Django vs. Ruby on Rails

A Pet Shelter Website

Zhen (Marco) Xu
zxu3@ualberta.ca

django

RAILS

Table of Contents

- Similarities
- Differences
- The Pet Shelter Website
- References

Similarities

- Model-View-Controller (MVC)
- Representational-State-Transfer (REST)
- Database integration (back end)
- Template utilization (front end)
- Test driven designing
- … …

Differences – installation and creation

```
sudo pip install django
sudo apt-get install python-mysqldb
django-admin.py startproject Django
cd Django; python manage.py runserver 8080
```

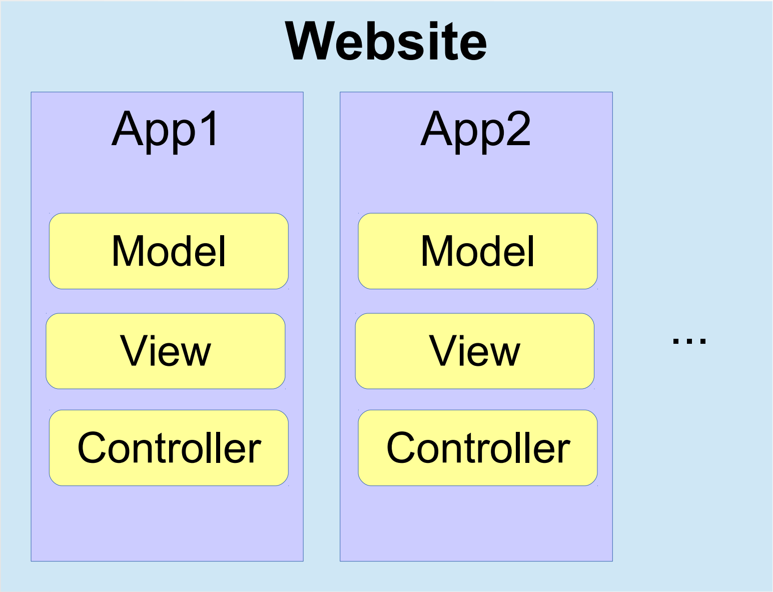
```
sudo gem install rails
# \curl -sSL https://get.rvm.io | bash -s stable --rails
sudo apt-get install libmysqlclient-dev
sudo gem install mysql2
rails new Rails -d mysql
cd Rails; rails server -p 3030
```

