Aptistudy

Team 04

Hatfield, Josh (jhatfie@iu.edu)
Johnson, Dylan (dylajohn@iu.edu)
McGaugh, Alex (amcgaugh@iu.edu)
Yen, Johnathan (jpyen@iu.edu)

This report is prepared for INFO I494/I495 Capstone.

Table of Contents

Overview

In essence, this Capstone experience encapsulates a team-oriented web application, where we derive a set of ideas and functionalities that ultimately intend to streamline a process for our predetermined target demographic. For this reason, as students who have worked across various study groups with different goals, we felt that we could address the shortcomings with current applications in regard to study group organization and ensure that, through a sociotechnical perspective, we can develop an equitable application facilitating our users' needs on top of providing them a level of user agency for interacting with our application. Given the multilateral scope and complexity of our application, we enumerate the purpose of this documentation and what each section broadly contains below.

For the team member composition, idea and purpose, and platform requirements/dependencies sections, we highlight many of the introductory components that elucidate why we chose the nature of our application in addition to how we plan to render it from both a teamwork and technical perspective. Since we individually possess many strengths that can coalesce with what types of contributions we can make, it is important that we can predefine our roles and therefore gauge which technical requirements we believe are feasible such as APIs integrations, the Rivet framework, etc. By additionally comparing potential competitors, we can apply a more holistic approach for differentiating our own application in terms of the functionalities we wish to provide as well as the depth of scope for the application itself.

In the subsequent sections containing our Epics / user stories, responsible computing statement, wireframes, and database design, we magnify the more technical aspects we plan to follow for our application since these sections essentially act as the foundation for what our application will entail regarding our overall logical data flows and visualizations. The Epics and user stories provide initial conceptualization as to how we revolve our functionalities around our intended user, which act as the lowest yet most important level for defining our application's logic. Then, the responsible computing statement assures that this logic we implement maintains the aforementioned equity and sociotechnical perspective so that our application provides a strong ethical framework in addition to assurance for our users. Next, the wireframes translate our application's logic to frontend visualizations, where we attempt to replicate and iterate on the perceived user experience that we hope to offer. Lastly, the database entity relationship diagram (ERD) design defines more granular flows/relationships on both the frontend and backend interactions that thus delineates our entire application.

With the next three sections pertaining to user experience (UX) methods, our project timeline, and methods for successful teamwork, these address more nuance with how we seek to establish team expectations and progress for our project throughout the Spring semester. Because we are developing a tangible application for a target demographic, we necessitate feedback so that we can make iterations to our expectations and subsidiarily update our timeline to reflect these iterations; the UX methods include different forms of feedback based on our wireframes and overall project. The project timeline offers both a broad and detailed view of our plan of attack based on our individual strengths we offer to our team mentioned previously.

Finally, the successful teamwork section exhibits responsibilities and actions that we can internalize so that we can conclusively overcome any roadblocks and meet our expectations for developing a strong application.

For the acknowledgement and appendix sections, these include outside resources that helped us throughout the semester for procuring our documentation. More specifically, acknowledgements relate to faculty resources or other students who assisted by providing valuable feedback and guiding us through this documentation process; the appendix sections contain raw results from our UX methods that were valuable for drawing insights for the UX methods section.

Team Member Bios/Composition

As students hoping to excel within Capstone, we understand the importance of applying our experience in addition to recognizing our various strengths and weaknesses so that we can find the most optimal ways to accomplish our project. Even though we may have been exposed to prevalent coursework prior to this course, we acknowledge that individual members may feel more comfortable with specific skill sets and intend to outline these for each member below. Collectively, this will help us better understand ourselves and each other for a successful year in Capstone.

Josh Hatfield

Josh Hatfield is a fourth year senior intending to graduate with a major in informatics and cognate/minor in security informatics. He feels generally proficient with various programming languages, which include Python from the foundational Luddy courses, R/RStudio through business modeling in Kelley, and introductory C language in CSCI-C 291. As a prior Kelley student, having additionally completed the INFO-I 300 course in UX design, Josh also feels that he can incorporate his knowledge of intuitive UX design to determine useful functionality for our project going forward. Lastly, he has experience from multiple consulting projects and has given effective presentations for risk assessments, SWOT analyses, target demographics, and industry insights, which may help when determining Epics for our users.

Dylan Johnson

Dylan Johnson is a senior at IU and is working towards a bachelor's degree in informatics. He has general proficiency in coding in Python, HTML, CSS, and some SQL. He will be able to help the team when it pertains to UX and UI design. Throughout his time at IU, he has plenty of experience being in a group and presenting projects. This will greatly help the team as we move forward and eventually present our project.

Alex McGaugh

Alex McGaugh is a fourth year senior studying informatics with a cognate and minor in business. While he has taken all core informatics classes proficient in Python, SQL, HTML, and CSS, his best skills include UX/UI design, SQL, and ethics. He plans to help his team mainly through his UX/UI design and his ethics background. Even though his top skills are not Python, he plans to help his team with every single aspect of the project using all his skills he has learned throughout his degree in informatics.

Jonathan Yen

Jonathan Yen is a fourth year undergraduate student studying Informatics at the Luddy School of Informatics with a cognate and minor in Psychology. Jonathan is more familiar with the coding end of things such as Python and MySQL. He has also had the opportunity to work

with a number of groups in the years leading up to his fourth year, affording him at least some baseline group experience and communication skills, if not more. Jonathan will be able to help handle much of the back end database development and management while providing feedback on the development of the other branches within the project.

Idea and Purpose

Since we recognize that many students tend to experience some sense of incongruity when utilizing beneficial opportunities such as collaboration or groups that can help maximize their ability to excel in courses and finding methods that tangibly facilitate these interactions, our team intends to design an application that promotes this interactivity through its central theme of course groups. Thus, we intend to solve the problem of students who, for example, do not know anyone else in their courses but hopes to identify others with similar academic and/or interpersonal goals in order to support each other for fostering success; we also intend to quell many anxieties for teams since we ourselves have encountered circumstances where organizing team meetings, times, and locations were infeasible given that we were not able to visually account for our needs in relation to each other.

While our project focuses on course groups, we want to distinguish our project by streamlining how students can expand initial interaction inside courses to beyond them as well as personalizing its functionality in a way that makes our application pervasive to student needs. Although explicated better in the first epic, one way of accomplishing these goals in a unique way is to offer a compatibility quiz for users, who correspond their desires for interactivity with what they hope to accomplish within their courses. Thus, two students who "match" based on their desires can connect and assist one another within the same courses, which may not have been possible if they had never met each other in class. Moreover, another way our application can uniquely solve the problem described for teams is coalescing tools that simplify their coordination so that they can focus on their work. These tools *may* include but are not limited to: group calendar integration so that teams can observe open time slots to meet, map integration to determine locations close to all members, and the ability to reserve open rooms based on team needs. Overall, we are genuinely interested in solving these problems since, as mentioned, every member on our team has experienced obstacles such as feeling isolated yet seeking help in classes or wanting a more effective way to delegate team logistics, so we hope to propagate a solution that can mitigate our described problems for other IU students.

Based on recent evaluation, we note that When2meet has some functionalities we intend to incorporate such as its calendrical function and meeting scheduling, but we consider it a baseline competitor in which we can consider its tools yet refine our product to offer more optionality for users. In terms of collective functionality, however, we consider Microsoft Teams as a competitor since the application allows individuals within schools and organizations to add themselves to teams, which eases their ability to utilize similar tools we intend to incorporate like meeting management and some form of calendrical integration. However, our team recognized that the application struggles to delineate attention to the same approachability for personalization we want within our application and how this personalization may help students benefit through a more social component of our application. In turn, our intended focus for students on campuses can limit the scope of application compared to systems like Microsoft Teams but will hopefully allow us to better address the needs of our target demographic concerning how we blend classroom and social components together. As for applications like GroupMe, we appreciate its flexibility for team management but felt that it felt somewhat lacking

in terms of the customization we seek (Ex. maps or group calendars). Lastly, after contemplation, we felt that the most comparable application regarding social features we want such as map integration and attendance would be TeamSnap, but this differentiates from our system since it emphasizes sport management rather than academic use.

Platform/Technical Requirements/Dependencies

Our team is choosing to render this project through a *web application*. This seemed to be the best route for us as our team's experience specializes in web development and for the fact that a web application would best integrate with the APIs we would potentially use. We would agree that while a mobile application would look and feel better, a web application would be the most accessible to all user groups and stakeholders, essentially working as long as the user has an internet browser. This being said, functionality is what we are looking for most in this application, and as such, knowing what we need and being able to do it becomes the foremost important technical skill we have. Therefore developing a web application with skills we practiced in the previous I-210, I-211 I-308, and other supplementary courses seems to be the best course of action.

This being said, even if we do stay true to what we have learned from previous courses, we will inevitably need to explore items that we are unfamiliar with. These items come in the form of integrations. Outsourcing from other systems or tools that have already developed is something our group is unfamiliar with, however it is a requirement for this prototype to be successful. This being the case, we still need to implement integrations that are meaningful to our project, and seeing as our project revolves around the student user base, location finding and scheduling, we decided that integrating resources from the Rivet Framework, Google Calendar, Google Maps, and possibly even the IU database of classes would best serve our image of a good student social planner.

Epics

For our nine epics, we wanted to keep the end user in mind and what was best for them. We decided to focus primarily on functionality and purpose. This meant creating features with purpose that are also easy to use and understand. We did our best to keep every epic within the scope of our project. Every single epic builds on the last and it is this combination that makes our system so powerful. We have listed below the nine epics that we finalized on for our project.

- Epic 1: User Profiles
- Epic 2: Compatibility Questionnaire
- Epic 3: Study Group Organization
- Epic 4: Host Group Study Session
- Epic 5: Join Group Study Session
- Epic 6: Ticket System Form and Support
- Epic 7: Classroom Availability with Maps
- Epic 8: Notifications with Group Calendars
- Epic 9: Login and Registration

Furthermore we will break down each epic and explain what data elements will be involved with sending, receiving, and more. Each epic listed above has many small components that make up it. We will also dissect those and explain the inner workings and the processes behind each one.

Epic 1: User Profiles

Functionality: Users will be able to create a profile and edit the settings to best match their self and preferences. Users will also be given the option to view other users profiles. When first setting up a profile, the user will be prompted to fill out logistic questions such as major, year, bio, interests, clubs, etc. This will help with finding better compatible people to work together or if users want more social functionality beyond just class work.

- Create User Profile
- Edit User Profile
- View User Profile
- Delete User Profile Fields

User Story 1: Create User Profile

As a user, I should be able to create an individual profile personalized to my preferences.

Trigger	A button in the navigation bar in the dropdown menu under ‘Profile’ labeled ‘Create Profile’
Details	A user clicks on the button and is redirected to a screen where they can begin to fill out questions to make their profile unique to them.
Access	Public for all users
Required Field(s)	User First Name (String, VARCHAR 20) User Last Name (String, VARCHAR 20) User Email (String, VARCHAR 50) User Classes (String, VARCHAR 50) User Major (Dropdown List) User College Standing (Dropdown List, ex: Senior)
Input Field(s)	User Bio (VARCHAR 150)
Action(s)	User clicks a button labeled ‘create profile’.
Exception(s)	If another user has already registered an account with a preexisting email, the user will not be allowed to include that email within the profile. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-Data validation comparing all emails in the database to the one currently being submitted for account creation* *-Attempt (harmless) code injection into name, email, and class parameters and see if sanitization has correctly applied to database insertion*

User Story 2: Edit User Profile

As a user I should be able to update my profile in accordance with my daily life.

Trigger	A button in the navigation bar in the dropdown menu under ‘Profile’ labeled ‘Edit Profile’
Details	User is able to modify certain elements of their profile after selecting this option
Access	Private for authenticated user
Required Field(s)	User First Name (String, VARCHAR 20) User Last Name (String, VARCHAR 20) User Email (String, VARCHAR 50) User Classes (String, VARCHAR 50) User Major (Dropdown List) User College Standing (Dropdown List, ex: Senior)
Input Field(s)	User Bio (VARCHAR 150)
Action(s)	Redirects not viewing the present user profile after clicking the save or cancel button.
Exception(s)	Invalid email and/or other inputs. Profile must be initiated and have inputs when created for edit functionality to apply. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-Test to see if the email is the same; if email changed then test to see if it is valid and if it exists already in the database* *-Attempt (harmless) code injection into name, email, class, club, and interest parameters and see if sanitization has correctly applied to database insertion* *-Consider (not concrete but is a consideration) sending a verification email before saving profile so that changes can only populate in database once verification has been confirmed*

User Story 3: View User Profile

As a user, I should be able to view my own profile.

Trigger	This workflow is triggered via a navigation bar with an image of the user profile and an indicator titled 'Profile.' The button is accessible from any page on the application and allows a user to view their own profile.
Details	The user should be able to view the updated sections of their profile once updates, additions, and deletions have been made. This contains the generic non-editable page that can only be changed via another user story function.
Access	All authenticated users logged in.
Required Field(s)	User First Name (String, VARCHAR 20, Pre-populated) User Last Name (String, VARCHAR 20, Pre-populated) User Email (String, VARCHAR 50, Pre-populated) User Classes (String, VARCHAR 50, Pre-populated) User Major (Dropdown List, Pre-populated) User College Standing (Dropdown List, Pre-populated) User Bio (VARCHAR 150, Pre-populated)
Input Field(s)	N/A
Action(s)	'Profile' button redirects to a static profile page and retrieves information stored within the database associated with the user. Contains buttons for other user story functionalities
Exception(s)	If none of the fields have been populated yet, page redirects to 'Create User Profile' indicated in user story 1 for Epic 1
Test Criteria	*-Test to see if user profile is in the database and is correctly grabbed by the query result on the frontend; test simple updates based on second user story and ensure that new changes are populated in database as well as frontend PHP page* *-Attempt invalid data types within specific fields, such as only space characters or integers within the user first and last name entries* *-Test to see if user profile and IU username match once authenticated*

User Story 4: Delete User Profile Fields

As a user, I should be able to delete populated data for any optional fields on my profile.

Trigger	After clicking on profile in the navigation bar, clicking on the edit button will give you the option to delete information out of the user profile's bio.
Details	Users are able to delete non required fields from their profile.
Access	.All users logged in with an account.
Required Field(s)	User First Name (String, VARCHAR 20) User Last Name (String, VARCHAR 20) User Email (String, VARCHAR 50) User Classes (String, VARCHAR 50) User Major (Dropdown List) User College Standing (Dropdown List, ex: Senior)
Input Field(s)	User Bio (VARCHAR 150)
Action(s)	Information in the optional fields; users choose to delete will be deleted from the database and not show up on users profile.
Exception(s)	Users have to have information in required fields.
Test Criteria	*-Once a user creates a bio, the user should be able to delete the bio and therefore, the user should no longer see this bio on their profile page. This deletion should also reflect in the database for that specific user, which should now be a NULL value* *-Attempt to delete required fields such as user first name, or perhaps only contain space characters*

Epic 2: Compatibility Questionnaire

Functionality: Users will be given the option to initiate, edit, and view compatibility quizzes, which utilize algorithms to "match" users based on the number of comparative independent questions, categories, and needs within respective course groups. Categories could expand beyond courses and integrate some granularity for groups that help facilitate clubs, groups seeking to help prospective students apply for internships, etc

- Initiate Compatibility Questionnaire
- View Compatibility Questionnaire
- Edit Compatibility Questionnaire
- Delete Compatibility Questionnaire

User Story 1: Initiate Compatibility Questionnaire

As a user, I should be able to take a compatibility questionnaire in order to be matched with individuals that best fit my study needs.

Trigger	A button in the navigation bar in the dropdown menu under ‘Compatibility Quiz’’ labeled ‘Take Compatibility Quiz’
Details	Once a user has registered an account, they can take a compatibility quiz which features rating questions. This information will be collected to match students with the best individuals for them
Access	Public access for all users who have registered an account
Required Field(s)	None
Input Field(s)	Most Active Study Hours (Dropdown List) Types of Individuals for Matching (Dropdown List with Multi-Value Options) ● Ex. Same major, same course(s), same year of study Looking for Friends? (Dropdown List) Online and/or In-Person Meetings Prefererable (Dropdown List) Prospective Intern (Dropdown List)
Action(s)	User clicks on the button and is redirected to a page where they can begin filling out questions If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Exception(s)	If a user has already taken a quiz, a pop up will tell them they can edit their results.
Test Criteria	*-Test to see if user has taken a quiz, where the algorithm compares between the user results and other users; if all entries are empty, ensure that a message reflects this on the frontend* *-Click on various combinations of the single and multi-value populated dropdown lists as well as save results; compare with database changes and match updates if applicable*

User Story 2: View Compatibility Questionnaire

As a user, I should be able to view the compatibility questionnaire I have taken.

Trigger	Button located on the user profile, viewing a list of previous questionnaires, clicking on a specific questionnaire pulls it up in a viewer.
Details	Once a user has taken a questionnaire it is stored to their account, a user can change/edit at their discrepancy.
Access	Being private or public depends on other features, whether a meeting session is going to be for users of a specific type or just an algorithm attempting to match them together.
Required Field(s)	Most Active Study Hours (Dropdown List) Types of Individuals for Matching (Dropdown List with Multi-Value Options) ● Ex. Same major, same course(s), same year of study Looking for Friends? (Dropdown List) Online and/or In-Person Meetings Prefererable (Dropdown List) Prospective Intern (Dropdown List)
Input Field(s)	The same as required fields, but those are now mutable.
Action(s)	Other than scrolling through their answers there shouldn't really be anything to change here, immutable
Exception(s)	N/A
Test Criteria	*-Once a user has taken their* questionnaire*(s) go to Profile >> View Test Results >> test to see specific* questionnaire *evaluation and* questionnaire *list; matches should have been initialized, with a score and percentage evaluating how strong the match is* *-Update results with third user story and navigate again to Profile >> View Test Results >>* questionnaire *comparison from previous view page and see if query correctly updates database's populated changes*

User Story 3: Edit Compatibility Questionnaire

As a user, I should be able to update my compatibility questionnaire , which reflects updated matches if applicable.

Trigger	This workflow is triggered via a button titled 'Edit Quiz' that is available once a user clicks on the initial button for user story 2 to view their compatibility quiz.
Details	The user should be able to edit and update the fields provided in their compatibility quiz, regardless of section or updates made. All fields should be editable by the user.
Access	All authenticated users who have completed a compatibility quiz.
Required Field(s)	N/A
Input Field(s)	Most Active Study Hours (Dropdown List) Types of Individuals for Matching (Dropdown List with Multi-Value Options) ● Ex. Same major, same course(s), same year of study Looking for Friends? (Dropdown List) Online and/or In-Person Meetings Prefererable (Dropdown List) Prospective Intern (Dropdown List)
Action(s)	'Update' button saves the updated changes for the information in the database. Proceeds back to the static compatibility quiz page for viewing answers. Updates compatibility matches based on changes.
Exception(s)	Do not proceed with saving if the user populates a field with an incorrect data type. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-Where multi-value options exist, try combination of changes to value options and save changes; compare database population and query result on frontend, where matches should update based on the changes if applicable within the algorithm* *-Change user profile parameters like User Major, save changes, then select the dropdown list option for the same major; compare database population with query result on frontend, where query should have the updated major value from the database and update matches if applicable with the algorithm (Ex. If updated_user_major == other_user_major…)*

User Story 4: Delete Compatibility Questionnaire

As a user, I should be able to delete my compatibility questionnaire, which removes information associated with my questionnaire and any matches generated from the algorithm.

Trigger	Located within the view compatibility quiz. After clicking button edit quiz users will have the option to delete the quiz entirely. Users will be shown a “Are you sure” pop up button to confirm.
Details	When a user deletes their compatibility quiz they will no longer receive matches with other users.
Access	All authenticated users who have completed a compatibility quiz.
Required Field(s)	N/A
Input Field(s)	Delete Compatibility Quiz and Confirm
Action(s)	Delete quiz button will have another pop up button that says confirm delete pressing this button will delete the quiz from the users account allowing them to create a new quiz or not have an active quiz in their account.
Exception(s)	N/A
Test Criteria	*When a user confirms delete, they should not see a compatibility quiz in the compatibility quiz section of their account.*

Epic 3: Study Group Organization

Functionality: Users can facilitate most of their study groups as both coordinators and members, where they can create new groups based on a set of parameters or with compatibility matches, edit groups based on these parameters in addition to member roles, join already-created groups as a user, leave groups at any time, and delete groups. Ultimately, study groups are tailored to their academic needs and can easily be generated based on the compatibility matches or through a more manual member selection process.

- Create Study Group
- Edit Study Group
- Join Study Group
- Leave Study Group
- Delete Study Group
- View Study Group

User Story 1: Create Study Group

As a user, I should be able to create a study group in order to increase interactivity with other users.

Trigger	Located as a button in the navigation bar in the dropdown menu under ‘Study Groups’ labeled ‘Create Study Group’ Once compatibility matches' profiles are generated, user can attempt to initialize a study group with a match, automatically populating the user and match within the same group as members
Details	The user will be able to define a set of parameters for their study groups, ranging from the number of member slots allowed, type of access (Ex. Only students within the same course, compatibility matches, etc.), and study goals
Access	All authenticated users can create a study group.
Required Field(s)	Study Group Name (String, VARCHAR 50) Study Group Description (String, VARCHAR 50) Number of Members Allowed (Dropdown List / Slider Integer Value) Study Goals (Checkbox List; String, VARCHAR 150)
Input Field(s)	Course Selection (String, Dropdown List from Populated User Profile or String VARCHAR 50) Color Choice (Radio Button Selection)
Action(s)	'Save Study Group' button proceeds to 'Home' page. Green bar and message with 'Study Group Saved' indicated on the top of the web application.
Exception(s)	If a required field is not populated or incorrectly populated, do not save the study group and indicate that the field needs to be addressed by the user. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-Test to see if the user has courses registered in their profile. Where applicable, users should be able use the populated dropdown list for assigning a study group to a course; add a new course to the user profile, save changes, and compared query result on the frontend with updated values to populated dropdown list* *-Attempt (harmless) code injection into name, description, and goal parameters and see if sanitization has correctly applied to database insertion*

User Story 2: Edit Study Group

As a user, I should be able to edit my study groups, including parameters specified and individuals allowed within the group if I am a coordinator.

Trigger	A button located in the top right corner on the study group's home page labeled 'Edit Group'
Details	Based on the parameters users set when creating study groups, users can edit these fields and change the color of the group, which reflects on the 'Home' page once updates have been pushed by the user.
Access	Private for users who are authenticated as a 'Group Moderator'
Required Field(s)	Study Group Name (String, VARCHAR 50) Study Group Description (String, VARCHAR 50) Number of Members Allowed (Integer) Study Goals (Checkbox List; String, VARCHAR 150)
Input Field(s)	Course Selection (String, VARCHAR 50) Color Choice (Radio Button Selection)
Action(s)	'Edit Group' button redirects the user to the edit page where they can change input settings for their group. Green bar and message with 'Study Group Updated' indicated on the top of the web application when finished making changes.
Exception(s)	If a required field is not populated or incorrectly populated, do not save the study group and indicate that the field needs to be addressed by the user. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-Once edited by the owner/moderator, ensure that updates are also populated for other members within the group; compared frontend query result with database update for a study group* *-Attempt (harmless) code injection into name, description, and goal parameters and see if sanitization has correctly applied to database insertion*

User Story 3: Join Study Group

As a user, if I find a study group I like, I should be able to join it.

Trigger	A button located in the top right corner on the study group's home page labeled 'Join Group'
Details	Once a user has gone to a study group's home page, they will find a 'Join Group' button that will allow them to join the group if there are available spots.
Access	Private only to users who are registered with the same corresponding class as the study group
Required Field(s)	Course Selection (String, VARCHAR 50) Number of Members Allowed (Integer)
Input Field(s)	N/A
Action(s)	User is accepted and joins a study group
Exception(s)	If the user is not taking the class that the study group corresponds to, an error screen will popup If there are no more available spots in the study group, the user will not be allowed to join
Test Criteria	*-Test to see if the user is registered with the correct class as the group based on the course selections in the user profile* *-Test to see if there are available spots in the specific group; also leave the group as another user and ensure that the popup no longer occurs (The user is able to now join the study group)*

User Story 4: Leave Study Group

As a user, if I no longer wish to be a member of a study group, I should be able to leave it..

Trigger	A button in the top right corner on the home screen of a group
Details	If a user wants to leave a study group they have joined, they can click the ‘leave study group’ button. This will remove them from the group and they will no longer receive notifications about that group.
Access	Public to any user who is a member of a study group
Required Field(s)	Member Status (VARCHAR 1)
Input Field(s)	None
Action(s)	Remove user as a member of a study group
Exception(s)	If the user is the host of the group they will not be allowed to leave. A prompt will appear stating “You must delete the group since you are the host”.
Test Criteria	Test to see if the user’s member status is “H” Test to see if the user’s member status is “M”

User Story 5: Delete Study Group

As a user hosting a study group, if I want to discontinue a study group, I should be able to delete the study group.

Trigger	A button located in the top right corner on a group's home page next to the 'Edit Group' button.
Details	If a user is hosting a study group and wants to no longer be the host they will be given the option to delete the study group. It will remove the group from the database and all members will be removed.
Access	Public only for users who are a host of a study group
Required Field(s)	Member Status (VARCHAR 1)
Input Field(s)	None
Action(s)	Specified study group is removed from the database
Exception(s)	If the user is not the host they will not be allowed to delete the group. A popup will appear saying "Sorry but you are not the host of this group".
Test Criteria	Test to see if the user's member status is "H"

User Story 6: View Study Group

As a user, I should be able to view specific study groups

Trigger	A button located in the navigation bar called study groups.
Details	Once clicked, if a user has previously joined a study group from user story 3, they should be able to see your active study groups they are currently in.
Access	Any logged in user who is a member of a study group.
Required Field(s)	Member Status (VARCHAR 1)
Input Field(s)	N/A
Action(s)	Any joined study groups are able to be viewed.
Exception(s)	If the user is not in any study groups there shouldn't be any group to view
Test Criteria	*Click on study groups while having a member status of at least one study group.*

Epic 4: Host Group Study Session

Functionality: If a user wants to "host" a group study session for a class, they can initiate that after confirming the courses they are taking. We will incorporate functions to allow the host to host a session only from the classes they are taking. A host user will be able to edit their group to personalize it for the specific group.

Note: After reading our reviews, there were a lot of questions regarding how we were planning to integrate live meetings on our website. Our group has made the decision to not integrate live meetings into our project for this reason. We deemed that it was out of the scope for our project in particular. If a user wants to "host" a group study session, they will be able to once they have registered their classes with our website. The user will then be able to create a name and provide a brief description of the study session that they are hosting. This should appear as a text listing for other users to find when they are searching for groups. If the host of the study group needs to edit the description they will be able to edit it along with the title. When the website is done, users will be able to click a button and search for groups. There will be a plus sign symbol listed with each group, giving users the option to join a group.

- Host A Study Group Meeting
- View a Study Group Meeting
- Edit a Study Group Meeting
- Delete a Study Group Meeting

User Story 1: Host A Study Group Meeting

User's (mainly students) should be able to create / host a meeting from the application within valid parameters.

Trigger	Located as a button on a dropdown list in the top right corner of the meetings screen called 'Host a Meeting'
Details	User goes through the process providing required information about the meeting, saving and pushing the meeting to the platform.
Access	Public for users within that class
Required Field(s)	Host First name* (string autofill via google?, VARCHAR 20 if manual) Host Last Name* (string autofill via google?, VARCHAR 20 if manual) Host email* (string autofill via google?, VARCHAR 50 if manual) Phone # (string, CHAR 14) Class / Subject* (string) Location* (string, viewable via google maps) Time/date* (MM/DD/YYYY HH:MM TT)
Input Field(s)	Purpose (string) Description (string)
Action(s)	Redirects to viewing the meeting after saving and pushing, wherein the host can edit the meeting or delete the meeting.
Exception(s)	If the user has a similar meeting during the same time, creation is rejected. If location is unavailable or any of the required fields aren't valid, it doesn't allow the user to proceed. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*Dynamic drop downs that pull from the user's data and campus locations. Data validation compared to the student's classes and location availability.*

User Story 2: View a Study Group Meeting

As a user hosting a meeting, I should be able to view the details of any meeting that I am hosting.

