San Francisco State University

SW Engineering CSC 648/848 Section 1

StudentConnect

Team 5

Team Lead: Kao Saephan, Scrum Master: Sean Ibarra Diaz, Back End Lead: Luke Thilgen, Front End Lead: Collins Gichohi, Git-Master: Zacharia Angha, Front Assist: Salvador Avila.

Milestone 2

March 18th, 2024

History Table:

Milestone 1	3/4/2024
Milestone 2	3/18/2024

Task 1: Data Definitions, V2

1. **User Account:** Each user will have a user profile that will contain their personal information and preferences. These accounts will be used by both the students and the faculty to gain access to the platform features and interact with them. There will be a karma point system where users can upvote or downvote posts they think are good/bad. The total number of upvotes that a user gets, will be tracked in their account info. Each account will be created as a "teacher role" or a "student role." The teacher role will have the power to create courses, make announcements, post questions, post comments, join classroom chat, join private chat, react to posts and delete posts. Meanwhile the student role only has the ability to post questions, post comments, join classroom chat, join private chat, and react to posts.

2. **Course:** Teachers will create courses that they teach. Once created the course will hold many students. The course will have class chatter and Q&A forums.

3. **Q&A Forum:** There will be an area in the course where the users will be able to post questions and answers regarding academics and reach out for support. Both questions and answers will be classified as posts. This data is meant to only hold information regarding the question asked.

4. **Class Chatter:** A channel within the course for users to post class notes and updates, share resources, and ask each other for help in their courses. This data is meant to hold all the class-wide discussion, about any area of the course.

5. **Insightful Reactions:** Users will be able to react to a post in one of three ways: "Answer", "Bad Information", or "Off-Topic." Each user can only react once per post.

6. **Messaging System:** A private channel where users will be able to send and receive messages from those within the educational community.

7. **Security Protocol:** The sequence of operations that will ensure privacy and the safety of personal information.

Task 2: Functional requirements, V2

Priority 1: (Must have)

1. Users should be able to sign up as a specific role.
2. Users should be able to log in.
3. Users should be a teacher or a student.
4. Users should be able to post in class chatter.
5. Users should be able to post questions.
6. Users should be able to post answers to questions.
7. Teachers should be able to make courses.
8. Courses should be able to hold many students.
9. Courses should have class chatter.
10. Courses should have Q&A forums.
11. Students should be able to join many courses.

Priority 2: (Desired)

12. Users should be able to gain karma points from upvotes to their posts.
13. Users should be able respond to posts with insightful reactions.
14. Users should be able to set their profile including name, pronouns, description, and profile picture.
15. Users should be able to switch to different courses they are in.
16. Users should be able to start private chats.
17. Users should be able to message in private chats.
18. Teachers should be able to send announcements to many students.

Priority 3: (Opportunistic)

19. Users should be able to customize their user interface.

20. Teachers should be able to promote a student to teacher's assistant.

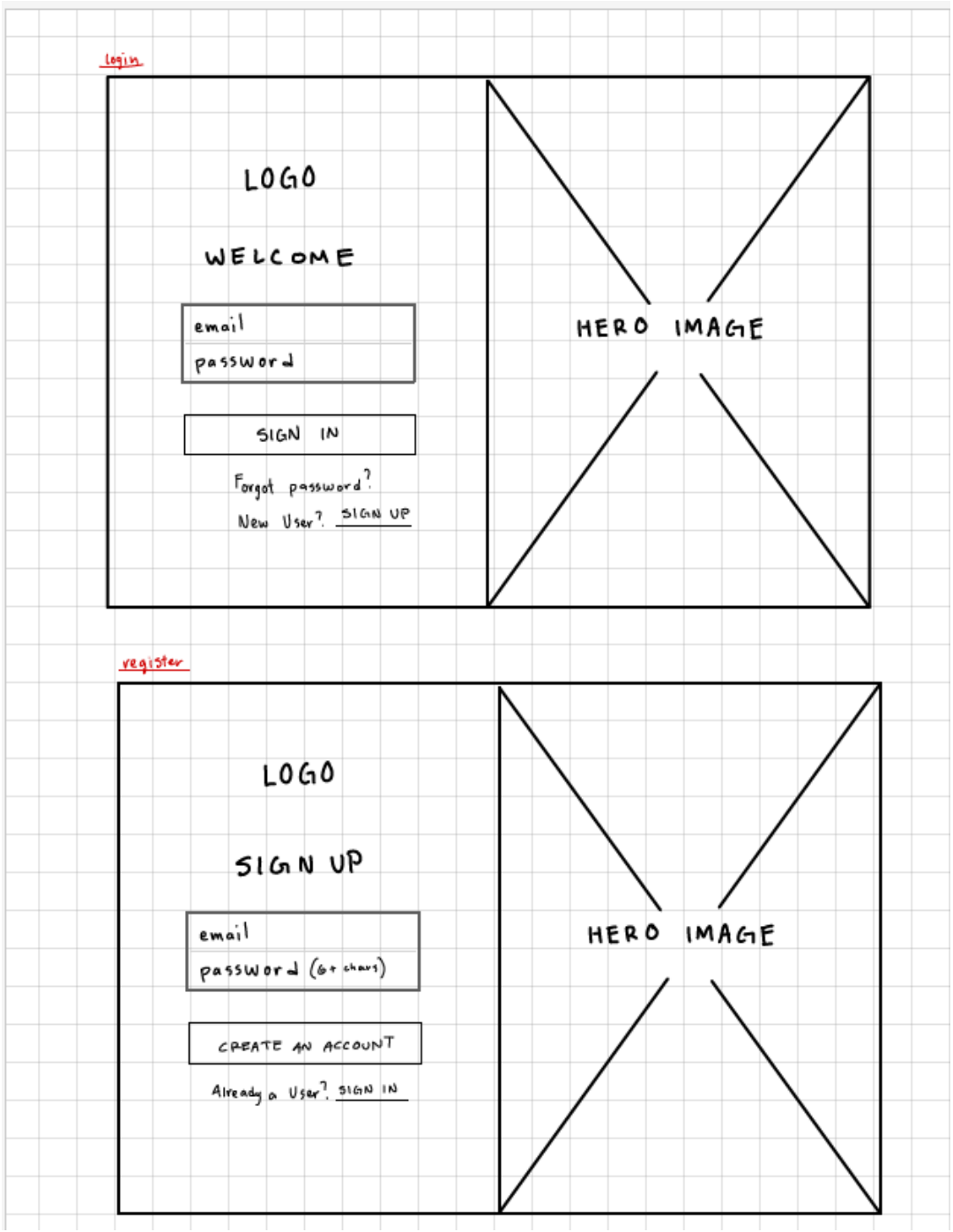
21. Teacher's Assistants should be able to make announcements.

