IFSC 3360: System Analysis and Design

Final Project Report

Twitter Social Media System

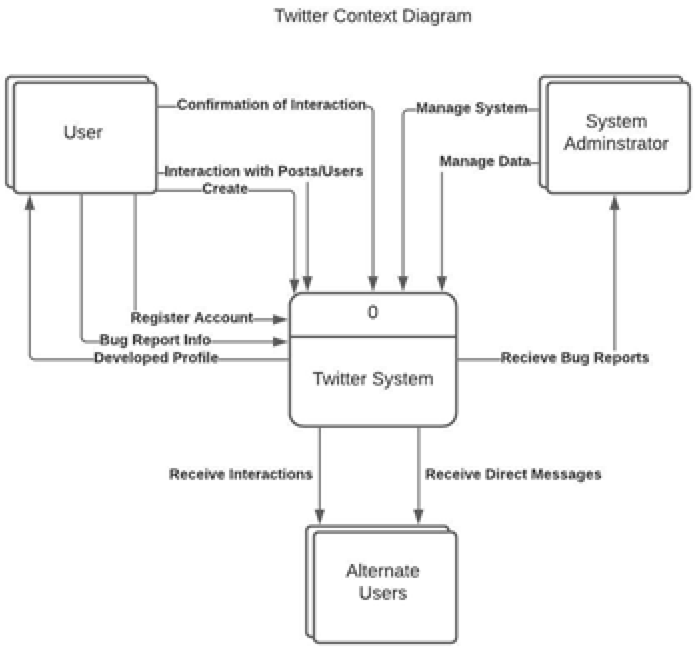
Alexander Sohn

1. Introduction *(500 words)*

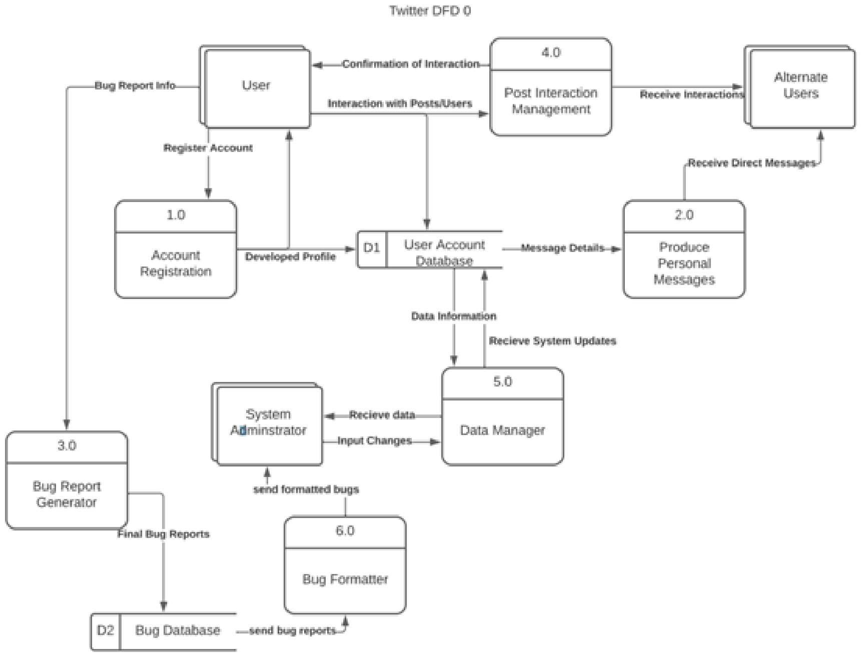
The goal of the system that I created was to provide a safe, secure, and simple social media system that anyone could use. This twitter system was designed so that people could have an easier take on the sometimes complex and confusing social media world out there.Some more objectives of the system are after a simple log in process, there will only be 8 to 10 key interactions when logged in. I also want to have a safer system where random people whose accounts you have never seen or interacted with are not allowed to message you. Personally, I have dealt with many scammers whether it be on instagram, twitter, or facebook. Another objective of my system is to have constant feedback and improvements. Any bug that is seen by a user will automatically be formatted, processed, stored in a database of total bugs, and also sent to the admin to be looked at. After an admin inspects the bugs and the percentages of how often they occur, they can push updates directly to the system. Since this system is smaller scaled and simpler, bugs will be easier to fix and faster to solve, while also hopefully having very little of them. My system improves an already existing system that is similar to twitter with the interaction aspect of it, but also has similarities to other systems such as instagram or facebook included. For example, all of these other social media systems have explore pages or pages where people can easily get lost in the enormous world of social media. However, in my system, I changed this to a simple system that is focused on seeing your friends' content and the people you follow in life, whether it be a sports team, celebrity, or even politicians. While I understand the advantages of having a page where people can find new people to follow or befriend, I think this take on social media could also work. This system is needed because of this need for simplicity. All of the current social media systems copy off each others expansions. An example of this can be seen with stories. Snapchat first introduced stories as a new type of way of interacting with friends. Then shortly after, instagram introduced stories as well on top of their already normal posts. Then, facebook also added stories after that, and now even twitter this past month added them. All of course with a different name for it. This cycle of copying the next big or already big social media systems leaves no room for new ideas, which is why my system is needed. Instead of complexifying something that is being copied and made bigger daily, why not try going the other way and simplifying it. Overall, I think this type of system could work for a majority of people, mostly older though. I think with everything going on in this world and the need for interaction or staring at a phone and getting lost in content for hours everyday being needed, something like this system could help counteract this problem. The only purpose is to see your world, and no one else's.