Trigger	Located as a button on a dropdown list in the top right corner of the meetings screen called 'View Meeting'
Details	The user is able to view an individual meeting that they are hosting and all of its details
Access	Private to all users who have an authenticated account and are registered to a group meeting
Required Field(s)	Any inputs given via creating the meeting: Host First name* (string autofill via google?, VARCHAR 20 if manual) Host Last Name* (string autofill via google?, VARCHAR 20 if manual) Host email* (string autofill via google?, VARCHAR 50 if manual) Phone # (string, CHAR 14) Class / Subject* (string) Location* (string, viewable via google maps) Time/date* (MM/DD/YYYY HH:MM TT) Purpose (string) Description (string)
Input Field(s)	No input fields here, as the host is just viewing The overall items in the meeting.
Action(s)	Pulls up a screen that displays the meeting information as well as buttons either going back to the calendar, dropping the meeting, or to edit the meeting, from there we should be able to save the information and push it to the database.
Exception(s)	There shouldn't be many exceptions here, other than if there is a fetching error from the database, wherein it sends an error popup for the user to refresh the calendar
Test Criteria	*Clicking through the meetings, and trying to edit or leave the meetings after the meeting has been removed from the database. Seeing if meetings are pushed correctly*

User Story 3: Edit a Study Group Meeting

As a user hosting a meeting, I should be able to edit the details of any meeting that I am hosting.

Trigger	Located as a button in the top right corner on the meetings screen called 'Edit a Meeting'
Details	User goes through the process of editing required information about the meeting, saving and pushing the meeting to the platform.
Access	Only authenticated users who initially hosted the study group meeting can edit the fields populated
Required Field(s)	Host First name* (string autofill via google?, VARCHAR 20 if manual) Host Last Name* (string autofill via google?, VARCHAR 20 if manual) Host email* (string autofill via google?, VARCHAR 50 if manual) Phone # (string, CHAR 14) Class / Subject* (string) Location* (string, viewable via google maps) Time/date* (MM/DD/YYYY HH:MM TT)
Input Field(s)	Purpose (string) Description (string)
Action(s)	Redirects to viewing the meeting after saving and pushing, wherein the host can edit the meeting again or delete the meeting.
Exception(s)	If edits made to the fields cause any invalid cases such as where location is unavailable or any of the required fields aren't valid, it doesn't allow the user to proceed. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-Dynamic drop downs that pull from the user's data and campus locations.* *-Data validation compared to the student's classes and location availability.*

User Story 4: Delete a Study Group Meeting

As a user hosting a meeting, I should be able to delete the meeting, which should be deleted for other users as well.

Trigger	Located as a button in the top right corner on the meetings screen called "Delete Meeting" after clicking the button you will get a confirm delete button to finalize deletion.
Details	Once a user has successfully joined a study group an option to delete that study group will be granted.
Access	All users who have an authenticated account and are registered to a group meeting
Required Field(s)	Course Selection (String, VARCHAR 50)
Input Field(s)	N/A
Action(s)	User deletes study group, therefore removing them from a study group
Exception(s)	If the user is not in a study group they will not have the option to delete one.
Test Criteria	*Once a user has successfully joined a study group an option to delete that study group will be granted. Press button delete study group hit confirm. The study group that was deleted should not be able to be viewed. (user story 2)*

Epic 5: Join Group Study Session

Functionality: If a user wants to ‘join’ a group study session, they will be able to once they have registered their classes. There will be functions that allow them to view, join, and leave any group within boundaries.

- Join a Study Group Meeting
- View a Scheduled Study Group Meeting
- Leave a Scheduled Study Group Meeting

User Story 1: Join a Study Group Meeting

As a user, I should be able to join hosted meetings once I am a part of a group.

Trigger	Located on the meetings page under each available group as a button labeled “Join Meeting”
Details	Once a user has successfully registered their classes they can search for groups hosting meetings in their particular class and join one.
Access	Public for only users associated with a specific class
Required Field(s)	User First Name (String, VARCHAR 20) User Last Name (String, VARCHAR 20) User Email (String, VARCHAR 50) Class / Subject (String, VARCHAR 15)
Input Field(s)	User First Name (String, VARCHAR 20) User Last Name (String, VARCHAR 20) User Email (String, VARCHAR 50) Class / Subject (String, VARCHAR 15) Location (String, VARCHAR 25) Date (DATE FORMAT(MM,dd,YYYY HH:MM)
Action(s)	When the button is clicked a user will be redirected to the group’s homepage where a pop up will show them they successfully joined the group.
Exception(s)	If a user is already in a meeting that is happening at the same time an error will appear saying they are unable to do this due to a conflict in time
Test Criteria	*Data validation compared to the student's current meetings and attempted joined meetings.*

User Story 2: View a Scheduled Study Group Meeting

Users should be able to view a scheduled study group meeting

Trigger	A user is able to view a joined study group meeting by either the calendar or the meetings page but selecting the specified meeting.
Details	Once a user has joined a group meeting that the study group has scheduled it should appear on their meetings list and calendar for them to review.
Access	Anyone that has joined the specific meeting under the same study group.
Required Field(s)	Any of the fields provided when the host made the meeting: Host First name* (string autofill via google?, VARCHAR 20 if manual) Host Last Name* (string autofill via google?, VARCHAR 20 if manual) Host email* (string autofill via google?, VARCHAR 50 if manual) Phone # (string, CHAR 14) Class / Subject* (string) Location* (string, viewable via google maps) Time/date* (MM/DD/YYYY HH:MM TT) Purpose (string) Description (string)
Input Field(s)	There shouldn't be any fields for input via the user here as they shouldn't have permissions to edit the meeting unless they are the host.
Action(s)	From here a user is able to select an option to go back to the previous page.
Exception(s)	If the meeting is deleted while the user has selected the meeting, it either returns them to an error page, or perhaps returns a deprecated version of the meeting that specifies that the meeting has been abandoned.
Test Criteria	*Loading the meetings page or calendar, then deleting the meeting and trying to load the previously mentioned meetings page.*

User Story 3: Leave a Scheduled Study Group Meeting

As a user I should be able to leave a group meeting if I do not wish to continue any longer.

Trigger	A button labeled ‘Leave Meeting’ in the top right corner of the specific groups home page
Details	If a user wants to leave a group they can select this button in order to leave and stop receiving notifications.
Access	Private to only users who are apart of a study group
Required Field(s)	None
Input Field(s)	None
Action(s)	User is removed from the specified study group
Exception(s)	If the user is the meeting host, a prompt will say ‘Sorry you are the host, you must delete the study group’.
Test Criteria	*-Test to see if the user is the group host* *-Test to see if they are a member of the group*

Epic 6: Ticket System Form and Support

Functionality: If users need any assistance or find any technical issues within the system, they can log these into a form based on the type of issue and any other relevant factors we consider. Users will also be able to access and delete previous tickets they made within the system if, for example, they need documentation of ticket support or have previously received administrative support.

Note: This Epic is replacing our chat functionality since we felt that this was both more intuitive and necessary for our system. We have also considered the possibility of the PayPal donation integration in place of this Epic.

- Initialize Ticket Support Form
- View Tickets Sent
- Delete Previous Tickets

User Story 1: Initialize Ticket Support Form

As a user, I should be able to fill out a ticket support form and send it for subsequent manual review of the ticket.

Trigger	Located on the navigation bar, this workflow is triggered via a button titled 'Support,' which allows navigation to the ticket support form if none have been generated. Otherwise, include a dropdown list on the 'Support' button, where form is accessible on the list below the 'Tickets Filed' link; can also be accessed using a button called 'File Ticket' on the 'Tickets Filed' page listing previous tickets filed
Details	The user should be able to define the type of support needed and manually document issues they have encountered within the web application. Our system should automatically save the time and date once the form has been sent
Access	All authenticated users who are logged in
Required Field(s)	See Input Fields
Input Field(s)	Type of Assistance Required (Multi-Value Dropdown List) Severity of Issue (Dropdown List of INTs from 1-5) Description of Issue (String, VARCHAR 256)
Action(s)	'Send Ticket' button saves information to the database and stores the ticket on the 'Tickets Filed' page for viewable access. Proceeds to 'Tickets Filed' page on successful save and includes a green bar on the top of the page with 'Ticket Successfully Sent'
Exception(s)	If a required field is not populated, do not proceed with the save and indicate to the user that the field is required. Ensure that notifications for other errors such as incorrect string usage or length are indicated. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-Attempt (harmless) code injection into the description parameter and see if sanitization has correctly applied to database insertion* *-If all tickets are deleted, test to see if the application automatically regenerates the form on the 'Support' button on the navigation bar and removes the 'Tickets Filed' link on the dropdown list*

User Story 2: View Tickets Sent

As a user, I should be able to view my previous tickets sent and note the ticket identification as well as date/time sent for each ticket.

Trigger	Located on the navigation bar, this workflow is triggered via a button titled 'Tickets Filed,' which allows navigation to a page where all previous tickets are listed. This page should not be accessible until a ticket has been filed by a user. Users can then click on individual previous tickets to view.
Details	The user should be able to see documentation of all issues they sent within the ticket support system. Previous tickets are filtered by time and date on the 'Tickets Filed' page, include the ticket identification number as well as the time and date for submission, and display static data of the inputs made by the user.
Access	Authenticated users logged into the system
Required Field(s)	Ticket Identification Number (Prepopulated) Ticket Date and Time submitted (Prepopulated) Type of Assistance Required (Prepopulated, Value(s) selected from Dropdown List) Severity of Issue (Prepopulated, INT from 1-5) Description of Issue (Prepopulated, String, VARCHAR 256)
Input Field(s)	N/A
Action(s)	'Tickets Filed' page contains viewable tickets and information populated from the database. Clicking on an individual ticket provides static information of prior tickets made as well as 'Return to Tickets Filed' and 'Delete Ticket' buttons on this page
Exception(s)	A user should not be allowed to view tickets generated by other users. Users should also not be allowed to view previous tickets if they have been deleted from the database.
Test Criteria	*-Submit form and check to see that the correct date and time stored within the database for a ticket entry matches when the ticket was sent on both the 'Tickets Filed' and viewable static ticket pages*

User Story 3: Delete Previous Tickets

As a user, I should be able to delete any of my previous tickets sent and remove them from my 'Tickets Filed' page.

Trigger	When a user accesses individual tickets from their 'Tickets Filed' page, this workflow process is initiated via a 'Delete Ticket' button
Details	This functionality exists on the premise that tickets become outdated or have been satisfied for the user. The user (as well as administrators / team members) can view prior tickets filed and stored, but they have the option to delete the tickets
Access	Authenticated users who are logged in; administrators (team members) within the database for tickets generated
Required Field(s)	Ticket Identification Number (Prepopulated, Delete and Confirm) Ticket Date and Time submitted (Prepopulated, Delete and Confirm) Type of Assistance Required (Prepopulated, Value(s) selected from Dropdown List) (Delete and Confirm) Severity of Issue (Prepopulated, INT 1-5) (Delete and Confirm) Description of Issue (Prepopulated, String, VARCHAR 256) (Delete and Confirm)
Input Field(s)	N/A
Action(s)	Prior to deletion, clicking on the button generates a popup message asking whether the user confirms the deletion for that ticket. If 'Yes' is selected, the viewable static ticket page is removed from the web application in addition to the prevalent information stored in the database. Depending on whether all tickets have been deleted or not, either navigate to an empty form when no tickets exist or back to the 'Tickets Filed' page when more than one ticket exists
Exception(s)	Only tickets pertaining to the user who filed the ticket should be able to delete them based on view access. Administrators, however, can delete any populated tickets in the database, though it is imperative that all tickets regardless of deletion should always be retrievable within a separate table collecting all users' tickets
Test Criteria	*-User first attempts to delete the ticket and renavigate with the back button on the browser back to the deleted ticket. If successfully coded, the back button should not repopulate the static display form or the prevalent database information for that deleted ticket* *-If all tickets are deleted, test to see if the application automatically regenerates the form on the 'Support' button on the navigation bar and removes the 'Tickets Filed' link on the dropdown list* *-Once the 'Delete Ticket' button is confirmed, the subsequent information for that deleted ticket should be removed from the 'Tickets Filed' page*

Epic 7: Classroom Availability with Maps

Functionality: As we mentioned before, we plan on adding classroom functionality, allowing students/users to see classroom availability as well as rooms with specific equipment users might need/want.

Note: In our reviews, we perceived safety concerns about where meetings would take place. Meetings will not take place in any location that we deem to be unsafe for users. For example no meetings will take place inside a user's house. We will only allow users to select from a predetermined list of acceptable locations like a library or classroom on campus. This is to ensure the safety of all of our users. The maps feature will allow users to visually see which locations or rooms are available and the quantity of available rooms. Special rooms with extra amenities will be marked, and users will be able to filter out by room amenities. Lastly, users will be able to select a location and submit a reservation request.

- Check For Available Rooms
- Selected Room Booking
- Edit Meeting Location Details
- Delete Meeting Location Details

User Story 1: Check For Available Rooms

As a user hosting a meeting, I want to check if there are any available rooms to host a study session.

Trigger	A button labeled ‘Check For Available Rooms’ at the bottom of the screen on the ‘Host A Meeting’ web page
Details	Once a user has finalized their meeting information they can select the ‘Check for Available Rooms’ button. It will show them a map and a list of rooms that are available.
Access	Only users hosting a meeting will have access to this button
Required Field(s)	See Input Fields
Input Field(s)	Location (Map Integration from Google) Date (Date, String with CHAR 8) Time (Time, Javascript for Slider) Extra amenities (Multi-Value Dropdown List)
Action(s)	When a user clicks the ‘Check For Available Rooms’ button, they are redirected to a new page with a map and a list of available rooms. They can select and reserve any rooms that are available.
Exception(s)	If a user tries to schedule two meetings at the same time, it will pop up with an error screen and not allow them to create a second meeting. If a user tries to schedule a meeting for a room that has no availability, it will pop up an error screen and let them know there are no available rooms at that location.
Test Criteria	*-Data comparison between users' request to reserve a room and the room availability.* *-Data comparison between all meeting times for a user and the current scheduled time they are submitting.*

User Story 2: Selected Room Booking

As a user, I want to be able to select a room and book it at specific times.

Trigger	A button labeled ‘Select This Room’ located underneath the room’s details
Details	Once they have located a room they might want to pull up the room in a 3d? and/or 3d? Detail. It also has a display that lists amenities in the room as well as maximum capacity
Access	Only users creating the meeting will have access to this feature.
Required Field(s)	Classroom infrastructure and timeframes from the known database.
Input Field(s)	Possibly a book from (MM/DD/YYYY HH:MM TT) to (HH:MM TT) Maybe even requests for certain amenities to be made available? (I know at my job we can help in certain situations where they might request a room to be handled for an 'event') book_time (date(YYY/MM/DD, HH:MM TT)) Host Last Name* (string autofill via google?, VARCHAR 20 if manual) Host email* (string autofill via google?, VARCHAR 50 if manual) Phone # (string, CHAR 14) Class / Subject* (string) Location* (string, viewable via google maps) Time/date* (MM/DD/YYYY HH:MM TT) Purpose (string) Description (string)
Action(s)	Once a room is selected from a list or display, the room is pulled from view with a close or a booking option.
Exception(s)	If someone had booked it as they were booking it, booked at invalid times, or with unavailable amenities. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*Room, datetime, and amenities are validated during the Book button action to see if all are available. Also refers to their attending and other hosted meetings, where it might toss them back into the editor if they have conflicting schedules.*

User Story 3: Edit Meeting Location Details

As a user, I should be able to change the details of a meeting that I am hosting, including the location.

Trigger	A button located in the top right corner labeled as ‘Edit Meeting’ on the individual meeting’s home page.
Details	Once a user has created a meeting they should be able to go back and edit the details if they need to change the time, location, or other details.
Access	Private only to users who are the host of a meeting
Required Field(s)	book_time from (MM/DD/YYYY HH:MM TT) to (HH:MM TT) Host Last Name* (string autofill via google?, VARCHAR 20 if manual) Host email* (string autofill via google?, VARCHAR 50 if manual) Phone # (string, CHAR 14) Class / Subject* (string) Location* (string, viewable via google maps) Time/date* (MM/DD/YYYY HH:MM TT) Purpose (string) Description (string)
Input Field(s)	book_time from (MM/DD/YYYY HH:MM TT) to (HH:MM TT) Host Last Name* (string autofill via google?, VARCHAR 20 if manual) Host email* (string autofill via google?, VARCHAR 50 if manual) Phone # (string, CHAR 14) Class / Subject* (string) Location* (string, viewable via google maps) Time/date* (MM/DD/YYYY HH:MM TT) Purpose (string) Description (string)
Action(s)	Redirected to an editable version of the meeting’s home page
Exception(s)	If the user is not the host of the meeting they can not edit the details. If any of the new updated fields are invalid the meeting cannot be updated. ←(not sure if this is for different user story) If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	Test to see if the user is the host of the meeting Test to see if all data input fields are valid

User Story 4: Delete Meeting Location Details

As a user, I should be able to delete the details of a meeting that I am hosting, including the location.

Trigger	After accessing a button located in the top right corner labeled as ‘Edit Meeting’ on the individual meeting’s home page, this workflow process is initialized by scrolling down the relevant meeting location details and clicking a 'Delete Meeting Location' details button. A popup box asks the user for confirmation prior to removing the meeting location details
Details	Once a user has created a meeting, they should be able to completely delete the details if they need to remove the time, location, or other details. This deletion should apply for all members associated with the meeting and meeting location details
Access	Private only to users who are the host of a meeting
Required Field(s)	N/A
Input Field(s)	Delete Meeting Location Details and Confirm Deletion
Action(s)	Redirected back to the 'Host a Meeting' page for the group of the deleted meeting location details
Exception(s)	If the user is not the host of the meeting, they can not delete the details. Hosts should not be able to delete meeting location details for other meetings that they are not hosting
Test Criteria	*-Test to see if the user is the host of the meeting within the database* *-Test to see if all data fields within the database are removed as well as the frontend page* *-Attempt to delete the meeting location details and renavigate to the previous page containing the editable fields with the browser's back button. If successfully coded, the page should not be repopulated nor should the database associated for that meeting*

Epic 8: Notifications With Group Calendars

Functionality: There will be a personal and group calendar feature. Users will be able to switch between the different calendars and put notifications on their respective calendars.

- Select Calendar
- Add Notifications
- View Notifications
- Delete Notifications

User Story 1: Select Calendar

A user is able to select a specific date notifications are posted to and the specific calendar account it is bound to, such that if a user wants the notification to be bound to their student accounts or personal accounts.

Trigger	Selecting the calendar option in the navigation bar or sidebar a user is redirected to the calendar page wherein they can choose which calendar they want to edit / access via a drop down or similar intractable.
Details	Once pulled up a calendar is pulled up that is able to display a single day, week, or month at the user's discretion as well as being able to view the events on the calendar display.
Access	Either can be a personal calendar, or a group's specified calendar for a planned event or meeting.
Required Field(s)	IU username, or specific email (to access a personal or group calendar list) Group_id varchar / int, as a group can be serialized as a code or a number string. Or calendar_id which turns out to be similar to group_id, it just depends on how we want to store the data in order to serve and display it
Input Field(s)	There shouldn't be any input fields here, unless a dropdown bar or similar counts as an input field in order to display the calendar contents.
Action(s)	From here a user is able to select their view such as mentioned before between, day, week, or month, as well as being able to select the specific meetings per date to view from this page. And if they are the host or creator of the meeting they are able to modify the notification / meeting. Via the button in the top right.
Exception(s)	If there is no calendar account attached to the account, a page is displayed to include an existing calendar or to create one.
Test Criteria	*-Attempting to access the calendar tab results in a page that displays create calendar or connect a calendar to this account.* *-Calendars can only be attached to the account if the calendar is bound to the account via the IU gmail system, or by the group including their email in the emailing list (maybe)*

User Story 2: Add Notifications

As a user, I should be able to add notifications listed on the calendar view for either the individual or group calendar.

Trigger	Selecting the calendar option in the navigation bar or sidebar a user is redirected to the calendar page. From here, a user can either click on an 'Add Notification' button or an individual calendar day to specify the notification for that day
Details	A user is able to add notifications from either the notification menu with the 'Add Notification' button or a specific day on the calendar integrated from Google. Notifications can be added for the individual calendar, a group calendar, or both
Access	Any authenticated user who has initialized the profile
Required Field(s)	See Input Fields
Input Field(s)	Notification Title (String VARCHAR 50) Notification for (Multi-Value Dropdown List based on individual and group calendars) Date (DATE, could be generated based on the Google Calendar API)
Action(s)	Once a user adds a notification and saves, the application should automatically navigate to the individual calendar integrated. Also, a green bar along the top should note that 'Notification Successfully Added' appears
Exception(s)	Do not save invalid data types such as the notification title field. If a user attempts to leave the page without saving any information input, indicate a pop-up alert message noting that changes will be lost.
Test Criteria	*-If the user attempts to input another data type in the notification title, ensure that a note is shown to the user illustrating this error.* *-Find the day and calendar(s) where the notification additions should be populated based on the third user story; confirm that the data input into the addition matches with the viewable information*

User Story 3: View Notifications

As a user, I should be able to view individual notifications listed on the calendar view for either the individual or group calendar.

Trigger	Selecting the calendar option in the navigation bar or sidebar a user is redirected to the calendar page. From here, a user can either click on a 'Notifications' button or an individual calendar day to view the notification for that day
Details	A user is able to view notifications from either the notification menu with the 'Notifications' button or a specific day on the calendar integrated from Google. Notifications populating the correct groups can be viewed for the individual calendar, a group calendar, or both
Access	Any authenticated user who has calendar notifications
Required Field(s)	Notification Title (String VARCHAR 50) Notification for (Multi-Value Dropdown List based on individual and group calendars) Date (DATE, could be generated based on the Google Calendar API)
Input Field(s)	N/A
Action(s)	Once a user adds a notification and saves, the application should automatically navigate to the individual calendar integrated. Also, a green bar along the top should note that 'Notification Successfully Added' appears
Exception(s)	A user attempts to navigate away from the notification form while fields are still populated and the notification has not yet been saved; a popup message appears asking for confirmation before the user leaves the page and loses the unsaved additions Users should not be able to access group notifications that they are not a part of
Test Criteria	*-Find the day and calendar(s) where the notification additions should be populated; confirm that the data input into the addition matches with the viewable information*

User Story 4: Delete Notifications

As a user, I should be able to delete notifications listed on the calendar view for either the individual or group calendar.

Trigger	Selecting the calendar option in the navigation bar or sidebar a user is redirected to the calendar page. From here a user can click on a 'Notifications' button or individual days with notifications present. A list of new and previous notifications will appear.
Details	A user is able to delete notifications from the notification menu. A dropdown of notifications should display a user can individually delete each notification. No warning of deletion.
Access	Anybody that is logged in and has calendar notifications
Required Field(s)	Notification (Dropdown List of VARCHAR 250)
Input Field(s)	N/A
Action(s)	User deletes notification therefore notification is no longer in the dropdown menu of notifications.
Exception(s)	The user has no notifications Users should not be able to delete group notifications that they are not a part of
Test Criteria	*User selects the calendar, then selects notifications then selects delete. When done the user should no longer see the notification they deleted.*

Epic 9: Login and Registration

Functionality: In terms of authentication, our web application will connect with the IU login page as an integration; because users are logging in rather than technically creating an account, we will "initialize" the user the first time they login to the web application, which will then associate these credentials with the user identification within the database so that their information saves for any subsequent login attempts. Deleting the account essentially just removes their user identification from the database as well as any associated credentials, but the IU login will always be the mechanism for maintaining all aspects of the Epic.

- Register/Initialize User Account via IU
- Login to User Account via IU
- Delete User Account

User Story 1: Register/Initialize User Account via IU

As a user, I should be able to initialize my account on the web application once I login with my IU credentials for the first time.

Trigger	To initiate this workflow process, users can click on a 'Login through IU' button on the introductory page, which initializes user identification for a new account based on these IU credentials
Details	When first navigating to the web application for the first time, the user should see a fairly static page indicating what the application is used for and how the user can begin utilizing the application by initializing through the IU login page. By initializing their account and authenticating that they are indeed an IU student, they can begin using the functionalities of this application. As noted, narrowing users to only IU students assisted with our map and course integrations
Access	All users with an authenticated IU account
Required Field(s)	See Input Fields
Input Field(s)	IU Username (String, VARCHAR, integration functionality based on authentication within the IU system) IU Password (String, VARCHAR, integration functionality based on authentication within the IU system; need to ensure encryption is present) DUO Two-Factor Authentication (integration functionality based on authentication within the IU system)
Action(s)	After initializing their account for the first time, navigate to an empty 'User Profile' page, where new users fill out a form and save these additions. The credentials, a new user ID, and these profile additions should be added within the database
Exception(s)	If the account has been initialized before, simply login to the user's account and output all the relevant input data and/or other saved features already populated
Test Criteria	*-Attempt to login using both valid and nonvalid IU credentials. Only valid credentials should have access to our web application* *-Test to see if the account is initialized in the database when the user fills out the required fields in the user profile for the first time* *-Delete an account and reinitialize it to see if the database entries have been deleted for that account. Should navigate to the empty user profile form when deleted and reinitialized*

User Story 2: Login to User Account via IU

As a user, I should be able to login to my account in the web application and still have access to any inputs I added for my account, including user profile, matches, courses, study groups, and meeting availability.

Trigger	To initiate this workflow process, users can click on a 'Login through IU' button on the introductory page, which searches the user identification for an existing account based on these IU credentials. We should also try to load the page already logged in based on the 'Remember Me' conditions for Duo and IU
Details	After initializing their account for the first time and authenticating that they are indeed IU students, they can use all the functionalities within this application on any subsequent accesses. As noted before, narrowing users to only IU students assisted with our map and course integrations
Access	All users with an authenticated IU account
Required Field(s)	See Input Fields
Input Field(s)	IU Username (String, VARCHAR, integration functionality based on authentication within the IU system) IU Password (String, VARCHAR, integration functionality based on authentication within the IU system; need to ensure encryption is present) Duo Two-Factor Authentication (integration functionality based on authentication within the IU system)
Action(s)	Once logged in, navigate to the 'Home' page on the web application, which should contain previous information populated in the database. For now, we are still determining how we want to structure the UI of our homepage, but we will likely contain separate containers pertaining to the user's courses, study groups, and potentially other simplified information sections such as matches or calendar notifications in a list format
Exception(s)	If the account has not been initialized before, complete the following workflow for the first user story, where the user is navigated to the user profile for filling out the required fields. The credentials, a new user ID, and these profile additions should now be added within the database
Test Criteria	*-Attempt to login using both valid and nonvalid IU credentials. Only valid credentials should have access to our web application* *-Test to see that all information generated for the user identification based on the logins is reflected on the frontend pages (Ex. User profile, compatibility questionnaire)*

User Story 3: Delete User Account

As a user, I should be able to delete all the information associated with my account so that if I attempt to login again with my IU credentials again, everything will be reinitialized.

Trigger	On the 'User Profile,' include another small navigation bar that must be clicked or hovered over before displaying a 'Delete Account' button. Clicking this button includes a popup message asking the user for confirmation, which then begins the process of deleting the user account
Details	After an account has been initialized, the user can choose to delete it at any time, which will remove any user identification and information tied to that account. Users can thereafter decide to create a new account with their same IU credentials, but all previously deleted information based on this functionality will be wiped from the database
Access	All authenticated users who are logged in; must obviously have an account created already
Required Field(s)	N/A
Input Field(s)	Delete User Account and Confirm
Action(s)	As noted, clicking and confirming the deletion will wipe out any populated information in the database associated with that account. The user is then redirected to the introductory page containing the static information about the webpage as well as the ability to create a new account again if desired
Exception(s)	Unless logged in using the IU credentials, users should not be able to delete other users' accounts. Moreover, if there is clear violations being tracked within an account, such as attempted code injection within the user forms, then the administrators / team members should also be able to deleted the account
Test Criteria	*-User first attempts to delete their account and renavigate with the back button on the browser back to the deleted account. If successfully coded, the back button should not repopulate the webpages or the prevalent database information for that deleted account* *-All populated information should be deleted and untraceable in the database since the account does not technically exist anymore* *-Test that deleted accounts do not leak into and delete other information that is pervasive for other users. For example, deleting an account should succinctly delete a user from the courses they listed on the profile, but the function should not delete the same courses themselves for other users*

Responsible Computing Statement

Given how the essence of sociotechnical systems, by definition, intertwines interdependencies between the social and technological processes that shape these systems, where one is not inherently deterministic over another but rather works in correspondence together as one system, this mutual shaping perspective helps us apply this theory when we consider the implications of our own Capstone project. For example, our system will be directly interacting with real users, and for this reason, we need to propagate a system using a rudimentary design philosophy in terms of the intention of the application yet ensure that it scrutinizes a deeper set of sociotechnical questions so that there exists a deliberate continuity in interaction between the user and the system. If users expect our system to only provide study groups functionalities and not obfuscate other technical objectives such as personal data collection, then we can only realize this expectation by holistically designing our system so that, as mentioned, our system relies on the contributions of the users but does so in a way that provides them the same or a similar level of agency with the system itself.

