The Loop: Design Documentation

ver. 2.0

Calvin Li

Richard Lu

Christian Mata

Mikael Mengistu

Motivation

The Loop is a web application that allows students to view and publicize MIT events in a centralized location. With *The Loop*, users will be able to filter and find events of interest, RSVP to events, and follow groups. Group admins will be able to create events, edit event information, add and remove group admins and count RSVPs on behalf of their group.

This app provides a more focused medium to share events and aims to supplement existing methods such as bulletin boards in the Infinite Corridor, spammy email blasts, pesky Facebook event invites, and the unfriendly and incomprehensive events.mit.edu. This allows students to effectively seek out events rather than having events presented to them when not sought after.

Purposes:

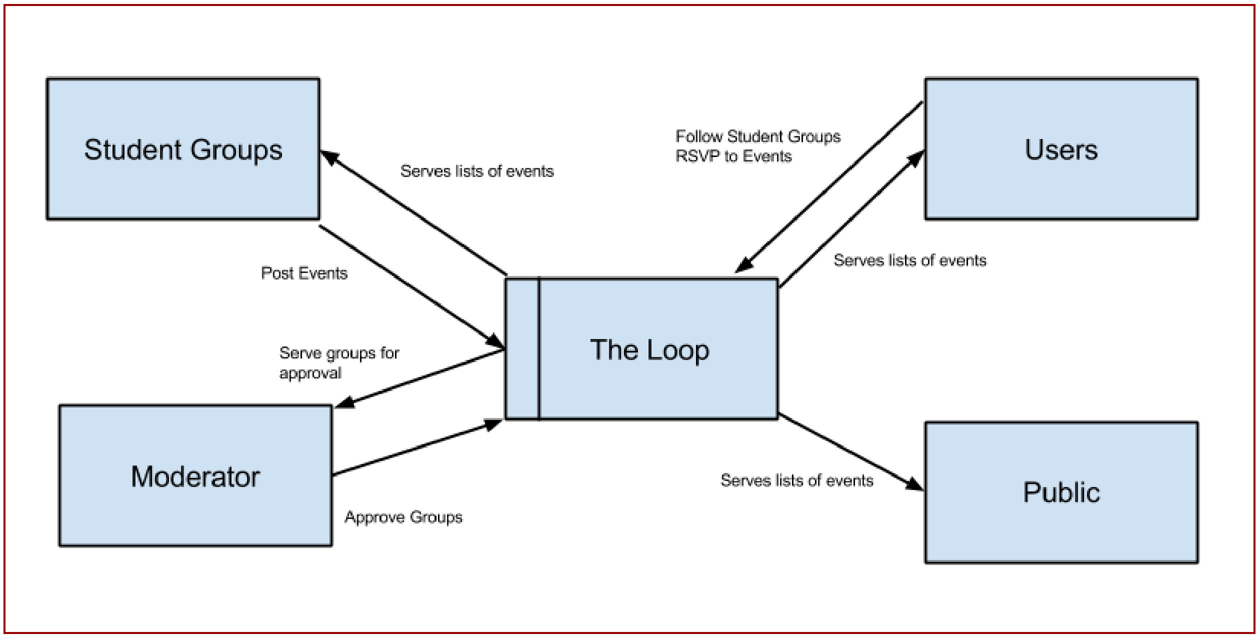
- **Provide a centralized portal for student run events happening at MIT**
- **Give the MIT community a focused medium for advertising events**
- **Allow students to express/gauge interest in the activities**

On “Provide centralized portal for student run events happening at MIT”: *The Loop* is the first MIT-oriented web app to post and view events in general. Currently if a student wants to find events that are occurring on campus, he/she is left looking through emails or staring at bulletin boards and *The Loop* aims to improve the effectiveness of this process.

On “Give the MIT community a focused medium for advertising events”: as it is now, student groups rely on various ineffective mediums for publicizing their events through emails, event invites and flyers. Using *The Loop* allows these groups to share their events through a medium that is completely dedicated to their cause and makes it easier for those who may interested in their events to find them.

On “Allow students to express/gauge interest in the activities”: *The Loop* includes a feature that allows students to RSVP to events and express their interest in attending. This allows students to keep track of events they want to attend and allows groups to gauge interest in the events they’re hosting and plan accordingly.

