INDEX

SRS

Software Requirement Specification Document

Introduction

Purpose

This SRS describes the scope and system design of “Wazifa” which is created in sake of making it easier to connect with other people works the same as you work or working in the same job Wazifa will minimize the time and efforts in searching for companies or other workers.

Scope

“Wazifa” is a business and employment-oriented social media service that works as web application that allows users to make friends with each other and post anything about jobs or advice to other employees.

Users can sign-up and make their profile as their CV they will add their skills past jobs, current jobs, education .
They can comment, like or share someone's post and they can chat with any user.
Users can add a page about the company they work in with logo, description, position.
Users can search about other users and companies and discover their profile or page.
Users will be able to update their profile info (CV)

an administrator will use web-portal to administrate the system and keep the information accurate. The admin can verify the company page and manage its information.

Out Scope

There is not job posting
no bidding in jobs
Not commenting someone's post if he isn't friend.
you can’t login by fake mail or password.

overview

“Wazifa” is built to make it easier to find other people works with you in the same company or works the same work as you are and make friends with them in order to learn from them and exchange your experiences.

General Description

Product Functions

This system helps you to connect with other people in your career to add more information and experience to you
You also can search about friends and companies.

Similar System Information

Our system is more similar to “LinkedIn”. it also can search about jobs in this application and make relation between people in your career like social media

User Characteristics

We have three actors in our application including admin, visitor and member
Admin is the person who control the whole application by adding and removing posts, users and accounts
Visitor is the person who sign up to the application and can view

someone's profile
member is the person who add friends, put posts, sending message, like, make comment and sharing the posts.

General Constraints

System can't work without DB working,
can't work locally,

Function Requirement

visitor

Code	*1.00*
Name	Sign up
Type	Functional Requirements
Criticality	High
Input	Name, Email, password
Output	Registered to the system and Logged in
Description	Sign up in the function that the user will register himself in the system to be able to sign in
Priority	10/10
Expected risks	Database registration failed
Preconditions	He shouldn't have an account
Post-conditions	The user is registered on the system
Dependencies	

Code	*1.01*
Name	View user profile
Type	Functional requirement
Criticality	Low
Input	Profile name
Output	Profile
Description	User wants to see someone's profile and last posts or see his/her skills
Priority	5/10
Expected risks	Database crashes while getting information
Preconditions	User have account already
Post-conditions	The profile is viewed
Dependencies	2.00

Code	*1.02*
Name	View post
Type	Functional requirement
Criticality	Low
input	
output	Post
Description	User wants to see any post of any profile
Priority	5/10
Expected risks	Database crashes while getting information
preconditions	
Post-conditions	The post is viewed
Dependencies	2.00

member

Code	*2.00*
Name	Sign in
Type	Functional Requirement
Criticality	High

Input	Email, Password
Output	Successfully logged in.
Description	Logging in is the entrance to the site every user should sign in to be able to connect with other
Priority	10/10
Expected risks	Database crashes while getting information
Preconditions	He must have an account
Post-conditions	The user is logged in the system
Dependencies	1.00

Code	*2.01*
Name	Log out
Type	Functional requirement
Criticality	Low
Input	
Output	Get out of from his profile
Description	This function make prevent user from access to the system by his user unless he logged in again
Priority	5/10
Expected risks	Bug, the logout button not working
Preconditions	He must be logged in
Post-conditions	He will return to system home page
Dependencies	2.00

Code	*2.02*
Name	Search
Type	Functional requirement
Criticality	Low
Input	Text
Output	Profile has the same text
Description	The user types the text he wants to fiend and start search and we provide options user can search by the name or the career

Priority	5/10
Expected risks	The system doesn’t respond
Preconditions	He must be logged in
Post-conditions	The result of search is given
Dependencies	2.00

Code	*2.03*
Name	Like post
Type	Functional Requirement
Criticality	Low
input	Post's id
output	You liked someone’s post
Description	It is a way of interaction with the post.
Priority	5/10
Expected risks	Bug, the like button not working, database not saving likes on post
preconditions	He must be logged in
Post-conditions	The user will show that he post liked and see users who liked his post
Dependencies	2.00 , 1.02

Code	*2.04*
Name	Comment
Type	Functional Requirement
Criticality	Low
input	Post id and text
output	Your comment will be shown to you
Description	Is the way of interaction with post by adding your comment about it
Priority	5/10
Expected risks	Bug, the follow button not working
preconditions	He must be logged in and friend or follower with who put the post
Post-conditions	The comment will be shown to anyone to see and will be added to the post
Dependencies	2.00 , 1.02

Code	*2.05*
Name	Post
Type	Functional Requirement
Criticality	Low
Input	Any text or link or image or video
Output	Showing the input as a post.
Description	Making new post is the way people will type about anything they want to share with other users
Priority	5/10
Expected risks	Crashes on post, crash while saving in DB
Preconditions	He must be logged in
Post-conditions	The post will be published and will be shown to other people
Dependencies	2.00

Code	*2.06*
Name	Share
Type	Functional requirements
Criticality	Low
input	Post's id
output	Post has been shared
Description	User can share any post he wants to post it in his timeline
Priority	5/10
Expected risks	Bug, the share button not working
preconditions	Must be logged in
Post-conditions	Post has been published in user timeline
Dependencies	2.00 , 1,02

Code	*2.07*

Name	Message
Type	Functional Requirement
Criticality	Low
Input	Text
Output	Message is sent
Description	User choose the user he wants to contact with him then type the message he wants
Priority	5/10
Expected risks	Database crashes while getting information
Preconditions	He must be logged in
Post-conditions	Message sent successfully
Dependencies	2.00,1.01

Code	*2.08*
Name	View Messages
Type	Functional Requirement
Criticality	Low
Input	
Output	Messages is viewed
Description	Able user to see his message in his inbox
Priority	5/10
Expected risks	User don't have any messages
Preconditions	He must be logged in
Post-conditions	User showed his messages successfully
Dependencies	2.00

Code	*2.09*
Name	Follow
Type	Functional Requirement
Criticality	Low

Input	Profile name
Output	Successfully followed a person
Description	Following someone will let you able to view his posts and interact with them by comment or like
Priority	5/10
Expected risks	Bug, the follow button not working
Preconditions	He must be logged in
Post-conditions	The followed user will be added to follow list and his posts will be shown to the follower
Dependencies	2.00 , 1.01

Code	*2.10*
Name	View followers
Type	Functional requirement
Criticality	Low
Input	
Output	List of user's follower
Description	User can see the people who follow him
Priority	5/10
Expected risks	Database crashes while getting information
Preconditions	He must be logged in
Post-conditions	User will see list of followers
Dependencies	2.00 , 2.22

Code	*2.11*
Name	View following list
Type	Functional requirement
Criticality	Low
Input	
Output	List of user's following
Description	User can see the people he followed to contact easy with them or see their last posts

Priority	5/10
Expected risks	Database crashes while getting information
Preconditions	He must be logged in
Post-conditions	User will see list of his followed profiles
Dependencies	2.00 , 2.22

Code	*2.12*
Name	Add Friend
Type	Functional Requirement
Criticality	Low
Input	Profile name
Output	send a request to person
Description	Adding someone make you contact with him easy , see his posts and comment on it .
Priority	5/10
Expected risks	Bug, the add Friend button not working
Preconditions	He must be logged in
Post-conditions	The added user will be receive add request from the user
Dependencies	2.00 , 1.01

Code	*2.13*
Name	Accept friend
Type	Functional Requirement
Criticality	Low
Input	Profile
Output	Successfully added a person
Description	Accept add request means add someone to his friend list
Priority	5/10
Expected risks	Bug, the accept Friend button not working
Preconditions	He must be logged in

Post-conditions	The added user will be added to friend list and his posts will be shown to the friends
Dependencies	2.00,2.20

Code	*2.14*
Name	View Friends list
Type	Functional Requirement
Criticality	Low
Input	
Output	List of user's Friends
Description	This function provide you easy way to see the list of your friends
Priority	5/10
Expected risks	Database crashes while getting information
Preconditions	He must be logged in
Post-conditions	You can see all your friends
Dependencies	2.00 , 2.22