2. System Requirements Model

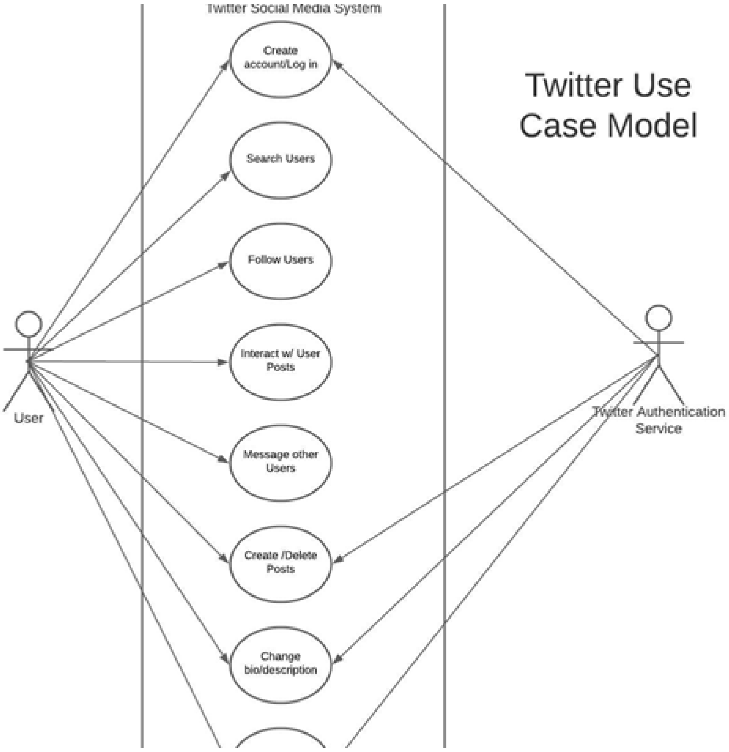
- Context Diagram
    - This context diagram is high level overview of the twitter system that shows the 3 entities and the main process of the Twitter System that represents the entire info system



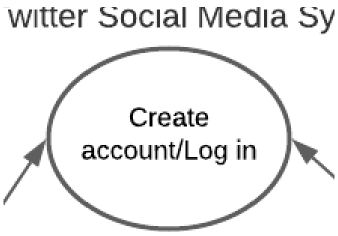
- DFD 0
  - This DFD 0 diagram takes process 0 from the context diagram and explodes it into different processes and databases that make up the whole information system. Everything in it basically runs through the user account database through one process or another.



