**Notes**

Green - Added for FP

~~Red~~ - Deleted from FP

# Overview (Tess Gustafson)

**e-Template Full Design Overview**

1. **Brief description of system to be built**

   When users first navigate to e-Template, they will immediately see published email templates with titles and descriptions about the intended uses. Without being logged in, any user can filter these templates on this page by topic, author, and location, helping users find templates that really matter to them. Clicking on one of these templates will lead to an expanded view of it, showing the email recipients, subject, and body. Users can then customize the parts that authors have included to make it more personal before sending this email template. Users can easily log in or create an account via a navbar. Signing in allows users to create, edit, and bookmark templates. They can also save drafted templates. Authentication also allows users to make template copy requests for templates in which they want to modify, say for instance, to make it applicable to a different location. Authors can thus see private statistics as to how many users have shared this template or have copied the template, helping activists see how helpful their templates have been.

2. **Key purposes**

- *Lower activation energy: e-Template will make it easy for activists to find email templates applicable to the causes they want to support.*

  e-Template will be a helpful resource to any activist for any cause. Users will be able to search for templates based on topic, author, and location. This lowers the activation for activists to find resources because anyone can simply search e-Template to find mailto links perfectly related to causes.

  e-Template will therefore be a library of email templates that people can easily search for and save for later, being able to easily navigate back to it days, weeks, and months later.

- *Ease of creation: e-Template will make it easy for activists with or without a tech background to make customizable email templates.*

  e-Template not only makes the generation of email templates uncomplicated, but also adds an aspect of customization. These email templates can be customized via adding fill in the blanks, multiple choice, and dropdowns, often things not present in normal mailto links. Thus e-Template not only makes creating email templates easier, but also adds more functionality for users to customize.

- *Activism insights: e-Template will allow authors to see insights as to how much reach their published templates have had.*
  e-Template hopes to give activists insights as to how much spread their templates have generated by allowing users to see how many times a template has been shared and customized. Users will also be able to see the number of requests to duplicate this template. This should give users more of an idea as to how helpful their templates actually are, allowing users to create better and better templates over time.

3. **Deficiencies of existing solutions**
   Current solutions for generating mailto links are not ideal. Often these email templates are often hardcoded (a tolerable yet unpleasant experience) and most people without a tech background are mystified by how these email templates are made. It also requires a higher activation energy since users must put in a lot of time and effort to hard coding these. Similarly, searching for mailto links for causes is also difficult. Many email templates shared are often spread via social media. This makes it difficult to actually search for applicable email templates. Similarly, being posted on social media platforms often is not permanent, where people post these templates only temporarily, so you cannot go back and find them later. Lastly, users cannot really see from current email templates how much spread their mailto links have generated. Posting on social media is difficult because you might not be able to track everyone who has used it and where they are sharing it. This is why e-Template will be a beneficial resource to any and all activists, as we hope to diminish these deficiencies.

# Design Commentary (Sabina Chen)

**A Note on Jargon:**

Individuals: people who use e-Template
Creators: individuals who create templates
Users: individuals who use templates created by Creators

## Email Recipients

1. **Template creators can group emails together into "Recipient Bundles"**
   Emails can be grouped together as "Recipient Bundles" by Creators so that Users can more easily identify recipient types. For example, grouping 5 emails together and naming the bundle "Deans of MIT". Creators can reuse their own recipient bundles, but recipient bundles will not be shared among Creators in order to limit the possibility of accidentally reusing inaccurate Recipient Bundles without explicit intent. Furthermore, we made a language choice to call this type of email grouping "Recipient Bundles" as opposed to "Mailing List", to prevent implications that e-Template is a complete email

client (which it is not). The number of emails within a Recipient Bundle is also limited (ie. below 100 recipients).

Alternatives Considered

We considered allowing Recipient Bundles to be re-used among Creators across the platform, so that other Creators, who are creating new templates that email to the same group of people ie. "Deans of MIT", do not have to spend time looking for individual emails. However, we decided against this because emails should be sent **with intent** to the individuals involved, so we did not want creators reusing Recipient Bundles without fact checking the emails first, nor do we want to encourage or recommend using Recipient Bundles without context.

