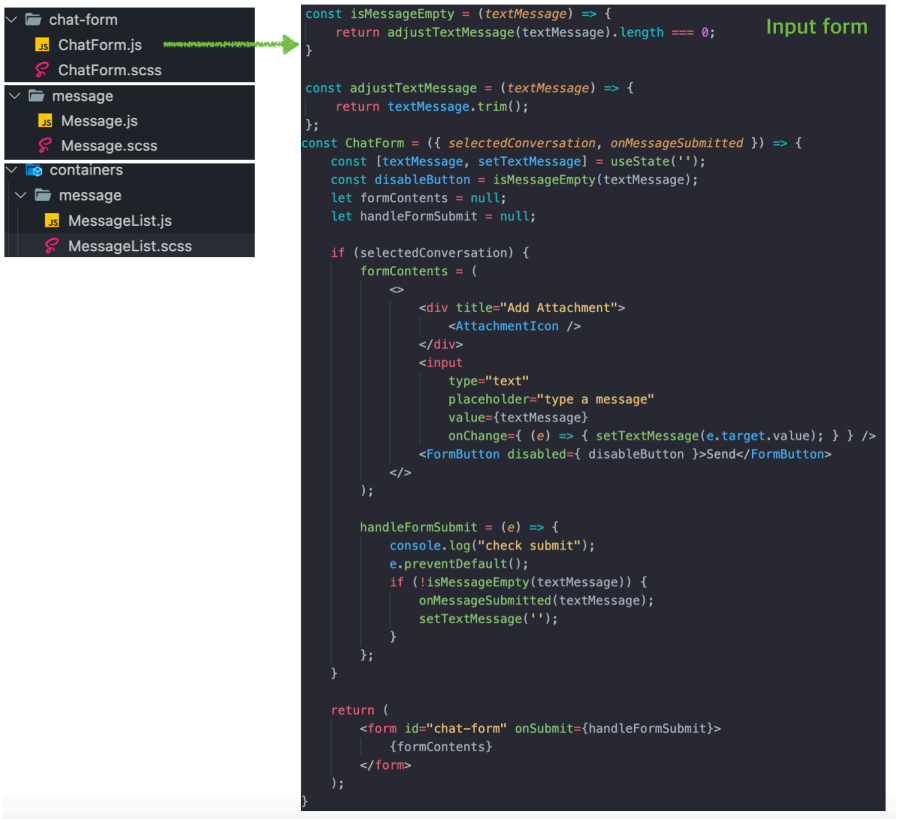
- Chat-box



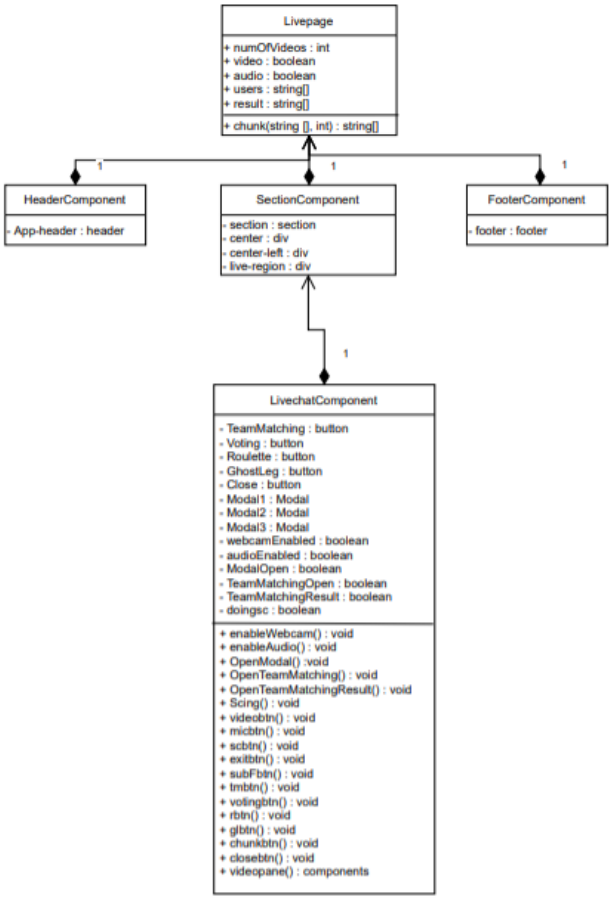



```
const Message = ({ isMyMessage, message }) => {
    const messageClass = classNames('message-row', {
        'you-message': isMyMessage,
        'other-message': !isMyMessage
    });

    const imageThumbnail = isMyMessage ? null : <img src={message.imageUrl.default} alt={message.imageAlt} />;
    const messageName = isMyMessage ? null : <span>test name</span>
    return (
        <div className={messageClass}>
            <div className="message-content">
                {imageThumbnail}
                <div className="message-name">
                    {messageName}
                </div>
                <div className="message-text">
                    {message.messageText}
                </div>
                <div className="message-time">{message.createdAt}</div>
            </div>
        </div>
    );
}
```

