

# Final Report

Team UMAaaS: Study Abroad Experience Review System

Arka Bose  
Molly Frost  
Anuj Ketkar  
Abhishek More  
Uma Sarkar  
Sarah Sotelo

1. Project Summary
2. User Stories
3. Approach to Legacy Project
4. Team Roles Throughout the Project
5. Scrum Iteration Summaries
6. Customer Meetings
7. Behavior and Test Driven Development
8. Tool Issues
  - a. Heroku
  - b. AWS and LightSail
  - c. Cloud9
  - d. Github
9. Tools and Gems
10. Repo Description
11. Links

## Project Summary:

This project is a convenient website for all Aggies who are interested in, have been on, or are currently on a study abroad trip. It is also for the faculty managing the study abroad trips and the study abroad office. When coming into possession of the project, the website had the bare needs: a main page where we could select a program to view, a program page where you can add tips or add/rate an experience, and an emergency contact page. The customer came to us asking for more from the website. The customer wanted the website to be completely configurable, a way to flag posts by customers, an admin page, and overall a better looking website.

We have gone above and beyond to meet these needs. We have added flags to tips and experiences. We have created an admin page where you can add a program, add participants to a program, make people admin in a program, and delete flagged items. We have modified how you add experiences so each experience is able to have a title, description, and a picture. We have added profiles for participants where you can see their profile picture and what they have posted. Lastly, overall, we have changed the design of the website to make a cleaner and more sophisticated look.

## User Stories:

### Implemented:

1. Feature: Move add experience button Point(s): 1
  - a. As a user adding an experience  
So that I can understand more easily how to add an experience  
I want a more visually appealing ‘add experience’ button that stands out better
2. Chore: When on the experiences page, add a picture on the banner Point(s): 1
  - a. As a user adding an experience  
So that I can have instant context on what trip’s page I am on  
I want a banner picture that identifies my location
3. Feature: Update Experience card to include title and description Point(s): 2
  - a. As a user adding an experience  
So that I can better define an experience  
I want to see both the title and description of the experience on the main page
4. Task: Refactor database to match uml changes (Sarah Sotelo) Point(s): 3
  - a. As a programmer  
So that I can have full confidence in my database meeting all of the project’s needs  
I want to refactor the database to match the uml diagram that the team discussed and created.
5. Bug: Put delete button back (Arka Bose) Point(s): 1
  - a. As a user  
So that I can delete an experience that I’ve created  
I want a functioning delete button
6. Bug: Make it so that the search feature searches both titles and descriptions of experiences instead of just description (Anuj Ketkar) Point(s): 3
  - a. As a user  
So that I can have the most comprehensive search experience as possible  
I want the search functionality to search both the title and descriptions of an experience
7. Functionality: Assign students and faculty to programs in admin portal (Abhishek More) Point(s): 1
  - As an admin  
So that I can assign users to programs  
I want to be able to assign students and faculty to programs in the admin portal
8. Functionality: Create study abroad programs: (Anuj Ketkar) Point(s): 1
  - As an admin  
So that I can create new study abroad programs

I want to have the exclusive ability to create new study abroad programs in the admin portal

9. Functionality: change a user's role in admin portal (Abhishek More) **Point(s): 1**

As an admin

So that I can account for changes in faculty and student lists

I want to be able to change a user's role within a program in the admin portal

10. Bug: Fix Program Dropdown (Sarah Eilene Sotelo) **Point(s): 5**

- a. As a user

So that I can more easily access and view the programs I am a part of

I want to be able to select the different programs more readily

11. Feature: Add Photo to Experience (Anuj Ketkar) **Point(s): 5**

- a. As a user

So that I can share a picture of my relevant experience with other users

I want to be able to add photos to my experience if I have some to share

12. Bug: Fix Crashes (Abhishek More) **Point(s): 1**

- a. As a user

So that I can access and view everything without running into any errors

I want to be able to explore the site and interact with features without technical difficulties

13. Bug: Fix Where Add Participant Button Is (Abhishek More) **Point(s): 1**

- a. As a user

So that I can add participants to any program

I want to be able to add participants to any program without having difficulty on locating where to do so

14. Bug: Delete Button for own User's Posts (Abhishek More) **Point(s): 1**

- a. As a user

So that I can remove my own posts whenever I choose

I want to be able to remove any of my own posts in case it is not relevant or it's inaccurate

15. Bug: Add Program Title to 'Add Participants' Screen (Abhishek More) **Point(s): 1**

- a. As a user

So that I can easily see which program I'm adding participants to

I want to be able to add participants to the right program without any difficulty

16. Bug: User can't flag their own post (Abhishek More) **Point(s): 1**

- a. As a user

So that I don't accidentally flag myself, and potentially get in trouble

I want to be able to add posts without the worry of flagging them on accident

17. Feature: Add search function to users page (Arka Bose) **Point(s): 2**

- a. As an admin

So that I can find users more quickly

I want to be able to find specific users through a search bar without scrolling and reading every single user for the right one