2. **Emails can be updated in the Creator's "Contacts" page**

~~When a creator adds new recipients to an email template, the emails will automatically be added to the creator's contact list for future use.~~ Creators can create both Individual Contacts and Recipient bundles in the "Contacts" page. Any emails in this "Contacts" list will be recommended to the user in the future when creating new templates. This decision was made because it is likely that the people to whom the Creators are trying to contact will not change greatly over time, especially if the Creators are active within the same area (aka. contacting the same local officials). Creators will also have the option to explicitly edit and delete these contacts and recipient bundles.

Alternatives Considered

Not providing a Contacts list for Creators. This was decided against because we needed a system to keep track of Recipient Bundles, since Recipient Bundles is one of our core concepts, especially in terms of User accessibility. Also, in most email clients such as gmail or hotmail, when users are completing the "to" field, relevant emails are recommended, so we wanted to have the same functionality for creators when they are making email templates, which meant there needed to be a Contact list to pull data from for individual users. We also decided against automatically adding recipients to a Creator's contact list when a Creator adds new recipients to an email template, since the Creator is not explicitly alerted about this functionality, and may be surprised if they did not intend to add these emails to their Contacts list. There can also be security concerns related to previous email usage (ie. if a Creator did not want to save information regarding their past recipients).

## Creating Templates

3. **Creators can duplicate other Creator's email templates**

If a Creator finds an existing template relevant, the Creator can choose to duplicate the existing template and modify it to their needs. For example, an email about global warming that was originally sent to local officials in Boston, can be duplicated, with the recipient field changed to officials in Los

Angeles instead. Duplicated emails cannot be exactly the same as the original. Duplicated emails will also have a tag in the template notifying Users that the template was based on another template. When a template is duplicated, Creators will be able to view their Insights to see that the number of templates duplicated from their original template has increased by one.

Alternatives Considered

Either not allow for duplication of templates, or require creators to ask for permission before being able to duplicate. We provided the option for duplication through our platform so that templates are not hard-copied by other Creators without notice. Copyright and reuse of content is already a big issue that many social media platforms have not yet completely resolved themselves, so rather than preventing or taking down the reuse of materials outright (since some reuse can be useful, ie the global warming example above), we provided an option for explicit duplication through the e-Template platform that will provide credit to the original creators. Furthermore, forcing Creators to ask for permission before usage is an extra step for Creators, especially if they are under time pressure, and may encourage Creators to just copy and reupload the content themselves, rather than going through our platform, which is not something we want to encourage.

4. **Template Preview Cards for quick overviews**

   Template preview cards include template title, blurbs, creator, and tags. The preview cards are loaded in the home page so that users can more easily search through templates by reading blurbs to understand the relevant context before clicking into each template. Template titles are kept short (within a specified character limit) to be meaningful and succinct to communicate to users the purpose of the template. Template blurbs and email bodies are not character limited to enable Creators to be detailed in providing context to their templates, as well as to be meaningful in their associated emails.

   Alternatives Considered

   Show the templates as-is in the homepage, as opposed to creating separate preview cards. This was decided against because some templates may be too large to preview. Furthermore, users will have a more difficult time searching through templates when there is too much visual/text input. Thus the preview cards will provide a succinct way to summarize templates meaningfully.

## Using Templates

5. **Users can interact with templates without signing in**

   Users can view, search for, and use templates without having to sign into the website.

   Alternatives Considered

Require users to sign-in in order to interact with the templates. This was decided against because one of the primary purposes of e-Template is to lower the barrier of entry in order to participate in activism. Requiring sign-ins may discourage users from participating in email campaigns.

6. **Users must be signed in to create, bookmark, comment, and report templates/comments.**
   For Users looking to become more involved in the activism community, Users can choose to create an account and sign in. Signing in gives Users the option for more actions, such as creating their own templates, saving searches via bookmarks, commenting on templates, as well as reporting malicious behavior on the platform. Having Users linked to personal accounts enables Users to save their data, as well as encourage responsible usage of the platform.

   Alternatives Considered
   Enable anonymous creating, commenting, and reporting. This was quickly decided against because we do not want anonymous individuals to be spamming the platform with unhelpful or malicious content.

