**Analysis**

**2. Create URL configuration and STUB view**

url(r'^qs/(?P<questionnaire\_id>\d+)/(?P<order\_info>\d+)/$',

view = 'get\_next\_questiongroup',

name = 'get\_next\_questiongroup'),

url(r'^qs/(?P<questionnaire\_id>\d+)/$',

view = 'get\_next\_questiongroup',

name = 'get\_next\_questiongroup'),

View :-

def get\_next\_questiongroup(request,questionnaire\_id,order\_info=None):

The URL takes at least questionnaire id or questionnaire id and order\_info digit. The argument order\_info = None in the view, indicates the argument can be empty

**3. Render a Group for a given Questionnaire (logic)**

View :-

**get\_next\_questiongroup**(request, questionnaire\_id, order\_info = None)

IF order\_info == None

order\_info = 1

ELSE order\_info = order\_info

(LOGIC)Based on order\_info, get **questiongroup\_id**

**#below is a method to build the form based on questiongroup\_id**

make\_question\_group\_form(**questiongroup\_id**)

IF request.POST #if the for is submitted

#get answers and save it.

Then order\_info = order\_info +1

Return questionnaire\_id, order\_info to the same method

Explanation :-

This view shows that at first instance with URL configuration of

../<questionnaire\_id>/<order\_info>/

If order\_info is not set, set it to 1, else set it on what the user has entered

Eg. question/1/1 or question/1/2

The parameter questionnaire id and order\_info, query will be met to check what questiongroup is associated with the two parameters.

And then the retrieved questiongroup\_id will be used in forms.py method named make\_question\_group\_form()

Skipping forward if the form has been saved or finalized etc, the set order\_info + 1 to retrieve the next set of questiongroup, using this method again

6. Logic to iterate the whole questionnaire, one group at a time

Question is – How do I make sure the method wont return out of range order\_info? (more @ 3rd Analysis)

It’s fairly simple, first get the count of all groups that belong to the questionnaire, then after each form save, check if current order\_info is equal to the count of all groups within that questionnaire. If it is. Then it means it is the last/final order\_info and the view will move on to the the questionnaire finish, else it will increment order\_info to the next one.

if request.method =='POST':

if order\_info == orderedgroups.count():

this = 'this is the last one!'

print this

else:

this = 'Continue! pass order\_info+1 !'

print this

order\_info = order\_info + 1

return HttpResponseRedirect(reverse('get\_next\_questiongroup', kwargs = {'questionnaire\_id': questionnaire\_id, 'order\_info' : order\_info}))