Context



Concepts

Feed - A list of upcoming events, filterable by group, location, tag and other categories. Allows any visitor (registered and non-registered) to see student-run events that are happening on MIT's campus. Users can filter the Feed by predetermined categories to find events that are of more interest, and can sort the Feed by either posted-time or event-time to find new or upcoming events, respectively.

Events - A group's entry for a specific activity, hosted by the publishing group, that will occur at some point in time. Events are submitted by group admins on behalf of the group, and may be modified or canceled after posting by group or site admins. Events are the primary gateway for groups to publicize their activities to a wider audience.

Tags (of events) - Predefined identifiers that can be used to help distinguish events and make events filterable. Tags help facilitate searching of events at a deeper level than just by group. Each event creator can apply tags to their events at their discretion. Tags can added to and deleted from events during or after posting.

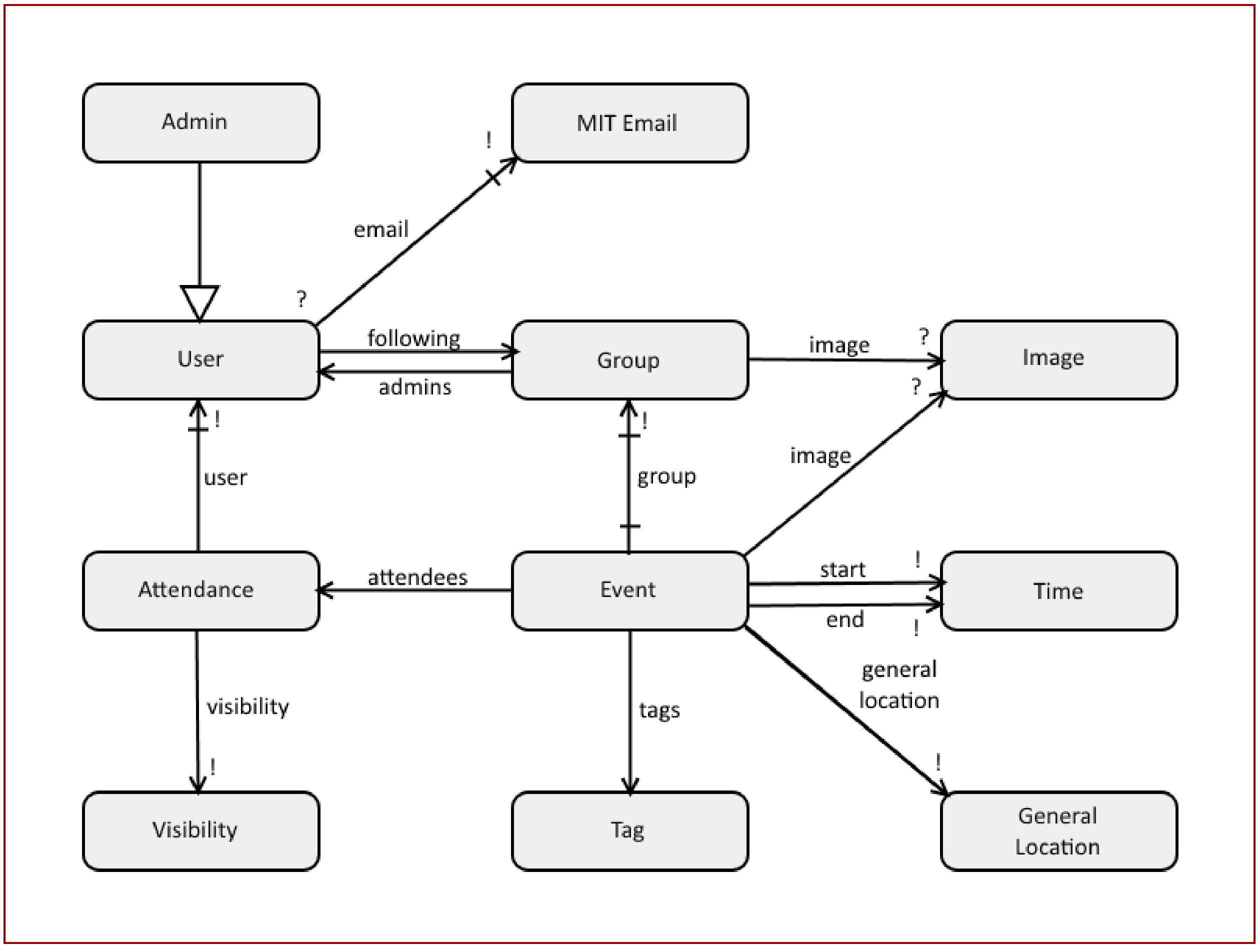
Groups - A campus student organization (e.g. club or student group) with the ability to schedule and host events. Users can request the creation of a group, subject to site administrator's approval. A group should correspond to an existing organization on campus and is managed by group administrators. Each group has its own Events Feed which advertises solely its own events.