To accomplish the methods in which we intend to maintain a sociotechnical system, we have enumerated social considerations on the subsequent pages, where we describe aspects pertaining to our social design philosophy, practices we employ for data collection and usage, and other factors such as harms or abuses that could result from the system. The first two sections, *Stakeholders and Design* and *Governance, Oversight, and Codes of Conduct*, directly associate with one another since it is imperative that we can consider the overarching stakeholders in order to codify an equitable set of rules that we intend to abide by. The latter two sections, *Data* and *Transparency*, also relate with each other regarding our sociotechnical approach since we need to understand the fundamental level of data that we are collecting so that we can accentuate this for our target users; if we abuse the data we collect outside of the rudimentary design philosophy, as mentioned, then there could be serious ramifications from the perspective of users since they had personally placed their trust in the equity of our system when choosing to interact with it. Lastly, the final three sections, *Harm*, *Limitations*, and *Abuse*, relate together well since a system's limitations generally exposes its potential measure of harm inflicted on users in addition to avenues where malicious users can abuse the system, either in relation to the technical vulnerabilities like code inject or social threats like threats within our meeting functionality. Overall, by delving into these sections with more granularity, we can better understand how and assure that our application upholds a strong predication for high ethical standards.

Stakeholders and Design

Because we want our integrations in terms of courses and our map to be specific for the IU-Bloomington campus, this will ultimately limit the scope of our primary stakeholders to be Bloomington campus students, which we can refine further to students who want to increase their social opportunities or bolster their success in difficult courses by utilizing study groups. To mitigate any exclusive design practices for our student base, therefore, we avoid storing any personal student data that identifies them based on demographics or other factors, which is

important from our design standpoint highlighted in subsequent sections since we will be formulating a compatibility questionnaire that identifies potential matches between students. Most importantly, if we had expanded our student base to all college students as a broad encapsulation for our application, then we would have conceptualized our application in a different way that we did not feel comfortable consummating, so we decided upon IU-Bloomington to be able to moderate our course database as well as map integration.

We also intend to incorporate the IU login integration, which we would theoretically be able to tailor our profiles around since users would already have their classes matched within the application, but because we as a team will not have access to the database containing this information stored by the IU administration, we can assume that a stakeholder under real-life circumstances would be IU administrators since they would need to uphold security practices for our application. This resides on the premise that our application would be implemented onto One.IU and therefore be managed by IU administrators and other personnel, but we can only enumerate this here as an example of a real-life scenario.

One functionality within our application will be classroom booking for facilitating meetings at locations on the Bloomington campus, and since many different types of groups necessitate these classrooms for their own purposes such as labs, clubs, or actual courses, these groups would be considered indirect stakeholders since they could be consequently affected if, for instance, a study group can book a classroom during the same time as a club meeting occurring within this classroom; this could insinuate some form of conflict if this classroom is the only one, for example, with access to a projector in the entire building. In turn, we need to consider the possibility of already-booked rooms as another example of a real-life scenario, which may be difficult to internalize in practice but is something that we will actively try to contemplate for our application.

Finally, many types of security personnel need to maintain the security of buildings on campus, which means that we need to consider them as stakeholders with the assumption again that IU would be managing our application through One.IU. Although this concern will be further exacerbated in the *Abuse* section of this document, malicious users could always pose as other people in addition to either threaten students through our application or cause damages to one of the rooms booked. Therefore, although we as group will likewise not have access to this information stored within an IU database and will try to minimize the potential for abuse within our application, under a real-world scenario, we would need to consider some form of quality assurance so that security personnel can document who abused the application with malicious intent in mind.

Governance, Oversight, and Codes of Conduct

Since our application is going to be created for aforementioned Indiana University students and used on and around Bloomington campus, we will considerably abide by the ethics and laws of the city of Bloomington in addition to Indiana University campus. Oversight for our system would include a ticket system that allows students to send in issues with the site; this

can include both technical issues and student profile or study group issues. This form of oversight trusts users to obey the honor system and report any user trying to cause harm on the site. As developers, we will abide by the code of conduct for creating a safe environment for users to socialize and create study groups, and we therefore need to honor the Indiana University Division of Student Affairs Cheating and Plagiarism Policy, Student Organization Misconduct Policy, and Sexual Misconduct Policy. It is important as creators, programmers, and designers that we shape our project to limit the overarching amount of harm, limitations, and abuse to users on our site in order to create a thriving community that allows users to achieve their studying goals.

Data

Data collected for the website is information non-personal to users. Most information is generally only "class" related and only collects user volunteered data through the compatibility questionnaire, which is used for nothing else apart from group matching and has no other purposes (again this is non-personal). Depending on how we are able to aggregate classes available and taken by users, we are able to either grab that information from IU itself, or by user input again, seeing as our application aims to be (ideally) integrated with One.IU, we would be able to grab information from student accounts to easily dynamically fill their account class information from the IUJ database, but that is not feasible, so users would select from a list of classes created by our application in order to be matched with the right study groups for those classes. Again, these two data types in our system, classes and the compatibility questionnaire, aim to be non-personal, and objective.

Transparency

For our application, we decided to keep specific parts of our information system transparent while other parts are hidden because the end users simply do not need access to them. For example, we decided that it is better to keep the purpose and values behind our system transparent to the user. This makes the user feel more comfortable knowing exactly what they are using and why it is doing what it is doing in terms of functionality. We don’t want users to question our applications intentions at all or think that is hiding valuable information. All users should be able to see that this application is meant to bring students together. On the other hand, it is not important for the user to see the decision making process behind our compatibility questionnaires. The data behind these questionnaires and how the results are ultimately determined could be misinterpreted by users. This would only cause confusion, and it could potentially cause harm to our end users. It is our goal to avoid this at all costs, so as a team, we decided it best to withhold this information.

Harm

Our application is based completely on interconnecting students with other students. With this can come great benefits and potential harm. As a group, we decided that harm can be defined as anything that causes an end user to have a negative experience, whether this is physical or mental. In order to reduce as much harm as possible, we created compatibility

questionnaires, which are tailored to match users with other students who have similar interests or study habits as themselves. We believe by using these questionnaires, we can eliminate users being stressed over finding a compatible study friend(s). Also, every user is given the ability to create their own study group or meeting. This ensures that every user can enjoy the same experience while utilizing our application. By using our application, users should and will be affected in positive ways through these social opportunities.

On the other hand, however, it is impossible to remove all potential harm. On an end-user level, there is the potential that some users didn't answer the compatibility questionnaire with full honesty. This could create situations where some users are not happy because they have been matched up with someone who is not compatible with them. This can also extend into study groups as well. In some cases, there might be some study groups that try to be exclusive or not welcoming to new members. With all of this, worst case scenario students would develop a distrust for other students on a community level. Therefore, users would discontinue using our application.

Limitations

The limitations of our application and its database is limited to the amount of data the application is actually able to accumulate. Depending on whether or not this application is integrated and officially used by IU, as an example, the application may have to make its own database of classes that students can take, or as another example, which classrooms are available, as building room information is not readily available to the public unless one personally enters the building. The information included in the database is limited depending on the integration level with IU itself, which therefore limits the officiality of the information used throughout application.

Abuse

As with any application, technical vulnerabilities can be abused if discovered by malicious actors, which can potentially affect other legitimate users in the process. In correspondence, it is our responsibility as the developers to find and alleviate these vulnerabilities, which we believe will mostly pertain to the various forms we implement for users in addition to the API integrations we aim to encompass within our application (Ex. library dependencies in APIs have seen rising instances of remote code execution through frontend processes). For the forms, we will include proper validation and sanitization methods applied from ITP 04 so that we can minimize the ways in which users can abuse the data entries, and these methods, in a sense, act as multilayered protection if users bypass the validation layer. In regard to library dependency vulnerabilities, we will import only the minimum number of packages we need from verified sources and if need be, receive approval from instructors directly in every instance where we are unsure about a library's source.

Other abuses befit our classroom booking functionality, where study groups may attempt to encumber the system for other users by either booking as many rooms as possible or

booking the same room consecutively once a meeting timeframe expires. In turn, we will establish a set of logical conditions to curtail these abuses so that, for instance, study groups can only realistically book about five classrooms per day as well as book the same classroom once per day. When receiving feedback from groups, they questioned the safety of classroom booking, which we will seek to pacify by only allowing designated locations on campus such as Luddy Hall or Wells Library and limiting the timeframe for booking to reasonable hours like 8:00am-8:00pm.

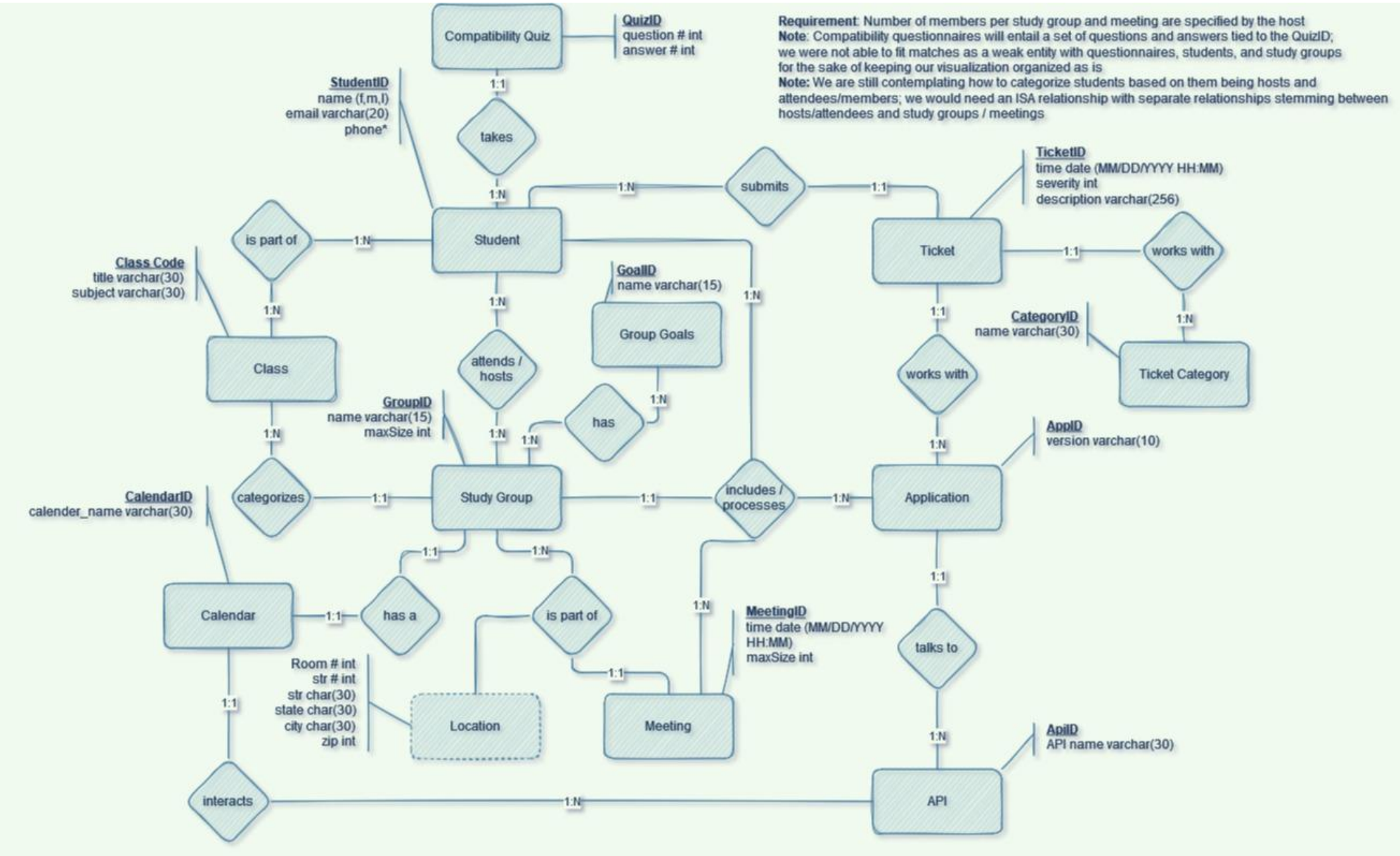
Lastly, malicious users may pretend to be other IU students through the user profile form as well as conduct other irregular activities like calendar notification or study group spam. To prevent the first abuse highlighted, we will have a ticket support system incorporated so that other users can report potential misuse of authentication from malicious users, which we can verify using our database and remove these users from our application; we can also manually analyze our database to flag users who have input considerably different names or email addresses in comparison to their IU usernames, though we will research ways to expedite this process if possible. To prevent the remaining abuses, we will also include logical conditions to flag whether users are attempting to join a vast number of study groups or sending out an irregular amount of spam through the calendar notifications, which we can likewise verify through our database and remove users if we deem them malicious.

Database Design

Using draw.io as introduced from I308, which allowed us to create an comprehensible semi-accurate ERD to our current design prototype, we believed this would be the best way to represent our current vision of our application when conceptualizing our ERD. Being semi-accurate, there are only a few facets we were unable to touch upon in the ERD, however we strongly believe that it is an effective visual representation of how our application and its database would work. For instance, we initially created and listed the strong and weak entities within our application (including the application itself), and from there, coalesced them together with their relationships with one another, progressively modifying its design as we went in order to keep it as simple and readable as much as we could keep it.

Addressing the semi-accurate nature of our ERD, we initially wanted to incorporate an 'ISA' relationship for our student entity; however we quickly realized that this was unfeasible as it would create a very complex and tangled ERD, which would defeat the purpose of creating an ERD in the first place. Therefore we have just included *students* as its own entity, as it both improves readability and comprehension of our ERD design.

As an effect of our assignment feedback, we changed the cardinality between student and meetings to a 1:N relationship, as students can have many meetings while a meeting can also have many students



Requirement: Number of members per study group and meeting are specified by the host
Note: Compatibility questionnaires will entail a set of questions and answers tied to the QuizID; we were not able to fit matches as a weak entity with questionnaires, students, and study groups for the sake of keeping our visualization organized as is
Note: We are still contemplating how to categorize students based on them being hosts and attendees/members; we would need an ISA relationship with separate relationships stemming between hosts/attendees and study groups / meetings

Wireframes

As aforementioned with our web application, we intend to facilitate and streamline the process by which IU students can utilize study groups to increase social opportunities with others in their courses as well as assist one another whenever they are struggling with course material. Because we intend to integrate our application into the IU system, we recognized the importance of familiarizing with the Rivet framework since this effort would help us alleviate the process of needing to idealize wireframes given that the Rivet framework already visualizes layout and component objects that we can incorporate into our own application. Moreover, Rivet helped us envisage more continuity with our wireframes since we did not need to predetermine our general design on top of attempting to keep it consistent with every user story since, as noted, many of the example layout pages already provided us a framework to build off of.

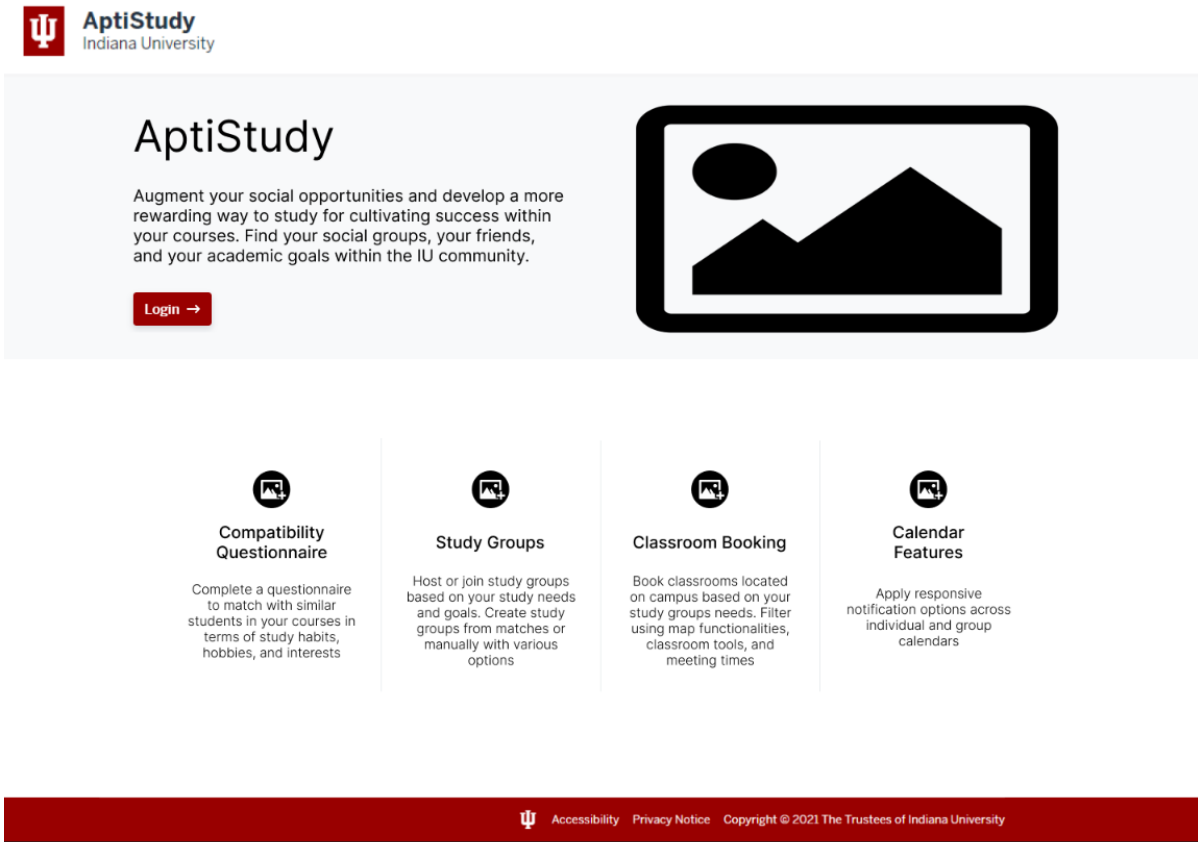
In regard to our specific wireframe approach, we accentuated how finishing the homepage first was imperative since many of our user stories would either directly or indirectly stem from our visualization of processes contiguous to this page; having created the homepage, we could then conceptualize and work on any immediate wireframes branching off our homepage since this approach would allow us to refine our wireframes over time in addition to minimize potential mistakes in the subsidiary branches we work finish later on in the creation process. Lastly, in terms of work distribution, we knew that some user stories pertaining to wireframes were more extensive than others, so we distributed our work based on the effort we predicted would be needed for each Epic and their relevant user stories.

To complete our wireframes, we collaborated using Figma since we were all familiar with this software from prior classes and already had sufficient knowledge of the UX component options that Figma offers like bevels, shadows, and transparencies. We also were aware that we could collaborate synchronously using Figma, and because wireframes may require a lot of iteration before they are refined to our liking, we wanted to be able to provide immediate and synchronous feedback with each other's work. This tool ultimately proved very useful for us when completing the wireframes.

For the process to create our wireframes, we separated pages based on a similar collection of user stories and included arrows when necessary to show the flow between parent and subsidiary wireframes; we also included our names on the pages we individually completed so that we could remain organized even with the 25 or so wireframes that we created. For consistency, we followed the Rivet framework regarding font sizes and styles, colors, and page layouts so that we could easily transcribe elements to other pages if need be; for example, after we incorporated boxes and elements for one form, we could transcribe many of these to other wireframes since some of our user stories necessitated forms. Finally, we made sure to avoid creating wireframes for every single flow or user story since one wireframe may encapsulate both the edit and delete functionalities for an Epic; we simply just copied one page and included an alert to indicate that the user is attempting to delete a destination designated for a user story.

Registration/Login Page

This acts as the introductory page where users can actually access the application by logging in using their IU account; depending on whether the user has created an account yet, the login button will flow to either the user creation page or homepage.



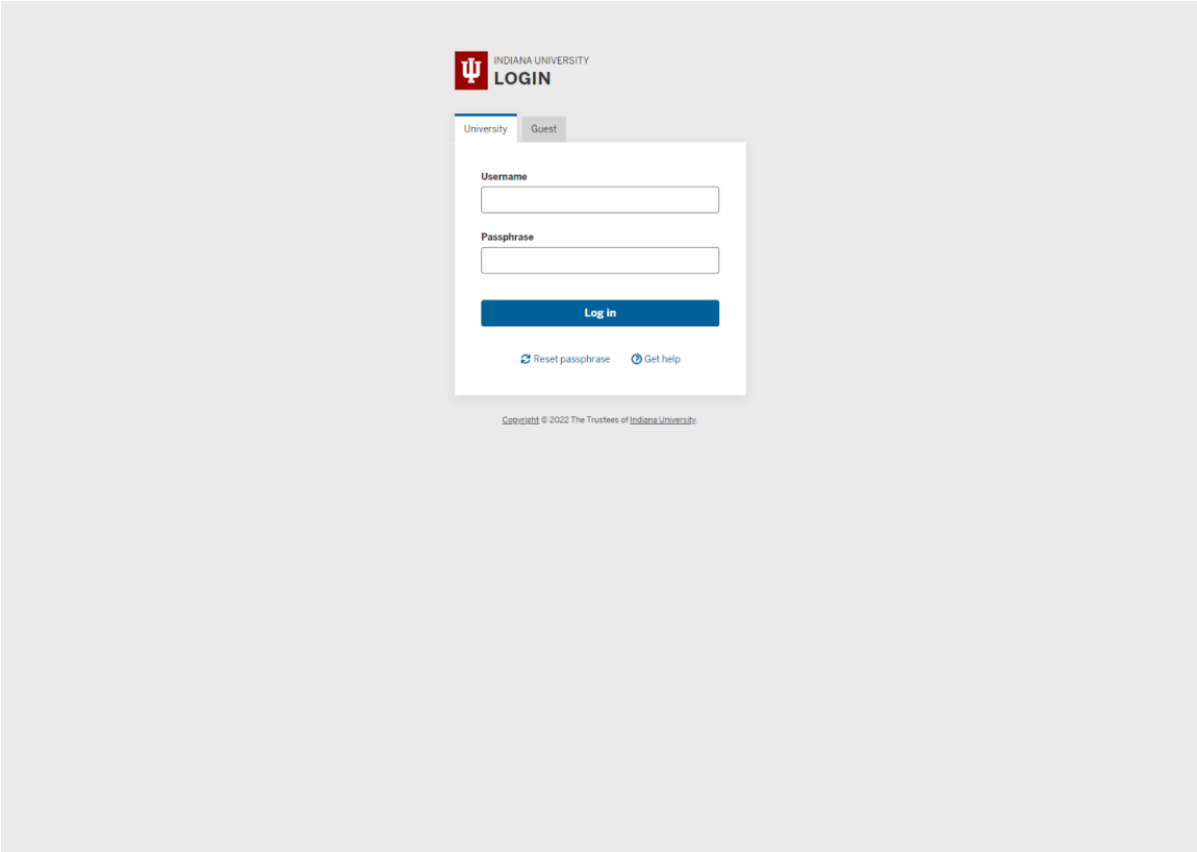
We did not want to populate too many elements onto this page, so we include information regarding what the application offers as well as the login functionality if the user intends to use the application. This page directly reflects the format of an initialization page used by the Rivet framework, where the four divisions illustrated below the login button can also be hoverable and expansive to improve our UX design.

Feedback Notes

No revisions were needed for this wireframe, yet participants from the UX methods noted that the design for this page obviously demonstrates that the application would be rendered and hosted by IU.

IU Login Integration Page

The following page simply demonstrates the IU login integration for our application, where only users with IU credentials and DUO two-factor authentication can access our application.



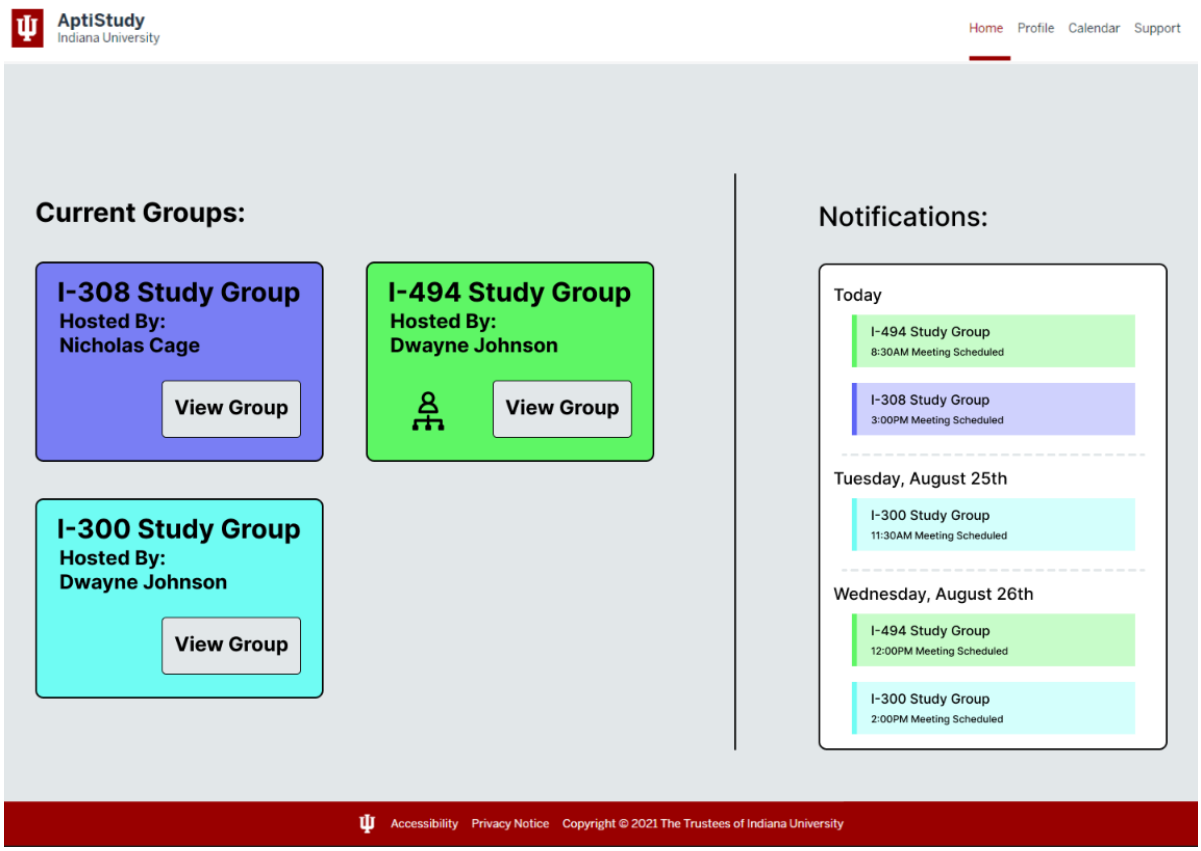
Since we did not want to replicate the parameters of this page on Figma, we simply took a screenshot and implemented it into our wireframes to demonstrate that login functionality will be required through IU; because our application is tailored to IU students, having IU login integration seems to be the most intuitive integration across all login options available.

Feedback Notes

No revisions were needed for this wireframe, yet participants from the UX methods noted that having this indicator page helped them imply that this application would only be accessible by students with IU credentials.

Home Page

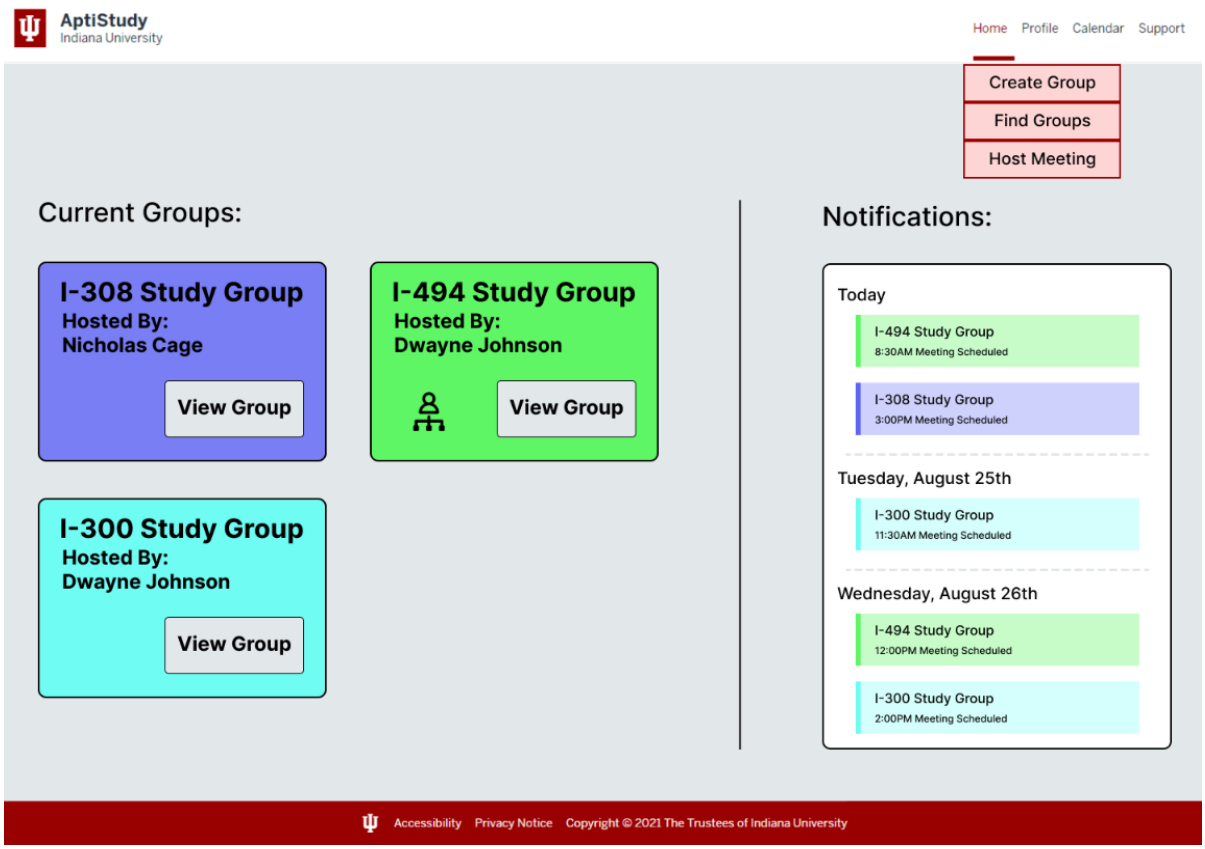
The home page displays current groups students have hosted or joined in addition to a set of more recent notifications that correspond to the calendrical function.