18. Functionality: create users in faculty portal (Arka Bose) **Point(s): 2**

As faculty

So that I can add new students to my program without involving admin

I want to be able to create users in my program

19. Functionality: Remove users in the faculty portal (Abhishek More) **Point(s): 1**

As faculty

So that I can remove students from my program without involving admin

I want to be able to delete users in my program

20. Functionality: Fixed 'Where to?' button (Sarah Eilene Sotelo) **Point(s): 2**

As a user

So that I can view the button better

I want to be able to click on 'Where to' and be confused about the extra space

21. Functionality: Fixed look of 'Add Participants' button (Sarah Eilene Sotelo) **Point(s): 2**

As a user

So that I can view the button better

I want to be able to click on the button and have the text show up properly

22. Functionality: Ban users in faculty portal (Abhishek More) **Point(s): 1**

As faculty

So that I can block users who are not acting appropriately

I want to be able to block (put on read only access) users in my program

23. Testing: Increase test coverage on the portals controller (Arka Bose) **Point(s): 3**

a. As a programmer

So that I have more confidence in the robustness of the portals controller in the legacy program

I want to increase test coverage for portals\_controller.rb to at least 90%

24. Feature: Increase test coverage on experience.rb (Uma Sarkar) **Point(s): 2**

a. As a programmer

So that I have more confidence in the robustness experiences and their associated arrays of tags

I want to increase test coverage for experience.rb that we started with to at least 90%

25. Testing: Test coverage on the reviews controller **Point(s): 1**

a. As a programmer

So that I have more confidence in the robustness of the reviews controller in the legacy program

I want to increase test coverage for reviews\_controller.rb to at least 80%

26. Testing: Increase test coverage on the routes **Point(s): 1**

a. As a programmer

So that I have more confidence in the robustness of the routes in the legacy program and eliminate routes that aren't needed

I want to increase test coverage for paths.rb to at least 90%

27. Functionality: Add multiple people to program at once (Uma Sarkar) **Point(s): 2**

As faculty and admin

So that I can multiple people at the same time without having to type each user in one at a time.

I want to be able to add multiple people on the same line separated by a comma to a program.

28. Feature: Flag tips (Uma Sarkar & Molly Frost) **Point(s): 5**

As a user

So that I can avoid and prevent others from seeing irrelevant, inappropriate, or spam content

I want to be able to flag tips if the need arises

29. Feature: Make pictures smaller in experiences (Arka Bose) **Point(s): 2**

As a user

So that pictures do not take up much space when I add them to my experiences.

I want to be able to post pictures and view them with a reasonable size.

30. Feature: View User's Profile (Anuj Ketkar) **Point(s): 3**

As a user

So that I can view any user's profile and see their activity (posts).

I want to be able to more easily access any user's posts through their profile

31. Functionality: Unauthenticated users can't view posts (Abhishek More) **Point(s): 2**

As an unauthenticated user,

So that I can be notified when I am accessing a restricted experience,

I want to receive an alert when I click a restricted post without crashing

32. Functionality: Allow background photo changes in admin portal (Anuj Ketkar) **Point(s):**

**5**

As an admin

So that I can change the background photo for any program

I want to be able to upload or remove background photos for any program that exists in the admin portal

33. Functionality: Complete access as admin without assignment to a program (Abhishek More) **Point(s): 1**

As an admin

So that I can access, view, and edit everything for any program

I want to be able to manage all programs without having to be assigned to any

34. Functionality: Review flagged content for all programs in admin portal (Arka Bose) **Point(s): 2**

As an admin

So that I can handle inappropriate or spam posts/comments/experiences in all programs

I want to be able to review content that is flagged 3+ times and decide to remove or leave it in the admin portal

35. Feature: Review flags in faculty portal (Arka Bose) **Point(s): 1**

As faculty

So that I can handle inappropriate or spam posts/comments/experiences in all programs

I want to be able to review content that is flagged 3+ times from my programs and decide to remove or allow it in the faculty portal.

36. Functionality: Shouldn't be able to rate twice when commenting (Abhishek More)

**Point(s): 1**

As a user

So that I don't re-rate everytime I comment when I've already done so

I want to be able comment without having to rate from the second comment onwards

37. Feature: Enhance User's Profile (Anuj Ketkar) **Point(s): 2**

As a user

So that I want to view my profile and other's profile and see/access their posts.

I want to be able to have an aesthetically pleasing profile page and see the posts I made.

38. Feature: Flag experiences (Uma Sarkar & Molly Frost & Abhishek More) **Point(s): 2**

As a user

So that I can avoid and prevent others from seeing irrelevant, inappropriate, or spam content

I want to be able to flag experiences if the need arises

39. Design: "View Another Program" moved (Abhishek) **Point(s): 1**

As a user

So that the overall design is more visually appealing

I want the "View Another Program" button moved to below the image banner

40. Design: "Sort By" dropdown moved (Abhishek) **Point(s): 1**

As a user

So that the overall design is more visually appealing

I want the "Sort By" dropdown to be below the line instead of above it

41. Design: Search bar and Location Dropdown Moved (Abhishek) **Point(s): 1**

As a user

So that the overall design is more visually appealing

I want the location dropdown to be inline with a centered search bar.

42. Functionality: Make banner redirect to mainf page of the current program (Uma Sarkar)

**Point(s): 1**

As a user

So that I can get back to the main page of any program easier while browsing through the different experiences

I want the image banner to redirect me to the main page of the current program

43. Bug: Make sure that the dropdown list of programs inside the user profile only includes active programs **Point(s): 1**

As an admin

So that I can only assign users to programs that are active

I want the associated dropdown list to only show programs that are active

44. Functionality: Add a “show” option in the admin portal program page (Uma Sarkar)

**Point(s): 1**

As an admin

So that I can easily view students assigned to a study abroad program

I want there to be an option to view the students assigned without having to edit.

