Team git push --force
UMOC Website
Spring 2018

Overview:

We built a website for the UMass Outing Club (UMOC), which has a large member body and organizes trips for students to go on. Our website provides a platform for club leaders to schedule trips and advertise them to members, who can browse upcoming trips and sign up online. Our website provides an easy interface for both leaders and members; leaders can see who signed up and can generate trip reports, which list the emergency contact information for signed-up users. Meanwhile, users can easily read about upcoming trips and use the built-in comment sections to ask questions.

Our website also implements a system of digital waivers, requiring users to sign an online waiver before being able to sign up for trips. This will replace UMOC's old paper system, which had been very difficult to maintain.

Overall, our website is innovative because it provides a full-featured digital platform for the club. It makes trip creation and advertisement intuitive, greatly simplifies club administration tasks, and implements digital waivers.

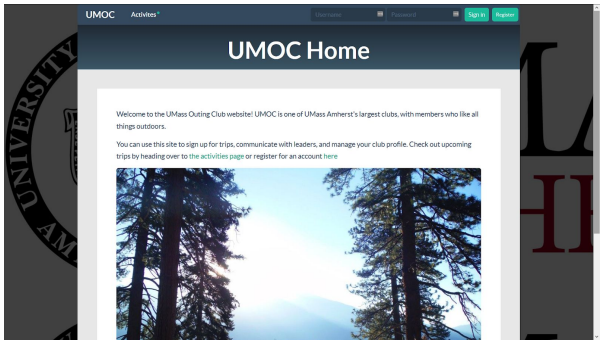
Team Members:

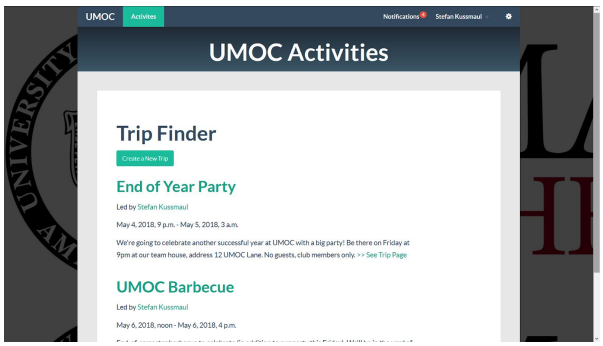
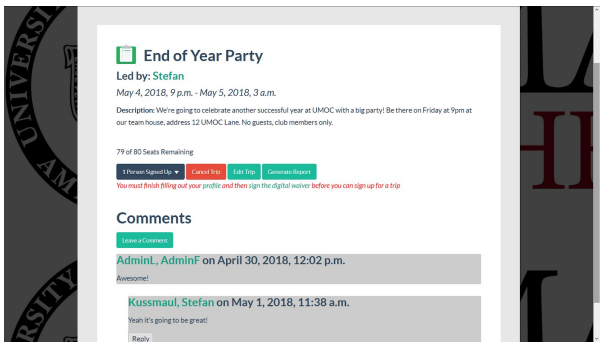
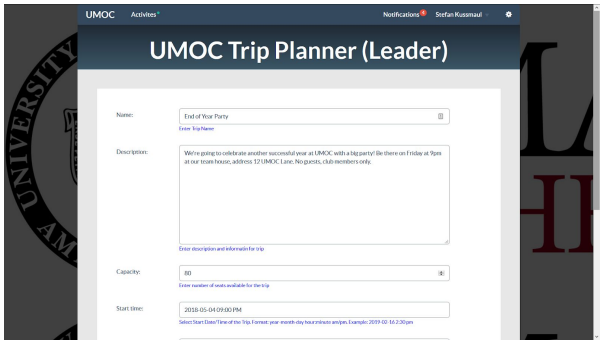
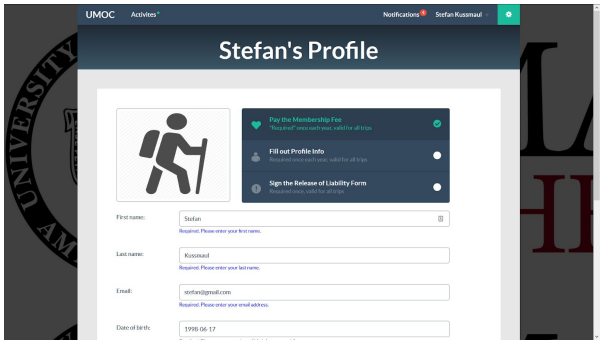
- Susan Pan
- Caroline Yu
- Kyle Vedder
- Jeffrey Shao
- Nonu Bajaj
- Stefan Kussmaul

