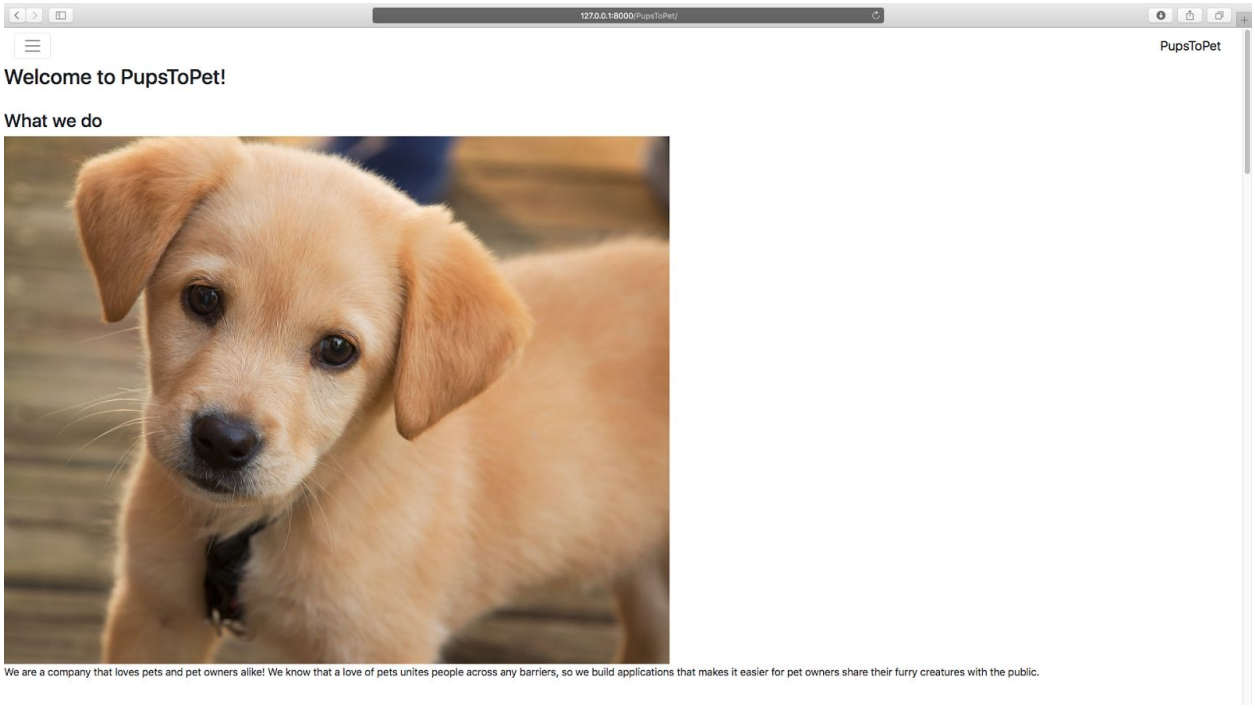
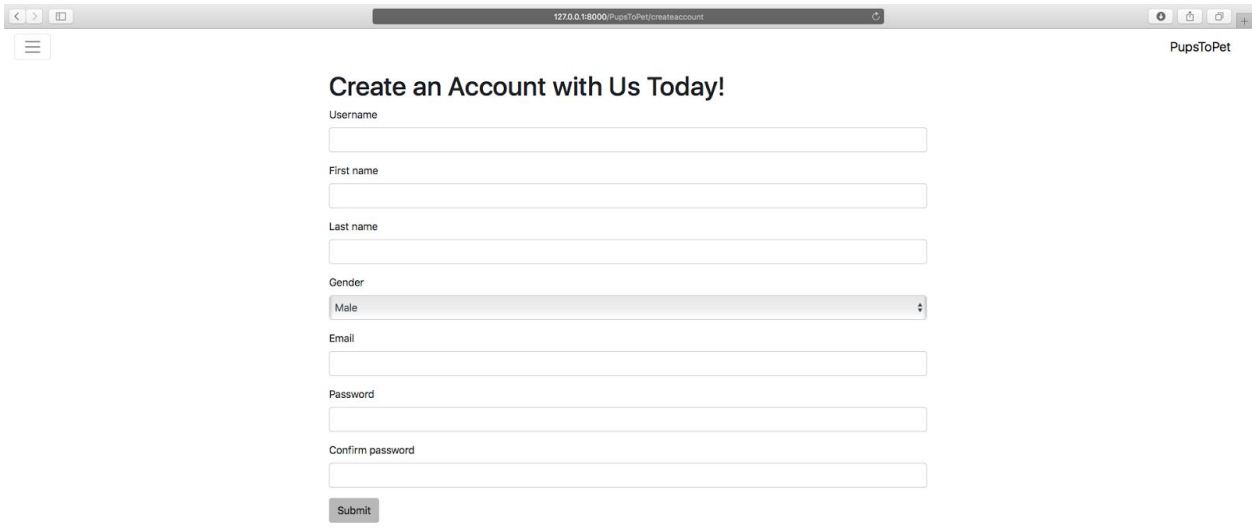
Title: PuppyCorps (though we may be written down as Deadmaus)
Subtitle: PupsToPet
Semester: Fall 2017
Overview: PupsToPet is a web based application that allows users to organise and find events wherein a pet is brought for others to pet and/or play with. It is the focus on pets that makes this application unique.
Team Members: Colin Lee, Arthur Normand, Thao Phan, Jared Yeager
Github Repository: https://github.com/normana10/CS326TermProject/

