CODE COMMENTS FOR ASHH

BY TEAM CYL

Lucy Yu, Chongyuan Xiang, Yang Chen

1. create method in requests_controller:

```
if request_params[:clothe_or_outfit] == 'c'
  @request = Request.new(:status => request_params[:status], ...

elsif request_params[:clothe_or_outfit] == 'o'
  Outfit.find(request_params[:outfit_id]).clothes.each do |item|
    @request = Request.new(:status => request_params[:status], ...
    @request.save
  end
end

respond_to do |format|
  if @request.save
….
  else        // ERROR HANDLING
     …..
  end
end
```

1. create method in requests_controller:

- **respond_to only correct when request_params[:clothe_or_outfit] == 'c'**

- **If request_params[:clothe_or_outfit] == 'o'**
    - Those requests already saved before respond_to |format|
    - Consider the case where some requests aren’t saved successfully

Request #1	✗
Request #2	✗
Request #3	✓

This situation is not handled, and may create some weird intermediate states in the database.

2. in request.rb:

```
# approve a request
 def approve
    self.status = "approved"
    self.save
  end

  # decline a request
  def decline
    self.status = "declined"
    self.save
  end
```

- Seems like one can change the status of a request multiple times & no error is given (e.g. approving a certain request from one open browser window then declining the request from another)
- If intended that request status can be changed again, add a notification?
- Otherwise, add a check to disable further status changes.
- Also, executing the sequence mentioned seems to have broken dashboard for some reason.

3. create method in outfits_controller:

```
def create
  @outfit = Outfit.new(outfit_params)
  clothing_ids = params[:clothes_ids]
  has_enough_clothes = false
  if(clothing_ids)
    has_enough_clothes = @outfit.has_enough_clothes(clothing_ids)
    if(has_enough_clothes)
      clothing_ids.each do |c_id|
        clothe = Clothe.find_by_id(c_id)
        if (clothe)
          @outfit.clothes << clothe
        end
      end
    end
  end
```

3. create method in outfits_controller:

- Consider the case:
  has_enough_clothes is true (there are >= 2 clothing ids) but only one of clothing IDs is valid,

  According to the code, it turns out the outfit will finally contain only one “clothe”.

- **Should do the has enough clothes check after filtering out invalid clothing ids.**