- UML
  - This Twitter use case model shows two actors, the user and the twitter authentication service. It also shows 8 different use cases and how the 2 actors interact with each. After this, I provide the 8 use cases in this model.



  -
  - Create account/Log in Use Case

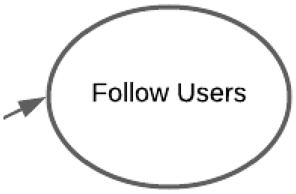
CREATE ACCOUNT/LOG IN Use Case	
Name:	Create Account/Log In
Actor:	User/Twitter Authentication Service
Description:	Describes process used to Create or Log In to an account
Successful Completion:	1. Person submits needed info to create account 2. Twitter Authentication Service checks if info is valid 3. If valid, an account is created and the

	person is now a user on twitter.
Alternative:	1. User has account and submits username and password 2. Twitter Authentication Service checks if given info matches 3. If valid, access is given to twitter
Precondition:	Person submits info to gain access to user status
Postcondition:	User is given full access to twitter

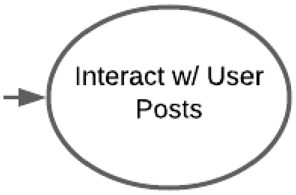
- Search User Use Case

SEARCH USERS Use Case	
Name:	Search Users
Actor:	User
Description:	Describes process of searching other twitter users
Successful Completion:	1. User types username in search bar 2. User is given multiple different accounts with similar usernames and exact username.
Alternative:	1. User types username in search bar 2. No accounts with that username match the intended search 3. Accounts with similar usernames are showed instead.
Precondition:	User has twitter account
Postcondition:	User finds the intended user searched for

- Follow User Use Case

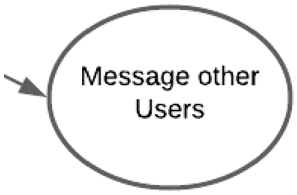
FOLLOW USERS Use Case	
Name:	Follow Users
Actor:	User
Description:	Describes process of User following Alternate Users on twitter in order to see/follow posts.
Successful Completion:	1. User is on Alternate Users page that they intend to follow 2. User clicks follow button and follows Alternate User 3. Alternate User is informed that User follows them and can follow them back or not
Alternative:	1. User is on Alternate Users page that they intend to follow 2. Alternate User is private and User requests follow 3. Alternate User decides to allow User follow or not(access to content or not)
Precondition:	User has twitter account and finds intended Alternate User
Postcondition:	User follows intended Alternate User and has access to feed.

- Interact w/ Users Posts Use Case

INTERACT W/ USER POSTS Use Case	
Name:	Interact with User Posts

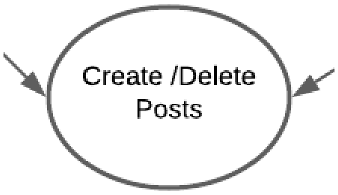
Actor:	User
Description:	Describes process of main User interacting (liking, commenting, retweeting) with Alternate Uses posts
Successful Completion:	1. User follows Alternate User and has access to their page 2. User goes to Alternates Users page and scrolls through posts 3. User Likes, Comments, and/or Retweets Alternate Users posts
Alternative:	1. User follows Alternate User and can see their posts 2. User scrolls through feed of followed users' posts. 3. User Likes, Comments, and/or Retweets Alternate Users posts that pop up on their timeline
Precondition:	User follows Alternate User
Postcondition:	User is able to share, like, and comment on an Alternate Users post who they follow