Because we are working with groups, we intend to differentiate them based on color as well as the host symbol specified in the green study group box since this is the primary information that users should have in regard to groups. Notifications also correspond to the color of the group, and we limited the value to five notifications so that users can quickly see upcoming meetings yet not feel overwhelmed since the calendrical functionality will accomplish this.

Home Page with Dropdown

This page shows the dropdown options present when users hover over the Home navbar link; these subsidiary navbar links pertain to functionalities with user groups as well as meetings.



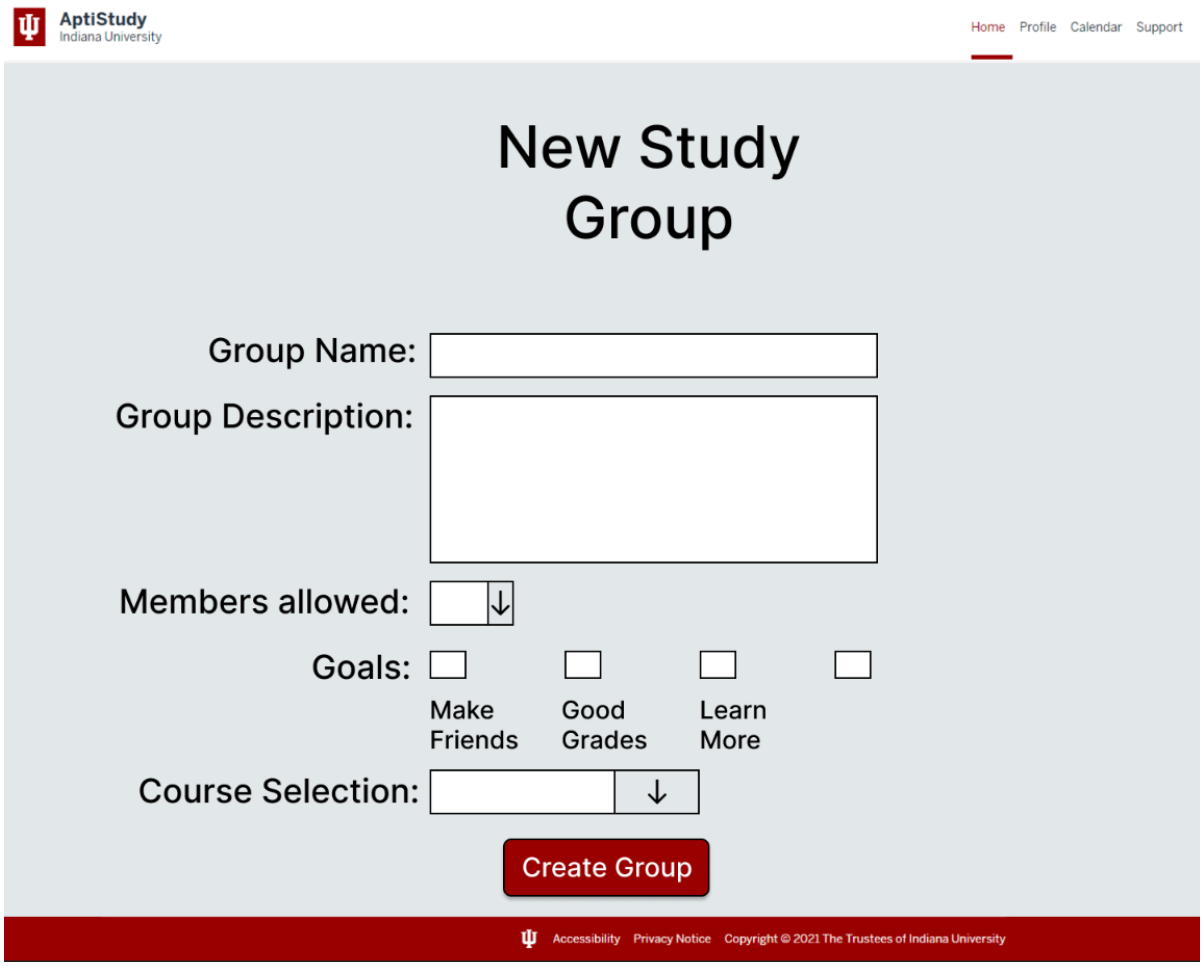
Rather than incorporate the buttons on the homepage, we initialize them using subsidiary navbar links so as to not overwhelm the user with the options they have available. Each link on the dropdown list pertains to the subsequent wireframes below.

Feedback Notes

One participant within a UX method indicated how the dropdown should rather be included with a new 'Groups' tab on the navbar given how these functions are conceptualized. However, we currently intend to keep this menu on the premise that these should be the first options that users see when navigating to the homepage for the first time, given that they will not be part of a study group yet and therefore not have any notifications. We may consider rearranging or adding an alternative homepage to make these three accentuated buttons that remain until the user joins a study group, but we still prefer this general layout compared to including another tab.

Create Study Group

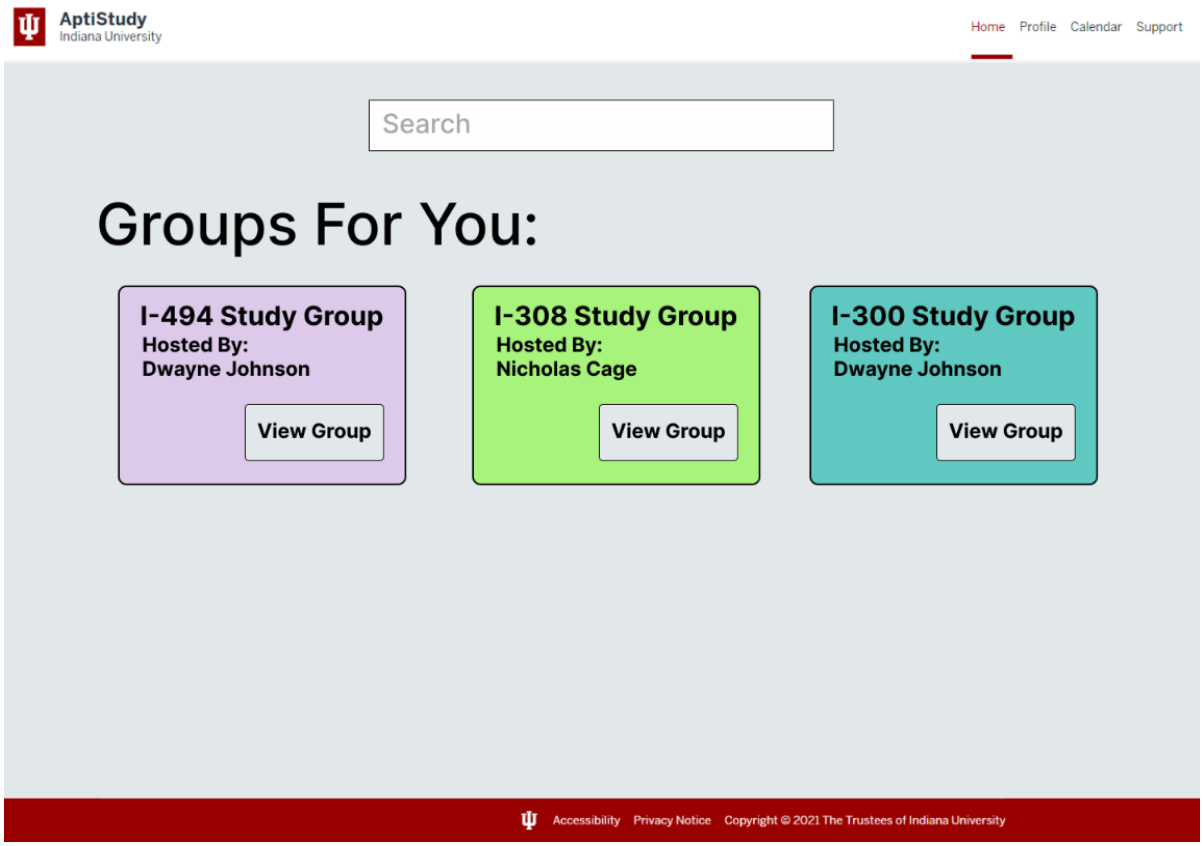
Users should be able to fill out a set of parameters for creating a new study group and restrict the amount of members for this particular study group; once users create and submit the study group, they are considered the host member with elevated privileges.



We wanted to include intuitive parameters for creating study groups, so users can ultimately specify study group names, descriptions, restrictions to the amount of members allowed, the set of predetermined or created goals, and course selections if they want to allow users from the same course to be able to join the study group whenever slots are open. The creation process is facilitated through a simple form that captures the user inputs and initializes them into a new study group on the web application.

View Recommended Study Groups

After accessing this page from the dropdown menu, users can see a predetermined set of recommended study groups based on a set of conditions like matchs' study group requests, study groups within the same courses, and so forth.



Although this page remains relatively static for the user outside of searching for study groups and viewing individual study groups, we want this page to automatically populate when a set of conditions are met with the users viewing the page. Therefore, study groups will populate when users finish other components of the application such as the compatibility questionnaire and user profile for course selection.

View Individual Study Group (General User View and Join)

Whenever an individual clicks on a viewable study group prior to joining, they can see more static information such as the meeting host, announcements made, and upcoming meetings for this particular study group.

For this page, users should have enough insight about the study group prior to determining if they want to join; for example, users may not want a stream of notifications for upcoming study group meetings if they are not able to attend the times illustrated, so this information assists with the process by which users choose to join or not join specific study groups.

View Individual Study Group (Host View)

This page illustrates the host view for a study group, which is only dependent on the study group elevated privileges and not individual study group meetings; different items are appended to the list on the right, where hosts can edit or delete groups if they choose to do so.

Apart from the previous wireframe, this page only differentiates by exemplifying the elevated privileges for users designated as hosts. Users can now see that they can either edit or delete a group as hosts, and these elevated functionalities should only be applicable for hosts since we do not want general members to be able to edit or delete study groups.

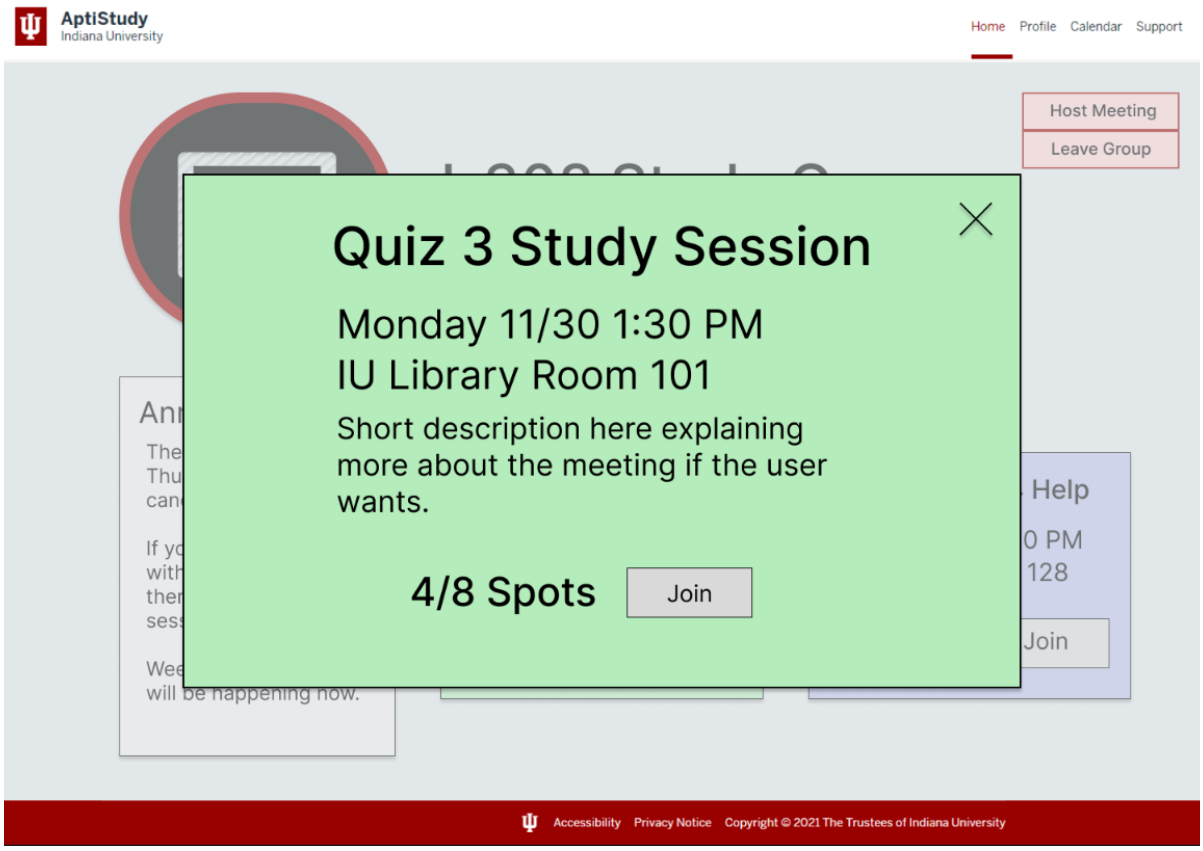
View Individual Study Group (General User View with Edit and Delete Button for Meeting Host)

If a user is the host for a meeting, regardless of if they are general members or hosts of a study group, then they should have elevated privileges for being able to edit the meeting specifications where necessary.

As illustrated with the edit button encapsulated in the yellow box (the yellow box will not be included on the actual webpage), users should have the option to edit meetings they are hosting based on the study group that a meeting pertains to; hosts can also choose to delete the meeting if necessary directly from this page by clicking the 'Delete' button.

View Joinable Study Group Meeting

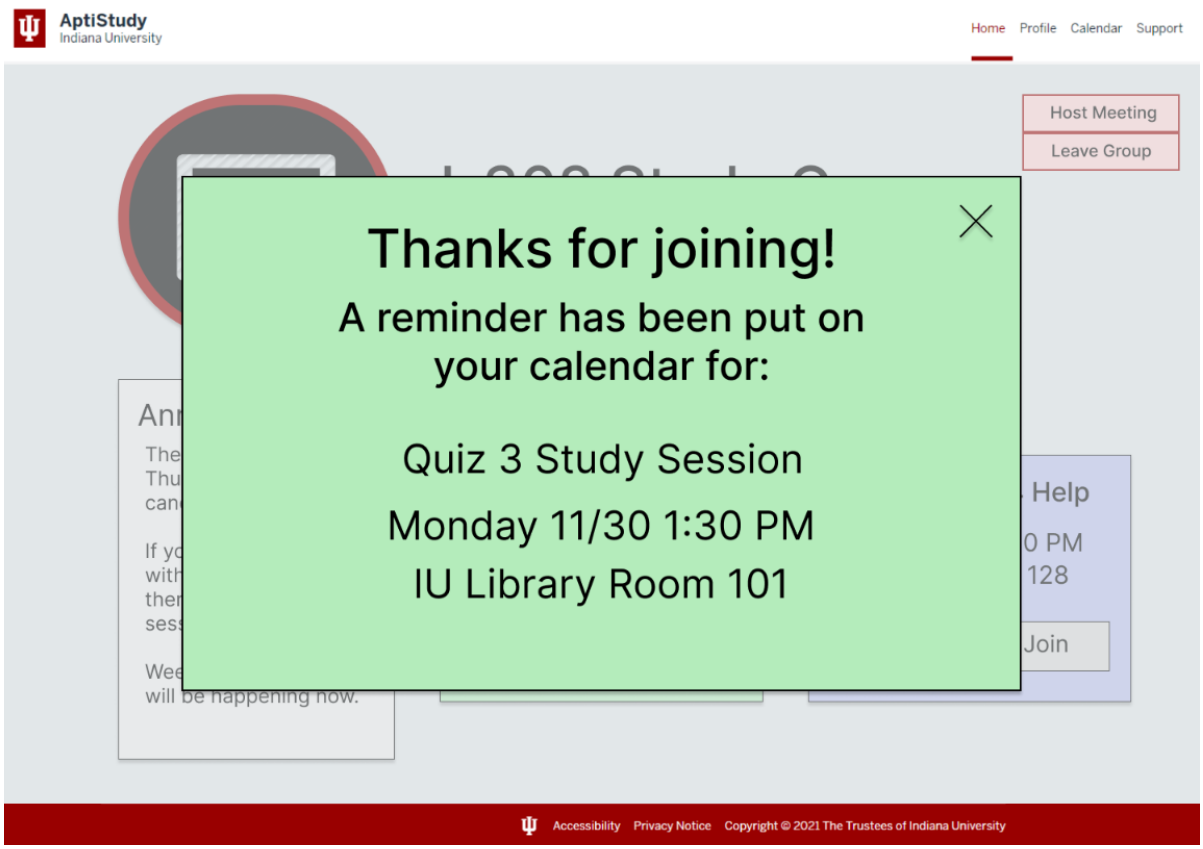
After users click on the 'View' button to see individual meeting pages, they should subsequently be able to see a box containing additional information about the meeting if they are inclined to gain insight before joining.



For this popup box, we simply wanted to include additional information including the meeting description and number of open spots available (assuming there are no spots available, the meeting will not be joinable in this context). Therefore, as with our other pages, we intentionally try to include intuitive parameters so as to not overwhelm the user with information yet provide enough that could alter their inclination to join a meeting.

Join Study Group Meeting

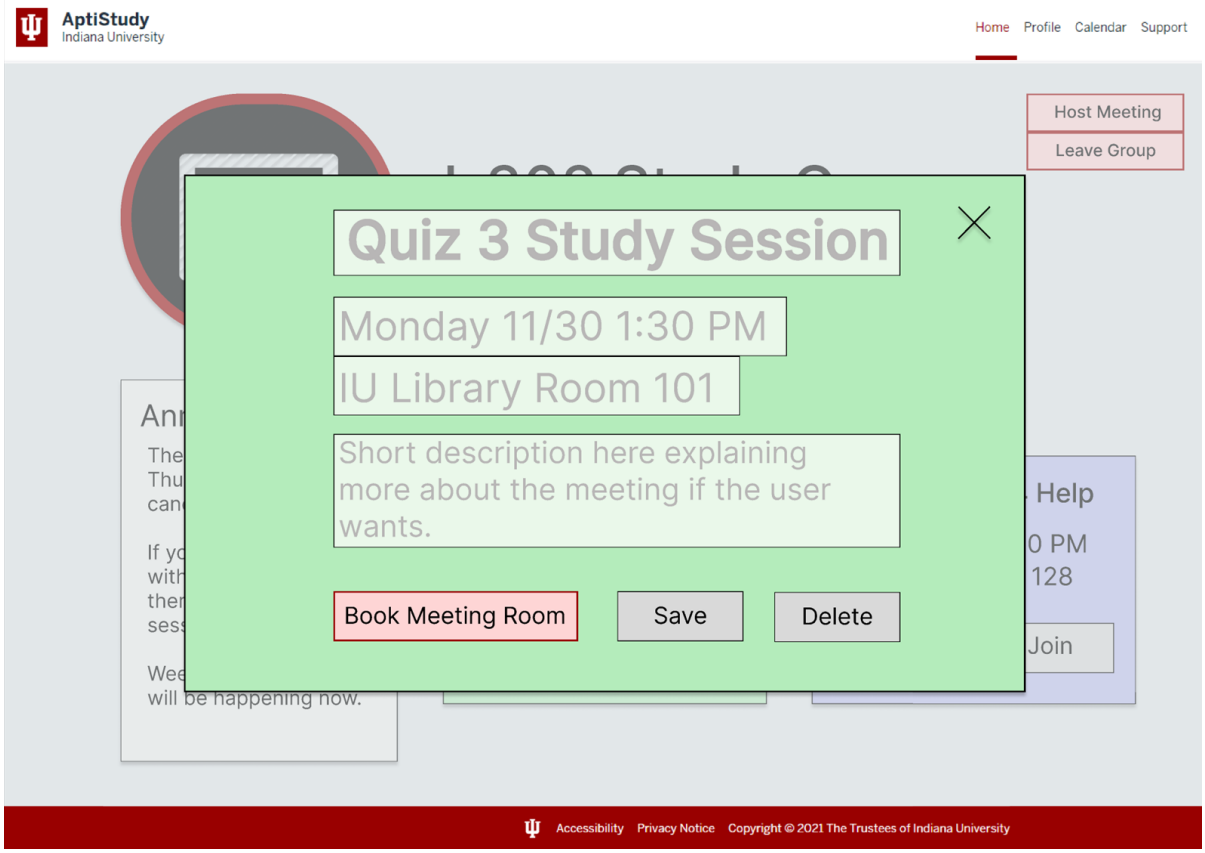
Once users join a study group meeting, they should receive confirmation that they were correctly initialized and accepted into the meeting.



Regardless of whether users join from the initial study group page or the viewable meeting page, we populate a static review of the parameters for the meeting so that the user can reiterate whether they are able to attend this meeting. For example, users may join a meeting but immediately realize after seeing the meeting time and location that they are not able to attend; therefore, they can easily leave the meeting after exiting the popup, redirecting them back to the initial study group page where the meeting functionality exists.

Host / Edit Study Group Meeting (after Selecting Edit Button)

If a user is a host of a meeting, they will be able to edit the details of the meeting. This includes everything from the title, time, location, and description of the meeting.



We decided as a group that it would be important for a user to edit the details of a meeting if plans changed. This screen shows what that exactly would look like. Every box that is edible has a transparent box around it. The user only has to click the box and they will be able to edit the contents. From there the user can either save the details, close out of the edit screen, or delete the group. If a user decides to close out of the screen they will be brought back to the study groups home page. On the other hand if they choose to save the details it will be displayed on the screen.

Revisions

There were not many changes made to this wireframe. The biggest change we made to this was including an additional button labeled ‘Book Meeting Room.’ We believe this will be more effective and convenient for users when editing group details. We also include Meeting on this button simply to help users subconsciously recognize what the room booking reservation applies to.

View Calendar with Notifications

This page acts as a monthly view calendar that allows individuals to see events upcoming or occurring within study groups.

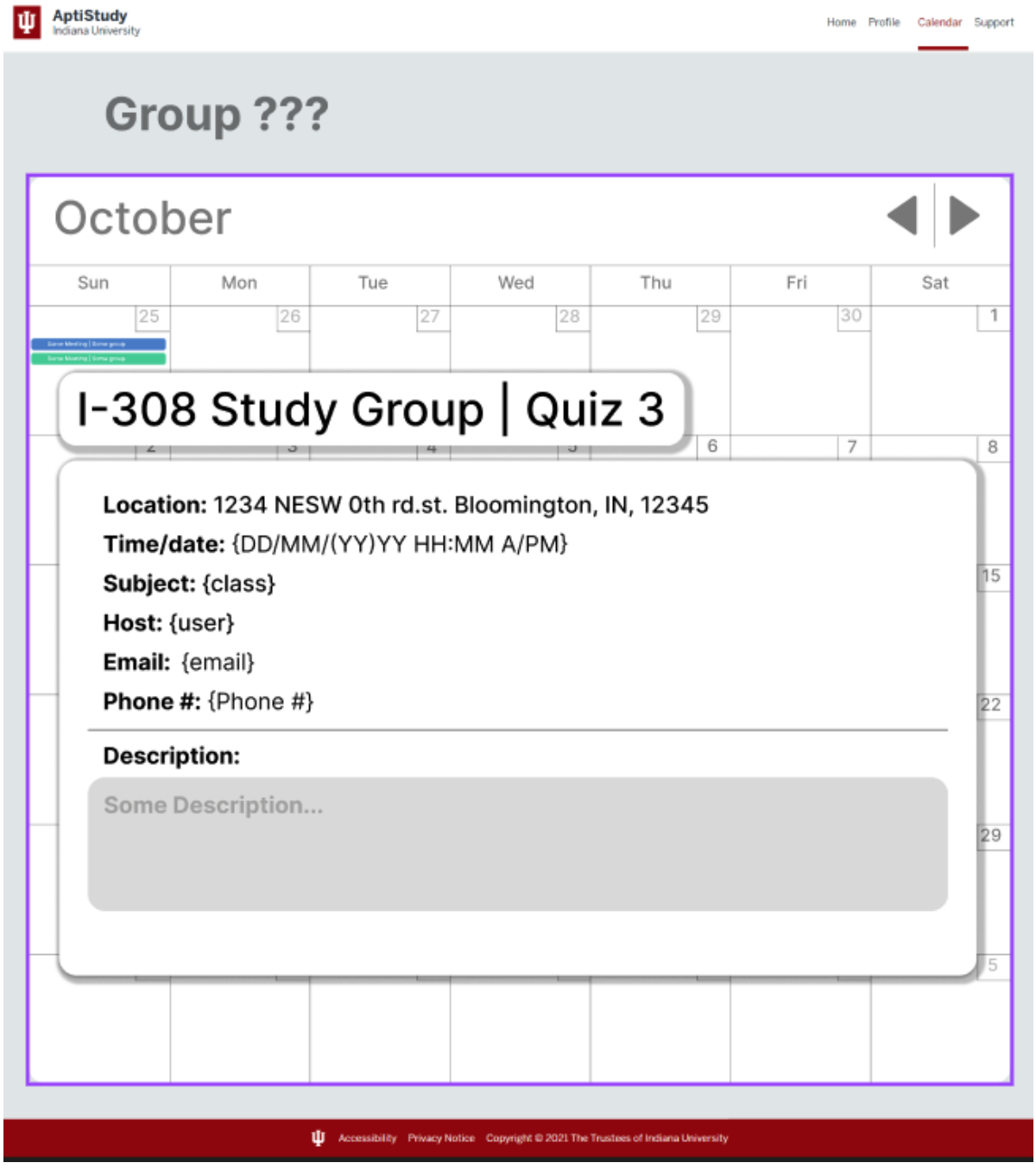
While there isn't much going on here in terms of visualization, this wireframe is a portrayal of our in-app calendar that illustrates a monthly calendar with specific group events that have been created and added as event notifications. Users are also able to add events from this page.

Revisions

Not much is revised here; the only minor/major change (depending on how you consider it) primarily only addresses quality of life changes, where we allow users to change between specific group calendars via a drop down menu added to this screen.

