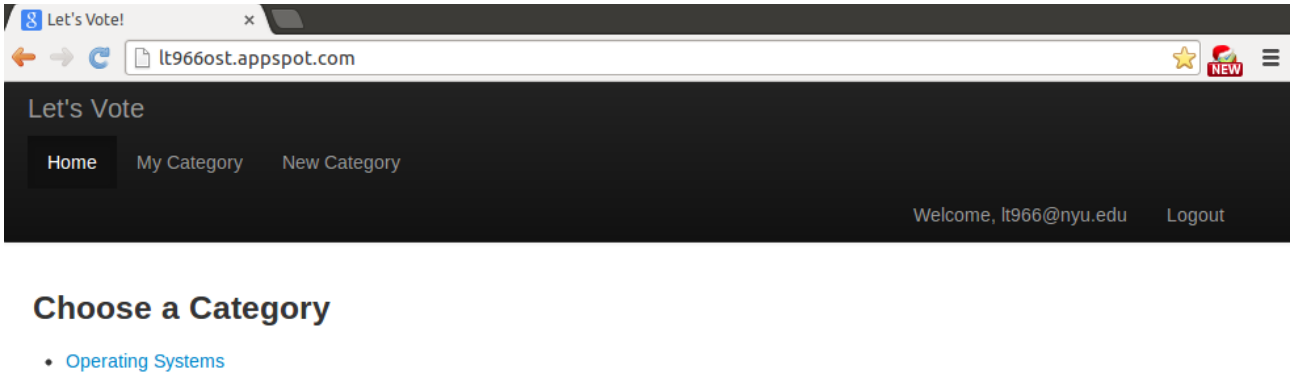
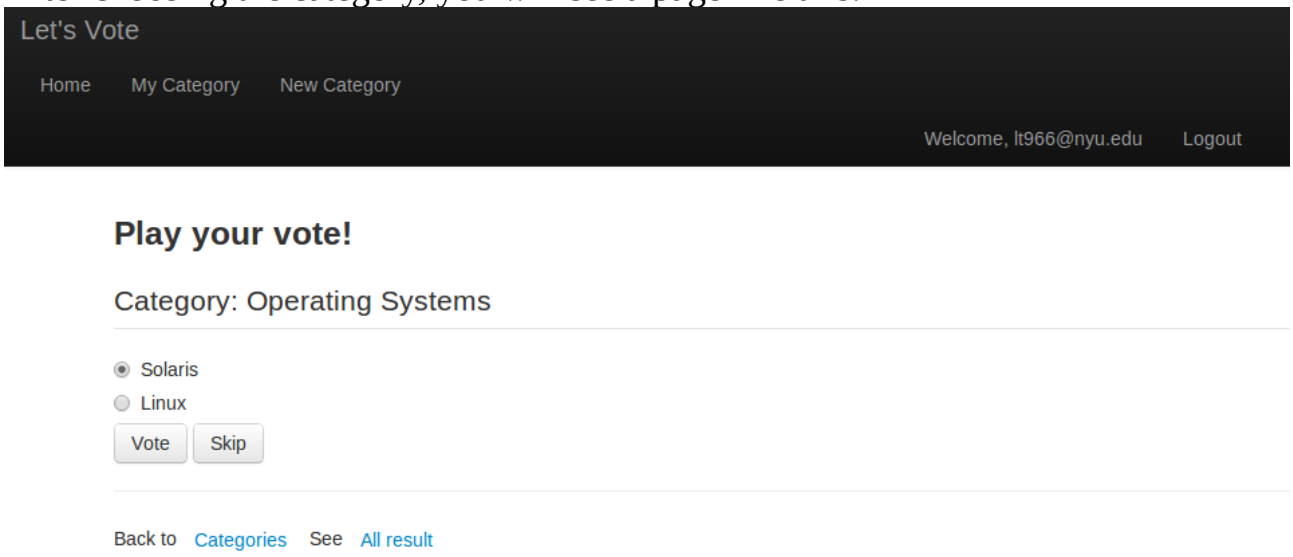
# User Guide

This is the index for my project called “Let's Vote!” .



