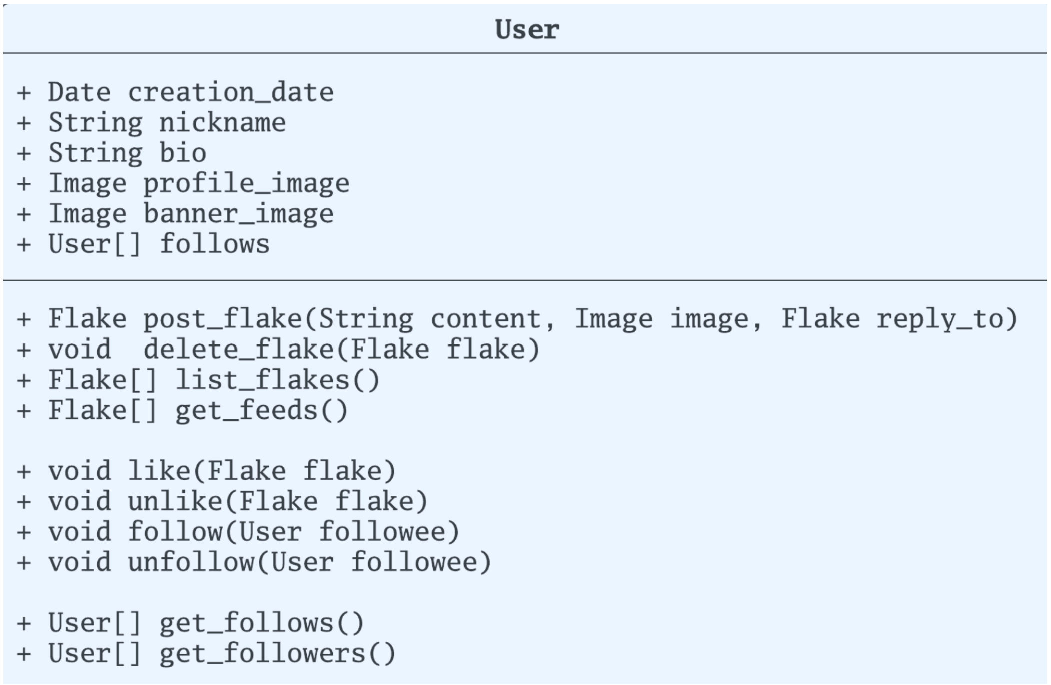

```
User

+ Date creation_date
+ String nickname
+ String bio
+ Image profile_image
+ Image banner_image
+ User[] follows

+ Flake post_flake(String content, Image image, Flake reply_to)
+ void  delete_flake(Flake flake)
+ Flake[] list_flakes()
+ Flake[] get_feeds()

+ void like(Flake flake)
+ void unlike(Flake flake)
+ void follow(User followee)
+ void unfollow(User followee)

+ User[] get_follows()
+ User[] get_followers()
```

Overview: The retweet functionality takes a flake from one user and reposts it as a flake again under the new user. We would need to create a Retweet method that returns a flake and takes in a Flake object and returns a new Flake.

Implementation: Within the User class, we would create a new method that returns a Flake and takes in a Flake object. We would then create a new Flake with the same information as the argument, and post the Flake under the user who is retweeting. But, we would also get the original poster information from the Flake argument and have that displayed in the retweet. You could then access the original Flake from the retweet somehow. This could be possible by creating a new Flake instance variable in the Flake class, which could allow Flakes to have parents, thus allowing you to see the original posts.