View Study Group Calendar Notification

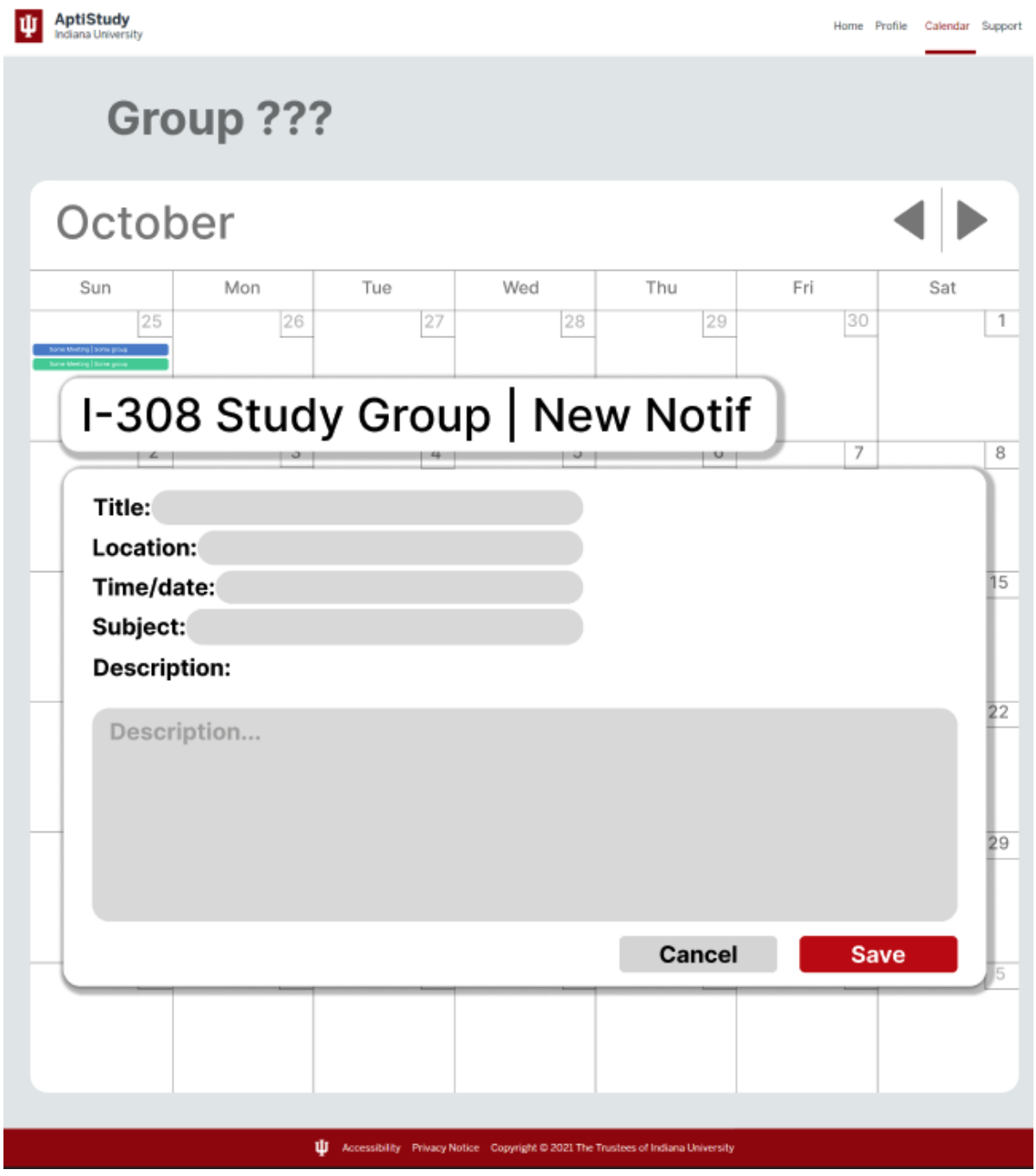
For each element contained within the calendar, users are able to view the individual details of the notification created.



This feature allows individuals to look at the details of a meeting specified by clicking on it and seeing its exact details as denoted by the meeting host.

Create Study Group Calendar Notification

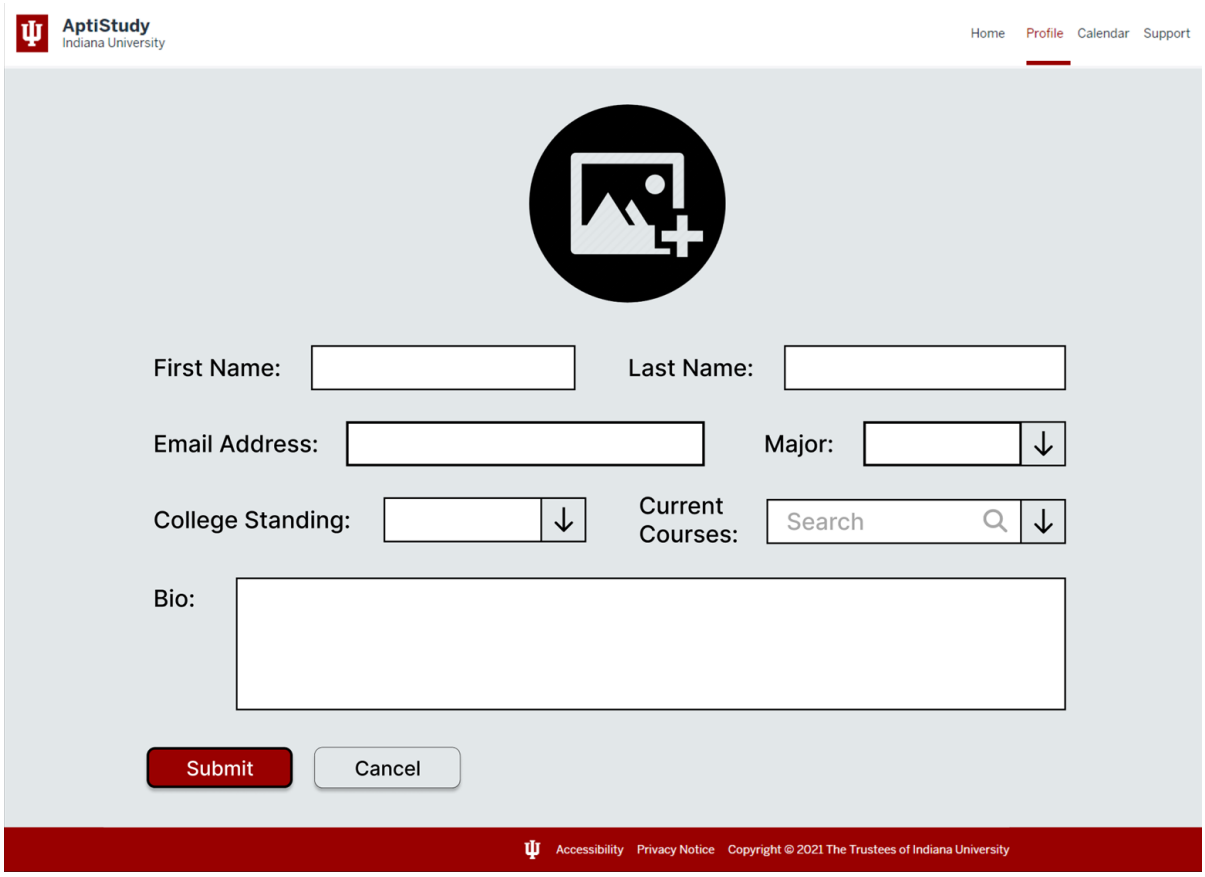
Here, users are able to create an event as well meetings from the calendars page.



This allows for individuals to seamlessly create a meeting on the calendars page without having to click to another page, automatically filling in user data for host and contact info as well as potentially pulling the date if they selected a certain day already.

Create User Profile

Users should be able to fill out a variety of parameters, ranging between a set of curricular and extracurricular elements, to display on their user profiles.



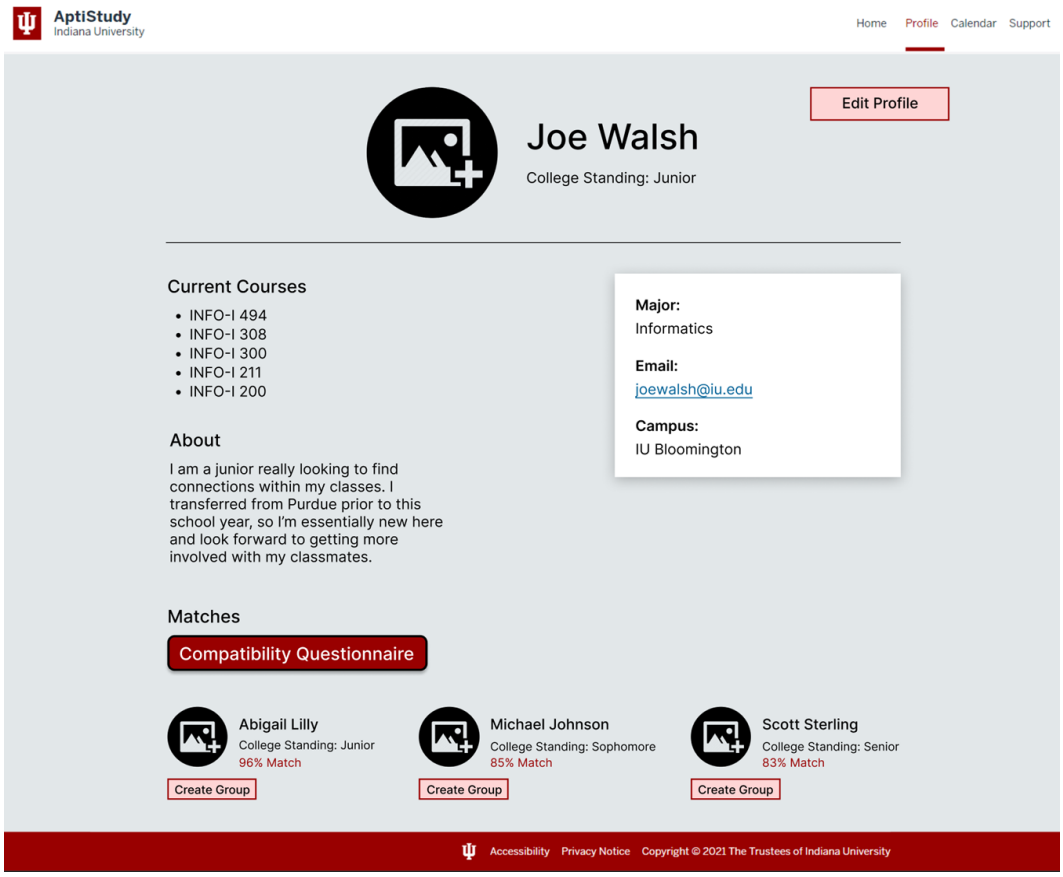
This page essentially acts as a form that captures data users submit onto their individual profiles. While the compatibility questionnaire may focus on a specific set of parameters to help students match with each other and find study groups, this form should encapsulate more "easy-to-fill" parameters outside of the scope of the compatibility questionnaire so that the user can be introduced to other functionalities within the application more quickly.

Revisions

Some minor revisions to this page entail removing the clubs and interest sections, as these seem somewhat counterproductive for the intent of this application since users can simply include these within their bio sections. We additionally include 'Current Courses' as a field indicator so that users are not confused whether to include current, prior, or combinations of current and prior courses within this field.

View User Profile

Once users complete their profile and/or make updates, they should be able to see these implementations on a viewable profile page in addition to concurrent information about potential matches for creating study groups.



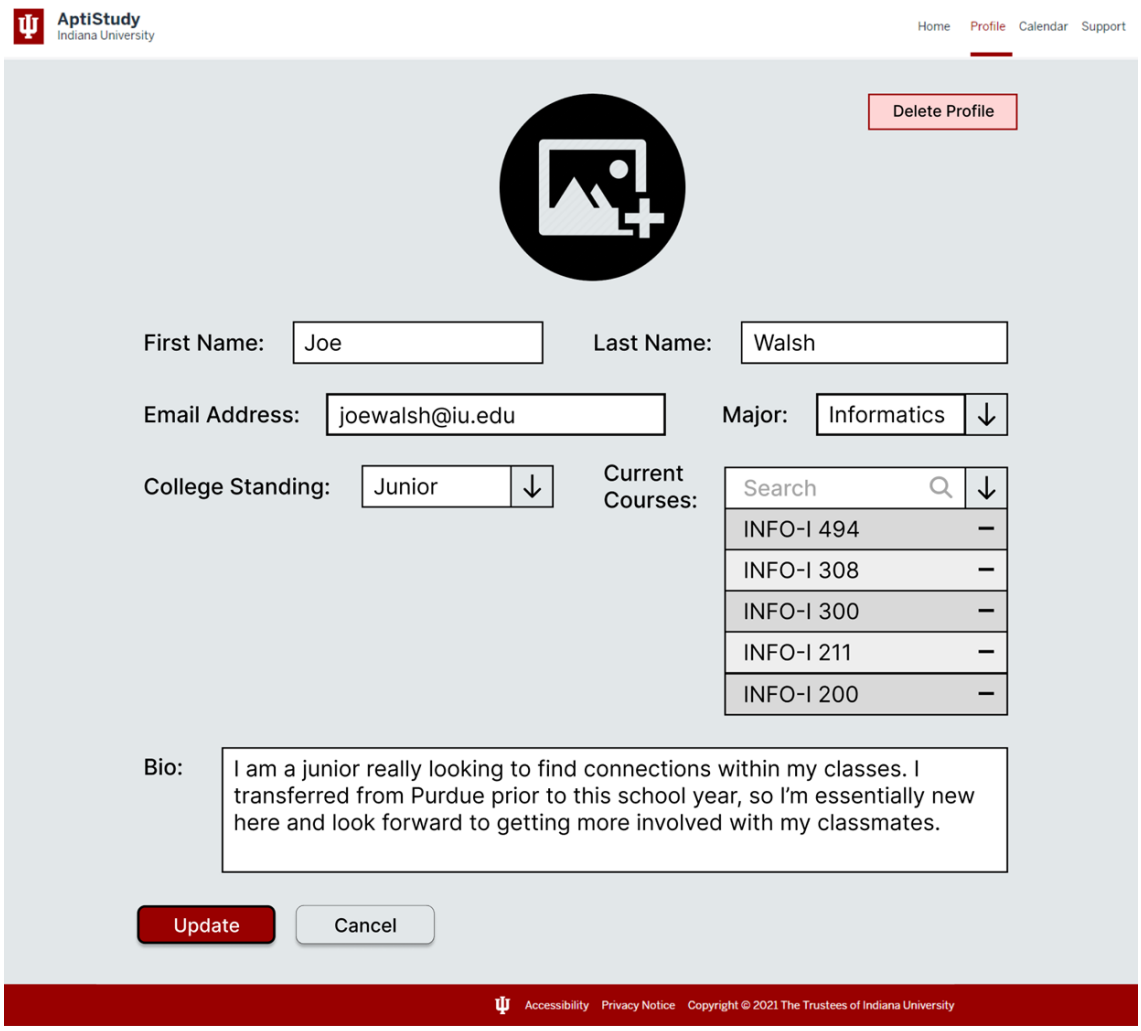
The viewable user profile page is designed with a two-pronged functionality so that users can easily view the information they input when creating or editing their profile as well as matches based on the compatibility questionnaire they filled out using the accessible button on the page. Outside of these design considerations, this page also attempts to replicate a standard user profile used within the Rivet framework so that the page feels contiguous in terms of design with the other pages in the application. Users should obviously also be able to edit their profile via a button whenever they choose.

Revisions

Revisions include removal of the club and interest sections also removed when creating/editing user profiles. The Matches section is also slightly shifted downward to differentiate on a user profile and be more visible for the users when viewing their user profiles for the first time. Lastly, the edit button in the upper right corner initially accidentally said 'Edit Group,' which has been updated here.

Edit User Profile

Whenever all required fields are submitted after creating their profiles, users should be able to edit these fields since students will obviously be enrolled in new classes every few months and need to incorporate updates to reflect this.



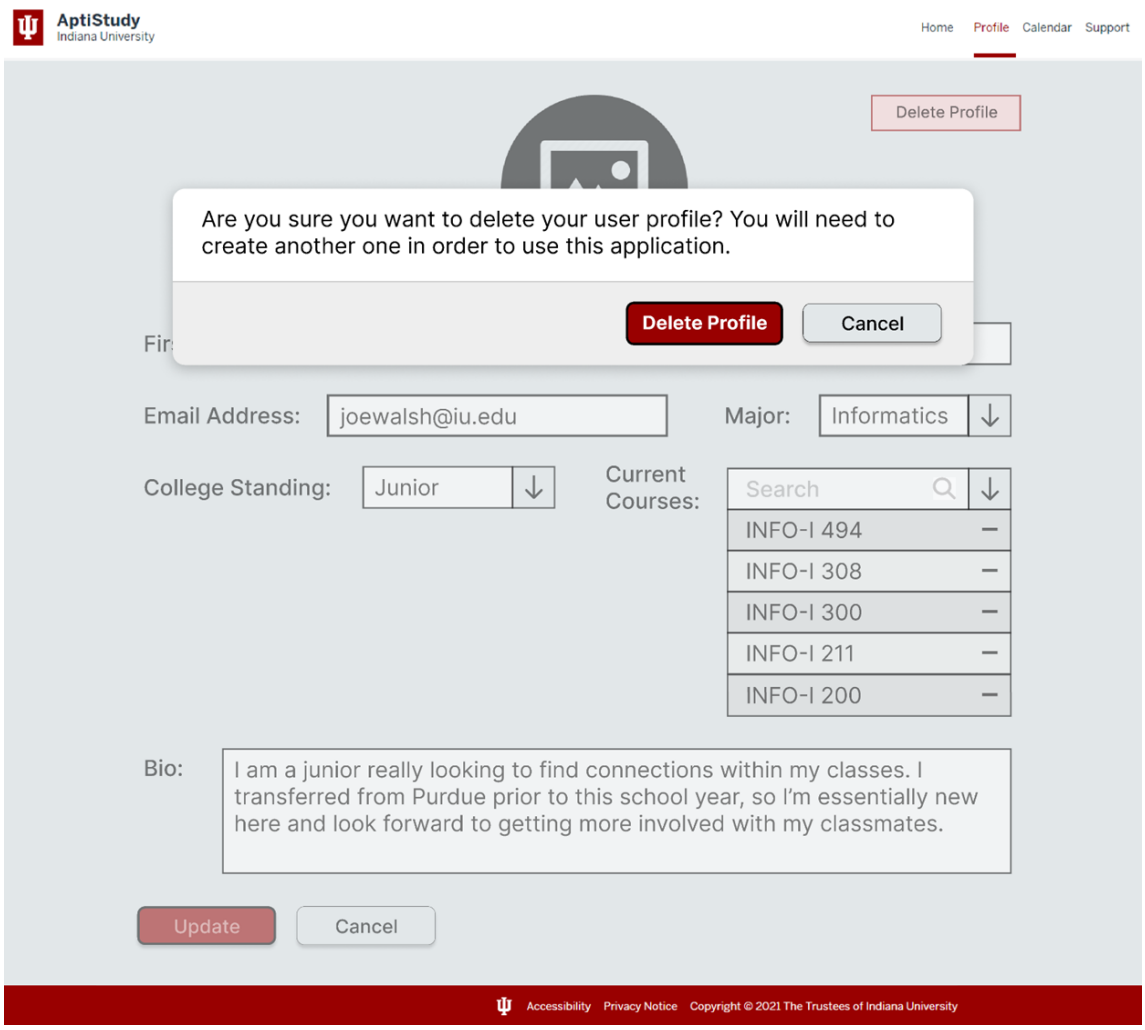
Our design rationale for this page reflects the idea that editable fields should be pre-populated with the information users already input so that they do not need to input it again. Likewise, multi-selective inputs such as courses, clubs, and interests should populate the list array generated by the user so that they can easily remove one or more inputs using buttons.

Revisions

Revisions include removal of the club and interest sections also removed when creating/viewing user profiles. Courses, as when creating profiles, have also been updated to note 'Current Courses' for the field indicator so as to alleviate this confusion brought up during the UX methods feedback.

Delete User Profile

If the user attempts to delete their user profile, a popup message will ask for confirmation so that users do not mistakenly delete their profile in the event that they unintentionally click the button.



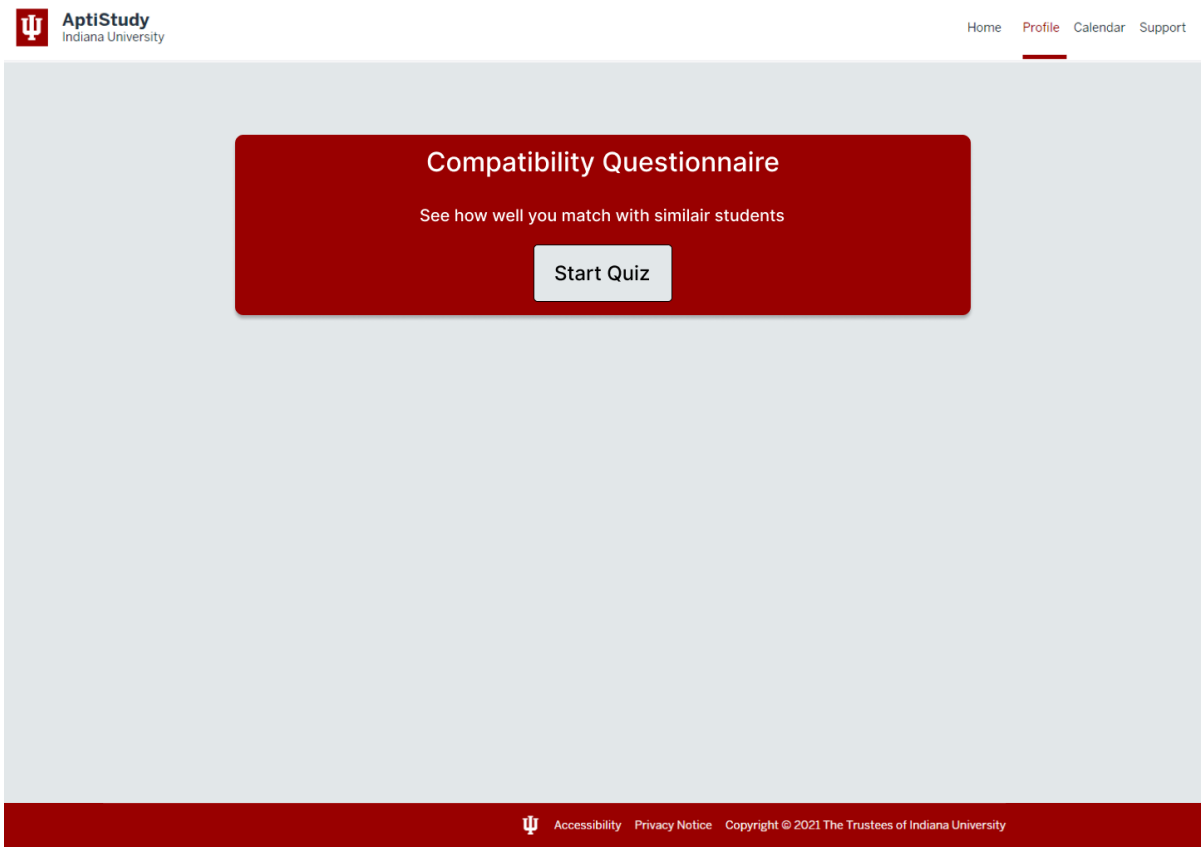
Although the message should not be too invasive or difficult to navigate, it is imperative from a UX design standpoint that we include confirmation for this delete action since we only want to initialize legitimate deletions. We will keep the user profile slightly transparent in the background to illustrate how the deletion will affect the amount of information they had previously input.

Revisions

Revisions include the same removal of the club and interest sections also removed when editing user profiles. Courses, as when creating profiles, have also been updated to note 'Current Courses' for the field indicator so as to alleviate this confusion brought up during the UX methods feedback.

Initiate Compatibility Questionnaire

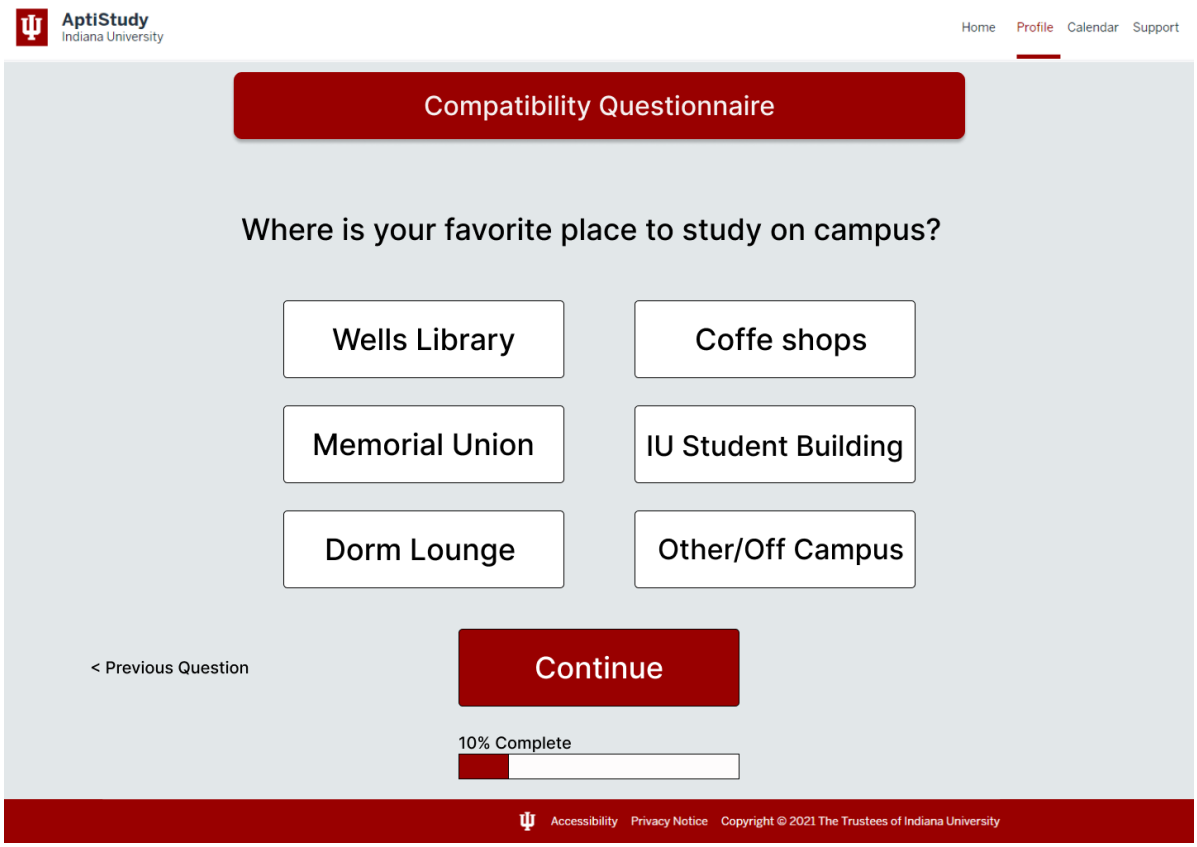
When the user has no active quiz, this is what the screen will look like in the Compatibility Questionnaire setting.



This screen was designed for users to easily know how to create a Compatibility Questionnaire. They only have one option so there is no confusion.

Fill Out Compatibility Questionnaire

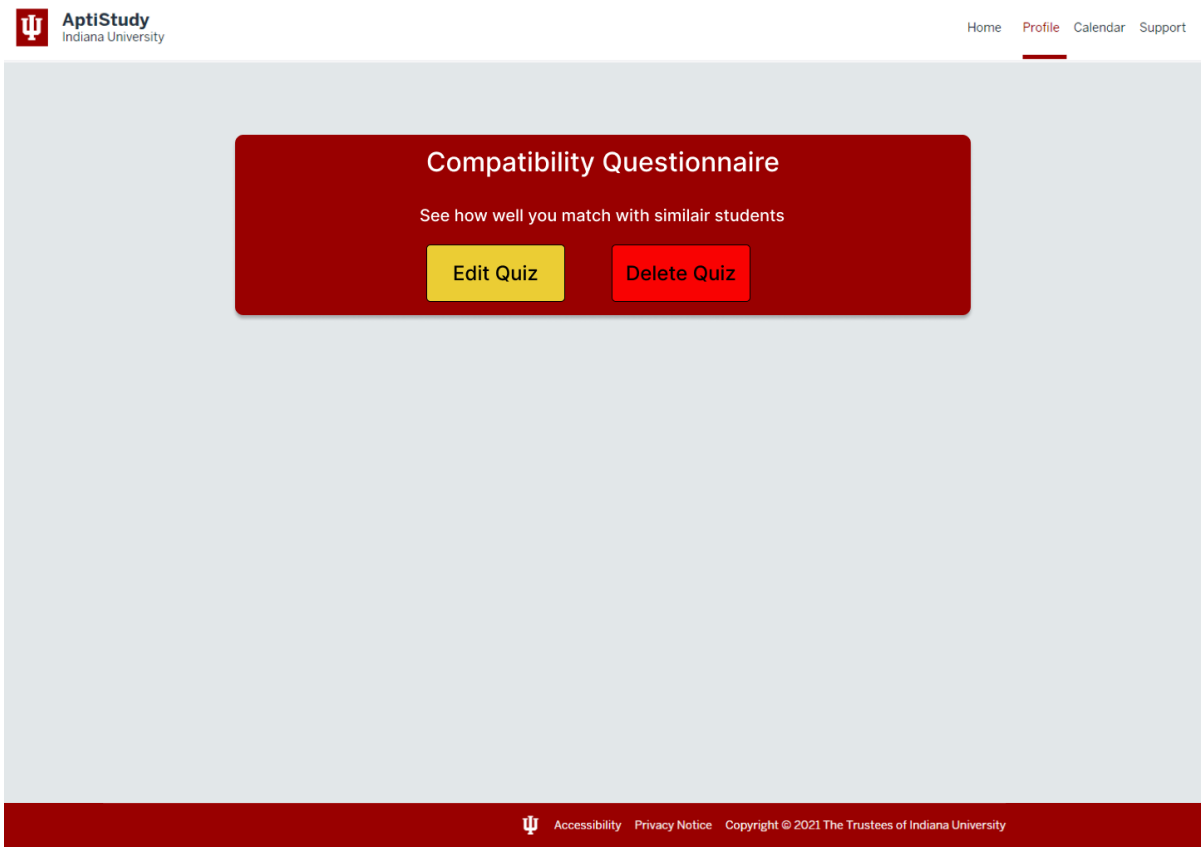
When the user clicks start quiz they will begin taking an easy multiple choice questionnaire about their studying habits to match with other students.



