



School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - GROUP ASSESSMENT

Unit of Study: INFO2222

Assignment name: Project2

Tutorial time: RE03 Tutor name: Tianyi Zhang

DECLARATION

We the undersigned declare that we have read and understood the [University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy](#), an, and except where specifically acknowledged, the work contained in this assignment/project is our own work, and has not been copied from other sources or been previously submitted for award or assessment.

We understand that failure to comply with the *Academic Dishonesty and Plagiarism in Coursework Policy* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

We realise that we may be asked to identify those portions of the work contributed by each of us and required to demonstrate our individual knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

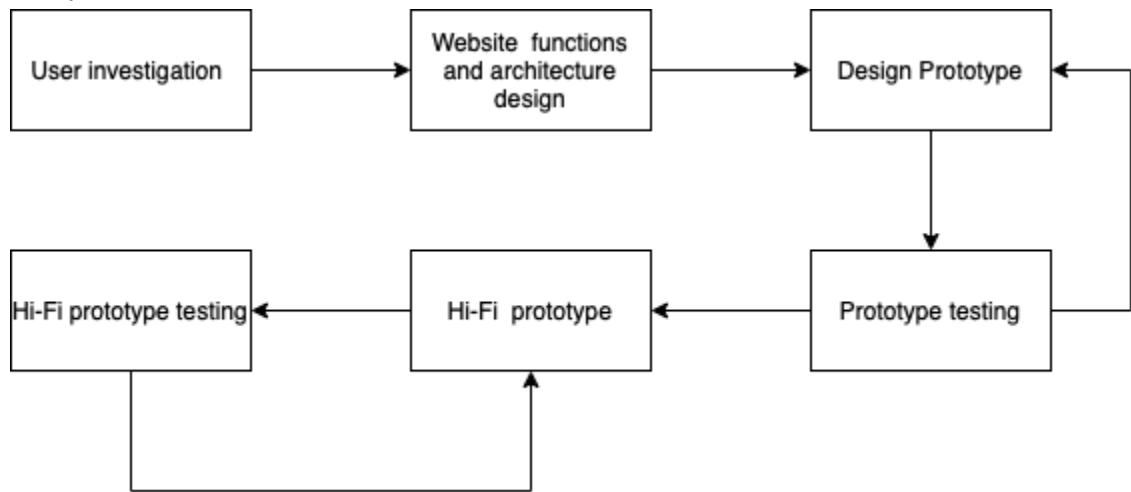
Project team members				
Student name	Student ID	Participated	Agree to share	Signature
1. Langyi Chen	510090391	<input checked="" type="checkbox"/> Yes / <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes / <input type="checkbox"/> No	陈朗毅
2. Wentao Gao	480187808	<input checked="" type="checkbox"/> Yes / <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes / <input type="checkbox"/> No	高文韬
3.		Yes / No	Yes / No	
4.		Yes / No	Yes / No	
5.		Yes / No	Yes / No	
6.		Yes / No	Yes / No	
7.		Yes / No	Yes / No	
8.		Yes / No	Yes / No	
9.		Yes / No	Yes / No	
10.		Yes / No	Yes / No	

INFO2222 usability report

Group name: RE03_Team4
Name: Langyi Chen, Wentao Gao

Summary

Lifecycle



Finished items

1. User investigation
2. Personas for our target users
3. Card sorting
4. Information architecture
5. Prototype
6. Hi-Fi prototype

Individual percentage

Name	Contribution
Langyi Chen	50%
Wentao Gao	50%

Body of report

Step1: User Investigation

Survey

We published a survey on Ed, the questions on the survey are for students and staff, and we received the respondents from both students and staff.

The link of our survey:

https://docs.google.com/forms/d/e/1FAIpQLSeygsmWWA0b8re8yQO7MYuSiEmjXJx128hJZJV4ctim5bIZQ/viewform?usp=sf_link

There are 15 responses in total, 13 from students and 2 from staff. We collected the basic information (age, major etc.) from them to narrow the target users and create the personas. We also collected the information about how they feel about canvas and Ed to help us design our website. Most students are in the age of 18-22 and their majors are related to computers. All of the respondents prefer text as the form to show the contents on the websites. Most respondents feel satisfied with Ed, but the satisfactory level of canvas is lower (The graphs of analysis of survey are shown in appendix 1-1, 1-2, 1-3, 1-4, 1-5).

We narrowed the target users to two different kinds, and created the personas for each kind.

The personas are shown as below.

Persona A

Fictional Name	James Li
Occupation	A student of USYD
Major	Advanced Computing
Age	22
Goals of using this website	Use this website to get the resources of his study easier. And use this website to communicate with others easier.

Persona B

Fictional Name	Eric Han
Occupation	A staff of USYD
unit	INFO3333
Age	40
Goal of using this website	Use this website to communicate with students and upload resources easier.