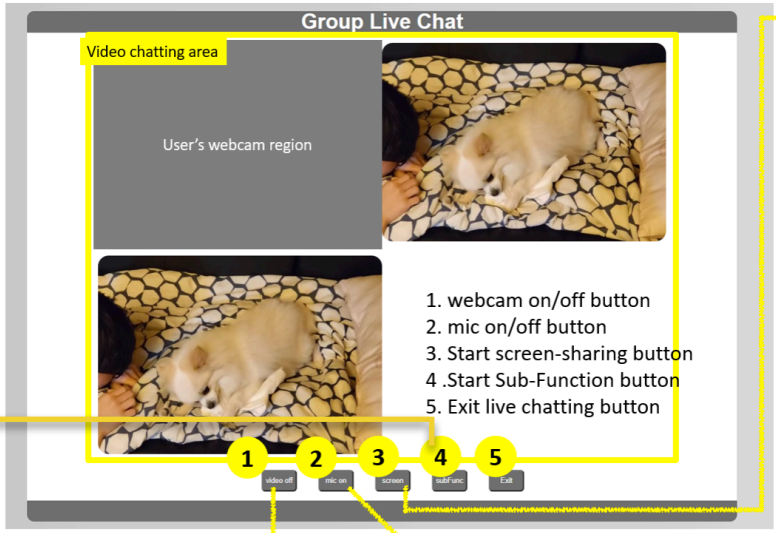
```
const MessageList = ({ conversationId, getMessagesForConversation, loadMessages }) => {
    const messageDetails = getMessagesForConversation(conversationId);
    const messages = messageDetails ? messageDetails.messages: null;
    let messageItems = null;

    useEffect(() => {
        if (!messageDetails) {
            loadMessages(conversationId, null);
        }
    }, [messageDetails, loadMessages, conversationId])

    if (messages && messages.length > 0) {
        messageItems = messages.map((message, index) => {
            return <Message
                key={index}
                isMyMessage={message.isMyMessage}
                message={message} />;
        });
    }

    return (
        <div id="chat-message-list">
            {messageItems}
        </div>
    );
}
```



-Class diagram(Live chat)



- Page development result (Live chat page)




```
scbtn =()=>{
  if(this.state.doingsc){
      alert("stop screensharing");
      this.setState({doingsc:false});

  }
  else{
    alert('start screensharing');
    this.setState({doingsc:true});
  }
}

exitbtn=()=>{
  window.location.replace("/");
}
//for scroll

subFbtn=()=>{
  //for team matching, voting, roulette, ghost leg
  this.setState({ModalOpen :true});
}

tmbtn=()=>{
  this.setState({ModalOpen :false});
  this.setState({TeamMatchingOpen : true});
}
```

In screen document num3 : screen sharing button

Onclick method : scbtn

If user click this button, screensharing state will be change (on/off)

In screen document num4 : sub function button

Onclick method : subFbtn

If user click this button, modal's(to show sub-functions) state will be change (on/off)

In screen document num5 : exitbutton

Onclick method : exitbtn

```
videobtn = ()=>{

  if(this.state.webcamEnabled){
    this.setState({webcamEnabled : false});
    video = "on";
  }
  else{
    this.setState({webcamEnabled : true});
    video = "off";
  }
}
```