- Javascript runtime on Linux

Differences – architecture

django

- App oriented MVC
- Easy to extend



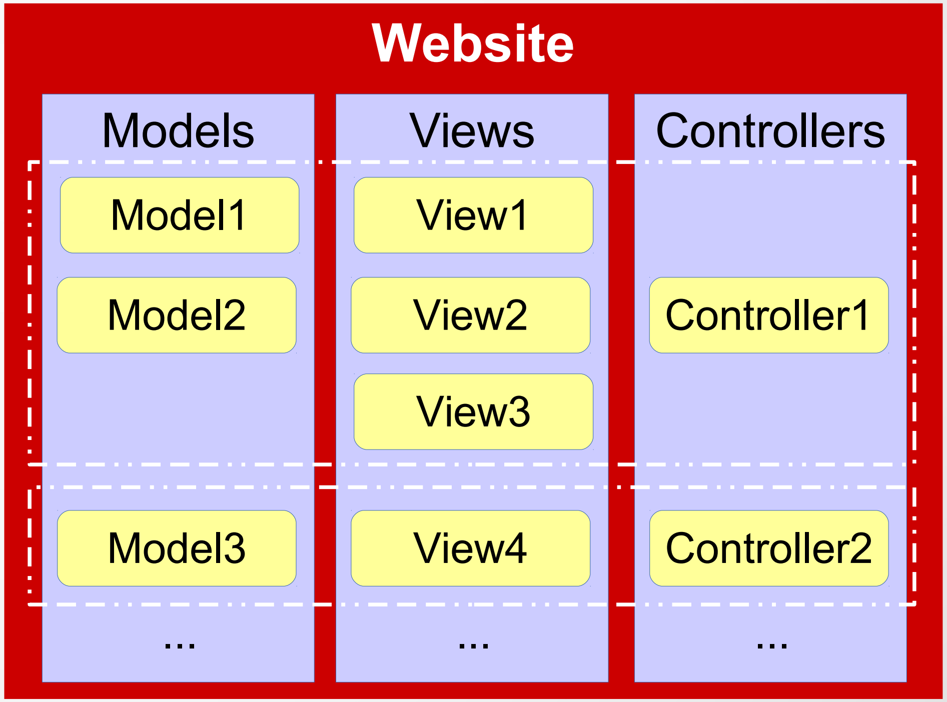
Website

App1: Model, View, Controller

App2: Model, View, Controller

…

- No apps



Website

Models: Model1, Model2, Model3, …

Views: View1, View2, View3, View4, …

Controllers: Controller1, Controller2, …

Differences – model

- Models: objects in database (tables)
- Each model -> table of database
- Variable -> column of table

django

- Source codes:

```
class People(models.Model):
  name = models.CharField()
  # ... ...
```

- Synchronize to database:

```
python manage.py syncdb
```

RAILS

- Terminal commands:

```
rails g model people \
  name:string \
[--force]
```

- Migrate to database:

```
rake db:migrate
```

Differences – controller

django

- Controllers are methods
- Located in “`views.py`”
- Controller methods may:
  - Render a template page
  - Respond message
  - Redirect to another method
- Pass variables by dictionary

RAILS

- Terminal commands:

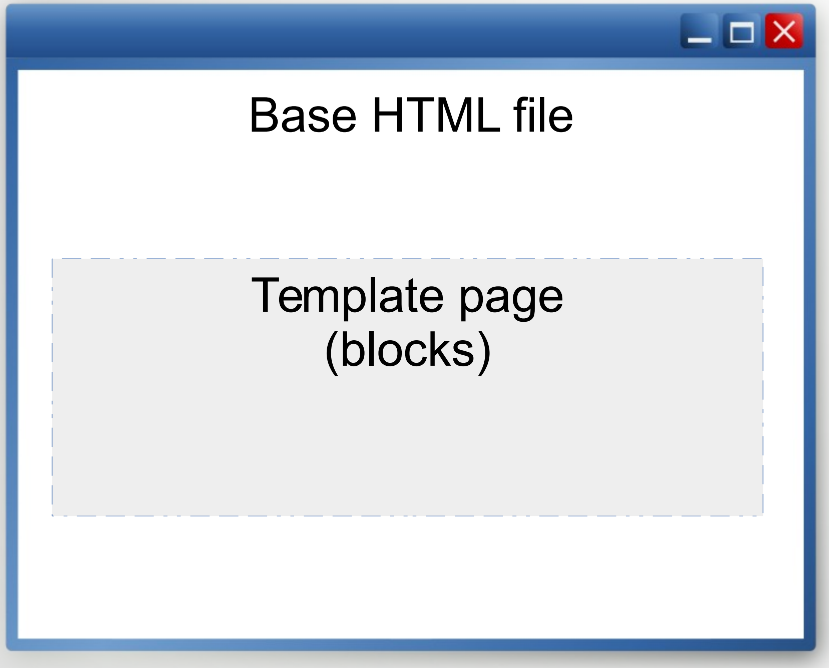
```
rails g controller welcome \
[--force]
```

- By default, controllers will render template pages in the same name
- Pass global variables to template pages

