



Group 8 - UI Report

Submitted By

64130500231	Supasek	Dhanabordeepeat
64130500202	Kyaw Swar	Hein
64130500206	Kantitat	Warawut
64130500265	Phongsaphak	Fongsamut
64130500255	Khush	Agarwal
63130500228	Sukanya	Chinwicha

Present

Dr. Vithida Chongsuphajaisiddhi
Course CSC234 User Centered Mobile Application Development
Semester 2, Academic Year 2022

Table of contents

0. Design rationale for WisMod's environment	3
1. Feed Page	4
2. Event Detail Page (Normal View,Owner View)	6
3. Create Event Page	10
4. Chat Room	12
5. Chat Page	13
6. Profile Page	15
7. Main Setting Page	17
8. Sub-Setting Pages	19
9. Block List Page	22
10. Admin Dashboard	24

0. Design rationale for WisMod's environment

Primary Color

We pick Purple as our primary color, it has an inner meaning which represents wisdom and luxury. We want to enhance the student experience of living together in the university, by increasing more activity done with inter-faculty students.

Secondary Color

By using the companion color theory, we choose pink as secondary color to be used within our application, this promotes the variability within application where user can differentiate each element easily.

Background Color

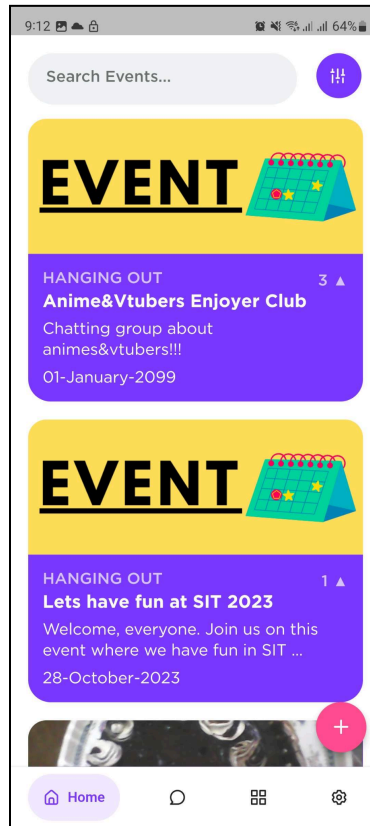
We pick white as our background color, white is a very preferable choice for making contrast with another color, and in order to maintain a good proportion for variability of color within our app, we have a background in white color.

Navigation Bar

This element acts as an index for the user to navigate them to their most preferred page, we select these 4 main pages to be in our navigation button; Feed page, Chat page, Profile page & Setting page. The leftmost shows the most used page which is the feed page, and consequently the rightmost is the least used page where we put the settings page there. The order of the pages in the navbar highlight the concept of *Human Visual*, which normally people read from left to right, top to bottom, the leftmost element represents the most highlighted event which the developer team wants to present to the user first.

1. Feed Page

1.1 Screenshot of the Feed Page



1.2 Functional Requirement

- User can surf the feed page and find all events available at that period.
- User can filter the event regarding their interest using the filter button. This will direct user to filter page.
- User can find specific or related events which the user is interested in via the search bar at the top.
- User can add the post to the page via the plus button, it will direct the user to create an event page.
- If any event looks interesting, the user can click on the event card and see more description about that event. This action will direct the user to the event detail page.
- This page is viewed as the main page, so our application has navbar which include the feed page as a frequently used menu. This allows user to direct to main page whenever they want.

1.3 Elements of the page

- Search bar
- Filter button
- Event card which contains

- Event picture
- Event type
- Event title
- Number of user upvote for that event
- Brief detail of that event
- The day event will be held
- Add event button
- Navigation bar (Navbar)

1.4 Design Rationale

We want our feed page to be the center of our application, where user can see all events recommended at the first page. This means that the first page should be most impactful to the user, so we choose this first page to present all interesting event which is being held at the moment.

Search Bar & Filter Button

Both the search bar and filter button are placed at the top of the screen. We use the idea of *Unity*, this is useful for user when they want to specify the event regarding their interests, they can choose it either via search bar or filter function. This also helps users to categorize the elements of the page as well. We also use the contrast of color to differentiate these 2 elements separately (*Variability*), the search bar is painted in white while the filter button is painted in purple as our primary color.

Event Card

Event card is what our application wants to present to user the most. We use the concept of *Hierarchy* to promote the visual of this particular element by putting picture of that event at the top of card, normally picture has the ability to attract the viewer naturally. So user will be first look at the top event first. The body of the card is painted in purple to promote the difference from the background color. Card is designed to be larger than other component in this page, by combining the use of primary color and size of the card, we are now able to promote the concept of *Dominance* element within the app. This card will play as a dominant element on this page.

