Petr: Uber for Pets

Seth Pendergrass (sethdp), Brendan Redmond (bredmond)

Vision

What is product?

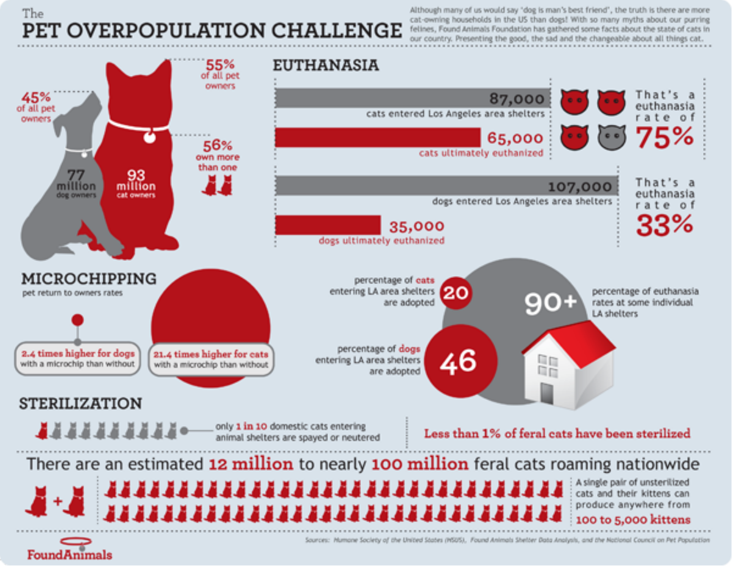
Our product would allow people to rent pets directly from other people. Think “Uber for pets”.

Whom is it for?

People who want a pet but don’t want to commit and people who have pets, but want to monetize an idle resource.

What problem does it solve?

Some people don’t want to commit to owning and caring for a pet long-term and some people have pets that they want to make money off of. Our product would match these people up (and take a commission).



What alternatives are available?

https://www.borrowmydoggy.com/ - Unfortunately, this site requires a membership, is not mobile-first and only allows dog to be rented out. We do not want to discriminate against other species.

Why is it compelling/worth developing?

Millions of people own pets that just sit in their house all day. In the spirit of the sharing economy, we have identified these pets as idle resources that their owners can profit from.

Thousands of pets are euthanized every year because their owners didn’t want them. Petr will reduce this by allowing (otherwise) negligent pet owners to rent a pet instead of owning one and then abandoning it.

Describe objectives, differentiators, target customers, scope, competitive analysis

Petr is different from alternatives like borrowmydoggy because it allows pets of all species to be rented, not just dogs. What if you always wanted a fish, but didn’t want to commit 6 months of your life to caring for it?

Software Architecture

What is the architecture? (Describe components)

Petr will use a database service to store records for user accounts, pet profiles, pet ratings, etc. The database will be accessed/modified through API servers. For our web application, we will have web servers which will talk to those API servers to build web pages. For our mobile application, the phones will talk directly to the API servers.

How will we implement functionality?
Our database service could be something like DynamoDB, and then our web servers and API servers could be EC2 instances. The servers will run Ruby on Rails. We will use React as our front-end framework. We will start with an Android app, then possibly progress to an iOS app if time permits.

What is interesting technically?
Allowing a user to use different devices and keeping the data synchronized across them.
Matching pet owners with pet requesters.

What languages/toolkits?
Ruby on Rails
React

Challenges and Risks
What is most serious risk?
Scaling the pet matching process could be an issue once Petr becomes as big as Uber.

How is risk minimized?
We will design with scale in mind so that we can identify issues early on, and we don't end up having to rearchitect Petr to work for billions of users.