Differences – view

django

- Template HTML pages
  - Pieces of HTML codes
  - Extend from base pages

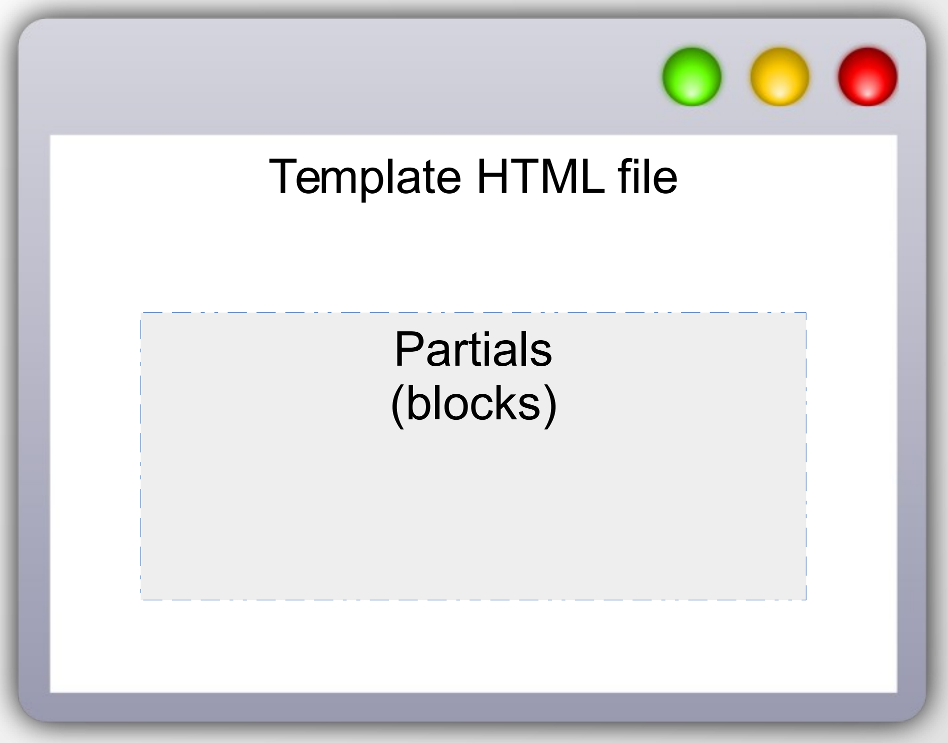


Base HTML file

Template page
(blocks)

RAILS

- Codes extended from pages
- Template extend outer “partials”



Template HTML file

Partials
(blocks)

Differences – view (continue)

django

- Load static files (CSS, js, etc.)
  - Separated from templates
  - Specify as “static”
- Serve multimedia
  - Same as serving static files

RAILS

- CSS, JS in Ruby on Rails
  - Static files are “assets”
  - Any file in “assets” directory will be applied automatically
- Multimedia
  - Mostly in “public” area
  - Specify and managed as “images”, “audios” and “videos”

The Pet Shelter Website - requirement

- Information:
  - shelters, staffs, animals
- Activities:
  - receipts, adoptions, surrenders, donations, volunteers
- Applications:
  - adopting, surrendering, donating, volunteering
- Management:
  - shelters, staffs, animals; requests from above applications

The Pet Shelter Website - configuration

- Database

django

```
# ./Django/settings.py

DATABASES = {
  'ENGINE': \
    'django.db.backends.mysql',
  'NAME': 'Django',
  'USER': 'MarcoXZh',
  'PASSWORD': 'project',
}
```

RAILS

```
# ./config/database.yml

development:
  adapter: mysql2
  database: Rails_development
  username: MarcoXZh
  password: project
test:
  database: Rails_test
production:
  database: Rails_production
```

The Pet Shelter Website - configuration

- Other configurations

django

```
# ./Django/settings.py
INSTALLED_APPS = (
  # ... ...
  'PetShelter',
)
TIME_ZONE = 'America/Edmonton'
USE_TZ = False


# MySQL terminal:
GRANT ALL ON Django.* TO 'MarcoXZh'@'localhost' IDENTIFIED BY 'project';
```

RAILS

```
# Nothing more to do!
```

The Pet Shelter Website - REST

- Routes (URIs)

```
# ./Django/urls.py
urlpatterns = patterns('',
  # .. ..
  url(r'', include('PetShelter.urls')),
)
# ./Django/PetShelter/urls.py
urlpatterns = patterns('',
  url(r'^$', views.index, name='index'),
  # ... ...
)
```

```
# ./config/routes.rb
Rails::Application.routes.draw do
  root 'petshelter#index'
  get 'animals/' => 'petshelter#animals'
  # ... ...
end
```