User Interface:
About: our homepage, a simple about page.

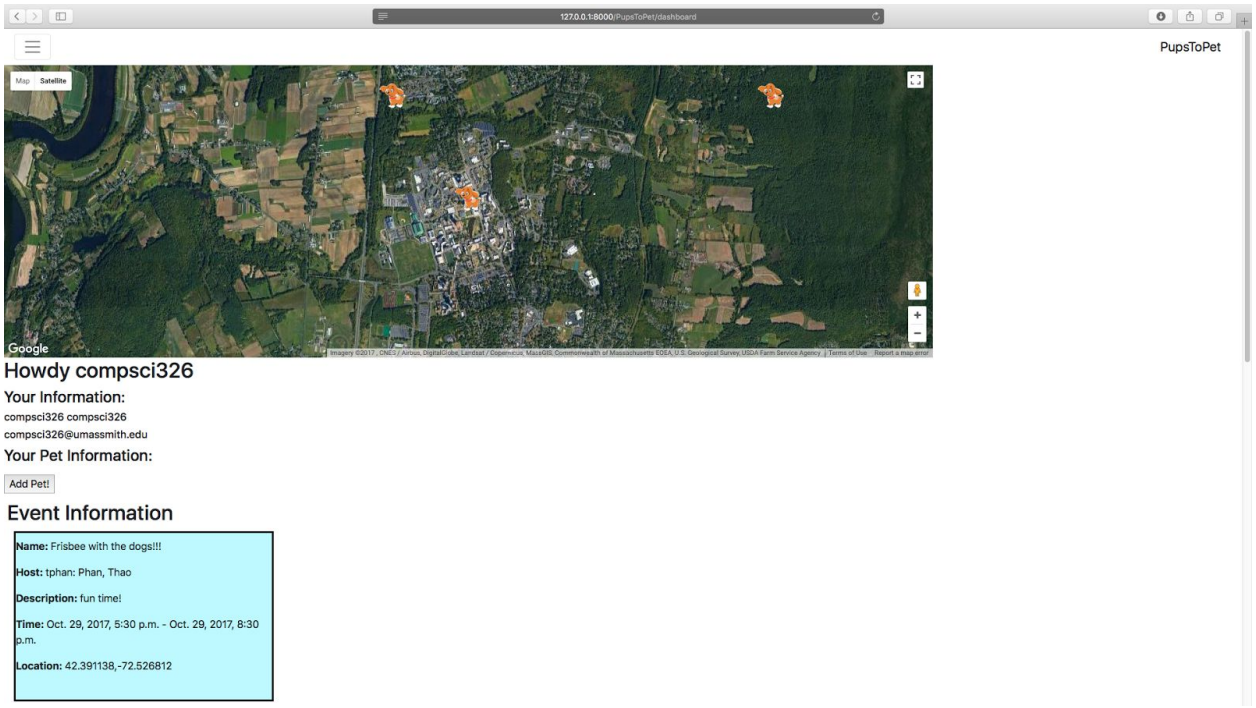


Create Account: a page from which a user can create a new account

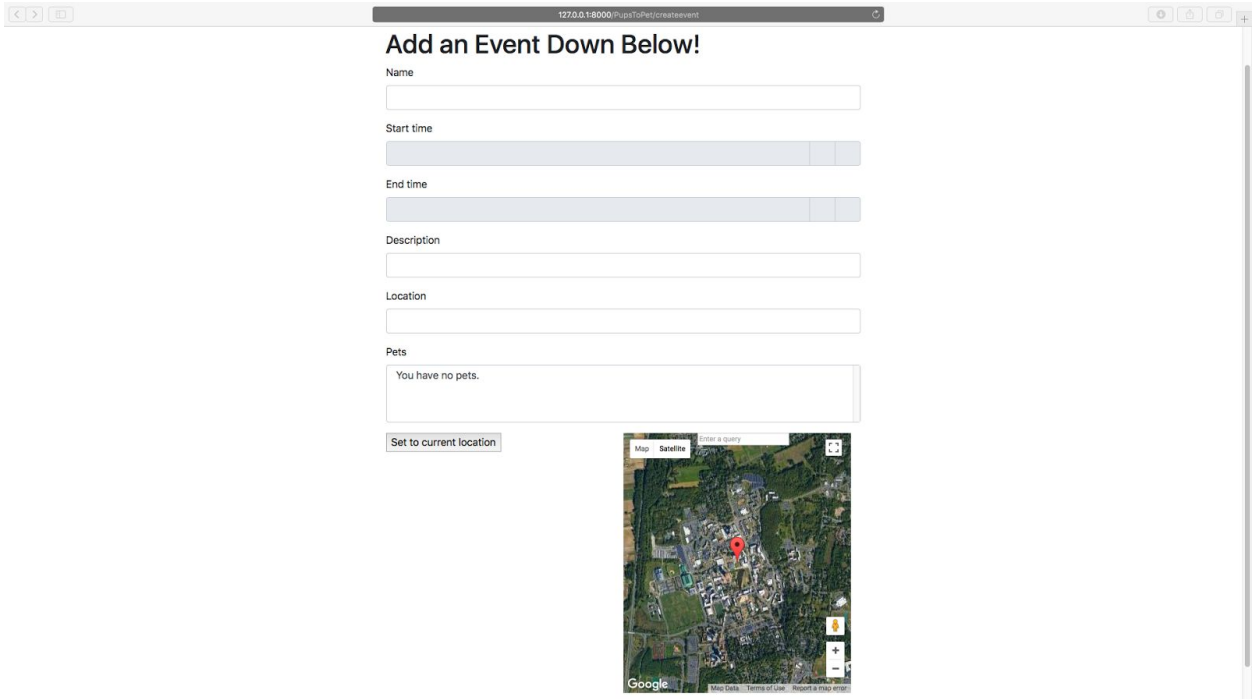


Login: a page which the user user to log in

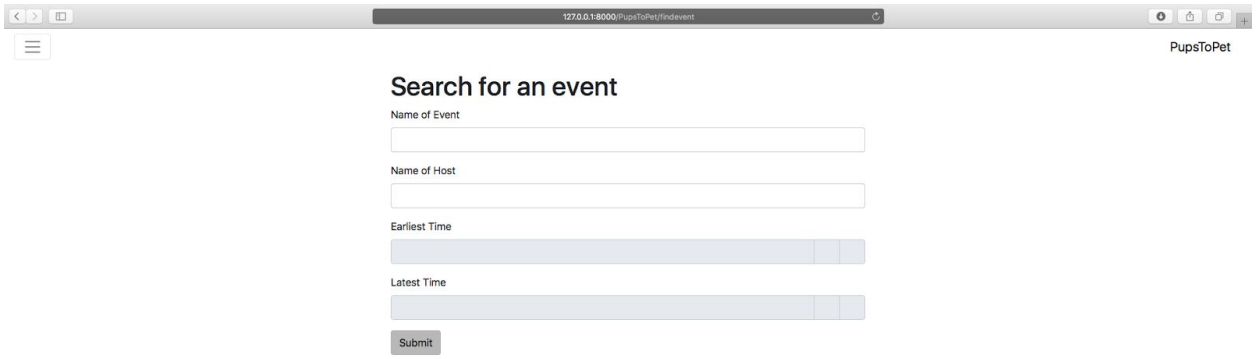
Dashboard: a page that displays events and shows their locations on a map (image from logged in state)