- Message other users Use Case

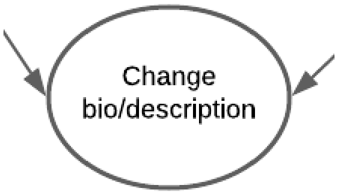
MESSAGE OTHER USERS Use Case	
Name:	Message Other Users
Actor:	User
Description:	Describes process of how a User messages an Alternate User
Successful Completion:	1. User follows Alternate User that they intend to personal message. 2. User clicks message button 3. User types message and clicks send 4. Alternate User receives message
Alternative:	1. User follows Alternate User that they

	intend to personal message 2. Alternate User has personal messages turned off 3. User can’t message Alternate User
Precondition:	User follows Alternate User and vice versa
Postcondition:	User messages Alternate User and Alternate User receives it

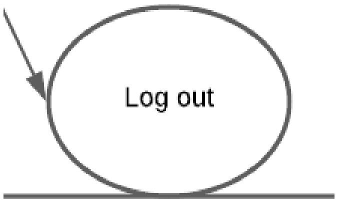
Create/Delete posts Use Case

CREATE/DELETE POSTS Use Case	
Name:	Create/Delete Posts
Actor:	User/Twitter Authentication Service
Description:	Describes process of how a User creates and deletes post from their feed
Successful Completion:	1. User has complete access to twitter system with their account 2. User clicks create post button 3. User types/adds media to create post area 4. User clicks post and the post is added to account feed for Alternate Users who follow to view
Alternative:	1. User has complete access to twitter system with their account 2. User goes to their account feed 3. User clicks delete post on the desired post 4. Confirms deletion and the post is removed from Alternate Users feeds
Precondition:	User is logged into twitter
Postcondition:	User creates and delete post from account and Alternate Users can view them

- ○ Change bio/description Use Case

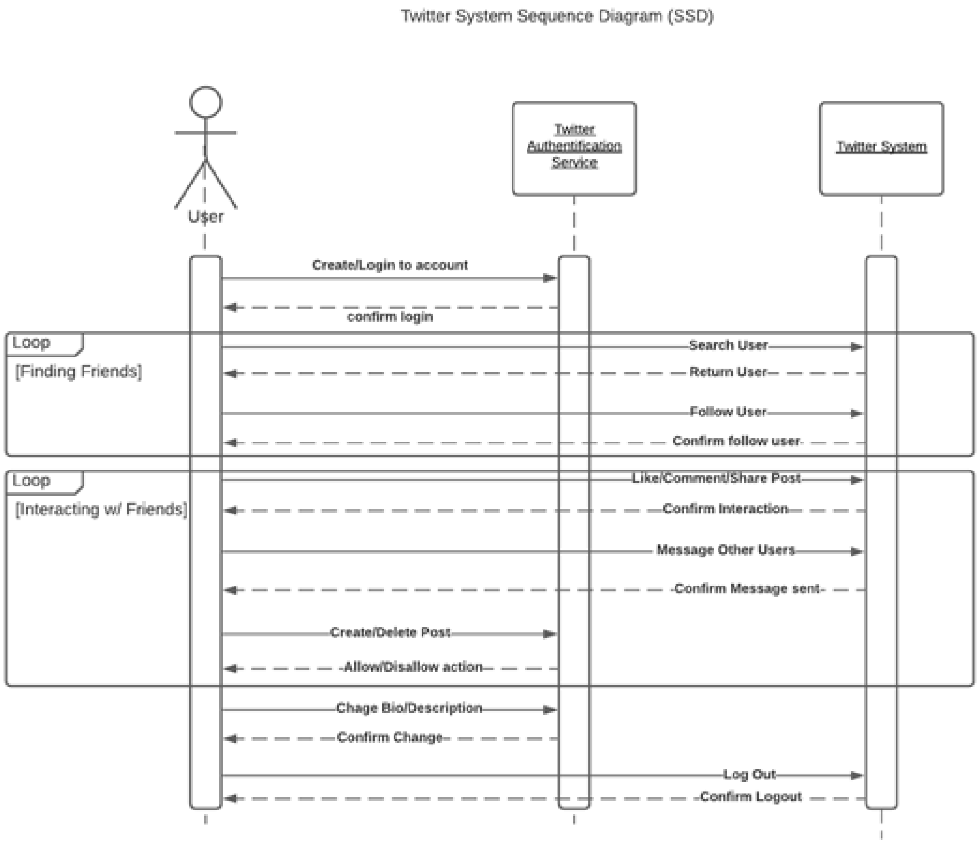
CHANGE BIO/DESCRIPTION Use Case	
Name:	Change Bio/Description
Actor:	User/Twitter Authentication Service
Description:	Describes process of how a User changes their bio/description
Successful Completion:	1. User is logged into their account on their accounts homepage 2. User clicks edit profile then types in desired bio/description 3. User clicks confirm and bio is changed. 4. Alternate Users can now see new bio
Alternative:	1. User is logged into their account on their accounts homepage 2. User clicks edit profile 3. User types bio that is not allowed 4. Twitter Authentication Service stops bio from being changed
Precondition:	User has an account on Twitter
Postcondition:	The Users account now has a different bio/description then before