Step2: Navigation design

Card Sorting

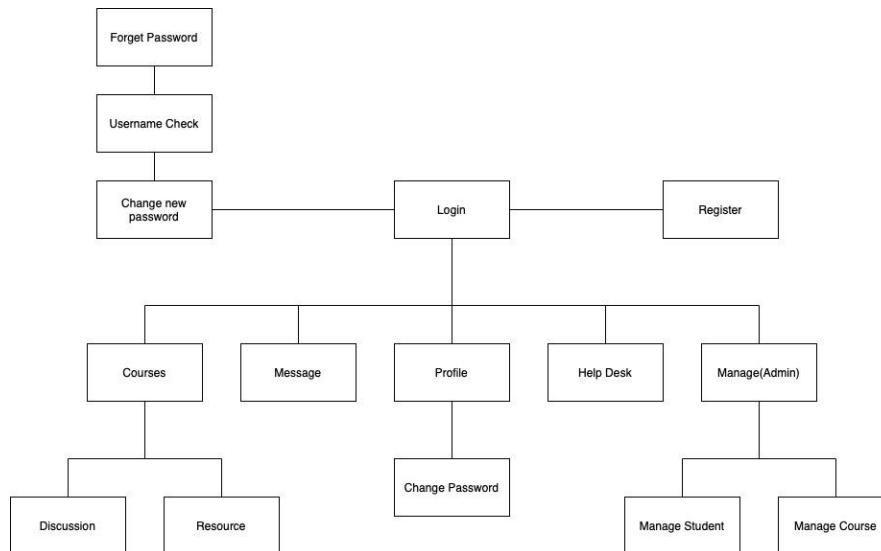
We invited 10 target users to do the open card sorting. There are 12 cards and invited target users would allocate these cards into different groups by themselves (The details of cards and how they are allocated are shown in appendix 2-1, 2-2, 2-3, 2-4, 2-5, 2-6). Some responses allocated these 12 cards according to functions, some responses allocated these 12 cards according to web pages, and some responses allocated these 12 cards according to characters

(students and admins). The similarity matrix of how the cards are allocated is shown in appendix (appendix 2-7).

This information helps us to allocate the functions to different kinds of users, design the webpage and decide the functions we will implement.

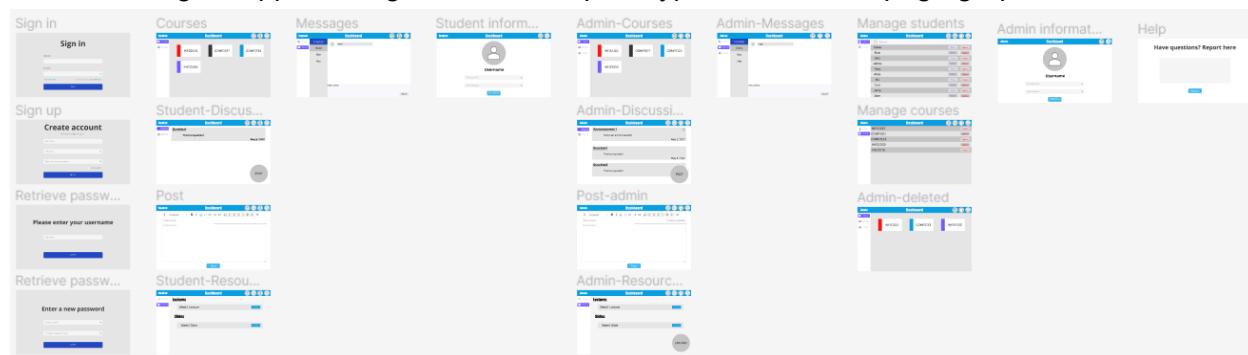
Information Architecture

This is the sitemap of our website.



Step 3: Design-Evaluate (Prototype)

We use the figma app to design our website prototype and the total page graph shown below.



We first designed a sign in page and it is the homepage of our website.(shown in graph 3-1). If the user doesn't have an account, just type **create now** at the bottom right of this page, and the website will jump to the sign up page. The user needs to create a user name and a password. If the user is a staff member, he only needs to put a check mark in the i am a stuff box in the lower right corner of the page to get administrator privileges(shown in graph 3-2). After sign up, the website will jump to the sign in page to let the user login. If the user forgot his password, just type **forgot password** at the bottom right of this page. The website will jump to the forgot password page, first the website will ask you to enter the username of the account where you

have forgotten your password(shown in graph 3-3). After that, you can set a new password for your account(shown in graph 3-4).

