Clippy Team Write Up: Gather

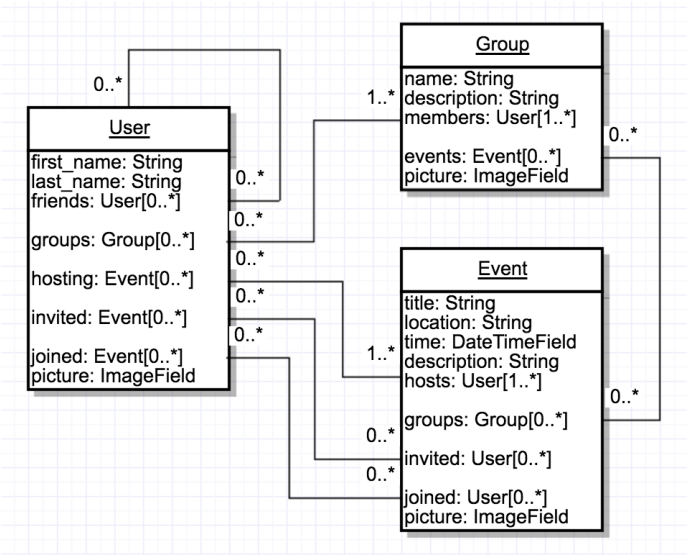
Overview:

The purpose of Gather is to allow users and groups to quickly organize and post location-based events. As laid out in the Project 1 submission, users can view upcoming events in various home, user, and group feeds and then either join or hide events. This submission implements many of these goals. There are further goals such as commenting and Google maps integration that we will continue to work on in future project submissions.

Team Members: Luis Quiterio, Cyrus Freshman, Connor Ingram, Michael Maio, Ryan Clayton

Github Repository: https://github.com/cfreshman/clippy

Design Overview:



Our data model contains 3 objects: events, users, and groups. Events are the core of the model because every event needs members, users can be placed into groups to easily be added to events together. The event model has a time, location, title, description, an optional picture and users who are attending the event. Users have a first name, last name, a list of friends and a picture. Users can be put into groups which have names, descriptions, an optional picture and a list of members.

For the UI, we implemented templates for each of our pages included in the mock UI, all based off of a generic template. This includes templates for the home feed, user feed, group feed, group manager page, create event page, and account settings page. The implemented URLs were able to make navigation possible on the website and the views ensure that dynamic content can be displayed from the database. Important URL routes include: /gather for home feed, /gather/u/<id> for user feeds, /gather/g/<id> for group feeds, /gather/manager for group manager, /gather/event for event creation, and /gather/settings for account settings. All of these except user feeds are accessible off the home feed, and user feeds can be reached through groups.

Problems/Successes:

The biggest problem our team encountered has been using git for version control, as all of us were unfamiliar with it and many had never used it before. Besides that we have generally avoided any serious roadblocks. Our greatest success was finally figuring out git and completing the project.

Figuring out how to integrate images was a minor roadblock, which required configuring media paths for the project. A success was using a base generic template for the navbar and another base feed template (extending the generic one as well) for the sidebar/event feeds, which removed a lot of duplicate code. For the next project, we can improve collaboration by meeting more regularly to discuss requirements and features for the application.

Individual Write Ups - percentages approximately equal

Luis Quiterio: So far, Luis' contributions to the project have consisted of

1. Creating the mock UI for the application's "Create Event" page.
2. Defining the model for the application's "Event" data model.

In creating the mock UI for the Create Event page, I first had to determine what type of data the page required and also come up with a design for the page. Once that was determined, I went on to develop the mock page by implementing the necessary HTML, CSS, and JS files using the Bootstrap framework.

In terms of defining the model for the Event data model, the team first decided on data models and attributes that made sense to our application. Team member Cyrus then created a diagram that shows each model's attributes and the relationships between different models if any. After the creation of the data model diagram, I went on to implement the Event model in our application's code. This primarily consisted of translating the attributes of the Event data model into Python code.

Cyrus Freshman: In project 1, I created the mock UI for the home feed, including the navbar and sidebar which are common elements across the site. For project 2, I drew up the data model diagram and populated the database with mock data. I implemented the URL mappings, templates, and view functions for the home feed, group feed, and user feed. I configured the project's media settings in order to display images from the database. I also redesigned some aspects of the user and group feeds.

Connor Ingram: In project 1, I created the mock UI page for our account settings html. Project 2 was largely a team effort. Once we solidified our model on paper (mostly Cyrus who also created the diagram above), we split our roles. I was responsible for implementing the 'User' model, and assisting in making sure our database migrations were successful, specifically fixing an issue with reverse relation in 'models.py' and for the ImageFields.

Michael Maio: For project 2, I implemented the templates for the group manager page, account settings, and the create event page. This included the URL mappings, to ensure that navigation worked properly, as well as creating views for each of these pages to ensure the content displayed correctly.

Ryan Clayton: This project our group completed part 0 mostly together during class (I just did the startproject and start app command), then split parts 1 and 2 amongst us. 3 of us made the data models (part 1) and 2 made the url mappings and templates (part 2). For our data model we had 3 models so it was easy to split the models between us. I was in charge of writing the event class. After the 3 of us had written our classes we had to ensure all the models worked together and made migrations. There were some problems with the compilation of the event class because there were multiple fields for users they needed to have different related names, Connor fixed this.

After that part 1 was finished, part 2 was completed by team members, part 3 I completed the design overview for the data models (not the picture though) and the general problems and successes, there weren't any real problems besides understanding git, and no real notable successes. Id say I contributed approximately an even amount, I think we split the work up fairly evenly, the url mappings and templates might have taken a bit longer though honestly but I'm not sure.