Group administrators - Users appointed by current group admins, can administer group's events. Each group can have multiple users appointed as administrators and each user can administer multiple groups.

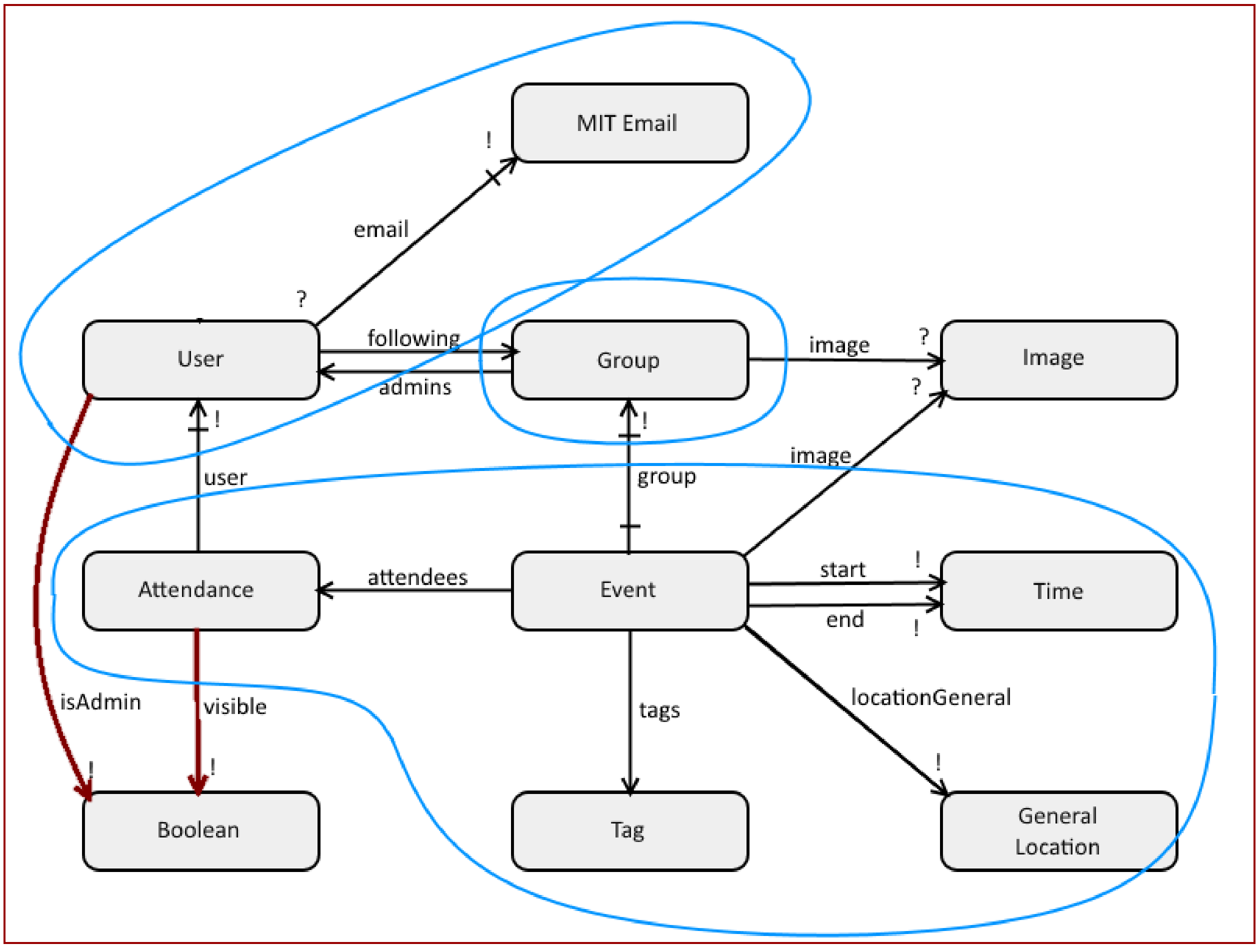
RSVP - A response that indicates a user's interest in attending a specific event. The user can choose whether to make the the RSVP public or private. This allows groups to get a head count of people planning on attending while also allowing users to be discreet about their attendance. Users are able to change their RSVP response.