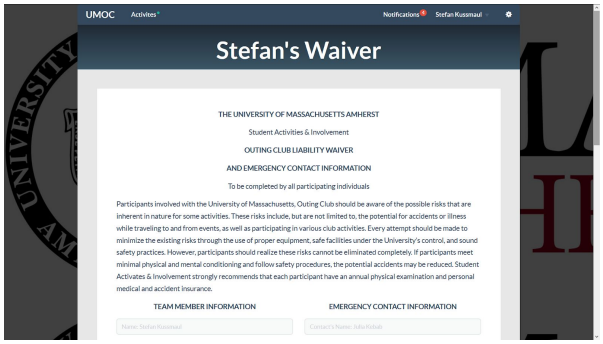
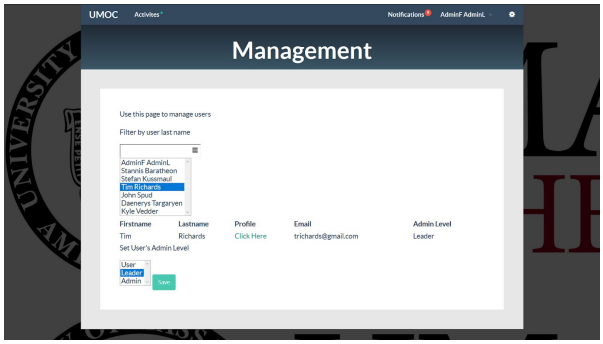
Github Repository:

https://github.com/desporous/CS326Project

User Interface:

Homepage *index* The site's welcome page gives an intro to UMOC and lists some of the site's features. We also include a slideshow of images taken by UMOC users.	

Dashboard *dashboard* The dashboard lists all upcoming trips, ordered chronologically by start date. Each listed trip has a link to its information page, which provides sign-up links and a comment section. To see trips that have already happened, we provide the "Click to See Past Trips" button with the "/trips" url. A User logged in with a Leader/Admin role can see the "Create New Trip" button, which is hidden from regular users.	
Trip Details *trip_info* Information page for a given trip. Provides all information, as well as a comment section. Signing up requires a user to be authenticated, and to have their profile filled out and their waiver signed. The "Cancel" and "Generate Report" functionalities are for Leaders/Admins only, and are hidden from regular users.	
Trip Planner *TripCreate, TripUpdate* Provides functionality to create or edit a trip that will be displayed on the Dashboard page. Restricted to Leaders and Admins only.	
Private Profile *profile* Allows a User to view and edit their information. There is also a to-do list of tasks that must be completed before they can join trips. First, they must pay (defaulted to true). Second, they must fill out their profile/emergency contact information. Finally, they must sign their waiver. This page can be accessed and modified at any time by clicking the gear icon in the navbar. It is only accessible to the currently authenticated user.	

Waiver *waiver* A digital waiver that users must complete before being able to sign up for trips. Users must be authenticated and have completed filling out their private profile to see it.	
Administration *admin_management* Allows a site administrator to view basic information on any signed up user, and set user permission levels. Only accessible to site administrators.	