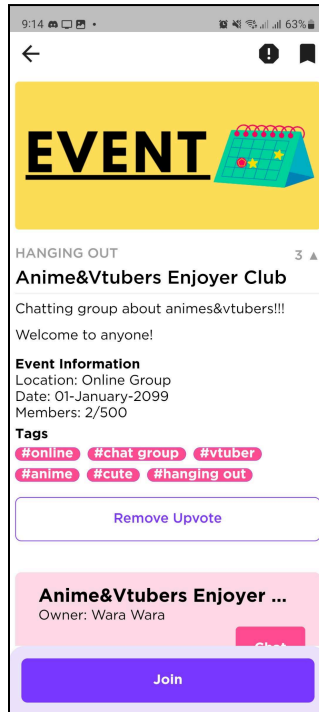
The top text is event type, placing the type at the top helps user to know first that what this event is related. While the title of the event card is in bold format, this also presents the dominant text from all other texts. Users can easily look at the card and tell the name of the event by the size of the text.

Add Event button

In order to differentiate this particular button from another elements, we use the secondary color as this button's color. The position of this button is at the rightmost bottom, so that the user can differentiate the button from another different part. This also supports the '*Economy of Motion*' where most users are right handed, so this position will be easy for them to reach the button.

2. Event Detail Page (Normal View,Owner View)

2.1.1 Screenshot for Event Detail Page (Normal View)



2.1.2 Functional Requirement

- User can report and bookmark this event.
- User can upvote/remove upvote this event.
- User can join this event.
- User can chat with the owner via group chat.
- User can see the tags of interest.
- Location,date and number of participants will be shown.
- Type of event is shown above.

2.1.3 Elements of the page

- Back button (Arrow)
- Report button
- Bookmark button
- Picture of an event
- Title of the event
- All details of the event
 - Event picture
 - Event type
 - Event title
 - Number of user upvote for that event
 - Number of current participants

- Number of maximum participant
- Full detail of that event
- The day event will be held
- Upvote button
- Owner card
- Chat button
- Join button

2.1.4 Design Rationale

Event detail page is where the application directs user after they have toggled the event they are interested in. The goal of this page is to allow user to interact and connect with the owner. Hence all contents displayed in this page are related to that particular event, which was input from the owner when they created the event.

3 Buttons on the top of the page

These 3 buttons represent the additional action users can take, it's not primary subject to this page, therefore we make them look not so *dominant* to tell user that these buttons are *not as important/ dominant as other components* on this same page.

On the other hand, we still keep them as a usable *function* via the concept of '*Affordance*', this makes the user know that even if it's *not dominant but user can still use them as a normal function*. We use the icon instead of normal text to indicate that this particular element is not text but it's interactive. The icons are made bold in order to stand out from the normal text too.

Body of the page

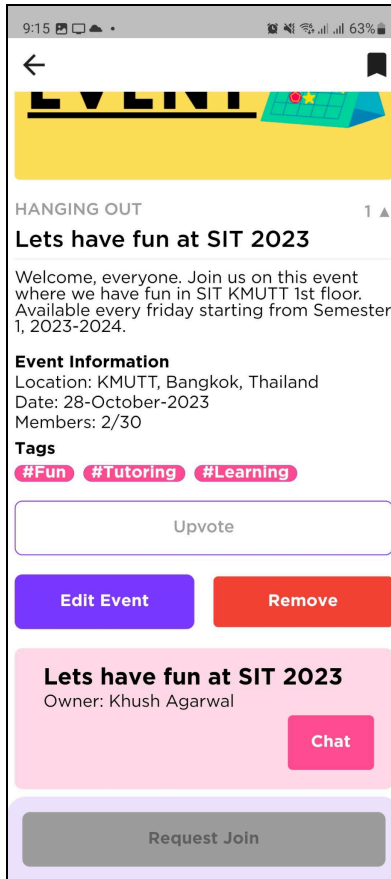
By using the '*Hierarchy*' concept, the most impactful element is placed at the top of the body, which is an event picture. From top to bottom, each elements are placed in order of impact respectively. The most impactful elements are placed on the top, while less impactful elements are placed below.

There is also use of '*Dominance*' element here where the tags are shown with the pink background. These tags are highlighted so user can have overview of what this event going to be. The owner card and its chat button are also created to be dominant too. The highly contrast of its color make user more focus on the element. It's rounded border also tell the user that this element is *interactive* like a normal button.

Join Button

It's *primary color highlights its role as the primary element*, we enlarge its size and also put bold white text inside to create *more dominance level*. The size and the color make it very clear to the user to see and know that this is the main feature of this page.

2.2.1 Screenshot for Event Detail Page (Owner View)



2.2.2 Functional Requirement

- User can bookmark this event.
- User can upvote/remove upvote this event.
- User can edit/remove this event.
- Owner can chat with the owner via group chat.
- User can see the tags of interest.
- Location, date and number of participants will be shown.
- Type of event is shown above.

2.2.3 Elements of the page

- Back button (Arrow)
- Bookmark button
- Picture of an event
- Title of the event
- All details of the event
 - Event picture
 - Event type
 - Event title
 - Number of user upvote for that event
 - Number of current participants

- Number of maximum participant
- Full detail of that event
- The day event will be held
- Upvote button
- Owner card
- Chat button
- Edit button
- Remove button
- Disabled Join button

2.2.4 Design Rationale

This is same page as the normal event detail page, except that it's the owner view. The owner here can do most of the action that normal user can, only that they are able to edit/remove this particular event.

