**Station Group**

* This section includes the station type selection dropdown, a search-bar to search for a specific station group and the list of station groups in card form.
* API for the station category options is: **/stations/options**. This API gets the category and subcategory and is stored in the state in **category\_list**.
* API for station group list: **/stations/groups/list**. This API gets the list of station groups and its data and is stored in the state in **station\_group**.
* API for station group list: **/stations/list**. This API gets the list of station groups and its data and is stored in the state in **station\_list**.

1. Category / subcategory selection

* The data will be stored in **category\_selected** and **sub\_category\_selected** in state.
* The default value selected is the category and subcategory in the URL **<Project URL>/settings/station-groups/?cat=[2]&subcat=[].**
* If no category or subcategory is given in the URL, then the default value selected is Pump Station under Flood.
* The filter is a multi-select type, i.e. multiple categories and subcategories can be selected.
* On clicking a category, that category is selected along with all the subcategories under it.
* If a subcategory is clicked, if it is already unchecked, then it will be checked, and if already checked, it will uncheck.
* Every time the subcategory or category selected is changed, it is reflected in the URL and accordingly the stations in the table are displayed.
* The **station\_group** is filtered according to the filters applied and stored in **filtered\_groups.**

1. Search

* The result after applying filters can be searched for by station group name or station name.
* The searched text is stored in the key **table\_search** in the state.
* The list after the search from **filtered\_groups** is stored in **searched\_groups.**
* If station name is searched, the groups under which that station is present is displayed.
* The searched key is reflected in the URL and the result of the search is displayed in the table.
* URL: **<Project URL>settings/station-groups/?cat=[2]&subcat=[]&search=demo.**

1. Add new Station group
   * On clicking the Add new station group button, the URL is updated and a popup appears to insert information about the group such as group name, description, and stations.
   * The URL is: **<Project URL>settings/station-groups/add/?cat=[2]&subcat=[].**
   * Multiple stations can be selected for a group.
   * The states for the data are:
     + selected\_group\_name
     + selected\_group\_description
     + selected\_group\_stations
   * On saving the information, the data is posted to the API: **stations/groups/new**. And if the data is stored in the data base, we get a response with the group id of the newly created group. The keys to be sent in body of API:
     + name
     + description
     + station\_list
2. Station group listing in card form

* The card shows the Station group name, info button to view description, menu button (three dots), created by, creation time and stations under that group.
* On clicking the menu button (three dots), option menu opens to delete or edit the group.
* The delete button will be disabled for some groups which are not delete-able.
* On hover over the non-delete able button, a tooltip will open to show that the group is not delete-able.
* On clicking the delete button from the option, the URL will be updated and a confirmation popup will open to confirm the delete. On confirming, an API to delete the group will be called and on success response, the group will be deleted. On cancelling it, the group will not be deleted.
* A button loader will be shown while waiting for the response.
* URL to open the delete confirmation popup:

**<Project URL>settings/station-groups/:group\_id/delete/?cat=[2]&subcat=[]**

* On clicking the edit button from the menu, the URL will be updated and a popup will open to edit the station group information such as group name, description and stations under that group.
* URL to open edit modal:

**<Project URL>settings/station-groups/:group\_id/edit/?cat=[2]&subcat=[]**

* After modifying the information and clicking the save button, an API is called to post the data. API: **/stations/groups/:group\_id/edit**.
* The states for the edit data are:
  + selected\_group\_id
  + selected\_group\_name
  + selected\_group\_description
  + selected\_group\_stations
* The keys sent in API:
  + group\_id
  + name
  + description
  + station\_list