

## Team P - Back to Basics

### PUPG: Cute Royale

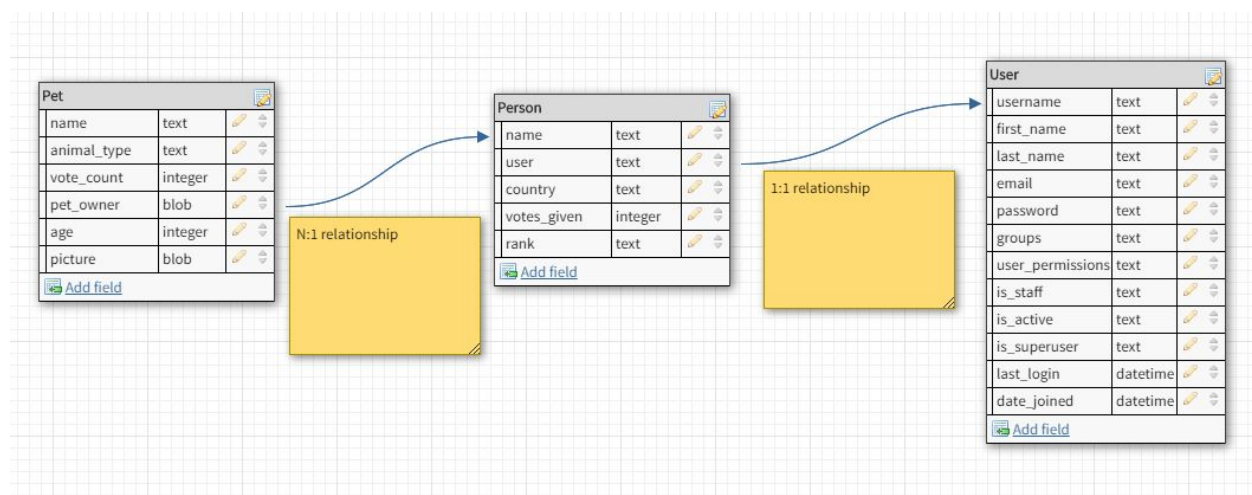
#### Fall 2018

**Overview:** Our group's project allows users to submit and rate pictures of cute pets. Users can create an account, and they are each ranked into tiers based on how many votes they distribute to other users' pets. The pets are ranked in a leaderboard and the number one pet is displayed on the homepage along with some random fun facts about other pets. The voting page also prevents users from voting for their own pet. Each pet has data on its image, name, species, age, and votes. Also, there is a help page and a logout function. Also, the leaderboard allows users to filter out only certain pet species. The database starts with 200 pets. The website also has a basic search function that returns the pet and user that you searched for. As well as a navigation bar at the top of the screen for users to navigate the website with.

**Team Members:** Joseph Capozzi, Patrick Robb, Jacob Grosner, Brandon Loo.

**User Interface:** \*See github docs\*

**Data Model:**



**URL Routes/Mappings:**

- /admin/ - Used to access the admin view
- /registration/ - Used to register a new user
- /updateprofile/ - Used to update Person information. This is login required.
- /accounts/ - Used for logging in and out
- /PUPG/ - Used for the homepage
- /PUPG/submit/ - Used for submitting pets to our database. Login required.
- /PUPG/leaderboard/ - Used for viewing the pet vote leaderboard.
- /PUPG/vote/ - Used for voting on pets. Login required.
- /PUPG/person/<int:pk>/ - Used for Person detail views.
- /PUPG/pet/<int:pk>/ - Used for Pet detail views.
- /PUPG/vote/vote/<int:id>/ - Used to update vote fields. Login required.
- /PUPG/search/ - Used for searching.
- /PUPG/profile - Used to access "my profile." Login required.
- /PUPG/leaderboardspecies/ - Used to see species leaderboard page.
- /PUPG/help/ - Used for help page which explains how to use the website.

**Authentication/Authorization:** We used Django's built-in authentication and authorization system. The User Class from Django's built-in authentication is paired with a Person class used in our data model. Users are able to register Users, log in, and log out.

**Team Choice:** Our team choice was storing pictures in the database. Our app allowed users to upload pictures of their pets, meaning each pet had an image field associated with it. These images were then displayed along with the pets names throughout the app(leaderboard, voting, etc.) We also implemented a search bar, that allows users to search for pets or owners, by first name.

**Conclusion:** In conclusion, our team's first difficulty was running the server, until we went to office hours and figured it out. After that, the website began to update smoothly and we learned many things about creating websites and design. The site gave many other additional issues such as storing image files to the database and logging users in and out. Also, everyone was able to come together with their ideas to filter out the site to make it better from the initial vision, and many changes were made. It would have made it easier to understand our final goals when we started, because the plan changed many times during the development stages.

**Overview:** *A brief overview of your application. This will be based on what you are submitting as your final web application artifact. You should also mention why your application is innovative.*

**Team Members:** *A list of your team members*

**User Interface:** *A final up-to-date list/table describing your application's user interface. This should include the name of the UI view and its purpose. You should include a screenshot of each of your UI views.*

**Data Model:** *A final up-to-date diagram of your data model including a brief description of each of the entities in your model and their relationships.*

**URL Routes/Mappings:** *A final up-to-date table of all the URL routes that your application supports and a short description of what those routes are used for. You should also indicate any authentication and permissions on those routes.*

**Authentication/Authorization:** *A final up-to-date description of how users are authenticated and any permissions for specific users (if any) that you used in your application. You should mention how they relate to which UI views are accessible.*

**Team Choice:** *You should briefly mention your team choice component for your application. You should highlight any additions to your application that are part of your team choice, such as, URL routes, UI views, data model, to make it clear what your team choice addition is.*

**Conclusion:** *A conclusion describing your team's experience in working on this project. This should include what you learned through the design and implementation process, the difficulties you encountered, what your team would have liked to know before starting the project that would have helped you later, and any other technical hurdles that your team encountered.*

Part 3 (slideshow) reqs:

Here is the info requested in part 3 of the final project specification, we took it off the slides and put it down here, for sake of convince.

- Title: PUPG Cute Royale
- Team Members: Jacob Grosner, Joseph Capozzi, Brandon Loo, and Patrick Robb
- PUPG allows users to submit and vote on cute pets, and a leaderboard tracks whose pets are the cutest. Users can find pets by searching via name, and can filter the leaderboard by species.
- Our audience will be people with cute pets, and people who like looking at cute pets. User do not need to upload a pet to use our site, to ensure that both target audiences are hit.
- Storing an image was definitely the most interesting part of our site, and it's pretty cool that now pets have their very own imageField model associated with them. Another interesting part was figuring out how to extend User to our Person class, allowing Users to be linked to Pets.

- A list of what your team found easy and what your team found hard: We found python scripting pretty easy, but designing our database was challenging, and our models changed multiple times throughout the semester. Storing an image was also a bit of a challenge.
- Future work - We'd like to go on a live server (Heroku) if possible, and in terms of functionality, would like to implement some basic filtering to keep the site family friendly. For example, if someone enters a curse word for a pet name, or uploads an inappropriate picture for a pet's image, we have no way of detecting that.

Note: To see our final UI presentation(besides just visiting the site), please check Docs/final\_ui\_pres2.pdf, on the Github repo.