

Overview

Many MIT students find it helpful to be able to work with others while doing psets or studying for tests; however, it can be difficult for students to find others to work with. To address this issue, we designed Cloop, a web application designed to allow MIT students to find classmates, build study groups, communicate with other students, and share resources in a comfortable, student-oriented forum. It features pages for each class offered at MIT, populated with all Cloop-registered users who are currently in each class. Users may make posts or comments on any of the class pages that they are associated with.

Purposes:

- 1) Finding classmates: Many students do not know how to find or reach out to other students in their classes.
- 2) Build study groups: Even if students find others in their classes, it may still be difficult for them to reach out for collaboration.
- 3) Communicate with other students: Students who may not want to form study groups or meet in person may still have questions or points of discussion for their peers.
- 4) Share resources: Students find it helpful to share resources such as notes, links, or other media.

On initial glance, Piazza and Facebook (or other social media platforms) have some aspects of what we are trying to achieve. However, neither of them provide a satisfactory solution to our problems. Piazza does not provide a comfortable place for students to interact because it is moderated by instructors and TAs; furthermore, many students feel scared to share their opinions out loud and instead post anonymously. While Facebook provides groups for general topics and purposes, it is not specifically structured for classes and education; users of Facebook are also limited by who they already know, and cannot reach out to those who are not already their friends.

Concepts

Class

- Purpose: To create an environment where MIT registered users in the same classes can interact freely
- Operational Principle: If a student joins a class, then they are added to the class page and can post or comment on the page, which they can navigate to via tabs on the user's interface

Post

- Purpose: To allow MIT registered users to share their ideas and ask questions.
- Operational Principle: If a student creates a post on a class page, then their post will be displayed for everyone in the class to see, upvote, flag, or comment on.

Resource

- Purpose: To allow MIT registered users to share resources such as notes, powerpoints, etc.
- Operational Principle: If a student clicks the upload resource button on a class page, then they can choose a file to display for everyone in the class to see, upvote, flag, or comment on.

Comment

- Purpose: To allow MIT registered users to reply or give feedback to a post.
- Operational Principle: Each post has a comment button on it that users can click. If a user clicks the comment button, then they can write a text response to the post that is viewable by everyone in the class group.

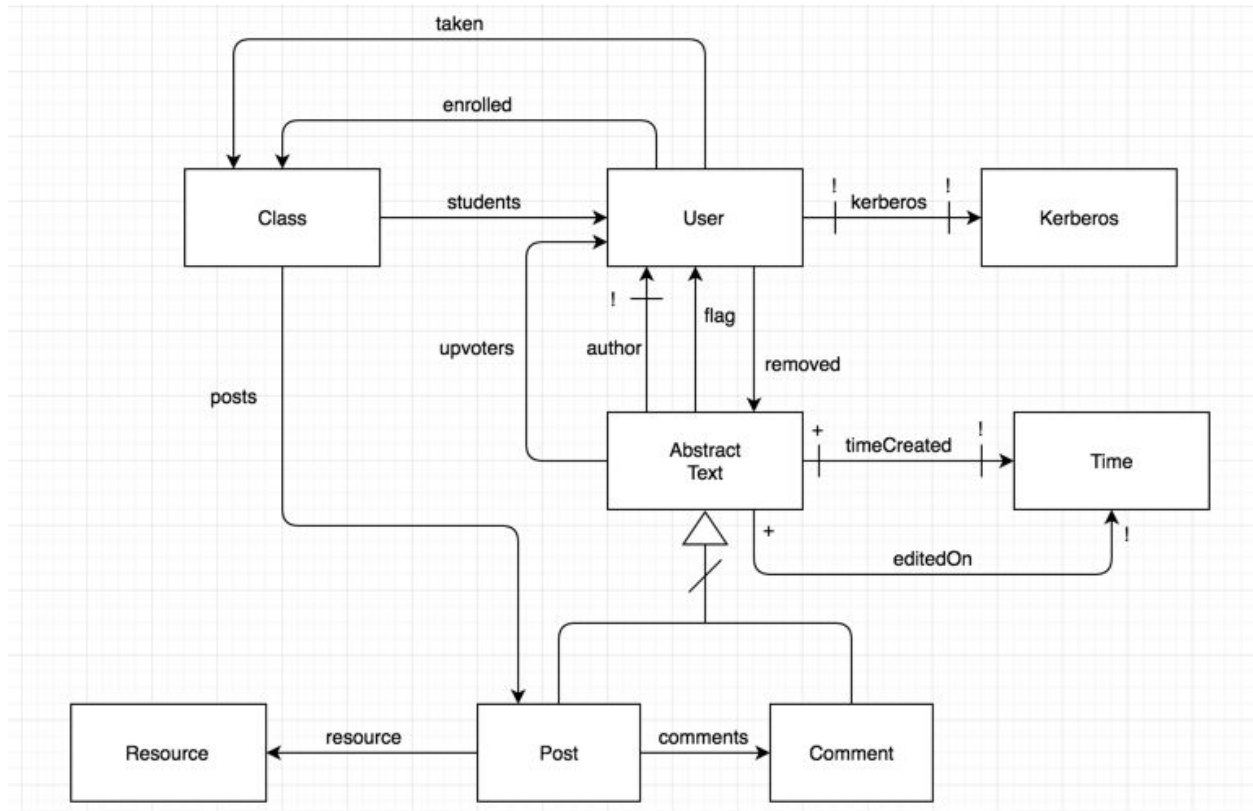
Flag

- Purpose: To allow users to secretly flag bad content for removal
- Operational Principle: Each post or comment has a 'flag' button that users can click. If enough different people flag it, it is automatically removed. If a user has enough removed comments, then they will receive a warning or penalty.