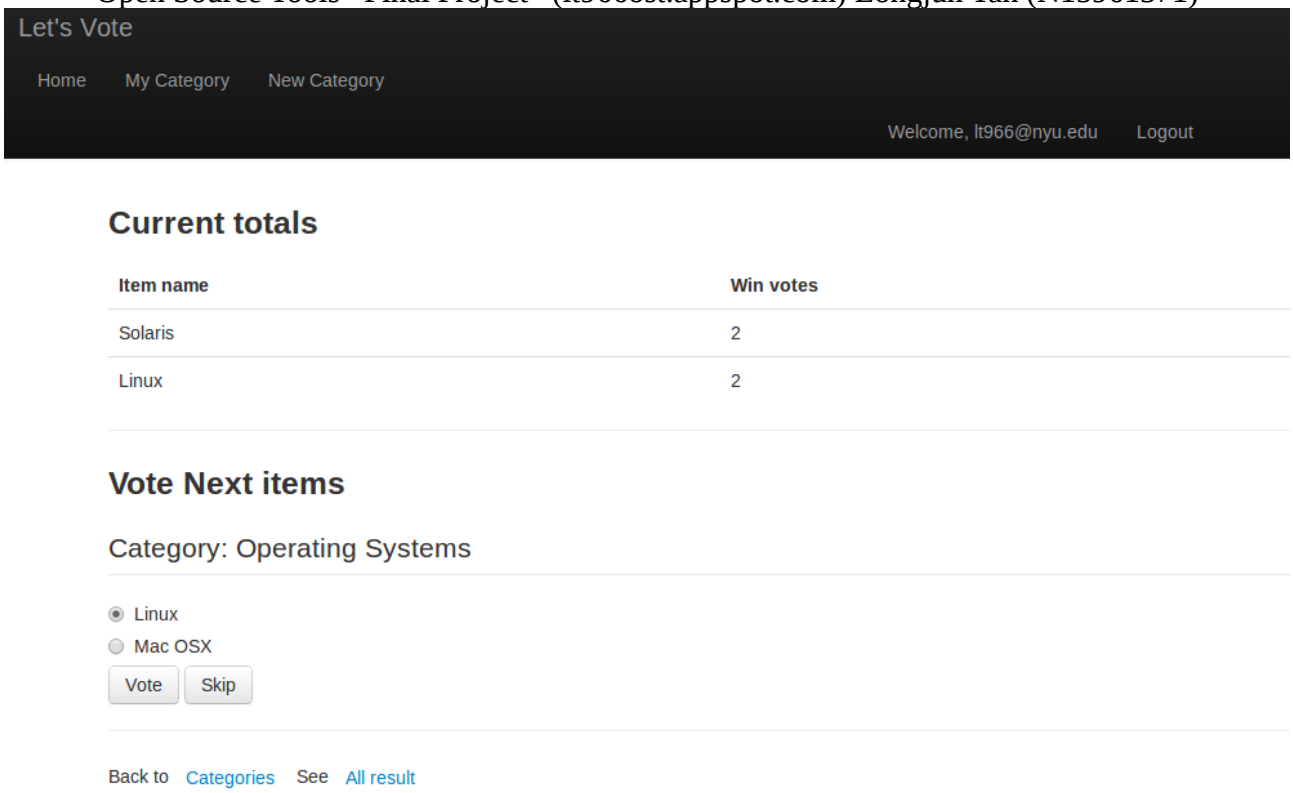

In this page, you will see the categories to vote. If no category in database, it will be blank.
You can click the link for the category you want to start a vote.

