Basic UML Diagram for Social Networking Site

A basic UML diagram for a social networking site.

Here, there are five Classes - User, Post, Comments, Friendship, Homepage
Database to be used is - Graph DB, such as Neo4j

Explanation User Class -

1. A simple class that holds user Information where password field is not public. (Denoted by -)
2. Has setter / getter method for the user.
3. Also a user can delete his account, which again will delete all his post (Will use graph QL to write the query)
4. User class is dependent on Post Class, as a user create a post, and user will be param in create post method.
5. User class is dependent on Comment Class, as a user create a comment, and user will be param in create comment method.
6. User class has composite relationship with Friendship class as Friendship is consisted of Users.
7. Each User object will have a unique id, which is system generated

Explanation Post Class -

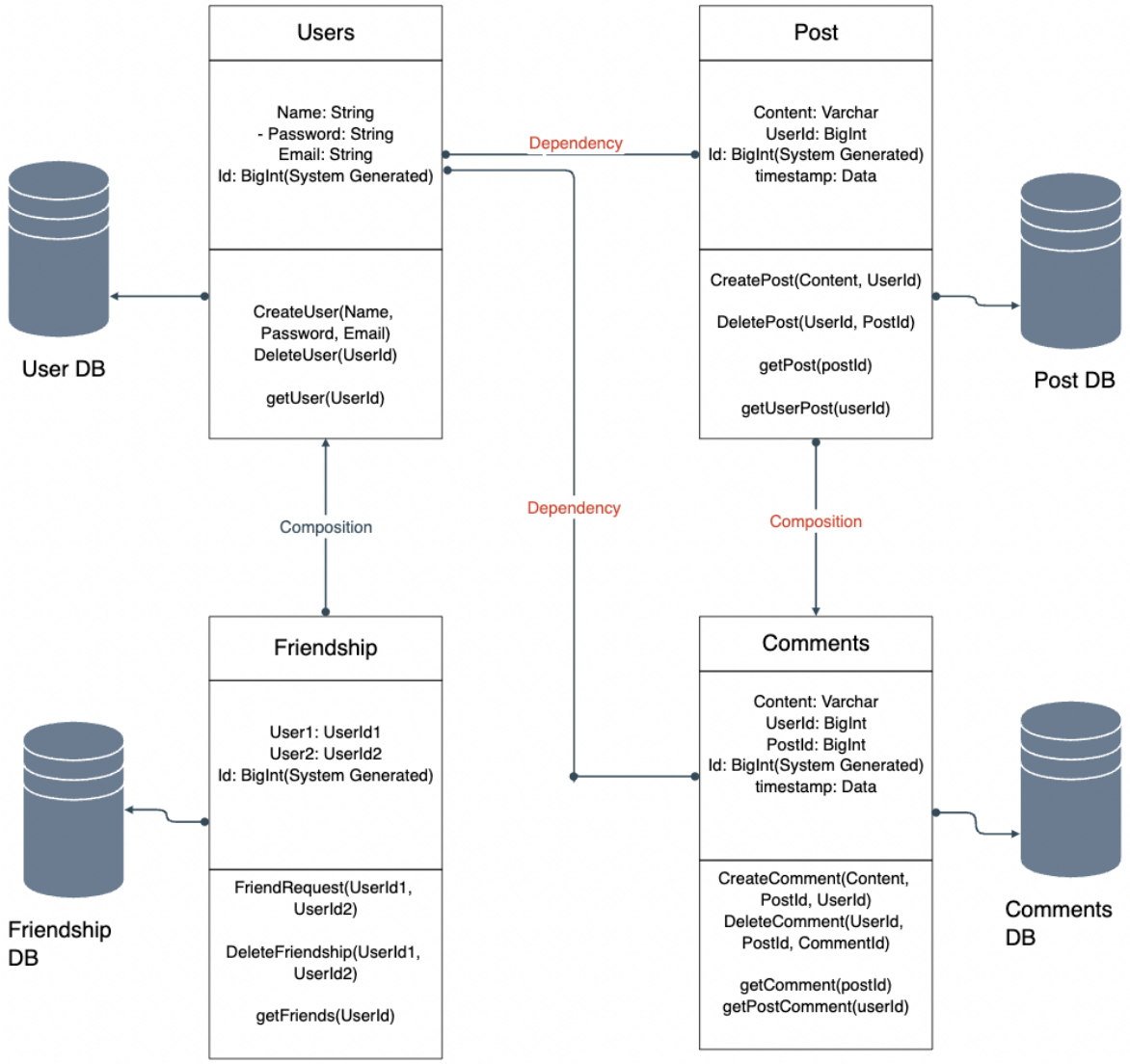
1. Hold information regarding the post and user who created the post.
2. Every post will consist of timestamp for homepage.
3. Has setter / getter method for the post.
4. Also a user can delete his post, which again will delete all comments in the post (Will use graph QL to write the query)
5. Post class has composite relationship with Comment class as Post is consisted of comments.
6. Each Post object will have a unique id, which is system generated
7. getUserPost will return a specific user post