Code	*2.15*
Name	Update profile
Type	Functional requirements
Criticality	Low
input	Name or password o skills
output	Profile has been updated
Description	User can update all of his information like name ,password, age or update his career and experiance
Priority	5/10
Expected risks	Bug, the update button not working
preconditions	Must be logged in
Post-conditions	Profile has been updated
Dependencies	2.00 ,2.22

Code	*2.16*
Name	View notification
Type	Functional requirements
Criticality	Low
input	
output	List of notification
Description	By this function user can knew how many member accepts his add request, like ,comment in his post or share it
Priority	5/10
Expected risks	Database crashes while getting information
preconditions	Must be logged in
Post-conditions	Notification has been showed
Dependencies	2.00

Code	*2.17*
Name	Retrieve Password
Type	Performance requirements
Criticality	Low
input	Email
output	Verification code
Description	If user forget his password he can retrieve it again
Priority	5/10
Expected risks	Database crashes while getting information
preconditions	Must have an account
Post-conditions	Password has been retrieved
Dependencies	1.00

Code	*2.18*
Name	Upload photo
Type	Functional requirements
Criticality	Low
input	Photo
output	Photo
Description	User can upload a photo to his profile easy
Priority	5/10
Expected risks	Bug, the upload button not working
preconditions	Must be logged in and in his profile
Post-conditions	User's photo has been uploaded
Dependencies	2.00 , 2.22

Code	*2.19*
Name	Show likes
Type	Functional requirements

Criticality	Low
input	Post
output	User who like the post
Description	User can see the member who like any post
Priority	5/10
Expected risks	Database crashes while getting information
preconditions	Must be logged in and viewed post
Post-conditions	User knew the member who like the post
Dependencies	2.00 , 1.02

Code	*2.20*
Name	Show comments
Type	Functional requirements
Criticality	Low
input	Post
output	User who comment in the post
Description	User can see the comments in any post
Priority	5/10
Expected risks	Database crashes while getting information
preconditions	Must be logged in and viewed post
Post-conditions	User see the comments in any post
Dependencies	2.00 , 1.02

Code	*2.21*
Name	Show shares
Type	Functional requirements
Criticality	Low
input	Post

output	User who share the post
Description	User can see the member who share any post
Priority	5/10
Expected risks	Database crashes while getting information
preconditions	Must be logged in and viewed post
Post-conditions	User knew the member who share the post
Dependencies	2.00 , 1.02

Code	*2.22*
Name	View own profile
Type	Functional Requirement
Criticality	law
Input	
Output	User profile oppend
Description	user can view his profile to edit in it ,remove things or add things in it
Priority	5/10
Expected risks	Database crashes while getting information
Preconditions	He must have an account
Post-conditions	The pofile page opened
Dependencies	1.00

Code	*2.23*
Name	View friend request
Type	Functional Requirement
Criticality	law
Input	Friend request IDs
Output	Names of friend requests

Description	It’s the place where the person is able to view people who send him add request with the ability to add them
Priority	5/10
Expected risks	Database crashes while getting information
Preconditions	He must have an account
Post-conditions	All requests data will be shown
Dependencies	2.00

Admin

Code	*3.00*
Name	Remove comment
Type	performance Requirement
Criticality	Low
Input	Comment's id
Output	Your comment successfully deleted
Description	It is an option to delete the comment you typed by mistake or you don’t want anyone to see it
Priority	5/10
Expected risks	Bug, the comment not deleted from the database
Preconditions	He must be logged in, the comment must be typed
Post-conditions	The comment will be deleted
Dependencies	2.00 , 2.17

Code	*3.01*
Name	Remove user
Type	preformance requirement
Criticality	Low
input	Bad user's profile name
output	User removed
Description	Admin can remove bad user
Priority	5/10
Expected risks	Bug, the user not deleted from the database

preconditions	Must be logged in and user exists
Post-conditions	User will be removed
Dependencies	2.00 , 1.01

Code	*3.02*
Name	Remove post
Type	performance requirement
Criticality	Low
input	Bad Post's id
output	Post removed
Description	User also can remove bad posts which include bad posts
Priority	5/10
Expected risks	Bug, the post not deleted from the database
preconditions	Must be logged in and page exists
Post-conditions	Page will be removed
Dependencies	2.00 , 1.01

Code	*3.03*
Name	Get Report
Type	Performance Requirements
Criticality	Low
input	Report's id
output	Report
Description	Admin can view a Report about the site user's number
Priority	5/10
Expected risks	Database crashes while getting information
preconditions	Must be logged in
Post-conditions	Report is viewed successfully
Dependencies	2.00

Performance Requirements

Site needed to load from 5-30 seconds at most
Needed at least 50 gigabyte storage

Other non-functional attributes

code	4.1
name	System dependability
type	performance requirement
description	The fault tolerance of the system ,if the system loses the connection to the internet or the user gets some strange input ,the user should be informed
Priority	10/10

code	4.2
name	System security
type	Security requirement
description	Security of the communication between the system and the server ,message should be encrypted for login communication so other can't see user password
Priority	10/10

code	4.3
name	User create account security
type	Security requirement
description	If the user wants to create an account and the desired name is occupied ,the user should be asked to choose a different user name
Priority	10/10

code	4.4
name	Admin login account security
type	Security requirement
description	If an admin tries to log in to the web portal with a non-exiting account then the admin should not logged in, the admin should notified the log in failure
Priority	10/10

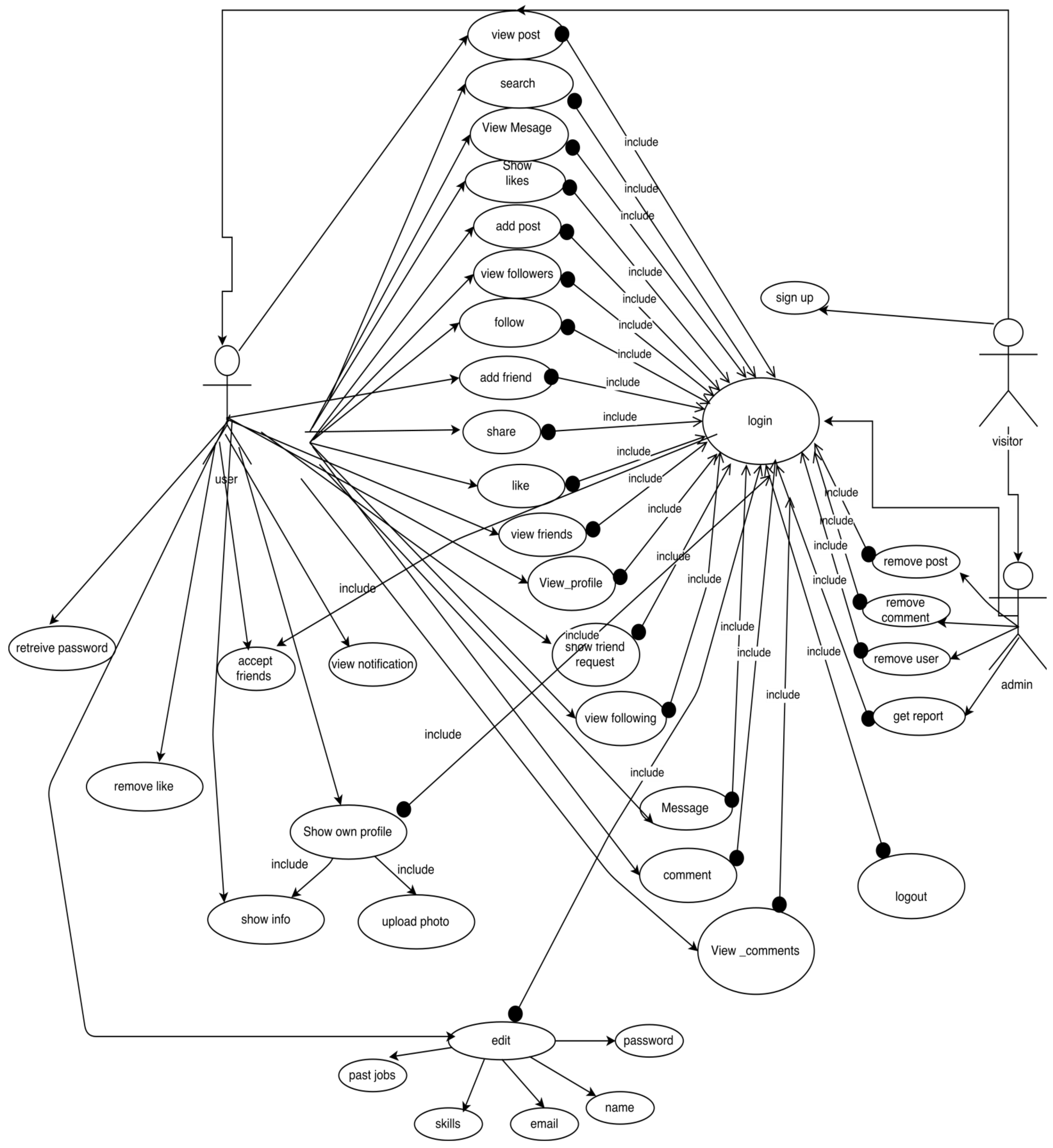
code	4.5
name	System reliability
type	Performance requirement
description	The reliability of the system that it gives that it gives the right result on a search
Priority	10/10