The Pet Shelter Website - controller

```
# ./PetShelter/views.py
def index(request):
  Return HttpResponse('Welcome to PetShelter!')
def animals(request):
  animals = Animals.objects.all()
  Return render(request, 'PetShelter/animals.html', animals)
# ... ...

class PetshelterController < ApplicationController
  layout 'base' # 'layouts/base.html.erb'
  def index
    # Rails renders "index.html.erb" by default
  end
  def animals
    @animals = Animal.all
    render 'petshelter/animals'
  end
  # ..  ...
```

The Pet Shelter Website - view

- Layout (base file)

django

```
<html>
<head>
  <title>
    {% block title %}
    {% endblock %}
  </title>
</head>
<body>
  {% block body %}
  {% endblock %}
</body>
</html>
```

RAILS

```
<html>
<head>
  <title>
    <%= yield :title %>
  </title>
</head>
<body>
  <%= yield :body %>
</body>
</html>
```

The Pet Shelter Website - view

- Template

```
{% extends "PetShelter/base.html" %}

{% block title %}
  Welcome to the "PetShelter"
{% endblock %}
{% block body %}
  <p> Welcome to PetShelter! </p>
{% endblock %}
```

```
<% content_for :title do %>
  Welcome to "PetShelter"
<% end %>
<% content_for :content_block do %>
  <p> Welcome to PetShelter! </p>
<% end -%>
```

The Pet Shelter Website - view

- Ruby on Rails: template rendering partials

```
<!-- index.html -->
<html>
<body>
  <%= render 'body' %>
</body>
</html>

<!-- _body.html.erb -->
<p> Welcome to PetShelter! </p>
```

The Pet Shelter Website - view

- Static files in templates

```
{% load staticfiles %}
<!-- ./PetShelter/static/PetShelter/css/style.css -->
<link rel="stylesheet" type="text/css"
      href="{% static 'PetShelter/css/style.css' %}" />
<!-- ./PetShelter/static/PetShelter/javascript/script.js -->
<script src="{% static 'PetShelter/javascript/script.js' %}">
</script>
```

```
<!-- app/assets/stylesheets/style.css -->
<%= stylesheet_link_tag "style" %>
<!-- app/assets/javascripts/script.js -->
<%= javascript_include_tag "script" %>
```

The Pet Shelter Website - view

- Multimedia files in templates

django

```
{% load staticfiles %}
<!-- ./PetShelter/static/PetShelter/mm/image.jpg -->
<img src="{% static 'PetShelter/mm/image.jpg' %}" />
<!-- ./PetShelter/static/PetShelter/mm/audio.wav -->
<audio src="{% static 'PetShelter/mm/audio.wav' %}" />
<!-- ./PetShelter/static/PetShelter/video.mp4 -->
<video src="{% static 'PetShelter/mm/video.mp4' %}" />
```

RAILS

```
<!-- ./public/images/image.jpg -->
<%= image_tag 'image.jpg' %>
<!-- ./public/audios/audio.wav -->
<%= audio_tag 'audio.wav' %>
<!-- ./public/videos/video.mp4 -->
<%= video_tag 'video.mp4' %>
```

The Pet Shelter Website - view

- Forms

```
<form method='POST' action="/">
  {% csrf_token %}
  <!-- input tags -->
  <input name="name" />
  <input type="submit" />
</form>
```

```
<form method='POST' action="/">
  <input name="authenticity_token" type="hidden"
         value="<%= form_authenticity_token %>" />
  <!-- input tags -->
  <input type="submit" />
</form>
```

The Pet Shelter Website - view

- Form

django

```
def index(request):
  if request.method == 'POST':
    v = request.POST['name']
    # do POST things
    pass
  # do GET things
```