Once you enter the created account and right password, the website will jump to the dashboard page. If you are a student, the leftmost top bar of the page will be displayed as student, and there will be **courses** and **messages** modules in the sidebar. Also, if you are staff or have administrator privileges, the leftmost top bar of the page will be displayed as admin, and there will be **courses**, **messages** and **manage** modules in the sidebar(shown in any graph in the dashboard page).In the right side of the top bar, there are three buttons: **report**, **log out** and **account information**. The report button allows users to report problems encountered while using the site and suggestions for improvement.(shown in graph 3-5). Log out button is used to allow users to log out of their current account and jump to the sign in page immediately. The account information button is used to go to the personal information page, where you can upload a picture to use as an avatar, or change your password.After change will jump to sign in page.(shown in graph 3-6-1 and 3-6-2) The Dashboard page initially showed the courses page. For students, this page will show all the courses you are studying, for admin, this page will show all the courses you are managing(shown in graph 3-7-1 and 3-7-2). Click on a course and you will go to the course's dedicated page, where the sidebar has **discussion** and **resources**. As a student, you can post discussions and questions about the content of this course. Click the post button in the lower right corner of the page and you will be taken to the post page, where you can describe your problem in detail, with any attachments (pictures, any form of files and videos). When you have finished describing the problem, just click on the post bottom below. Your post will be shown in the discussion area(all posts will be sorted by time, the newer the post, the higher the position in the discussion area.)(shown in graph 3-8-1 and 3-8-2). As a admin, most of your permissions will be the same as a student, but on the right side of the title bar in post page, you have an option **announcement**, and if selected, your post will be displayed on top in the discussion area (the horn logo in the upper right corner of the post indicates that the post is a topped announcement)(shown in graph 3-9-1 and 3-9-2). Click **resources** in the sidebar and you will jump to the resources download page, which contains all the learning materials for this course. Students can only download it, and the admin can upload. In the right side of the top bar, there is a fourth button **go back**, which is used to go back to the last page user viewed(this button is only available in the secondary page, because the first page have no page to go back)(shown in graph 3-10-1 and 3-10-2). If the user clicks the go back now, the website will go back to the initial page with **resources**, **messages** and **manage**(admin only). If you click **messages** in the sidebar, the website will jump to the messages page. Any user can send private messages to other users here(there is no difference between student and admin on this page)(shown in 3-11-1 and 3-11-2). For users with administrative privileges, there is an additional **manage** module in the sidebar. Click it and you will jump to the admin only page. In this page, you can manage all students and courses that you are managing. For a student, you can mute or delete him. If a student is muted, he will not be able to post in the Discussion area of any courses. If a student is deleted, his account will no longer be able to pass this website.(shown in 3-12). For a course, you can delete this course, the entrance to this course will not be visible in the courses page(shown in 3-13).

During the design, we had two iterations. The first iteration is **forget the password**. We found that users could no longer log in if they forgot their password, so we decided to add this feature to the **sign in** page(shown from graph 3-iteration1 to graph 3-1, graph 3-3 and graph 3-4). *The second iteration is a report problem.* Users may encounter many kinds of problems when using the website, so we added an additional report function to the top bar of each page after we finished the website prototype(shown form graph 3-iteration 2 to graph 3-5 and all pages with top bar).

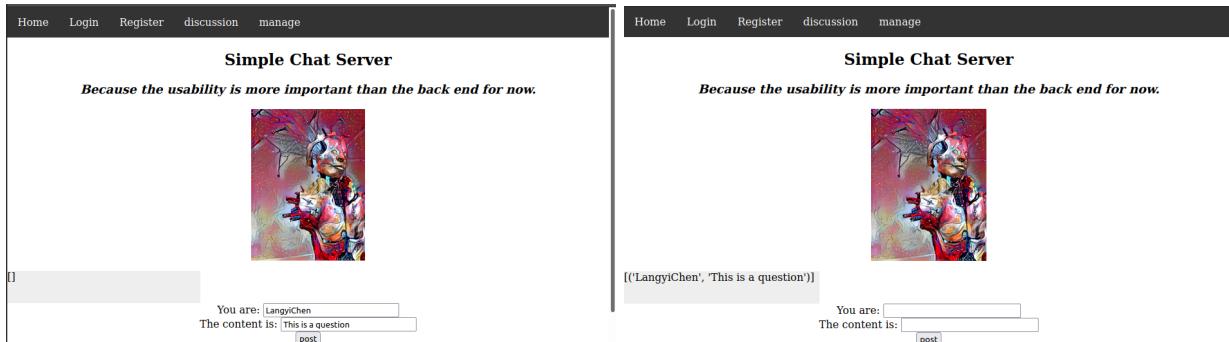
Our tutor did a usability test for our prototype. Before testing, we prepared two tasks for our design. First task is delete a course as an administrator, second task is post a question in the discussion area of a course. During the test, our tutor did not encounter any problems in testing these two tasks, and it took only a little time.

Step 4: Design-Evaluate (Hi-Fi Prototype)

Based on the code of the previous security part, we added two functions for the website, the discussion function and manage function.

Discussion function

We implemented a simple discussion function on our website. Users can post their discussions on the page and all users can see the discussions on the discussion page. User needs to enter the contents of the discussion and his username then click the post button to post the discussion. The discussion will show on the discussion page with its poster's name. The result of our page is shown below.



(The frontend code and backend code of discussion function is shown in the appendix 4-1, 4-2, 4-3, 4-4)

Manage function

We finished a manage function for the administrator of this website. In this function, the administrator can mute or delete users. In the manage page, If admin input the username he

wants to manage and choose mute and click confirm.

