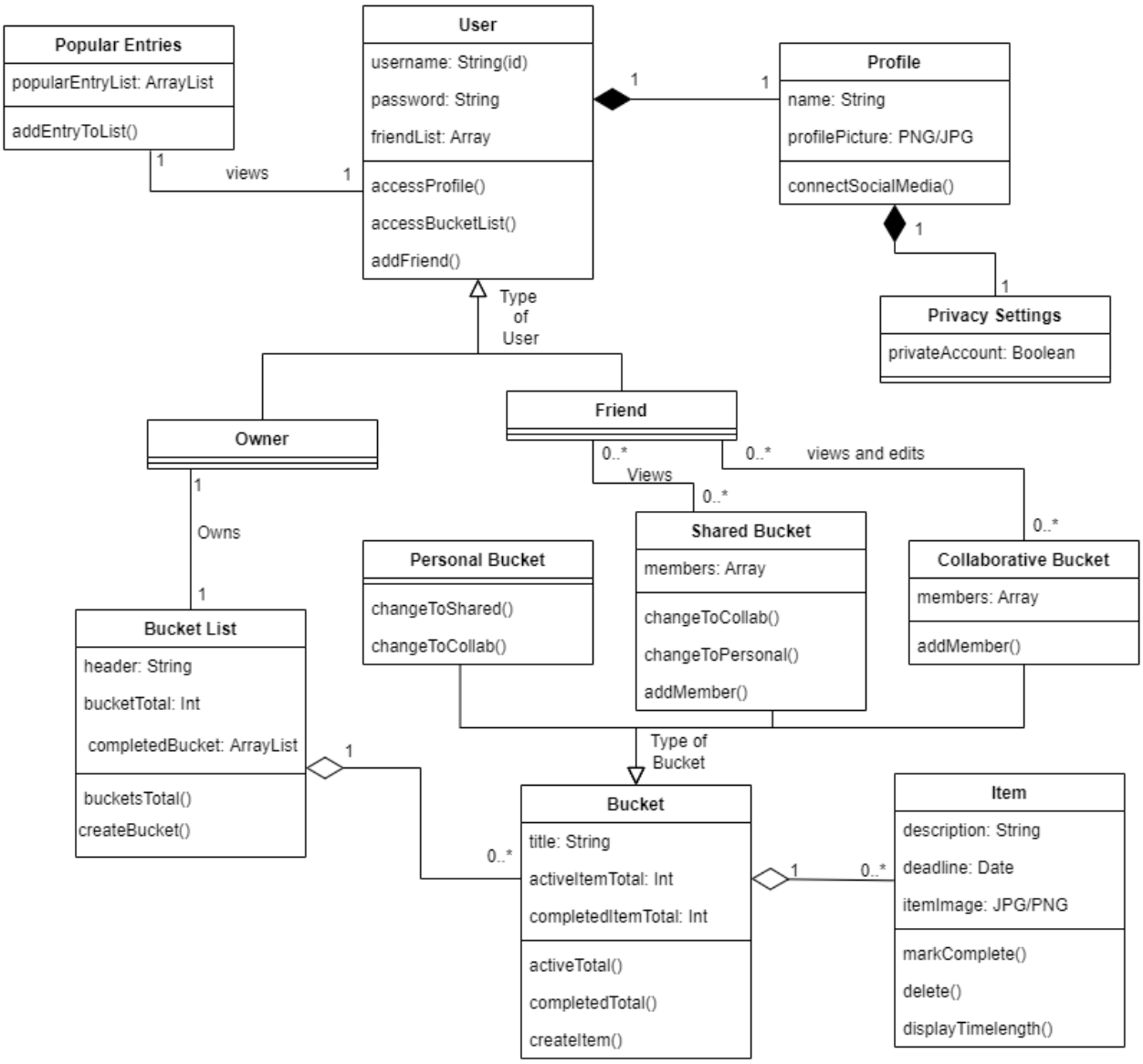
Domain Model Document



Description: The domain model above includes all of the user stories that we created for our application. The user class of the application has a username and password as well as a friend list that includes all of their friends. There is a “Profile” class that is owned by the “User” class, within this “Profile” class there is an attribute that holds the name of the user as well as a profile picture for the user. The “User” can also connect to their social media accounts in the “Profile” class. There is then a “Privacy Settings” class that is owned by the “Profile” class and in this class the “User” can choose to make their account private or public. In the “Privacy Settings” class there is an attribute “privateAccount” which is of the type boolean, if this is 1 then the account is private, if it is 0 then it is a public account. Going back to the “User”, they can access their profile, access their list of buckets, and they can add friends. There are two types of users

in this application, an owner and a friend. Even though these are two different types, a “User” can be each of the two types, one type at a time. We separated the “User” into two separate classes to show the capabilities of each user. When a user is in the “Owner” class they own a bucket list class where they can view how many buckets they have and they can also add buckets. The bucket list contains a header which is of the String type which just states “Bucket List”. It also contains an attribute named “bucketTotal” which is of the int type and it shows how many buckets an owner owns. There is also a bucket that contains all of the completed items from other buckets. The “Bucket” class is a part of the “Bucket List” class and in this class there is a title, and two other attributes which show the amount of active items within the bucket and the amount of completed items within the bucket. The methods for this class include seeing the amount of active items and completed items, and there is also a method to create a new item. The “Bucket” class is the superclass of three subclasses. These subclasses represent the type of bucket that each is, there is a subclass for a personal bucket, a collaborative bucket, and a shared bucket. A personal bucket is restricted to the owner, a collaborative bucket can be viewed and edited by the owner and shared friends, and lastly a shared bucket can be viewed and edited by the owner and only viewed by shared friends. There is then an “Item” class that is a part of the “Bucket” class. Within this class, items can be marked as complete, they can be deleted, and they can show the time they took to complete. This class contains a title, deadline, and image to represent the item. This takes care of the “Owner” class, when a “User” is in the “Friend” class they can do a few things. They can view buckets that have been shared with them and they can also view and edit collaborative buckets that have been shared to them. These types of buckets will only appear for a user in the “Friend” class if they have been shared to them. One last thing about our model is that there is a class for “Popular Entries”, this class shows the popular entries in our application and a user can choose to add these items to a specific bucket by using the method “addEntryToList()”. The “User” class is associated with this class so that they can view the popular entries.