Data Model:

UserProfile	Represents a club member. Stores all profile information, and linked one-to-one with a User for authentication purposes.
Trip	Represents a trip hosted by club leadership. Includes all relevant trip data, with a ForeignKey to the UserProfile of the trip's assigned leader, and a ManyToMany field for all the UserProfiles that have signed up.
Comment	A comment made on a particular Trip. Stores ForeignKey to the UserProfile of the comment's author, as well as a ForeignKey to the comment's "parent" (the comment it is replying to), which may be null. Stores a ForeignKey to the trip the comment is on.
Notification	A notification for a particular user, reminding them of something. Includes a ForeignKey UserProfile for the recipient of the notification.

URL Routes/Mapping:

* = requires user authentication

‘’	Home page
notifications*	Renders notifications for currently-authenticated user

dashboard	Shows list of upcoming trips
register	New-user registration page
profile*	Authenticated user's profile, which allows editing
profile/<int: pk>	Public profile of user with the given id. Shows very limited information, unless the authenticated user is an administrator
trip/<int: id>	Shows information about the given trip, and comment section
trip/<int: id>/comments	Renders comment section for the given trip. Accepts POST to create a new comment if the user is authenticated. Used with AJAX calls
waiver*	Digital waiver for the user
administration*	Allows viewing all users and setting permission levels. Admin-only
administration_edit*	Handles AJAX GET and POST requests created by the administration path. Admin-only
trip/create*	Allows creation of a new trip. Leaders and admins only
trip/<int: id>/edit*	Allows editing the given trip. Leaders and admins only
trip/<int: id>/cancel*	Cancels the given trip. Leaders and admins only
trip/<int: id>/join*	Adds the user to the given trip
trip/<int: id>/leave*	Removes the user from the given trip
trip/<int: id>/report*	Generates a report on the users signed up for the given trip. Leaders and admins only
trips	Shows all trips that were ever created

Authentication/Authorization:
Users are authenticated using Django's default authentication system. We decided to implement our own permission system, because it was better suited to our site's architecture. Each UserProfile has an admin_level field, which can be "User", "Leader", or "Admin". All users also have a "can_join_trip" permission, which determines whether they are allowed to sign up for trips. This defaults to false, but becomes true once the user finishes their profile and signs the digital waiver.
Leaders have the ability to create new trips and generate trip reports, and Admins have the added ability to set other user's access levels.

Team Choice:
The team choice component involved building working comment sections, as well as a user notification system. This involved writing/adding *comments.js* and *notifications.js* as well as the *notifications* and trip/<int: id>/*comments* URL routes, which provide GET and POST functionality used by the Javascript. Comments and notifications are now rendered separately from the main templates, and are requested through Javascript. The server then renders and returns the html, which is inserted in the website.