The screenshot shows the homepage of the "Simple Chat Server". At the top, there is a navigation bar with links: Home, Login, Register, discussion, and manage. Below the navigation bar, the title "Simple Chat Server" is displayed, followed by the subtitle "Because the usability is more important than the back end for now.". A colorful illustration of a stylized robot or alien head is centered on the page. In the bottom left corner, there is a message box containing the text "[('LangyiChen'), ('WentaoGao',)]". Below this, there is a form for managing users. It includes a "Username:" input field with "LangyiChen" entered, and three radio buttons: "mute" (selected), "unmute", "delete", and "confirm".

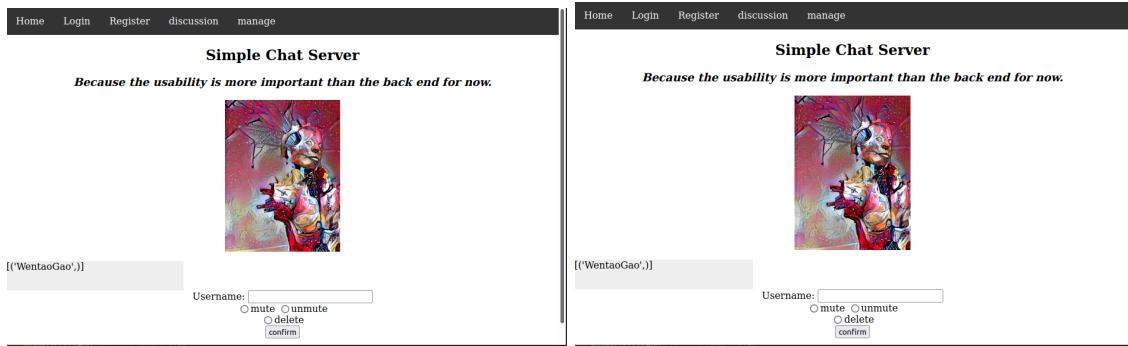
this user will be muted, then if he posts anything in the discussion area, there will be an error message and his post will fail.

This screenshot shows two side-by-side pages. The left page is the user management screen for "LangyiChen", where the "mute" option is selected. The right page shows the result of a failed post attempt. The user "LangyiChen" has posted the message "This is a question". However, an error message "ERROR: \${error_type}" is displayed, along with the page number "Page \${error_msg}".

If now the admin wants to unmute this user, just input username and choose unmute and click confirm. This user can post again in the discussion area!

This screenshot shows two side-by-side pages. The left page shows the user management screen for "LangyiChen", where the "unmute" option is selected. The right page shows the result of a successful post attempt. The user "LangyiChen" has posted the message "This is a question".

If the admin wants to delete a user, input his username, choose delete and click the confirm. This user will be deleted from the website and database.



(The frontend code and backend code of discussion function is shown in the appendix 4-5, 4-6, 4-7, 4-8)

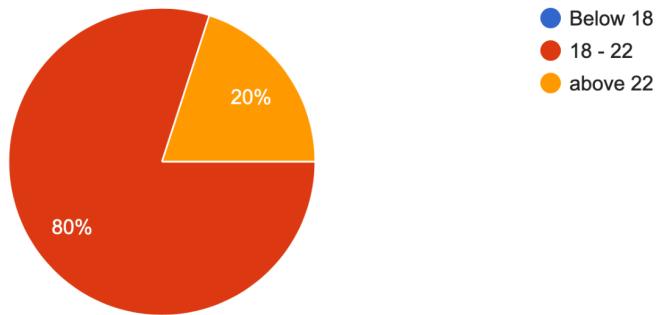
Incremental development plan

1. Make the website pages style be similar to our prototype, to let users have better experience and visual feeling while using our website.
2. Complete all functions on our prototype to make the usability of this website higher.

Appendix

Age

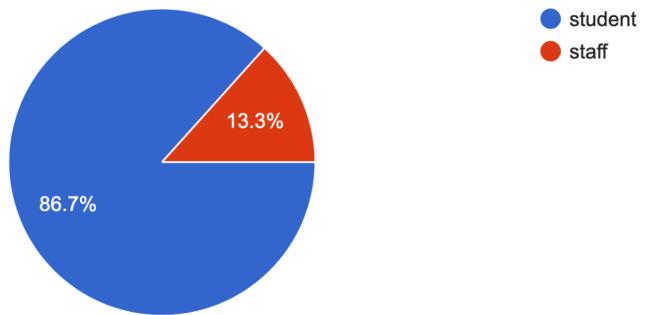
15 responses



(graph 1-1)

Are you a student or a staff of USYD?

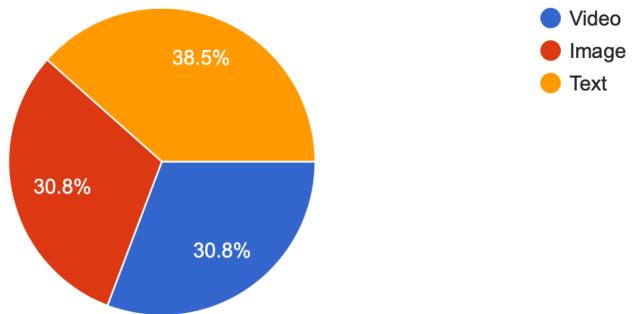
15 responses



(graph 1-2)

What form of context you prefer?