All components are the same as the normal view on the event detail page. Except that there is no need for joining the event by the owner, so the join button is set to disabled mode by changing its theme color to gray.

Edit & Remove Button

These 2 buttons are placed side by side, since they share the similar usage purpose which is making changes to the event. But they are made to be different by the use of different dominant colors. Edit button has primary purple, while the remove button gets the pinkish tone.

3. Create Event Page


3.1 Screenshot of the Create Event Page

9:12 64%

← **Create Event**

Your Event's Name

Event's Name



Your Event's Detail

Event's Detail

Amount of People

2 - 500

Location of the Event

Location

Date of event

4 / 6 / 2023

Category

Other ▼

Tags

Add Tags +

☐ Allow Automatic Join ⓘ

Create Event

3.2 Functional Requirement

User will be able to do these things in order to create an event

- Create event title
- Choose event type
- Select picture for that event
- Select amount of maximum people who can join
- Select the location of that event
- Choose date for event
- Define tags which emphasize what event is about

- User can either choose to allow automatic join or approved by owner mode (Default)

3.3 Elements of the page

- Back button
- Event name textbox
- Picture upload area
- Event detail text area
- Amount of people text box
- Location text box
- Date of event button
- Category dropdown
- Add tag button
- Radio button for automatic join
- Create event button

3.4 Design Rationale

This page is where user can write down what is going to be for their event. Therefore most of the elements here are the input box.

Create Event Body

Hierarchy plays a major role for the input area. It ranked what is most important at the top and the less important are placed below respectively. *The size of some input indicates the meaning of that information* such as the input for the picture which is the biggest box in the body, as well as the event description which is the biggest text area since detail is the longest text part of the event.

Create Event button

Economy of motion and the *visual levels* are determined to position the create event button. The create event button supposed to be the *last element user interacts with* in this page since they must have completed the other input area first before publish the event. So we place this button at the bottom of page, which follows the natural visual of human that they will *focus anything at the bottom at the end*. *Economy of emotion* is also used here where user can easily press the create button, the create button is designed to take the whole bottom space so user can *reach easily* with their thumb.

4. Chat Room

4.1 Screenshot of the Chat Room



4.2 Functional Requirements

- Users can see different chat groups of different events that they joined

4.3 Elements of the page

- Title of the page
- Chat group cards which contains
 - Event photo
 - Event title

4.4 Design Rationale

This page was designed for communication purpose where users can chat and stay in touch with the events and members that they joined

Title of the page

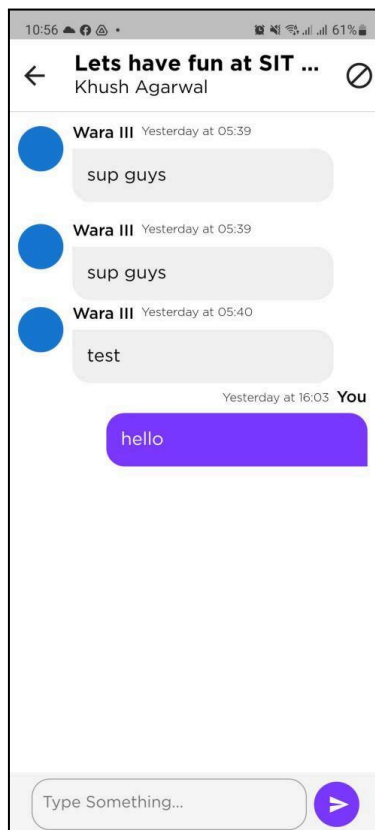
The title was put on top of the page by applying the **Consistency and Standards** of Usability Heuristics. Furthermore, we applied the rule of **Visibility of System Status** by letting users know where they are at now.

Chat Group Cards

There are many information in each event and we provided only two pieces of information (Event photo and title) on the chat group card, to allow users to easily see and access each event chat. In doing so, we have to applied **Aesthetic and Minimalist Design** of Usability Heuristics by keeping things simple and straightforward.

5. Chat Page

5.1 Screenshot of the Chat Page



5.2 Functional Requirements

- Users can see other users' messages in this page, who joined the same event
- Users can send their messages to chat

- Users can block the chat if they do not find it interesting or suitable for them

5.3 Elements of the page

- Title of chat which contains
 - Event title
 - Post owner name
- Block Icon
- Chat messages and sender profiles
- Message writing bar and sending icon

5.4 Design Rationale

Chat room is designed for the purpose of allowing users to share their skills and experiences for the specific event they are joining.

Title of chat

The title of the chat which contains the event title and the post owner was put on top of the page by following the **Hierarchy** design principle to make it more prominent. So that, users can easily see and check which event chat they are in. This could be useful when users have joined many events to avoid confusion.

Block Icon

Block icon was put on the top right corner of the page to make it a bit harder to navigate to avoid allowing users to mistakenly press it. Thus, we made use of **Economy of Motion** of design principle for this. Furthermore, we show a prompt when users press that icon to allow users to confirm their action, which is using the **User Control and Freedom** of Usability Heuristics, which is to prevent users from mistakenly pressing it and making errors.