In screen document num1 : videobutton

Onclick method : videobtn

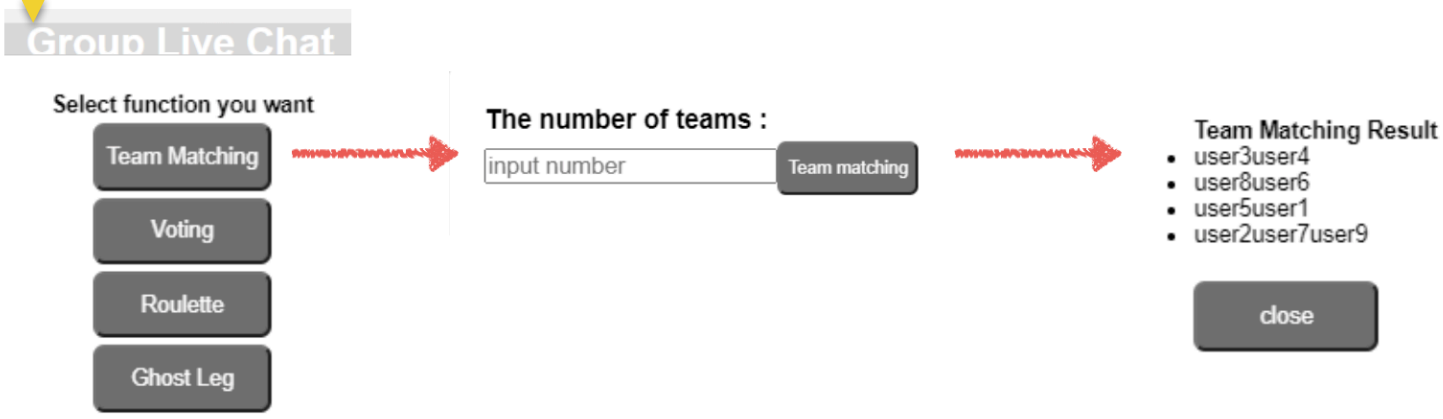
If user click this button, webcam's state will be change (on/off)

```
micbtn =()=>{
  if(this.state.audioEnabled){
    this.setState({audioEnabled:false});
    alert('muted');
    audio = "on";
  }
  else{
    this.setState({audioEnabled:true});
    alert('unmuted')
    audio="off";
  }
}
```

In screen document num1 : micbutton

Onclick method : micbtn

If user click this button, mic's state will be change (on/off)




```
<Modal class ="Modal3" isOpen = {this.state.TeamMatchingResult}>
<h4>Team Matching Result</h4>
<ul>
{result&&result.map( (v)=>{
  return(
    <li>{v }</li>
  )
})}
</ul>

<br></br>
  <button class="button2" onClick={this.closebtn}>close</button>
</Modal>
```

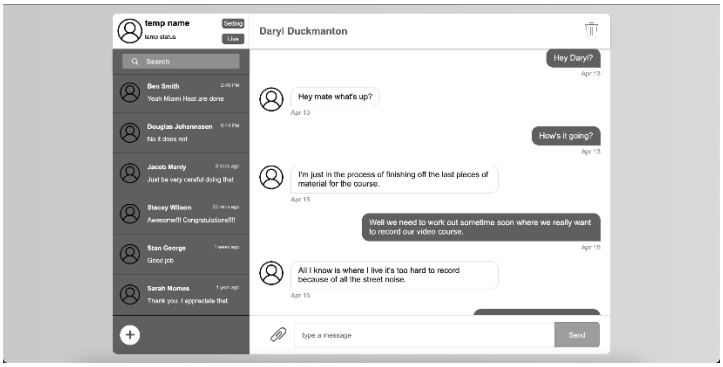
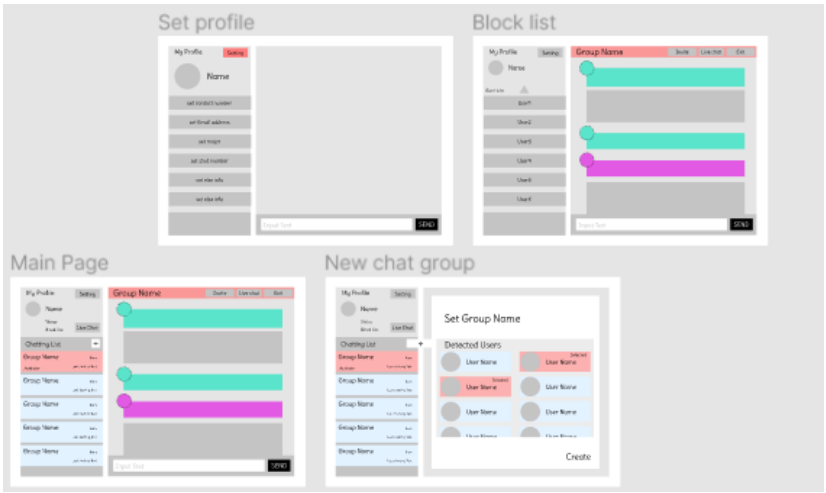
If user click team matching button, method chunk called to divide team