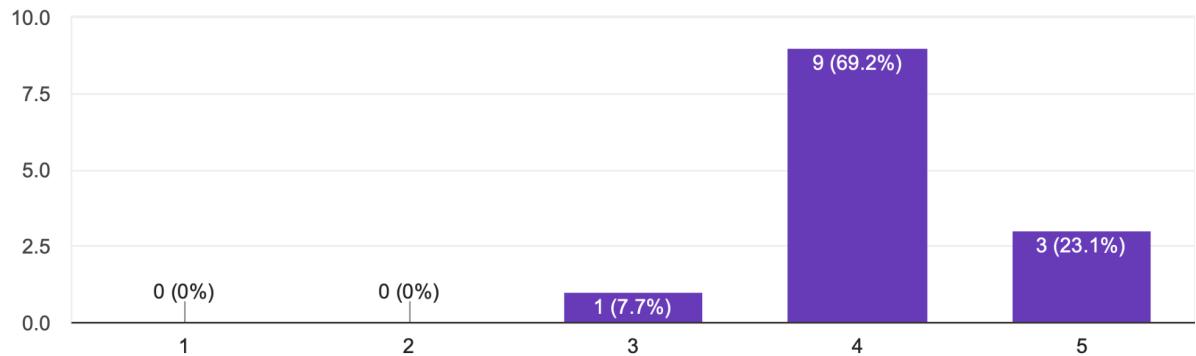
13 responses



(graph 1-3)

From 1 to 5, how much do you like ED website?

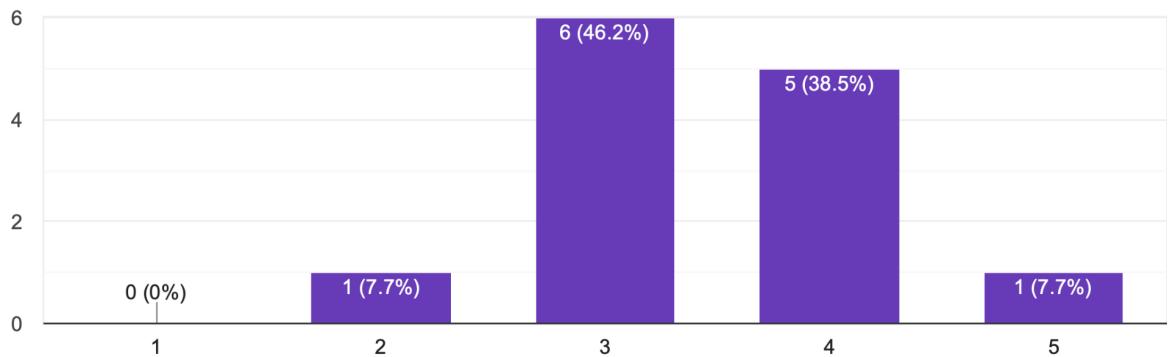
13 responses



(graph 1-4)

From 1 to 5, how much do you like Canvas website?

13 responses



(graph 1-5)

Card	Sorted into	Categories	Frequency	Position
Ask for help	6 categories	<ul style="list-style-type: none"> Help desk Help Help page Related to password All user functions All users 	3 3 1 1 1 1	1.0 1.3 1.0 3.0 7.0 9.0

[Show fewer categories](#)

Change password	7 categories	<ul style="list-style-type: none"> Password Profile Help Profile page Related to password All user functions All users 	3 2 1 1 1 1 1	2.0 1.0 1.0 1.0 3.0 4.0
-----------------	--------------	---	---------------------------------	--

(graph 2-1)

Delete a course	7 categories	<ul style="list-style-type: none"> Admin Delete a course Manage Manage course Management Admin functions Admin page 	4 1 1 1 1 1 1	2.0 1.0 1.0 1.0 2.0 2.0
-----------------	--------------	--	---------------------------------	--

[Show fewer categories](#)

Download the resource of one unit	5 categories	<ul style="list-style-type: none"> Resource Course page Learning All user functions All users 	6 1 1 1 1	1.2 2.0 2.0 6.0 7.0
-----------------------------------	--------------	--	-----------------------	---------------------------------

[Show fewer categories](#)

(graph 2-2)

Forget password	7 categories	Password	3	2.3
		Login	2	1.5
		Login page	1	2.0
		Related to password	1	2.0
		All users	1	3.0
		Login/Register	1	3.0
		All user functions	1	5.0

[▲ Show fewer categories](#)

Login	7 categories	Login/Register	3	1.3
		Login	2	1.5
		Login page	1	1.0
		All user functions	1	2.0
		All users	1	2.0
		Password	1	2.0
		Related to password	1	5.0

[▲ Show fewer categories](#)

(graph 2-3)

Mute an account	7 categories	Admin	4	1.3
		Admin page	1	1.0
		Manage users	1	1.0
		Mute account	1	1.0
		Manage	1	2.0
		Management	1	2.0
		Admin functions	1	3.0

[▲ Show fewer categories](#)

