

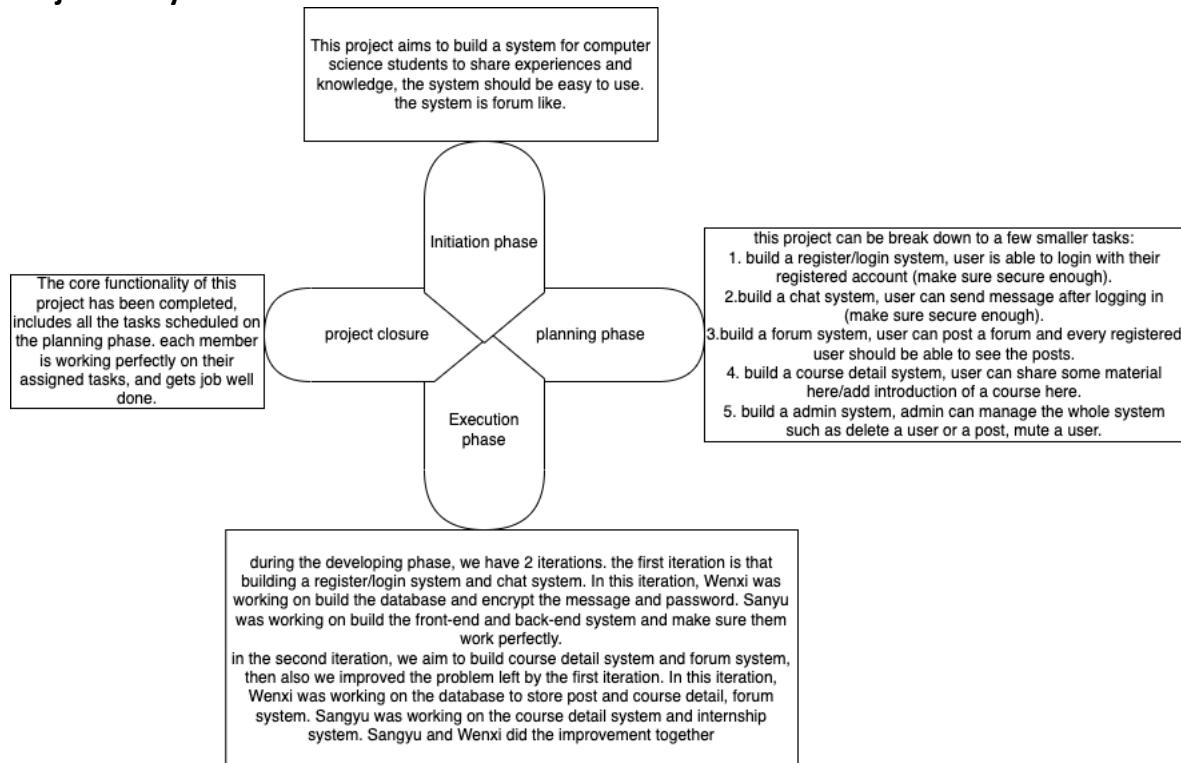
Report

Group name:RE03_Team5

Name: Wenxi Zheng(Edward), Sangyu Shi(Frank)

Summary

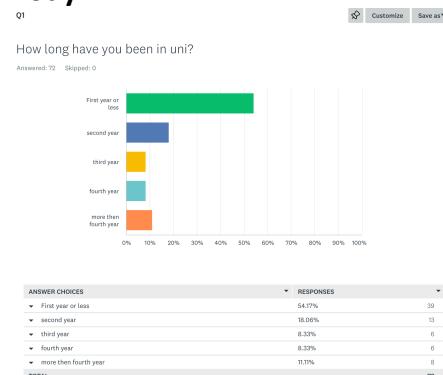
Project lifecycle:



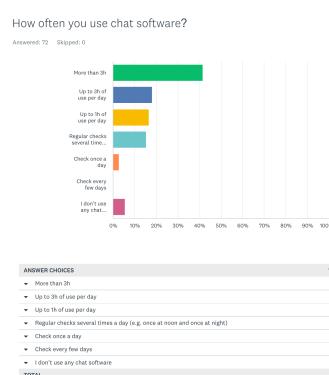
Contribution:

At each LOFI design iteration, Sangyu did the lofi design and Wenxi did the Guerrilla test. For the HIFI design (actual code), Sangyu build the front-end and back-end system and all the page layout design, Wenxi built the database and did the encryption in the first iteration. In the second iteration, Sangyu did the course detail and internship system design and Wenxi did built the additional database table and forum system, we did the improvements in this iteration together.