7. **Editing options for template users**
   How to communicate to Users regarding what part of the template is optional or required to fill out when editing a template:
   - Blank —> to be filled in
   - Dropdowns —> multiple choice
   - Optional Paragraph —> checkbox
   - Text —> add in preview
   - FilterBuster —> add random characters (optional)

   Alternatives Considered
   How to more explicitly communicate to Users how to use the template editor. We do not have any alternatives, better solutions will most likely be found through actual user testing and implementation.

8. **Enabling FilterBuster**
   Enabling FilterBuster via a checkbox (optional) will add random characters to the body of the email. This is done in order to get past email filters so that the generated emails will not be sent directly to spam. The purpose of most email campaigns and protests is to provide a voice to the people so that their voices are heard by the government. This involves taking up space digitally / physically and making their voices and presence known. Stock email templates without FilterBuster can be easily ignored and filtered away, thereby ignoring the concerns of the people.

   Alternatives Considered

Disable FilterBuster altogether. Although the act of attempting to out-maneuver spam filters may be an ethical dilemma in itself, we have decided that this functionality adds to the one of the overall goals of e-Template, which is to enable activists to have a voice online by providing a resource for sharing email templates. Given this decision, in order to mitigate malicious usage of the platform, we have also included multiple design decisions throughout the e-Template to prevent the platform from becoming an outright spam bot (techniques are described throughout the Design Commentary). Furthermore, FilterBuster is optional, so Users uncomfortable using this functionality can choose not to use it. Ultimately, the option for FilterBuster is a choice among many within the e-Template platform that Users can decide to use.

## Organizing Templates

9. **Templates can be filtered by search location radius**

Users can search for templates relevant to their location of interest. The location filter narrows down the results to only those templates whose locations are within a 50 mile radius to the desired search input.

Alternatives Considered

By enabling a search location radii, as opposed to matching *only* specific locations, users can search for templates relevant to general areas of interest, as opposed to being limited to having to search for exact location names. This feature makes templates more likely to be found by enabling grouping of templates targeted for general communities.

10. **Templates are organized by date of creation.**

Templates are organized by date of creation because most templates are created to be actionable and effective within a set timeframe, thus by organizing the templates by date of creation, most relevant templates will appear at the top of the list. For example, a template created urging officials to pass a specific policy will most likely not be relevant 2 years from now, so it will be pushed to the bottom of the list. When creators edit and create new templates, they can also see their “Last modified” and “Created on” times respectively, so that they are aware when they last interacted with their templates.

Alternatives Considered

Recommend or organize templates by popularity (ie. most used, most likes, most comments, etc) or alphabetically. These options were decided against because we do not want e-Template to become a social media echo chamber where the “most popular” templates get reused the most often. We want to encourage Users to read individual templates and make educated decisions on what templates to use. Thus besides the option for filtering in the home page, templates are organized by date of creation, with most recent on top.

## Usability

11. **User actions show explicit feedback to signify server state**

    Various prompts are shown to the user when waiting for server responses, so that it is explicit what the state of the website is at at any given time. The following are a few examples: When templates are being loaded into the home page, a "Loading Templates" prompt is shown. When no templates are found, a "No templates found" prompt is shown. When users are saving templates, the save button is disabled into a loading button until the server responds successfully. When a template is successfully saved, the "Save" button turns green. If a user starts editing the template again, the "Save" button is re-enabled. If a server error occurs, the user is logged out and taken to the home page with a "Session Timeout" error. When users hover over templates, the templates "pop-out". Selectable items turn the cursor into a pointer. And templates for which the user is the owner are differentiated by different colors. Dropdowns and user options for templates and accounts are also dependent on users who are logged in vs logged out. Successful password changes are also signified with "Success" prompts. Help icons are also interlaced within the app to teach users how to use the website. These are only a few of the many usability examples we have considered and applied when designing the website.

    Alternatives Considered
    Log errors and server states into the console, as opposed to enabling explicit, reactive feedback in the graphical user interface. This was decided against when we started user testing our UI, and found that we had difficulty knowing if an action has completed successfully without looking at the browser inspector. We realized that we could not rely on the general user to know or understand our application innately. Furthermore, it is surprising for users if they expect a certain action to cause state changes to the browser, but nothing changes. Therefore, in order to follow general industry convention and speak the user's language, we decided to make server states and actions explicit in the GUI.