Post discussion	5 categories	Discussion	6	1.5
		Admin functions	1	1.0
		Discussion page	1	2.0
		Learning	1	3.0
		All users	1	5.0

[▲ Show fewer categories](#)

(graph 2-4)

Register	7 categories	Login/Register	3	1.7
		Register	2	1.0
		All user functions	1	1.0
		All users	1	1.0
		Password	1	1.0
		Register page	1	1.0
		Related to password	1	4.0

[Show fewer categories](#)

Send message to others	6 categories	Message	4	1.0
		Chat	2	1.0
		Message page	1	1.0
		Send Message	1	1.0
		All user functions	1	4.0
		All users	1	6.0

[Show fewer categories](#)

(graph 2-5)

Upload the resource of one unit	5 categories	Resource	5	1.8
		Admin	2	2.5
		Course page	1	1.0
		Learning	1	1.0
		Admin functions	1	4.0

[Show fewer categories](#)

View discussion	5 categories	Discussion	6	1.5
		Discussion page	1	1.0
		Learning	1	4.0
		All user functions	1	8.0
		All users	1	8.0

[Show fewer categories](#)

(graph 2-6)

Similarity matrix



[View discussion](#)

90	Post discussion									
10	20	Upload the resource of one unit								
30	20	70	Download the resource of one unit							
20	10	0	20	Login						
20	10	0	20	80	Forget password					
20	10	0	20	40	60	Change password				
20	10	0	20	70	50	40	Register			
20	10	0	20	30	30	40	30	Ask for help		
20	10	0	20	20	20	20	20	20	Send message to others	
0	10	30	0	0	0	0	0	0	Mute an account	
0	10	30	0	0	0	0	0	0	Delete a course	
80										

(graph 2-7)

Sign in

Username

Password

[Forgot password](#)

Don't have an account? [Create now!](#)

[Sign In](#)

(graph 3-1)

Create account

Already have an account? [Sign in](#)

Username

Password



Type your password again



I am a stuff

[Sign up](#)

(graph 3-2)

Please enter your username

Username

Confirm

(graph 3-3)

Enter a new password

New password



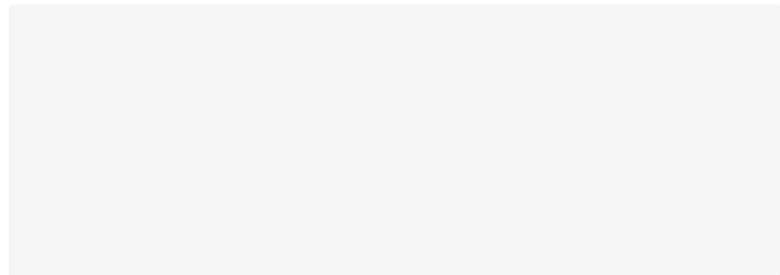
Type your password again



Confirm

(graph 3-4)

Have questions? Report here

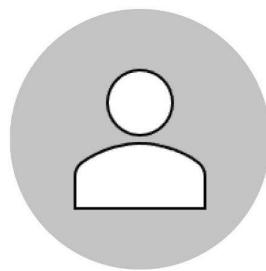


Report

(graph 3-5)

Student

Dashboard



Username

Old Password

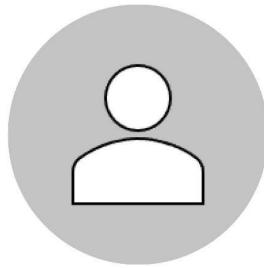


New Password



Confirm

(graph 3-6-1)



Username

Old Password



New Password



Confirm

(graph 3-6-2)



Discussion

Qusetion1

Resources

This is a question!

May 5, 2022

POST

(graph 3-6-1)

Student **Dashboard**

Paragraph **B** *I* U \leftrightarrow \otimes $\vdash \dashv$ Σ \leftrightarrow \oplus Σ \otimes \oplus

Title here
Content here

Post

(graph 3-6-2)

Student **Dashboard**

Courses

Messages

INFO2222

COMP2017

COMP2123

INFO3333

(graph 3-7-1)

Admin

Dashboard

The Admin Dashboard features a header with 'Admin' and 'Dashboard' on the left and three icons on the right. A sidebar on the left includes 'Courses' (selected), 'Messages', and 'Manage'. Four course cards are displayed: 'INFO2222' (red), 'COMP2017' (black), 'COMP2123' (blue), and 'INFO3333' (purple).

(graph 3-7-2)

Student

Dashboard

The Student Dashboard features a header with 'Student' and 'Dashboard' on the left and three icons on the right. A sidebar on the left includes 'Discussion' (selected) and 'Resources'. A central card displays 'Qusetion1' and the text 'This is a question!'. A timestamp 'May 5, 2022' is at the bottom right.

POST

(graph 3-8-1)

Student

Dashboard



Title here

Content here

Post

(graph 3-8-2)

Admin

Dashboard



Discussion

Announcement 1



This is an announcement!

May 3, 2022

Qusetion1

This is a question!

May 5, 2022

Qusetion2

This is a question!

