Beer Me

UML Class and method descriptions

**UI**

UI is the class that will handle all elements of the GUI, including format and outputting to the screen as well as user input.

Main() is the main loop of the program.

displayMenu() displays the entire GUI. It will call displayBeers() and possibly displayProfileAttributes() if the enum ActiveTab is set to “PROFILE”.

displayBeers() depends on the enum BeerSource and the enum ActiveTab. It checks what beers should be displayed on the highest level, calls the appropriate function from Discover or Profile to actually get the beers to be displayed to the user, and displays them on the GUI. For example, if the enum ActiveTab is set to “DISCOVER” and the enum BeerSource is set to “FILTER”, displayBeers() would call the function filter() in class Discover to get the beers to displayed. As another example, if the enum ActiveTab is set to “ONTAP”, then the enum BeerSource is not even looked at because the only list of beers to be displayed would be onTap, which is a field that will be retrieved from class Profile.

displayProfileAttributes() will display the attributes of beer that Beer Me has determined to be the best match for the user based on their likes and dislikes through the application.

**Discover**

Discover holds methods and fields that are relevant to the “Discover” tab in the GUI.

search() takes a String parameter which is inputted by the user in the search bar and returns an array of six beers that most closely match that String.

filter() takes an array of ints corresponding to the beer attributes that the user has selected from the filter screen. These attributes are shown on the GUI so the user has no confusion about what they are setting. The attributes are rated on a scale of 1-10. The method returns an array of six beers that most closely match the attributes of the ones provided by the user.

getRecommendations() looks at the attributes of the beers that the user has liked and disliked and returns an array of six beers that closely resemble the preferred attributes as well as are not similar to the disliked attributes. Note that the beers that are returned will not be in the user’s likes or dislikes.

**Profile**

Profile holds methods and fields that are relevant to the “On Tap” and “Profile” tab in the GUI.

handleFeedback() is called if the user adds a beer to their tap, or likes or dislikes a beer. It adds that beer to their dislikes if it is disliked, adds a beer to their likes if it is liked, or adds the beer to their tap if it is to be added to the tap.

getLikes() scans the “likes.txt” file and loads those beers into the “likes” ArrayList.

getDislikes() scans the “dislikes.txt” file and loads those beers into the “dislikes” ArrayList.

getOnTap() scans the “ontap.txt” file and loads those beers into the “onTap” ArrayList.

getPreferences() looks at the likes and dislikes ArrayLists and determines what the user’s general preferences are based on the data found.

**Beer**

Beer is a class holding immutable data. As such, the only method it holds is one beer’s name and its attributes.

Beer() initializes the Beer class for one beer.

Note that there will be as many instances of the Beer class as there are beers in out beer database (implemented as a .CSV).

**FileManager**

FileManager is the class that handles all file operations.

add() takes two String parameters, what to add and a filename, and adds what should be added to the filename provided.

remove() takes two String parameters, what to remove and a filename, and removes what should be removed from the filename provided.

**<<enum>> ActiveTab**

ActiveTab holds one of three values:

* DISCOVER if the user is currently viewing the “Discover” tab
* ONTAP if the user is currently viewing the “On Tap” tab
* PROFILE if the user is currently viewing the “Profile” tab

**<<enum>> BeerSource**

BeerSource holds one of three values:

* AUTOREC if the beers to be displayed are determined by an automatic recommendation based on the user’s likes and dislikes
* FILTER if the beers to be displayed are determined by the user’s input to the attribute filter
* SEARCH if the beers to be displayed are determined by the user’s inputted String containing the name of a beer

**<<enum>> BeerType**

BeerType holds the type of beer (IPA, pilsner, etc.) that the beer is.