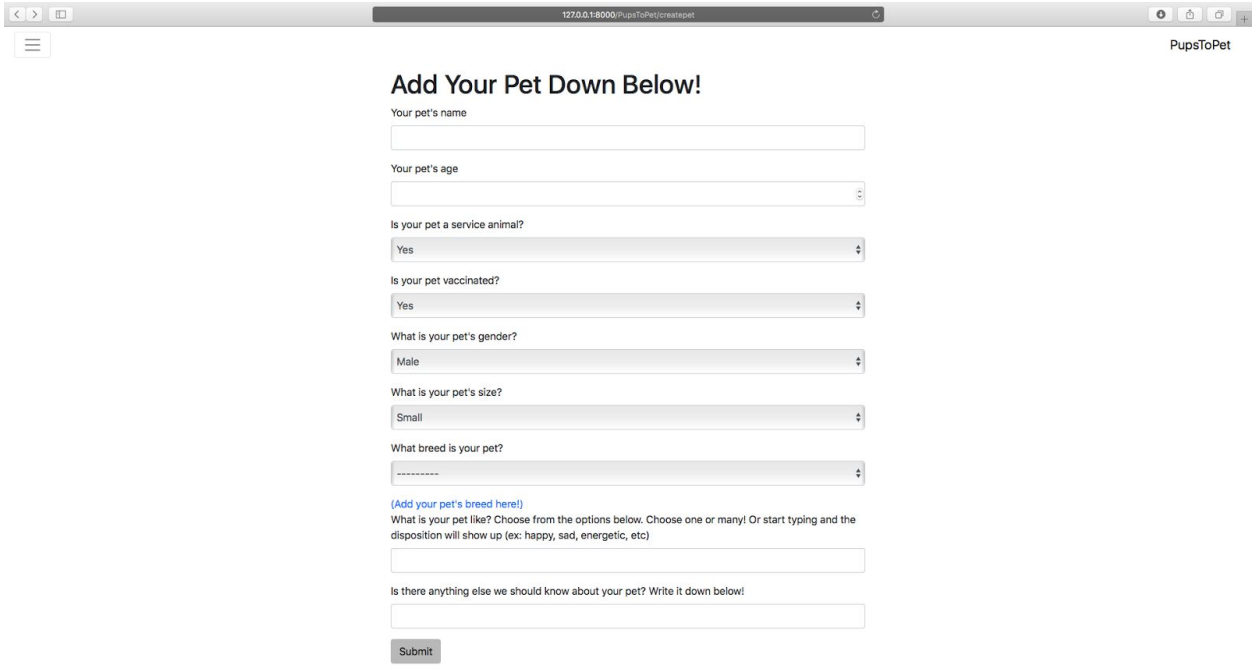
Create Event: a paged that an authenticated user can use to create an event



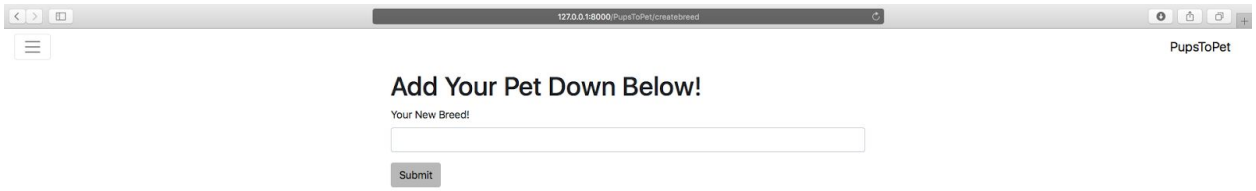
Find Event: a page any user can use to filter events displayed on the dashboard