45. Functionality: Shouldn’t be able to rate twice when commenting (Uma Sarkar) **Point(s):**

**1**

As a user

So that I don’t re-rate everytime I comment when I’ve already done so

I want to be able comment without having to rate from the second comment onwards

46. Bug: Make sure that the rating system is consistent (all stars instead of some stars and some thumbs ups) (Sarah Eilene Sotelo) **Point(s): 1**

As a user

So that I can rate experiences with a uniform scaling system

I want to be able to rate with no confusion, and see the consistency of the rating system

There weren’t any changes made to the user stories. However, some of them were revised to break up the bigger tasks into smaller functions which will be better explained in the ‘Not Implemented’ section below.

### **Not Implemented:**

1. Feature: Add Faculty Users to tripAgvisor

As a faculty member on a study abroad program

So that I can manage/monitor the students on the same program I am assigned to

I want to be able to have the same features as a student and on top of that be able to edit/delete posts or ban users according to the situation

2. Feature: Add map feature to tripAgvisor

As a study abroad student

So that I can share specifically where the experience/places I've visited is with other study abroad students, and also see what experiences are in my vicinity

I want to pinpoint the location of where I go whenever I add my experiences and be able to view a map containing all of the experiences

3. Feature: Favoriting/Interested/Recommended/Done feature

As a study abroad student

So that I can help other study abroad students of the same program can consider if they would like to do the same activity

I want to mark an experience as favorited, interested, recommended, or done

4. Feature: Add photos to tripAgvisor

As a study abroad student

So that I can share the sites I've visited with other study abroad students

I want to add images to the tripAgvisor feed whenever I add my experiences

5. Feature: Update tag features in tripAgvisor (study abroad student)

As a study abroad student

So that I can filter experiences by category when looking for new experiences

I want to be able to tag an experience as a number of predefined categories (ex: restaurant, outdoor activities, museum, gallery, etc)

6. Feature: Update tag features in tripAgvisor (faculty)

As a faculty member on a study abroad program

So that I can manage experience categories

I want to be able to review, merge, or create new tags customizable to the study abroad location

7. Feature: Make contacts configurable in faculty view (faculty)

As a faculty member or administrator for a study abroad program

So that I can keep contact information as updated as possible

I want to be able to edit contact information as the need arises

8. Feature: Create faculty console for all admin tasks

As a faculty member

So that I can manage lists of verified students in one place

I want a faculty console where I can do all administrative tasks in one place

9. Feature: Bookmarks/Favorites page

As a study abroad student (current or prospective)

So that I can view all of the experiences that I've done/am interested in in one place

I want a bookmarks or favorites page

10. Feature: Increase test coverage

As a programmer

So that I have more confidence in the robustness of the legacy program

I want to increase test coverage for the legacy code that we started with

11. Feature: Flag comments (Molly Frost)

As a user

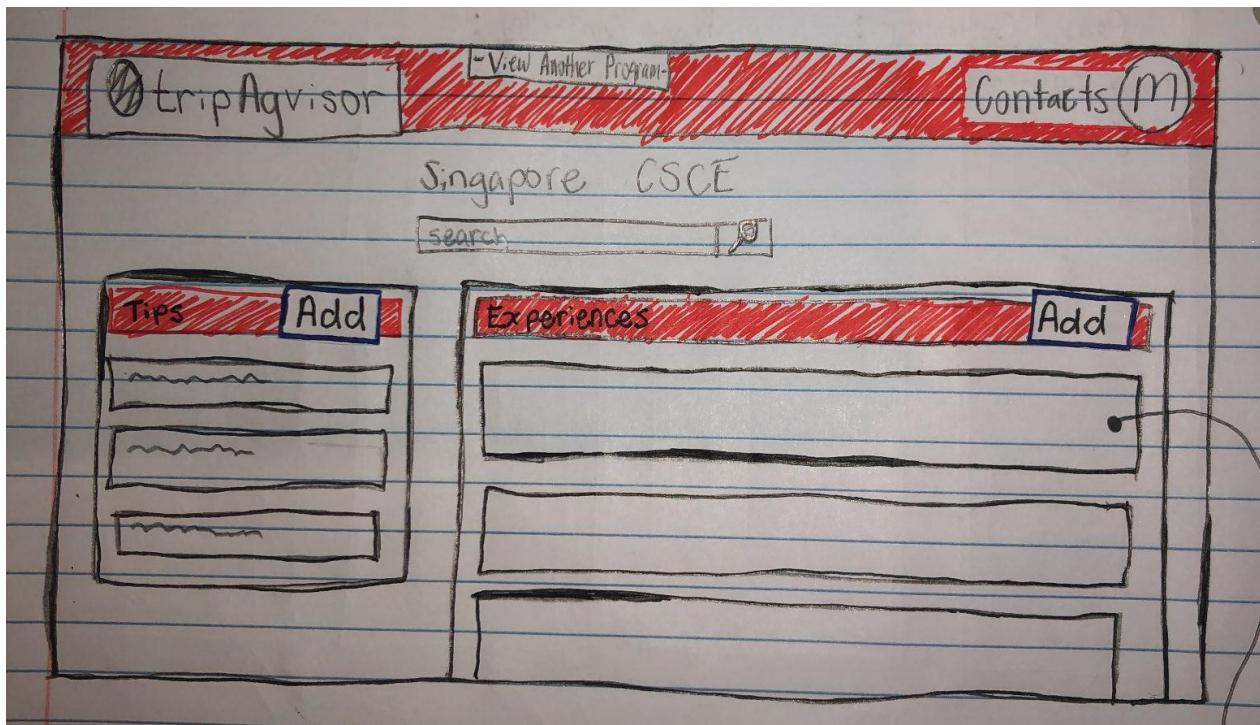
So that I can avoid and prevent others from seeing irrelevant, inappropriate, or spam content

