

What is this:

A Slack wrapper that uses the Slack IRC integration and has visualizations and organization for meeting times and locations

PROJECT TODO:

- 1) Location and personal preference selection screen
- 2) Time grid selection
- 3) User auth (OAuth with Slack?)
- 4) Main window with IRC client and schedule stuff
- 5) Banners for next/requested meetings
- 6) Request meeting popup/screen
- 7) Way to straight up just enter a meeting time that is agreed upon
  - a) Screen shows recommendations or something
- 8) Way to edit existing meeting information
- 9) Backend:
  - a) Database to hold information about meetings/users
    - i) User preferences
    - ii) Credit card information
    - iii) Location votes
    - iv) Meeting time/location/date
    - v) Active meeting requests
  - b) Google Maps integration
  - c) Slack IRC integration
- 10) Frontend:
  - a) Make it look gud idk

What we want to do:

- 1) Sam
  - a) Database stuff kinda
  - b) API stuff
- 2) Sean
  - a) Is willing to listen to Sam for help
  - b) Please
  - c) "Google it"
- 3) Ditto
  - a) Ace
  - b) Sehat

What we want to learn how to do:

PROPOSAL TODO:

- A summary of the technologies you intend to use

- Any functional and non-functional requirements
- An estimated project schedule

# Web Systems Development

## Project Proposal

[\Collude.it]

Powered by MEESE

09.13.2019

**M**udse  
coh**E**n  
h**E**ffernan  
bayu**S**ik  
s**E**hat

Jonah Mudse, Sam Cohen, Sean Heffernan, Kayla Bayusik, and Ace Sehat

# SUMMARY

---

Collude.it is a chat application which will focus on making the scheduling process for team meetings more convenient and easier to navigate. When MEESE was planning its first meeting, we went through various websites to collect information that was relevant to the meeting; we used When2Meet to coordinate times, Strawpoll to decide on an optimal location, and Slack to discuss the results and decide on a final meeting point and time. Collude.it will combine the features of Strawpoll, When2Meet, and Slack into one application so that users can easily list their preferences and schedule meetings that work for every team member, or as many members as possible, on the fly.

The userbase of Collude.it will primarily consist of RPI students who need to come together for group meetings, whether it be for group projects, club gatherings, or other events outside of the curriculum. With so many different sports, clubs, and classes, most students have busy schedules with rare moments of overlap between any two of them. Collude.it is perfect for these sorts of students, allowing them to efficiently and conveniently communicate and coordinate with one another. Better yet, Collude.it will save the user's preferences, allowing them to more readily accommodate for each other and schedule meetings in the future with less work.

Most groups at RPI utilize some form of online chat service to keep students in touch, but they can't agree on one platform. Every class and club picks for themselves Discord, Slack, Mattermost, or an other, less common service to connect with, resulting in students using a different platform for everything. We hope that Collude.it will be a good alternative to these other chat applications and can become a primary source of communication between students here, with the added benefit of helping them schedule meetings without the hassle.

# SCHEDULE

---

Provisional Date:	Task to have been accomplished
Friday, Sept 20	<ul style="list-style-type: none"> <li>Complete Independent Outlines <ul style="list-style-type: none"> <li>For collective review and finalized wire frame</li> </ul> </li> </ul>
Tuesday, Oct 1	<ul style="list-style-type: none"> <li>Basic Research on Implementing Private Messaging <ul style="list-style-type: none"> <li>Slack IRC no longer viable, need to look toward alternatives</li> </ul> </li> <li>Front End Architecture <ul style="list-style-type: none"> <li>To follow website tree structure as outlined in finalized wire frame</li> </ul> </li> </ul>
Friday, Oct 4	<ul style="list-style-type: none"> <li>Simplistic Implementation of MySQL for User Accounts</li> <li>Develop Thematic Consistency and Marketable Design</li> </ul>
Friday, Oct 11	<ul style="list-style-type: none"> <li>Develop Account Creation <ul style="list-style-type: none"> <li>User signup form</li> <li>Hashed Password Creation and Verification</li> </ul> </li> </ul>
Tuesday, Oct 15	<ul style="list-style-type: none"> <li>Private Group Channel Creation <ul style="list-style-type: none"> <li>Maintain chat record</li> <li>Display uploaded files</li> </ul> </li> </ul>
Friday, Oct 18	<ul style="list-style-type: none"> <li>Availability Input Menu</li> <li>Location Input Menu</li> </ul>
Tuesday, Oct 22	<ul style="list-style-type: none"> <li>Collectivized Optimal Meeting Conditions <ul style="list-style-type: none"> <li>Algorithm used to display recommended meeting information based on what works best for most people in the group</li> </ul> </li> </ul>
Friday, Oct 26	<ul style="list-style-type: none"> <li>Implement Notification Bar <ul style="list-style-type: none"> <li>Displays upcoming meets that are agreed upon</li> </ul> </li> </ul>
Friday, Nov 8	<ul style="list-style-type: none"> <li>Implement Request for Meeting <ul style="list-style-type: none"> <li>Suggest a meeting based on optimal data that is sent out to all individuals joined in the group</li> </ul> </li> </ul>
Friday, Nov 15	<ul style="list-style-type: none"> <li>Catchup Week/ Time for Potential Displaced Schedule</li> </ul>
Tuesday, Nov 19	<ul style="list-style-type: none"> <li>Finalize/Polish Backwork</li> </ul>
Friday, Nov 22	<ul style="list-style-type: none"> <li>Prepare Project for Release/Presentation</li> </ul>