12. **Mobile device and Chrome browser compatibility**

    The website is tested and compatible with Chrome. Additionally, along with regular computer browsers, we have designed the application with mobile phone usage in mind, so that the app also looks good and can be accessed through a smartphone browser. Since many young users now spend the majority of their time accessing apps and websites on their mobile phones, ensuring that e-Template is also viewable and usable on mobile is important to enable greater audience reach.

    Alternatives Considered
    We considered also making the website compatible with Safari and Firefox, however, due to time constraints, we decided to only prioritize the website and mobile versions of Chrome. We also

considered our mentor's feedback to use tooltips, however mobile devices do not allow for tooltips, so we decided against this, and opted to use popup modals instead.

13. **Providing security when storing and accessing user data**
    Passwords are hashed when stored in the database. User accounts are referred to as UUIDs, as opposed to their usernames or emails. Usernames, emails, and passwords are never sent to the client, once they have been initially created and stored in the database. Furthermore, the URL is redirected to HTTPS to ensure secure connections between the client and the server. Cookies are used to provide persistent state on the browser. Additionally, if users who are not signed in attempt to access the profile or admin portal, they are automatically redirected to the login page.

    Alternatives Considered
    Store passwords in plain text, and use usernames or emails as account identifiers as opposed to UUIDs. Enabling any user to access the admin and random user profile pages. These options were decided against for obvious reasons. Identifying user data can be stolen from multiple different avenues. If information from the database or the server is accessed maliciously, personal user data can be stolen and misused. Furthermore without explicit redirection when users attempt to access unauthorized pages, personal user data can be stolen. Therefore adding various sources of security can better help prevent common security hacks.

14. **Error prevention via visual design choices**
    When users decide to report or delete templates and comments, a pop up modal appears asking users to confirm their action. This helps ensure that actions are intentional, and not just accidental misclicks. In particular, these pop-up modals are *especially* important for admin actions within the Admin Portal, since their actions hold much more power than the general user. Admins can add/delete other admins, resolve/delete templates, resolve/delete comments, and ban/unban users. The platform therefore uses colors (on top of pop up modals) to convey whether an action is "good" or "bad". For example, for actions involving deleting or banning items, the button is colored red. For actions involving resolving or unbanning items, these buttons are colored green. Using colors further helps users recognize the impact of their desired actions, thereby adding another layer of error prevention and user confirmation.

# Changes Made to Address Feedback
# (Sabina Chen, Stephanie Yoon)

1. **Enhance discoverability that partial matches to search words**

   Changes Made

   Previously, users can enter any text within the search bar. Since our previous application uses exact matching of words to find appropriate tags, if a search text was misspelled, then no templates were found. Given our mentor's feedback, we have converted the search bar to give recommendations to users of all the current available matching tags. Users can only select tags that are within the recommended dropdown list, so that it is not possible to load a feed with no templates found. This decision was both for error prevention, as well as to increase recognition, as opposed to relying on recall, for the user.

2. **Provide Insights to show efficacy of templates**

   Changes Made

   Insights such as: # viewed, # mailed, # filterbustered, # bookmarked, and # comments,  will be made available to ONLY template creators. Previously, we did not want to provide users with this type of information because we did not want to encourage the popularity-focused social-media type addiction and echo chamber to our application. However, this decision made it difficult to know if a template was effective or if it is being used by others or not. Our mentor suggested that rather than allow all users to see these insights, instead, only allow template creators to see these insights. These insights will not be shared among users, and template statistics can only be viewed by the creator themselves. This will hopefully incentivize template creators to create more effective and meaningful templates, since they can see if their templates are actually being used or not, while preventing these insights from being used like social media popularity contests.