After choosing the category, you will see a page like this:



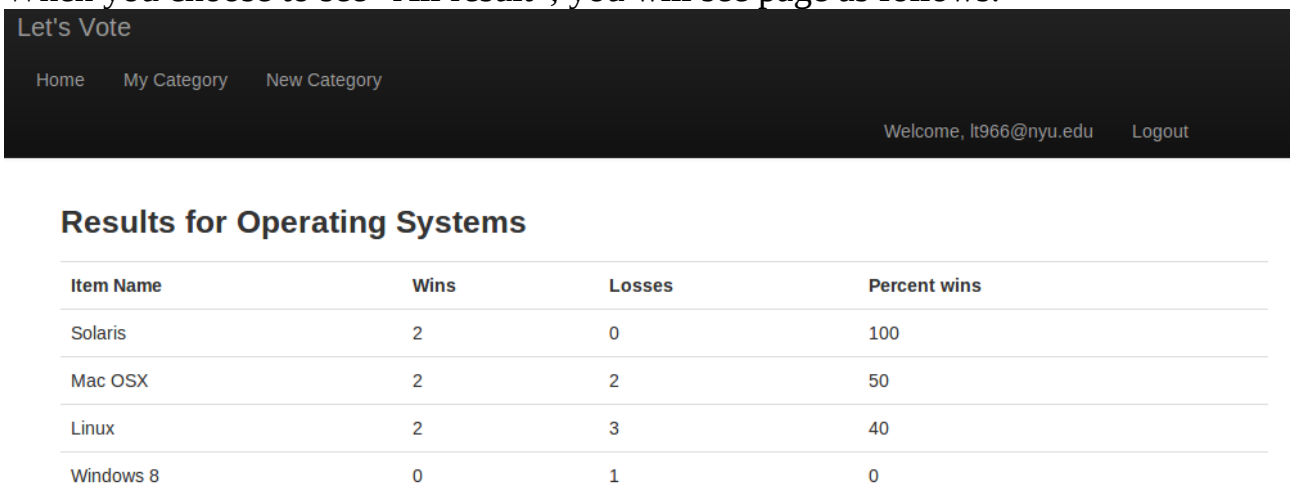

It's almost the same what shown in assignment 4.(http://cs.nyu.edu/~lt966/cgi-bin/test.cgi)

After voting, the page will be shown as follows:

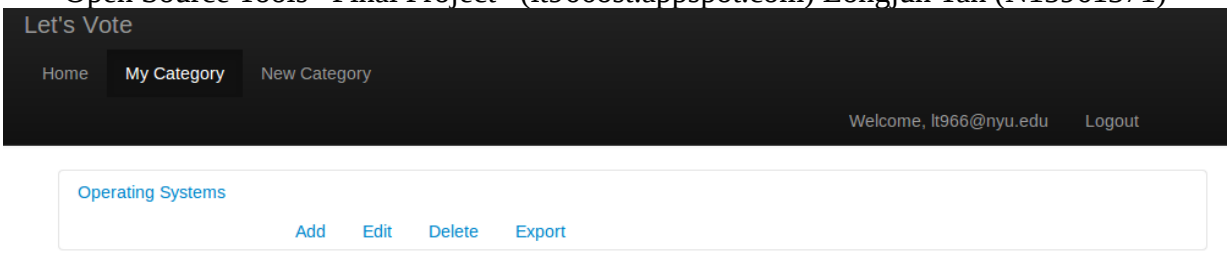



You will know the votes for both items related just know. Also, the another round voting starts.

If you click on the link ”Categories”, you go back to the home page.
When you choose to see “All result”, you wiil see page as follows:

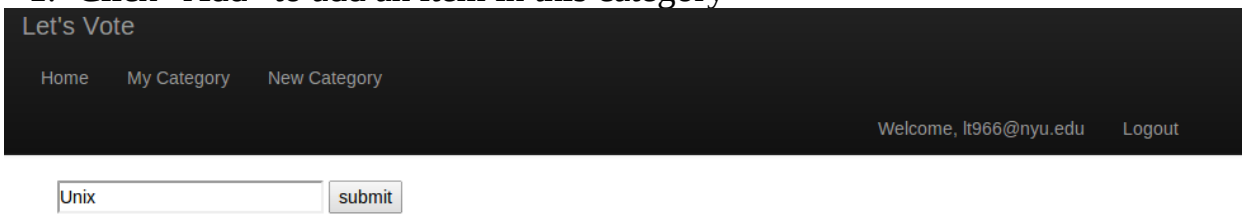



Above are all voting functions.
Next, you can click “My Category” to see the categories you create.