I want to be able to flag comments if the need arises

All of these user stories, except one, were not implemented because they were created in iteration 0 and were too high level for this project. We have broken these user stories into much smaller problems/features in the later iterations. The last user story, ‘Flag comments’, was not implemented due to the complexity and lack of time.

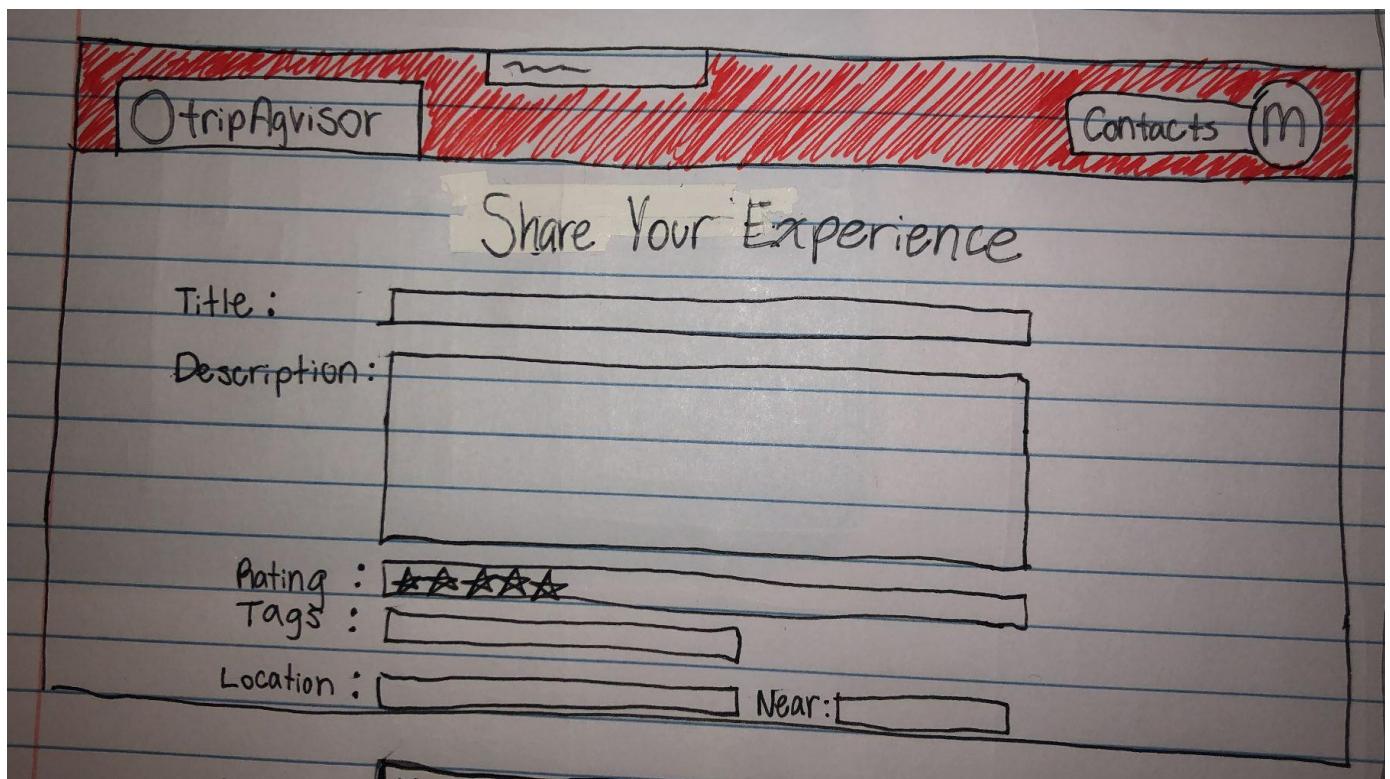
UI Mockups:

Edit Add “Tips” and “Experiences” Buttons:

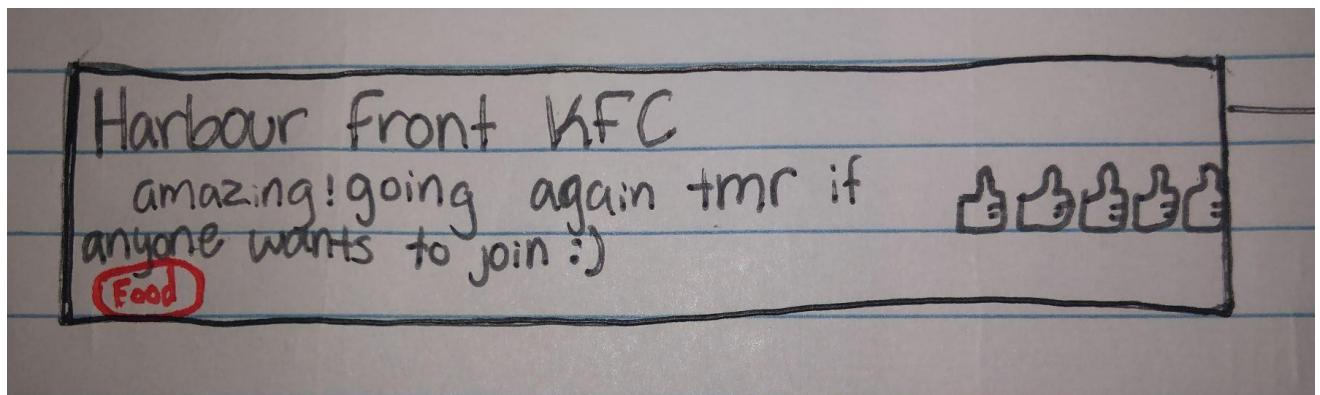


To the lo-fi UI, we modified the add button. In iteration 0, the Add button was not clearly visible. To fix this, we changed the “Experiences” and “Tips” titles to a solid color background and added a box around the add button. This creates a more distinct division between the two indices.

Create a Title and Description for Adding an Experiences:



For adding a title to an experience, we added a title text box at the top and a description text box underneath it. The next picture is how the experience will show on the page. The title will be in large text and the description will be in smaller text underneath it.



## Admin & Webmaster console design

The screenshot shows the tripAgvisor Admin & Webmaster console. At the top, a banner states: "Admins can delete any tips/comments/experiences, users can only delete their own". The main header includes "tripAgvisor", "--View Another Program--", "Contacts", and an "Edit background picture for a program (admin)" button.

The interface features a search bar labeled "Observation". Below it, there are two sections: "Tips" and "Experiences".

- Tips:** A list of user-generated tips. One tip has a white flag icon next to it, while others have maroon flags. A blue arrow points to one of the flagged tips with the annotation "Tip not flagged".
- Experiences:** A list of user-generated experiences. One experience has a white flag icon next to it, while others have maroon flags. A blue arrow points to one of the flagged experiences with the annotation "Experience flagged".
- Comments:** A section showing a comment from "Sarah Elene Sotelo Jan 11, '23" with an average rating of 5.0. A blue arrow points to the comment with the annotation "Comment flagged".

The picture above shows what an admin and webmaster console would look like. This also applies to the faculty if they are assigned to this program. There are now flags that are white if not selected, and filled in with maroon if they are. The search bar can now search by title (it already searches by description, tips, and tags). Admins/Webmasters are also able to delete experiences, tips, and comments by other users. There is also an edit background picture feature.

The screenshot shows the "CREATING NEW USER (ADMIN)" page. The header includes "tripAgvisor", "Admin", "Contacts", and a user profile icon.

The main content area features a banner with the text "New Participant". Below it, there is a form for creating a new participant:

- A "Participant Email" input field with a blue arrow pointing to it and the annotation "email creates new participant".
- A "Faculty" checkbox with a blue arrow pointing to it and the annotation "can mark user as faculty".
- A "Save" button.

At the bottom of the page, there is a link "View all participants in a program" with a blue arrow pointing to it and the annotation "in a program".

The picture above shows the Admin/Webmaster console when creating a new user by email on the program page. They are also able to select if they'd like to mark that user as a faculty for the program. This page also allows viewing of other participants, faculty or not, as well as remove users from the program.

**REVIEW FLAGS (ADMIN)**

--View Another Program--

Contacts

tripAgvisor

Singapore CSCE Wintermester

search

Can also sort by # of flags and done flagged

Sort by: Type

**Flags**

Description	Creator	Date Created	Flagged By	Date Flagged	# of Flags	Type	Action
If you're a student it's about \$22	Sarah Ellene Sotelo	January 11, 2023	Uma Sarkar	January 11, 2023	1	Comment	Warn
Marina Bay Sands Observation Deck	Sarah Ellene Sotelo	January 11, 2023	Uma Sarkar	January 11, 2023	1	Experience	Approve
Do not litter in Singapore! I just did and was...	Anuj Ketkar	January 11, 2023	Uma Sarkar	January 11, 2023	2	Tip	Ban

The picture above shows the Admin/Webmaster console when viewing the flags from experiences, comments, and tips on a program page. This can be filtered by type, number of flags, and date. There are also three dropdown actions the admin/webmaster can make which are to warn the user, approve of the flag, or ban the user.

**CREATE NEW PROGRAM (ADMIN)**

Admin Contacts

tripAgvisor

New Study Abroad Program

Program Name:

Country:

City:

Instead of Location and Region, it will be Country and City.

Save

The picture above displays the Admin/Webmaster console when creating a new study abroad program. Instead of what was previously ‘Location’ and ‘Region’, we are replacing them with ‘Country’ and ‘City’.