Explanation Comment Class -

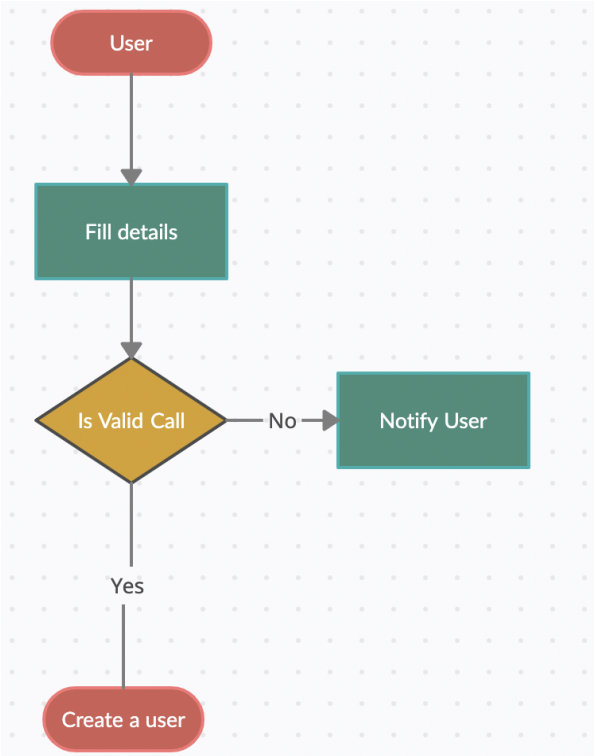
1. Hold information regarding the comment, post and user who commented on a given post.
2. Every comment will consist of timestamp for homepage.
3. Has setter / getter method for the comment.
4. Also a user can delete his comment.
5. Each Comment object will have a unique id, which is system generated.

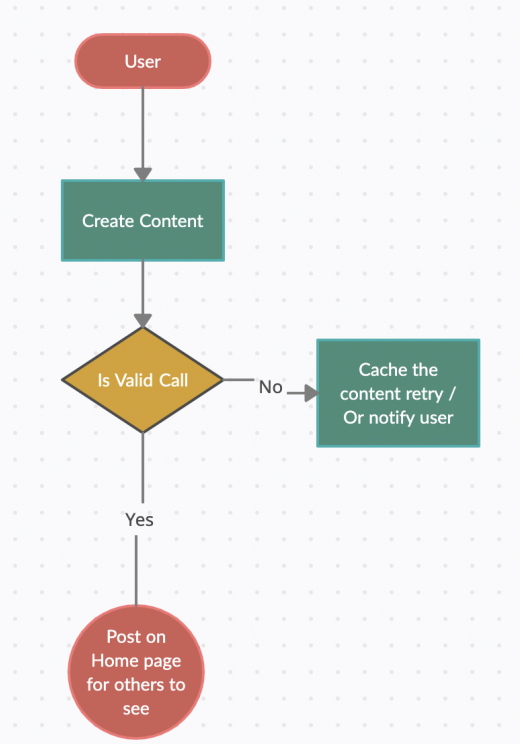
Explanation Friendship Class -

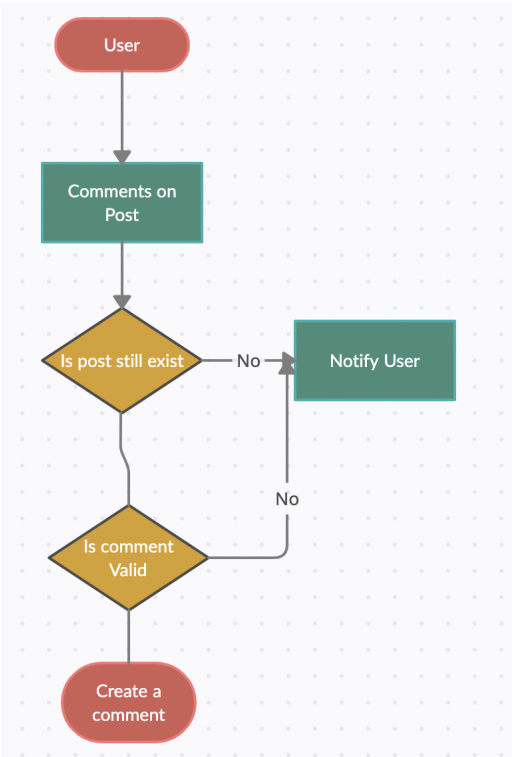
1. Hold information regarding the friendship between two users.
2. Has setter / getter method for the friendship.
3. Also a user can delete his friendship with other user.
4. Each Friendship object will have a unique id, which is system generated.