3. **Preventing bad actors from misusing the platform**

   Changes Made

   From our mentor and staff feedback, one concern is the possibility of bad actors misusing the platform (ie. spamming civilians with malicious content). In general, we recognize that we cannot prevent the malicious usage of our platform completely, since this is a known issue among larger social media companies online. Any website created can be misused even if it was not part of the creator's original intent. However, given this, we still recognize the importance of addressing the possibility of bad actors misusing our platform. In that vein, we have implemented various prevention methods to discourage malicious users from using e-Template specifically for spamming purposes. These methods include: not allowing templates to be sent to more than 20 recipients at a time, allowing general users to report inappropriate content, creating an admin portal for template and comment moderation (as well as for user banning), notifying admins of templates that have been filterbustered a standard deviation above that of regular templates, redirecting users back to the home page when “Open in Mail” is clicked so that it is harder for bad actors to repeatedly filter buster the same template to mass spam general users, making filterbuster an optional choice so that users can select this action **with intent**, and using mail:to links that only enable text within emails (no attachments and no links allowed).

# Ethics Protocol Analysis (Tess Gustafson)

1. **How are useful templates found?**

*Choice: Users can search for templates related to tags or location. At least one tag and location is mandatory for a template. Any user (signed in or not) can use a template they find.*

- Process Lens: Privacy is compromised because users can see authors and location of templates, thereby being able to infer a user’s likely location (only likely because you can make templates for locations in which you do not reside, but there is likely to be a correlation with where users create templates and where they reside).
- Outcome Lens: This might increase the number of users who use templates because we lower activation energy by not requiring users to be logged in to use templates.
- Process Lens: Privacy is less compromised than in other solutions where we use the author’s current location at time of creation for the location of the template. This gives authors the option to identify a different location for templates, even though it is still required.
- Structure Lens: This is the most user friendly option to find templates, as it is straightforward in its capability and similar to most search engine functionality. Therefore it should be simple to find templates for anyone of any age, race, and ability.
- Structure Lens: This model may work best for English-speakers. Our website is strictly in English, so it will likely be that tags are in English, thus harming non-English speakers. This also might mean there is a higher percentage of locations residing in English speaking countries, decreasing the usage in non-English countries.

2. **How do we allow users to see how effective their email templates are?**

*Choice: Authors can see insights as to how many users generated mail-to links for their template, how many users made a copy of their template, and how many users have viewed their template.*

- Outcome Lens: Making these insights private decreases the chance of e-Template becoming an echo chamber and mirroring social media sights.
- Process Lens: Users consent that, by using an email template, authors can get insights as to usage but

no personal information will be shared with the authors, thereby maintaining privacy.

- Structure Lens: All users are treated equally since any author can get these insights, and no personal information (such as race, age, gender, ability, etc.) is shared back to authors via insights.
- Legitimacy Lens: Users see this sight as more legitimate since you can gain insights back about your templates, which is not something you could otherwise have.
- Outcome Lens: Authors gaining insights encourage them to write better and better templates to get better insights. This overall will be more beneficial for activists in the long run since templates will be encouraged to be improved.
- Process Lens: Process is easy for all authors since they consent to gain insights when publishing a template and can easily see these insights.

3. **How do we prevent users from writing malicious email templates?**

*Choice: All published templates are public. Users can report harmful email templates and moderators will assess if the author did in fact generate a harmful email. Users also cannot have more than (MAX_RECIPIENT_BUNDLE_SIZE) recipients in an email, to prevent phishing or other harmful emails.*

- Process Lens: Authors consent that any email templates are public and can therefore be reported for being malicious.
- Outcome Lens: Having a cap on the number of recipients minimizes the risk that email templates are generated for mass phishing attacks.
- Legitimacy Lens: Individuals take the app more seriously because they know everything is public and can be reported.
- Process Lens: Authors can immediately post their templates without having to wait for it to be preapproved for malicious intent, making the process easier for stakeholders.
- Structure Lens: All authors are treated equally, with the assumption that email templates are created to promote positive changes until proven (reported and investigated) that they are malicious.
- Outcome Lens: There may be a decrease in the number of templates published if users are worried about getting reported or cannot send to all the recipients they would like to. However, we believe that it will not affect that many authors' choices and overall increase safety.