The screenshot shows the 'EDITING USER (ADMIN)' page for a user named 'Sarah Eilene Sotelo'. The top navigation bar includes 'Admin' and 'Contacts' links. The main content area features a banner image of the Singapore skyline at sunset. On the left, there's a sidebar titled 'Admin Functions' with options: 'Demote Admin' (highlighted with a blue arrow), 'Ban User' (highlighted with a blue arrow), and 'Save'. To the right, there are fields for 'Email Address' (s.sotelo@tamu.edu) and 'Abroad Program' (set to 'Singapore CSCE Wintermester'). A blue arrow points from the 'Ban User' link to the 'Ban User' field in the sidebar. Another blue arrow points from the 'Save' button to the 'Abroad Program' dropdown.

For the picture above, admins/webmasters are able to promote users to admin or demote them to a normal user. Banning users is also an option. On this page, they are also allowed to assign users to a specific study abroad program.

The screenshot shows the user profile page for 'Sarah Eilene Sotelo'. The top navigation bar includes 'Admin' and 'Contacts' links. The profile picture is a circular image of a flower. The user's name, 'Sarah Eilene Sotelo', is displayed prominently below the picture. An 'Edit' link is located just below the name. Below the name are two large, rounded rectangular boxes containing the text 'Sarah's Post #1' and 'Sarah's Post #2' respectively.

The picture above shows an example of what a user's profile page would look like. In this profile page, there will be the user's name, avatar, and all of their posted experiences, more easily findable than by search.

This screenshot shows a user profile page for 'Singapore CSCE Wintermester'. The top navigation bar includes 'Admin' and 'Contacts' buttons. Below the title, there are buttons for 'View Another Program', 'Search', 'Participants', and 'Flags'. The main content area is divided into two sections: 'Tips' and 'Experiences'. The 'Tips' section contains three items: 'Tip #1', 'Tip #2', and 'Tip #3'. The 'Experiences' section contains one item: 'Experience #1'. Each section has an 'Add' button at the top right. A 'sort by' dropdown is present in both sections.

The picture above shows our most recent look of the program page with experiences and tips. The difference between this look and the previous is the 'Search' bar and 'View Another Program' button placement, and the addition of the 'Participants' and 'Flags' buttons.

This screenshot shows a program listing page titled 'Study Abroad Programs'. At the top, there is a 'tripAgvisor' logo, 'Admin' and 'Contacts' buttons, and a user profile icon. Below the title, there is a 'Study Abroad Programs' heading. On the left, there is a legend for 'Disabled' programs and a checkbox for 'Show Disabled Programs'. On the right, there is a table listing four study abroad programs:

Name	Location	Region	Show	Edit
Singapore CSCE Wintermester	Singapore	Southeast Asia	Show	Edit
Greece CSCE Summer	Athens, Greece	Europe	Show	Edit
Vietnam CSE	Hanoi, Vietnam	Southeast Asia	Show	Edit
Antarctica CSCE Fall Semester	Antarctica	South Pole	Show	Edit

The picture above portrays the addition of the ‘Show’ option to the left of ‘Edit’ in the ‘Study Abroad Programs’ page. The purpose of ‘Show’ is to allow easier access for the admin to see participants and information about the program.

## Approach To Legacy Project:

Our first priority when we started this project was to get the legacy version of the website deployed again, since it had been over 2 years since anyone else had worked on it. We ran into some roadblocks at first, with their lack of documentation and the use of packages that have been updated since.

One thing we changed about the prior code was the use of environment variables. Environment variables are set outside of the program and can be accessed by the code. This is beneficial because they aren’t accessible by the public and they can be easily changed by the developer. Environment variables become useful when it comes to API keys and other secrets. When trying to get the code deployed, we noticed that API keys were blatantly listed in the code and were referenced in numerous files. Environment variables would provide a way to hide these values and also allow current and future developers to edit those values when API keys expire.

To understand the code, our team tried to correlate each block of code with a functionality. We originally intended to write characteristic tests for these functionalities, but we found it hard to do so with such little Ruby experience at the start of the project. Instead, we conducted these tests manually and figured out how the codebase operates.

## Team Roles Throughout the Project:

Every Iteration (besides iteration 1), there was a change in Scrum Master and Product Owner. If a teammate was not scrum master or product owner, all other teammate roles were a developer.

### Iteration 0:

*Arka Bose* - Scrum Master

*Molly Frost* - Product Owner

### Iteration 1:

*Arka Bose* - Scrum Master

*Molly Frost* - Product Owner

### Iteration 2:

*Uma Sarkar* - Scrum Master

*Anuj Ketkar* - Product Owner

### **Iteration 3:**

*Abhishek More* - Product Owner  
*Sarah Sotelo* - Scrum Master

### **Iteration 4:**

*Anuj Ketkar* - Scrum Master  
*Arka Bose* - Product Owner

### **Iteration 5:**

*Uma Sarkar* - Product Owner  
*Abhishek More* - Scrum Master

## **Scrum Iteration Summaries:**

### **Iteration 0:**

Iteration 0 we focused on planning out the trajectory for the entire project. We did this by planning out features that we wanted to implement, drafting up our first round of user stories, and planning out our strategy for how to go about understanding the legacy code. We also had to all learn how to execute test driven development and behavior driven development since none of us had worked with this style of software development before. We also had to figure out how to get the website deployed on heroku so that we could see what we were working with and begin to draft alternative design choices. We wanted to make sure our documentation was as up to date as possible, so we added to the readme in order to best express how we went about starting off our work with the project and getting it deployed in order to get all members of the current team set up on the right foot, but also future teams working on this project as well.