Chat Messages and Sender Profile

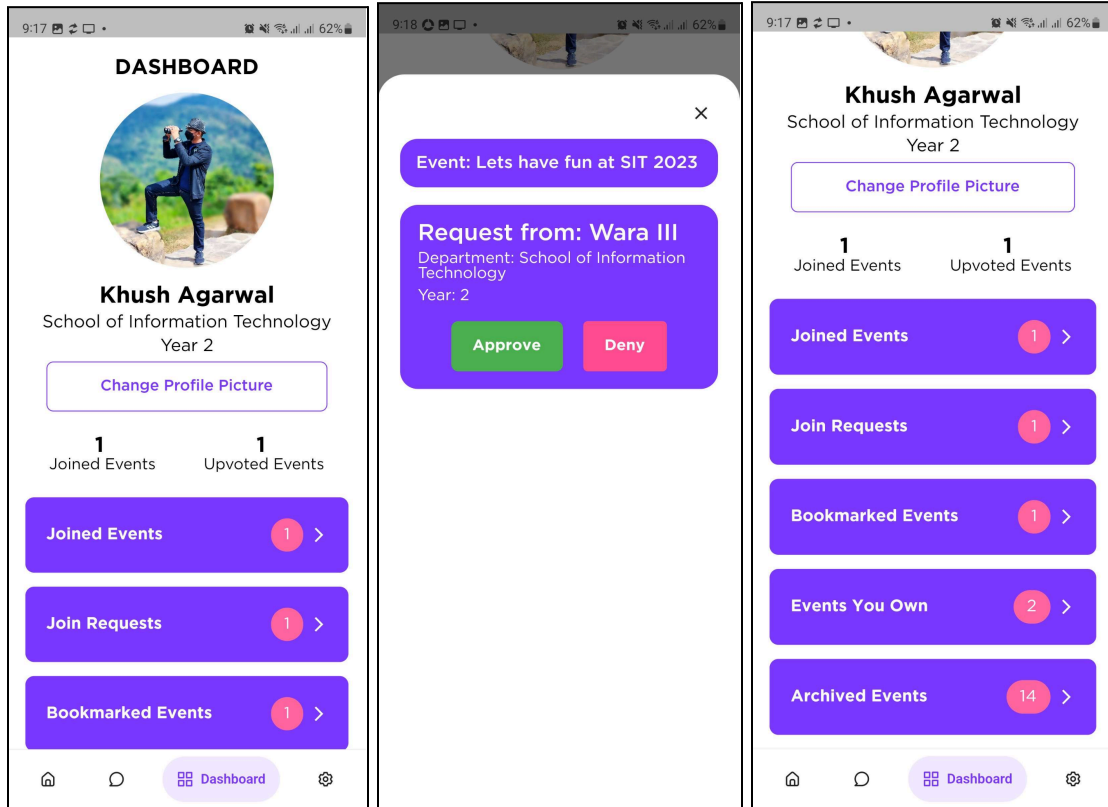
Chat messages and sender profiles were shown on the left side of the screen. This idea is following the **Consistency** and **Standards** of Usability Heuristics, since many other systems use this design for chat feature. Sender profiles were put next to each text by following the **Unity** of design principles to allow users to see which text is sent by who.

Message Writing Bar and Sending Icon

Message writing bar and sending icon were put together by following the **Unity** of design principles to show users they are related and they can do one action by combining two components which is writing message and sending action.

6. Profile Page

6.1 Screenshot of the Profile page



6.2 Functional Requirement

- User can have overall look on the events they interacted with
 - Check the event they joined
 - Check for the join request to their event
 - Check the event they bookmarked
 - Check all events the owner has
 - Check all archived events
- User can change their profile picture

6.3 Elements of the page

- User picture
- User's name
- User's faculty
- User's year
- Change profile picture button
- Summary of the event they join

- Summary of the event they upvoted
- Joined events button
- Join request button
- Bookmarked event button
- Events you own button
- Archived event button
- Request card

6.4 Design Rationale

This is one of the main pages we select to include in the navbar. Its main purpose is to collect all user's activity within the app in one place so it helps user in managing all join request or even keeping track of the event they are looking at.

Profile Part

User profile is presented as a *dominant element* in this area. This page is easily identified by user who just saw a picture and they will naturally know that this page is profile page. The texts are shown in bold texts at the center of the page where user will first look at their profile picture first and then followed by the details of their own profile. The center *bold text* represents the *key information of that user* whether it's name, the user's faculty or user's year.

The change profile picture is painted in white with the purple border, this soft theme represents the less important component of this page. The edit profile picture is *not as dominant as* the profile and its detail.

Lastly for the profile part is the summaries of all user's activity in 2 numbers; the first one on the left is the total joined events, the right one is the total upvoted event. The number is presented using big font. It helps user more focused on the number of their total activity.

Lists of all interacted events

These lists are varied based on the user's interaction on that specific event. But all the buttons here are painted in purple with rounded borders so that users can *naturally understand* that these are the major features in a button format.