POST

(graph 3-9-1)

Admin **Dashboard**

Title here **Announcement**

Content here

Post

(graph 3-9-2)

Student **Dashboard**

Discussion

Lectures

Resources

Week1 Lecture **Download**

Slides

Week1 Slide **Download**

(graph 3-10-1)

Admin

Dashboard



Discussion

Resources

Lectures

Week1 Lecture

Download

Slides

Week1 Slide

Download

UPLOAD

(graph 3-10-2)

Student

Dashboard



Courses

Messages

Friends

Steve

Bob

Alex

S

Hello!

Hello, Steve!

Send

(graph 3-11-1)

Admin

Dashboard

The screenshot shows a messaging application interface. On the left, there's a sidebar with 'Courses', 'Messages' (selected), and 'Manage'. The main area has a header 'Friends' with a search icon and a user icon. Below it is a list of names: Steve, Bob, and Alex. To the right, a message window for 'Steve' is open, showing a message from the user 'Hello!' and a response 'Hello, Steve!'. There's a 'Send' button at the bottom right of the message window.

Courses

Messages

Manage

Friends

S

Hello!

Hello, Steve!

Send

(graph 3-11-2)

Admin

Dashboard

The screenshot shows a student management interface. On the left, there's a sidebar with 'Students' (selected), 'Courses', and a search bar. The main area lists ten students: Steve, Bob, Alex, James, Tony, Alice, Ela, Tom, Jerry, and Sam. Each student entry includes a 'Mute' button (blue) and a 'Delete' button (red).

Students

Courses

Search...

Steve	Mute	Delete
Bob	Mute	Delete
Alex	Mute	Delete
James	Mute	Delete
Tony	Mute	Delete
Alice	Mute	Delete
Ela	Mute	Delete
Tom	Mute	Delete
Jerry	Mute	Delete
Sam	Mute	Delete

(graph 3-12)

Admin		Dashboard				
Students	INFO2222			Delete		
Courses	COMP2017			Delete		
	COMP2123			Delete		
	INFO3333			Delete		
	INFO1110			Delete		

(graph 3-13)

Sign in

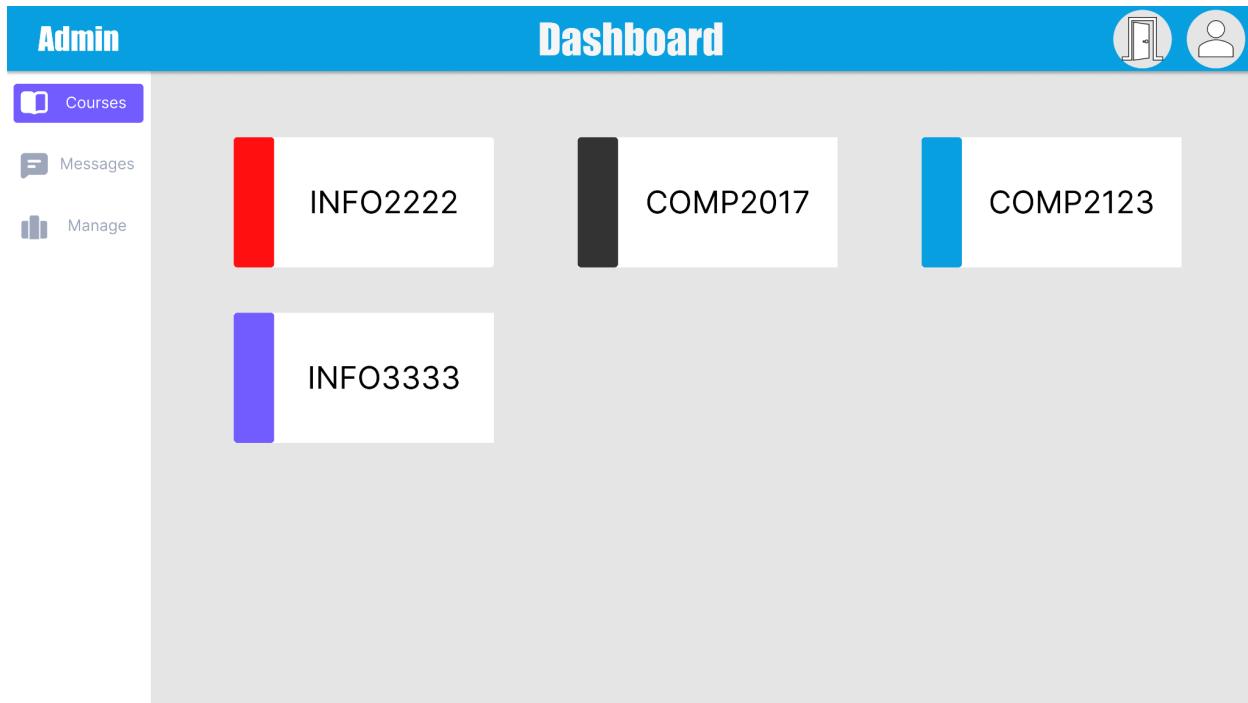
Username

Password

Don't have an account? [Create now!](#)

[Sign In](#)