22. Teacher's Assistants should be able to delete posts.

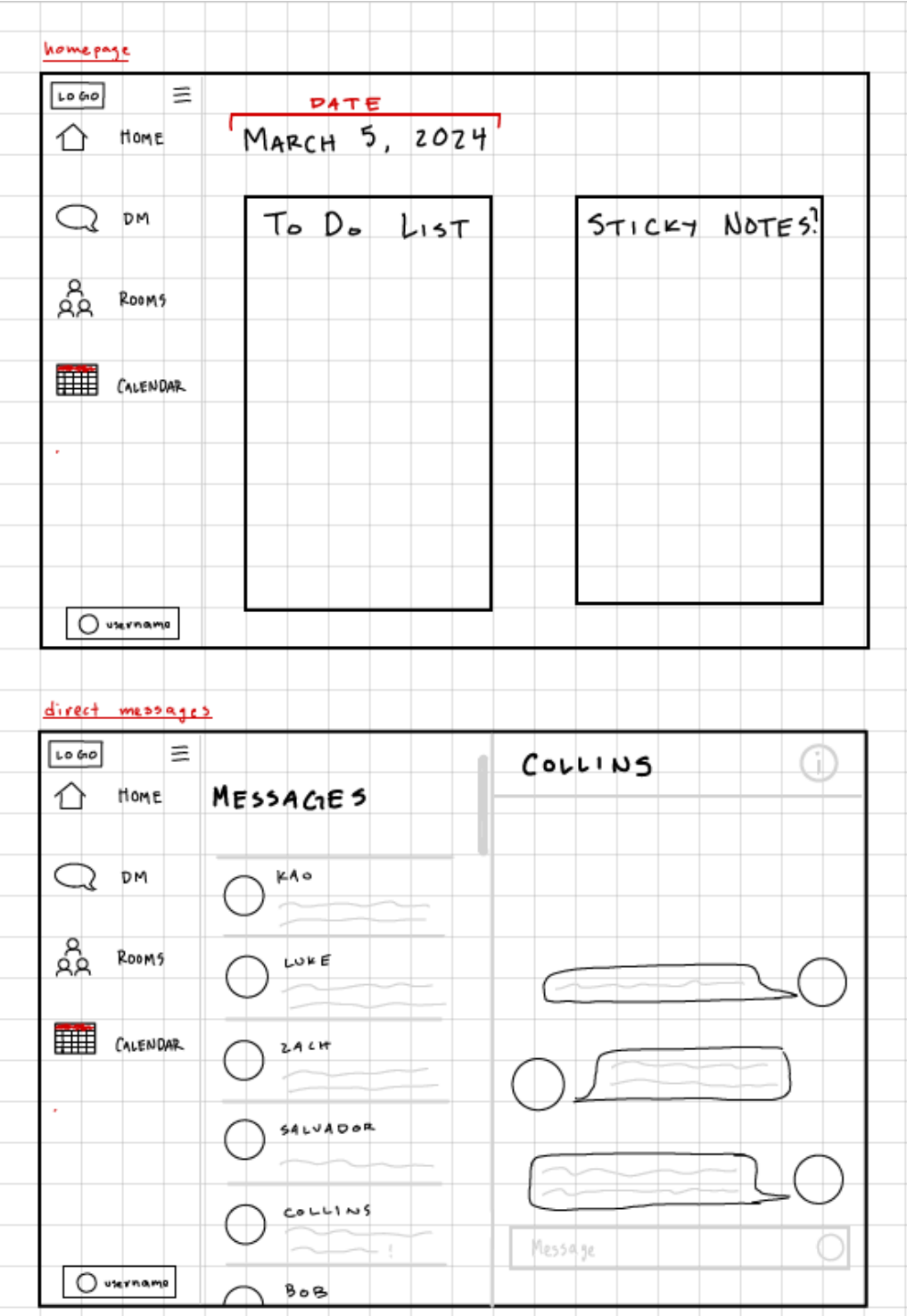
3. UI Mockups and UX Flows

UI Mockups:

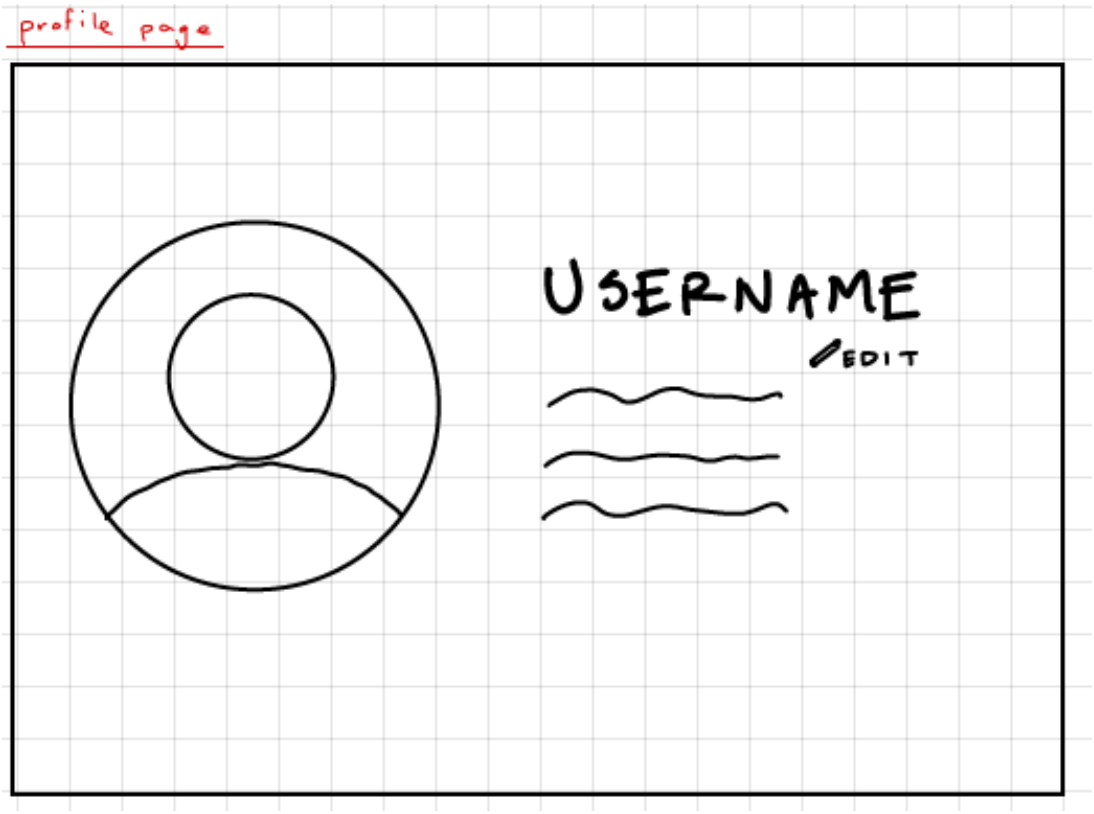
- Login and Register Mockup



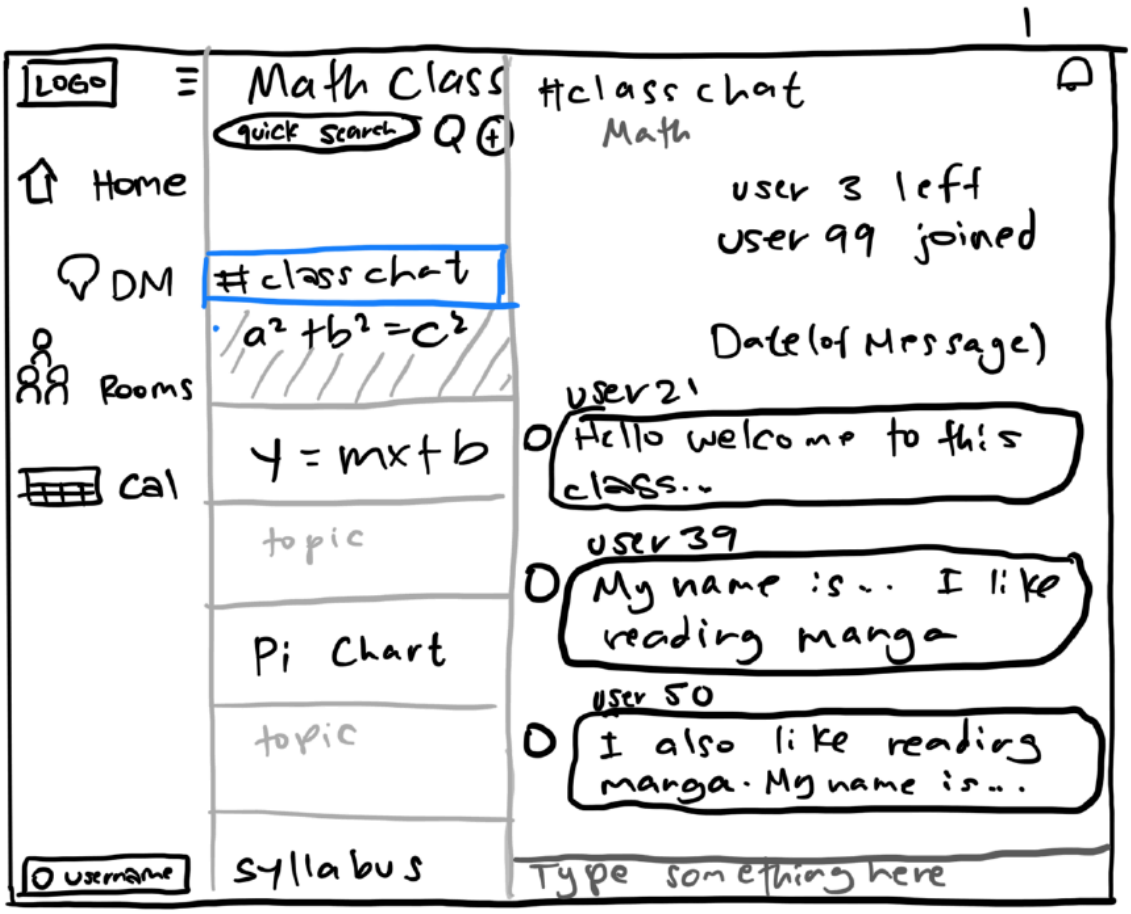
- Home Page and Direct Messages Mockup