code	4.6
name	System availability
type	performance requirement
description	The availability of the system when it is used (not considering network failing)
Priority	10/10

code	4.7
name	Response time
type	usability requirement
description	The fastness of the search the response time of a search
Priority	10/10

Operational Scenarios

Use case diagram



Use case scenario

USE CASE	REGISTER
ACTOR	Visitor
PRE-CONDITION	He shouldn't have an account
BASIC FLOW	First the user click on register button then the system will show a register form he must fill in the data then click submit then the system will show him his profile and then the user must enter his skills and jobs and his education and complete registration then submit his changes
ALTERNATIVE FLOW	The system will show a message register not completed
POST CONDITION	The user is registered on the system

USE CASE	LOGIN
ACTOR	Member,admin
PRE-CONDITION	He must have an account
BASIC FLOW	First the user click on login button then the system will show a login form he must fill in the user name and the password click login
ALTERNATIVE FLOW	The system will show a message no account and then will direct him to register form
POST CONDITION	The user is logged in the system

USE CASE	ADD POST
ACTOR	Member
PRE-CONDITION	He must be logged in
BASIC FLOW	The user will click on new post button then he will write his post then click post button
ALTERNATIVE FLOW	The system will show him try again message
POST CONDITION	The post is shown in his timeline

USE CASE	FOLLOW
ACTOR	Member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on follow button
ALTERNATIVE FLOW	The followed person won't be added on the followed list
POST CONDITION	system will add the followed person on the follower list of the user

USE CASE	ADD FRIEND
ACTOR	Member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on add friend button
ALTERNATIVE FLOW	The chosen friend won’t be added on the friends list of the user and the system will show him try again message
POST CONDITION	The user chosen will be added in the friends list of the user

USE CASE	VIEW FRIENDS
ACTOR	Member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	the user click on view friends button and the system will respond
ALTERNATIVE FLOW	The system will show him no friends message
POST CONDITION	system will show the friends list

USE CASE	VIEW FOLLOWING
ACTOR	Member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view following button
ALTERNATIVE FLOW	The system will show him no following users
POST CONDITION	system will show the following list

USE CASE	LOGOUT
ACTOR	Member,admin
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on logout button
ALTERNATIVE FLOW	The system will not logout
POST CONDITION	The user must be logged out

USE CASE	VIEW POST
ACTOR	Member, admin, visitor
PRE-CONDITION	
BASIC FLOW	the user click on view profile button
ALTERNATIVE FLOW	The system may crashes and doesn’t response to the request so he will stay in the same page
POST CONDITION	system will show the profile of the user

USE CASE	**VIEW MESSAGES**
ACTOR	member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view messages button
ALTERNATIVE FLOW	The system will show him no messages
POST CONDITION	system will show the user his messages

USE CASE	**SEND MESSAGE**
ACTOR	member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user must choose the person he wants to contact with then click on message button then type his message then click send button
ALTERNATIVE FLOW	The system will not send the message
POST CONDITION	The message must be sent to the selected user

USE CASE	**SEARCH**
ACTOR	Member, admin
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user type the thing he want to fiend in the search bar then click search button
ALTERNATIVE FLOW	The system will show him not found or will show him similar things
POST CONDITION	system will view result

USE CASE	**VIEW PROFILE**
ACTOR	Member, admin, visitor
PRE-CONDITION	
BASIC FLOW	the user click on view profile button
ALTERNATIVE FLOW	The system may crashes and doesn’t response to the request so he will stay in the same page
POST CONDITION	system will show the profile of the user

USE CASE	**LIKE POST**
ACTOR	member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view like button
ALTERNATIVE FLOW	Like will not be added on the post
POST CONDITION	A like must be added to the post

USE CASE	**COMMENT**
ACTOR	member
PRE-CONDITION	He must have an account and be logged in
BASIC FLOW	First the user will choose the post then click add comment then type his comment then

	click ok
ALTERNATIVE FLOW	The system will not add the comment to the post
POST CONDITION	system will show the comment to the post

USE CASE	**REMOVE COMMENT**
ACTOR	Admin,member
PRE-CONDITION	He must have an admin of the system and be logged in
BASIC FLOW	The admin will click on X button in the top of comment
ALTERNATIVE FLOW	System shows message “comment not deleted”
POST CONDITION	The comment will be removed

USE CASE	**REMOVE MEMBER**
ACTOR	Admin
PRE-CONDITION	Must be logged in and member exists
BASIC FLOW	First the user click on view friends button
ALTERNATIVE FLOW	The system will show him no friends message
POST CONDITION	User will be removed

USE CASE	**SHOW INFO**
ACTOR	member
PRE-CONDITION	Must be logged in and open his profile
BASIC FLOW	The user will click on view profile button then he will click on view info
ALTERNATIVE FLOW	His personal info will not viewed
POST CONDITION	The member info will be shown in the current page

USE CASE	**REMOVE LIKE**
ACTOR	member
PRE-CONDITION	Must be logged
BASIC FLOW	Go to home then click on add page then add page info
ALTERNATIVE FLOW	The system will show him a message “page added successfully ”
POST CONDITION	system will add the page to his account

USE CASE	SHARE
ACTOR	member
PRE- CONDITION	He must have an account and be logged in
BASIC FLOW	First the member click on share button on the post he wants to share
ALTERNATIVE FLOW	The content won't be shared in the user profile
POST CONDITION	The content will be shared successfully

USE CASE	RETRIEVE PASSWORD
ACTOR	member
PRE -CONDITION	He must have an account
BASIC FLOW	First the user will try to log in if he failed he will click on forget password
ALTERNATIVE FLOW	The system will crashes
POST CONDITION	system will sent him a confirmation message on his mail

USE CASE	UPDATE PROFILE
ACTOR	member
PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view profile button then click on update profile button
ALTERNATIVE FLOW	The system show the user an error message
POST CONDITION	The profile must be updated successfully

USE CASE	ACCEPT FRIEND
ACTOR	member
PRE -CONDITION	He must have an account and be logged in and he must get an accept friend notification
BASIC FLOW	The user will click on accept button
ALTERNATIVE FLOW	The system won't add the user in his friends list the request won't be dismissed from the notification list
POST CONDITION	The accepted user will be added in his friends list

USE CASE	VIEW NOTIFICATION
ACTOR	member
PRE -CONDITION	He must have an account and be logged in and he must get a notification alert
BASIC FLOW	First the user click on view notification

	button then list will be shown
ALTERNATIVE FLOW	The system won't response to his request
POST CONDITION	The notification list will be shown to the user

USE CASE	**UPLOAD PHOTO**
ACTOR	member
PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view profile button and must view his profile and then update profile then click upload photo button
ALTERNATIVE FLOW	The photo won't be uploaded
POST CONDITION	The photo will be added