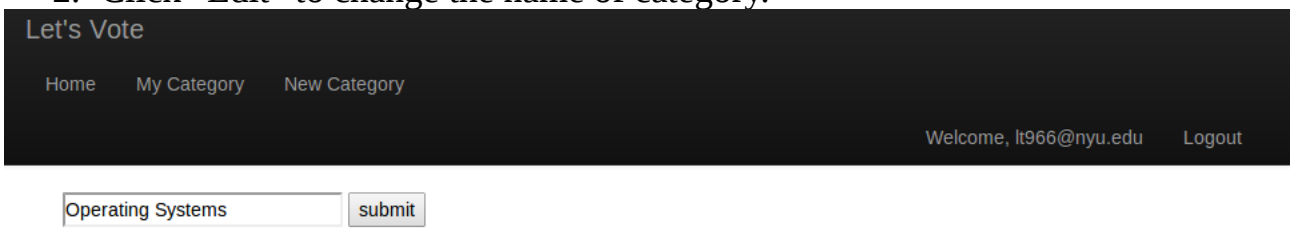



Here, the category Openrating Systems is created by the current user.

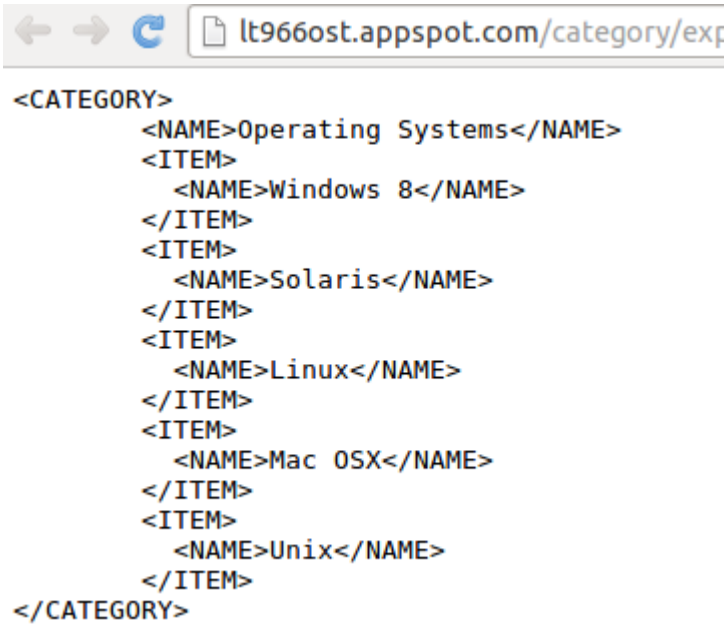
1. Click “Add” to add an item in this category