- ○ Logout Use case

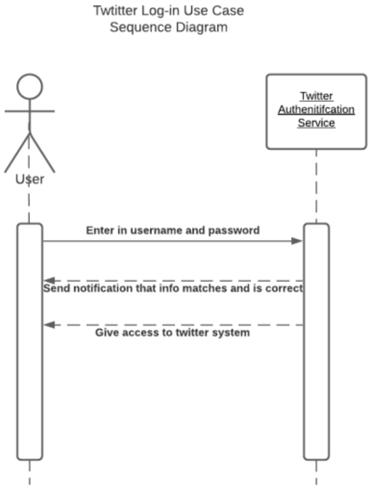
LOG OUT Use case	

Name:	Log Out
Actor:	User/Twitter Authentication Service
Description:	Describes Process of how a User logs out of the twitter system
Successful Completion:	1. User is logged into twitter system 2. User clicks logout button 3. User confirms that they want to logout 4. User is kicked out of twitter system and are back to login/create account screen
Alternative:	1. User is logged into twitter system 2. User clicks switch accounts button 3. User logs out of current account and is given ability to login to another
Precondition:	User is logged into to twitter account
Postcondition:	User is logged out of twitter account and doesn’t have access to twitter system

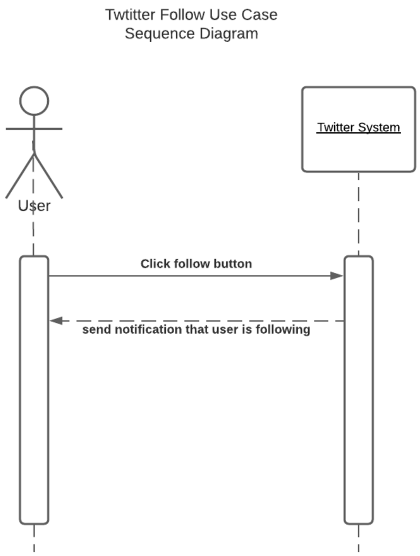
- SSD
    - This is the System Sequence Diagram of the twitter system, with 2 loops of finding friends and interacting with friends. Shows the user actor, and the 2 objects.



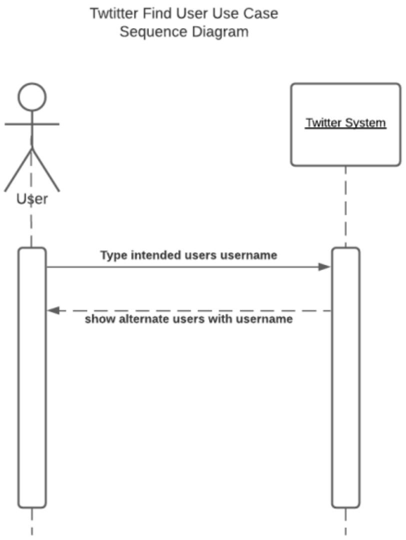
- Sequance Diagrams
  - #1 The login use case interacts with the user and authentication service to allow access into the system.