USE CASE	**SHOW LIKES**
ACTOR	Member, admin
PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view post then he will click on view likes button
ALTERNATIVE FLOW	The likes number will not appear
POST CONDITION	The likes number will appear

USE CASE	**SHOW COMMENTS**
ACTOR	Member, admin
PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view post then he will click on view comments button
ALTERNATIVE FLOW	The comments will not appear
POST CONDITION	The comments will appear

USE CASE	**SHOW SHARES**
ACTOR	Member, admin
PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view post then he will click on view shares button
ALTERNATIVE FLOW	The shares number will not appear
POST CONDITION	The shares number will appear

USE CASE	**SHOW FOLLOWING**
ACTOR	Member, admin
PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	First the user click on view following list
ALTERNATIVE FLOW	
POST CONDITION	The following list will not appear

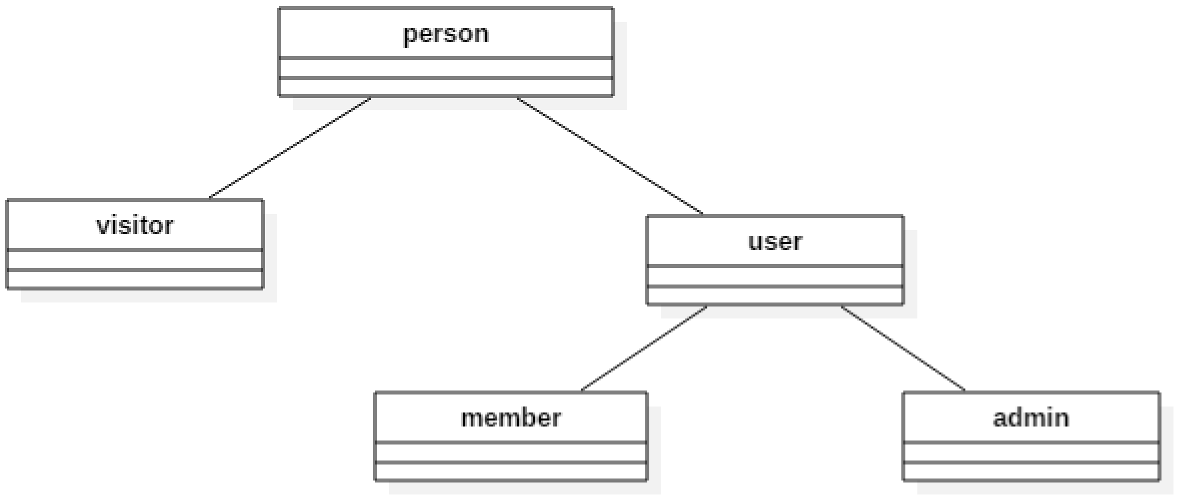
USE CASE	**SHOW OWN PROFILE**
ACTOR	Member

PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	the user click on button “profile”
ALTERNATIVE FLOW	
POST CONDITION	The profile page will appear

USE CASE	**SHOW FRIEND REQUEST**
ACTOR	Member
PRE -CONDITION	He must have an account and be logged in
BASIC FLOW	the user click on friends request button
ALTERNATIVE FLOW	No friend requests.
POST CONDITION	Friend requests will appear

Preliminary Object-Oriented Domain Analysis.

Inheritance Relationships



List of Super classes:

Person class
user class

List of Sub classes:

User class
visitor class
member class
admin class

Class name

1. Class Person
2. Class Post
3. Class User
4. Class Profile
5. Class Member
6. Class Friend
7. Class Admin
8. Class Validation
9. Class Visitor
10. Class Report
11. Class Data Base

Purpose:

1. Class person
   Include two function view profile and view posts that is help sub class that inherit from it to view their profile and posts

2. Class user
   It helps to represent the information of the user like name, email and password
   It can also make the user log in, search, add or remove posts and log out too

3. Class visitor
   To help the new visitor to registration for the first time

4. Class Admin
   It controls the application by adding or removing posts and accounts

5. Class member
   It helps the members of the application to like, comment, share, adding or removing posts
   And also following, adding friends and make communication with other friends

6. Class message
   To make communication more easy to our members they can also send and receive messages from other friends and followers

7. Class friend
   To view friends list and adding or removing friends easily

8. Class post
   To help members to like comment and sharing posts easily

9. Class report
   To send reports to the admin about whole the application for more control to the application

10. Class validation
    To handle all of errors it can happens in the application to ignore errors and make the application more safety

11. Class Data Base
    To storage and save all of data in the application for access and return the data at any time you need.

Attributes

Class name	Attribute
User	User name / email / password
Profile	Jobs /interest / education / skills
Member	Age /profile / posts
Post	Like / comment / share
Report	Date / content

References

The whole idea was adapted from the idea of LinkedIn.com
references:
Wikipedia.com

SDD

Introduction

Purpose

This SDD describes the scope and system design of “Wazifa” which is created in sake of making it easier to connect with other people works the same as you work or working in the same job. “Wazifa” will minimize the time and efforts in searching for companies or other workers.

scope

“Wazifa” is a business and employment-oriented social media service that works as web application that allows users to make friends with each other and post anything about jobs or advice to other employees.

Users can sign-up and make their profile as their CV they will add their skills past jobs, current jobs, education .
They can comment, like or share someone's post and they can chat with any user.
Users can add a page about the company they work in with logo, description, position.
Users can search about other users and companies and discover their profile or page.

Users will be able to update their profile info (CV)

an administrator will use web-portal to administrate the system and keep the information accurate. The admin can verify the company page and manage its information.

Out Scope

No job offers.

no bidding on jobs.

Not a freelancing application

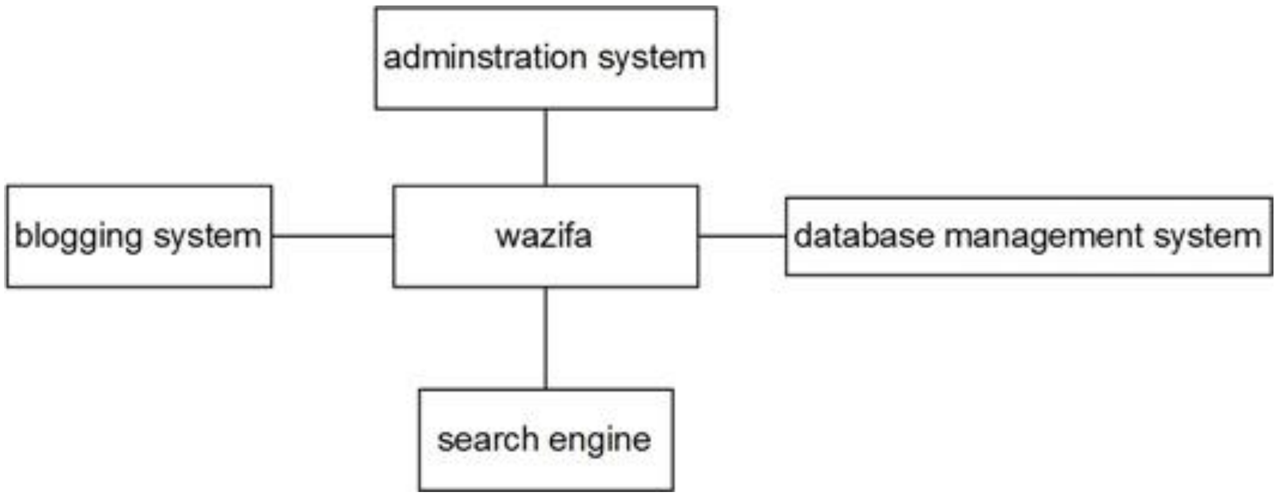
you can’t login by fake mail or password.

System Overview

“Wazifa” is built to make it easier to find other people works with you in the same company or works the same work as you are and make friends with them in order to learn from them and exchange your experiences.