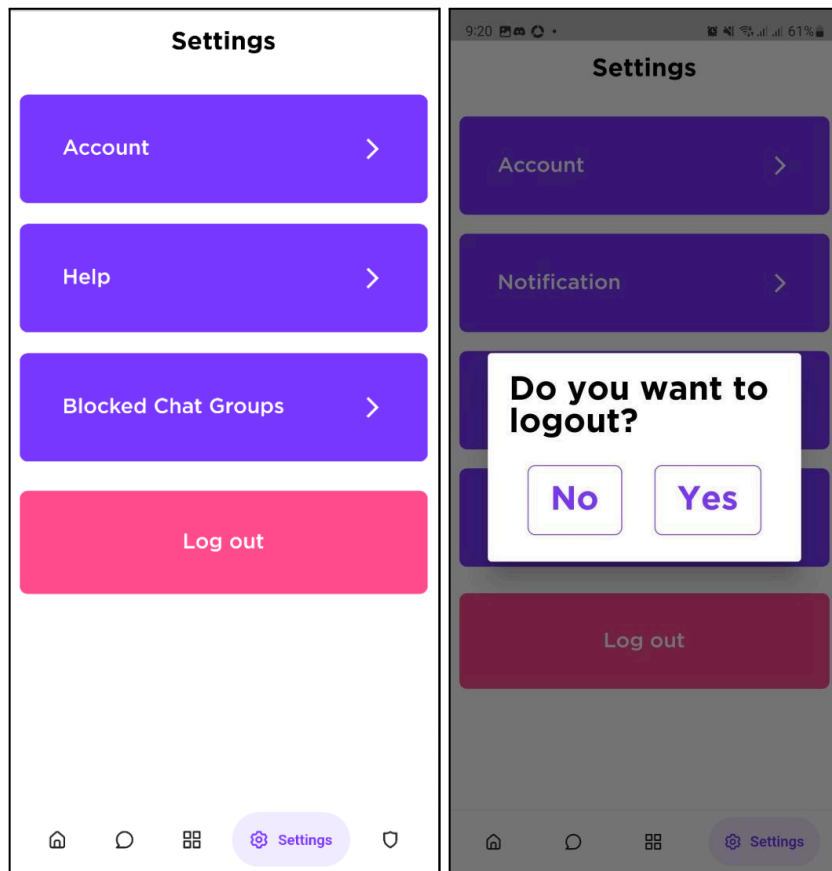
If we look closely, there is also number placed inside the button. These numbers are given pink background so user can differentiate this area with the rest part of the button. The pink color plays an opposite feeling of the purple *where user can sense the alert feeling in the button*. This helps user to be easily notified by all upcoming join requests that they are asked to check *as soon as possible*.

User request card

This card shows the user a request from someone to join the user's event. The actions to be taken here can be Approve or Deny. These 2 choices for user are ***painted differently to give user a natural feeling of that decision***. The approve button is painted in green, this color naturally gives positive, acceptance, welcome feeling. While the deny button is painted in secondary pink color, this tone gives the same feeling as red color which is the tone of anger, denial, aggression.

7. Main Setting Page

7.1 Screenshot of Main Setting Page



7.2 Functional Requirements

- Users can adjust the account setting
- Users can go and read documentation
- Users can log out of our system

7.3 Elements of the page

- Page title
- Account setting button
- Help button
- Block Chat Groups button
- Log out button

7.4 Design Rationale

On this page, we can see all the functional components that are hardly used by users, yet still important to have in every application. Users can see all the setting in this page and adjust as they desire.

Page Title

The title was put on top of the page by applying the **Consistency and Standards** of Usability Heuristics. Furthermore, we applied the rule of **Visibility of System Status** by letting users know where they are at now.