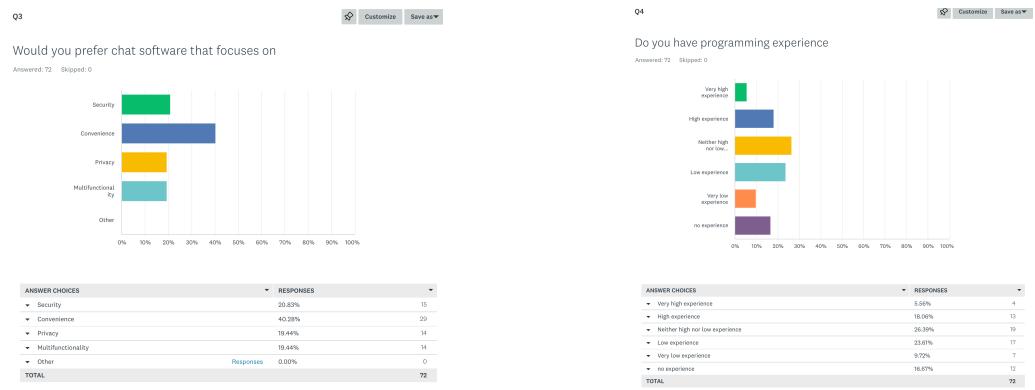
Body



Histogram (1) How long have the respondent been in uni.



Histogram (2) How often does the respondent use chat window



Histogram (3) what's the most important topic for respondent Histogram (4) Does the respondent have any programming experiences



Histogram (5) which group of students should our application focuses on

From histogram (1), we could know 54% of respondents are in year 1. From histogram (5), we could know 40% of the respondents prefer to build the application for computer science students. From histogram (4) and (5), they show most respondents (26%) have neither high nor low experience on programming and most respondent (41%) would like to spend more than 3 hours on our application. 40% of the respondent prefer convivence than other features. Thus, the target user is computer science students who are in year 1, the application should be easy to use since they are new to the system and they prefer convivence more and they do not have too much experience of programming.

Research

Themes	Number of participants	%
Unfamiliarity with Weebly	37	35.2
Slow Internet connection	20	19.04
Lack of time	14	13.3
Lack of confidence	12	11.4
Lack of mutual engagement	8	7.6
Lack of interesting topics	7	6.6
No notifications sent by Weebly	4	3.8
Too many posts	3	2.9
Total	105	100

Table 1: factors hindering students from participating in the forum (cite:

[https://www.researchgate.net/publication/311669156 To Participate or Not Identifying the Factors Affecting University Students 'Participation in an e-Forum](https://www.researchgate.net/publication/311669156>To+Participate+or+Not+Identifying+the+Factors+Affecting+University+Students+'Participation+in+an+e-Forum)

From table 1, the main reason of students not participating in the forum is unfamiliarity with the forum. Furthermore, about 20% of students participate in the forum for enhancing knowledge and related skills. Thus, we can assume the most important feature a forum/knowledge share like application should be easy to get familiar and easy to seek for help/post helpful answers.

Target user: first year computer science students

Persona:



Voski Baghdasaryan

Age: 19

Status: undergraduate

Customer profile: A first year undergraduate student who is always curious to the course he is taken and willing to help other students as he can. He wants a freshman-friendly system such that he can discuss and solve problem easily.

Motivations:

1. Easy to get familiar
2. Can help others with their question and get supported for himself
3. Get respond/support in a short time

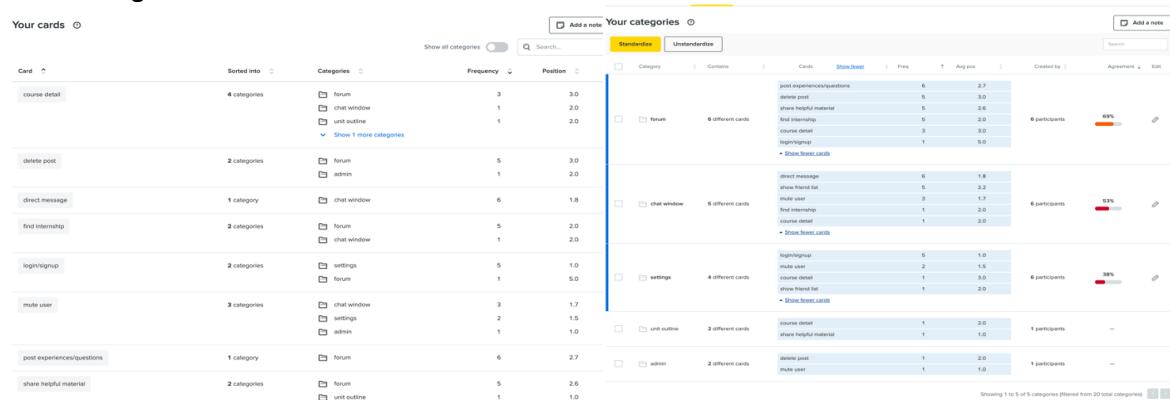
Frustrations:

1. Hard to find a solution on the internet for questions of a specific course.
2. Takes a lot of time to get responds/answer in non-uni forum (StackOverflow)
3. Need to sign up a new account for forum and hard to get familiar with the forum

Goals:

1. He wants to have a guide when he don't know how to achieve a function
2. He wants to get responds as quickly as possible
3. He wants to login with student account directly.

Card sorting

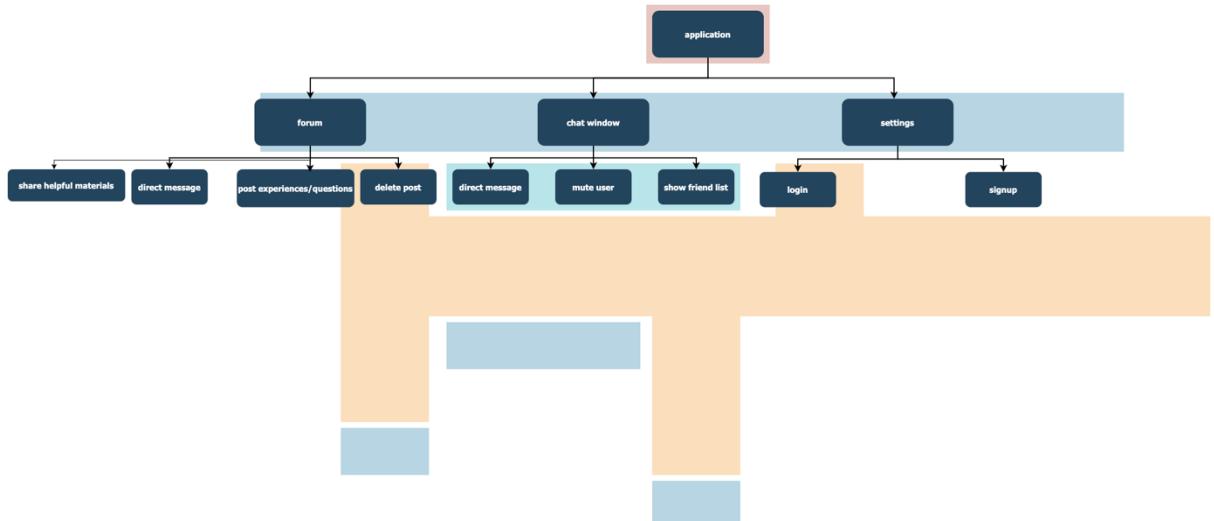


See use case in appendix

According to the result of the card sorting, forum should have course detail, delete post, find internship, post experiences/questions, share helpful materials. Chat window should be able to send direct message, mute user and show friend list. Login and signup should be included by settings.

Information architecture

1. sitemap:



Prioritized list of additional features

- 1.signup/login
2. Show friend list
- 3.direct message
- 4.post experiences/questions
- 5.share helpful materials
- 6.course detail
7. Delete post
8. Mute user
9. Find internship

Guerrilla test

To achieve the ‘best design’, we keep designing our prototypes and doing the guerrilla test with our potential users. We listen to our potential users and try to improve the design flaws.

In the first iteration, we designed two tasks for the participants. The first one is that try to play with the sign up and login functionalities. The second one is that try to send a message to their friends and check receive functionality. Most of the users feel uncomfortable with the way we show our message page, they reflect that the words/information in the message page is hard to see clearly due to the background. Some of the participants also talked about the way they interaction with sending message, they find it inconvenient to send messages by inputting the receiver. Thus, in this iteration, we will focus on making words showed clearly and improving the send message interaction.