### **Iteration 1:**

During this iteration we set up the database after planning it out in an ERD diagram so as to best accommodate our workflow. This iteration, we made some slight UI changes, but primarily focused on testing the legacy code and increasing our test coverage before we added any substantial features of our own. This testing process took a bit longer than expected, due to our unfamiliarity with test driven development, behavior driven development, and the legacy code. We did have to create environment variables so as to make the testing environment for each of us more manageable. We were able to increase our test coverage up to 51%. We also spent some time breaking down and reworking our user stories for the next couple of iterations to make them more specific and more manageable, since up until this point they were written at the feature level.

### **Iteration 2:**

During this iteration we designed our admin console and planned out all of the associated functionalities, from adding users to programs to banning users from the site entirely. Since this iteration was during our transition period of returning back to the United States, we had a slight adjustment period

of figuring out our new schedules and how to work together despite not all having the same schedule anymore. This caused us to fall slightly behind schedule, although we were able to make up for this in the following iterations. We did successfully refactor some of the controllers in order to make it so that only users assigned to a program can post and interact with its contents. We also increased our code coverage from 51% to 65%.

#### Iteration 3:

This iteration we actually worked on implementing the admin portal features planned and designed in the last iteration. We began work implementing a flagging feature for tips and experiences. We implemented RSpec tests for as much code as we could after receiving feedback about this from the last iteration, but our code coverage dropped back down to 52% after creating some new features and shifting our focus to TDD instead of BDD. We used rubycritic for the first time this iteration, and got up to a 92.41/100 for our syntax.

#### Iteration 4:

This iteration we continued work with the flag feature functionality, and also sped up work with implementing new features, such as the ability to upload images for experiences, and also for the background of each program page. This is the first iteration we were able to accomplish most of what we had planned, aside from the flag functionality. We added features such as the ability to add multiple students at once (as a comma separated list), and implemented a profile page for each user where you could view all posts made by each user. We drastically improved our test coverage, bringing it up to 81%, made sure that we had RSpec tests for every new code file we created, and maintained our high rubycritic score.

#### Iteration 5:

This iteration we finished the flag feature and test cases for the flag feature for both tips and experiences. We also met with the client to ask if they had any finishing touches to request, and implemented those as well. Our test coverage remained consistent at around 80%, as we continued to test our new features as we wrote them, and our rubycritic stayed consistent at around 90% once again.

### Customer Meetings:

#### December 31st 2022, 1:00 pm SGT :

The entire team met with the client for our very first meeting. We discussed who the target audience was and expectations for future iterations of the project. From this, we derived key features that the client would like to see and in what order they would prefer to see them in. Some features that were brought up include: configurability throughout the website, redefining the faculty/admin console, adding a map, allowing a user to bookmark an experience as an interest, and adding photos to your experiences.

#### January 5, 2023 at 12:00 pm SGT :

Meeting with our product owner (Molly Frost). Presented updates, asked for feedback about the UI mockups

January 9, 2023 between 8:00am-12 pm SGT :

Meeting during class time with our product owner again (Molly Frost).

Jan 10, 2023, 10:21 am SGT:

We presented the work completed with iteration 1 including changes to the UI, test case coverage, and adding a title button to experiences. The client then explained new changes that they would prefer such as being able to delete an experience, and creating a separation between different categories of users. From this, we derived three different tiers of users (User, Faculty, and Admin) that will be implemented with portals for Faculty, and Admin.

Jan 23, 2023, 3:10 pm CST:

Our product owner this iteration (Anuj) met with the client and presented the work completed with iteration 2 including changes to the UI, and admin console. The client then explained new changes that they would prefer such as adding a user profile page that includes all the posts. The client also had issues with the current design of things such as how it's possible to review things multiple times and how the admin console is difficult to navigate. Finally, the client wanted some bugs fixed and some UI tweaks to make the product more accessible.

Feb 17, 2023, 1:45 pm CST:

Arka, our product owner at the time, met with our client this time and presented the work completed with iteration 4 including more changes to the UI, and admin console. The client then explained new changes that they would prefer including some bug fixes and styling decisions.

March 6, 2023, 3:15 pm CST

With our client, Uma, our current product owner, presented the work completed for iteration 5 so far including changes to the UI, the creation of a functioning flag feature for tips and experiences, and a new profile page for each user. The client then requested some slight UI changes and additional functionality in order to improve ease of use.

## Behavior and Test Driven Development:

Behavior Driven Development:

During the course of this project, we learned about what is referred to as behavior driven development. We learned that this entails testing the buttons and overall UI functionality of our application rather than just the logic. In order to correctly implement behavior driven development, we used Cucumber and Capybara. Both of these softwares are made for the intention of being able to easily conduct BDD. Some of the BDD related challenges that we encountered during the course of this project were that since this was a legacy project, we had expected the code coverage to be near an acceptable level when we began the project. When we were able to finally clone the project correctly and ran our test cases we started off at 44% code coverage. This made it a struggle since we found ourselves having to write BDD test cases for code that we didn't write. The previous team did not do a good job setting up the structure of their test

cases and made it really difficult for us to improve the overall coverage as well as writing our own test cases.