Following groups - Users can choose which groups they would be interested in when returning to the site. The subset of groups is saved as a Main Feed filter. The user can add and remove groups from this subset.

Data model



Data Design



Security Concerns

Threat Model

- Users may be able to hack into the database and steal information, but cannot hack the host server nor source code
- Hackers will not want to steal account information (email, password, etc.) since we store minimal user information
- Site admin will not have its login information compromised
    - No hacker will have super-access to web application
- Assume group admins add only people they personally know as admins, not random users
    - Current group webmaster/social chair gives admin permission to new officers of their groups
- Hackers may attempt to impersonate an MIT member

Possible Risks

1. **Account Hacking**

   Some of the users are admins for the groups, so anything the user posts will be representative of the group. As a result, if a user has his/her account compromised, it will provide a negative image on the group if any malicious information is added onto the website, where the public can see it. In addition, admins of the entire site have all access to creating, editing, and deleting anything, so having their login information compromised is a security breach for our entire application.

   Mitigation: We will use a strong password encryption such as bcrypt so that no one can access the information if they somehow have access to our database. In addition, we will also require a strong password (greater than 8 chars, with letter and number) for all users so hackers cannot easily decrypt them.

2. **Injection**

   Because our website primarily displays user-created information, we have to worry about malicious users injecting HTML or javascript into our database that may be displayed on our website.

   Mitigation: We will sanitize our text inputs (names, descriptions, username, etc) by escaping dangerous special characters like ampersand, less than, greater than, etc into &amp;, &lt;, &gt;, etc. before inserting it into the database. The only exception would be password, which will never be displayed on the website.

3. **User Authorization**

   We do not want people outside of MIT using the application as they may promote activities or events that are irrelevant to MIT students. We will need a method of confirming if they are a member of the MIT community.

   Mitigation: We will use only @mit.edu emails for users. We will use email confirmation to make sure that the email exists and is active.

4. **Malicious Groups**

   Users may make groups that are not approved by the MIT community, so we need a way to filter groups from being created. In addition, groups that are already created may maliciously post content onto the website, so we need to regulate those groups from doing so again.

Mitigation: Site admins must approve of a group in order for it to be created. In addition, site admins have full control of the application, so if they choose so, they can ban a group by deleting it.