System architecture

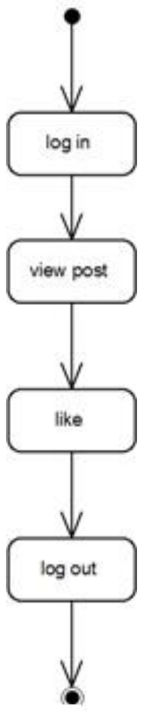
architecture design



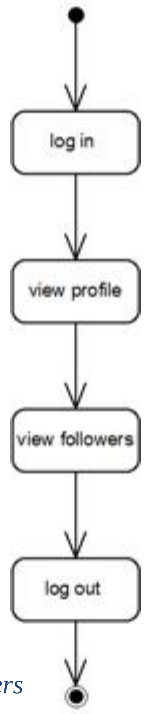
block Diagram 1

Decomposition Description

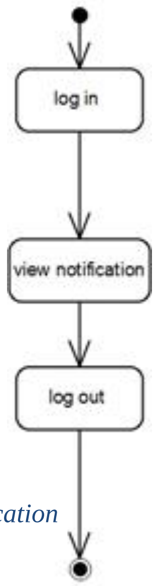
activity diagrams



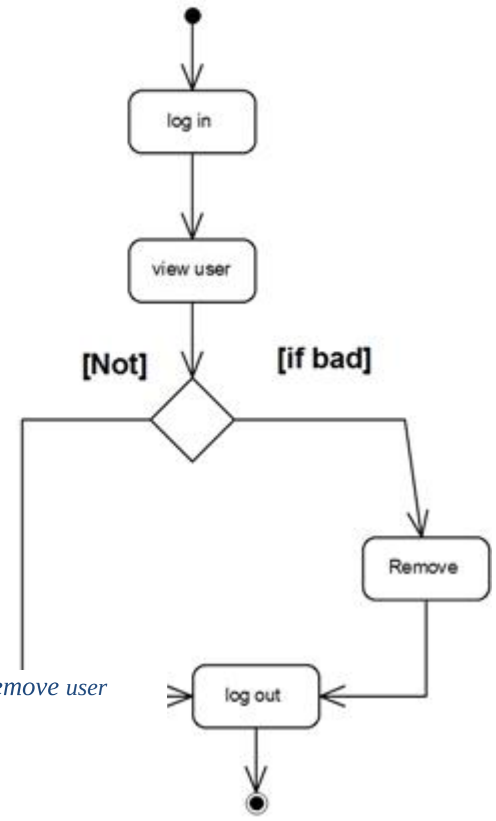
like post



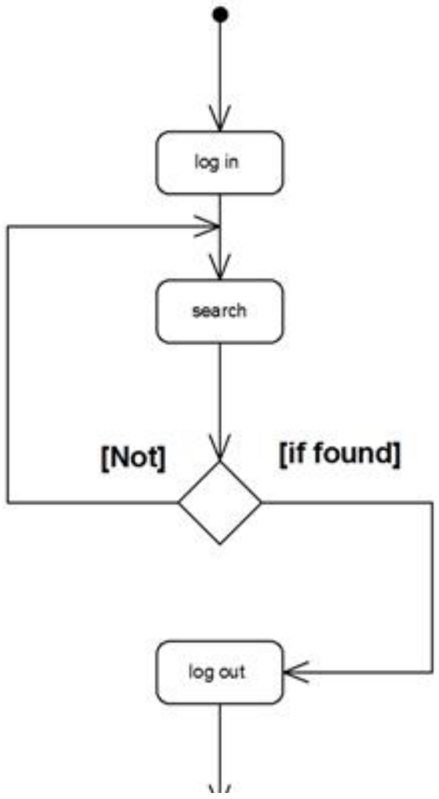
view followers



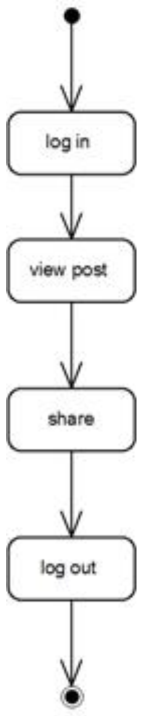
View notification