The user should not feel overwhelmed or really have to think too hard when filling out the questionnaire. Big buttons and only a couple selections make the quiz easy to navigate and understand.

Edit / Delete Compatibility Questionnaire

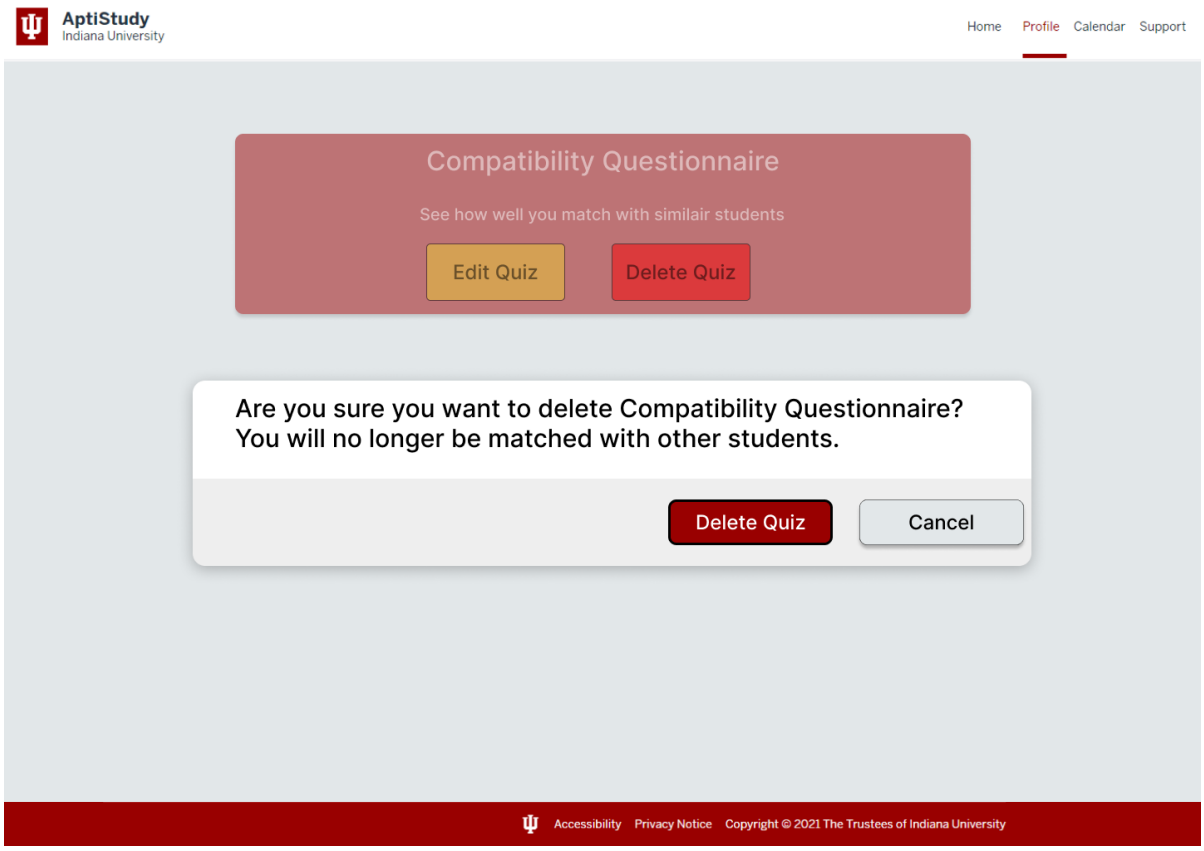
Users that already have a questionnaire active will have two options to either edit their responses or delete



Red delete button is to make sure users don't accidentally click delete. I colored it red as that usually means stop similar to a stop sign. Yellow was used for the edit button. Yellow was chosen because it stands out as my prediction is more users will want to edit the questionnaire than delete it.

Delete Compatibility Questionnaire

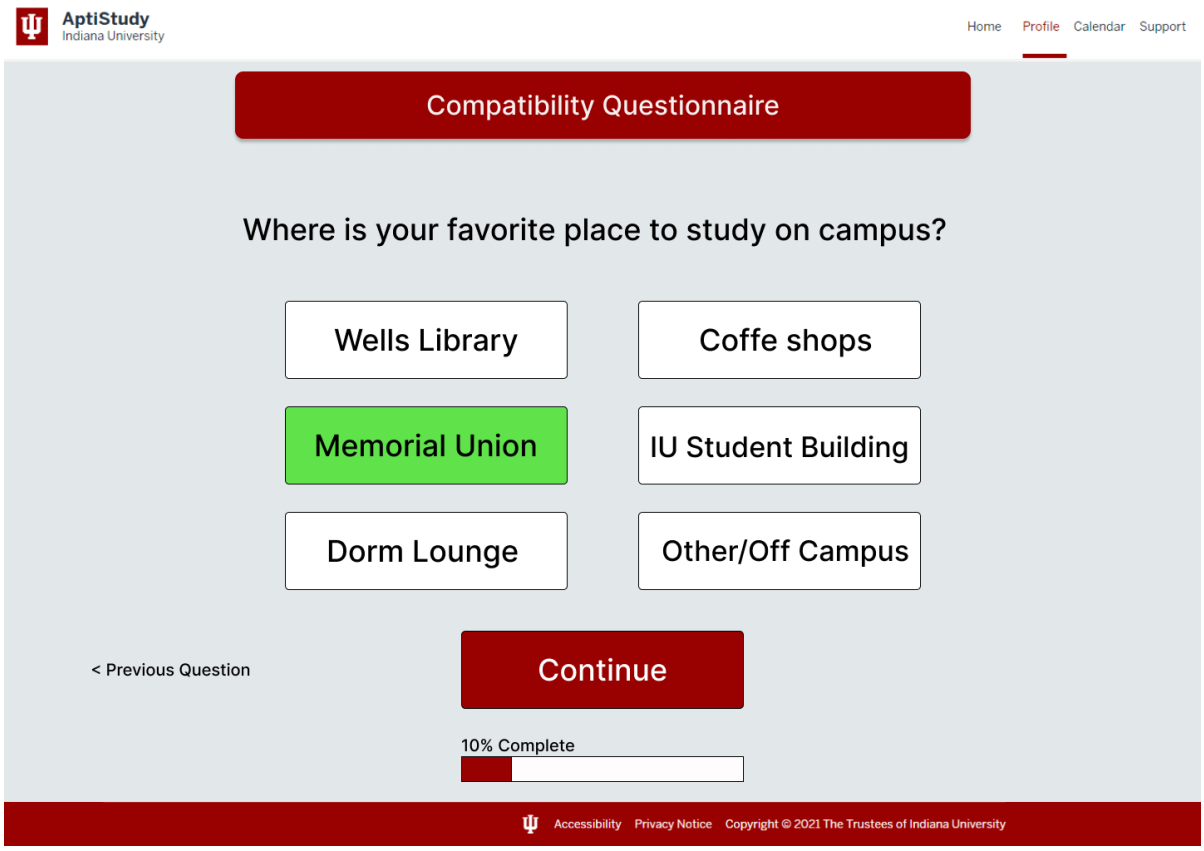
Confirmation delete message when a user selects delete quiz.



Allows the users to confirm delete to prevent accidental deletes. Helps tell users what will happen if the questionnaire is deleted.

Edit Compatibility Questionnaire

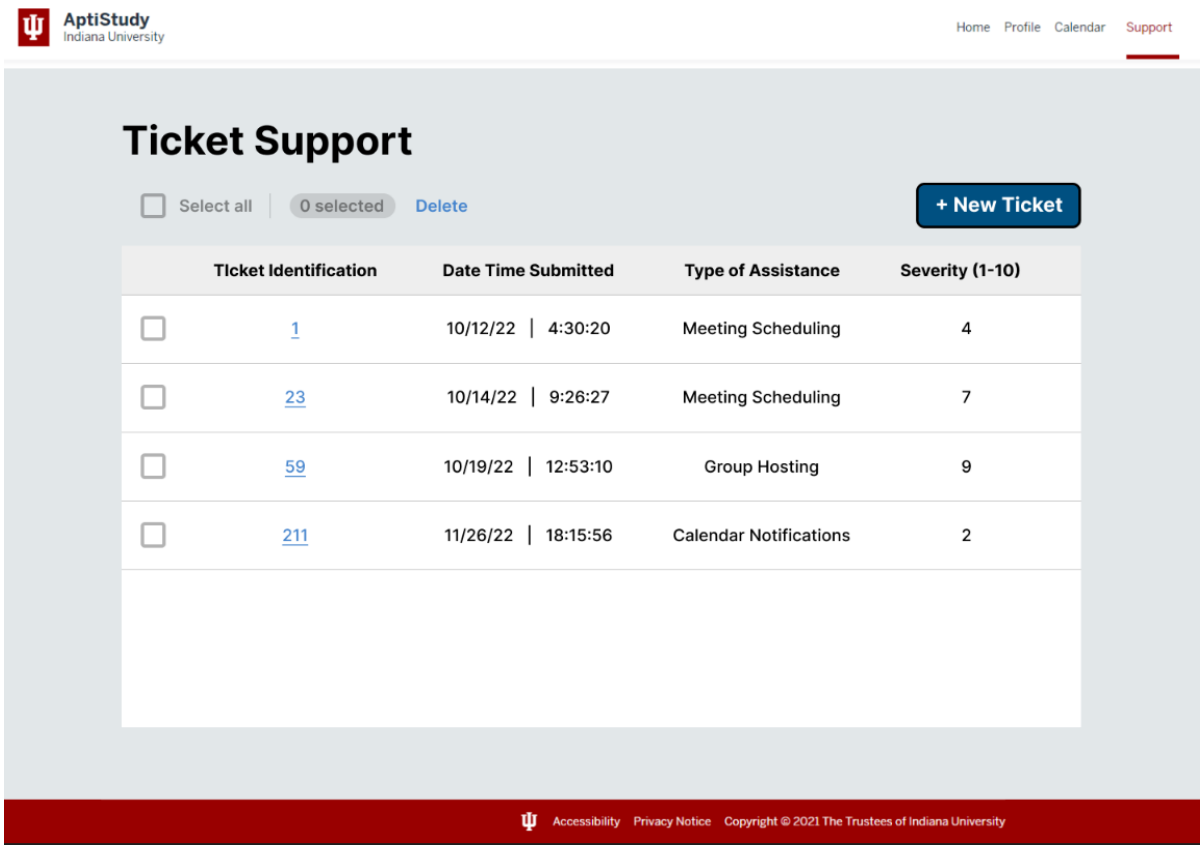
Allows users to go back to the questionnaire and change responses.



Green is what color the button changes to after you select that answer. Selecting a different response will then turn that response green. This allows users to know what they put for a question they already answered.

View Tickets Sent

If users want to be able to document the tickets they have sent requesting support, this page will show general information associated with each ticket as well as accessible links for viewing each ticket individually; we can see that users are able to also delete multiple tickets seamlessly if they want to do so.



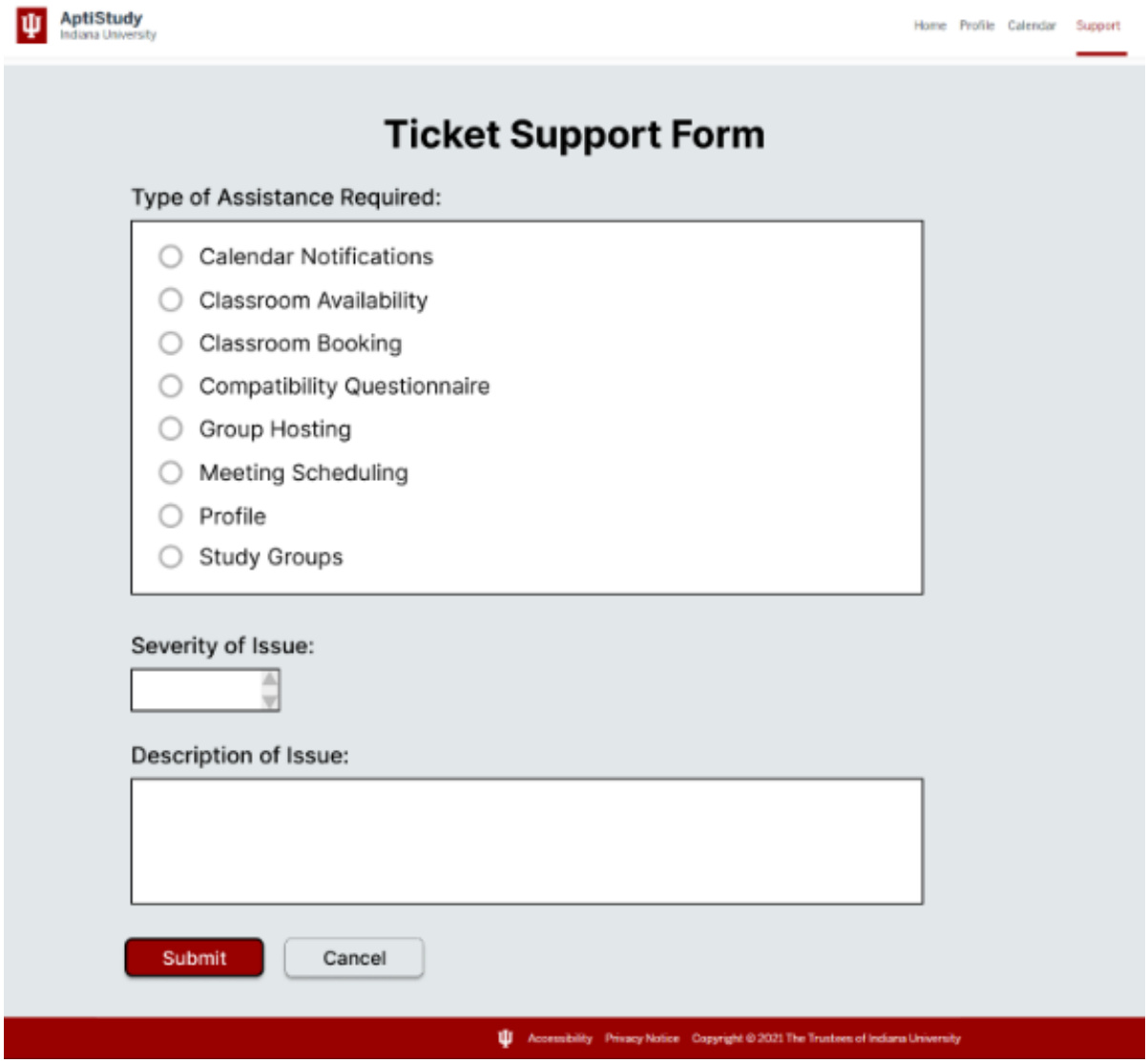
If technical problems persist after a ticket has been filed in the support system, it would make sense from a design perspective for users to be able to show documentation illustrating severity and the fact that the problem has not been resolved. This design transcribes prior ticket inputs into an easy-to-read format and is filtered based on times in which users had previously sent tickets. Users should also be able to easily add and delete new tickets on this ticket support hub page.

Feedback Notes

One participant noted that we may want to potentially replace our ticket support system with messaging functionality. While we obviously see the benefit of messaging, we felt that we did not have adequate experience with this functionality even though Josh has completed a bit of socket programming in C language. Having a ticket system also ensures that we can mitigate abuses by users and potential threats based on the reserve room functionality.

Initiate and Send Individual Ticket

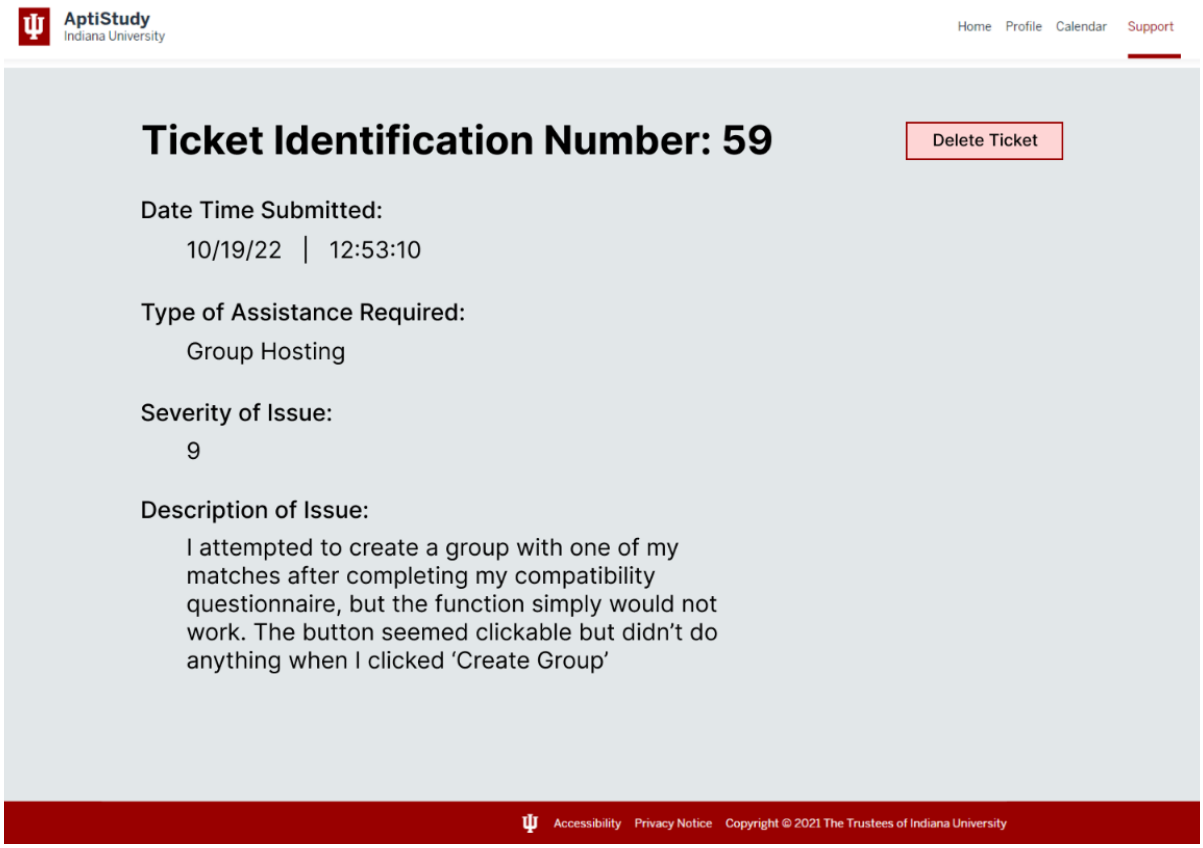
Users can specify parameters for support tickets based on the type of assistance required, the severity of the issue ranging from 1-10, and a description of an issue for subsequent validation and updates made by administrators.



Since we want to incentivize users to send tickets when a functionality is not working as intended, we need the ticket support form to be simplified yet insightful based on the issue presented. Therefore, the type of assistance required initially narrows down the issue to a functionality or Epic, the severity gives weight to the necessity for technical support, and the description allows users to detail what they were not able to do in relation to the type of assistance required.

View Individual Ticket

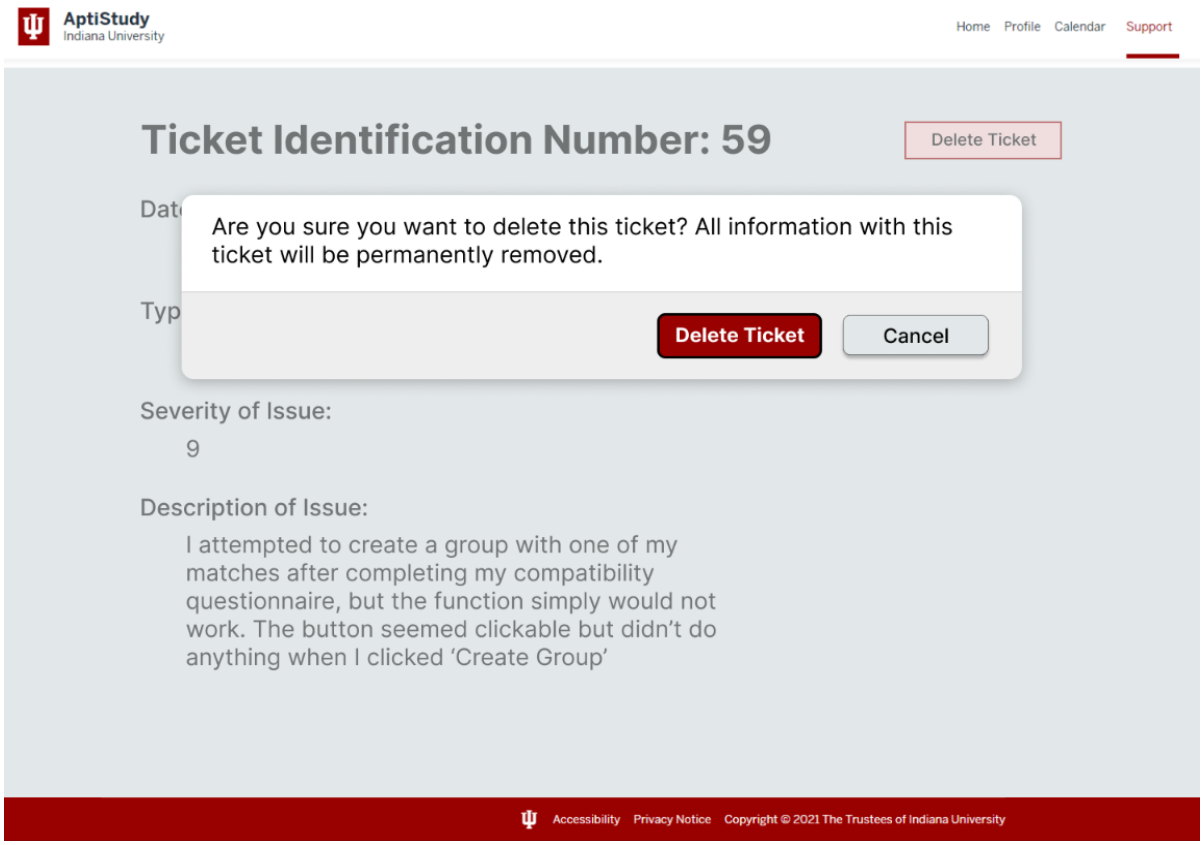
Users can view their previous inputs for a specific input when they select the identification number; this page reiterates most of the information from the ticket support hub page yet also includes the description of the issue presented.



Although we could have included the description with the other supplementary information on the ticket support hub page, we wanted to include these pages for better readability so that users do not feel overwhelmed when viewing many tickets at once. Users should not be able to edit tickets since this could create documentation issues for administrators, but they can individually delete tickets if the issue either has been resolved or is not relevant anymore.

Delete Individual Ticket

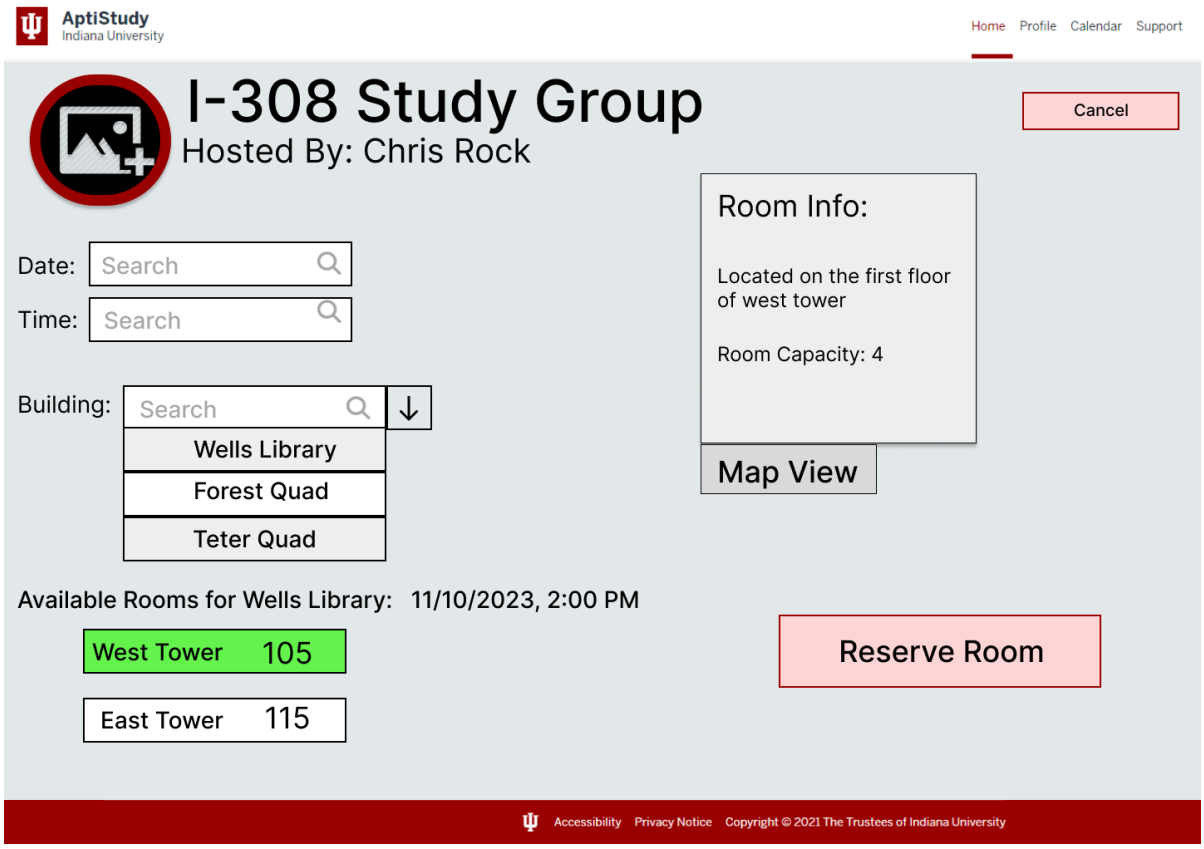
As with the other deletions, this page will include a confirmation popup to ensure that users do not unintentionally delete tickets after clicking the 'Delete Ticket' button.



Users do not have the same level of control compared to us as administrators, so we need to limit the amount of unintentional consequences for them by including confirmations prior to deletion actions. Therefore, we include an easy-to-navigate popup where users can confirm the deletion or cancel if they do not legitimately intend to delete the ticket; we will keep the individual ticket slightly transparent in the background to illustrate how the deletion will affect the amount of information they had previously input and which ticket will be deleted.

Reserve Room on Campus

This page illustrates where users will be able to reserve rooms around campus for study groups.