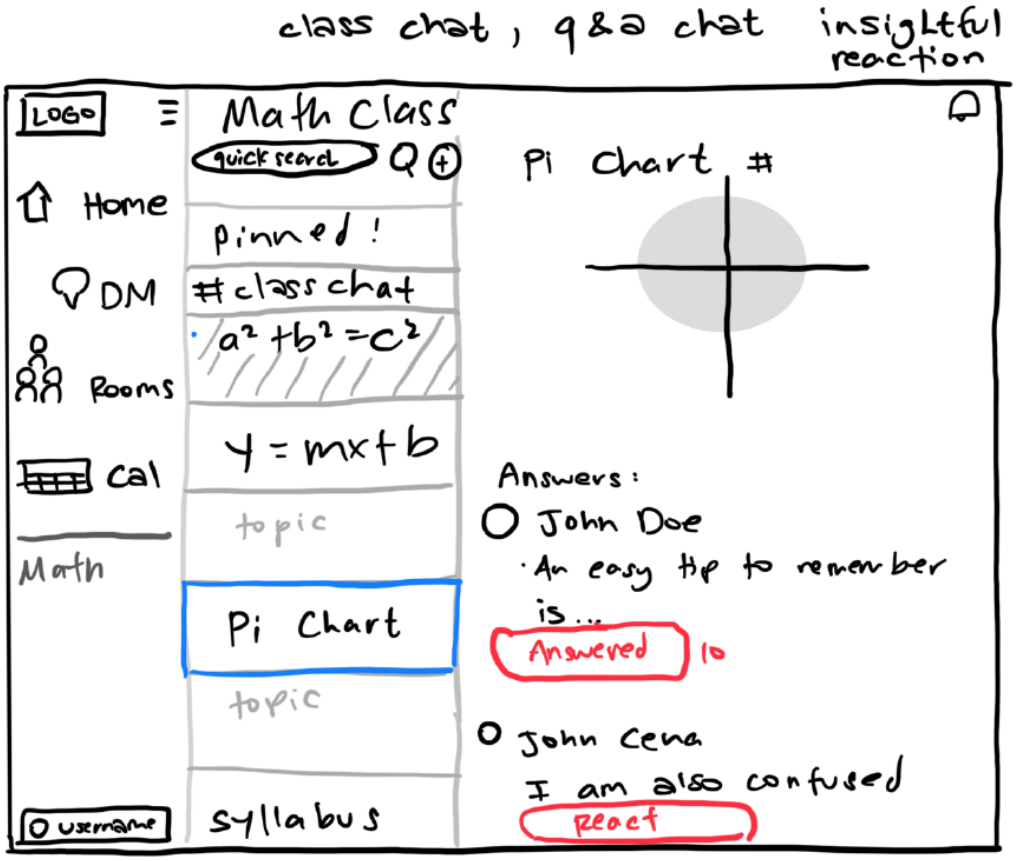
- Profile Mockup



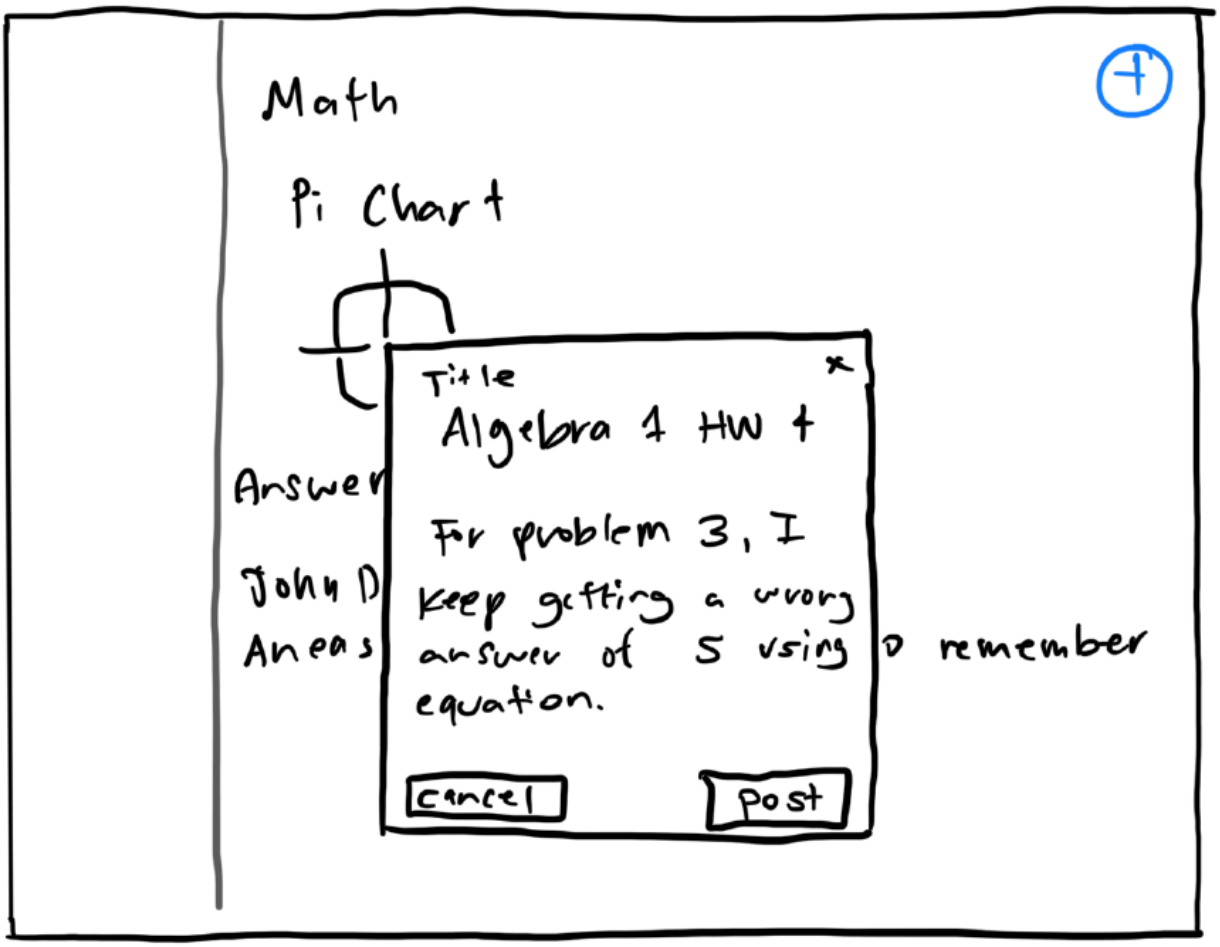
- Class Chat Mockup



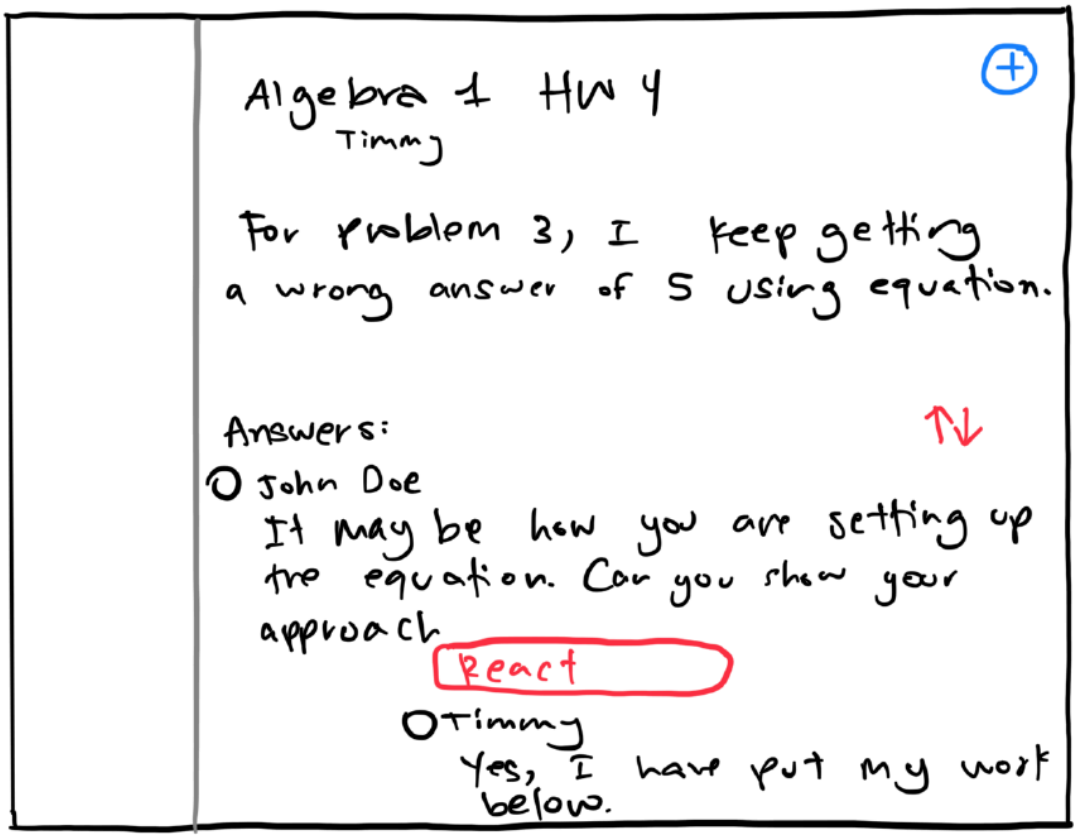
- Q&A Mockup