4. **Should we include FilterBuster, a way for users to bypass email filters?**

*Choice: We decided to allow FilterBuster because it allows users to make their voice heard, such that if multiple users send similar templates, people cannot filter out these emails. FilterBuster adds a random letter in the middle of every word so that it is still human readable but can bypass filters.*

- Outcome Lens: Having filter buster as a option can allow bad actors to spam emails accounts that cannot filter out these spam emails, however we think this will not be that big of a concern because FilterBuster would not help phishing attacks and spamming people is more easily done outside this app so phishers/bad actors are more likely to already use whatever they use for their attacks and they will not be encouraged to use e-template.
- Process Lens: Having a button that allows filter buster allows an easier way for users to generate mail-to links that they know will be read, making this filter buster easier for users since we automatically generate it
- Structure Lens: All users are treated equally and have equal accessibility to use FilterBuster
- Legitimacy Lens: Users could possibly think that allowing emails to bypass filters is risky, making it seem less legitimate and safe

- Legitimacy Lens: Users could also think that allowing emails to bypass filters helps spread their voice, making it seem more legitimate
- Outcome Lens: Having filter buster as an option gives users a way to send emails about things they care about without a fear of immediately getting sent to a trash, spam, or other ignored folder for the person they are trying to contact

# Heuristic Evaluation (Jonathan Wang)

1. Visibility of system status
    - When adding or removing tags, authors, or locations to filter by on the main page, the current set of filters being applied is shown beneath the search bar. Similarly, when creating or editing a template, the lists of tags and recipients that have been recognized are shown as pills, which allows the user to identify what the system is considering a single tag or recipient.
2. Match between system and the real world
    - The template-editing and template-using pages are similar to any email client, with fields for *to*, *cc*, *bcc*, *subject*, and *body*. This consistency with what users are familiar with helps them to better understand components of the template.
3. User control and freedom
    - For actions such as sharing and deleting templates, a modal dialog box appears allowing users to cancel the current interaction or confirm that it is what they want to do. On the admin portal, the admin is prompted to confirm their action before they delete or ban a user, and allows them to cancel the action if it was by mistake.
4. Consistency and standards
    - The home page follows industry convention by placing the login and signup buttons at the right side of the navbar at the top. Users are typically already acquainted with this pattern, so it is easier to figure out where to make an account on our service. We also follow standards by a context menu hidden behind a "…" button for actions users can take on individual template cards. When viewing a template, there is a back arrow icon near the upper-left of the view that takes the user back to the homepage. Following these conventions help users navigate the site more fluently and efficiently. The icons we use for things such as deleting and sharing are commonly used elsewhere and thus easily understood.
5. Error prevention
    - For irreversible actions like deleting a template as its owner or deleting templates or banning users as an admin user, the site prompts the user for confirmation, which helps to prevent errors. The sign-up and login forms cannot be submitted unless all required information has been filled out, which is communicated to the user by the grayed out submit button. This is also true of the template creation and editing page, which grays out the submit button if not all required fields are filled in.
6. Recognition rather than recall
    - When filtering templates on the homepage, an autocomplete feature looks at all of the locations, tags, and authors and makes suggestions on what the user is looking for. This is also true of adding tags to templates when creating them. This turns a recall problem into a recall problem.
7. Flexibility and efficiency of use
    - The bookmarking feature allows speedily finding specific templates, without having to recall all of the necessary filters. In addition to typing filters by hand, users can click on authors, locations, or tags in template cards they see and have them be added to the list of filters.

8. Aesthetic and minimalist design
    - All of our pages attempt to minimize extraneous information, providing only information that will be helpful in the user's task.
9. Help users recognize, diagnose, and recover from errors
    - This point is currently a shortcoming of our design. For instance, if the save and publish buttons are disabled on the template creation page because the form is not considered complete, there is currently no indication of what else needs to be filled in for the form to be complete.
10. Help and documentation
    - For the typical user who enters the site to just search for and use email templates, a help modal outlines the concrete steps required for the most common task of using a template to send an email.

# Conceptual Design (Jonathan Wang)

## Template

### Purpose

The meat of our application. A template is created by Users and can be used to send email by other people.