Account, Help and Block Chat Group Buttons

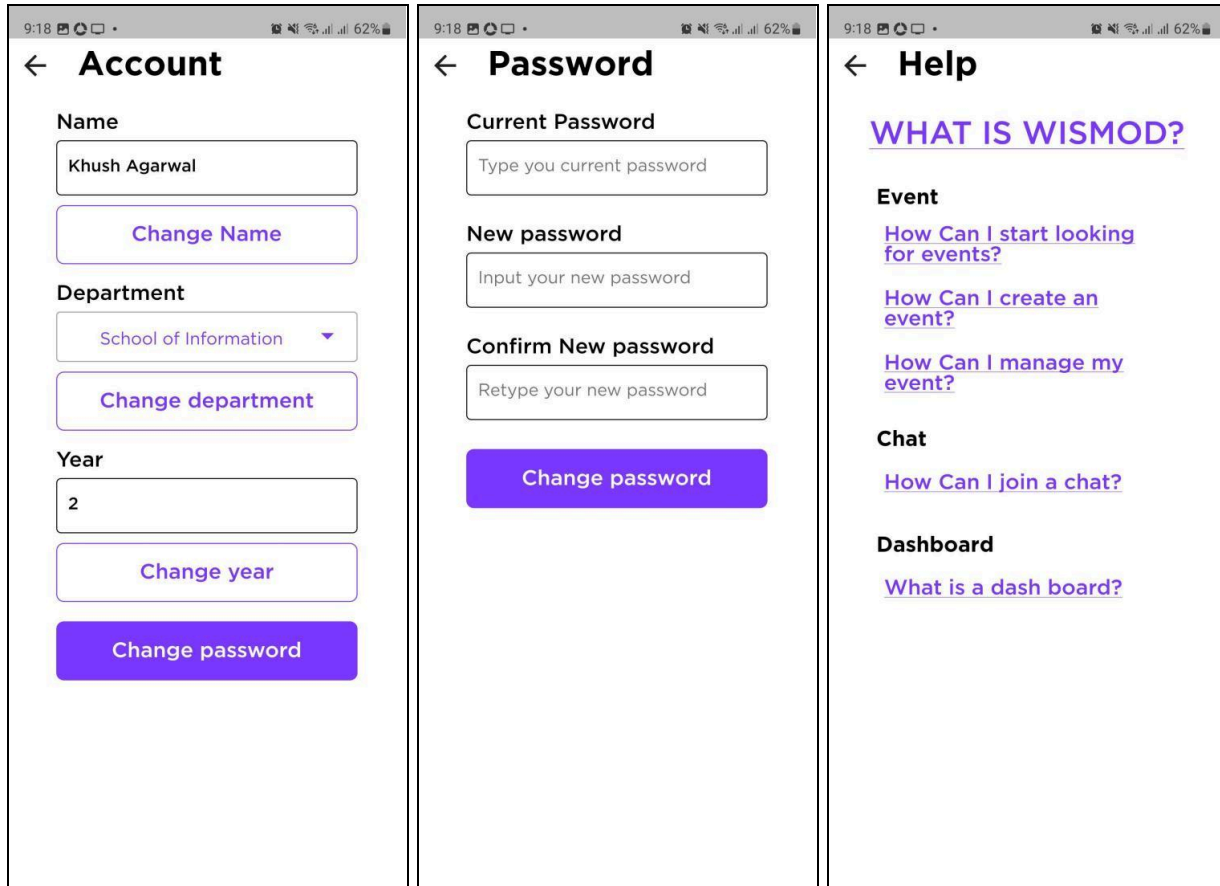
These three buttons are given the same background colors although they are different components because they all are navigating to a new page, which is doing the same action. Thus, the design is based on **Unity and Variety** of Design Principles. In addition, we also make sure that users know they will be navigated to a new page by giving a forward arrow on the button. This one is following the rule of **Affordance** of Design Principles, by applying an arrow sign as a navigation as in real world.

Logout Button

This button is given a different color since it is not navigating to any other pages. Instead, we make a prompt when users click that button to let users confirm that they want to log out of the system and to avoid mistaken press by users. Therefore, **Error Prevention** of Usability Heuristics was used for this design.

8. Sub-Setting Pages

8.1 Screenshot of the Sub-Setting Pages



8.2 Functional Requirement

- Users can see their current name, department, year and password.
- Users can change name, department, year and password.
- Users can read documentation about our application although our application is simple to understand

8.3 Elements of the page

For Account Setting Page

- Back button (Arrow)
- Title of page
- A form which contains
 - Name input and change button
 - Department selection box and change button
 - Year input and change button
 - Changing password button

For Password Changing Page

- Back button (Arrow)
- Title of page
- A form which contains
 - Current password label and its input field
 - New password label and its input field
 - Confirm new password label and its input field
 - Changing password button

For Help Page

- Back button (Arrow)
- Title of page
- Link text called “WHAT IS WISMOD?”
- A title “Event” which contains
 - Three link text related to “Event”
- A title “Chat” and its link text
- A title “Dashboard” and its link text

8.4 Design Rationale

For Account Setting Page

Account setting page is designed for users who want to change their information for some reason. This page allows users to change their information that is displayed on their profile.

Back Button and Page Title

Back button to go back and page title to let users know where they are now. They both are designed following the **Established Industry Conventions**, which is part of the **Consistency and Standard** of Usability Heuristics.

Form Component (Name, Year and Department)

Name, year and department are in the same form, but they are different elements. Thus, we make sure they are not the same by differentiating with titles and inputs, and make each part as a group although they are from the same form. Therefore, we applied the rule of **Unity and Variety** of Design Principles for it.

Password Changing Button

Password changing button was separated from the form above by giving our primary background to the button. We applied **Unity and Variety** of Design Principles for this one as well by giving a purple background.

For Password Changing Page

Password changing page was designed for users to allow them change passwords in case of security threat or other reason. And you may be wondering why we did not put it in the

account setting page. Mainly because password is not part of display information for the profile.

Back Button and Page Title

Back button to go back and page title to let users know where they are now. They both are designed following the **Established Industry Conventions**, which is part of the **Consistency and Standard** of Usability Heuristics.

Form Component (Current Password, New Password and Confirm Password)

Current Password, New Password and Confirm Password are different elements but they all work together. So we labeled each input field to let users know what each input is intended for. Therefore, we applied the rule of **Unity and Variety** of Design Principles for it.

Password Changing Button

Password changing button was separated from the form above by giving our primary background to the button. We applied **Unity and Variety** of Design Principles for this one as well by giving a purple background.

For Help Page

Password changing page was designed for users to allow them read and explore about our application for more information. In this page, users can learn more about our application and how to use it.

Link text called “WHAT IS WISMOD?”