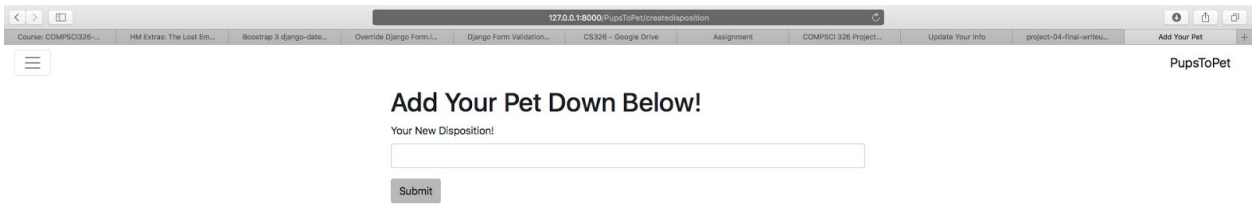
Add a Pet: A page an authenticated user can use to add a pet



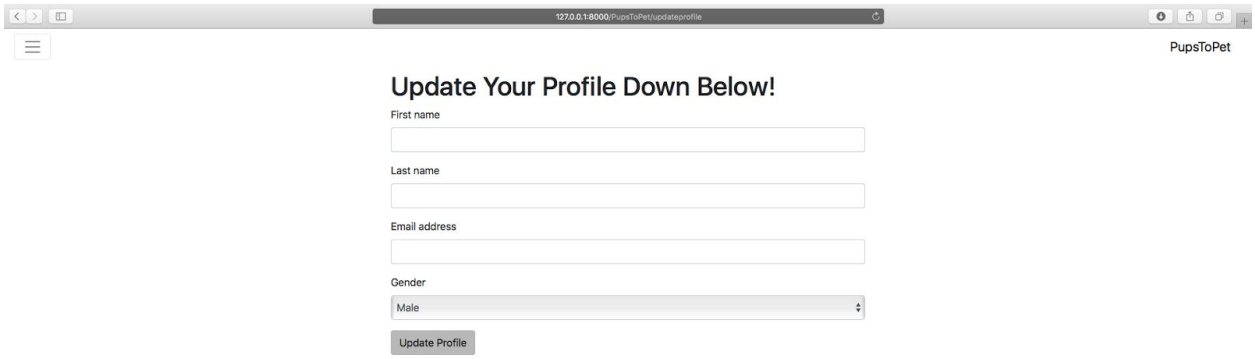
Add a Breed: a page an authenticated user can use to add a new breed to the database