(graph 3-iteration1)



(graph 3-iteration2)

```
<style type="text/css">
  #discussion-list{
    display: block;
    margin: 0;
    padding: 0;
    width: 300px;
    height: 50px;
    background: #EEE;
    list-style-type: none;
  }
</style>
<ul id='discussion-list'>
  <li>${discussionlist}</li>
</ul>
<form action='/discussion' method='post'>
  You are: <input name="poster" type="text" />
<br>
  The content is: <input id="discussion" name="discussion" type="text" />
<br>
  <input value="post" type="submit">
</form>
```

(graph 4-1)

```
def discussion_form():
    discussionDB = sql.SQLDatabase('database.db')
    result = discussionDB.get_discussion()
    return page_view("discussion", discussionlist=result)

# -----
```



```
def discussion_post(poster, contents):
    userDB = sql.SQLDatabase("database.db")
    if userDB.check_mute(poster) is True:
        discussionDB = sql.SQLDatabase("database.db")
        discussionDB.post_discussion(poster, contents)
    else:
        return page_view("error")
```

(graph 4-2)

```
@get('/discussion')
def get_discussion():
    return model.discussion_form()

# -----
```



```
@post('/discussion')
def post_discussion():
    poster = request.forms.get('poster')
    contents = request.forms.get('discussion')
    return model.discussion_post(poster, contents)
```

(graph 4-3)

```
# Get Discussion
def get_discussion(self):
    discussion = []
    sql_query = """
        SELECT poster, contents
        FROM Discussion
    """
    sql_query = sql_query.format()
    results = self.execute(sql_query)
    self.commit()
    for row in results:
        x = (row[0], row[1])
        discussion.append(x)
    return discussion

# -----
# Post Discussion
def post_discussion(self, poster, contents):
    sql_query = """
        INSERT INTO Discussion
        VALUES ('{poster}', '{contents}')
    """
    sql_query = sql_query.format(poster=poster, contents=contents)
    self.execute(sql_query)
    self.commit()
    return True
```

(graph 4-4)

```
<style type="text/css">
  #users{
    display: block;
    margin: 0;
    padding: 0;
    width: 300px;
    height: 50px;
    background: #EEE;
    list-style-type: none;
  }
</style>

<ul id='users'>
  <li>${users}</li>
</ul>
<form action='/manage' method='post'>
  Username: <input name="user" type="text" />
<br>
  <input type="radio" name="m" value="mute">mute
  <input type="radio" name="m" value="unmute">unmute
<br>
  <input type="radio" name="m" value="delete">delete
<br>
  <input value="confirm" type="submit">
</form>
```

(graph 4-5)

```
@post('/manage')
def post_manage():
    m = request.forms.get('m')
    user = request.forms.get('user')
    if m == 'mute':
        return model.mute_user(user)
    elif m == 'unmute':
        return model.unmute_user(user)
    elif m == 'delete':
        return model.delete_user(user)

# -----
# Help with debugging

@post('/debug/<cmd:path>')
def post_debug(cmd):
    return model.debug(cmd)
```

(graph 4-6)

```
def users():
    usersDB = sql.SQLDatabase('database.db')
    result = usersDB.get_users()
    return page_view("manage", users=result)

# -----


def mute_user(user):
    usersDB = sql.SQLDatabase('database.db')
    usersDB.mute_user(user)

# -----


def unmute_user(user):
    usersDB = sql.SQLDatabase('database.db')
    usersDB.unmute_user(user)

# -----


def delete_user(user):
    usersDB = sql.SQLDatabase('database.db')
    usersDB.delete_user(user)
```

(graph 4-7)

```

# -----
# Get users
def get_users(self):
    users = []
    sql_query = """
        SELECT username
        FROM Users
    """

    results = self.execute(sql_query)
    self.commit()
    for row in results:
        users.append(row)

    return users

# -----
# mute user
def mute_user(self, username):
    sql_query = """
        UPDATE Users SET mute = 1 WHERE username = '{username}'
    """
    sql_query = sql_query.format(username=username)
    self.execute(sql_query)
    self.commit()
    return True

# -----
# unmute user
def unmute_user(self, username):
    sql_query = """
        UPDATE Users SET mute = 0 WHERE username = '{username}'
    """
    sql_query = sql_query.format(username=username)
    self.execute(sql_query)
    self.commit()
    return True

# -----
# delete user
def delete_user(self, username):
    sql_query = """
        DELETE FROM Users WHERE username = '{username}'
    """
    sql_query = sql_query.format(username=username)
    self.execute(sql_query)
    self.commit()
    return True

# -----
# check mute
def check_mute(self, username):
    sql_query = """
        SELECT mute from users WHERE username = '{username}'
    """
    sql_query = sql_query.format(username=username)
    result = self.execute(sql_query)
    self.commit()
    x = ''
    for row in result:
        x = row[0]

    if x == '1':
        return False
    else:
        return True

```

(graph 4-8)

Reference

HUSSAIN AHMED. (n.d.). *Education dashboard design*. Figma.
<https://www.figma.com/community/file/1099980142157479414>