# TECHNOLOGIES AND REQUIREMENTS

---

## Front End Technologies:

- ❖ HTML
  - Foundational content, structure, and semantics
- ❖ CSS
  - Developing a professional thematic look that outlines an intuitive design without detracting from the functionality
- ❖ Javascript
  - Giving life to the platform by creating dynamically updating content and giving users responsive feedback given their use and input

## Back End Technologies:

- ❖ Python
  - Creating an algorithm that optimizes preference data given any group size and outputs suggestions accordingly
- ❖ PHP
  - Server side programming for saving files on the server, email notifications, database queries, and developing a secure login system
- ❖ MySQL
  - Maintaining a database of account data, meeting information, preference data

# SITE DESIGN

---

When Collude.it first loads, it takes users to a homepage<sup>1</sup> divided into three general sections. The main, largest section of the screen is devoted to the chat window, where users can view messages from their group members and send messages of their own. Off to the left of the message board is a collapsible sidebar with two subsections: the top contains information about times when the group is most likely to be able to meet, and the bottom one shows the optimum meeting locations.

Above the chat window is a notification banner, which updates users with information about pending requests and upcoming meetings. On the right side of the banner is a button to allow users to request a meeting time and location to the rest of the group.

From the homepage, the user can navigate to two other pages in order to edit their meeting preferences using buttons contained in the sidebar. The first of these is the location preference page<sup>2</sup>. Here users can enter the name of a convenient location for them to meet their teammates and add it to their list. Once additions to the location list have been made, up and down arrows beside the place names offer users the opportunity to change the preference ranking of each location. The second of those pages is the time preference page<sup>3</sup>. This page consists of seven boxes, one for every day of the week, in which the user can enter the time slots when they will be free on each given day.

In the case of both pages, upon clicking the submit button, a POST request is made, and the user's new choices and availabilities will be stored in the MySQL database. The user will be returned to the homepage.

Finally, when all of the team members have entered time and location preferences, the user can click on the "Request a meeting..." button found at the top of the homepage, which will create a dialogue window<sup>4</sup>. Here, the user can select a day,


time, and location that work best for them and, ideally, the rest of the team for a potential meeting. Clicking the “Request” button will create a formal meeting suggestion and cause a notification to appear in the banner on the homepage for all team members. From that banner, users will be able to accept or reject the request, if they can or cannot make it. Once enough members have accepted the meeting request, it will become official, and an upcoming meeting banner will replace the confirmation banner.



# DESIGN REFERENCE PHOTOS



---

Reference 1:



	Sam wants a meeting at 2 AM on Wednesday		Your next meeting is on Monday at 5 PM	<a href="#">Request a meeting...</a>
	<a href="#">Confirm</a>	<a href="#">Reject</a>		
<b>Availability:</b> 1) Mon @ 5 PM 2) Sat @ 12 PM 3) Wed @ 2 AM  <a href="#">Edit time availability</a>	<b>Sam:</b> Hello how are you?			
	<b>Jonah:</b> Good.			
	<b>Sean:</b> Good.			
	<b>Kayla:</b> Good.			
	<b>Ace:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
	<b>Sam:</b> Good.			
<b>Locations:</b> 1) Ricketts Building 2) Student Union 3) RAHPS A 4) VCC  <a href="#">Edit location preference</a>				

Reference 2:



Preferences:





1) Ricketts Building



2) Student Union



3) Pittsburgh, PA



4) Dr. Jackson's Mansion

Submit

Reference 3:

Time Preference:

Sunday:

Monday:

Tuesday:

Wednesday:

Thursday:


Friday:

Saturday:

Submit

Reference 4:

Request a meeting:

Day:  

Time:

☒

or

☐

Location:

