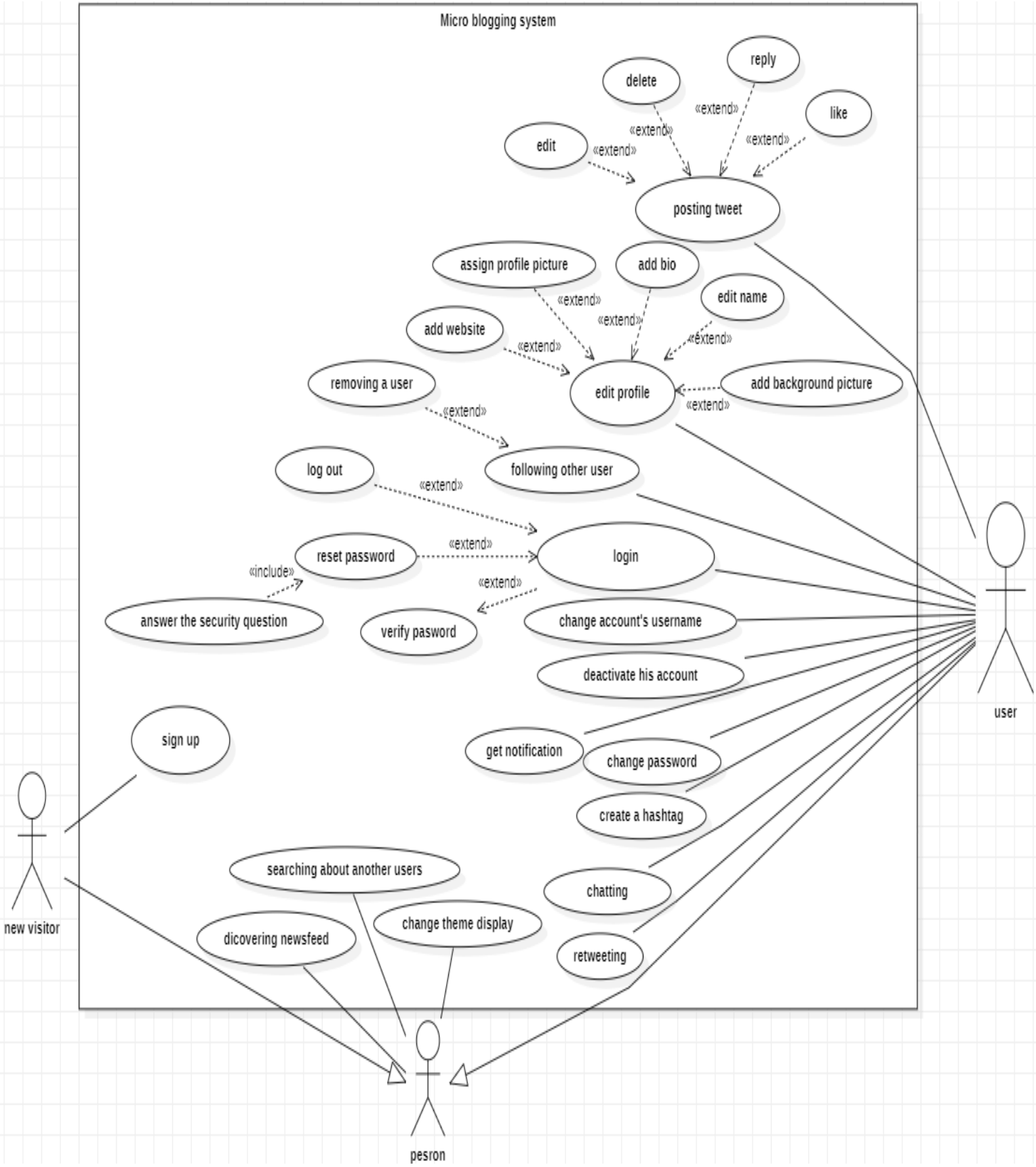
Use case:


Micro blogging system
reply
delete
like
edit
«extend»
posting tweet
assign profile picture
add bio
edit name
add website
removing a user
edit profile
add background picture
log out
following other user
reset password
login
«include»
answer the security question
verify pasword
change account's username
deactivate his account
user
sign up
get notification
change password
create a hashtag
searching about another users
chatting
dicovering newsfeed
change theme display
retweeting
new visitor
pesron

Descriptions:

Signup description

Identifier and name	Us1 Sign Up
Initiator	Visitor
Goal	Signup to the system for the first time to store his information.
Precondition	Has a valid email
Postcondition	Stored information about the visitor and become a user
Main success scenario	1-Visitor puts his information 2-the system valid the visitor information 3- sign up completed and the visitor became a user and can use all the services the system provides
Extension	1.1The visitor must have an email. If he doesn't have one, he must create an email. 2.1 if the visitor enters invalid information (weak password or repeated email address) 2.1.1 The system will ask to enter the information again

Login description

Identifier and name	Us2 Login
Initiator	User
Goal	Login into the system to use its services
Precondition	Signed up once before (having an account
Postcondition	Login in successfully
Main success scenario	1-user enter his email and password which he signed up with 2-The system makes sure it's The same stored in the database 3-login success
Extension	1.1 the user must have signed up once before. 2.1 if the email/password incorrect 2.1.1 The system will ask to enter the email/password again

Forgot password description

Identifier and name	Us3 Login
Initiator	User
Goal	Change user's password and login into the system
Precondition	Having an account on the website
Postcondition	Change user's password
Main success scenario	1-user enter his email and password which he signed up with and answer el security question 2-The system makes sure it's The same stored in the database 3-change password succsessfully
Extension	1.1 the user must have signed up once before. 2.1 if the email/password/security question incorrect 2.1.1 The system will ask to enter the email/password/security question again

Posting tweet description

Identifier and name	Us4 tweet
Initiator	User
Goal	Posting a tweet about the user's interest or opinion or using its options (like, reply)
Precondition	Login
Postcondition	System updates newsfeed with the new posted post
Main success scenario	1-The user can post a tweet. The user can see other tweets and 2-The user can delete unwanted tweets or edit them 3-The user can share a tweet to other social media sites.
Extension	1.1posting a tweet failed because the tweet exceeds the number of words limit. 1.1.1 The System will display an error message reminds the user about the limit

Interacting with tweets

Identifier and name	Us5 tweet
Initiator	User
Goal	User can interact with tweets are interested in
Precondition	Tweet is already exists
Postcondition	User interacts with tweets (reply, like, retweet,).
Main success scenario	1- user can reply on other tweets 2- user can like other tweets 3-user can retweet another tweet.

Edit profile description

Identifier and name	Us6 profile
Initiator	User
Goal	editing his profile and improve his identity by assigning himself a profile picture, add bio expresses about his interests, add a background picture, and updates his name
Precondition	An account without identity
Postcondition	Well known account to users
Main success scenario	1-The user assigns himself a new profile picture. 2-The user assigns himself a new background picture. 3-The user add bio to his account 4-The user edits his account name.
Extension	1.1adding bio failed because bio characters exceed the number of characters limit. 1.1.1 The System will display an error message reminds the user about the limit

Following/unfollowing other users description

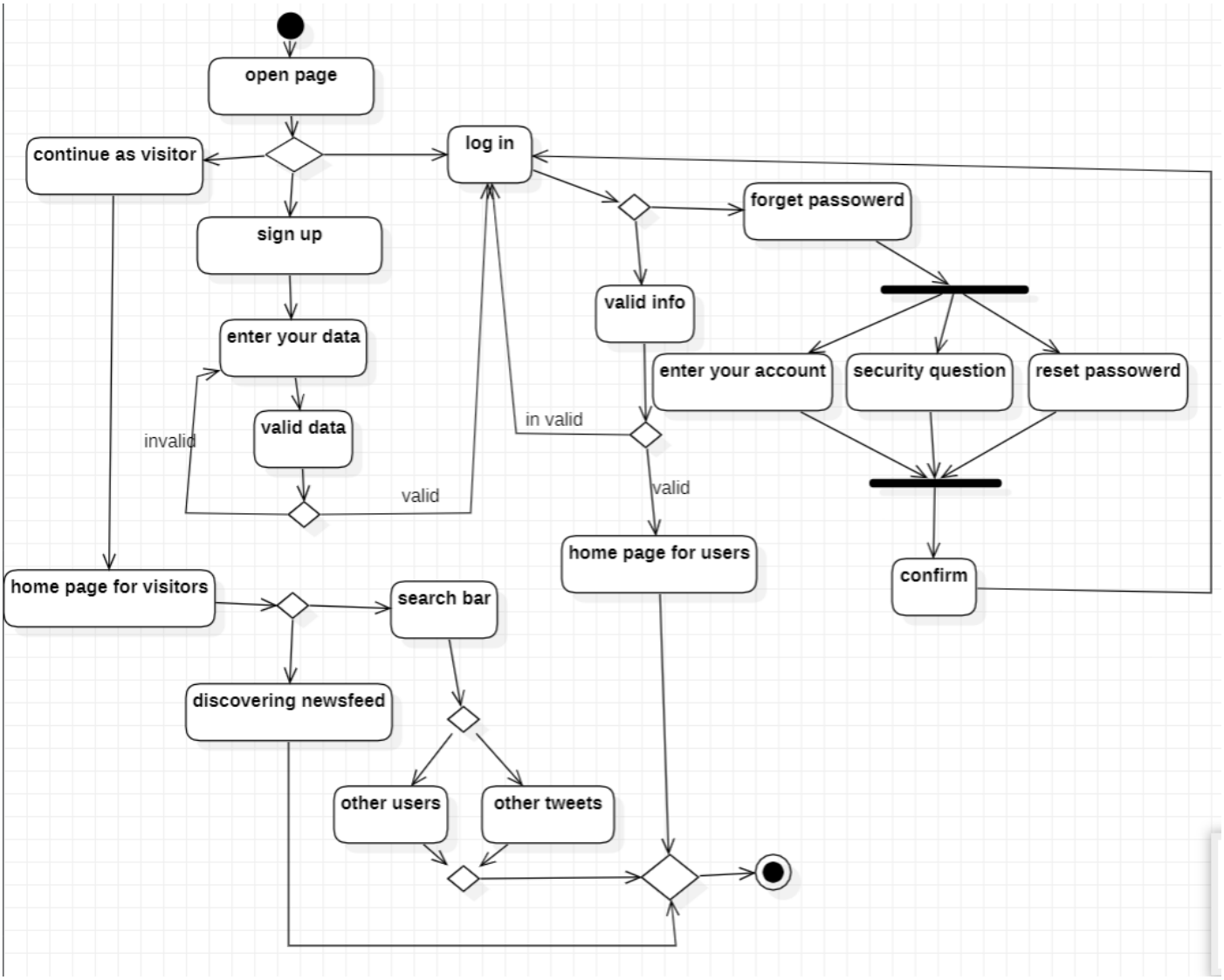
Identifier and name	Us7 following
Initiator	user
Goal	Follow another user to view posts that's shared by him in the newsfeed
Precondition	The user must be logged in the system.
Postcondition	Following counter increases one by one at every following process and decreases one by one at every unfollowing process
Main success scenario	1-user follows another user 2- user unfollow another user

Deactivate his account description

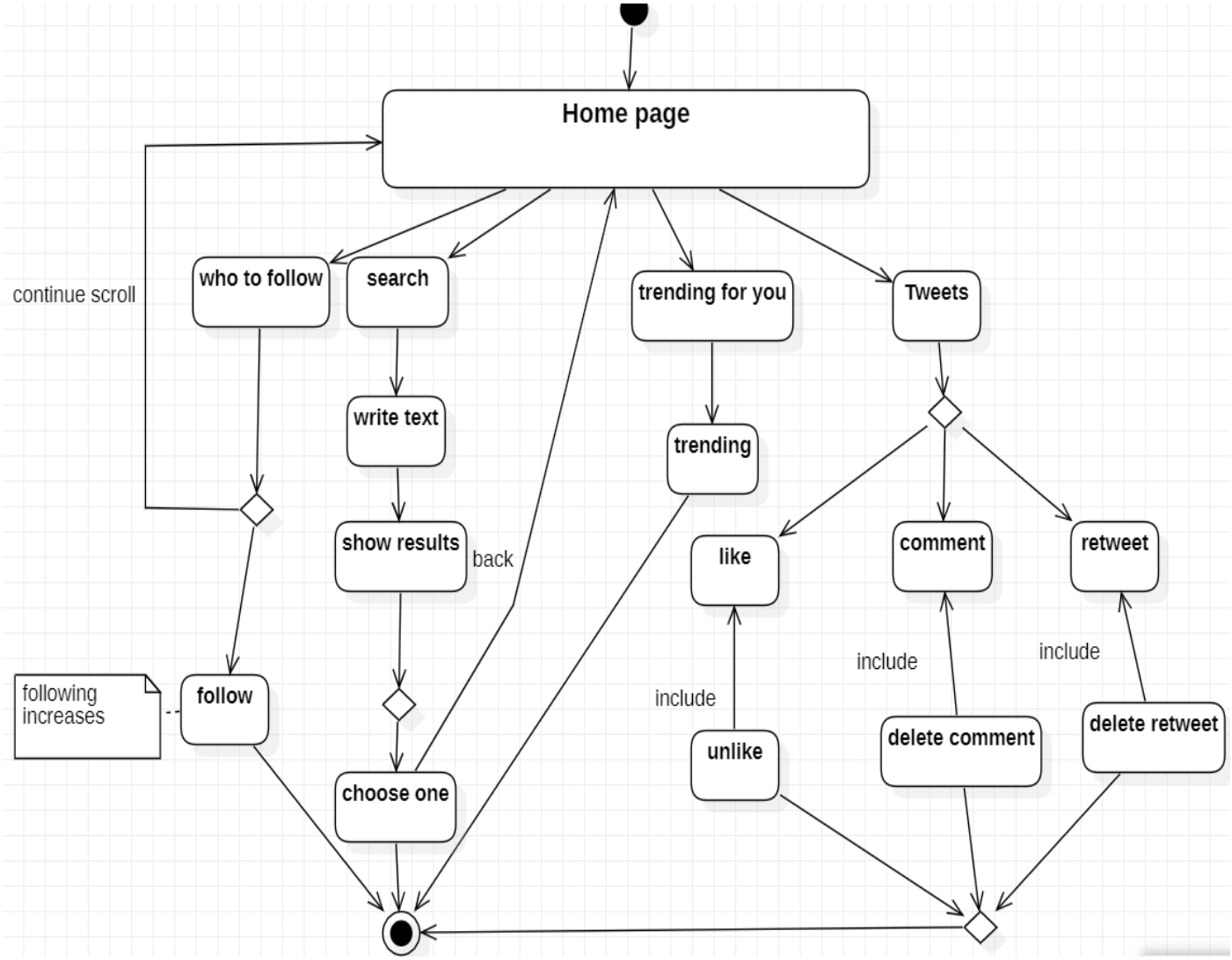
Identifier and name	Us8 deactivation
Initiator	user
Goal	Delete his account
Precondition	The user must be logged in the system.
Postcondition	User's data will be deleted from the data base
Main success scenario	user deactivates his account.

Activity diagram:

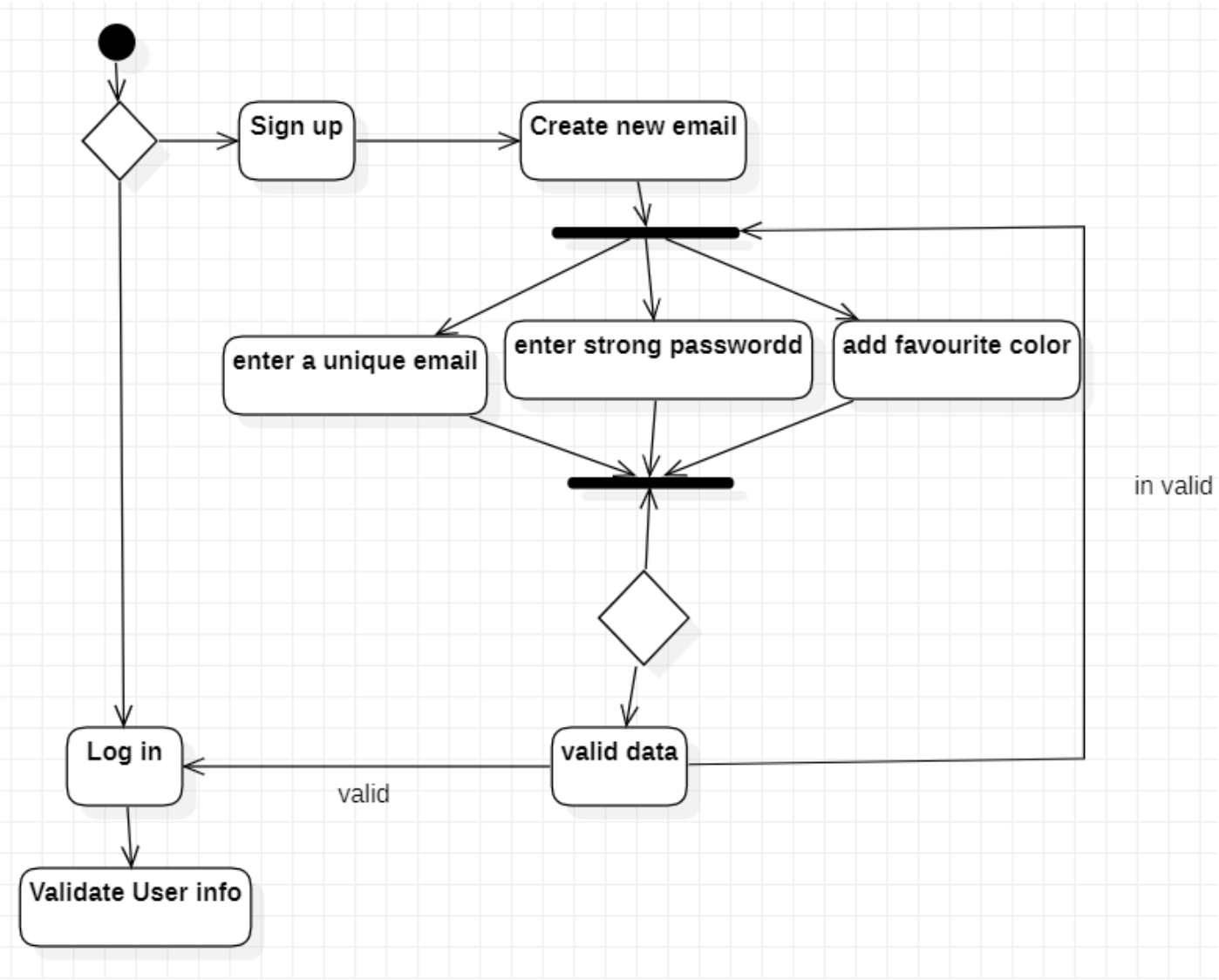
Sign in & sign up



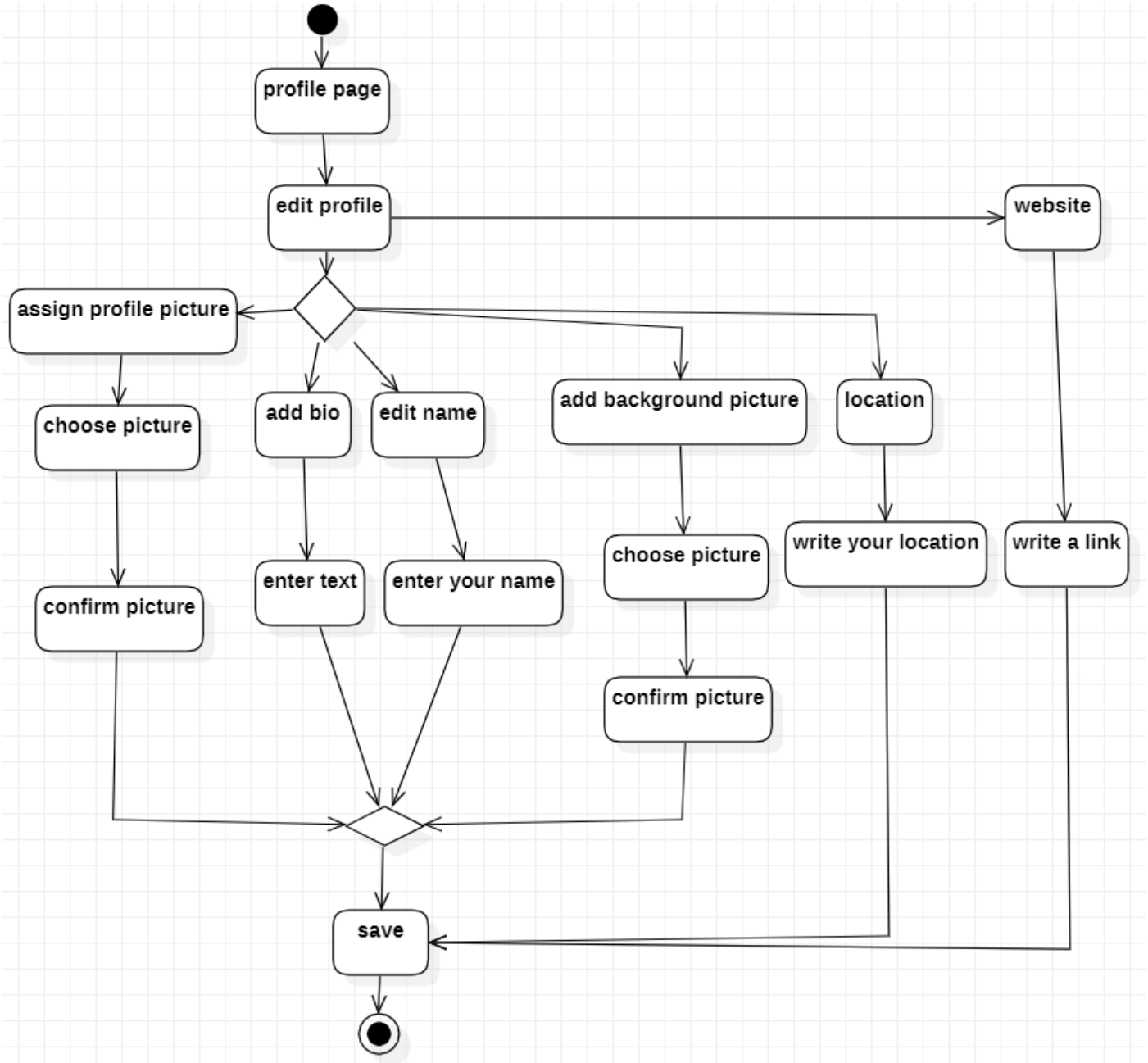
Homepage


Home page
who to follow
search
trending for you
Tweets
continue scroll
write text
trending
show results
back
like
comment
retweet
include
include
include
following increases
follow
unlike
delete comment
delete retweet
choose one

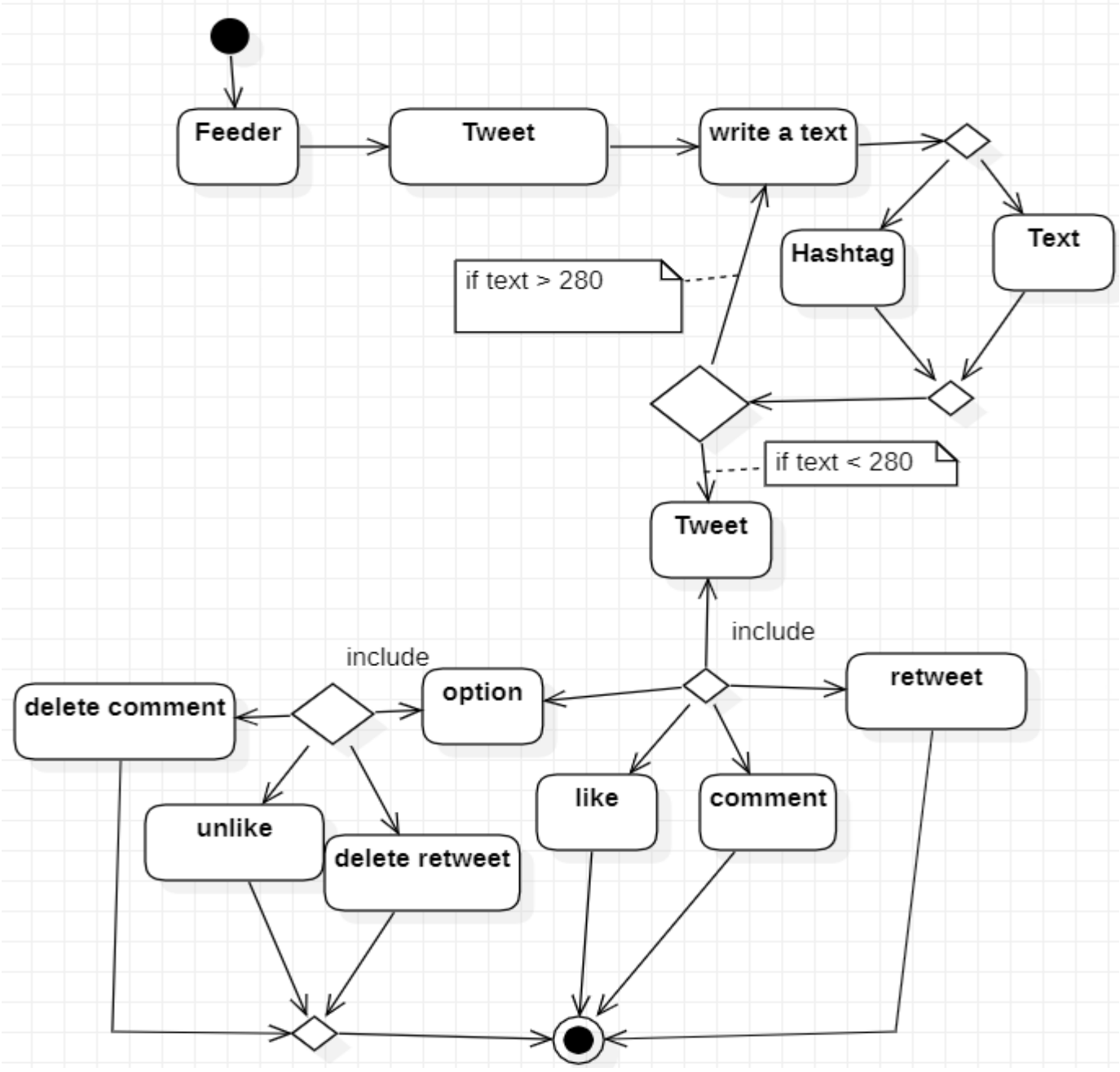
Login form


Sign up
Create new email
enter a unique email
enter strong passwordd
add favourite color
in valid
Log in
valid data
valid
Validate User info

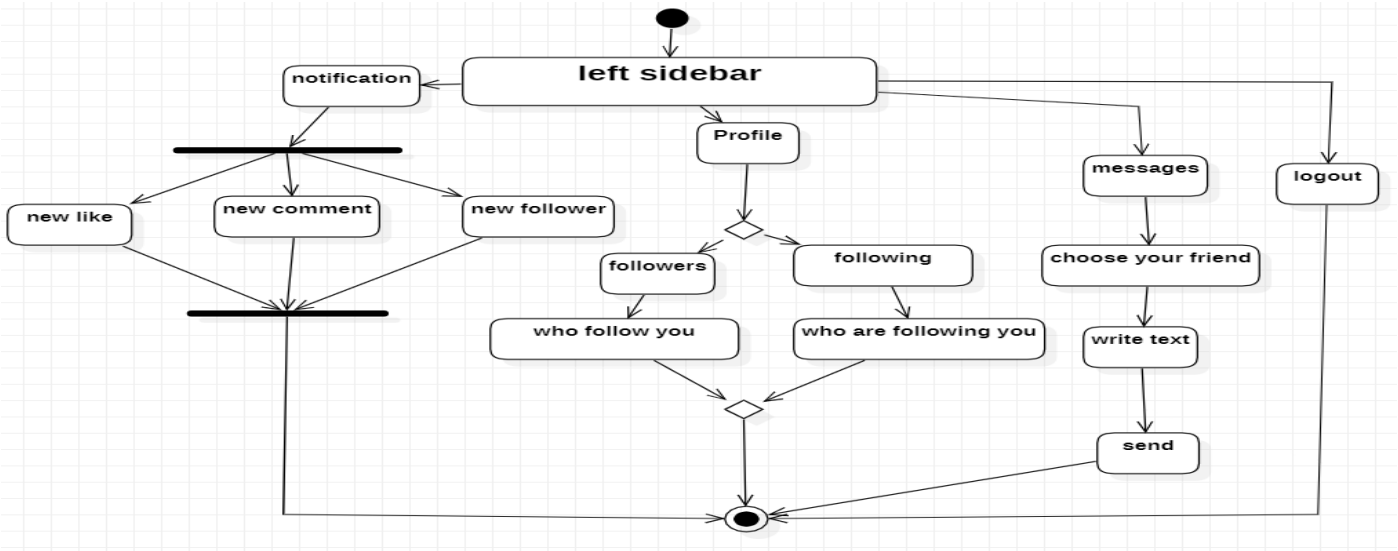
Edit profile



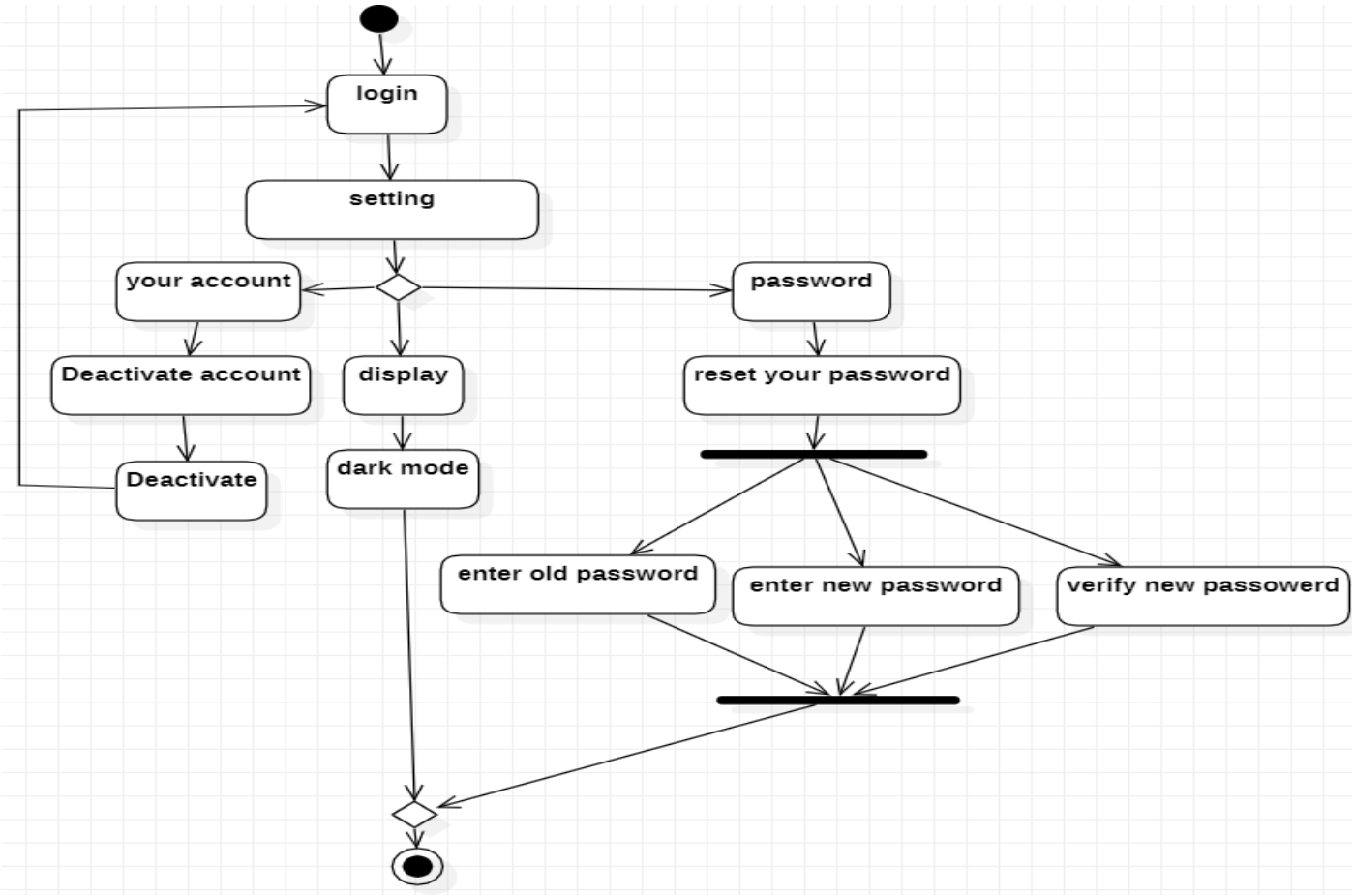
Posting a tweet



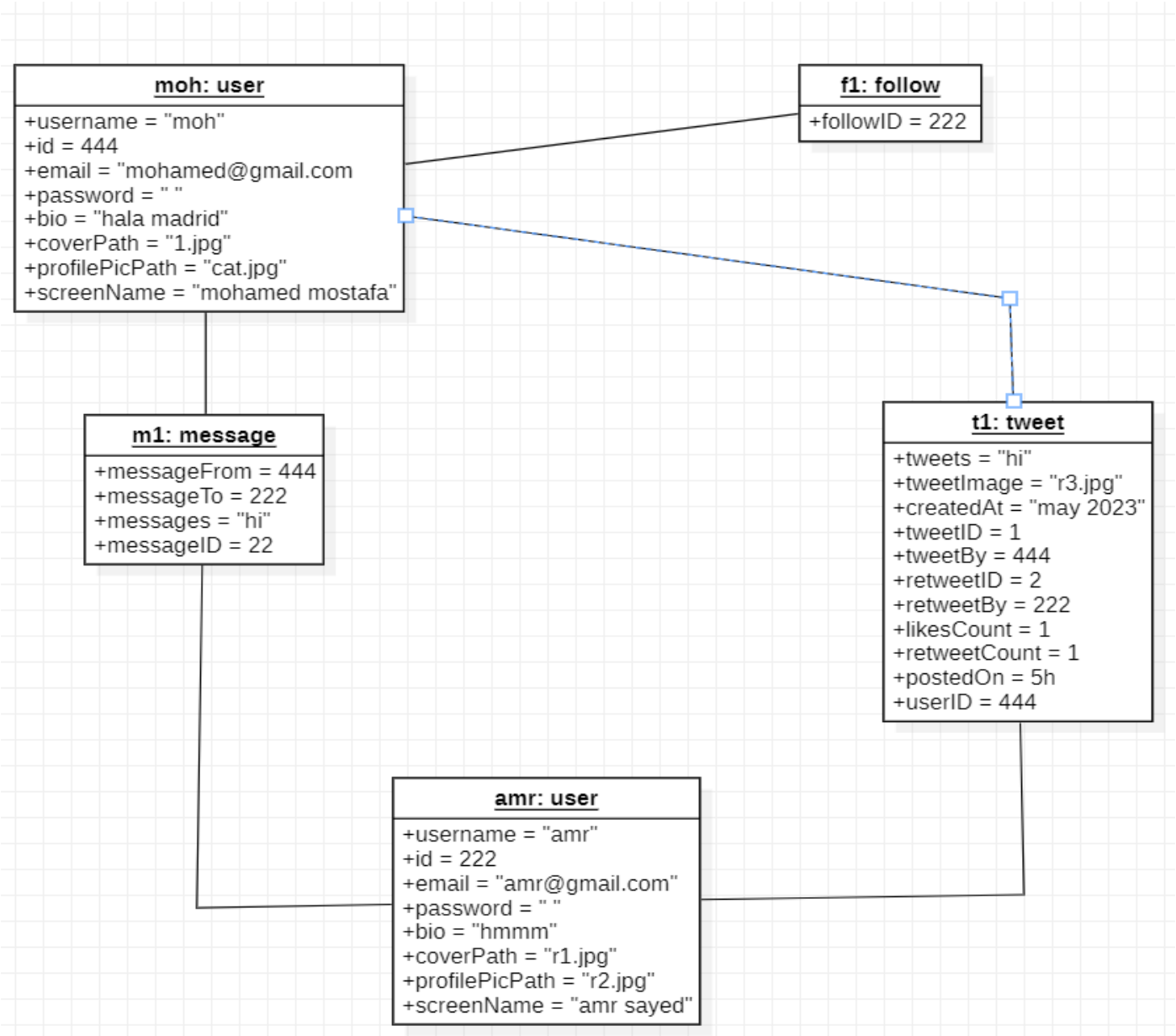
Sidebar



Settings

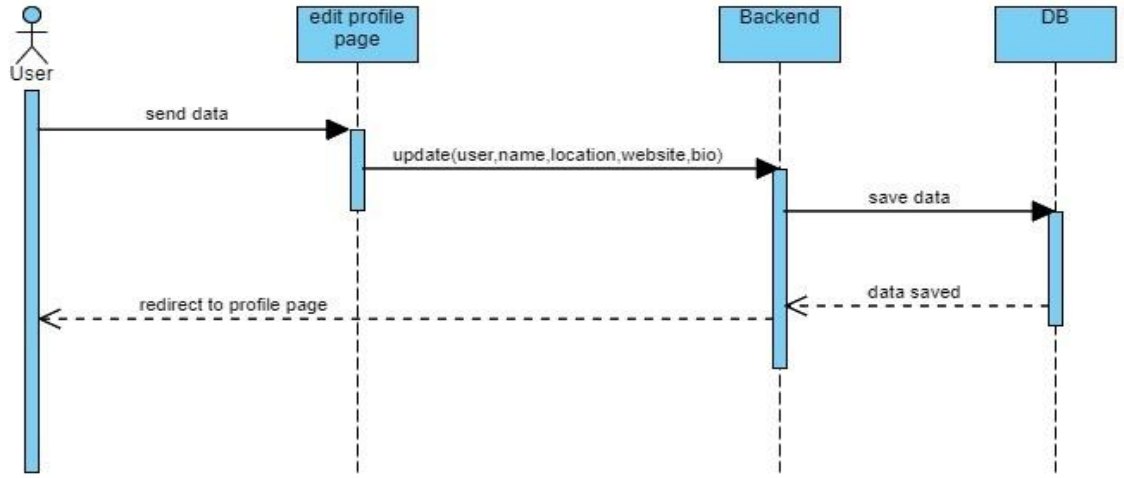


Object diagram:

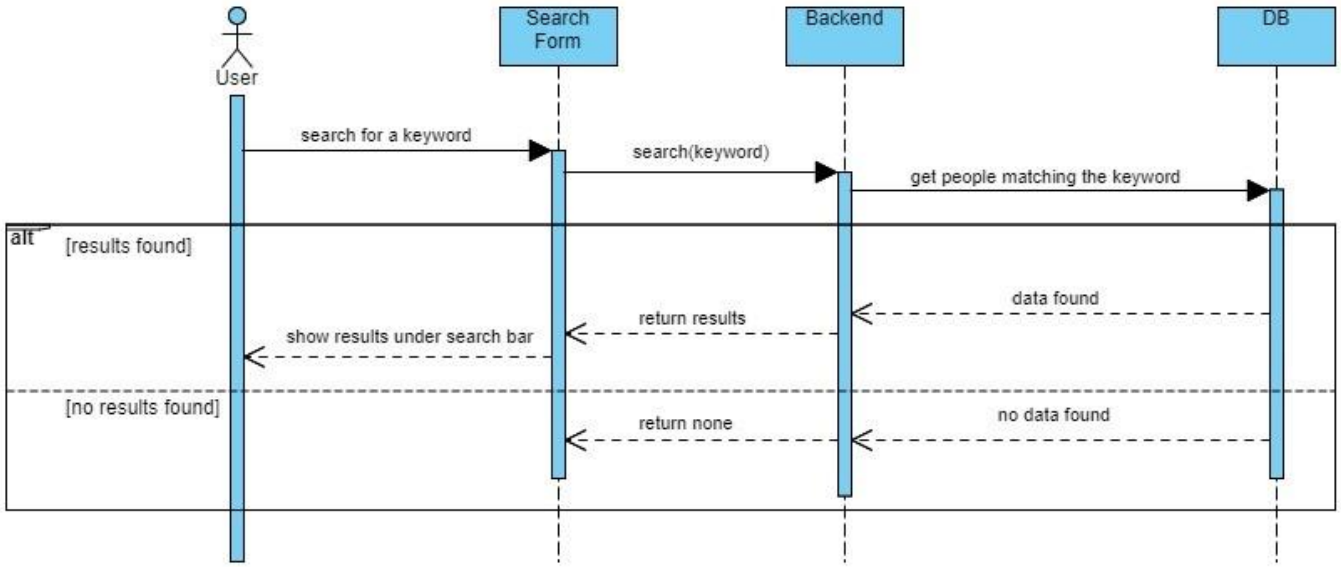


Sequence diagram:

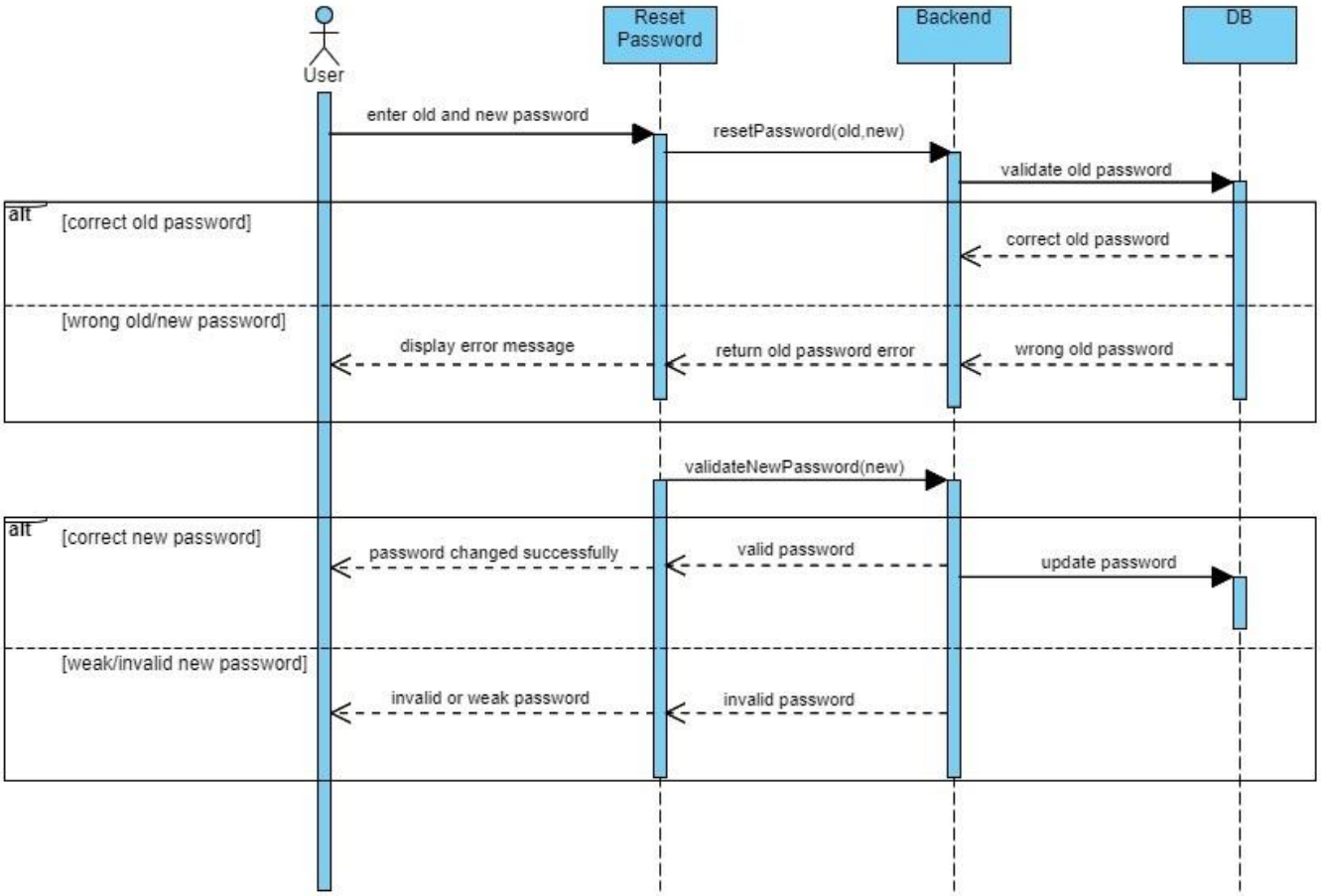
Edit profile



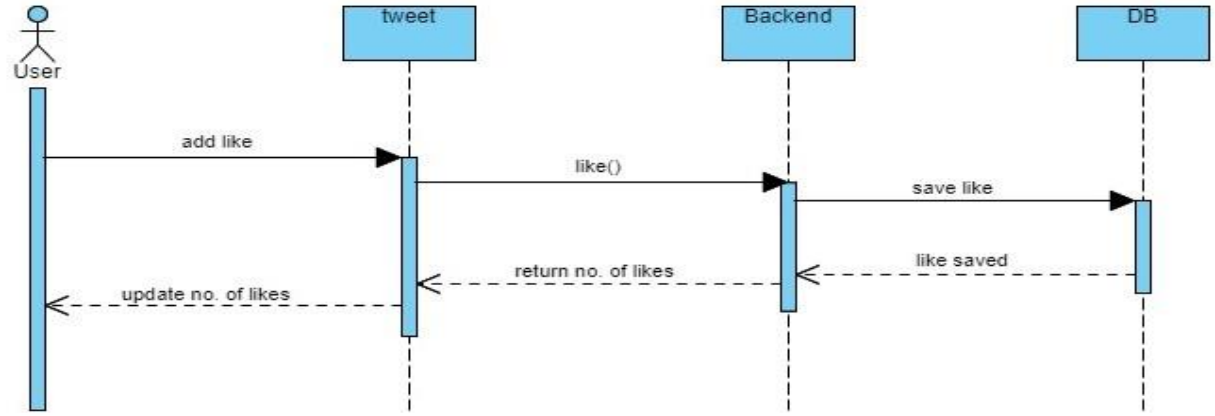
Search bar



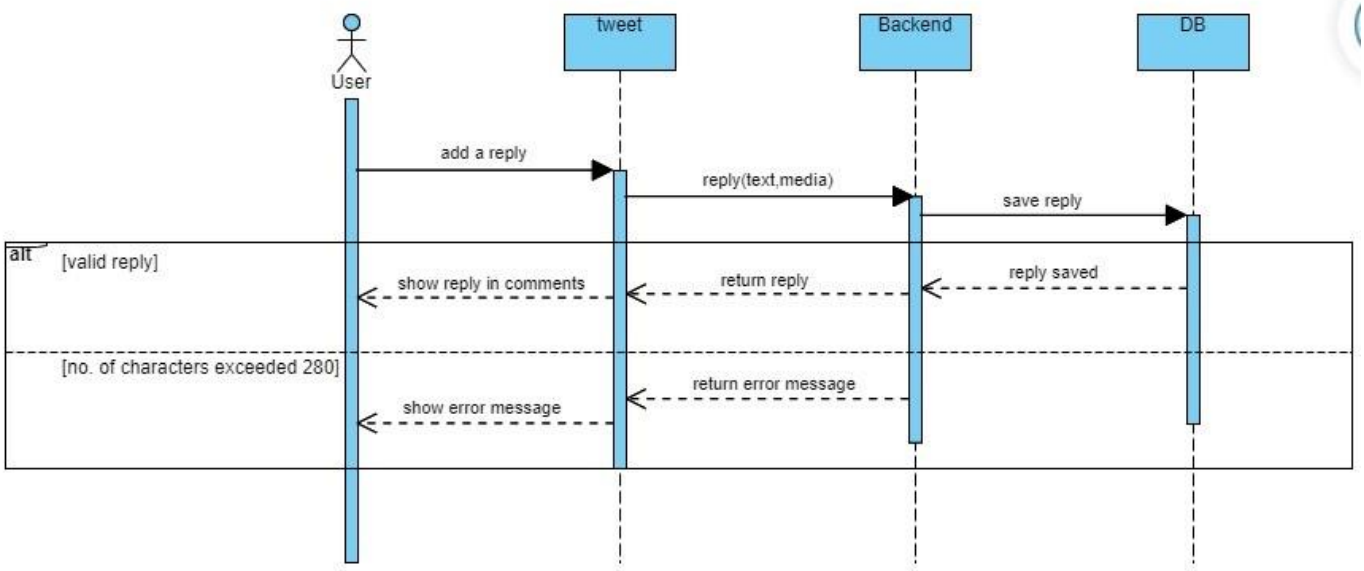
Reset password



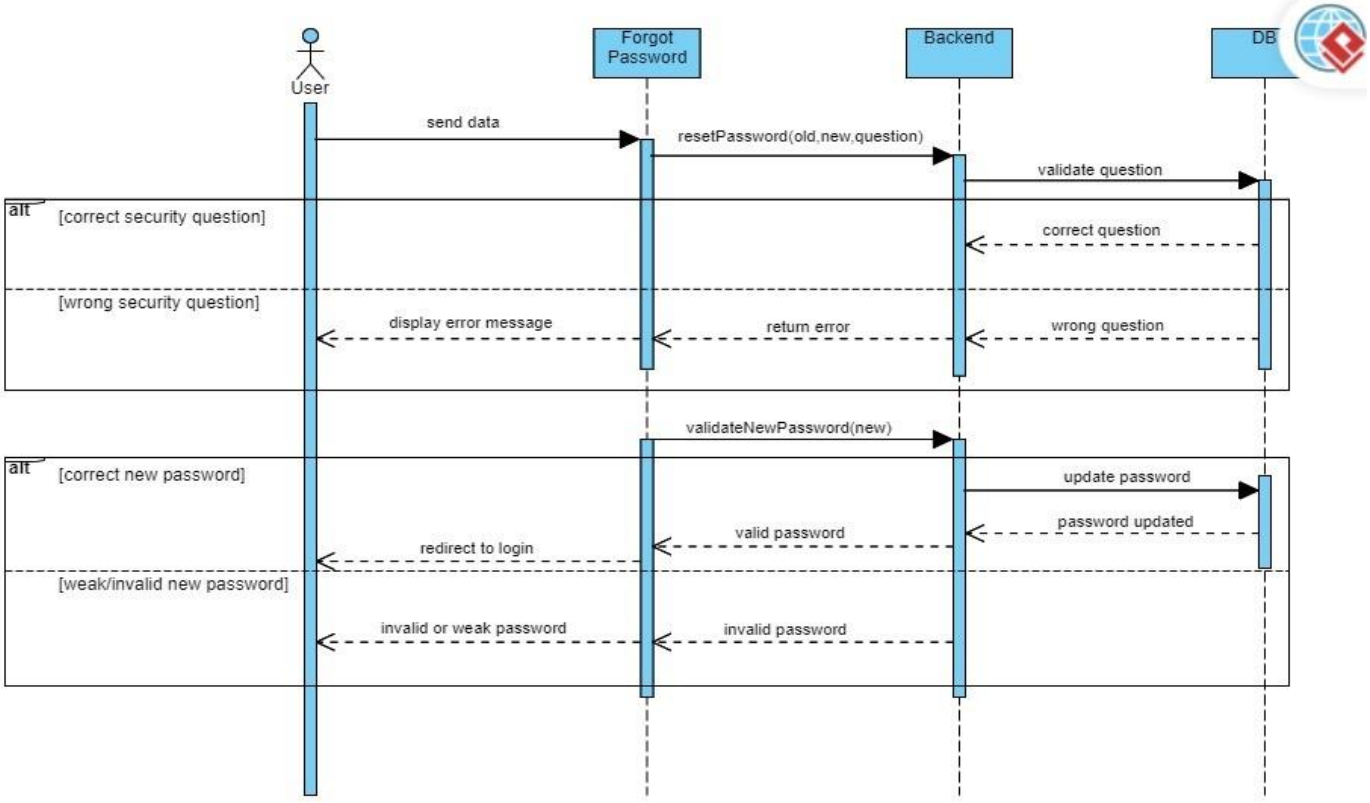
Adding a like



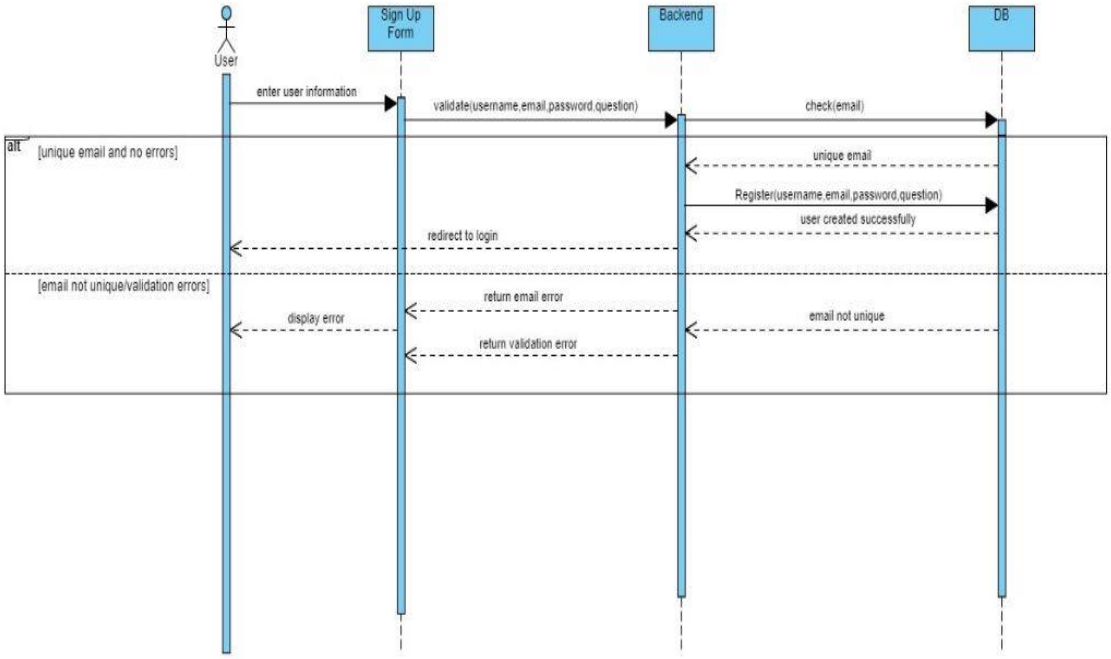
Adding a comment



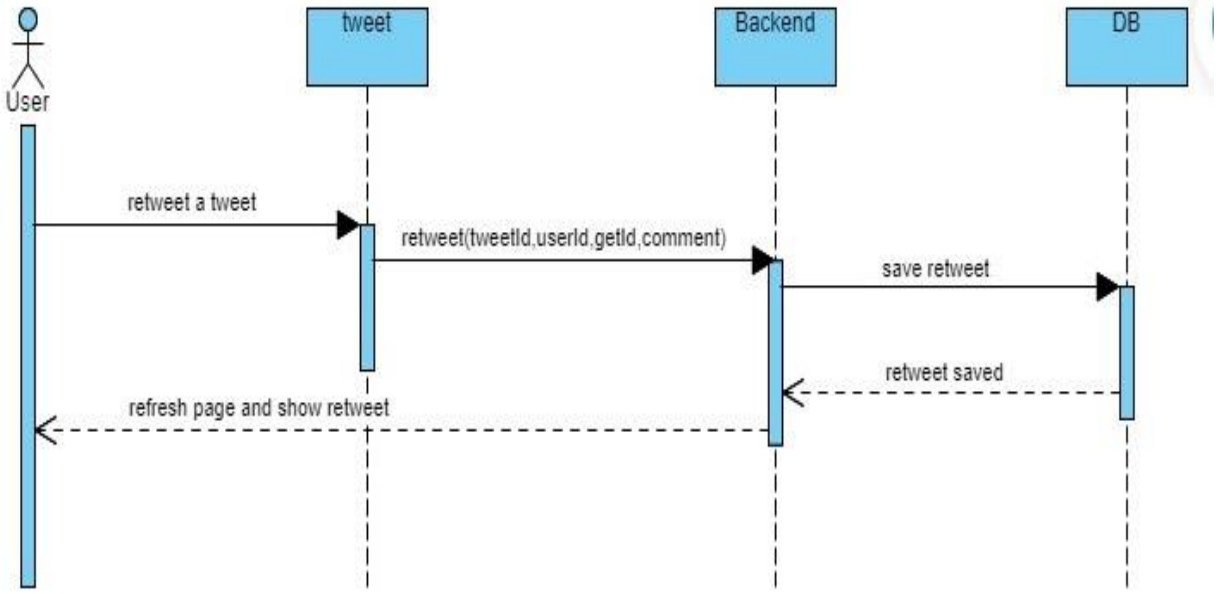
Forget password



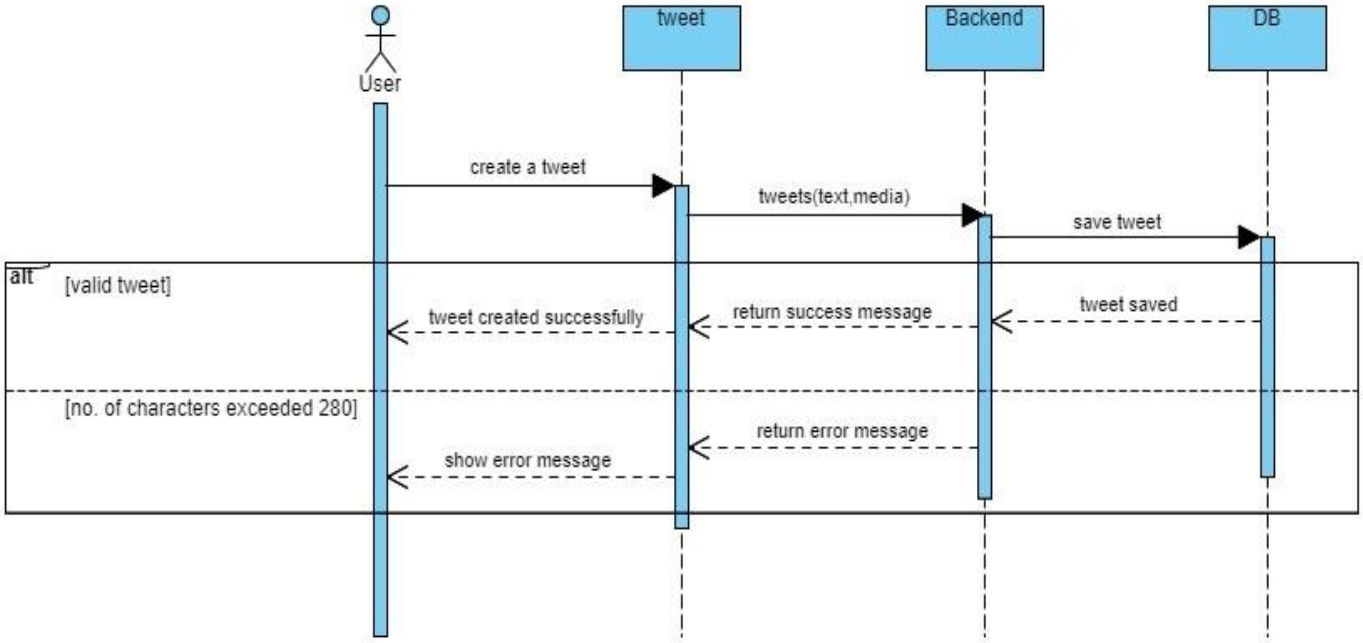
Sign up



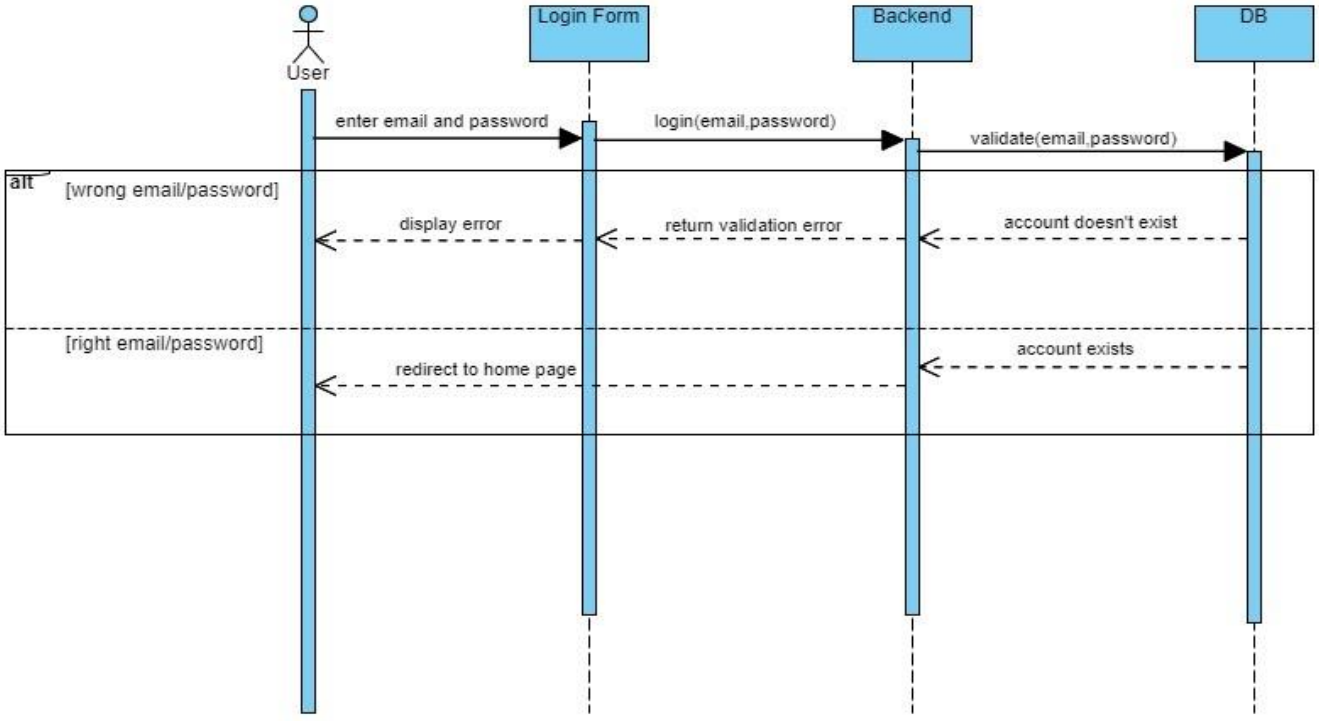
Retweet



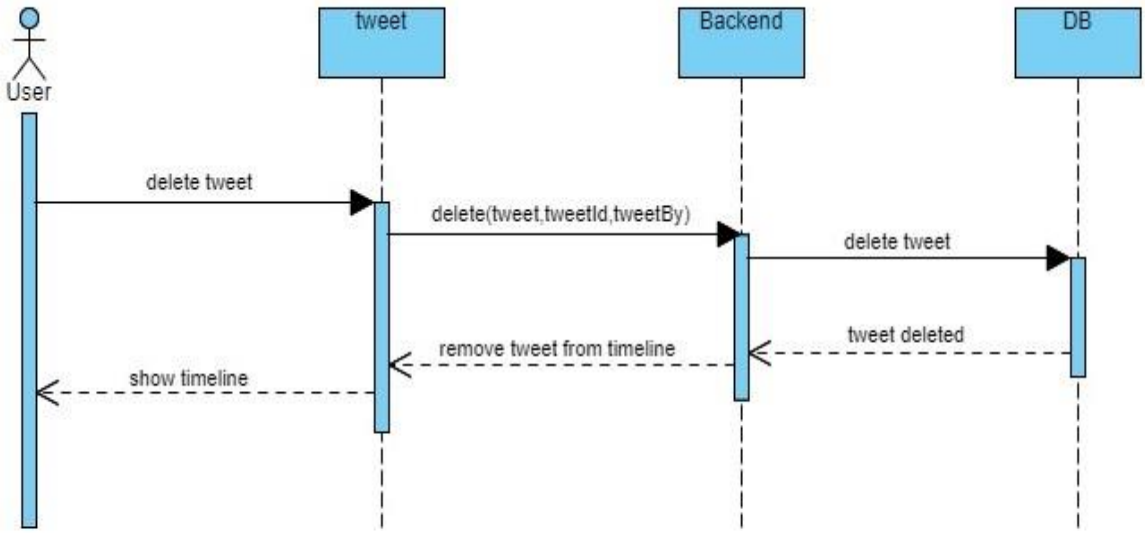
Posting a tweet



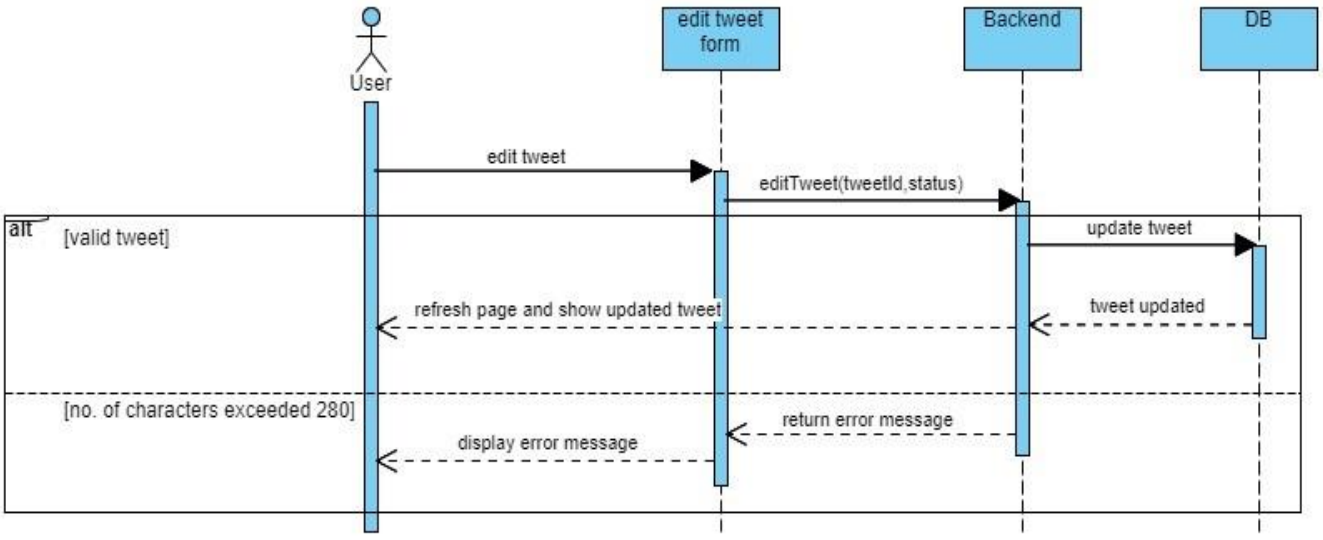
Login



Delete a tweet

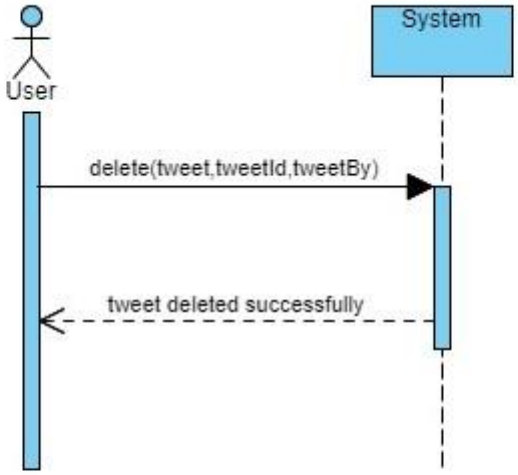


Edit a tweet

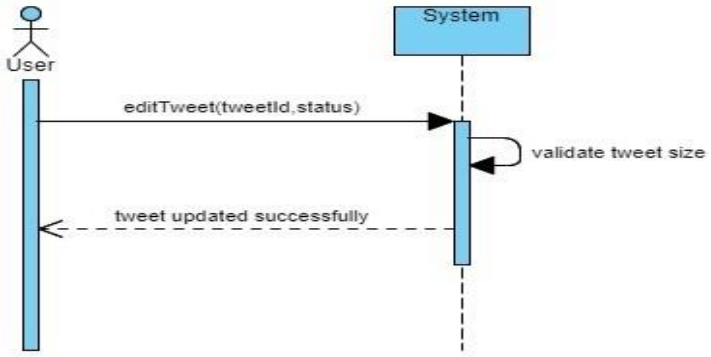


System sequence diagram:

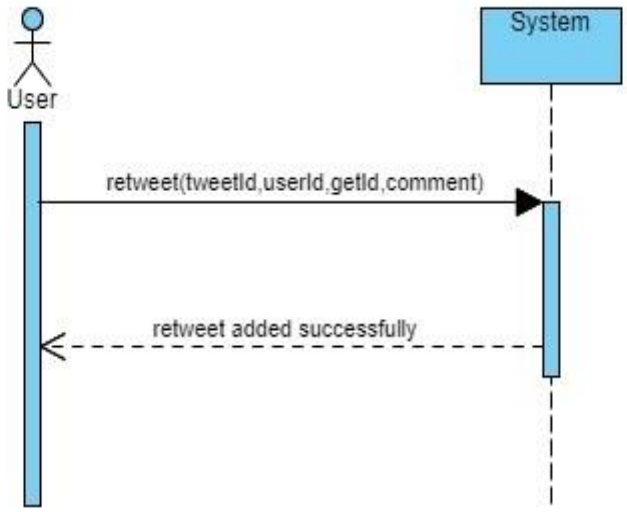
Delete a tweet



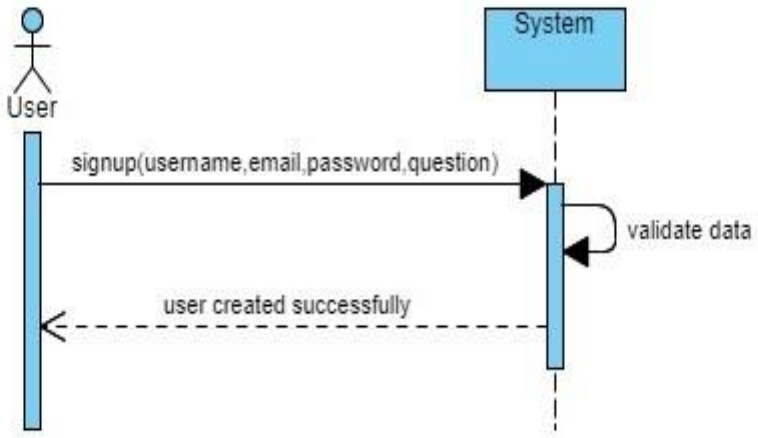
Edit a tweet



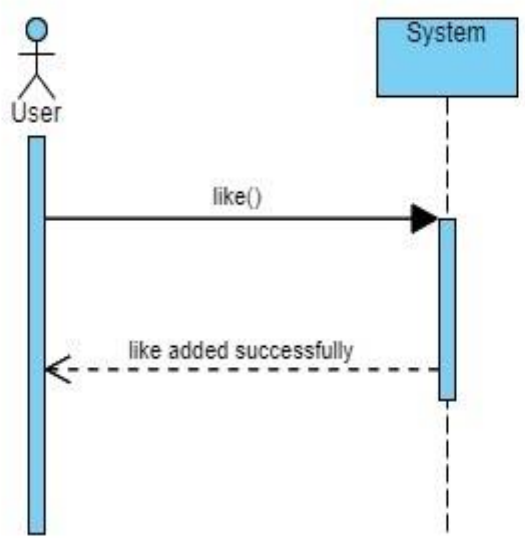
Retweet



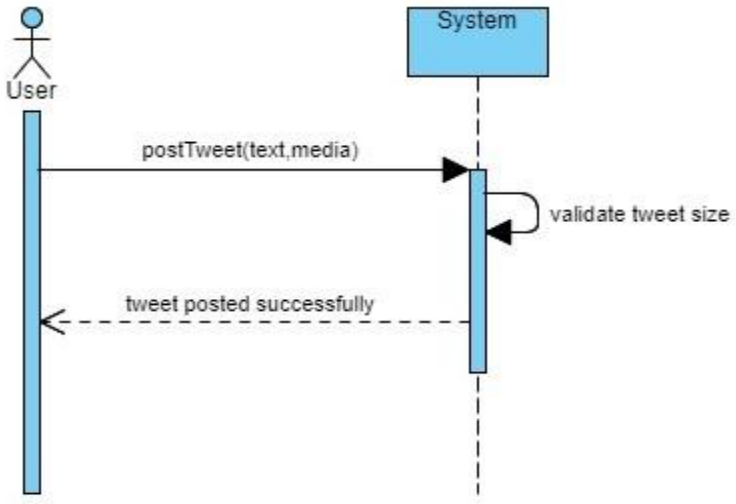
Sign up



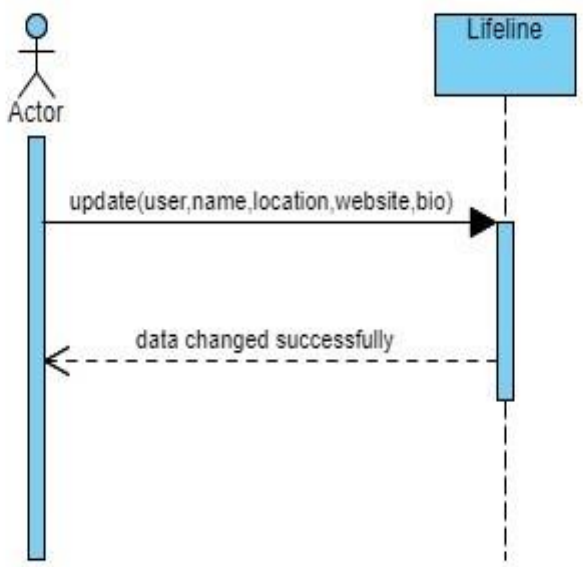
Adding a like



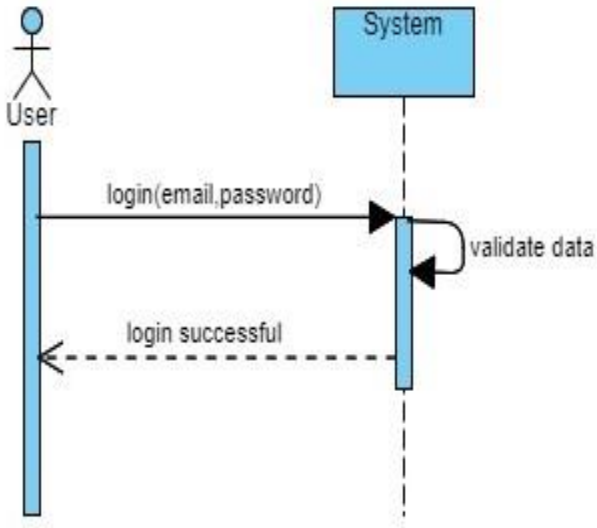
Posting a tweet



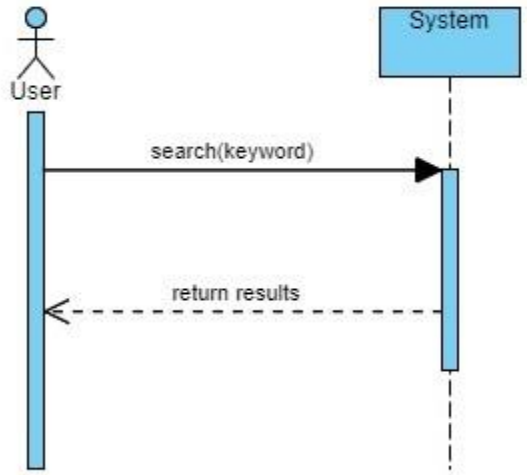
Edit profile



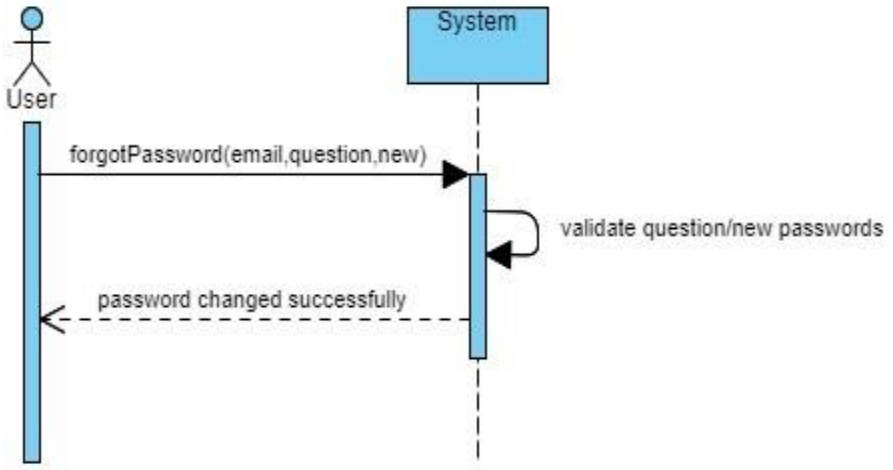
Login



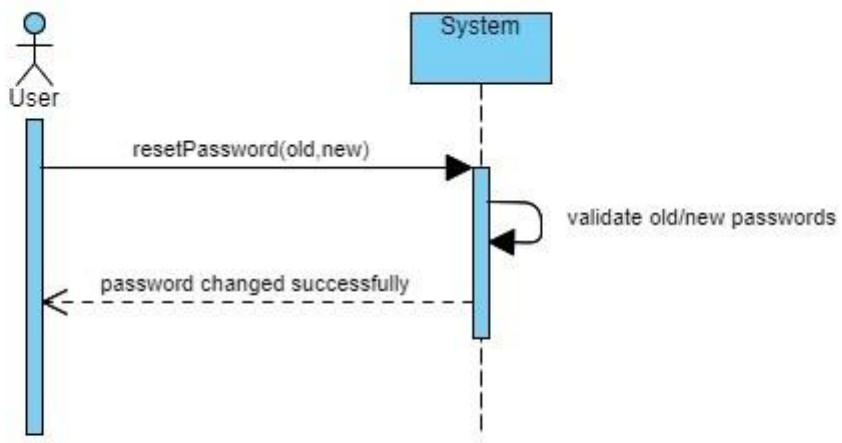
Search bar



Forget password (login form)



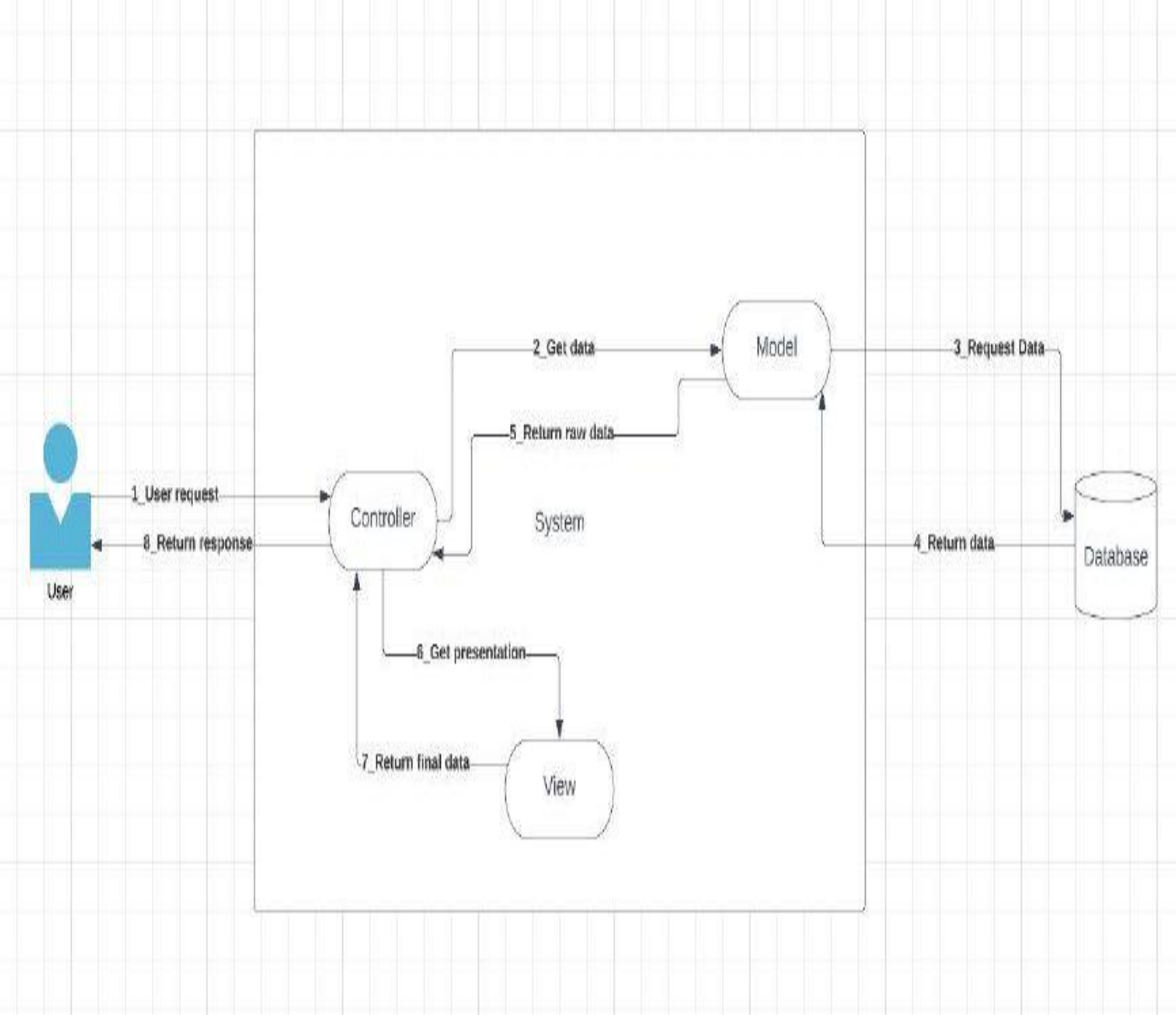
Reset password (settings)



System Architecture:

- **The system architecture consists of three elements:**

1_Controller: handles request flow and never handle data logic
2_model: handles data logic and interacts with the database
3_view: handles the data presentation and is what the user sees.



•The description of how the system will work:

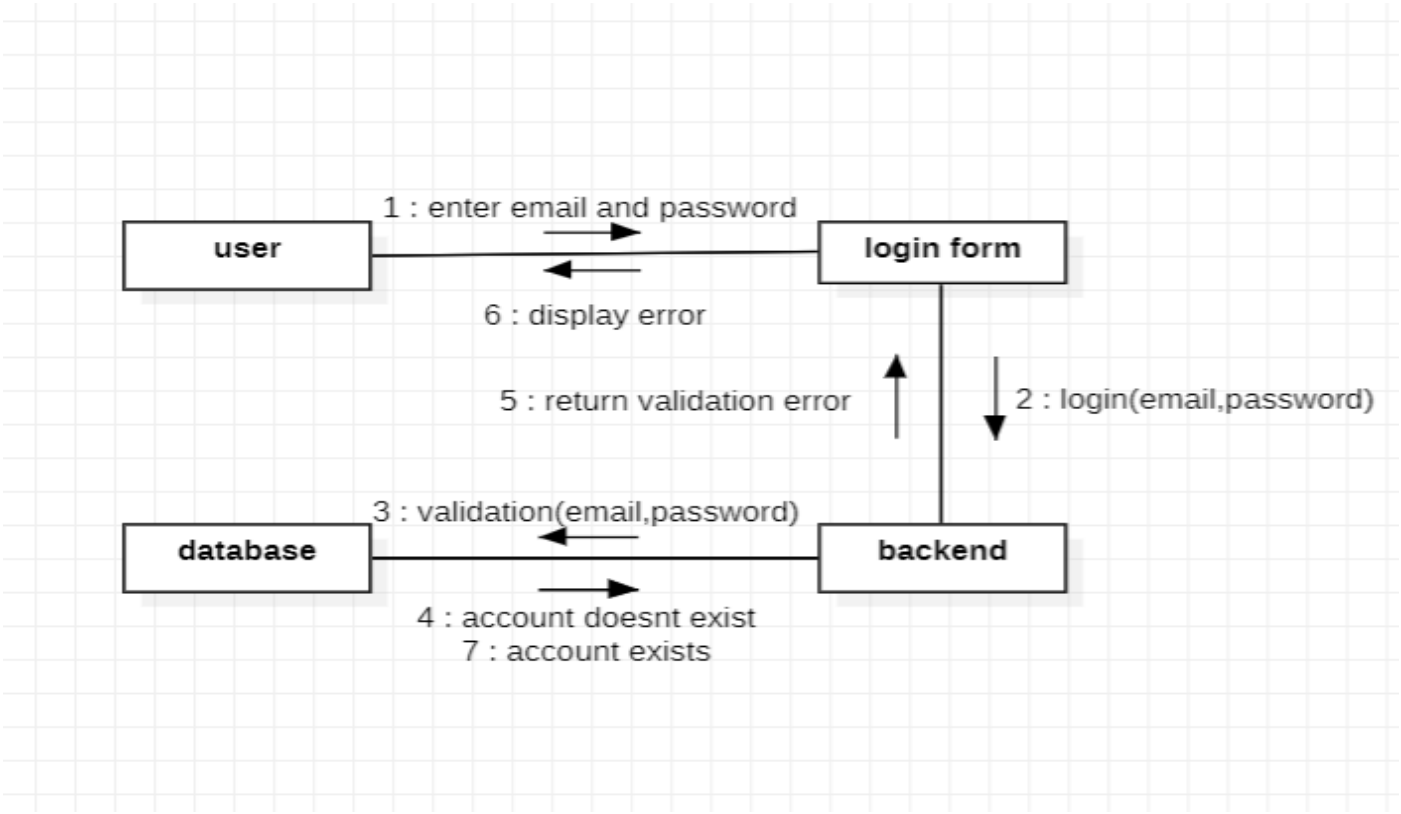
1_User interacts with the system in any way like (retweeting, replying or blocking …etc.)
2_The **controller** will take the request and ask the model to get the data that the user wants
3_The **model** will then request the data from the database and handles it and return it to the controller
4_The **controller** will take data and forward it to view to get a presentation of the data
5_The **view** will take the data from the controller and return a presentation of the data
6_The **controller** will finally take the presentation of the data from the view and return it to the user

• Benefits of using MVC pattern:

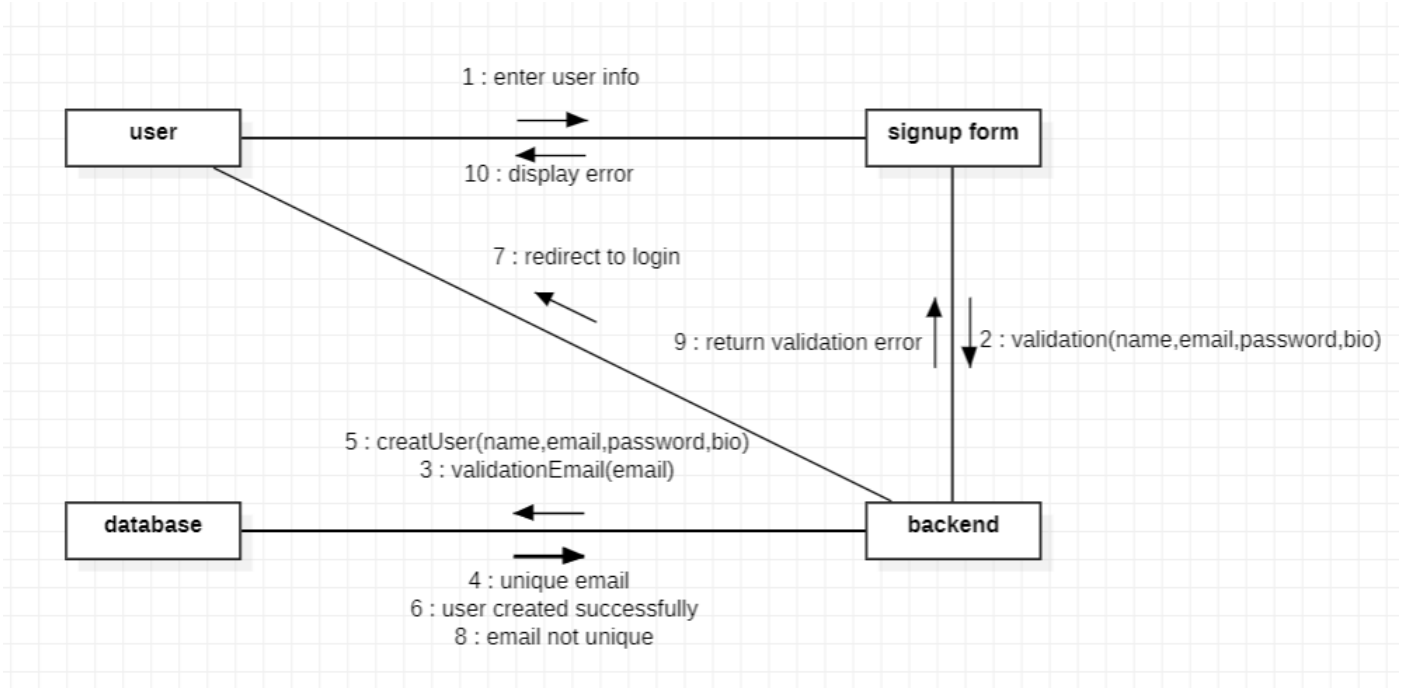
1_ Easy code maintenance
2_ It avoids complexity by dividing the system into three parts
3_ Modification does not affect the whole system

Collaboration/Communication Diagrams:

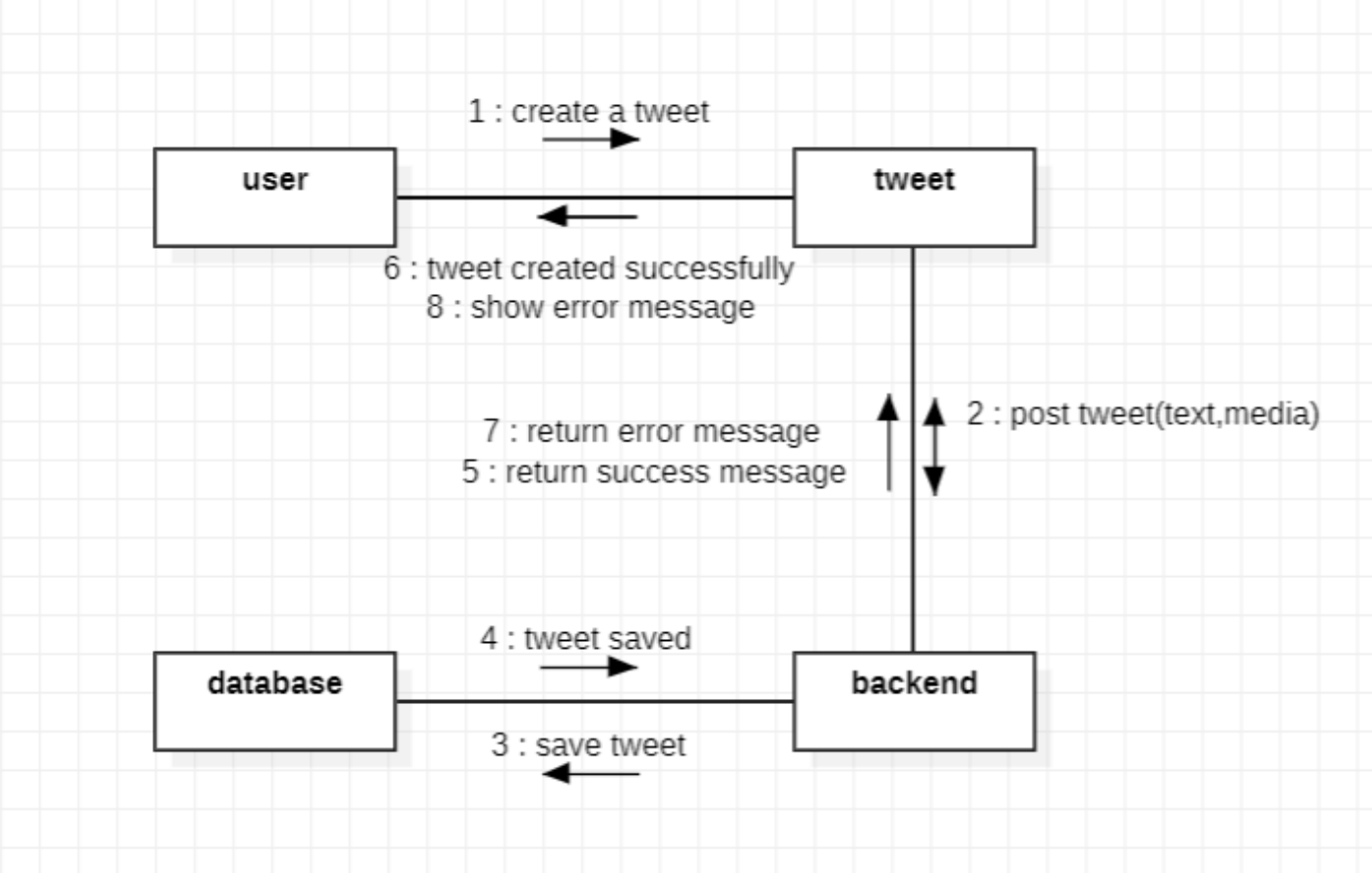
Login



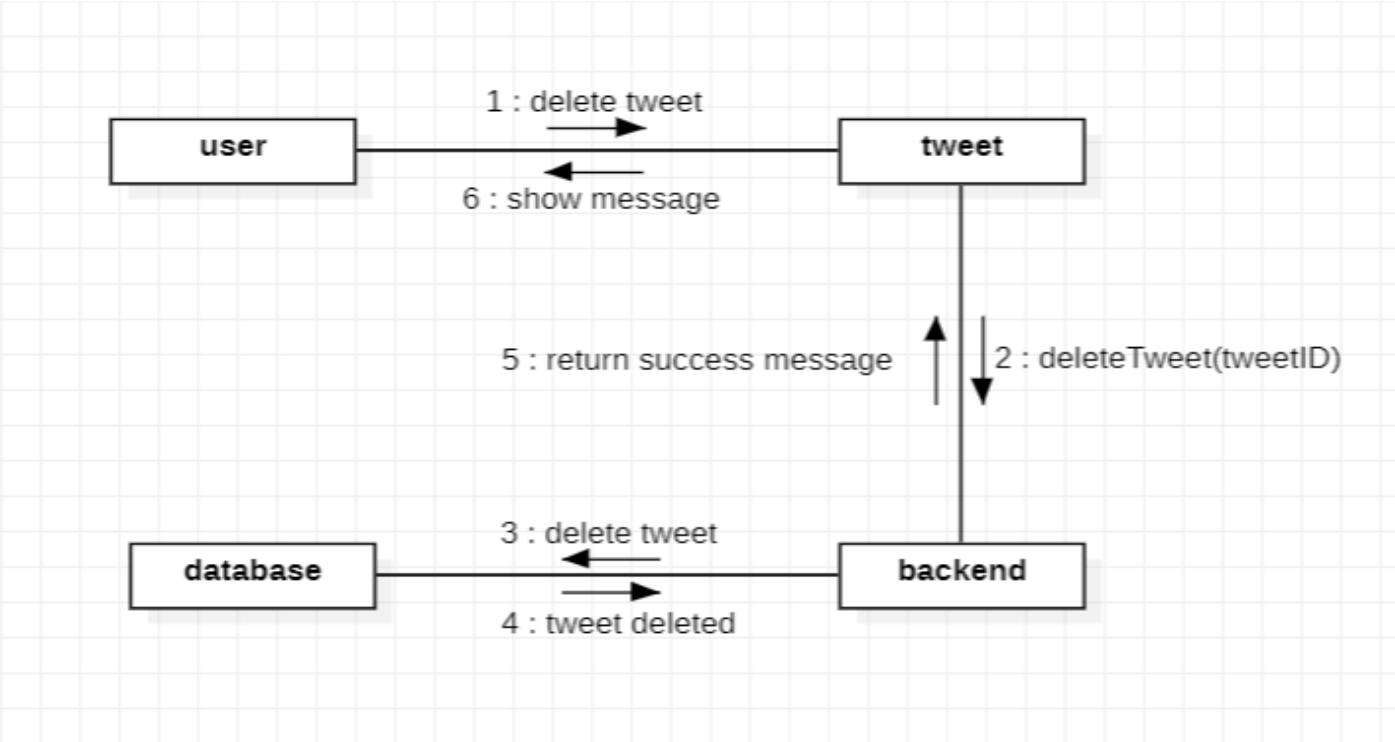
Sign up



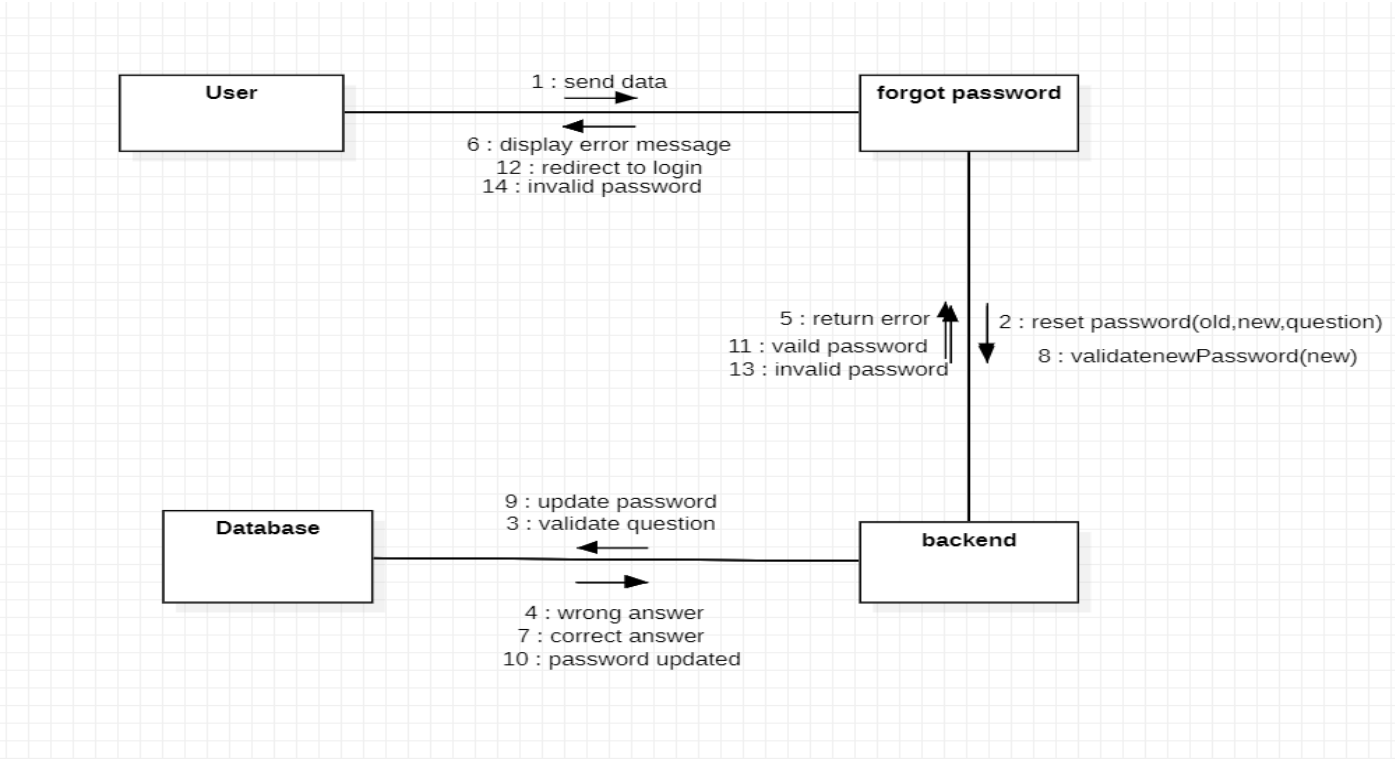
Posting a tweet



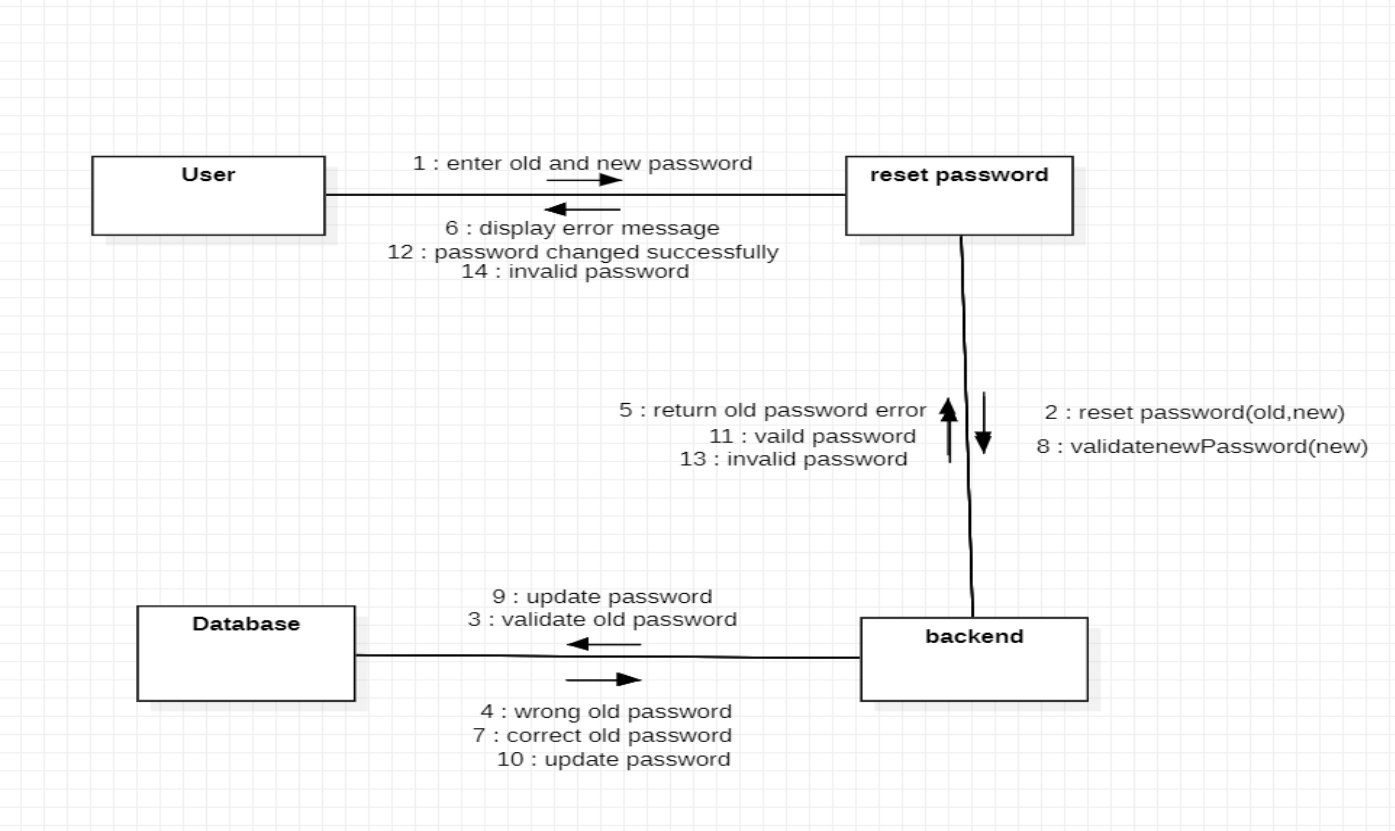
Delete a tweet



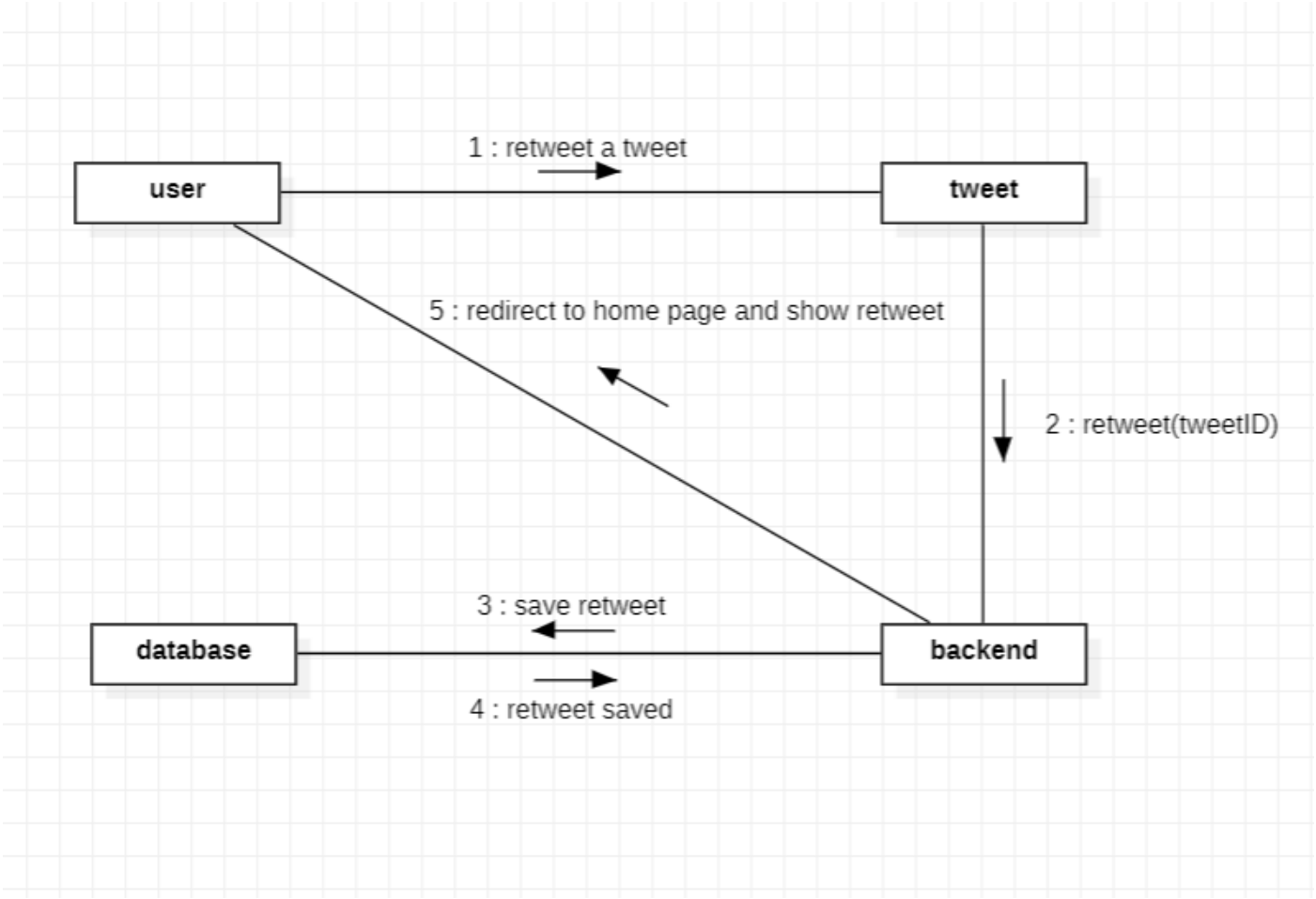
Forget password (login form)



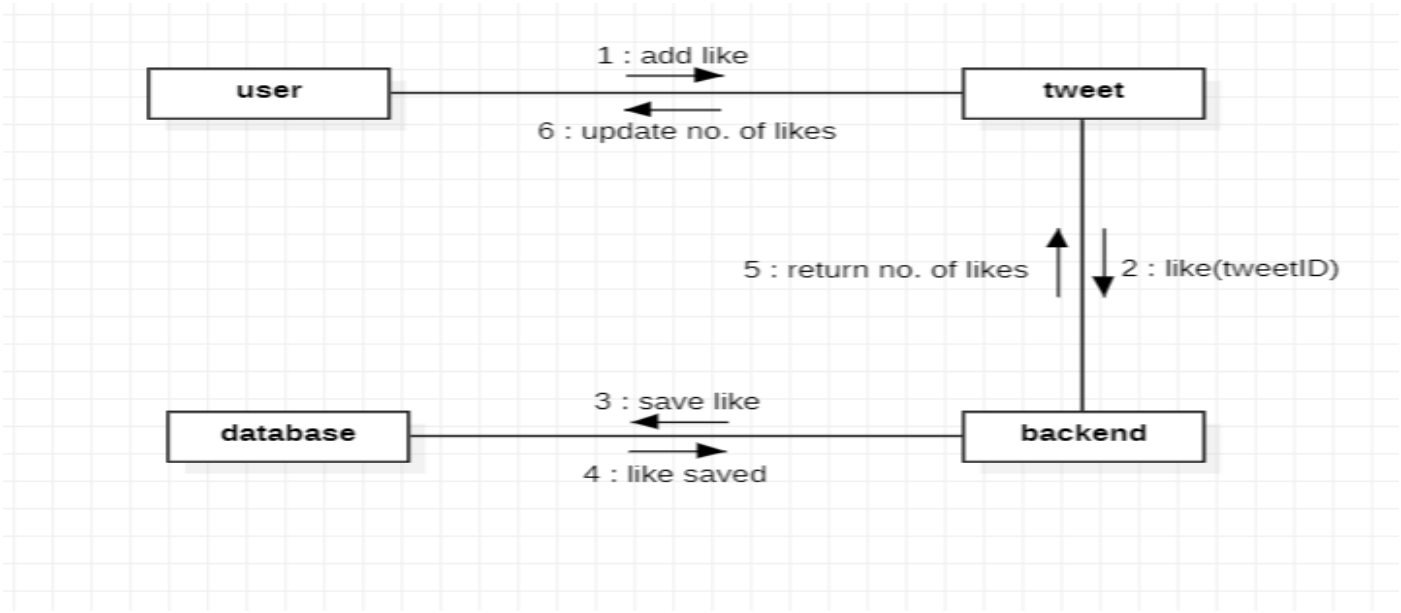
Reset password (settings)



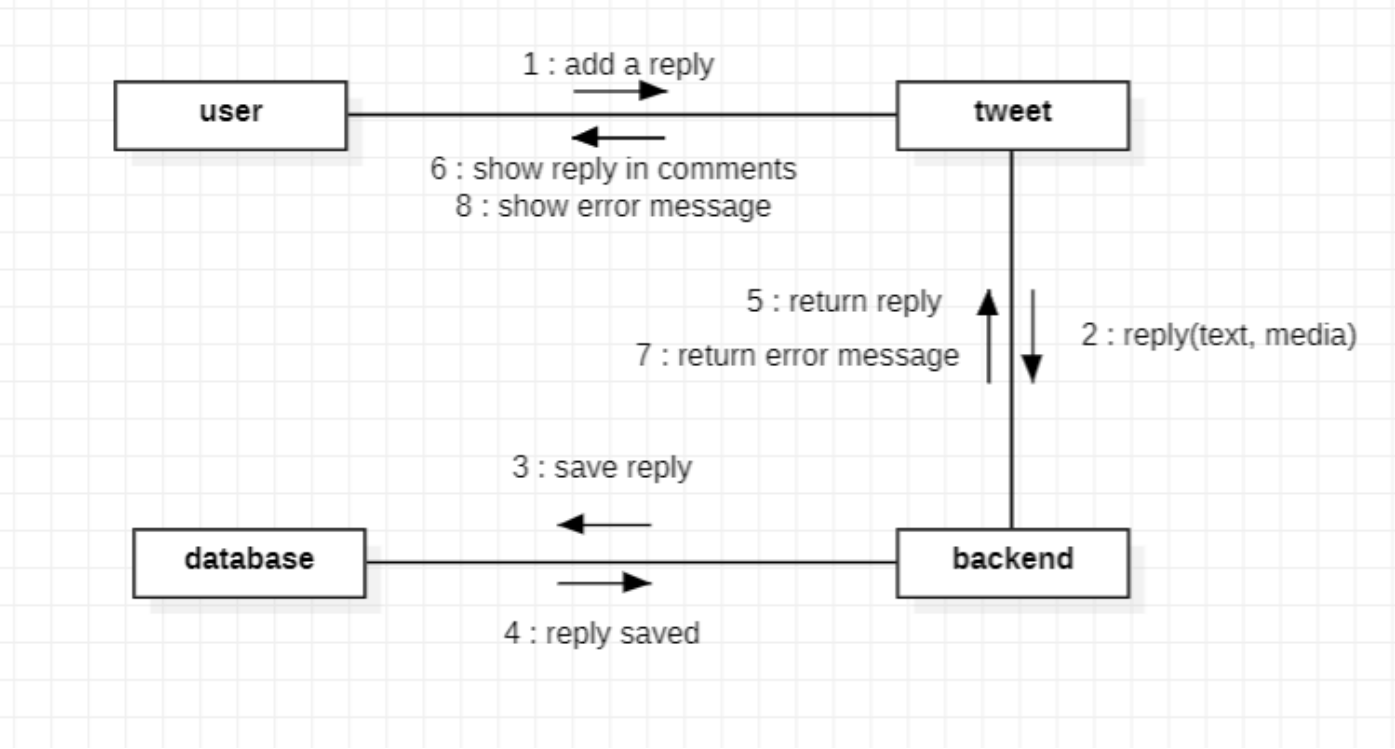
Retweet



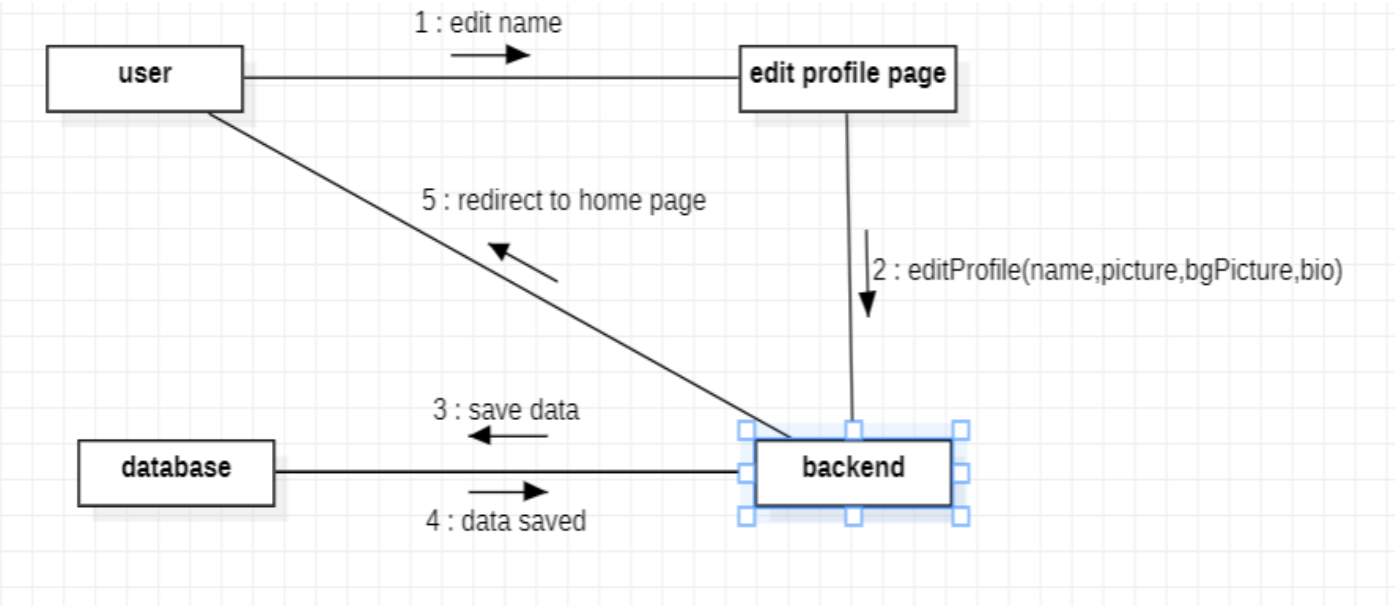
Adding a like



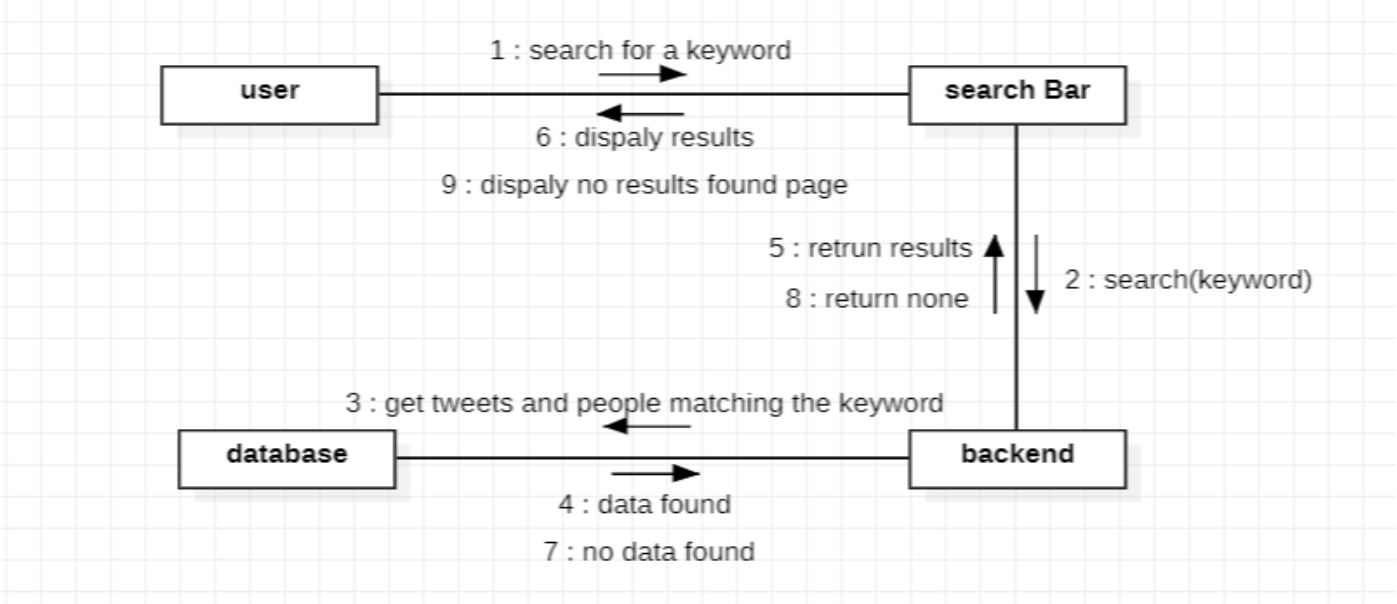
Adding a comment



Edit profile

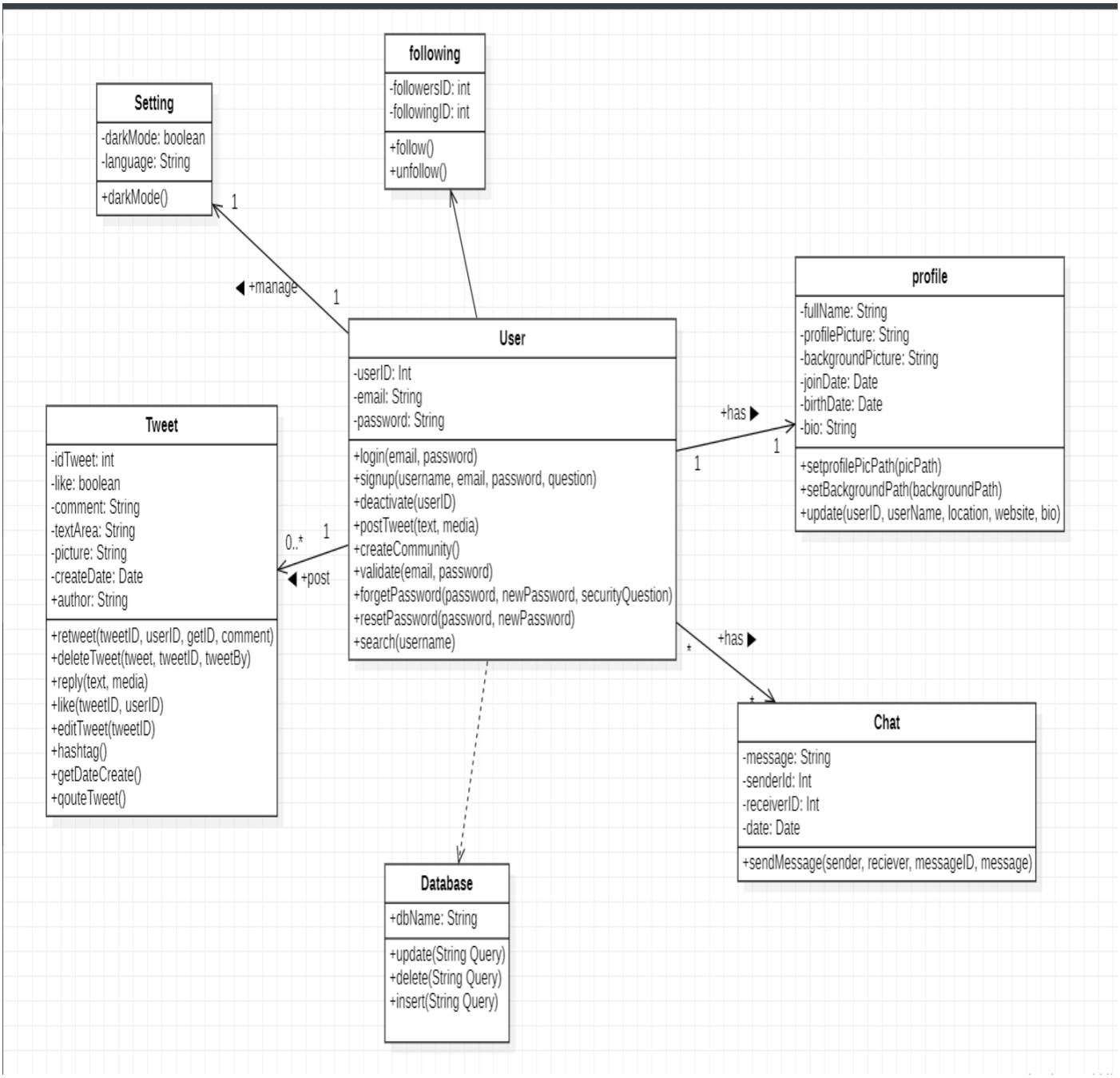


Search bar

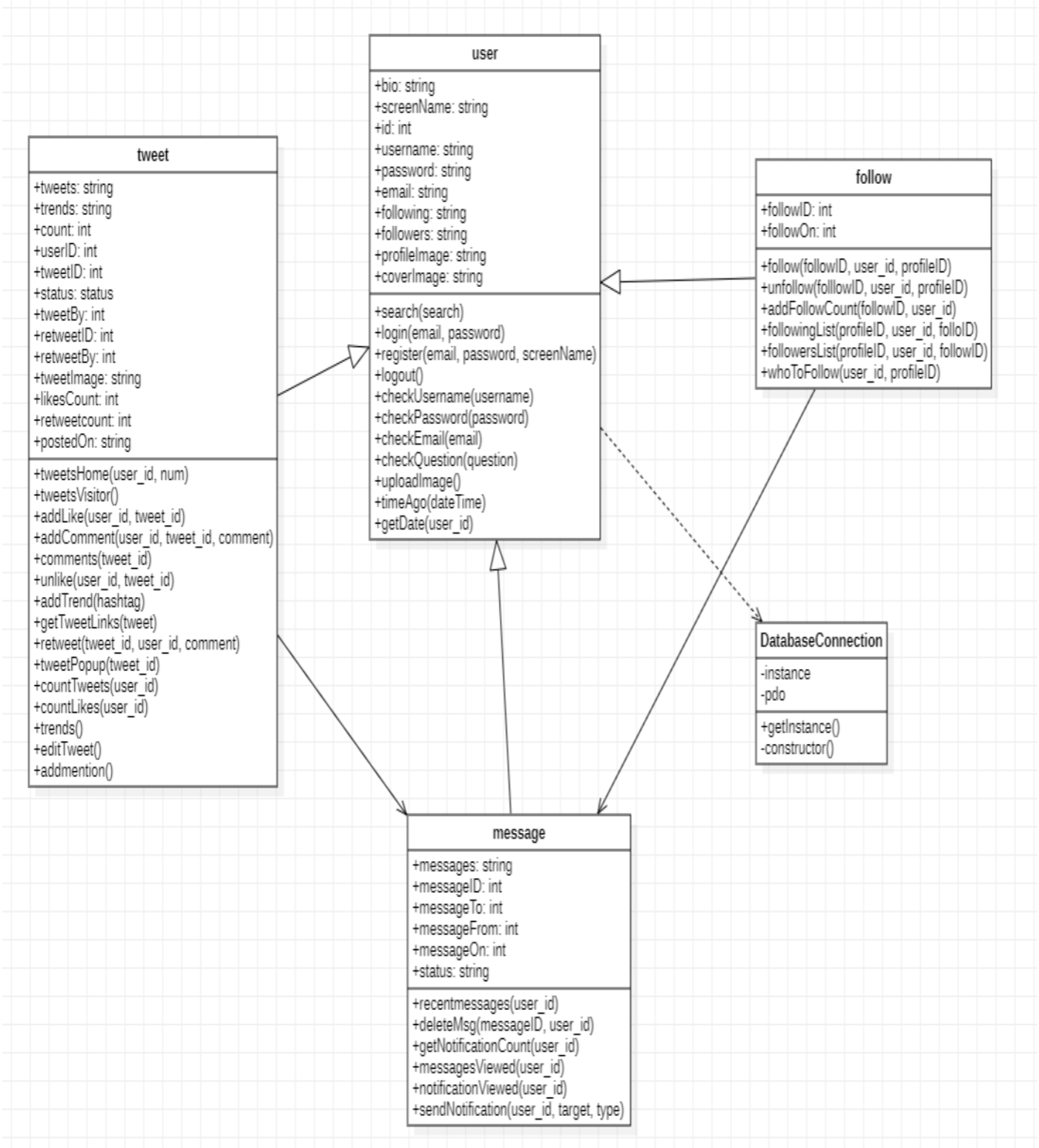

user
search Bar
database
backend
1 : search for a keyword
6 : dispaly results
9 : dispaly no results found page
5 : retrun results
8 : return none
2 : search(keyword)
3 : get tweets and people matching the keyword
4 : data found
7 : no data found

Class diagram:

(An intermediate version)

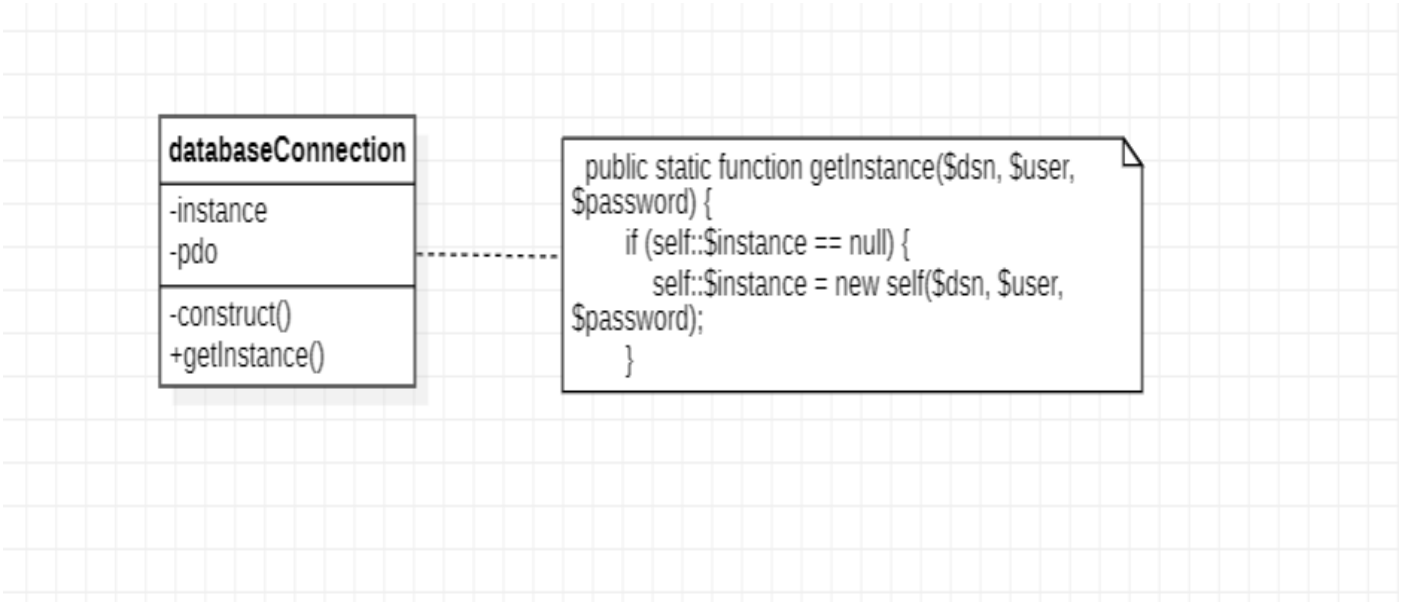


(Final version)

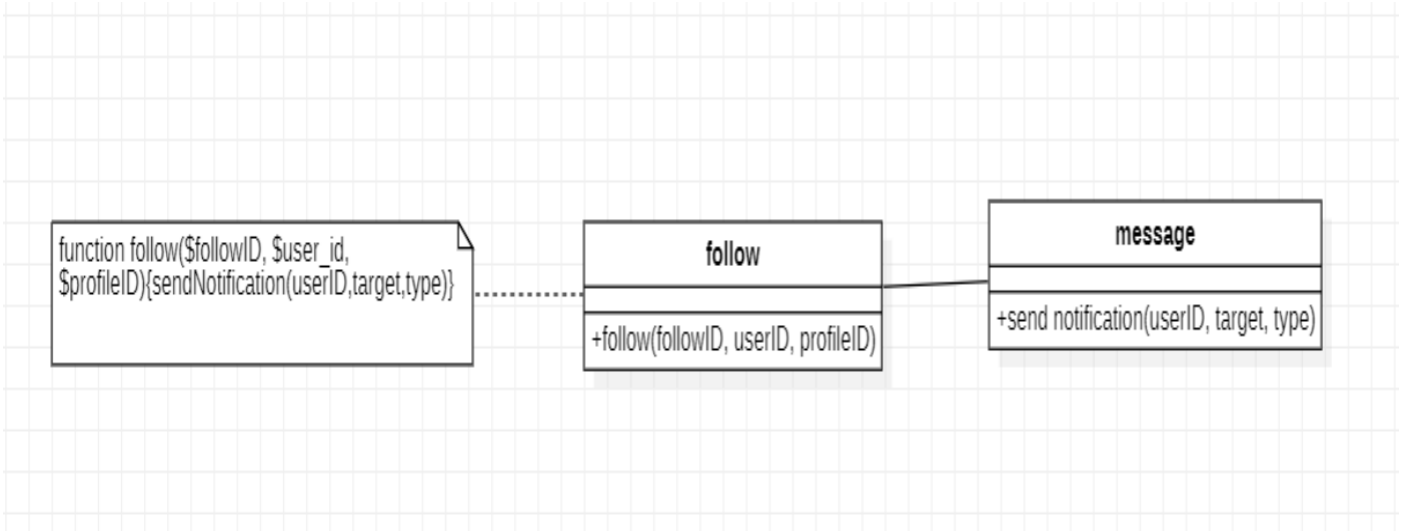


Three Mandatory Design Pattern:

Singleton pattern:



Delegation pattern:



- **Description:**

Context:

the system only has one database connection

problem:

make more than one instance for a single class that refers to one database that the system connected with

forces:

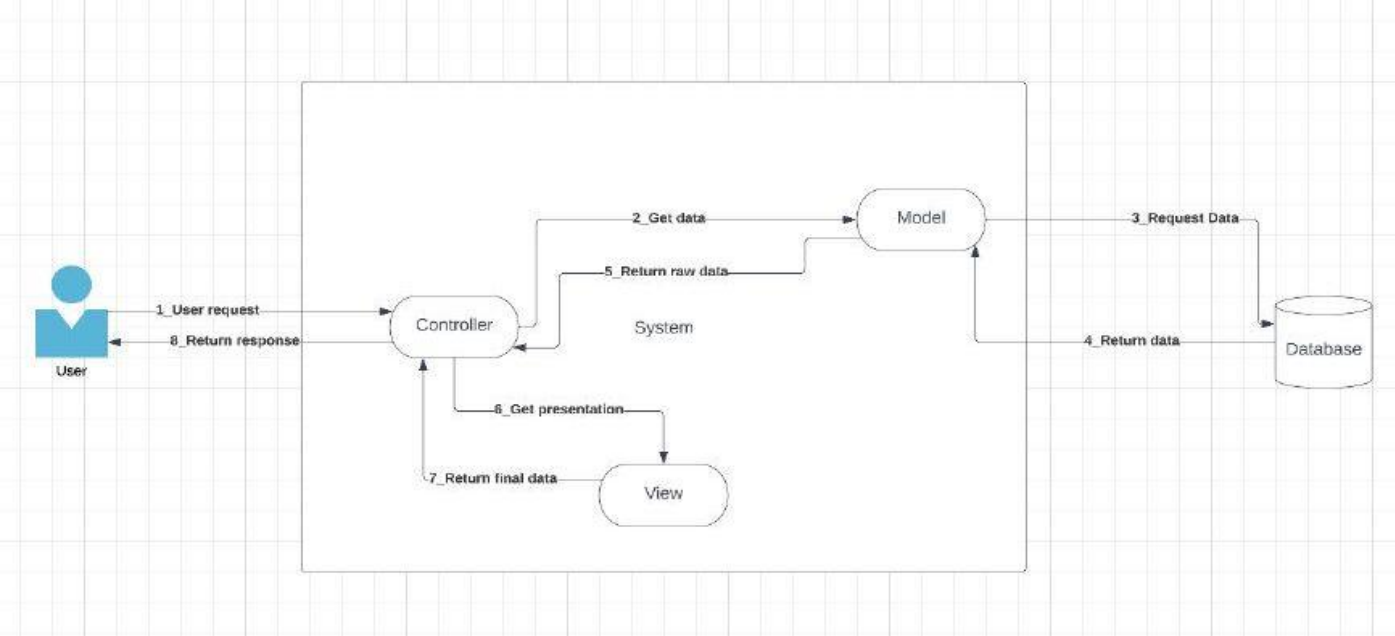
public constructor can make more than one instance.
The instance must be accessible to all other classes.

solution:

make the constructor private to ensure no other instances is created.

make the getInstance function public so it will be accessable to all classes.

System Architecture (MVC):



1- Models: which has the classes

2- views: has the interface the user interacts with

3-controllers: controls the variables in the models