5. **Malicious Posts**

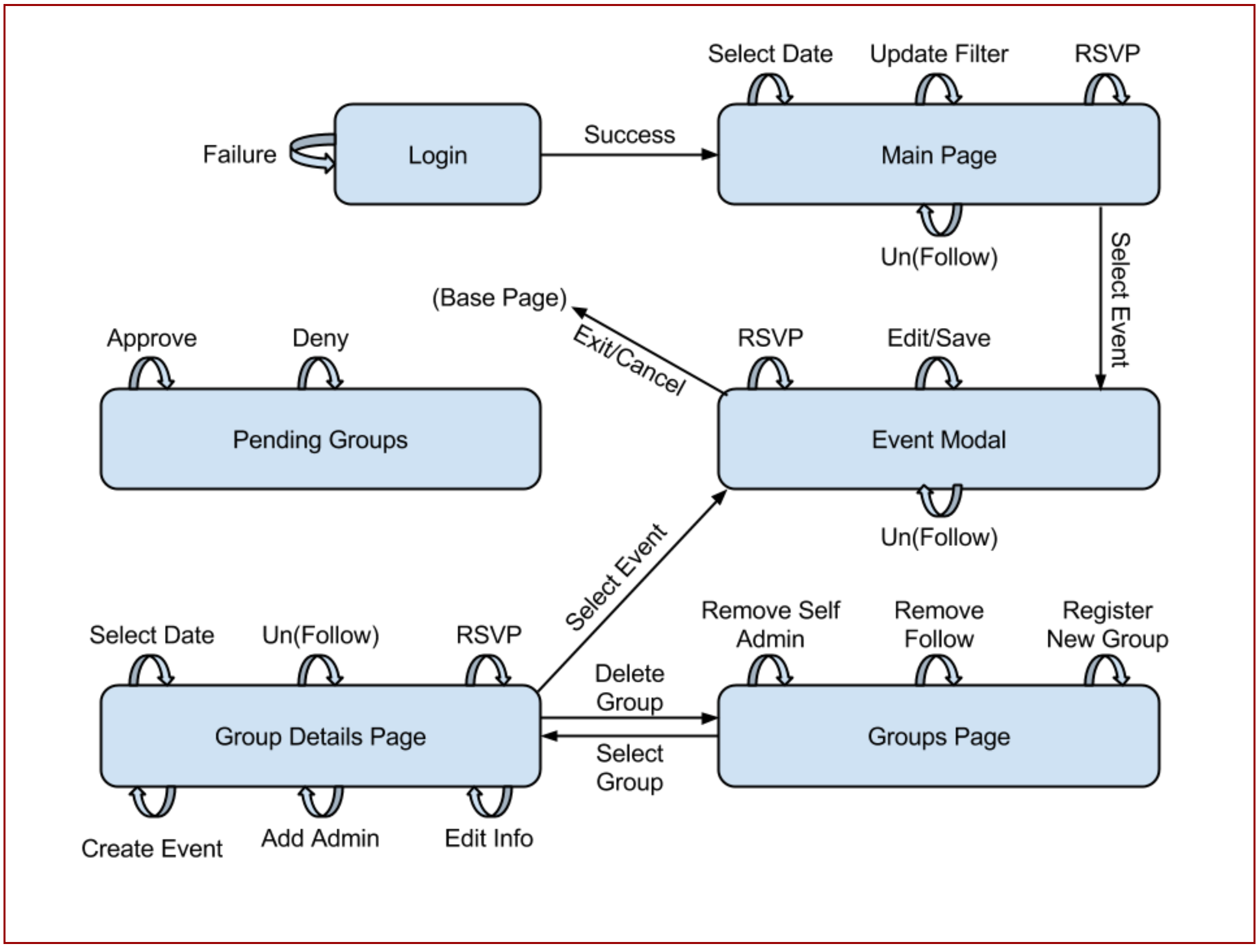
Similarly, groups may post malicious posts or display information in a negative manner, so these posts must be moderated.

Mitigation: Site admins have full control, so they can edit the events. In addition, as stated before, if the situation gets out of hand, they have the right to ban and delete the group.

User interface

Wireframes - see UI_Wireframes.pdf

Interactions Diagram



Design challenges

At what stage should we perform post moderation?

Pre-publication

- + Clamps down on spam
- - More urgent work for moderators
- - Higher submission-to-publication latency

Post-publication

- + Events are published more quickly
- + Less work for moderators
- - Problematic posts may persist for some time before action is taken

Resolution: Post-publication. To encourage groups to submit events to our site and students to visit the site more often, we must prioritize shortening the submission-to-publication latency. We also want the site to be relatively low-maintenance, so less work for moderators is more desirable. As far as problematic posts not being filtered, we believe the group approval process plus warning and/or banning groups after the fact will be more than sufficient.

Do we want to implement a separate my-followed-groups feed page in addition to the main feed page, or should "My Followed Groups" just be a filter option in the main feed?