2. Click “Edit” to change the name of category.

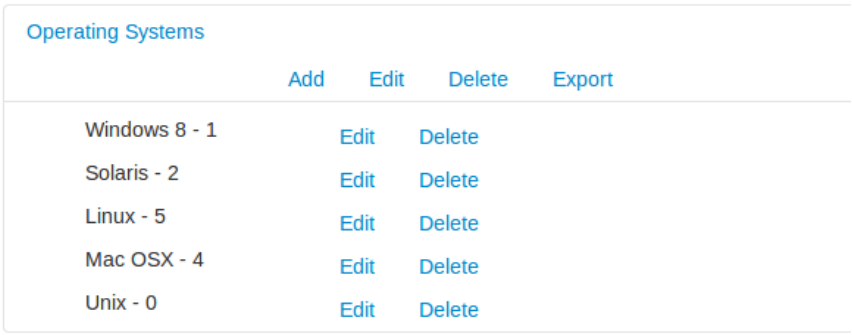



3. Click “Delete” to delete the whole category
4. Click “Export”, you will see the xml for this category



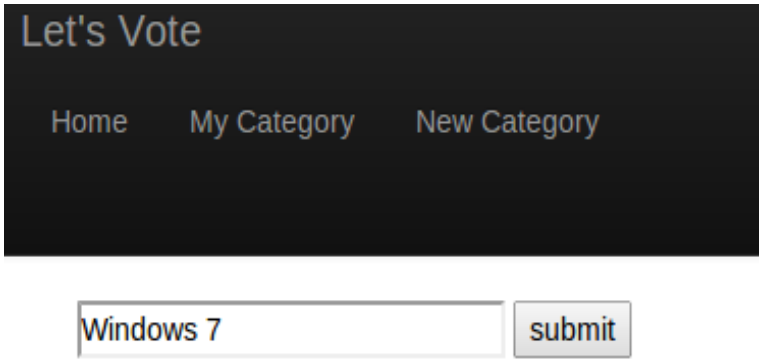

You can copy this to save as a XML file, to import later.

5. Click the link on category, you can see the list of items in this category.

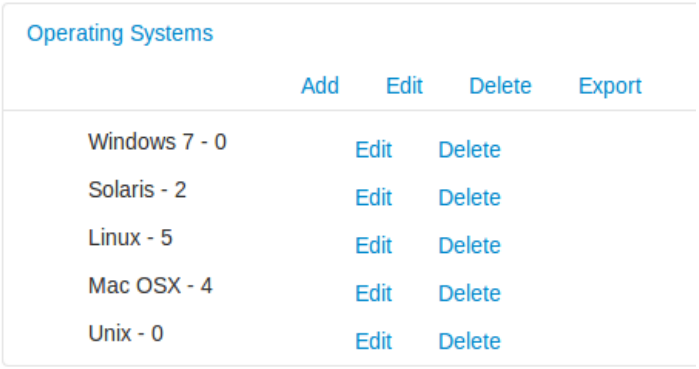



The number after the item shows current tatol votes it has, including losing ones.

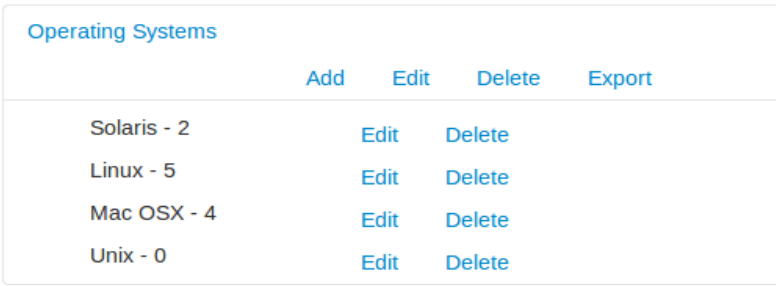
1. Click “Edit” to edit the name of item.




Then, the votes for this item will be deleted.




2. Click “Delete” to delete the item in this category.



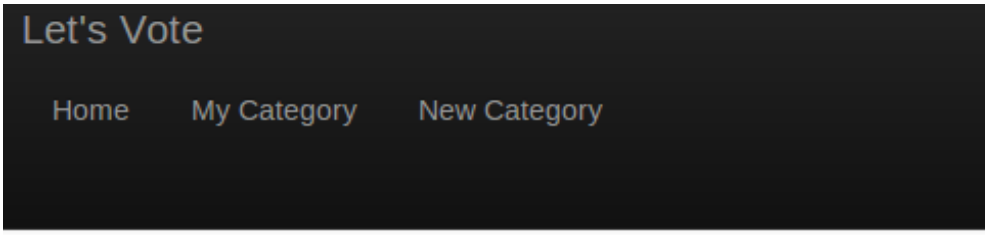

So, functions above meet parts of the first advanced feature: Allow a user to change the items in a category. When this happens, the results for that item must be removed.