RAILS

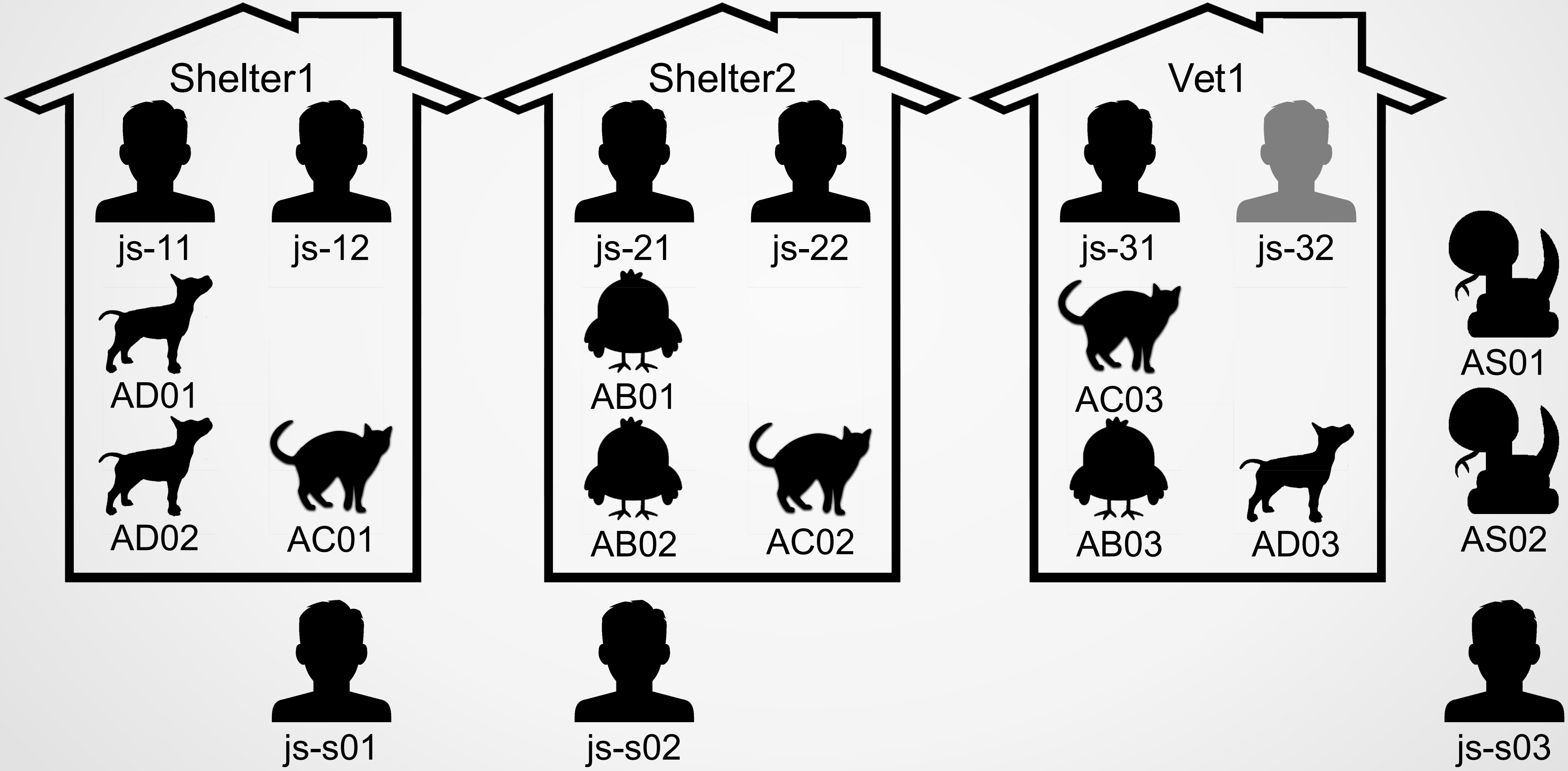
```
# route for POST
post '/' => 'petshelter#index'

def index
  if nil != \
   params['authenticity_token']
    # do POST things
  else
    # do GET things
  end
end
```

