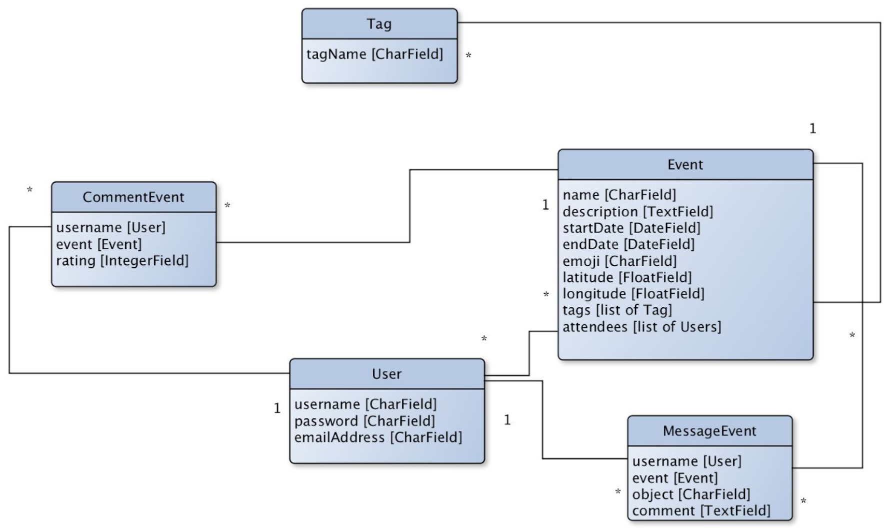
COMPSCI 326 Project 2: Data Model and Views

Team Write Up

- **Overview:**
  Our web app is a tool to discover events going on in your area. Users can drop pins on a map and attach information about an event so other users can find it.
- **Team Members:** Miles Black, Jordan Chen, James Curry, Mégane Michaud
- **Github Repository:** https://github.com/m3gan3/ZooMaps
- **Design Overview:**
  Our data models included tags, users, events, ratings and messages. Events will be rendered on a map in the final version and have a list of applicable tags for the event, as well as a list of users attending the event. The main URL routes are using views to display the index page, the list of events, which appear on the map page, the detailed pages of each event, and the user's account. We also have two static pages for now.
- **Problems/Successes:**
- We spent a long time making sure our models would be able to hold all of the information we wanted. We wanted to make sure we were all on the same page and knew exactly what functionality we would be implementing. When we attempted to add static pages to our site, we ended up breaking the admin functionality of Django. This went unnoticed for several days which meant it was difficult to track down exactly what was causing the error. It appeared after all that it was due to some django flatpage package that we imported. Since we ended up not using it, we got rid of the import and this fixed the admin page.
- A lot of time has been spent on the views and the templates: two of our members spent time trying to create the correct views, which would for instance display events that are in the future only. They also had to find a way to create a view that takes data from two different models: they used chain from the intertools package.
- Using the base_generic template for all pages added in a few issues in which page designs became inconsistent and some features such as page tracking was lost. We added the use of some page data such as current url, etc. to organize what css attributes to apply to things such as the nav bar. As well, adding model data made a few of the pages look slightly unaligned due to the length of certain data.
- So far, the search is done on the whole database: it either retrieves all the events to come, or, if a search term has been entered, it returns the events having the term in their name. Ideally, we would like to implement different filter, not limited to the event name, but also to the description, the location, the date, etc. For now, this was too demanding, but our team will ideally fix this later.
- We had an issue setting up the Github repository so that the database and pycache files would initially be empty. This was because development started without a proper gitignore. The issue was resolved later by creating a branch, deleting the unwanted files on that branch, setting up files to programmatically populate the database, adding a gitignore, and then merging it back into the master branch.
  - After this step, we needed to re-add a sample database file because some of our templates weren't working correctly without previous data that was added manually. Thus another branch was used to merge a sample db.sqlite3 file into master.

Data Model



Individual Write Up

- Miles Black: I worked on cleaning up the front-end design in order to properly view the dynamic models we added to this project. I worked with the HTML/CSS in order to add dynamic links and page tracking through the navbar based on the Django {url} data. Adding in new data seemed to make our footers and side navigation slightly messy, so most adjustments were just taking data and adjusting the site layout so everything fits nicely. I also helped with git tasks such as handling some of the pull requests and merges.

- Jordan Chen: I was in charge of cleaning up the repository (pycache files) and handling the gitignore. I added ways to populate the database programmatically (models, dummy data, superuser). I handled administrative stuff like creating this document, organizing the files into a Google Drive folder, and creating and filling in a Trello board. I also facilitated the creation of a project spec so that the team was on the same page regarding the specifics of the project.

- James Curry: I added the ability for users to search for upcoming events which involved learning a lot more about how Django views work. I also solved a bug that broke our admin site, which went unnoticed for several days of development. I helped finalize the models so that we knew the scope of our project. I helped Miles with some small CSS issues and Mégane with a templating and views issue.

- Mégane Michaud: I worked on the data model and implemented the diagram, and after brainstorming with my team and taking their view into perspective, I implemented models.py. I also coded the different templates as well as their urls, and worked on different views, so that users could see their own upcoming events, ordered, that a search would return different ordered events and with a good pagination, and so that each detailed event views would return the messages and ratings associated. I also populated the database with some mock data.