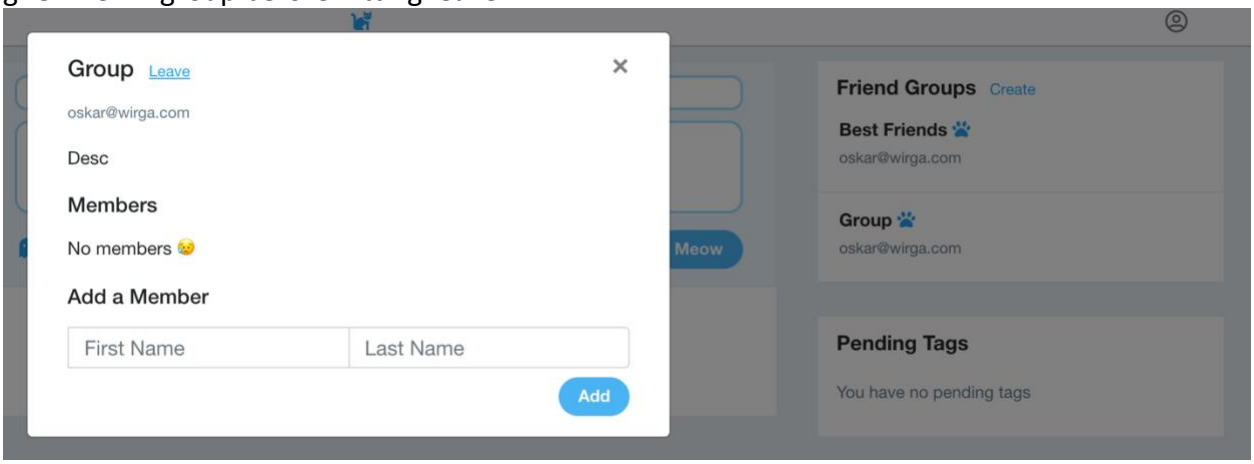
a. Oskar Wirga
b. Leaving a Friendgroup
c. Some people are fickle and don't deserve to have us in our lives and after we realize that, we need a mechanism in order to get rid of them. Leaving a Friendgroup is an important feature because after all, it is private content sharing, so we may want to increase our privacy by defriending others or leaving Friendgroups.
d. Wherever a Friendgroup primary key is, we need to add ON DELETE CASCADE in order to ensure that changes made to the main Friendgroup and Belong table are reflected throughout the entire database.
e. I use "SELECT * FROM Friendgroup WHERE fg_name = %s AND owner_email=%s" in order to check first of all if a group actually exists, then "SELECT * FROM Friendgroup WHERE fg_name = %s AND owner_email=%s" is ran with the leaving user's email to check if they own the group, if they do I run "DELETE FROM Friendgroup WHERE fg_name=%s AND owner_email=%s" in order to get rid of the Friendgroup entirely, otherwise I run "DELETE FROM Belong WHERE fg_name=%s AND email=%s AND owner_email=%s" in order to remove the user from the group.
f. db.py: Lines 366-387 show the flow of checking if the group exists, then if the user is the owner of the group, we delete it entirely, and if not we simply remove him from the group by deleting their entry in the Belong table.
app.py: Lines 295-302 show the simple flow of getting the group info from the form, then simply passing the data to the db function.
g. Owner in group before hitting leave



Friend Groups list following the leave function