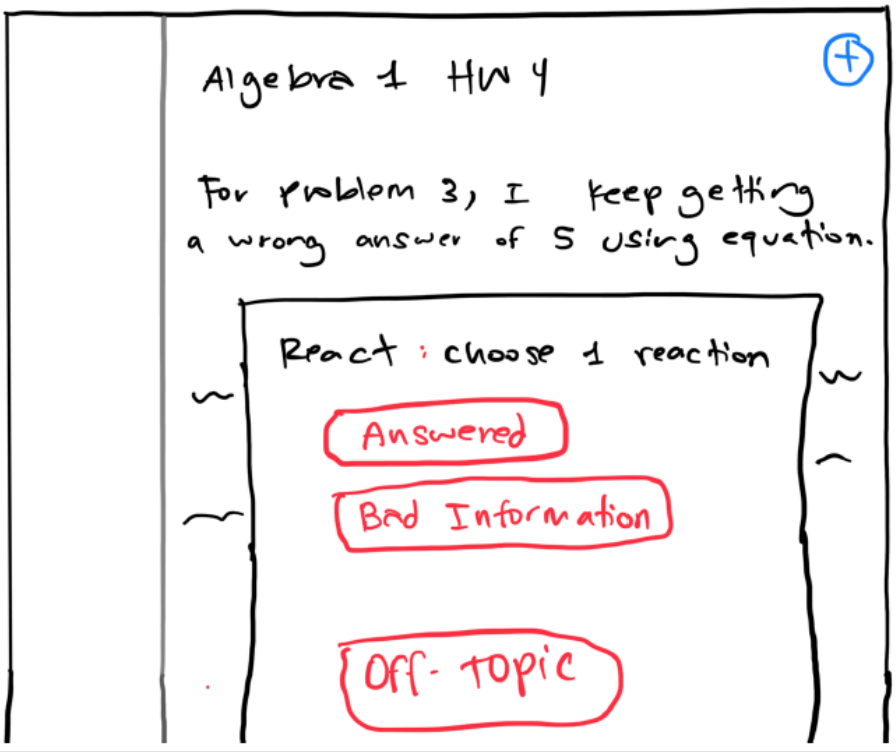

class chat, q&a chat
insightful reaction
LOGO
Math Class
quick search
Q+
Home
DM
Rooms
cal
Math
pinned!
#class chat
$a^2+b^2=c^2$
y = mx+b
topic
Pi Chart
topic
syllabus
username
Pi Chart #
Answers:
John Doe
An easy tip to remember is...
Answered
10
John Cena
I am also confused
React