1. User evaluation:

- User will try and achieve to login and send message by typing the receiver they expect
- User will associate send message with typing the name they expect to receive
- User will notice the correct action is available to them
- User finds it inconvenient to send message by typing the receiver’s name

In the second iteration, we designed two tasks for the participants as well. The first one is that try to send message by clicking the name listed in the friend list. The second one is that play with the forum page. Most of the users find the improved message page is quite interactive, it’s more observe that send a message by clicking instead of inputting the receiver. For the forum page, most of the participants find it’s easy to get started. One of the participants still find it a little bit hard to know the functionalities as a beginner. Thus, we will need a guideline to help the users to learn the usage of our application.

2. User evaluation:

- It might be a little bit hard to make user find out send message by clicking the name
- User might associate click name with send message but still not obvious
- User can see the progress by clicking the thread listed on the left

Hi-Fi prototype (actual code)

In the first iteration, we were focusing on making friends listed in friend list clickable and improve the way user send messages to their friends. Also, we made the forum functionality come true. Then we changed the layouts of pages a bit to make user see the information clearly.

3. user evaluation:

- user will try and achieve sending message by clicking the name listed in the friend list

- user will try and achieve post a thread by clicking the button and see details of threads by clicking the small thread card listed on the left
- user will notice that that send message can be triggered by clicking the friend's name
- user will notice that post threads can be triggered by clicking the button
- it might be hard for users to find that click the threads listed on the left can see more details
- user will associate click name with sending message
- user will associate click new thread button with posting a thread
- it might be hard for users to associate click the threads listed on the left with showing more details

In the second iteration, we planned to give some hints of the functionalities that is mentioned above to users to help them get familiar with the system. Then we were focusing on implementing a course detail page and career hub page.

4. User evaluation:

- User will try and achieve see the details of courses and see the available job in the career hub
- User will try and achieve see the usage of the web application when they are unsure with some operations
- User will notice that click the button in the career hub page to see more details of a job
- User will notice that every single functionalities need to login first (forum, course detail and career hub)
- User will associate see more details of a course with clicking the button
- User will associate forum, course detail and career hub with logging in

Demonstration of core functionalities:

See image in appendix

5. Forum:

- When user click forum in the header, it will show forum page image (1). User can choose to post a new thread by clicking the highlighted button 'New Thread', when the button is clicked, there will be an input box pop up, type the content in the input box and click 'click to share'. When there's post being shared, every registered user is able to see the post. Click the thread listed on the left to see the detail (forum page image 4).

6. Course detail:

- When user click CourseDetail in the header, it will show course page image (1). There are some pre-listed course.

7. Admin role:

- When admin choose to mute user, admin can type the user he expect to be muted in the input box. Then the muted user will not be able to share anything in the forum page as admin page image (2).
- When admin choose to delete user, admin can type the user he expect to be deleted in the input box. Then the user will be not exist in the system. If the deleted user try to log in, it will show as admin page image (3)
- When admin choose to delete a course detail, admin can type the course code in the input box. The deleted course will not be shown in the course detail page anymore. The given example (admin page 4) deletes INFO2222.

Every functionality mentioned above are based on login, application will jump to relevant page only if user are logged in.

Bonus functionality:

See image in appendix

Career hub:

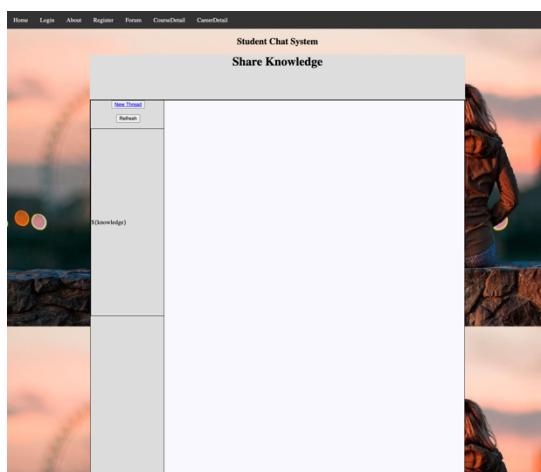
When user click CareerHub in the header, it will show career hub page image (2). There are some pre-listed jobs. When user choose to see details of a job (click the 'see detail' button), it will jump to the detail page.

When admin choose to delete a job in career hub, login as an admin, type the id of the job in the input box, for example the id of first job is 1. The deleted job will not be shown in the career hub page anymore as admin page image (5), the given example deletes Software Engineering job.

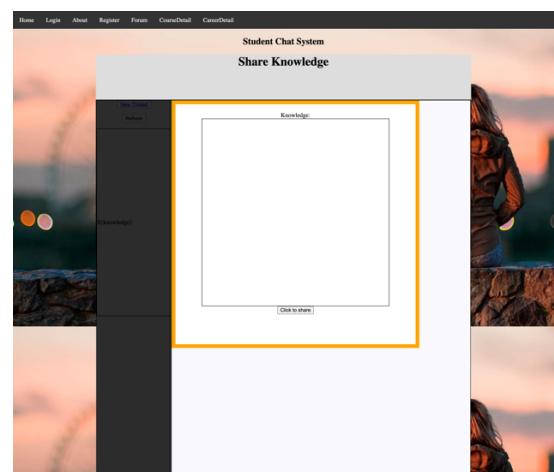
When admin choose to add a job in career hub, login as an admin, type the job name and job description in the input boxes. The job will be added and shown in the career hub page. The given example is shown in the admin page image (6).

Appendix

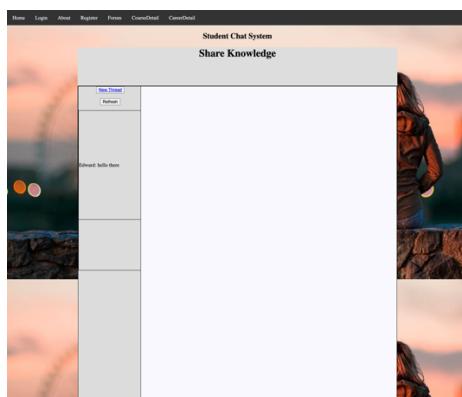
8. Forum functionality



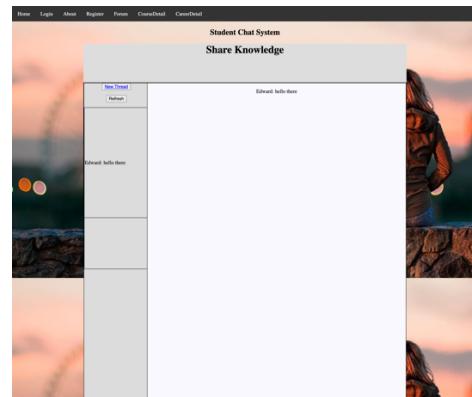
Forum page image (1)



Forum page image (2)

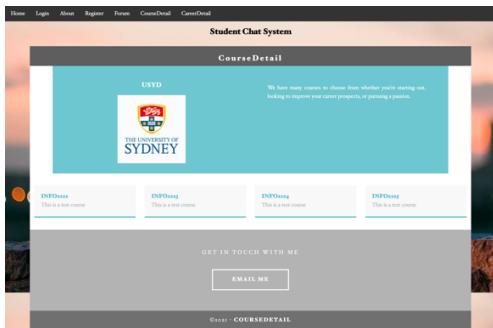


Forum page image (3)



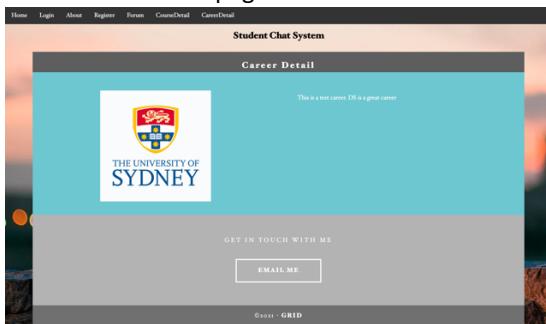
Forum page image (4)

9. Course detail page

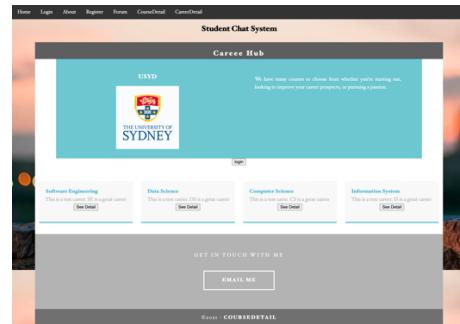


Course page image (1)

10. Career hub page



Career hub page image (1)