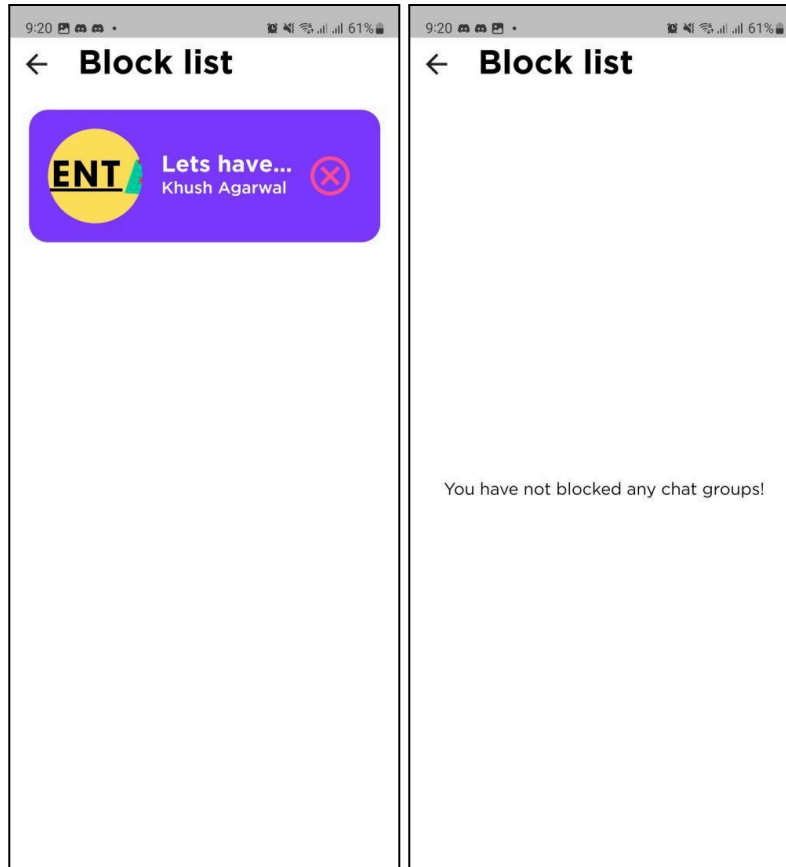
The text was put at the very top level of the page since it is important to let users know “What is WisMod” before coming with other questions and answers. Thus, this one title was designed by applying the rule of **Hierarchy and Dominance** of Design Principle. In addition, we underlined the text by applying the **Interaction** rule of Design Principles, which let users know that it is clickable to see more information.

Event, Chat and Dashboard

These three components share the same action, which is providing more information. Although they are different components, they are aligned symmetrically since they perform the same tasks. Therefore, we applied the **Unity and Variety** of Design Principles for designing those components. Moreover, we underlined the text by applying the **Interaction** rule of Design Principles, which let users know that they are clickable to see more information.

9. Block List Page

9.1 Screenshot of Block List Page



9.2 Functional Requirements

- Users can see the list of chat group that they blocked
- Users can unblock the chat group

9.3 Elements of the page

For Block List Page with Blocked Chat Group

- Page title
- Blocked chat group cards which contains
 - Event photo
 - Event name
 - Event owner
 - Cross Icon

For Block List Page with Blocked Chat Group

- Page title

- “You have not blocked any chat group” title

9.4 Design Rationale

This page was designed to let users see which chat groups are blocked by them and to allow them unblock if they want to.

For Block List Page with Blocked Chat Group

Page Title

The title was put on top of the page by applying the **Consistency and Standards** of Usability Heuristics. Furthermore, we applied the rule of **Visibility of System Status** by letting users know where they are at now.

Block Chat Group Card

Since we are using **Aesthetic and Minimalist Design** idea, we only included event photo, event name, and event owner on the card although there are many information in each event. And we put the cross icon to unblock the blocked chat group on the rightmost side of the card by following the **Economy of Motion** of Design Principles.