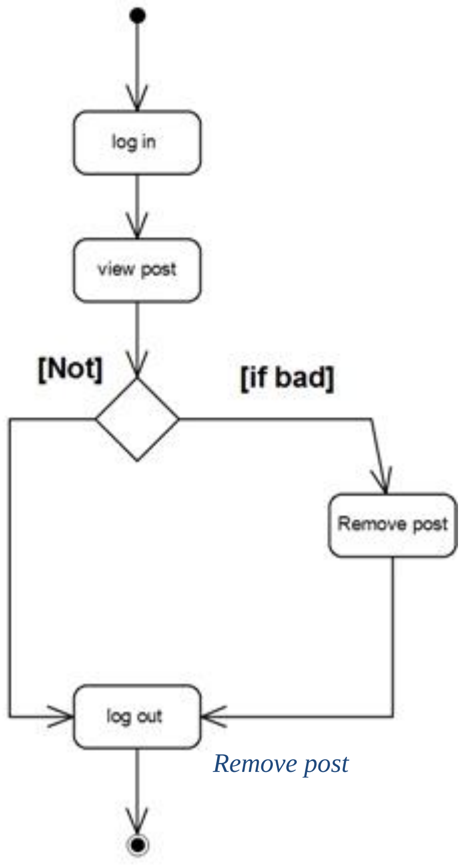
Remove user



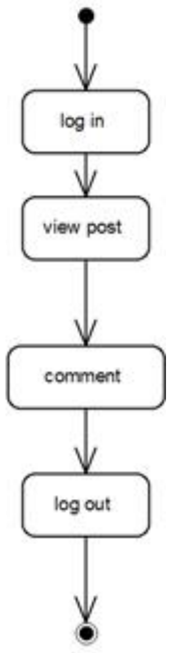
Search



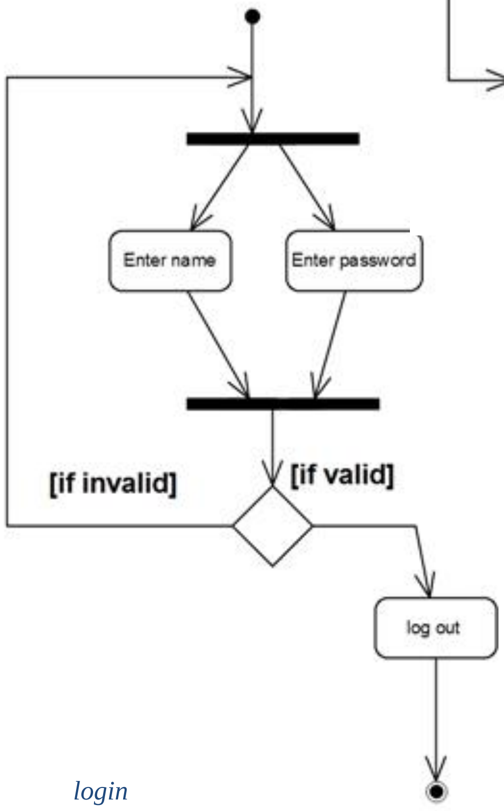
share



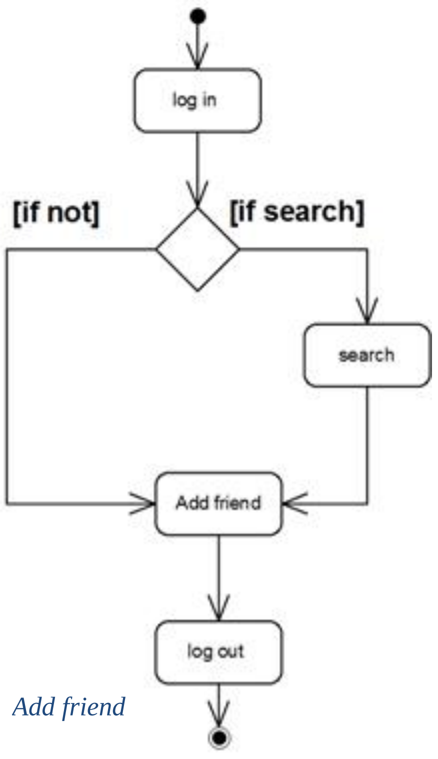
Remove post



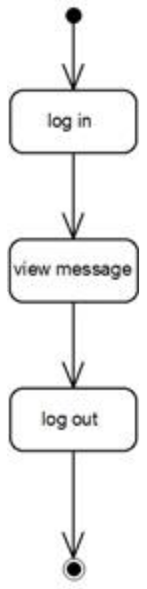
comment



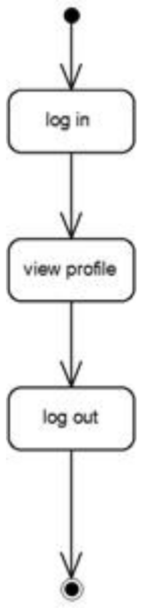
login



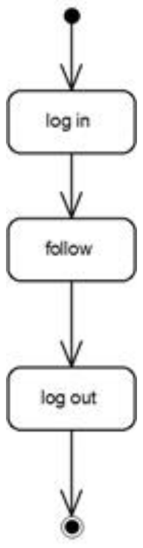
Add friend



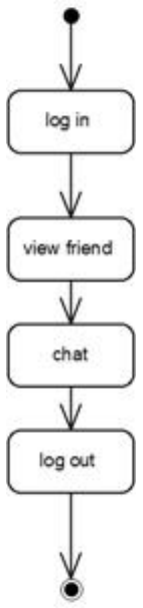