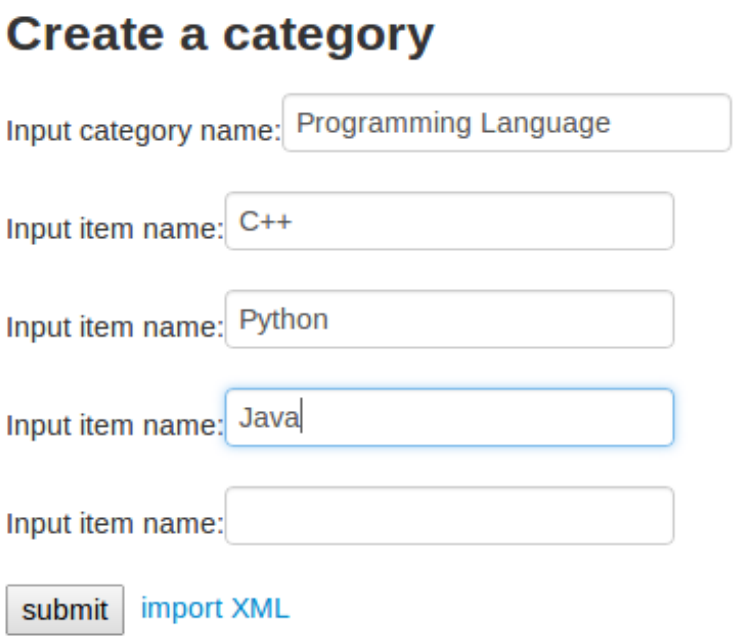
The rest of features, Also, imports must be supported that replace a

previous import (meaning items that do not change are preserved, items that are removed have results deleted)., will be shown next.

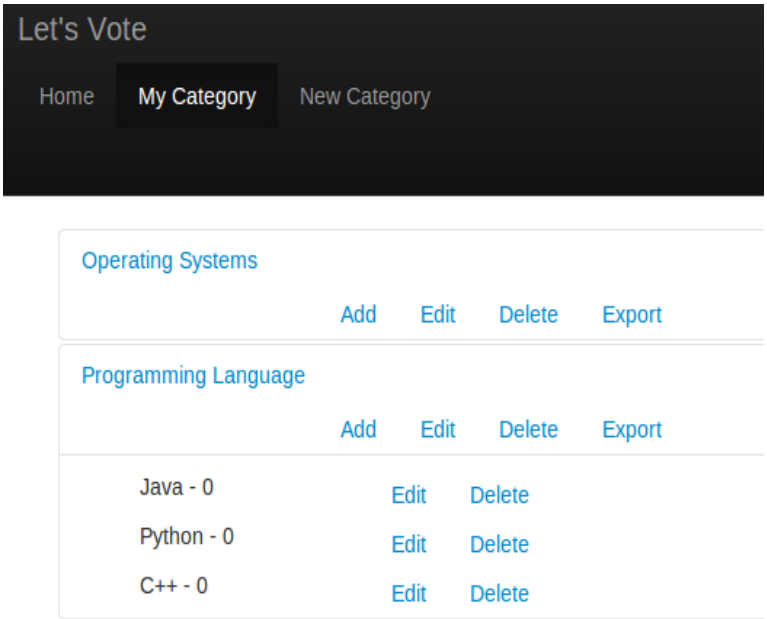
Next, click “New Category” to create a new category.

This time, it support to add more than one item once, which is more user friendly.