Upvote

- Purpose: To allow MIT registered users to express that they like a post or comment
- Operational Principle: If a student upvotes a post or comment, then that post or comment will gain one upvote. Posts will display their number of upvotes. Posts and comments can be sorted by number of upvotes.

Data Model:



Security Concerns:

- To make sure users actually attend MIT, we will require users to log in with MIT certificates
- To protect intellectual property and leaking of MIT exclusive material (ie. through posting resources), we require users to log in with MIT certificates
- To protect against injection attacks and XSS, we will use templating (ie. Handlebars) and try to implement other strict input validations for fields and special characters
- To protect against CSRF, we will attempt to implement hidden tokens for each session or auto logout the user after a period of inactivity.

Challenges:

Risks:

- Students may enroll in classes that they are not actually taking and disrupt the class page
 - Solutions: Posts or comments may be flagged, and students with too many removed posts or comments will be warned or penalized
- Students may abuse the flagging system and try to get every post or comment deleted

- Solutions: Students have a limit on the number of flags (10 flags) they can give out every 12 hours
- Not enough students may use the app and it may die very quickly
 - Solutions: We will try our hardest to make this app the best that we can, so that users will want to keep using it.
- Students may spam class pages
 - Solutions: Posts or comments may be flagged, and students with too many removed posts or comments will be warned or penalized

Design choices :

- We allow users to flag “bad” posts instead of downvoting them, because we don’t want to discourage or scare users from posting if they think they will only receive downvotes
- We allowed users to post resources because other students might find it helpful to look at notes or links. We chose to allow the posting of resources in posts to make it easier for students to access everything on one page (as opposed to a question/answer page and a resources page).
- We set the homepage as one of the user’s classes instead of the user’s profile (if user has not classes added, then the homepage will be their profile + an add class option). This caters to the user in the sense that when the user visits the web app, they want to see their class’ feed instead of their own profile.
- Instead of wiping a class’s page after every semester, we decided to create separate pages for every offering of the class. Old class pages will be archived. We did this so that old information will not be lost, in case anyone wants to refer back to it.

UI Design:

The design of the user interface consists of a single main view, for which almost all of the relevant data is displayed. From this view, information about a registered class shows up on a class page, along with the collection of posts for that class. The registered class shows up like a tab on the page (design choice) - one for each registered class. Furthermore, the content of the page is determined by the opened class tab. The UI then allows on the open class to add posts and skim through the collection of posts for a class. The wireframe for this main page looks like so:

Cloop

Profile


6.170

6.034


14.01

Find

Write a post




Full Name


2 hours ago 0 

Hi guys i don't know what I'm doing please help

2 Responses




Full Name

5 days ago 3 


Here is this nifty resource



0 Responses



Full Name

10/15/16 1 

Does anybody want to pset in the stud tomorrow at 7pm?

4 Responses



An example of a more thorough design implementation that could arise from this general UI specification is as follows:

Cloop - Manuel Guillen

← → ↺ 🏠

🔍

Cloop

Log Out

6.034 Artificial Intelligence

6.170 Software Studio

6.837 Computer Graphics

18.701 Algebra I

Search

Profile

6.170 Software Studio

[Class Website](#) [Students Enrolled](#)

Write a Post:

Submit

☐ Resource


☒ All Posts ☐ Resources

🔍 Search

Danny Tang

November 13, 2016 - 10:53 PM

Found this really good book on Web App Development

 **Getting MEAN with Mongo, Express, Angular, and Node**

Manuel Guillen

November 13, 2016 - 6:34 PM

Can anyone explain how sessions work, both the concept and its use in Node?

[View 3 comments](#)

Both wireframes capture the important aspects of the UI design, and have details that will end up being implemented in the interface. Following the implementation of the main view as described, we also have a login page - nothing special, just a log in form. Also, a new page is viewed when accessing one's profile, allowing for registration of more or less classes:

Cloop

Profile

6.170

6.034

14.01

Find

Your Name

Your Kerberos

Classes

6.170

×

6.034

×

14.01

×

add class

A thing to note is that, this is simply another view generated on the same page as the main view discussed before. The profile view outputs information about the user and his/her profile (including classes said user is in) and the user is able to manipulate their own information. For a user searching for classes or other users, we have the search or find view

Cloop

Profile

6.170

6.034

14.01

Find

Find Classes and Students

Joan

×

Joanna Han

i

jmhan

Joanna Man

i

jmman

From here, other users and other classes can be found, and their information (not the entire collection of posts, but characteristic data of the entity - user or class) is displayed in a small popup on the middle of the screen. The choice of popup rather than new page was chosen for the UI as a consequence of the relatively small amount of data needed to be displayed for a user profile or a class profile.