Twtitter Log-in Use Case
Sequence Diagram
User
Twitter Authenitifcation Service
Enter in username and password
Send notification that info matches and is correct
Give access to twitter system

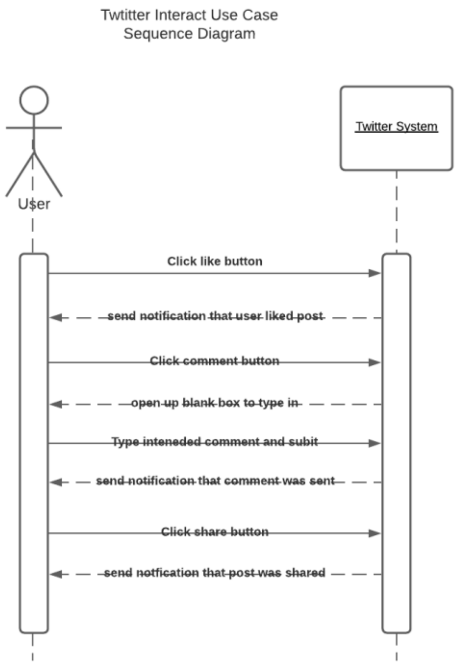
  - #2 The follow use interacts the user with the system to allow the user to follow an alternate user and gain access to their content/posts



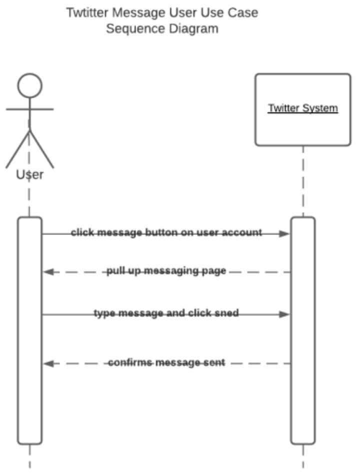
-
- #3 The find user use case allows for a user to access the twitter system and find alternate users whose posts they want to see or message.


Twtitter Find User Use Case
Sequence Diagram
Twitter System
User
Type intended users username
show alternate users with username

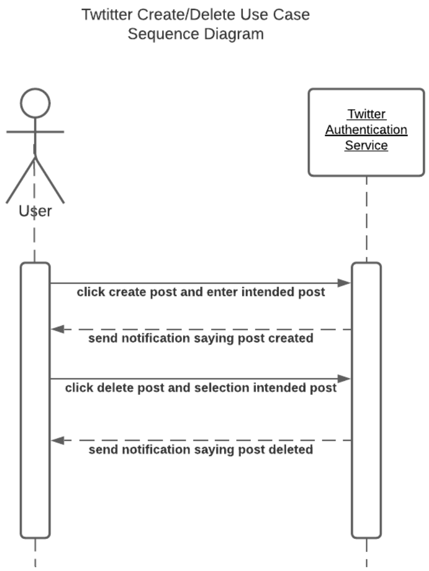
-
- #4 The interact use case is between the user and the system, showing how the user can interact with another users posts and provide feedback to them, through liking, commenting, and sharing; the system then notifies them of this action


Twtitter Interact Use Case
Sequence Diagram
Twitter System
User
Click like button
send notification that user liked post
Click comment button
open up blank box to type in
Type inteneded comment and subit
send notification that comment was sent
Click share button
send notfication that post was shared

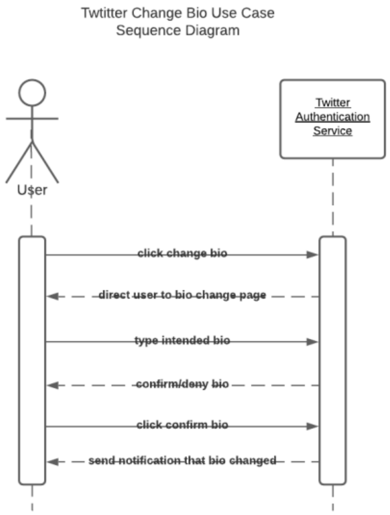
- ○
- ○ #5 the message interaction between the system and the user allows two users to interact with each other through a messaging system and notifies each other when a message has been sent.