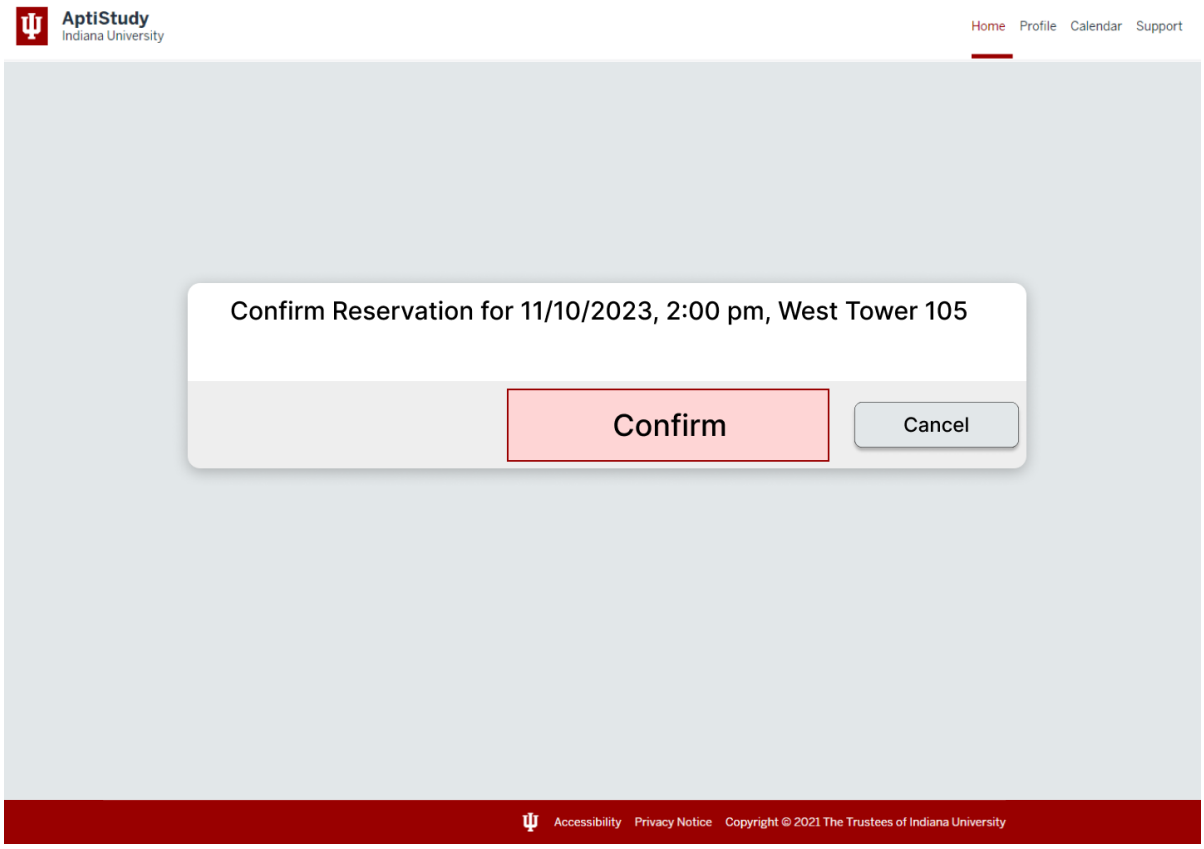
The user will be asked to enter in a date and time. After that all the buildings available during that date and time will be shown. After picking a building all the rooms available for that date and time will be displayed. Green means that the user has selected a room. I wanted to make this page very user friendly only giving the user valid options to choose from after selecting a date.

Revisions

This was essentially depreciated and moved onto the reservation screen, as a result of our feedback review having stated that it was a little tedious to have to create a meeting and then to to a separate screen in order to book a room. Meaning that we should really just integrate the room booking on the meeting set up form.

Confirm Reservation

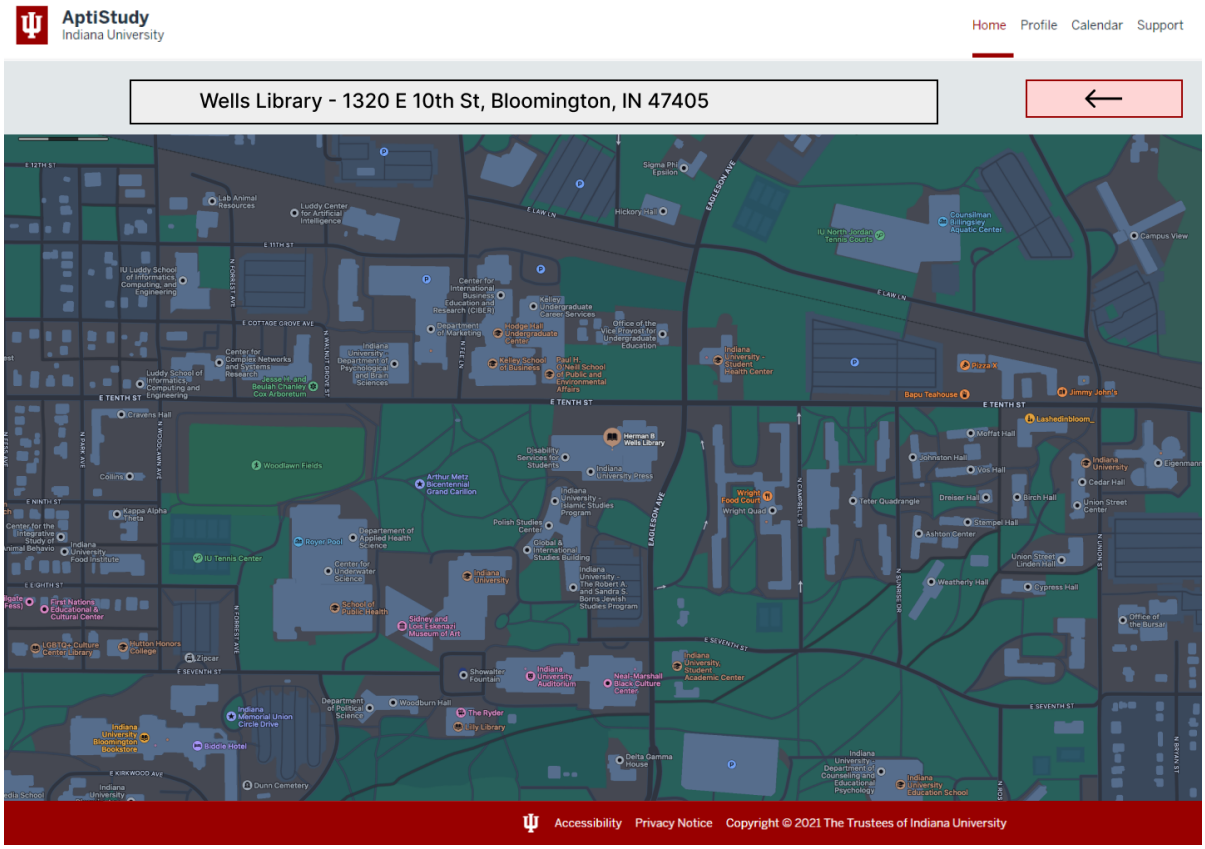
This visualizes a prompt confirming that a user wants to reserve a room.



It is important that users don't accidentally reserve rooms. For this reason, I created a pop up for a user to confirm exactly when and where he or she is reserving a room. Red is for users to stop and look at what they are clicking on.

Map Integration

When reserving a room users will have the option to look at the building in which the room they reserved is on a map.



At the top is the address of the building followed by a full screen map; I created a full screen map so users can easily visualize where on campus they need to go.

UX Results

To better understand the current progress and architecture of our project from the perspective of others, we utilize a series of user experience (UX) methods, including Observations, Focus Groups, Interviews, and Think-Aloud Sessions to be able to generate unique insights for accomplishing this overarching goal. UX studies also bridge the current gap between us as developers and our target demographic as IU students, because it is imperative that we practice applying socio-technical perspectives to ensure that our system is not one-sided in favor of our initial technological considerations as developers. In other words, we gear our system toward IU students who either particularly struggle to make connections with other students or need assistance within their courses, so utilizing this sociotechnical perspective gives more weight toward their needs for developing our system going forward. Most importantly, in respect to the "big picture" and our prospective actions in correspondence to these studies, we can apply the constructive criticisms and desires cultivated from the insights for improving upon our application prior to implementation since there will always exist limitations or shortcomings that can be elucidated based on interactions with our target demographic.

Although the subsequent pages will include more granular aspects of these UX methods conducted, we will note more broad summaries here. To differentiate between project and prototype studies, we utilized Observations and Focus Groups for our project topic since we felt that these methods were more appropriate when generating insights regarding our overall project. This indicates that we may have less constructive criticism toward our current progress since we more so want to understand how participants view academic groups as they are and whether they can be improved through an application with a set of features; we can also observe whether there exist any similarities or differences from the discussions expounded and if we can apply these to our prototypes and/or UX. As for Interviews and Think-Aloud Sessions, these pertain to our actual prototypes, where we receive more direct feedback on functionalities that were proficient as well as functionalities that were confusing or could be improved upon in some way. In consensus, most feedback exacerbated that the overall prototypes were easy to navigate but could incorporate a few tweeks since some participant tasks were not as intuitive as we initially predicted. Overall, we conducted four UX methods with eight total instances, where the Observations and Focus Groups focused on our project topic relating to study groups while the Interviews and Think-Aloud Sessions focused on our prototypes and direct visualizations of our project scope.

Observations

Observation 1 (Project) - Alex McGaugh

Date/time: 11/16/22 / 2:00 - 3:30pm
Location: Wells Library

About: Wells Library is a prime hotspot on campus for student gatherings and study sessions. This spot is great for observation since almost all the kids that study in Wells are IU students that perfectly fit our target user group. Wells also has a variety of different style rooms for studying.

Overview: The main focus of this observation is to see how students interact with each other and the environment around them while studying. In Wells Library, I will be conducting two observations in order to gather more accurate data. My first observation will be on the first floor in the West tower, and my second observation will be on the 9th floor. Some observations that I will be looking out for are the use of closed off study rooms, typical group sizes, average time spent occupying spaces, and any relevant information regarding our problem space.

Results: For my first observation in Wells library, I sat on the first floor in the West wing. I sat across from the elevators at a computer desk for about 30 minutes; it was a very busy time in the day for studying at 2:00pm. Regarding the number of people in groups, I could easily tell from looking around that there are more people working within groups than are working individually. Students would appear to sit alone at a round table or in a closed off study room, but then it would appear that other students would come meet them to form a group. I would assume that this is students trying to save a group spot for the rest of their party. Average group size on the first floor was 3 or 4 people. In all 30 minutes, I was only able to spot one group with a size of over 5 people. Every single round table and glass room was filled. Some groups of kids formed groups at computer tables all sitting side by side. Most groups observed outlasted me in time spent studying; therefore, I would conclude that the average study group spends more than 35 minutes at each location. For the 9th floor in the West tower, there were way less groups compared to students studying individually. Overall, the floor was quieter since there was no chatter from groups. On this floor, the glass rooms were a lot smaller, making the average group size at 2 people. People on this floor seemed to come and go at a faster rate compared to the bigger groups on the first floor. Overall, it seemed that students studying in groups outweighed the students studying by themselves creating a bigger need for more group style study locations in the library. Students in groups seemed to stay longer than individuals. From my observation in Wells Library, I can conclude that students would benefit from a more organized style of studying rather than from finding a space or finding other individuals to study with and make friends.

Observation 2 (Project) - Alex McGaugh

Date/time: 11/17/22 / 2:30 - 3:30pm
Location: Indiana Memorial Union (IMU)

About: The Indiana Memorial Union is another prime spot for studying at IU. Similar to the library, if not all most students studying at the IMU are IU students which is our specific user group.

Overview: The main focus of this observation is to see how students interact with each other and the environment around them while studying. In the IMU, I will be conducting my one hour observation in the south lounge of the building. I felt this was the best spot due to both groups and individuals studying in the same place. I sat in a single couch chair so I will be taking up as little space as possible for a better observation.

Results: Since there were no dedicated closed off study group rooms, I expected the number of groups to be less. Instead, groups and individuals were about equal. Groups would take over the facing couches in the center and sides of the room. Average stay for a group was about 20 to 30 minutes, as 3 groups came and left in the span of 1 hour. Average group size was surprisingly higher than the library at about 5 people. Couches throughout the IMU seemed to be filled with students, and overall space was more crowded than any precious location. Many students walk around for seating then eventually leave to find seating elsewhere, as there are no seats left. Overall students need a more organized space to study throughout the IMU. There are plenty of unused rooms that students should be utilizing instead of trying to cram into the lounge and cafeteria areas.

Focus Groups

Focus Group 1 (Project) - Josh Hatfield

Date/Time: 11/15/22
Subjects: Holt Henke, Ben Malone, Nick Mathein

About: These individuals were Josh's team members in INFO-I 330 and were easily accessible for conducting this focus group study, which also benefited in regard to our methodology since they all fall within our target demographics as IU students. Because these participants have also acquired professional experience working with various UX systems, we wanted to discuss their insights in relation to iterations or ideas toward our own project so that we can coalesce these discussions with the other focus group conducted. All note their prominent involvement with curricular and extracurricular groups.

Overview: To consider our overarching project topic, where we intend to streamline the process of finding and utilizing study groups within academic settings, focus groups allow us to generate more broad sentiments in addition to ways in which we can ruminate upon our topic from a more preliminary or rudimentary level. They additionally offer more fluidity compared to interviews since participants have more freedom to promulgate thoughts among each other while employing a group moderator to be able to guide the discussion at hand. The first focus group is given a set of five broad questions (differentiating from the second focus group), where participants are provided the opportunity to voice their thoughts within a time duration of five minutes for each question. Each participant can discuss as many or little points within the timeframe as possible.

Results: Participants were receptive to study groups within courses since they all noted that there were at least some instances in which they felt that having at least a study partner or group would have benefited them due to the nature or difficulty of the course. They had additionally encountered previous difficulties with groups as well, because communication was sometimes disorganized given how messages could oftentimes be a long wall of text and therefore were not always easy to remember, or members were less incentivized for keeping others accountable by constantly sending texts to the group. Lastly, within the focus group, they noted how COVID in particular demonstrated that online meetings were not always effective since members could be less inclined to remember meetings, and there could be technology barriers due to them trying to do too many tasks with their devices while on Zoom.

Focus Group 2 (Project) - Josh Hatfield

Date/Time: 11/19/22
Subjects: Jacob Hatfield, Elijah Hatfield, Jonah Sullivan

About: The following participants are relatives to Josh but were chosen due to their relevance as individuals within our target demographic outside of the informatics department, which may help diversify our insights since they may not be experienced with technology systems and therefore can help contribute to or help guide our sociotechnical perspective for our project. Jacob (Josh's brother) is senior intending to graduate with a biochemistry major, who often needs to work in many groups for his laboratory work and joins study groups for course exam preparation. Eli (Josh's brother) is attending an undecided college as a freshman next year, and while perhaps not exposed to study / academic groups thus, may offer some expectations based on his research on colleges thus far. Lastly, Jonah (Josh's cousin) is also an incoming freshman for an undecided college next year and can provide similar or unique insights drawn from Eli given their similar situations.

Overview: By diversifying our target demographic outside of the first focus group, we can maintain the same level of fluidity for discussion yet incorporate a different perspective when considering the overarching theme of our project. Here, the conversation remains more open-ended, where participants can answer individually or communicate points on top of prior participants' responses. The second focus group is given a set of five broad questions (differentiating from the first focus group), where participants are provided the opportunity to voice their thoughts within a time duration of five minutes for each question. Each participant can discuss as many or little points within the timeframe as possible.

Results: For this focus group, all importantly noted that they prefer to select groups over being selected for groups based on the professor / teacher's discretion. This sentiment was based on the premise that although selecting groups with friends could affect individuals' obligations toward making contributions for their groups, they generally have a better awareness of how much their group members will contribute and feel that selecting removes much of the initial awkwardness with being placed in groups with people they don't know or even dislike. All acknowledged the importance of groups in college since there seems to be a higher standard, and professors want to gauge how well you work with others when working collaboratively on a larger project. In regard to "good" and "bad" group environments, the participants made note that communication, favoritism, and schedules allow play some role into the spectrum of successful groups since any one of them can change the dynamics of the group to create unequal member contributions. Lastly, the current technologies in place like GroupMe or Microsoft Teams seem to have some useful features for the participants, but they ultimately feel that there is not as much relevance or depth for using these applications.

Think-Alouds

UX Think-Aloud 1 (Wireframes) - Dylan Johnson

Date/Time: 11/16/22
Subject: Mikayla Knierim

About: Mikayla is currently an undergraduate student at IUPUI. She lives a very social and active life within her community. She has a serving job and tends to be very active with her friends and family.

Overview: This think-aloud session was based around all of the epics that we decided to include in our project. Each test is broken down individually to the feature being tested. All of the tests were conducted using Figma. There was no prototyping involved since it is a medium-fi prototype. The user had to assume the natural flow of the application. I recorded the response after each test along with quick notes.

Results: Overall, our application is a great start up but it could still be improved upon. There were only minor questions with the profile process. The end user didn't have much trouble navigating through the application. However there were a few issues with locating features and figuring out the functionality of some features. To improve upon this, the user recommended creating a new tab on the navigation bar for groups and designing our prototype further.

UX Think-Aloud 2 (Wireframes) - Jonathan Yen

Date/Time: 11/14/22
Subject: Brian Yu

About: Brian is a graduate student at IU Bloomington, and is rather social. He takes the time to interact with the social circles that he is a member of as commonly as possible and separates his work from his personal life.

Overview: This think-aloud session was driven by a list of tasks the user had to complete for each epic or feature of our application. The application was shown in the Figma format without prototyping, letting the user just pan through the images acting as though they were actually using the application. While I did not take navigation time into consideration, I did ask after the test over all how well each feature was able to be piloted.

Results: In the most general sense, it seems that the application is at its very baseline, and even so is a little hard to understand. That being said, the application seems to be intuitive and easy to navigate, just that some of the concepts that we tried to portray were not displayed correctly. It seems that we just need to specify some categories a little better as well as meld some functions together to make it a more seamless process to create a meeting.

After the conducting the think-aloud method, he had given it an overall good review, saying that it had the good works to become a good application, although maybe missing a few things or including a few maybe not needed features; for example, from the think-aloud, he thought it was weird that we didn't put location reservation under our meeting creation tab, and that why would we have to insert our data into our profile after signing into the application via IU, seeing as the matching algorithm and the profile information could be taken from IU. In general, if something like this were to be implemented through One.IU, it might be lost to the masses, as it seems from his experience One.IU is a pretty bad application aggregation tool. Regardless of this, he thought that if we implement it well enough, many incoming freshmen would find it a great application to easily find study groups and even maybe social circles in their first year of college. Seeing as this is a UX report and not exactly a true functionality report, we derailed somewhat with not a lot of time left. This being said, Brian did mention that he does like his applications to at least have some streamline designs and conventions used by most applications today, but each one with unique application specific flavoring, but seeing as our application would be IU based, an IU stylized interface would fit the bill quite well.

Interviews

UX Interview 1 (Wireframes) - Dylan Johnson

Date/Time: 11/16/22
Subject: Mikayla Knierim

About: Mikayla is currently an undergraduate student at IUPUI. She lives a very social and active life within her community. She has a serving job and tends to be very active with her friends and family.

Overview: In order to capture the best data, I conducted two interviews. One interview was completed before the think-aloud and then I followed up with one after. This was done to see if our application meets the needs and wants of our target audience.

Results: Mikayla explained to me that finding a good group to study with has been a very difficult process in the past. She said she is a social person but finding ‘good’ study partners causes her anxiety and then she ends up studying alone. She went on to say that juggling a full school schedule and over twenty hours of work in a week makes it hard to find free time to work with others on homework. Most of the time now she is only a part of groups because she is assigned to them in class. She explained to me that time, location, and class were the most important parameters if she was creating a meeting. With a busy schedule, it is important to her that she is available and the location isn't far away. Regular meetings aren't too difficult for her to arrange as long as they are over zoom and not on the weekend. Mikayla said if she had to create a meeting scheduler application like ours, she would be sure to include time, location, messaging, meetings, and notifications. She thought having a calendar and ticket system isn’t necessary for an application like this. The calendar could be confusing and there are other applications that can keep track of this. Instead, she recommended including a messaging system instead. That way, if you were running late or needed to share a file with a group, it might be more convenient to just send a message.

After reviewing the whole application, Mikayla gave our project a good review. She said overall the navigation and functionality was on the right track. We included everything and more that she would want from a meeting scheduler application. There were a few minor issues that she encountered with our application. She said she would remove the dropdown from the home tab and put it under a new tab labeled ‘Groups’. On top of that, she expressed that she would remove the group header on the calendar page. This made it confusing to determine if it was a group or personal calendar. Besides that the ticket system works flawlessly. It was very straightforward for her and there were no issues figuring this out. Most features on this application were intuitive based on visual elements. The only thing that took learning were the calendar events. It took a second to learn that you need to click on them to view or edit. Lastly she said our application had a very similar feel to Canvas. She highly recommended that we push for a similar design because it helped her figure out the flow of our application quicker. Having the convenience of not learning a new layout was a big factor for her.

UX Interview 2 (Wireframes) - Jonathan Yen

Date/Time: 11/14/22
Subject: Brian Yu

About: Brian is a graduate student at IU Bloomington, and is rather social. He takes the time to interact with the social circles that he is a member of as commonly as possible and separates his work from his personal life.

Overview: The interview session was conducted before and after the think-aloud testing, and therefore helped me gauge what the user would think they want as well as how well our application matched their vision of utility.

Results: Brian had only ever really encountered the problem of not being able to find a group and not wanting to reach out after the start of the semester once or twice before, leading to a few awkward times in his second year. However before and after that group was chosen for him and generally worked out. In most cases when setting up a meeting with his groups he hadn't really thought much other than the location and time, and the groups only ever really worked with each other for class activities and assignments, and meetings only happened as catch ups after each class. Their outside of class meetings turned out to almost always be online Discord voice calls, therefore only making time an issue when scheduling group work times. Specifically Brian had never really had a study group he attended for a class before. When creating a vision for an application he had a few points:

- Time and location
- Subject
- Messaging

Other than that, he didn't really think it was necessary for much else. Asking whether or not messaging would clog up his messaging push notification with as many messaging applications as there are nowadays, he did mention that at times it would become pretty annoying managing so many messaging systems between Google, SMS, Email, Discord, GroupMe, Instagram, and FaceBook.

Project Timeline

Sprint	Task (Owner)	Dependencies
SP-08	Task 1: Initial application pages, including navbar and other elements (Josh)	None
	Task 2: Database setup based on ERD and feedback (Dylan)	None
	Task 3: Database population and tests based on various queries (Alex)	Task 2 (SP-08)
	Task 4: API integration research (Jonathan)	None
	Task 5: IU login integration and sign-in page (Josh)	Task 3 (SP-08)
SP-09	Task 6: User profile pages (create user profile through form, view user profile, delete user profile) (Josh)	Partially dependent on Task 5 (SP-08) for initializing user profile page
	Task 7: Study group pages (create study group through form, view suggested study groups, join study groups) (Dylan)	Partially dependent on Task 5 since only returning users can host or join study groups (SP-08) Dependent on Task 8 in regard to viewing suggested study groups (SP-09)
	Task 8: Compatibility questionnaire pages (initiate questionnaire, fill out / edit questionnaire, delete questionnaire) (Alex)	Partially dependent on Task 6 (SP-09) since only returning users can fill out questionnaires
	Task 9: Calendar integration without notifications (Jonathan)	None
SP-10	Task 10: Ticket support system pages (initiate ticket through form, view all tickets sent, view individual tickets, delete tickets) (Josh)	Partially dependent on Task 6 since only returning users can send tickets (SP-09)
	Task 11: Host and join study group session pages (host / edit meetings, join meetings, view meeting confirmation) (Dylan)	Task 7 (SP-09)

	Task 12: Reserve room pages (reserve room, confirm room reservation, map integration) (Alex)	Task 7 (SP-09) Task 11 (SP-10)
	Task 13: Notifications for calendar integration (view notification, create / edit notification, delete notification) (Jonathan)	Task 7 (SP-09) Task 9 (SP-09)
SP-11	Task 14: Initial iterations based on feedback and basic debugging (Josh)	
	Task 15: Database iterations and debugging (Dylan)	Task 3 (SP-08)
	Task 16: Webpage iterations and feature debugging (Alex)	Previous tasks
	Task 17: User permission debugging and API checkup (Jonathan)	Task 6 (SP-09)
DEMO	Task 18: Complete last minute iterations and debug checks (Josh)	
	Task 19: Complete last minute iterations and debug checks (Dylan)	Task 17 (SP-11)
	Task 20: Prepare project for first presentation and demo (Alex)	Partially dependent on all previous tasks for the demo presentation, but the demo itself will need to be flexible based on existing features or bugs present
	Task 21: Prepare project for first presentation and demo (Jonathan)	Partially dependent on all previous tasks for the demo presentation, but the demo itself will need to be flexible based on existing features or bugs present
-	"Spring Break"	
SP-12	Task 22: Utilize focus groups for set of UX studies and make UI iterations based on first project demo (Josh)	Task 20 (DEMO) Task 21 (DEMO)
	Task 23: Utilize interviews & think-aloud sessions for UX studies and make database iterations based on the other feedback (Dylan)	Task 20 (DEMO) Task 21 (DEMO)
	Task 24: Utilize observations for UX studies and consider other UX iterations outside of the UI (Alex)	Task 20 (DEMO) Task 21 (DEMO)

	Task 25: Improve the prototype based on both interviews and think aloud sessions, as well as DEMO time feedback (Jonathan)	Task 20 (DEMO) Task 21 (DEMO)
SP-13	Task 26: Conduct a second set of UX studies using focus groups with iterations now implemented (Josh)	Task 22 (SP-12)
	Task 27: Conduct a second set of UX studies using interviews and think-aloud sessions; consider any other slight improvements (Dylan)	Task 23 (SP-12)
	Task 28: Conduct a second set of UX studies observations for further UI iterations (Alex)	Task 24 (SP-12)
	Task 29: Testing prototype iteration based on Task 26 data with new and old stakeholders (Jonathan)	Task 26 (SP-13)
SP-14	No project work; grading	

Sprint 08 (Weeks 01/02)

**Note: We discussed completing some of these steps during winter break if time permits; this may entail initializing a test database using one of our individual accounts and beginning our code with the Rivet framework for our navbar as well as any other elements like the IU login integration, which could free some of our time for focusing on other features*

- Task 1: Initial application pages, including navbar and other elements (Josh). This will ensure that we can prioritize the features going forward and know which pages ascertain which features using various include statements for better organization. Not dependent on any other tasks.
- Task 2: Database setup based on ERD and feedback (Dylan). Database setup will correspond to our data collection and how we display information per user for our system through the semester. Setup will include all entities, relationships, cardinalities, and attributes as illustrated by our final ERD design. Not dependent on any other tasks.
- Task 3: Database population and tests based on various queries (Alex). Having tests will allow us to determine whether all the attributes are correctly specified; we also intend to include manually curated courses to demonstrate functionality for the application yet assume that these are procured from the IU database that we do not have access to as a team. Partially dependent on the database setup (Task 2) of this Sprint for applying the data population and tests into the actual database.
- Task 4: API integration research (Jonathan). Because we are importing various libraries hosting the APIs or making external calls to the APIs, we need general knowledge and

testing parameters to determine whether these APIs will be feasible for our application as well as whether we need to, for instance, create our own calendar with the Google Calendar API. Not dependent on any other tasks.
- Task 5: IU login integration and sign-in page (Josh). This represents the user initialization process so that only IU students with credentials will be able to access our application. We will need to automatically generate data for our database given this login process so that we can differentiate whether the user has already created a profile or is a new user. Not dependent on any other tasks.

Sprint 09 (Weeks 03/04)

- Task 6: User profile pages (create user profile through form, view user profile, delete user profile) (Josh). Profiles will differentiate and identify users across many of the other functionalities present such as study groups and compatibility questionnaires for matching between users. Partially dependent on login integration (Task 5) of Sprint 08 since a "Create User Profile" page will display after a user logs in for the first time.
- Task 7: Study group pages (create study group through form, view suggested study groups, join study groups) (Dylan). These pages will pertain to the functionalities with study groups such as hosting or joining study groups comprising multiple students, which encapsulate the overarching goal of our application. Partially dependent on created user profiles (Task 6) of this Sprint since having a created account acts as a verification step for then allowing users to host or join these study groups.
- Task 8: Compatibility questionnaire pages (initiate questionnaire, fill out / edit questionnaire, delete questionnaire) (Alex). Users will be able to initiate and fill out their own compatibility questionnaires after creating their profiles, which include a set of parameters to match students based on the algorithm we decide upon. Partially dependent on the created user profile (Task 6) of this Sprint since even though the compatibility questionnaire and algorithm can be coded independently, users should not be able to fill out a questionnaire or match with other students until the profile exists.
- Task 9: Calendar integration without notifications (Jonathan). Here, we will confirm that we can actually integrate the calendar API prior to focusing on the calendar's subsidiary user stories like notifications across study group sessions. Not dependent on any other tasks.

Sprint 10 (Weeks 05/06)

- Task 10: Ticket support system pages (initiate ticket through form, view all tickets sent, view individual tickets, delete tickets) (Josh). The ticket support system helps users note bugs or issues with the system, including abuses being committed by other users. Once users begin filling out tickets, these will populate on a general ticket support page, viewable based on date in addition to the individual tickets generated. Partially dependent on the created user profile (Task 6) of Sprint 09 since even though this feature can be coded independently, we only want users with existing accounts to be able to send tickets.

- Task 11: Host and join study group session pages (host / edit meetings, join meetings, view meeting confirmation) (Dylan). Regarding these pages, study group members can choose to either host a meeting or join an existing meeting given a set of parameters like descriptions or datetimes. Dependent on the study group existing (Task 7) of Sprint 09 since study group sessions correspond to the particular study group wanting to do a study session.
- Task 12: Reserve room pages (reserve room, confirm room reservation, map integration) (Alex). Hosts of study group meetings can determine where they want to meet based on datetimes and available classrooms within IU buildings. The building of the classroom selected can then be displayed through the map integration for helping visualize where the building is on campus. Dependent on the existing study group (Task 7) of Sprint 09 and meeting (Task 11) of this Sprint since only hosts of study groups meetings can reserve rooms.
- Task 13: Notifications for calendar integration (view notification, create / edit notification, delete notification) (Jonathan). Users will now be allowed to interact with the group calendar by seeing the notifications present and how they apply to study group meetings. Dependent on the existing study group (Task 7) and calendar integration (Task 9) of Sprint 09 since we want notifications to populate for all members of a study group using the calendar API we integrate.