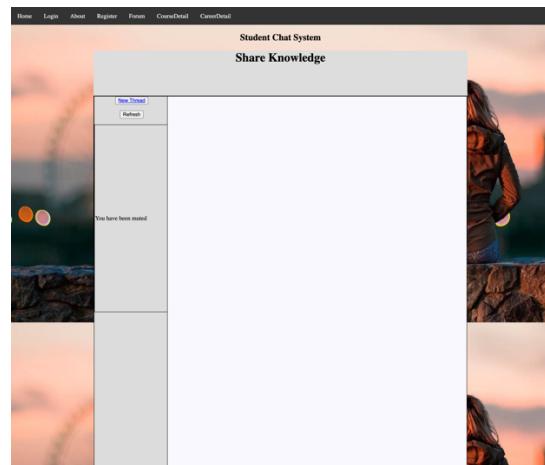


Career hub page image (2)

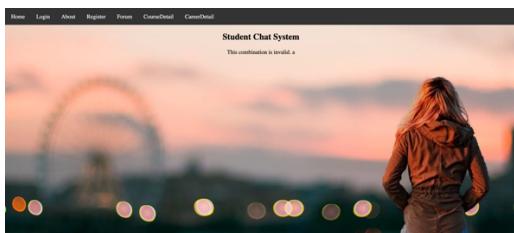
11. Admin role



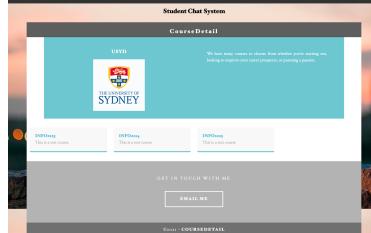
Admin page image (1)



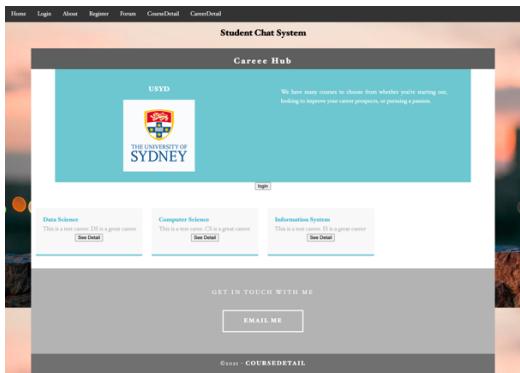
Admin page image (2): mute user result



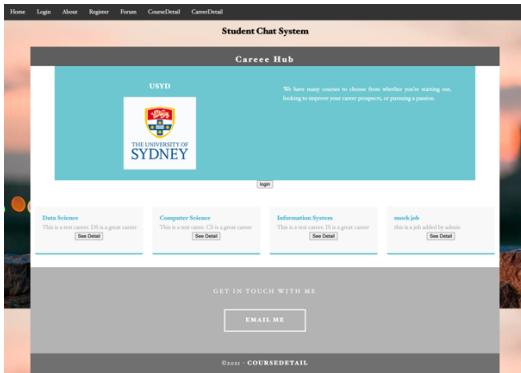
Admin page image (3): delete user result



Admin page image (4): delete INFO2222



Admin page image (5): result of deleting a job

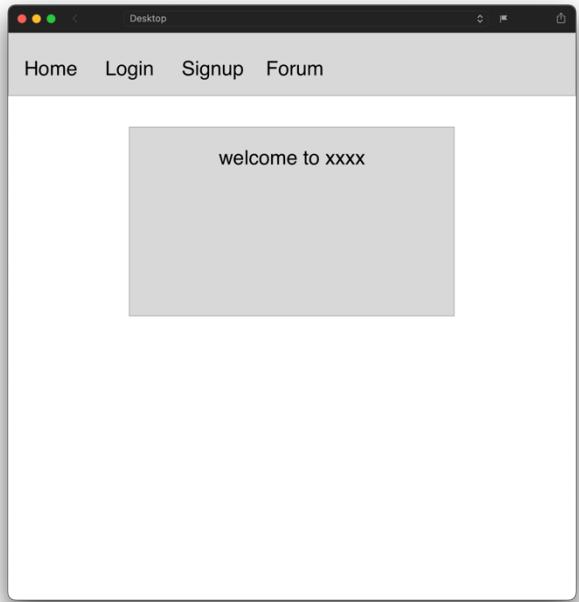


Admin page image (6): result of adding a job by admin

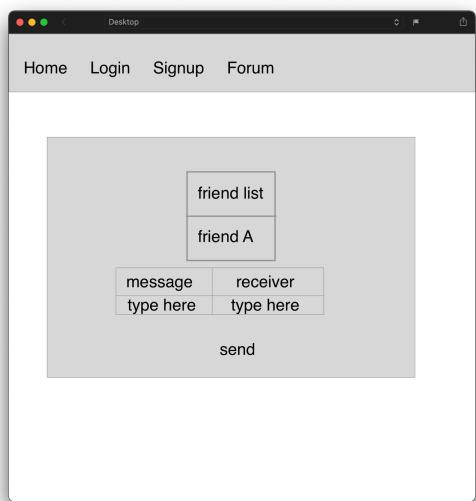
Use case of card sorting:

- **Signup:**
 - Primary actor: user/admin
 - Description: this use case descripts how a new user create his/her own account
 - Triggers: user wants to prepare for logging in
- **Login:**
 - Primary actor: user/admin
 - Description: this use case descripts how user login with his/her own account
 - Triggers: user wants to login to the system for further functionality
- **Post experiences/questions:**
 - Primary actor: user/admin
 - Description: this use case descripts how a user post a new thread in the forum page
 - Triggers: user wants to post a thread
- **Share helpful materials:**
 - Primary actor: user/admin
 - Description: this use case descripts how a user share materials in the course detail page
 - Triggers: user wants to share materials
- **See course detail:**
 - Primary actor: user/admin
 - Description: this use case descripts how a user see the detail of a course
 - Triggers: user wants to see course details
- **Direct message:**
 - Primary actor: user/admin
 - Description: this use case descripts how a user send messages to his/her friends
 - Triggers: user wants to send messages to his/her friends
- **Delete post:**
 - Primary actor: admin
 - Description: this use case descripts how admin delete a post
 - Triggers: admin wants to delete a post
- **Mute user:**
 - Primary actor: admin
 - Descriptions: this use case descripts how admin mute a user
 - Triggers: admin wants to mute a user
- **Find internship:**
 - Primary actor: user/admin
 - Description: this use case descripts how user finds internship
 - Triggers: user wants to find a job

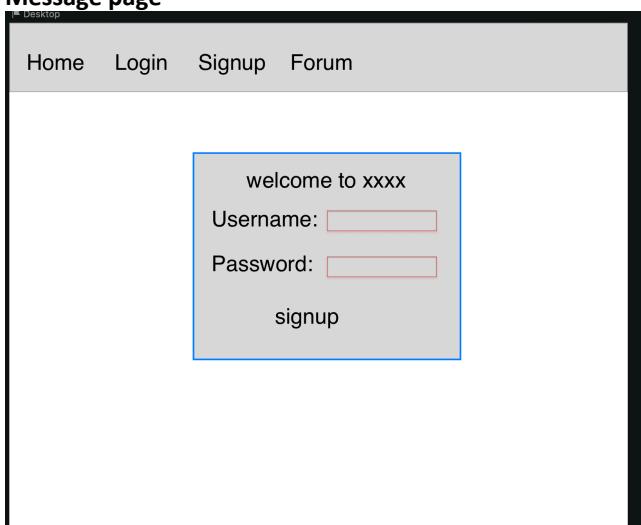
Original lofi prototype:



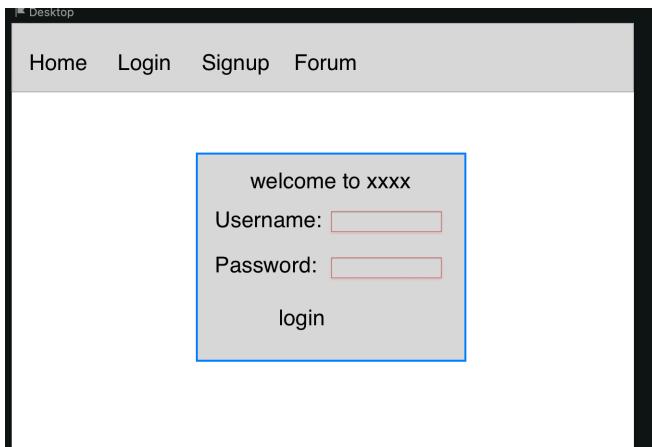
Home page



Message page

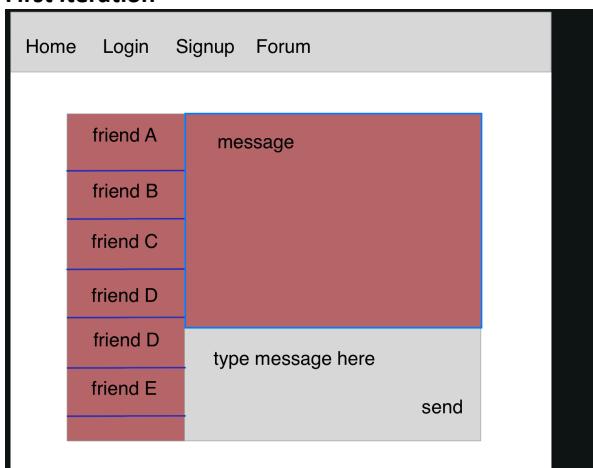


Signup page

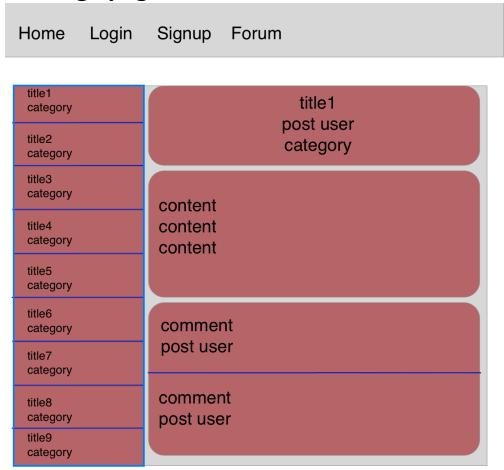


Login page

First iteration

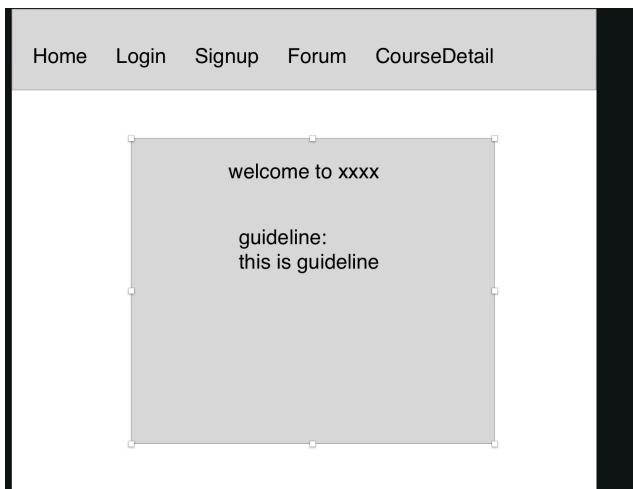


Message page

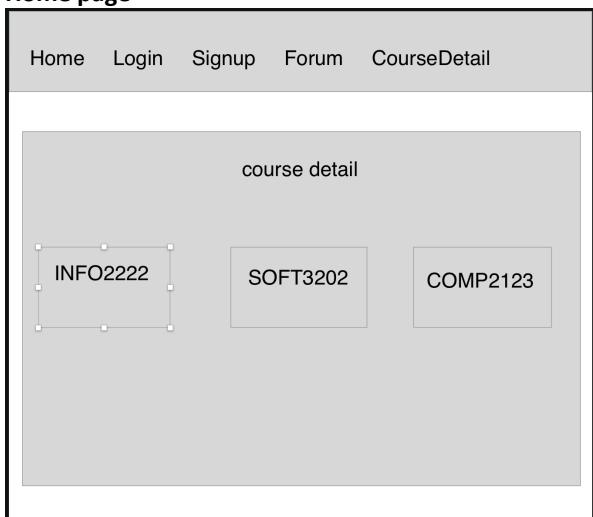


Forum page

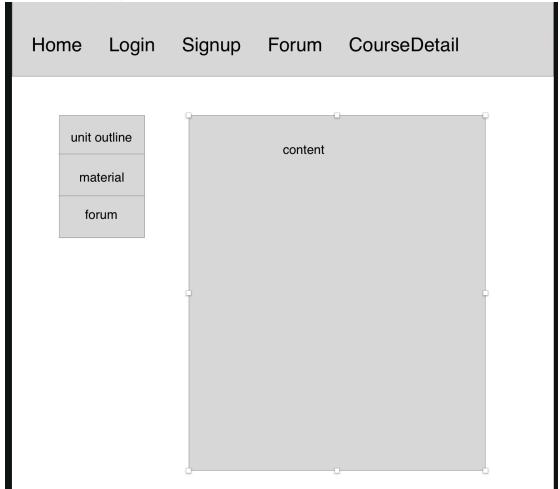
Second iteration prototype:



Home page



Course page



Course detail page