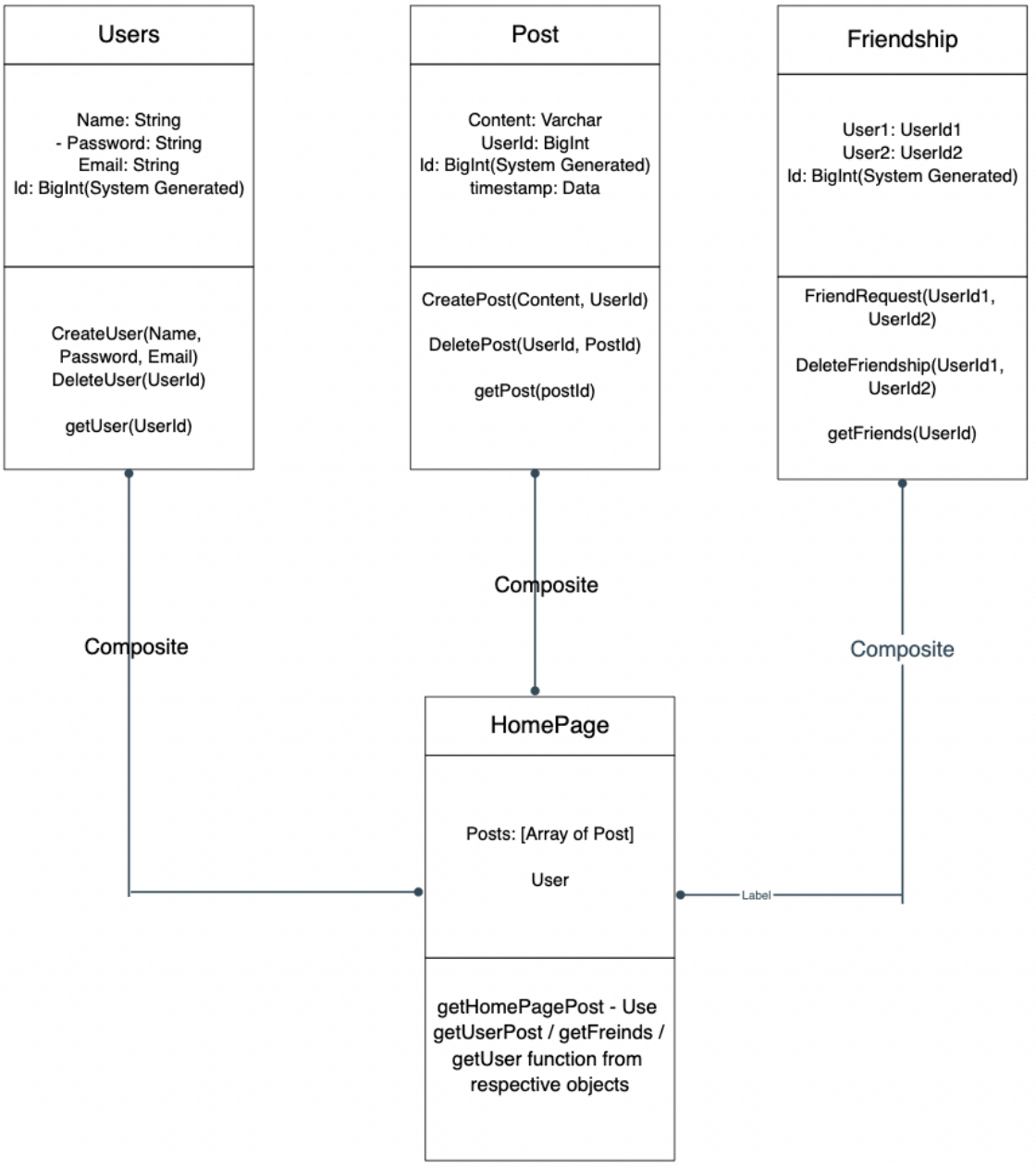


Few activity chart of user


User
Fill details
Is Valid Call
No
Notify User
Yes
Create a user


User
Create Content
Is Valid Call
No
Cache the content retry / Or notify user
Yes
Post on Home page for others to see


User
Comments on Post
Is post still exist
No
Notify User
No
Is comment Valid
Create a comment



1. Explanation Homepage Class -
    1. Hold information regarding the posts of user and the user itself.
    2. getHomePagePost method - will make use of getUserPost / getFriends / getUser method of respective abstract classes to create posts array for a specific user.
    3. Homepage class will form a composite relationship with rest of the three classes ie User, Post and Friendship

Improvements for a bit more complex design system

1. We can introduce permission block for functions within the Site, which then can be used for other scenarios, incase User1 blocks User2, User1 won't be able to call function friendRequest.
2. HomePage can work on observable pattern, where observers can registered for each user homepage (observable class), so to get instant updates
3. Search functionality which will again dependent on all the four base class ie User, Friendship, Post and Comment.