#### Test Driven Development:

During this project, along with behavior driven development, we also worked on implementing test driven development. This is different from behavior driven development because test driven development focuses on testing the logic of the code rather than the front end. This means that we would be testing the individual functions that we wrote with expected outputs based on certain inputs. The tool we used to do test driven development was RSpec. Some of the issues we encountered when working with test driven development and RSpec was initially understanding how the files interacted with each other. We struggled to find the connection between the RSpec files and the controllers, as well as the connection between the RSpec files and the RSpec helper files. Another issue that we ran into while working on this was that it was difficult to test some of the logic without a user being logged in and that caused us issues for a while, but we were eventually able to overcome it by setting a default user when the environment was the test environment.

#### Configuration Management Approach:

We have 33 active branches, most of which are merged into main. These graphs represent our github commit history to the master branch.



Jan 5, 2020 – Mar 21, 2023

Contributions: Commits ▾

Contributions to master, excluding merge commits and bot accounts



Since we only picked up this project in late 2022, we are only responsible for the latter half of the graph. As you can see here, there are some spikes, but more noticeably there is a dip. This dip occurs during our transition from Singapore back to Texas, as we all took some time to readjust and figure out how we were going handle our new schedules and meet up and work consistently towards finishing. The spikes occurred as usually we were successful in figuring out a problem that we had been working towards for quite a bit, and were thus able to push a lot of related features at once. We had 5 releases, one for each interaction, each accommodated with its own iteration report and .tar file.

## Tool Issues:

### Heroku:

When deploying to Heroku initially, we ran into a few issues. In Singapore, it took us 3-5 days just to get the project deployed because it was our first time working with Heroku. For some reason, Heroku wouldn't download the gems automatically or run DB functions, so we had to use the Heroku CLI to install the gems and run `rails db:migrate`. Auth also wasn't working on Heroku. This was because we had to set up a Google Cloud project, which wasn't mentioned in the previous documentation. Once we did that, our project was up and running smoothly.

### AWS and LightSail:

We originally tried to get the project working in Lightsail, as that was what we had used during lecture. However, we ran into some issues since Lightsail did not have some dependencies preinstalled that were essential for the project. We took the prior team's advice and switched to Cloud9, which magically started working since it had the dependencies already.

### Cloud9:

We had no issues with Cloud9, though the development process was unappealing to the eye. The search functionality was unintuitive.

## GitHub:

The only issue we really had with GitHub is merge conflicts. This is a common issue in GitHub, so it was not a stressful issue. In order to fix this issue, we edited the merge conflicts in our own IDE and then pushed once the conflicts were resolved.

## Tools and Gems:

All of the gems that are in use came with the legacy project, we did not have to add more gems than what already existed. Still, I will explain some of the main gems that we used below:

### SimpleCov:

This is a code coverage analysis tool that helps you identify which parts of the code are being tested and which are not. SimpleCov is integrated into our test suite to automatically generate coverage reports which highlight areas that require more testing.

### Omniauth:

This gem allows us to use Google Oauth to handle user authentication on the website.

### Shoulda Matchers:

A gem that provides a set of matchers for use with RSpec and Test::Unit that make it easier to write tests for Rails applications. It simplifies the process of writing tests by providing pre-built matchers for common Rails features such as validations, associations, and scopes.

### Cucumber-Rails:

A gem that provides integration between Cucumber and Rails, allowing you to write acceptance tests for your Rails application using Cucumber.

### capybara, selenium-webdriver, and webdrivers:

Gems used for browser-based testing in the production environment.

### pg and rails\_12factor:

Gems used for deploying a Rails application to a production environment on Heroku.

### SQLite3:

A lightweight database interface for the SQLite engine

### Factory Bot:

A gem that provides a way to define and create test data objects in a flexible and maintainable way.

### RSpec-Rails:

A gem that provides integration between RSpec and Rails, allowing you to write tests for your Rails application using RSpec.

### AWS-SDK-S3:

A gem that allows active storage attachments in Rails to synchronize with an Amazon S3 Bucket on production to allow us to use image attachments in the application

## Repo Description:

The repository contains a standard Ruby on Rails project, which includes all the files and directories typically found in a Rails project. This includes files for the models, views, and controllers, as well as configuration files and various other resources.

In addition, a shell script setup.sh, was written to automate the setup of a Ruby on Rails development environment by installing dependencies such as Yarn, Ruby 3.1.3, Rails, and Bundler. It also installs required gems and JavaScript packages, runs database migrations, and starts the Rails server. This script can save time and reduce errors when setting up a new development environment, and includes helpful messages and error checking to ensure successful completion of each task.

The repository also includes a README file, which provides instructions for setting up external services such as the Google Authentication, the .ENV variables, and the S3 Bucket.

## Links:

- Pivotal Tracker: <https://www.pivotaltracker.com/n/projects/2427229>
- Github: <https://github.com/tamu-edu-students/UMAaaS>
- Slack: [umaaas.slack.com](https://umaaas.slack.com)
- Heroku: <https://tripagvisor.herokuapp.com/>
- Presentation Video: <https://youtu.be/LIjLGA6p0X8>
- Demo Video: <https://youtu.be/C35w-FmvsCo>