View message



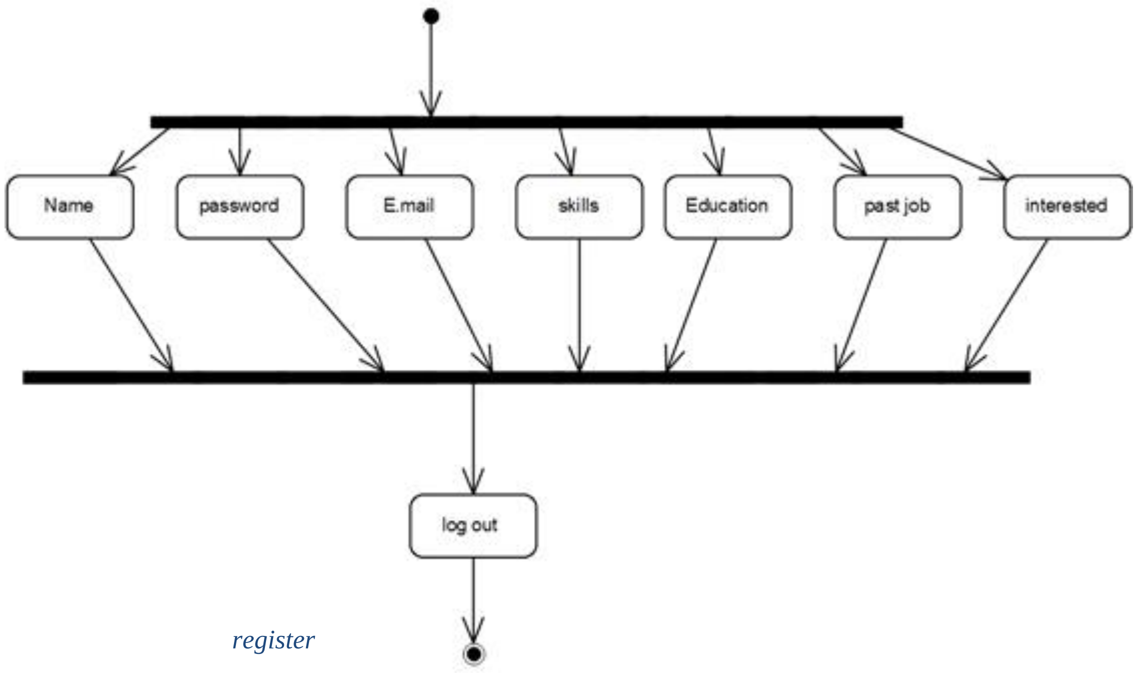
View profile



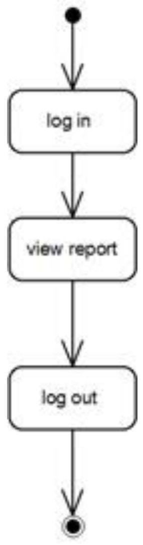
follow



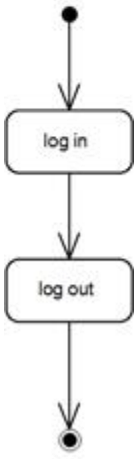
chat



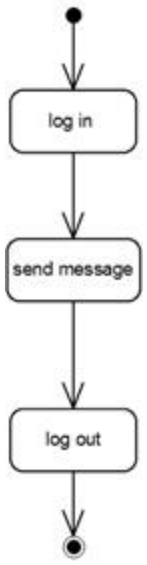
register



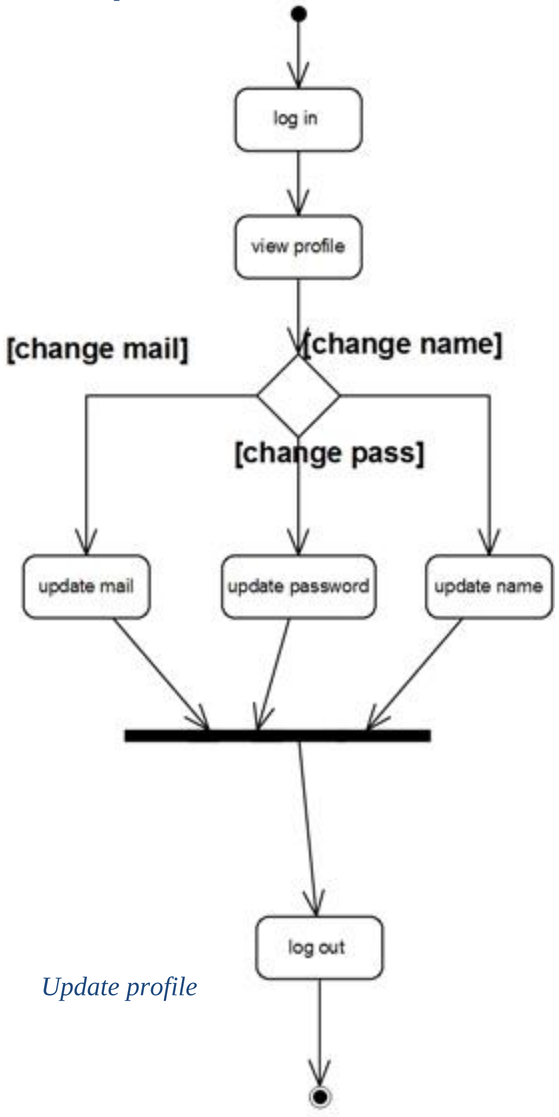
View report



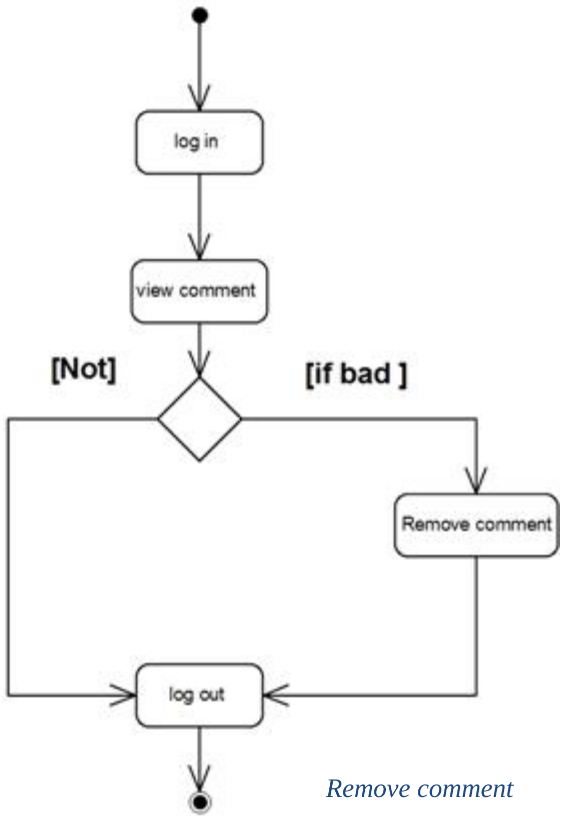
Log out



Send message



Update profile

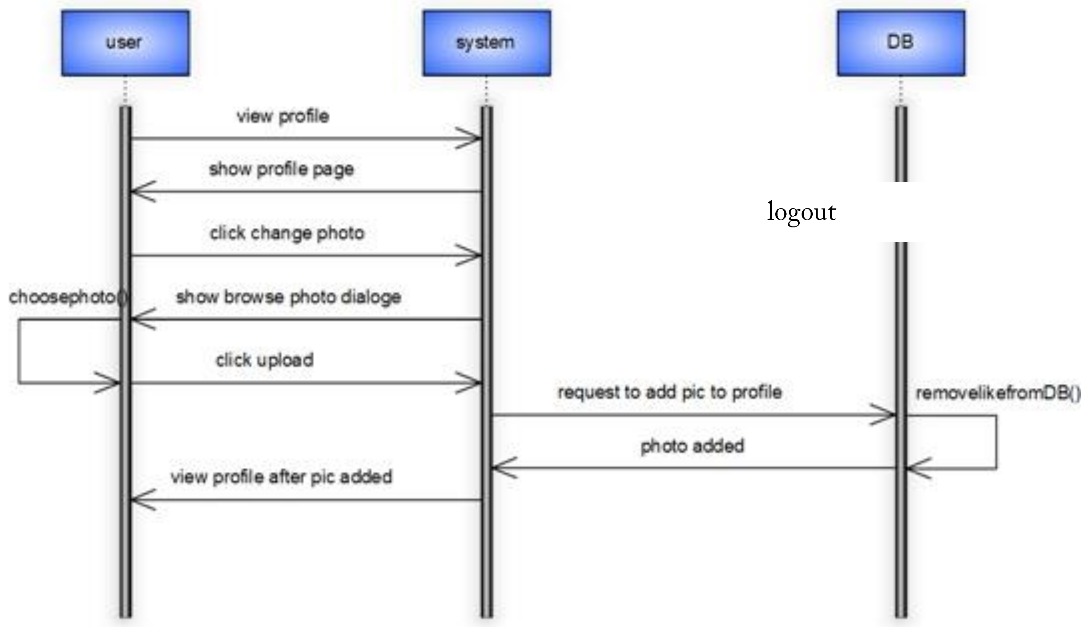