- Post on Q&A Board Mockup

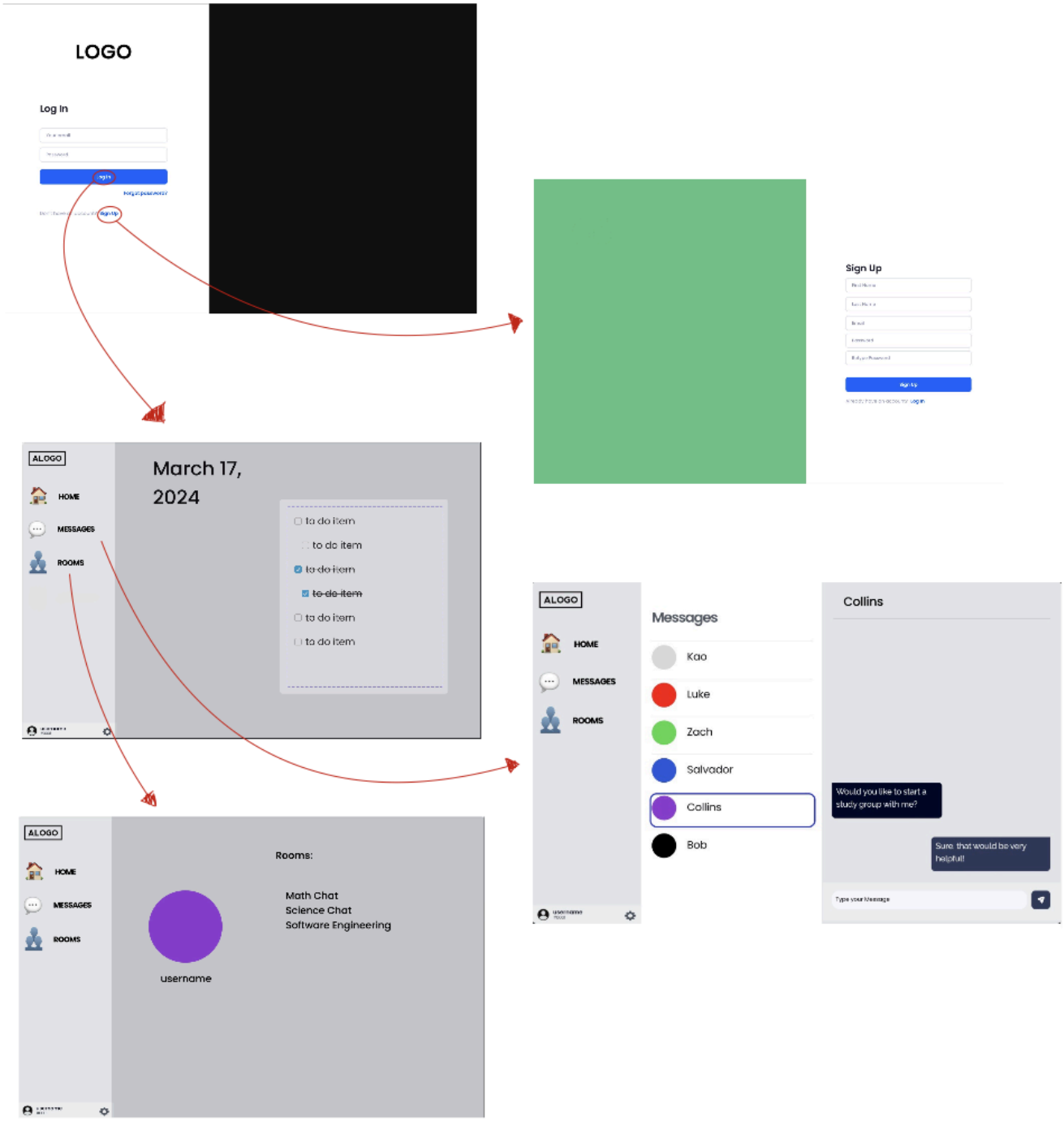


- React to Post on Q&A Board Mockup





- UX Flow: From Login to User Profile & Direct Messages



- UX Flow: From Home Page to Class Chat & Q&A Boards

4. High level Architecture, Database Organization, APIs

Schemas:

1. Users
    a. username, string - User's username
    b. userID, int - User's ID number
    c. email, string - User's email
    d. password, string - User's password
    e. firstName, string - User's first name
    f. lastName, string - User's last name
    g. karmaPoints, int - The amount of karma points a user has from interacting with other users
    h. pronouns, string - The pronouns the user has set for their profile
    i. role, string - The role the user has either student, teacher, TA
    j. dateJoined, date - The date the user joined StudentConnect
2. Posts
    a. postID, int - Post's ID number
    b. authorID, int - ID number of the user that made the post
    c. chatterID, int - ID number of the class chatter channel the post is in
    d. title, string - title of the post
    e. description, string - description of the post
    f. datePosted, date - date the post was made
3. Courses
    a. courseID, int - Course's ID number
    b. teacherID, int - ID of the teacher that made the course
    c. courseName, string - Name of the course
4. Q&A Forums
    a. forumID, int - Forum ID number.
    b. courseID, int - ID of the course the forum entry is in

   c. authorID, int - ID of the user that made the forum entry
   d. datePosted, date - Date that the forum entry was posted
   e. type, string - Specify whether it's a question or an answer
   f. message, string - The message of entry
   g. questionID, int - If it is an answer it will have the forumID of the question it's responding to
5. Class Chatter
   a. chatterID, int - class chatter ID number
   b. courseID, int - The ID of the course the class chatter is linked to
   c. channelName, string - The name of the class chatter channel
6. Messaging System
   a. senderID, int - The userID of the sender of the message
   b. receiverID, int - The userID of the receiver of the message
   c. dateSent, date - The date that the message was sent
   d. message, string - The content of the message
7. Insightful Reactions
   a. userID, int - ID of the user that made the reaction
   b. postID, int - ID of the post that the user reacted to
   c. reaction_type, string - The reaction that had, either "Answer", "Bad Information", or "Off-Topic"
   d. timestamp, date - Date the reaction was made
8. Security Protocols
   a. timestamp, date - Date the security protocol occurred
   b. description, string - Description of the security protocol
   c. eventType, string - The type of security protocol
   d. ip_address, string - The IP address of the user that performed the security protocol
   e. userID, int - The ID of the user that performed the security protocol

API Endpoints:

1. Users
    a. GET /users: Retrieve a list of all users.
    b. POST /users: Create a new user.
    c. GET /users/id: Retrieve a user by their unique ID.
    d. PUT /users/id: Update user details by ID.
    e. DELETE /users/id: Delete a user by ID.
2. Posts
    a. GET course/courseID/class_chatter/chatterID/posts/id: Retrieve a post by its unique ID.
    b. POST course/courseID/class_chatter/chatterID/posts: Create a new post.
    c. PUT course/courseID/class_chatter/chatterID/posts/id: Update a post's details by its ID.
    d. DELETE course/courseID/class_chatter/chatterID/posts/id: Delete a post by its unique ID
3. Courses
    a. GET /courses/id: Retrieve a course by its unique ID.
    b. POST /courses: Create a new course.
    c. PUT /courses/id: Update course details by ID.
    d. DELETE /course/id: Delete a course by ID.
    e. GET /course/id/forum: Retrieve a course's Q&A forums data by ID.
    f. GET /course/id/class_chatter: Retrieve a course's Class Chatters by ID.
4. Q&A Forums
    a. GET courses/courseID/forums/id: Retrieve a forum by its unique ID.
    b. POST courses/courseID/forums: Create a new forum entry.
    c. PUT courses/courseID/forums/id: Update a forum's details by its ID.
    d. DELETE courses/courseID/forums/id: Delete a forum entry by ID.
5. Class Chatter
    a. GET course/courseID/class_chatter/id: Retrieve a class chatter by ID.

   b. POST course/courseID/class_chatter: Create new class chatter.
   c. PUT course/courseID/class_chatter/id: Update class chatter by ID.
   d. DELETE course/courseID/class_chatter/id: Delete a class chatter by ID.
6. Messaging System
   a. GET /users/userID/messages: Retrieve a users messages by ID
   b. POST /users/userID/messages/receiverID: Create new messaging system entry.
   c. GET /users/userID/messages/receiverID: Retrieve a messaging chat with another user.
7. Insightful Reactions
   a. GET /posts/id/insightful_reactions: Retrieve all insightful reactions from a post.
   b. POST /posts/id/insightful_reactions: Create a new insightful reactions entry to a post.
   c. DELETE /posts/id/insightful_reactions/id: Delete an insightful reaction by ID.
8. Security Protocol
   a. GET /securityprotocol/id: Retrieve a security protocol by its unique ID.
   b. POST /securityprotocol: Create a new security protocol.

5. Identify actual key risks for your project at this time

- Skills Risks
    - Any issues with the development tools will hinder any further progress. Having one member suffering from errors reduces the amount of people working on the code.
    - We made sure that there would be little complications with the development tools and picked some that are compatible with our local machines. Everyone is holding each other accountable for their share of the work.
- Communication Risks
    - If there was a lack of communication it could result in a lack of group cohesion. There would be areas that have too many or too few people working on it, and no one would hold each other accountable.
    - To make sure that we do not break down communications we hold two meetings a week. We constantly communicate frequently on Discord to update everyone on their progress.
- AWS Risks
    - Amazon AWS has a limit to the storage the basic plan has. If the limit is reached it will bring our project to a halt.
    - We are confident that we will not exceed the limit but it is a possibility. If it does exceed we can upgrade the plan to one with more storage.
- Deadlock Risks

    - If there is a deadlock where production is in a cycle of waiting for one task to be complete so we can move on to another, it greatly reduces the speed of production. If this happens we will be drastically off schedule and not meet the milestones on time.
    - We will talk to each other to create a plan to streamline the production process to reduce the possibility of deadlocks. With the right communication and planning we will have an efficient production process with no deadlocks.
- Schedule Risks
    - Scheduling conflicts and members not showing up to meetings would be another issue which could pose a problem to the group. Not being able to join the meetings would lower the chances of communicating to the other team members.
    - One way that we could fix this issue is to reschedule the weekly meetings to a time that is more suitable for everyone. If this does not work, then send a message to those who could not attend and give them a summary of the meeting's contents.
- Teamwork Risks
    - The possibility of team members not cooperating with each other or not contributing. This would greatly hinder our progress or even remove some of the progress we have already made.

    - One way that we could avoid this is by communicating with each other and coming to a peaceful resolution. If the individual refuses to listen to reason we will report them to the teacher. As of right now there is no issues with team members and we do plan to use communication to solve any issues.

- Legal Risk
    - There is the potential risk of getting into legal trouble with any copyright infringement. If our product is too similar to any preexisting product, the rights holders could order a cease and desist order.
    - In order to avoid legal action from large corporations, we will make our product its own independent idea so the corporations can not claim we stole their intellectual property. We are also not profiting from this project, which further adds protection against facing legal action.

6. Project management

We hold two meetings a week where we go over the progress made on the project. Each member transparently presents their progress and issues to the entire team. Open discussion allows our team to reallocate resources if a section is falling behind. We use Discord as our primary means of communicating with each other and divide the tasks among us. Progress is tracked on Jira as well. As an easy to use tool, Jira allows for a quick top level view of each milestone. Each step of the milestone is broken down into parts that are then assigned to members.

Although useful, the free version of Jira has limitations. To make up for it, we have to communicate through voice chat, the responsibilities each person has and any difficulties that arise. As people, there are also challenges outside of the project. So, we require that every member checks our discord server at least once a day. Working together with open lines of communication is how our project will be successful.