### Structure

#### Sets

```
Template = Set of templates
Title = Set of titles (string)
Blurb = Set of blurbs (string)
Location = Set of locations (string)
Tag = Set of tags (string)
Recipient = Set of recipients (name/email pair) or RecipientBundles (concept: see below)
Subject = Set of subjects (string)
Content = Set of Contents (string that is interactive on the client side)
User = Set of Users (Concept: see below)
Comment = Set of comments
```

#### Relations

```
creator: Template -> User
title: Template -> Title
blurb: Template -> Blurb
location: Template -> Location
tags: Template -> Tag
to: Template -> Recipient
cc: Template -> Recipient
```

```
bcc: Template -> Recipient
subject: Template -> Subject
content: Template -> Content
comments: Template -> Comment
published: Template -> Template (reflexive relation (t, t) in published iff Template t is
   published)
```

### Additional Constraints

```
all t: Template |
    #(t.title) = 1 and #(t.blurb) = 1 and #(t.location) = 1
    and some t.tags and #(t.creator) = 1 and #(t.published) = 1
```

## Actions

```
create(u: User, s: Title, b: Blurb, l: Location, g: Tag):
  t = new Template
  creator += (t, u)
  title += (t, s)
  // similarly for blurb, location, tag

edit_title(u: User, t: Template, s: Title):
  (u in t.creator)
  title -= (t, t.title)
  title += (t, s)

// edit_blurb, edit_location, edit_subject, edit_content omitted for brevity

add_tag(u: User, t: Template, g: Tag):
  (u in t.creator)
  tags += (t, g)

remove_tag(u: User, t: Template, g: Tag):
  (u in t.creator)
  tags -= (t, g)

add_recipient_to(u: User, t: Template, to: Recipient):
  (u in t.creator)
  to += (t, to)

remove_recipient_to(u: User, t: Template, to: Recipient):
  (u in t.creator)
  to -= (t, to)

// add_recipient_cc, ..., remove_recipient_bcc omitted for brevity

comment(t: Template, c: Comment):
  comments += (t, c)

publish(u: User, t: Template):
  (u in t.creator)
  published += (t, t)

unpublish(u: User, t: Template):
  (u in t.creator)
  published -= (t, t)

copy(u: User, t: Template):
  (u in t.creator or some t.published)
  let r = new Template
  title += r -> t.title
  // similarly for blurb, location, tags, to, cc, bcc, subject, content
  creator += (r, u)
```

## Operational Principle

```
After:
  t := create(u1, “Template1”, “My first template”, “Cambridge MA”, “MIT”)
Observe:
  t in Templates
  and t.creator = u1
  and t.title = “Template1”
  and ...
  and t.tags = { “MIT” }
After:
  add_tag(u1, t, “COVID”)
Observe:
  t.tags = { “MIT”, “COVID” }
After:
  add_recipient_to(u1, t, “reif@mit.edu”)
  edit_subject(u1, t, “Rising COVID cases at MIT; shut it down”)
  publish(u1, t)
Observe:
  t.to = { “reif@mit.edu” }
  and t.subject = “Rising COVID cases at MIT; shut it down”
  and some t.published
After:
  t2 := copy(u2, t)
Observe:
  t.creator = u1 and t2.creator = u2
  and all field: { title, blurb, location, ... } | t.field = t2.field
  no t2.published
```

# RecipientBundle

## Purpose

Allows grouping of a number of recipients (name/email pairs) together under one readable name (i.e. “Deans of Harvard” grouping together all of the deans at Harvard), allowing template users to better understand who they are emailing.

## Structure

### Sets

```
Bundle = Set of RecipientBundles
User = Set of Users (concept: see below)
Label = Set of names/labels (string)
Recipient = Set of name/email pairs
```

### Relations

```
label: Bundle -> Label
recipients: Bundle -> Recipient
bundles: User -> Bundle
```

### Additional Constraints

```
all b: Bundle | #(b.label) = 1
```

## Actions

```
create(u: User, l: Label):
  let b = new RecipientBundle
  bundles += (u, b)
  label += (b, l)

add_recipient(u: User, b: Bundle, r: Recipient):
  (b in u.bundles)
  recipients += (b, r)

remove_recipient(u: User, b: Bundle, r: Recipient):
  (b in u.bundles)
  recipients -= (b, r)
```