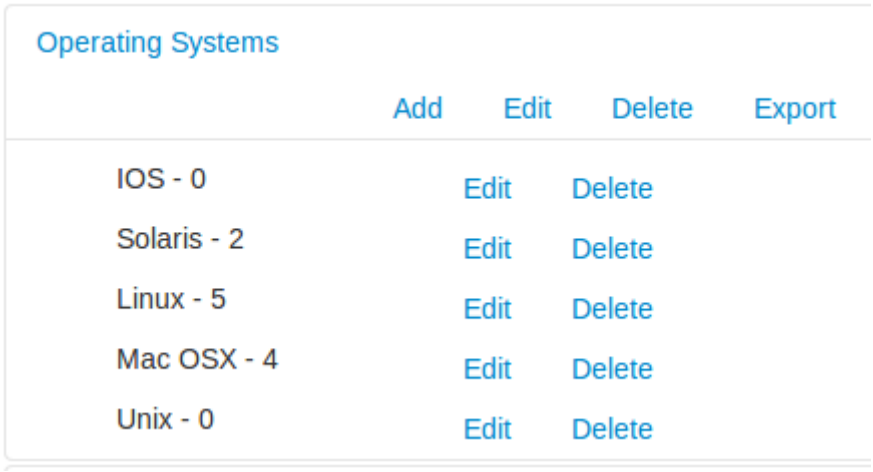


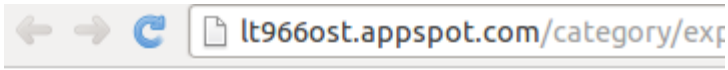

After submit, the category is created successfully.



Next step, let's try import XML file.
Before:

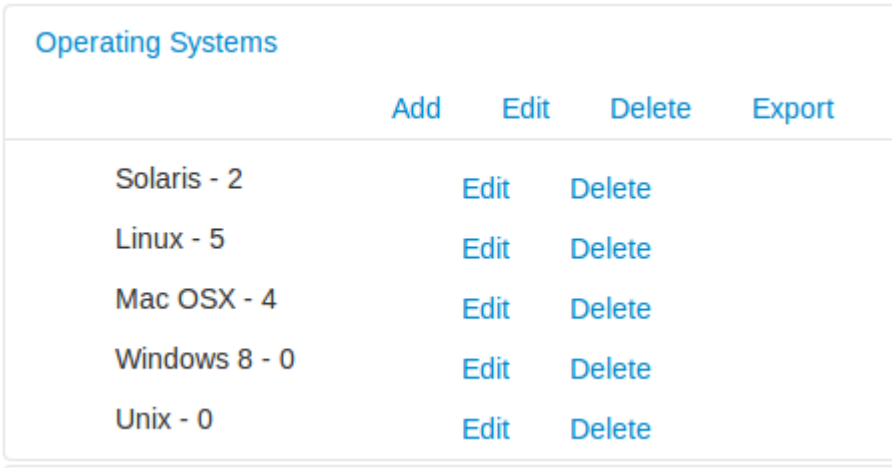



Using the XML created above:




```
<CATEGORY>
        <NAME>Operating Systems</NAME>
        <ITEM>
          <NAME>Windows 8</NAME>
        </ITEM>
        <ITEM>
          <NAME>Solaris</NAME>
        </ITEM>
        <ITEM>
          <NAME>Linux</NAME>
        </ITEM>
        <ITEM>
          <NAME>Mac OSX</NAME>
        </ITEM>
        <ITEM>
          <NAME>Unix</NAME>
        </ITEM>
</CATEGORY>
```

So, IOS should be deleted, Windows 8 should be added, other items' vote should preserved.
After importing, we get :




Everything goes fine!

Enjoy testing! Thanks for your work!

# Developer Guide

In my project folder, there are several python files, one yaml file and two subfolders. I will explain them in the order to create this website.

1. app.yaml is an configuration file for an App Engine application. Index.py is the entrace of this app.

2. Router.py is about webapp framework.

3. The first class View in views.py is in charge of users service, which meets one of professor's requirement – logged-in user only.

4. Models.py is for using datastore.
   By using db.ReferenceProperty, I design the database structure as:
   a) Category
      Category has a name and an owner, which is a User type. Also, one category has one collecion called items.
   b) Item
      A Item has its name. Also, an item has a collections as votes. Meanwhile, there are several methods for item, such as remove votes, counting winning votes.
   c) Vote
      A vote has a type (win or lose), also has a voter, which might be used in the future.

5. Views.py is the main file for this app.
   a) MainPage class is for the home page, to show all categories, then render to index.html
   b) Category_View is for the methods happened with category, such as create, delete, edit and so on. Since the voting is more likely category's function, I also put vote() in category class.
   c) Item_View is in charge of the methods on items, like delete, edit ,etc.

6. The template folder is using for the all HTML files which will be rendered in the views.py, which meets professor's requirement – must render HTML with templates.

7. The image folder includes css, images and js subfolder, which are used to optimize the User Interface. It's just for future extension.