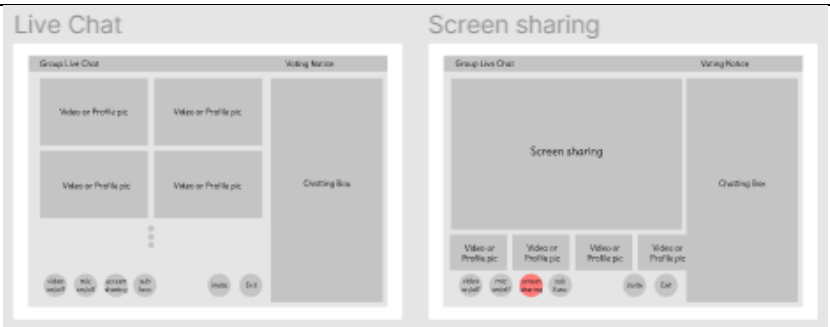
And TeamMatchingResult is switch to "true" to open teammatching result Modal.

```
chunkbtn=()=>{
 result = chunk(users,this.state.numOfteams);
 this.setState({TeamMatchingOpen : false});
 this.setState({TeamMatchingResult:true});
 console.log(this.state.numOfteams);
 console.log(result);

}
closebtn=()=>{
  this.setState({TeamMatchingResult:false});
}
```

```
function chunk(arr, size) {
  arr.sort(() => Math.random() - 0.5);
  var i, j, temparray = [], chunk =arr.length/size;

    for (i = 0, j = arr.length; i < j; i += chunk) {
      temparray.push(arr.slice(i, i + chunk));
    }
  return temparray
}
```

MINUTES : version1

Date	2021/11/08	Meeting place	IT4 - 101
Participant	Jaesung Ahn, Harin Seo		
Contents	1. The application's theme is decided.  2. Main page, chatting page should be completed implement by this week (~11/14) 		



Except sub-functions, all pages' frame should be completed implement by 11/14

3. In implementation, data will be static, later connect with backend's
4. More study about camera, screen sharing in webapp (React module)

   Reference

   - https://www.twilio.com/blog/video-chat-react-hooks
   - Video Chatting and Screen Sharing with React, Node, WebRTC(peerjs) - DEV Community
5. Technology documentation writing method:

   Focusing on used modules and code in each page, will upload on GitHub

MINUTES : version2

Date	2021/11/15	Meeting place	IT4 - 101
Participant	Jaesung Ahn, Harin Seo, Daniel		
Contents	1. Should more study about - How to implement screen sharing - How to encode/decode components to send to backend : Maybe need database Speed of video streaming send/receive Data send/receive unit - How to recode and send webcam video to other users - https://www.npmjs.com/package/react-webcam : for usage of react-webcam module - https://sendbird.com/developer/tutorials/screen-sharing-javascript : for screen sharing on js 2. Complete last meeting's contents 3. Apply theme on implemented code		

MINUTES : version3

Date	2021/11/22	Meeting place	IT4 - 101
Participant	Jaesung Ahn, Harin Seo, Daniel		
Contents	1. Video box - More closer each component(padding :5px) - 2,4,9 have each view, more than 9 -> scroll 2. Implementation of profile setting component 3. Write screen document specifically : flow between screens should be expressed. 4. Integration main page and live page 5. Discuss about data send/receive : serverless application – we need something acts as server and privacy is important too		

MINUTES : version4

Date	2021/12/01	Meeting place	IT4 - 101
Participant	Jaesung Ahn, Harin Seo, Daniel		
Contents	1. Sub-function implementation : Team matching Voting Roulette Ghost-leg 2. Unit testing of each component : buttons' onClick methods Team matching – chuck function Video chatting area – dynamic structure Chatting room lists 3. Integration with backend		

MINUTES : version5

Date	2021/12/08	Meeting place	IT4 - 101
Participant	Jaesung Ahn, Harin Seo, Daniel		
Contents	1. Make presentation ppt 2. Write documents - Requirements documents (functional/interface) - Screen documents (focus on flow between components) - Technology documentation (based on code) - User Tutorial - Class diagram		