## Operational Principle

```
(recipients emails omitted for brevity)
After:
  b := create(u1, “Cool Course 6 People”)
Observe:
  b in u1.bundles
  b.label = “Cool Course 6 People”
  no b.recipients
After:
  add_recipient(u1, b, “Daniel Jackson”)
  add_recipient(u1, b, “Arvind Satyanarayan”)
  add_recipient(u1, b, “EJ Sefah”)
Observe:
  b.label = “Cool Course 6 People”
  b.recipients = { “Daniel Jackson”, “Arvind Satyanarayan”, “EJ Sefah” }
After:
  remove_recipient(u1, b, “Arvind Satyanarayan”)
  add_recipient(u1, b, “Emily Hu”)
Observe:
  b.recipients = { “Daniel Jackson”, “EJ Sefah”, “Emily Hu” }
```

# Moderation

## Purpose

Allow malicious users and templates to be reported and for Admins to delete offending templates or ban offending users.

## Structure

### Sets

```
Template = Set of Templates (concept: see above)
User = Set of Users (concept: see below)
Admin = Set of Admin users (Admin in User)
```

### Relations

```
reportedUsers: User -> User
reportedTemplates: Template -> Template
```

### Additional Notes

```
Reflexive relation (u, u) in reportedUsers iff User u is reported
```

## Actions

```
report(t: Template):
  (no reportedTemplates.t)
  reportedTemplates += (t, t)

// similarly for report(u: User)
```

```
resolveReport(a: Admin, t: Template):
  (some reportedTemplates.t)
  reportedTemplates -= (t, t)

// similarly for resolveReport(a: Admin, u: User)

delete(a: Admin, t: Template):
  (some reportedTemplates.t)
  resolveReport(a, t)
  Template -= t

ban(a: Admin, u: User):
  (some reportedUsers.u)
  resolveReport(a, u)
  User -= u
  // also delete all templates where template.creator = u
```

### Operational Principle

```
After:
  report(t1)
  report(u3)
Observe:
  some reportedTemplates.t1 and some reportedUsers.u3
After:
  resolveReport(a5, t1)
  ban(a5, u3)
Observe:
  no reportedTemplates.t1 and no reportedUsers.u3
  u3 not in User
  (all of u3’s templates are deleted)
```

## User

### Purpose

Allows users of the platform to have an associated identity and thus privilege them to things such as create and edit their own templates.

### Structure

#### Sets

```
User = Set of Users
String = Set of one-word strings
```

```
Email = Set of emails
Template = Set of Templates
RecipientBundle = Set of RecipientBundles (Concept: see above)
```

### Relations

```
username: User -> String
password: User -> String
email: User -> Email
templates: User -> Template
bundles: User -> RecipientBundle
bookmarked: User -> Template
```

### Additional Constraints

```
all u: User |
  #(u.username) = 1 and #(u.password) = 1 and #(u.email) = 1
```

## Actions

```
create(un: String, m: Email, pw: String):
  let u = new User
  // omitted for brevity

change_password(u: User, pw: String):
  // omitted for brevity

// change_username and change_email omitted for brevity

create_template(u: User, (template parameters)):
  let t = create(u, ...) // see Template create above
  templates += (u, t)

create_bundle(u: User, (bundle parameters)):
  let b = create(u, ...) // see Bundle create above
  bundles += (u, b)

add_bookmark(u: User, t: Template):
  bookmarked += (u, t)

remove_bookmark(u: User, t: Template):
  bookmarked -= (u, t)
```

## Operational Principle

```
After:
  u := create(“jw”, “jonathan.wang@mit.edu”, “******”)
  create_template(u, ...)
  create_bundle(u, ...)
  add_bookmark(u, t1)
  add_bookmark(u, t2)
  add_bookmark(u, t3)
  remove_bookmark(u, t2)
Observe:
  u in User
  some u.templates
  some u.bundles
  u.bookmarked = { t1, t3 }
```

## Wireframes (Stephanie Yoon)

https://www.figma.com/file/a0qOtYL1myWN9yyHuP8Pap/e-Template?node-id=0%3A1