For Block List Page with Blocked Chat Group

Page Title

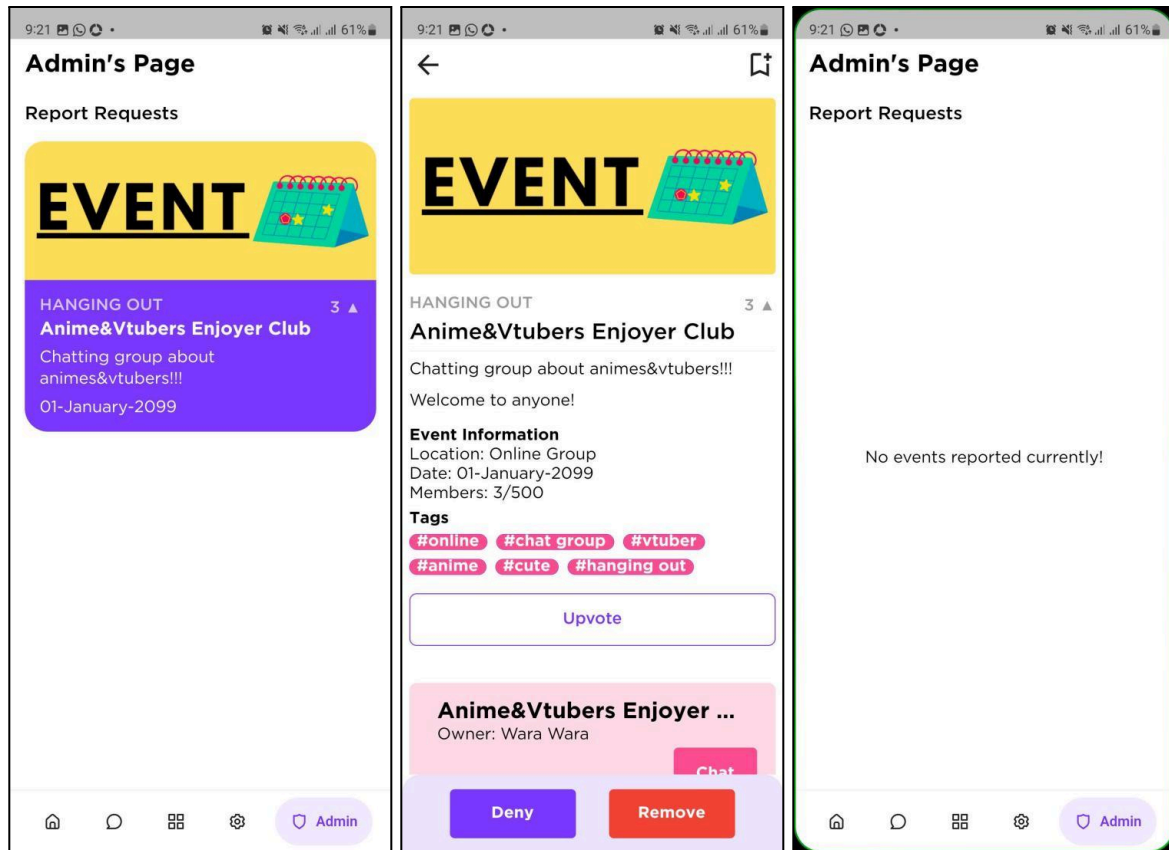
The rule we applied for this page title is the same as in “For Block List Page with Blocked Chat Group”

“You have not blocked any chat group” title

This title was placed in the middle of the page to inform users about the status of the page, which is having no blocked chat groups at the moment. Therefore, we applied the **Visibility of System Status** of Usability Heuristic by informing about the current status, instead of just displaying a blank page.

10. Admin Dashboard

10.1 Screenshot of the Admin Dashboard



10.2 Functional Requirement

- Admins can see events that are reported by users.
- Admins can click the event to see detail
- Admins can remove the event if the event is violating the rules
- Admins can deny the report if the event is not violating any rules

10.3 Elements of the page

For Admin Page with Reported Event

- “Admin’s Page” title
- “Report Requests” title
- Event card which contains
 - Event picture
 - Event type
 - Event title
 - Number of user upvote for that event

- Brief detail of that event
- The day event will be held

For Event Detail Page with Admin View

- Back button (Arrow)
- Bookmark button
- Picture of an event
- Title of the event
- All details of the event
 - Event picture
 - Event type
 - Event title
 - Number of user upvote for that event
 - Number of current participants
 - Number of maximum participant
 - Full detail of that event
 - The day event will be held
 - Upvote button
 - Owner card
 - Chat button
 - Deny button
 - Remove button

For Admin Page without Reported Events

- “Admin Page” title
- “Report Requests” title
- “No Events Reported Currently” title

10.4 Design Rationale

Admin page was designed for controlling users’ activities on our platform. We keep our platform clean and relevant for every user by not allowing irrelevant contents, and we monitor and control those contents from this page.

For Admin Page with Reported Event

Admin’s Page Title

The title was put on top of the page by applying the **Consistency and Standards** of Usability Heuristics. Furthermore, we applied the rule of **Visibility of System Status** by letting users know where they are at now.

“Report Requests” title

Since there are event cards in this page, admin may confuse what those cards are doing on this page. We, therefore, provided the “Report’s Requests” title on top of the page to

avoid discombobulating admin by applying the **Hierarchy and Dominance** of Design Principles.

For Event Detail Page with Admin View

Please refer to section 2.2.4 Design Rationale for Event Detail Page

Deny and Remove Buttons

The only difference from section 2.2.4 is that Event Detail Page has a “Deny” button in “Edit” button place. “Deny” button and “Remove” button are placed side by side, since they perform the opposite actions to the same event. For instance, they are like “Yes” or “No” options, which are opposing each other, yet should still be put together. Since Remove is a dangerous operation, it was given a red background. But for “Deny” button, we simply gave our primary color background since it is a normal operation. Basically, we applied the Consistency and Standards of Usability Heuristics by giving those colors.

For Admin Page without Reported Events

Admin’s Page Title and Report Requests Title

Admin’s Page title and Report Requests title share the same rules that I mentioned in the first part of this section 8.4 Design Rationale.

“No Events Reported Currently” title

This title was placed in the middle of the page to inform admin about the status of the page, which is having no reported events at the moment. Therefore, we applied the **Visibility of System Status** of Usability Heuristic by informing about the current status, instead of just displaying a blank page.