Sprint 11 (Weeks 07/08)

- Task 14: Initial iterations based on feedback and basic debugging. (Josh) Josh will begin reviewing feedback we received and start making appropriate changes. If there is not a lot to iterate, then the process of debugging will begin. Review of web pages and initial HTML/PHP code will be first on the list for debugging checks. Not dependent on any other tasks apart from existing web pages.
- Task 15: Database iterations and general debugging. (Dylan) Dylan will begin looking over database feedback and checking for bugs in this process, attempting various inputs in forms to see how our database responds. Not dependent on any other tasks apart from the existing database.
- Task 16: Alex will assist with any last minute features that need worked on and begin working on checking for bugs within our code. (Alex) This will pertain more so to web pages like Josh but will take a more broad approach to iterations outside of debugging. Not dependent on any other tasks apart from existing web pages.
- Task 17: Dummy data testing such as inserting into the API and reviewing output. (Jonathan) Jonathan will be testing this to ensure our system is properly functioning with the APIs we should have implemented at this point. Not dependent on any other tasks apart from existing API implementations.

DEMO (Week 09)

- Task 18: Complete last minute iterations and debug checks. (Josh) Josh will ensure that our debugging continues to reflect the overarching representation that we want for our

first project demonstration and especially prioritize this since we will want to avoid any bugs within our demo. Review JavaScript and PHP functions to ensure that they function correctly in relation to web pages. Not dependent on any other tasks apart from existing web pages.

- Task 19: Complete last minute iterations and debug checks. (Dylan) Dylan will need to continue debugging to also reflect how we want our project demonstration to be since we want holistic feedback based on the larger features rather than the bugs present. Briefly review database iterations from previous Sprint and API operations that Jonathan tested for Task 17. Dependent on API testing and dummy inputs (Task 17) from Sprint 11 since we will not be able to confirm any debugging required until these inputs occur.
- Task 20: Prepare project for first presentation and demo. (Alex) Although our application should be nearly usable at this stage based on implementations, iterations, and any debugging, we will need a distributed plan for demonstrating our project to others. Alex will focus on planning the demo presentation through example runs we can highlight, benefits for our target audience, and answers to potential questions testers may have. Partially dependent on all previous tasks for the demo presentation, but the demo itself will need to be flexible based on existing features or bugs present.
- Task 21: Prepare project for first presentation and demo. (Jonathan) Jonathan will consider potential visualizations or other appendix items we can provide for testers to and conduct most of the note-taking during the demo to identify any facets we can improve upon or need to reconsider based on the feedback received from others. Partially dependent on all previous tasks for the demo presentation, but the demo itself will need to be flexible based on existing features or bugs present.

Sprint 12 (Weeks 10/11)

- Task 22: Utilize focus groups for set of UX studies and make UI iterations based on the first project demo. (Josh) Here, Josh will conduct the first focus group study for the UX method as the first wave of feedback for making final iterations. The results should mostly pertain to the UI and allow him to make final iterative touches for the project without potentially jeopardizing established features. Partially dependent on the feedback generated from the demo presentation (Tasks 20 and 21) from the DEMO week.
- Task 23: Utilize interviews & think-aloud sessions for UX studies and make database iterations based on the other feedback from the demo presentation. (Dylan) To differentiate from focus groups, Dylan will conduct one-on-one interviews to obtain more granular insight for any slight modifications or improvements that can be made to the project. Partially dependent on the feedback generated from the demo presentation (Tasks 20 and 21) from the DEMO week.
- Task 24: Utilize observations for UX studies and consider other UX iterations outside of the UI prototyping. (Alex) Alex will conduct a set of observations, but this will be dependent on whether another UX study may benefit more given the extensiveness of the project itself at this point. Make slight modifications to any other UX considerations outside of UI prototyping such as components like fonts, images, and shapes while

staying within the scope of the Rivet framework. Partially dependent on the feedback generated from the demo presentation (Tasks 20 and 21) from the DEMO week.

- Task 25: Improve the prototype based on both interviews and think aloud sessions, as well as DEMO time feedback. (Jonathan) Jonathan will be implementing API call improvements if needed taking into account user feedback on the flow or usability of certain implemented or nono-implemented features of the calendar or meetings page from both presentations as well as additionally conducted interviews and UX methods. Partially dependent on the feedback generated from the demo presentation (Tasks 20 and 21) from the DEMO week.

Sprint 13 (Weeks 12/13)

- Task 26: Conduct a second set of UX studies using focus groups with iterations now implemented. (Josh) With a new set of participants acting as a focus group, Josh will employ different questions from the first one to consider other features that were overlooked during the first study. Make any additional slight modifications to the UI without compromising features already implemented and ready as a representation for the final project. Partially dependent on the first set of UX studies conducted (Task 22) of Sprint 12 since the questions generated will need to reflect updates or other topics outside of this study.
- Task 27: Conduct a second set of UX studies using interviews and think-aloud sessions; consider any other slight improvements. (Dylan) Dylan will find a new set of participants for interviews and think-aloud sessions using similar yet perhaps new questions to generate the insights from these UX studies compared to the first wave. Make any slight modifications to the database in relation to potential feedback offered during the demo presentation, but only do so with discretion from the team so as to keep the overall database intact during the final weeks. Partially dependent on the first set of UX studies conducted (Task 23) of Sprint 12 since the questions generated will need to reflect updates or other topics outside of this study.
- Task 28: Conduct a second set of UX studies observations for further UI iterations (Alex) Alex will find a new sample of users to conduct observation testing on alongside the data gathered from observing people during the DEMO in order to create improved prototype iterations. Partially dependent on the first set of UX studies conducted (Task 24) of Sprint 12 since the questions generated will need to reflect updates or other topics outside of this study.
- Task 29: Testing prototype iteration based on task 26 data with new and old stakeholders. (Jonathan) Jonathan, after polishing up the API calls as well as implementation (in areas of the applications that utilize the tool), will test the iteration with both previous tested users/stakeholders as well as new users/stakeholders to gain better insight as to whether the changes made created a better workflow/user use environment within the application.

Successful Teamwork

Communication Plan

Types of media that our group will use to communicate are GroupMe, Zoom, and Google Docs. We are all comfortable using these outlets of communication as students. Documents will be stored on Google Docs in a team shared folder. All team members will have access to our group documents on Google Docs and can edit them anytime they want. Collaboration will similarly be done through Google Docs and Zoom. As a team we will collaborate by assigning roles to assignments, utilizing each group member's strengths and weaknesses.

Every week, our team will hold a group meeting through Zoom. For the fall semester, we decided to meet after class on Wednesdays, as that time slot fits our schedules best given our differentiating schedules right now. Next semester, we will decide on a more appropriate time slot and will likely decide to meet more than once a week since the second semester will constitute the bulk of our synchronized work. Expectations of meetings include obviously that everybody attends if available so we can discuss, go over, and plan out group assignments; if somebody on our team is absent, we will record the meeting so they can go back and watch it whenever they are available to do so. Since we are recording our meetings, we do not need to take too many notes; however, all information in the Zoom chat will also be kept for documentation if need be.

Shared expectations for all team members include attending class and meetings on time unless a member has an emergency or informs the other members indicating that this member will not be present. Moreover, expectations for assignments will be that everybody works on their assigned section, getting it done prior to the deadline with proficient results demonstrated. If team members need help, it should be brought to the attention of the team as soon as possible so that we can proactively collaborate together on solutions.

Team members who are dealing with interpersonal issues among each other should be honest and converse with the team in a group meeting in order to either solve or find an intermediary agreement on said issues. We acknowledge that interpersonal issues are possible but will immediately mediate any criticisms on the personal characteristics of a member.

Training Plan

This project requires more cogs in order to work than projects we have faced before such as in INFO-I 211; for example, this project brings into mind external tools such as APIs in order to allow our application to better serve what we envision our product to do and to create the working prototype. This being said, many of the skills we need also come from our previous experiences within the informatics program and are either in need of de-rusting, polishing, or necessitating even more in-depth learning, perhaps by browsing literature on how similar products were created and what methods were followed as a guideline.

The specific skills we *do* need to learn or re-learn in no particular order thus becomes: HTML, PHP, CSS, Python, Javascript, SQL, API integrations, and finally (perhaps not concretely, however) Rivet Framework. These are the baseline skills that our group has identified as being crucial to generating a working prototype. However, having created this list, we acknowledge that it is not immutable, meaning that the skills needed can and almost certainly will be expanded upon with other hard or soft skills. Our training plan mainly revolves around each team member's cogent and "hard" skills, seeing as in-class lectures will primarily address the soft skills we need.

Finally, having identified the hard skills we need in order to create our product, it is imperative that we either learn them or reorient ourselves with the works we had been familiar with previously. Either way, the most easy and accessible path of action for this would be to browse external resources. External resources are the most accessible resource to anyone and prove to be effective when trying to brush up on skills that we may have forgotten. However, with this being said, simply reading up on coding is not always proficient enough, as experience is important and almost always remains the best way to learn the most difficult skills. This consideration, along with using external resources and the ITPs assigned to us in the class, will provide us additional exposure to what skills we need going forward. As for which kind of external resources we might use, we assume it would mainly be any (mostly online) literature addressing the hard skills we need like HTML or SQL, review from the bottom up, and finally more obscure topics that each team member denotes they might need to know. However, beyond just learning how each language works and the nuances, we may need to learn how to organize it effectively and efficiently to best suit the application we plan to build for our finalized prototype. We will achieve this by learning what other products like ours have done in order to achieve what they created as a product, which kinds of systems they used and proved to be effective, and what types of things to avoid. Perhaps even learning their general framework or system itself would be a great boon for our training plan, as we would subsequently be able to formulate our own system tailored to our purpose.

Risk Factors / Exposures

Although we believe that we have coordinated a strong foundation for the documentation, implementation, and deployment of our project so far, we acknowledge that circumstances can derail our project if we do not remain vigilant in deterring them. These may entail a combination of both technical and non-technical events, and because it is our responsibility to alleviate them together as a team, we need to initially enumerate a few risk factors that we believe are the most determinative in terms of extent or possibility for affecting our project. Below, we have highlighted four risk factors, which include how we will proactively mitigate them and apply backup plans if possible.

Firstly, one risk factor we noted is the feasibility of relegating and organizing our work on multiple repositories connected to our remote repository on GitHub. If we do not communicate our updates and which additions we have made, especially prior to commits and pushes we propagate to our remote repository, we could dismantle code that depends on sensitive

parameters or integrations to function. Likewise, any removals within our SQL database are permanent and cannot be recovered, which can cause incongruities between our past and current progress. To mitigate this for any additions we make, we will constantly communicate which files we will be updating or adding, note the versions for these files specified on GitHub, initialize new branches for any considerable work we complete at one time, and delineate any integrations in these files that could affect other files, therefore requiring manual review from team members. For worst case scenarios, we will look to export our database data before significant changes are made and commit often so that we can revert to previous versions.

As mentioned in one of our discussions, one risk could pertain to the actual security of our project once we begin to push our work onto our web application. If we manage PHP files to allow user inputs into forms, include executable code through JavaScript or other coding languages, and integrate our backend to an address facing the internet, it is imperative that we ensure we sanitize any inputs based on our SQL logic so that we inhibit the possibilities for exploits (Ex. code injection) within our web application. While we will not be integrating specific user data like real students, we will also have login integration and need to ensure the same methodology. Unfortunately, there may not be any immediate backup plans that we recognize we could utilize, but since we will have strong version control through GitHub, we would technically be able to recreate our project even if the timeframe is prolonged.

Apart from technical events, one non-technical risk we specified, dependent on our ability to coordinate meetings and work, is our differentiating schedules. If our schedules, both this and next semester, are disparate enough that we cannot meet until later times throughout the week, then this obviously runs the risk that we cannot ascertain what progress we have made or who is contributing when necessary. Although we have done a great job being active in our communication both in-person and online, we can more proactively mitigate this risk by planning out schedules in correspondence to Sprint timeframes so that we can alleviate most of our scheduling obstacles far in advance to due dates. Again, as a worst-case scenario backup plan, we can use intermediary correspondence and in-class discussion time so that we still have a strong idea of coordination even though it may not always be collective or together.

Lastly, we identify a second non-technical risk factor as any instances where one or more team members suddenly cease correspondence for concrete deadlines that we have established as a team, including assignment submissions. While we feel that we do not have much concern for this risk factor at the moment, it still certainly poses a hindrance if it does happen since it affects all team members who did contribute for a deadline. To concurrently mitigate this, we will try to memorize each other's plans based on our SCRUM meetings and vocalize short updates before working on a task for the team so that we have documentation of our tasks. Moreover, we will try to reach out and communicate to a team member if we feel that the person has been silent for an extended period of time (Ex. More than a week on top of not attending class). To conclude, as a backup plan, we should be able to cover if a team member does not correspond prior to a deadline since we will have been sufficiently aware of everyone's tasks; with multiple team members withdrawn, making the remaining work infeasible for the

other members, however, this would impel us to reach out to our associate instructor and potentially the instructors for guidance depending on the severity of the situation.

Acknowledgements

I-308 lecture/slides
Creation of the ERD diagram of the entities, attributes, relationships, and key constraints in our database. Looking back at our lecture and slide notes we used this information to design and document our database, as well as to visualize and understand the data and relationships within the database.

INFO-I 300 Lectures and Slides
Design process such as Wireframing, Prototyping, Sketching, User testing, observation, wizard of oz, focus Groups, Interviews, and Think-Aloud Sessions. In depth detail of each method can be pulled from lecture and slide notes.

INFO-I 370 Lectures and Slides
Three of our four group members are currently in this class. An extension of I-300, this class uses problem statement design such as identifying the problem you want to solve, defining the problem, identifying stakeholders and competitors, impact of the problem, and goals our project should accomplish. Lecture slides/notes helped provide a clear and focused direction for our team regarding our problem statement and design to ensure that the solution effectively addresses the problem and meets the needs of those who use our information system.

INFO-I 494 TA - Renu Jaiswal
Renu Jaiswal helped guide overarching project direction through weekly check ups in lecture and feedback for grading.

I-494 Other Group Feedback
Feedback from other teams helped in evaluating project decisions and ensuring that they align with the project goals and user needs. This helped avoid design, ethical, and ERD choices that may not be the most effective at meeting the needs of the users. Other groups can bring fresh perspectives which can help to spark new ideas and approaches. This helped our team explore different ways of creating our information system.

Appendix

The following pages contain additional items that support this document.

Focus Group 1

Date/Time: 11/15/22
Location: Zoom Virtual Meeting
Interviewer: Josh Hatfield
Subjects: Holt Henke, Ben Malone, Nick Mathein

Focus Group Questions:

- Q1: If you could join a study group within a course at IU, would you want to do so? Why or why not?
- Q2: How have your experiences been with study / academic groups as an IU student thus far?
- Q3: Did you ever become friends or better acquainted with other group members through study / academic groups?
- Q4: If you have been in study / academic groups, what were the main means for accomplishing your goals (Ex. Meeting online versus in-person, doing work synchronously or asynchronously, etc.)?
- Q5: Have you encountered any other barriers with study / academic groups outside of, perhaps, issues with group members? If so, what were these barriers?

Results:

- Q1: If you could join a study group within a course at IU, would you want to do so? Why or why not?
    - Holt
        - Yes, especially w/ upcoming exams. Having different perspectives really broadens the amount of knowledge you may have overlooked from the course
        - Easy way to make friends since you all have a pretty common goal
    - Ben
        - Maybe, but probably would during freshman year
        - Have a pretty defined friends group as a senior & doesn't need new study groups since friends are in a lot of the same courses
    - Nick
        - Like Ben said, maybe, depending on the course & whether I know anyone within it
        - Always nice to know at least one person within a course since it can alleviate a lot of the burden when studying
        - Common in coding classes where the exam study guide is somewhat ambiguous, so sharing ideas help
- Q2: How have your experiences been with study / academic groups as an IU student thus far?
    - Holt

        - Not great apart from this group. Either I feel inclined to join with people I know, & we're not always productive, or I'm stuck with people who don't contribute an equal amount
        - A lot of times, communication is uneven. Some members will communicate a lot and go silent or vice versa
        - This leads to a lot of quarrels within the group
    - Ben
        - To add onto Holt, communication doesn't work because messaging becomes overwhelming
        - Have a lot of obligations & don't want to sit reading a long block of text on top of the teamwork needed
    - Nick
        - Yes, because it's hard to keep team members motivated. I can't be overwhelmed trying to be both the team facilitator & member doing work if no one else is participating
        - Again, organization is hard w/ communication. I liked Holt saying communication is uneven because it feels that way
- Q3: Did you ever become friends or better acquainted with other group members through study / academic groups?
    - Holt
        - Yes, I've met a few friends that way
        - I think it happens more with academic groups because you're sort of obliged to interact w/ the others, not that this is a bad thing
        - Like I said w/ study groups, you have a pretty common goal, so this can motivate you to be better friends if you want to accomplish that goal
    - Ben
        - I'd say somewhat, because most of the time I group w/ people I already know
        - I haven't really had good experiences w/ people I don't know, so I sort of just stop communication after the group ends
    - Nick
        - I did with some people during freshman year, but COVID somewhat ruined a lot of communication I had w/ these people since they were not my main friends
- Q4: If you have been in study / academic groups, what were the main means for accomplishing your goals (Ex. Meeting online versus in-person, doing work synchronously or asynchronously, etc.)?
    - Holt
        - Obviously during COVID, a lot was done online asynchronously
        - This caused its own issues since it was hard to visualize each other's work & keep each other on task
        - We still meet a lot online since it's seamless for the work we do, but there's still a lot of issues w/ members not attending meetings or forgetting

    - Ben
        - On top of that, I don't think people prioritize remembering online meetings since it's different than meeting in-person
        - There feels like a lot more obligation to meet in-person
        - Do work mostly asynchronously, but synchronous work helps a lot when we need to be on the same page or are more in discussion about something
    - Nick
        - Study groups need to be synchronous since that provides the most benefit. Academic groups not so much since most of the work needs to be distributed and done individually to speed up time
- Q5: Have you encountered any other barriers with study / academic groups outside of, perhaps, issues with group members? If so, what were these barriers?
    - Holt
        - Yes, different schedules always provide a lot of issues. It's hard to see each other's schedule & build meetings around that
    - Ben
        - Again, a lot of it is technology-oriented, so having to much text & communications for delegating things
        - Hard to visualize what is being done & whether there's any personal benefit for being in the group
    - Nick
        - If we're working online to do presentation slides, it's sort of hard to practice together since it doesn't simulate all presenting together
        - Some of the technology is limited like brainstorming - could be better in-person

Focus Group 2

Date/Time: 11/19/22
Location: Zoom Virtual Meeting
Interviewer: Josh Hatfield
Subjects: Jacob Hatfield, Elijah Hatfield, Jonah Sullivan

Focus Group Questions:

- Q1: Do you prefer selecting who you wish to be with for groups or have the professor / teacher select this for you? What do you find as the pros and cons for each?
- Q2: What were (are) your expectations for study / academic groups going into college? Did (Do) you feel that they would be utilized a lot and could help with your courses?
- Q3: What helps encourage or generate “good” and “effective” group environments?
- Q4: Outside of group members simply not contributing, are there other factors that may cause “bad” group environments?
- Q5: If you’ve used technologies to manage groups (Ex. Microsoft Teams, GroupMe, etc.), what do you like and dislike what they offer as they exist now?

Results:

- Q1: Do you prefer selecting who you wish to be with for groups or have the professor / teacher select this for you? What do you find as the pros and cons for each?
    - Jacob
        - I prefer selecting who I wish to be with for groups, but this depends on whether I know anyone in the class
        - Selecting allows you to better predict the contributions that will be made my the group members
        - Teacher selecting for you helps make sure that there’s more accountability & professionalism w/ the group; people will be a bit more serious w/ each other
    - Eli
        - Prefer selecting who I wish to be with
        - As Jacob said, selecting allows you to know your group better, but to add on, you’ve already removed a lot of the initial awkwardness that exists w/ people you don’t know
        - Teacher selecting for you may know your strengths/weaknesses, so puts you in a group that maximizes these for a good group
    - Jonah
        - Definitely the first option
        - Con not said for selecting is that you can choose to group w/ friends, so you may not contribute the level of work you need even though you can since you’re comfortable w/ them
        - Con w/ teachers selecting is that group members that don’t mesh well will lead to bad groups & maybe lots of arguments

- Q2: What were (are) your expectations for study / academic groups going into college? Did (Do) you feel that they would be utilized a lot and could help with your courses?
    - Jacob
        - Felt that I would maybe struggle at first but then find a niche group that would be in many of my classes; I could group w/ them a lot then
        - I knew that some classes had hundreds of people, but I didn't think they would be used as much as I know now. Labs will tend to always have some partner or group that you need to dedicate a lot of time to
    - Eli
        - I think that they'll become a bigger part compared to high school since this will be important in the future
        - Professors basically want to see how you perform in a group on a larger project
        - Study groups will also be common since courses will have a higher standard than high school. I've heard people are more receptive to being friends w/ people they don't know since it's a new experience for everyone
    - Jonah
        - I actually like to do a lot of studying myself since I know what I still need to learn, so having an overdependence on them could hurt the college experience
        - You'd have to deal w/ someone you don't like for the entire semester if you're put w/ them in the same group
- Q3: What helps encourage or generate "good" and "effective" group environments?
    - Jacob
        - Communication is a must
        - Group members understand their responsibilities & can help each other
        - The group is on the same page like Jonah said
    - Eli
        - Group members are receptive to helping each other
        - They set goals to meet so that the group isn't a waste compared to what could be accomplished
    - Jonah
        - Everything is organized, so the group is on the same page
        - Schedules match well so that there is an expectation on what needs to be done when
- Q4: Outside of group members simply not contributing, are there other factors that may cause "bad" group environments?
    - Jacob
        - Bad communication - we can't assume that someone is getting something done if they don't communicate to others
        - When a group member is toxic, so the member doesn't really fit well w/ the group
        - People have different hobbies & interests

    - Eli
        - Maybe favoritism between particular members so that others are left out
        - Not knowing when something is due
    - Jonah
        - Lack of communication
        - Obviously people who don't want to be in the same group together
- Q5: If you've used technologies to manage groups (Ex. Microsoft Teams, GroupMe, etc.), what do you like and dislike what they offer as they exist now?
    - Jacob
        - Haven't used Microsoft Teams much, but liked how easy it was to set up and understand; didn't like the lack of customization, felt a bit lacking
        - GroupMe is okay but isn't the best for "managing" groups, doing labs require lots of documents to be shared & GroupMe is not the best for this; still easy to set up among people
    - Eli
        - Haven't use any of those, but could consider Discord; likes # of features offered but feels like they don't fit together well or feel really different
    - Jonah
        - Although not for managing groups, likes Canvas because it has a simple interface and allows me to see all the people in a course / group; thinks the group aspect is pretty lacking since even sending messages is hard

Think-Aloud 1

Date/Time: 11/16/22
Location: In-Person
Interviewer: Dylan Johnson
Subject: Mikayla Knierim

User Tasks

- **Profile**
  - After login please create your profile
  - From the homepage, view your profile and edit it
  - Delete your profile and information
- **Compatibility Questionnaire**
  - From the login page, please navigate to and initiate the compatibility Questionnaire
  - From the home page, please find and edit or delete a Compatibility Questionnaire
- **Groups / Meetings**
  - From the home page please:
    - Join a group
      - Then view it
    - Create a group
      - Then view it
    - Leave group
    - Host a meeting
      - Then edit or delete it
    - Join a meeting
- **Calendar**
  - From the home page, please navigate to the calendar
    - Add an event/notification through the calendar
    - View an event/notification
- **Ticketing system**
  - From the homepage, please navigate to the ticket support page
    - Create a ticket
    - Then view it
    - Then delete it
- **Room reservation**
  - From the homepage please reserve a room

Results:

- **Profile**
  - Easy to find and complete this task
  - Confused as to why clubs are an option for your profile
  - User asked if the courses section would be populated by their current courses or all courses

- **Compatibility Questionnaire**
    - Took a second to find but once found all tasks were straightforward
    - Would like to see more questions, but they seem purposeful
- **Groups / Meetings**
    - No problems hosting a meeting or joining a group
    - Clear ‘view’ and ‘join’ buttons helped a lot
    - Confused on where ‘creating a group’ would take the user
- **Calendar**
    - Group header at the top confused the user whether they were in a group or personal calendar
    - Differentiated colors for events was helpful
- **Ticketing System**
    - Layout was very easy to navigate
    - No issues with completing tasks
- **Room Reservation**
    - Critique on why the room reservation wouldn't just come with the create meeting function so we didn't have to reserve a room on one page and tie that room with the meeting.

Think-Aloud 2

Date/Time: 11/14/22
Location: In-Person
Interviewer: Jonathan Yen
Subject: Brian Yu

User tasks

- **Profile**
  - After logging in please create and edit your user profile
  - From The homepage please view your profile and edit it
  - Delete Your profile
- **Compatibility Questionnaire**
  - From the login page, please navigate to and initiate the compatibility Questionnaire
  - From the home page, please find and edit or delete a Compatibility Questionnaire
- **Groups / Meetings**
  - From the home page please:
    - join a group
      - Then view it
    - Create a group
      - Then view it
    - Leave group
    - Host a meeting
      - Then edit or delete it
    - Join a meeting
- **Calendar**
  - From the home page please navigate to the calendar
    - Add an event/notification through the calendar
    - View an event/notification
- **Ticketing system**
  - From the homepage please navigate to the ticket support page
    - Create a ticket
    - Then view it
    - Then delete it
- **Room reservation**
  - From The homepage please reserve a room

Results:

- **Profile**
  - Thought that the clubs tag would be a little weird, as we already bio section
  - Also didn't know if the courses were current courses being taken or all courses ever taken

    - They also didn't really understand why we needed a separate profile for the application when we had already signed in with the IU login
- **Compatibility Questionnaire**
    - Seemed pretty straight forward if not a little bit lacking in questions and and no real ending screen
- **Groups / Meetings**
    - Was confused on whether or not creating a group redirected them to the home page or to viewing the meeting.
- **Calendar**
    - Seemed to be unclear as to how to click on an event
    - There was confusion on whether or not there was a group calendar or a personal calendar
- **Ticketing System**
    - No problems here
- **Room Reservation**
    - Critique on why the room reservation wouldn't just come with the create meeting function so we didn't have to reserve a room on one page and tie that room with the meeting.

UX Interview 1

Date/Time: 11/16/22
Location: In-Person
Interviewer: Dylan Johnson
Subject: Mikayla Knierim

Question Guidelines (most questions if not all are followed up with a why question):

- **Before**
  - Have you ever had problems finding a group to study with when you wanted one?
  - Have you ever found it difficult to schedule meetings with a group of people and find some location to study/work at?
  - What parameters do you think of when you are setting up a meeting yourself, or attending a meeting (i.e. if location, time, class, dress, presentation, etc.)?
  - How often do you find it difficult to create meetings for regular work / study groups?
  - How often do you find it difficult to find a location to conduct work/study sessions at?
  - Given that you make or find an application for a group meeting scheduler kind of application, what features would you implement and how?
- **After**
  - How did our application fill the role of a group meeting scheduler?
  - What would you change, add, or remove UI wise?
  - What works and what doesn't?
  - What was intuitive or learnable? What wasn't?
  - What other applications does this seem to be similar to in your experience? Should we incorporate designs you are familiar with or a unique one of ours?

UX Interview 2

Date/Time: 11/14/22
Location: In-Person
Interviewer: Jonathan Yen
Subject: Brian Yu

Question Guidelines (most questions if not all are followed up with a why question):

- **Before**
  - Have you ever had problems finding a group to study with when you wanted one?
  - Have you ever found it difficult to schedule meetings with a group of people and find some location to study/work at?
  - What parameters do you think of when you are setting up a meeting yourself, or attending a meeting (i.e. if location, time, class, dress, presentation, etc.)?
  - How often do you find it difficult to create meetings for regular work/study group?
  - How often do you find it difficult to find a location to conduct work/study sessions at?
  - Given that you make or find an application for a group meeting scheduler kind of application, what features would you implement and how?
- **After**
  - How did our application fill the role of a group meeting scheduler?
  - What would you change, add, or remove UI wise?
  - What works and what doesn't?
  - What was intuitive or learnable? What wasn't?
  - What other applications does this seem to be similar to in your experience? Should we incorporate designs you are familiar with or a unique one of ours?