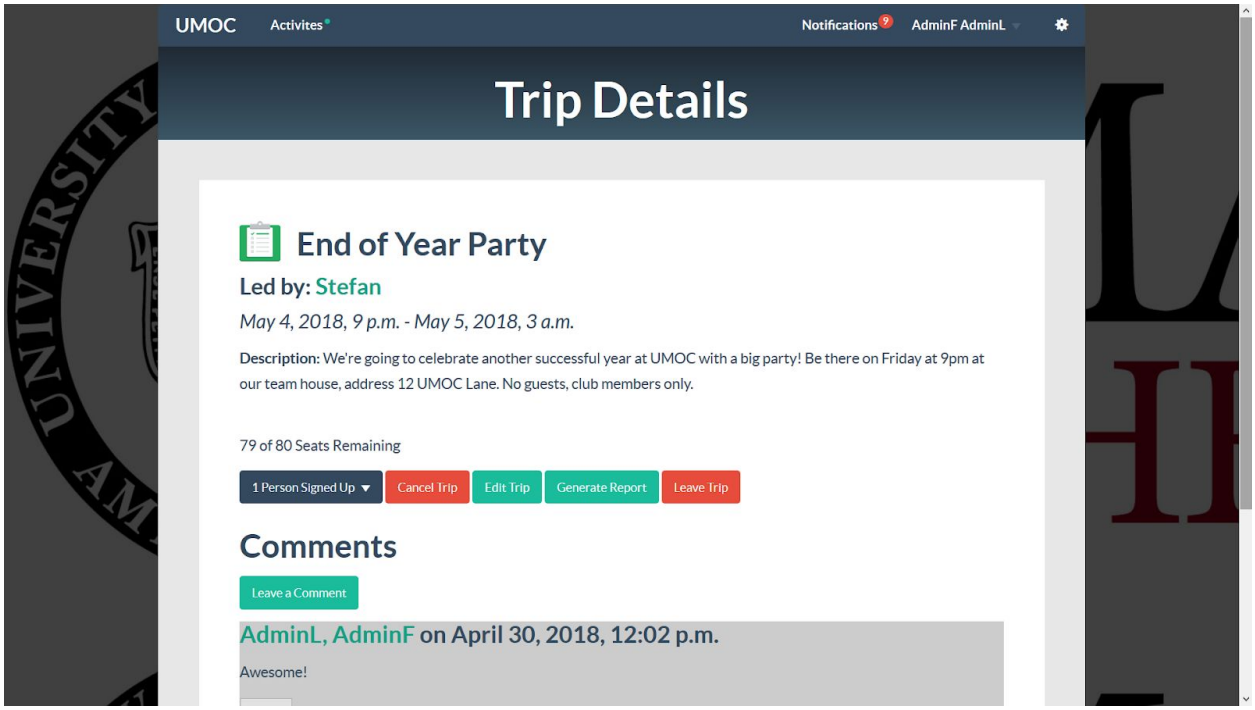
Conclusion:

- **Conclusion:** A conclusion describing your team's experience in working on this project. This should include what you learned through the design and implementation process, the difficulties you encountered, what your team would have liked to know before starting the project that would have helped you later, and any other technical hurdles that your team encountered.

All the functionality we wanted for the website has been done, even the stretch goals. The backend is very clean. The UX is a bit non-friendly only for the start date/time and end date/time for a trip because there was no calendar WITH clock widget for Django forms. However, all the other UX forms/navigating through pages/viewing information are well-implemented. We followed the Django tutorials and completed the steps from the different iterations in timely manner. The UI is clean and easy to follow. There were some difficulties understanding the differences between the templates and UI mockup HTML pages at first, but after working through some more Django library book tutorials, our whole team understood it. We helped each other clear up parts of confusion by communicating frequently and drawing things out during in-class meetings. We also had trouble understanding the hierarchy and naming of the folders because a lot had the same name, but were consistent with the Django tutorials. Creating forms to POST was a bit difficult because we did not know that in order to make a POST function, we **had** to overwrite the default GET function except for registering users. That piece of info was pretty intuitive, but in the future, it would helpful if it were said during class prior to project 3 to avoid confusion for the students. Adding comments and notifications was a bit tricky since a lot of it was self-learning since they were part of the team choice component. We were confused whether to use AJAX, JavaScript, or Django for the comments. After working through all the options, JavaScript seemed to work and was the best choice. Overall all of our problems came to a solution.

Link to Presentation
https://docs.google.com/presentation/d/1BlbDkt1h0sT4Q9EaMs_-aDnzheAEwmb3vBBFGx3oTVg/edit?usp=sharing

Our Single Slide (last slide in the above presentation)


UMOC
Activites
Notifications
AdminF AdminL
Trip Details
End of Year Party
Led by: Stefan
May 4, 2018, 9 p.m. - May 5, 2018, 3 a.m.
Description: We're going to celebrate another successful year at UMOC with a big party! Be there on Friday at 9pm at our team house, address 12 UMOC Lane. No guests, club members only.
79 of 80 Seats Remaining
1 Person Signed Up
Cancel Trip
Edit Trip
Generate Report
Leave Trip
Comments
Leave a Comment
AdminL, AdminF on April 30, 2018, 12:02 p.m.
Awesome!