Twtitter Message User Use Case
Sequence Diagram
Twitter System
User
click message button on user account
pull up messaging page
type message and click sned
confirms message sent

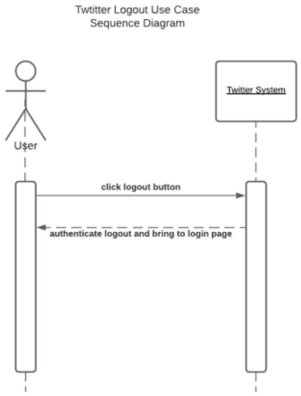
- ○
- ○ #6 The create/delete use case is an interaction between the user and authentication service, where a user submits the desired post and the authentication service allows or disallows it then notifies the user.


Twtitter Create/Delete Use Case
Sequence Diagram
User
Twitter Authentication Service
click create post and enter intended post
send notification saying post created
click delete post and selection intended post
send notification saying post deleted

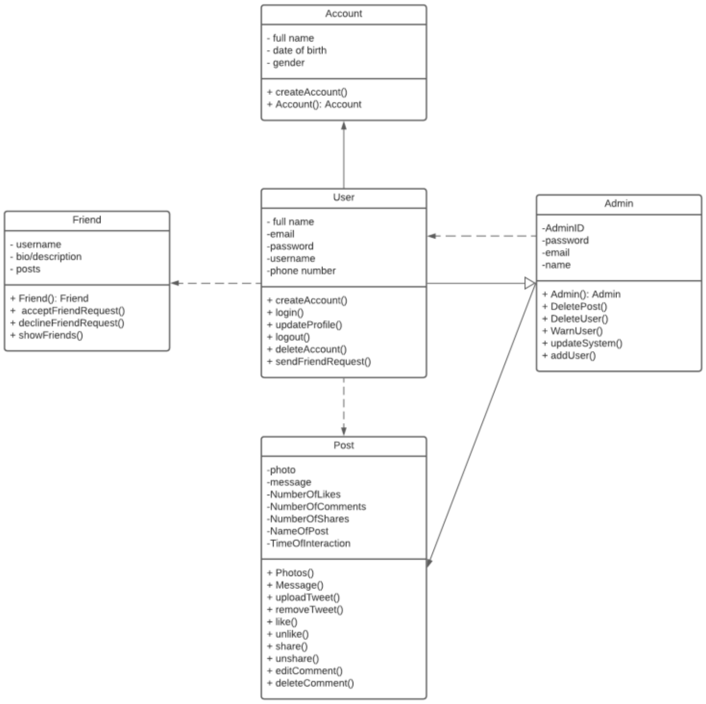
-
- #7 The change bio user case between a user and the authentication service allows the user to change what other users see below their profile, and the authentication service notifies whether or not its allowed or not.


Twtitter Change Bio Use Case
Sequence Diagram
User
Twitter Authentication Service
click change bio
direct user to bio change page
type intended bio
confirm/deny bio
click confirm bio
send notification that bio changed

-
- #8 The logout use case is a simple interaction between a user and the system, where the user submits a request to be logged out of the system, and the system allows it and disallows the user from seeing posts.



- Class Diagram
    - The class diagram has 5 classes, which are account, user, admin, friend, and post. The user has an association relationship to the account class because it has attributes of it. User also has a dependency relationship to friend because user sends messages to friend. Admin inherits user attributes, and admin sends messages to the user about their usage of the system. User also has a dependency relationship with post because it sends messages to post, for example what to do.



3. Lessons Learned

If I could do something differently a second go around, I probably would have created a system for something I enjoy more, like a video game or a sport. I also would have maybe implemented a new feature into a social media system, not sure what though