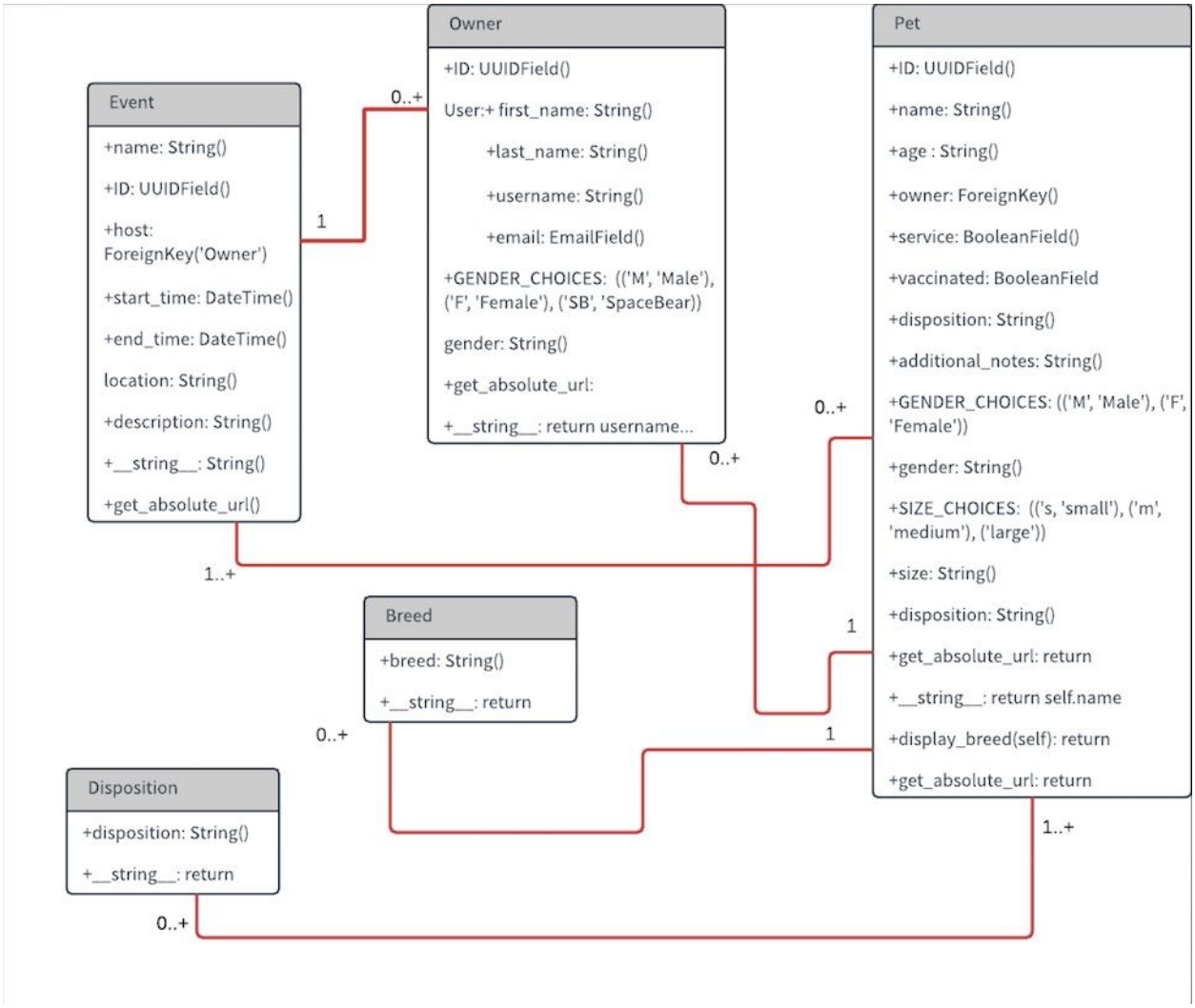
Add a Disposition: a page an authenticated user can use to add a new disposition to the database



Update Profile: a page an authenticated user can use to update their profile



Datamodel:
The model is as previously stands. We have owners, that are extensions of django's users and have can have zero to many pets and can create 0 to many events. Events have a single host (owner), but may have 1 or more pets. Pets have one owner, and can be of mixed breed, having 1 to many breeds, can attend 0 to many events, and can have 1 to many dispositions. Breeds and dispositions can be associated with 0 to many pets each.



URL routes/maps:
/: our about page
dashboard: a page displaying events and their locations on a map
createaccount: must be logged out, account creation form
updateprofile: must be logged in, form to update user profile
userinfo: must be logged in, displays information about the user
createevent: must be logged in, form to create an event
findevent: form to filter events visible on dashboard
createpet: must be logged in, form to create a new pet
createbreed: must be logged in, form to create a new breed for pets
createdisposition: must be logged in, form to add a new disposition for pets

Authentication/Authorisation: Our authentication is fairly minimal, as we really only have one type of user. Logging in is done though a login page users can find from the nav-bar. Logging out is done from the nav-bar. Authenticated users can create events and manage their account.

Unauthenticated users can create accounts. Either state of user can search for events and view the dashboard and about pages.

Team Choice: We had been notified that our user interface was less than satisfactory, and a large part of the final submission was an attempt to overhaul it. Unfortunately, given the time involved in this task and the time we each had available, the navigation bar was the only thing we were really able to substantially improve, much of the rest of the formatting was lost in the process.
We also wanted to try and give users the ability to filter the events displayed on the dashboard by means of a search. The ability to select criteria (specifically the name of the event, name of the host, earliest beginning of the event, and latest end of the event) on the search events page that would then take the user to a filtered dashboard was implemented.
More than anything, we wanted to polish out what we had...but the semester end rush and work to do was something we underestimated.

Conclusion: We had a number of difficulties. The team suffered from a number of mismatched schedules and a general shortcoming of communication. There were a number of cases where it seemed that people's understanding of certain parts of the project did not align. It seems that the front end was in general our team's greatest weakness. One thing that probably would've been useful to know in advance is that one can extend django's user model.