Personalized Feed

- + Good visual separation between two concepts
- - Building two pages with exact same layout, but with reduced content

Filtering Option in Main Feed

- + Minimal UI change, since two feeds utilize same page structure
- + Easy to switch between "Personal" and "Main Feed" with no page reload
- - Less learnability since feeds are meshed together into one

Resolution: Filtering Option. As long as we implement the filtering feature for followed groups so that users can easily learn how to use it, they will eventually be able to conceptualize the two "separate" feeds as one. Then, we would not have to implement another feed that filters based off of different parameters and just combine them into one. Less work for us as coders, and once users learn the feature, less work for them too.

Should RSVP identities be anonymous or public?

Public

- + MIT students will be able to see who is going and can identify friends
- - Users may not RSVP, making our attendance count less accurate

Anonymous

- + Users may be more willing to RSVP
- + Users can still have their identities public if they choose so
- - User's friends may be less encouraged to participate

Resolution: Anonymous. Having the choice to RSVP publicly or anonymously encourages more users to want to RSVP, which will optimize the accurateness of attendance number. Users can easily figure out if a specific person is going by asking them in real life.

How can users find a group which interests them, without having to find an event hosted by that group?

Alphabetized groups listing page

- + Displays all groups, even if they have not posted events
- - List could potentially grow very long, hard to parse visually

Groups search bar

- + Lean interface, one less page to implement
- - Cannot find interesting groups that one does not already know about
- - Introduces free-form text input which we must sanitize

Resolution: Alphabetized listing page. There are many ways to solve the long-list problem (columns, collapsible section headers, etc.), and the benefit of providing visibility to all groups outweighs the implementation cost.

Should our data model allow for events hosted by multiple groups?

Yes

- + Properly advertises multi-group events such as Nightmarket

No

- + Keeps data model and UI simple

Resolution: No. We feel that multi-group events occur with low frequency and does not sufficiently justify the added complexity, especially to the UI. It is not unreasonable to ask multiple hosting groups of an event to designate a representative host group, who can then credit the other hosting groups in the event title or description.

What unique identifier should users authenticate with?

Self-selected username

- - Introduces another frustrating name selection process to users' lives
- - Not strongly tied to MIT identity

Kerberos ID

- + Guaranteed to be unique for different legitimate users
- - Typing this without the '-@mit.edu' suffix tends to feel unnatural, especially if not accompanied by a text hint

MIT email address

- + Guaranteed to be unique for different legitimate users
- - Adds unnecessary bloat to database

Resolution: MIT email address. Prompting users for their MIT email address is more foolproof than asking for their Kerberos ID. We can simply remove '@mit.edu' when saving to the database.

Do we want to limit the length of events?

Yes, to 24 hours

- + Simplifies date selection UI
- + Forces groups to consider splitting long-running events (CPW) into smaller sub-events (Athletic Gateway, Nitrogen Ice Cream)
- - Excludes the possibility of legitimate, atomic, multi-day events

No

- + Allows multi-day events such as hackathons
- - Adds clutter to event time display (must display both start and end date)

- Must implement own duration selection UI module (currently using a modified timepicker, which only goes up to 24 hours)

Resolution: Yes, to 24 hours. While there could be legitimate events which last for longer than 24 hours, it is likely rare enough that excluding them will not be too detrimental to the value of our site. The benefits of allowing them does not outweigh the savings of excluding them.

How should we differentiate between view and data routes?

<u>Categorize view and/or data routes - under /view or /api</u>

+ Clean and semantically intuitive
- Requires restructuring all existing route definitions and calls

<u>Add extension to view (.html) and/or data (.json) routes</u>

+ Negligible refactoring required
- Looks clunky

Resolution: Put all data routes under /api. We want browser-displayed URLs for our site to look as clean as possible, so extensions are no-go; and since view routes are what most users will regularly deal with, the leanness of view URLs takes precedence over that of data URLs.

How should we handle errors in API routes?

<u>Use customized error object for all errors</u>

+ Consistency across all error types
+ Can create human readable messages to display/interpret client-side
- Requires more server-side code to interpret errors into a given format

<u>Use Mongo error object for DB errors and customized error object for others</u>

+ Easy to see DB errors client-side for debugging
- Mongo errors less readable for users

Resolution: Use customized errors for all. Error responses should be tailored towards the users, so they should be human readable. Consistency is more important than having users who can’t interpret the errors.