The Pet Shelter Website - view

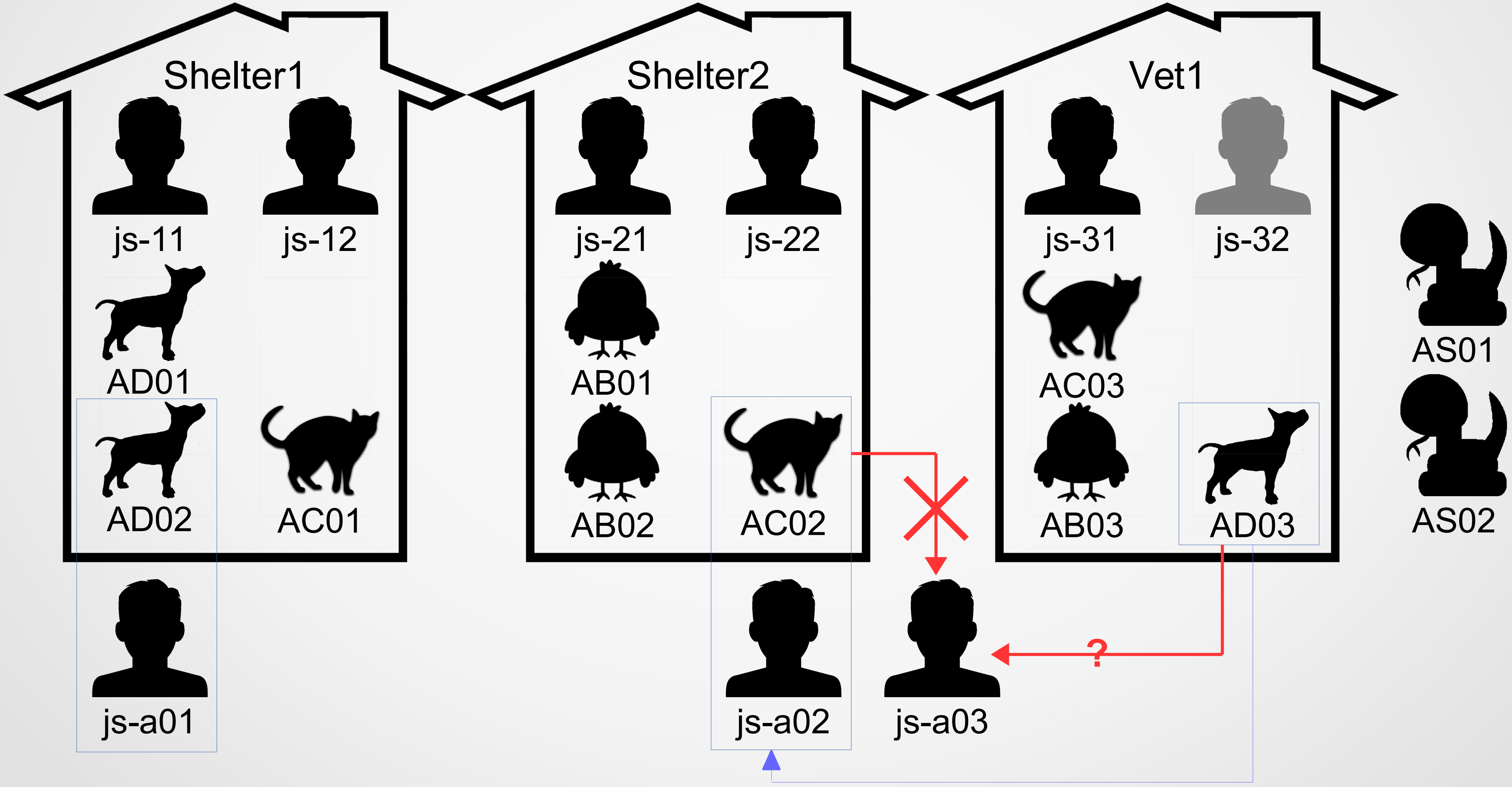
- Ruby on Rails: `form_for` method

```
<%= form_for :post do |f| %>
  <!-- input tags -->
  <%= f.text_field :name %>
  <%= f.submit %>
<% end %>

# route for POST
post '/' => 'petshelter#index'

def index
  if params[:post] != nil
    # do POST things
  else
    # do GET things
  end
end
```

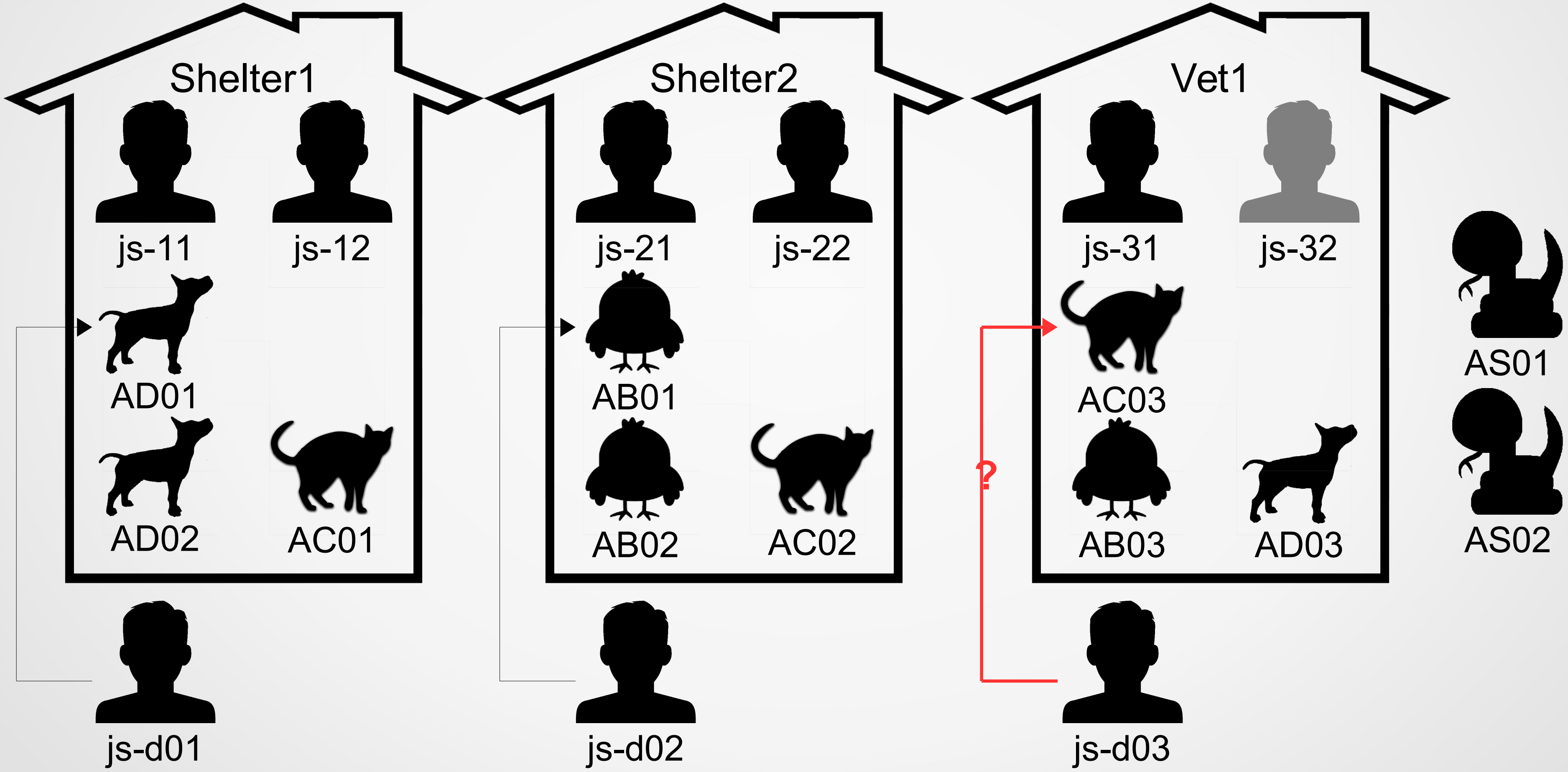
The Pet Shelter Website - scenario



The Pet Shelter Website - scenario



The Pet Shelter Website - scenario



References

- https://www.djangoproject.com/
- https://docs.djangoproject.com/en/1.6/intro/tutorial01/
- http://rubyonrails.org/
- http://guides.rubyonrails.org/getting_started.html
- http://guides.rubyonrails.org/active_record_querying.html
- http://guides.rubyonrails.org/layouts_and_rendering.html