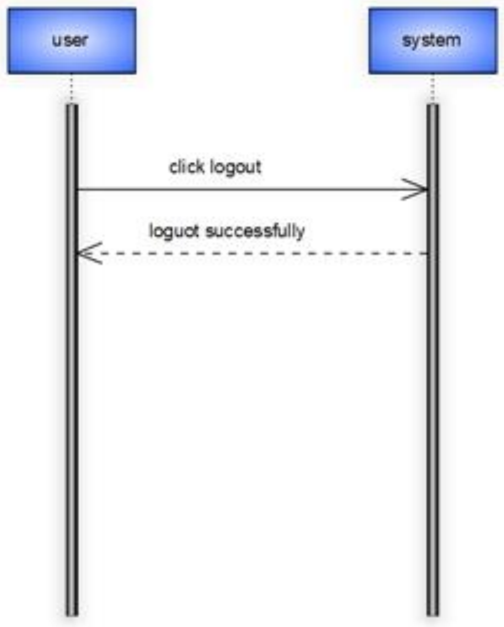
Remove comment

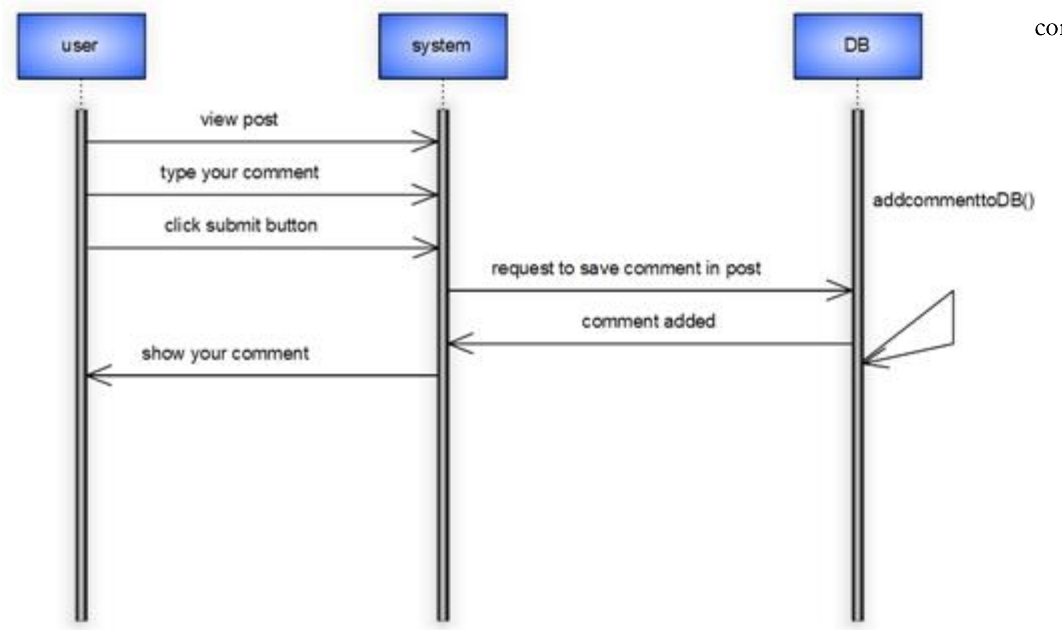
sequence diagrams

Upload photo

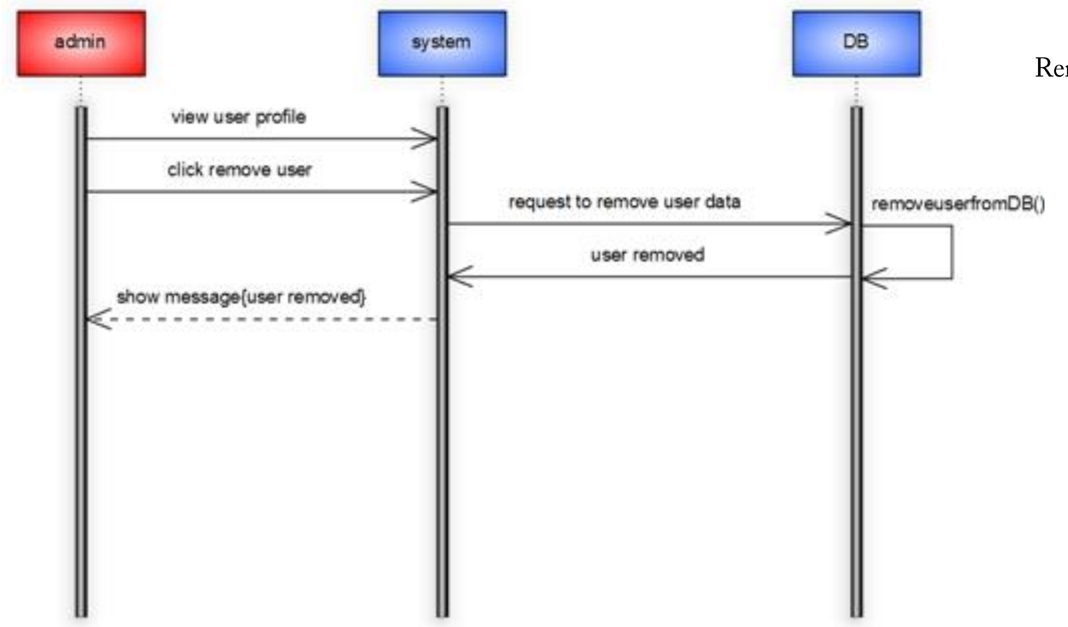
logout





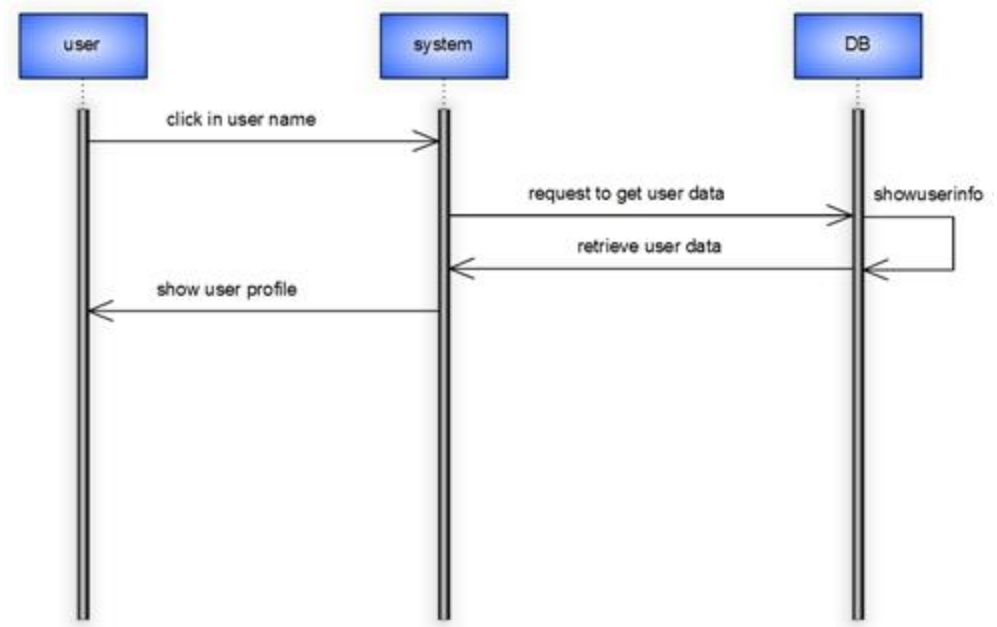


comment

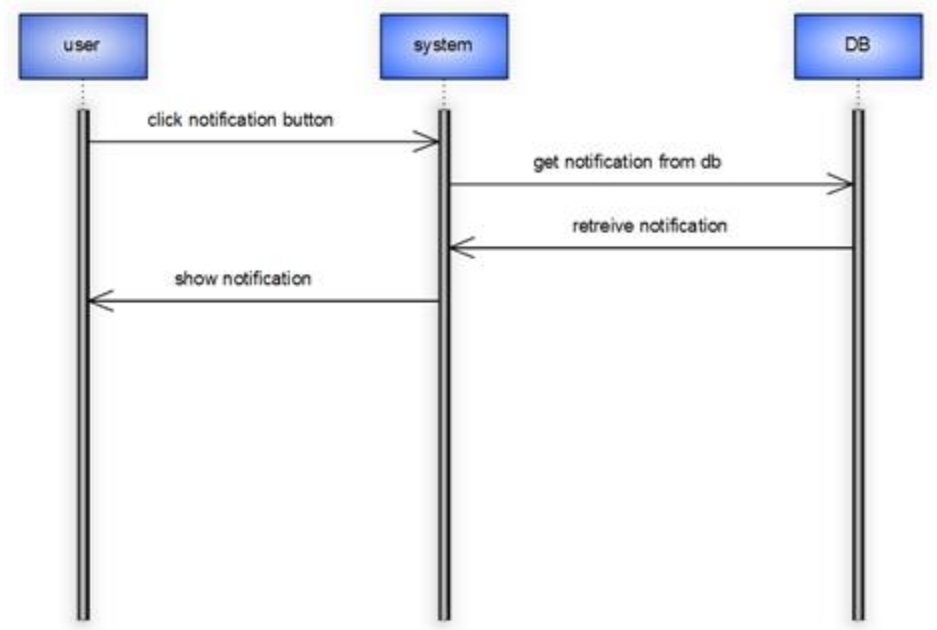


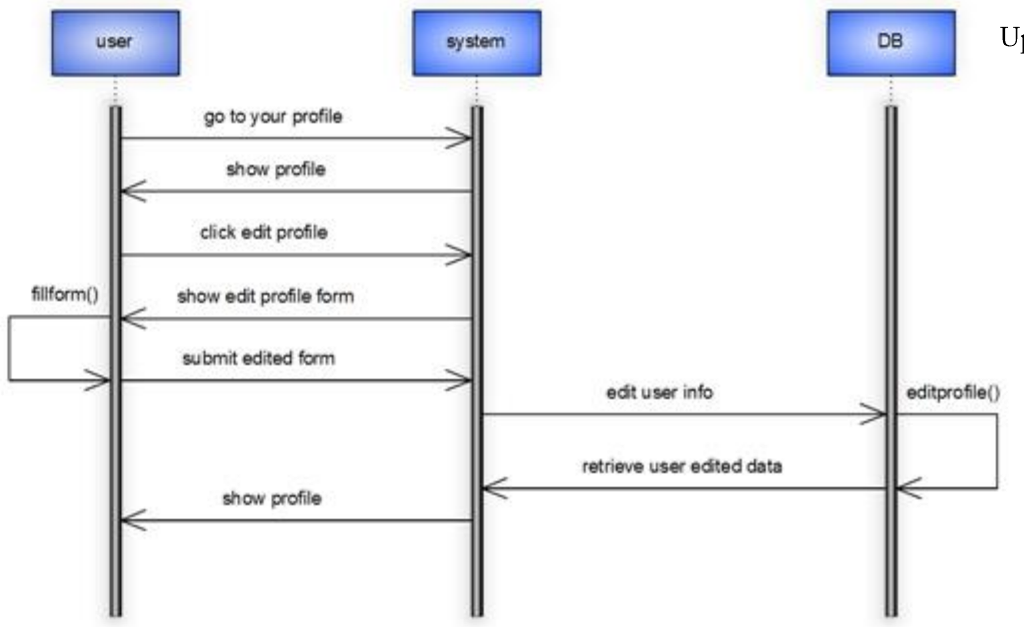
Remove user

Show user profile

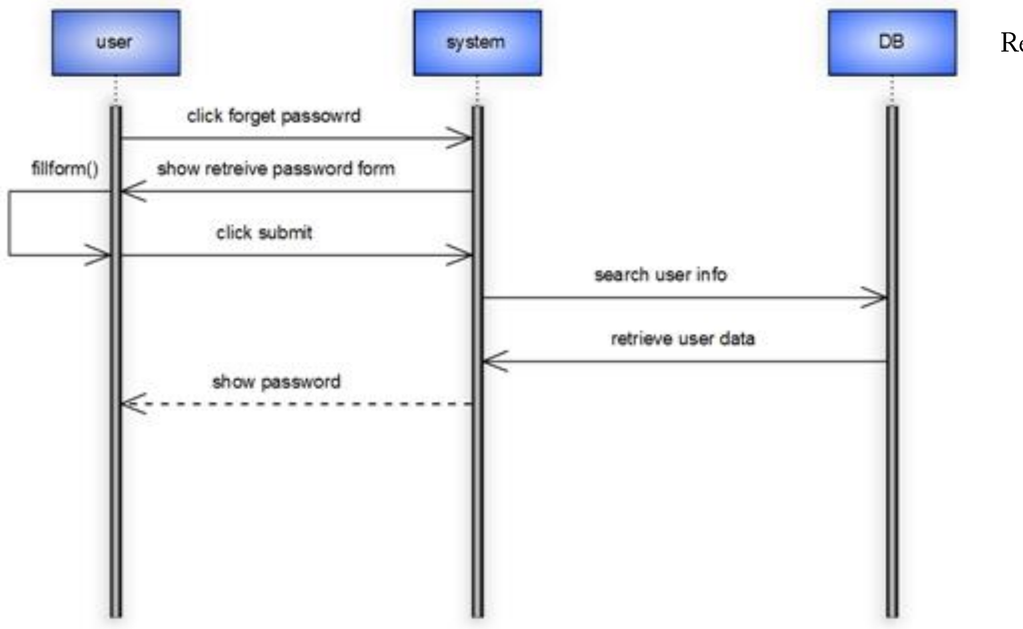


View notification

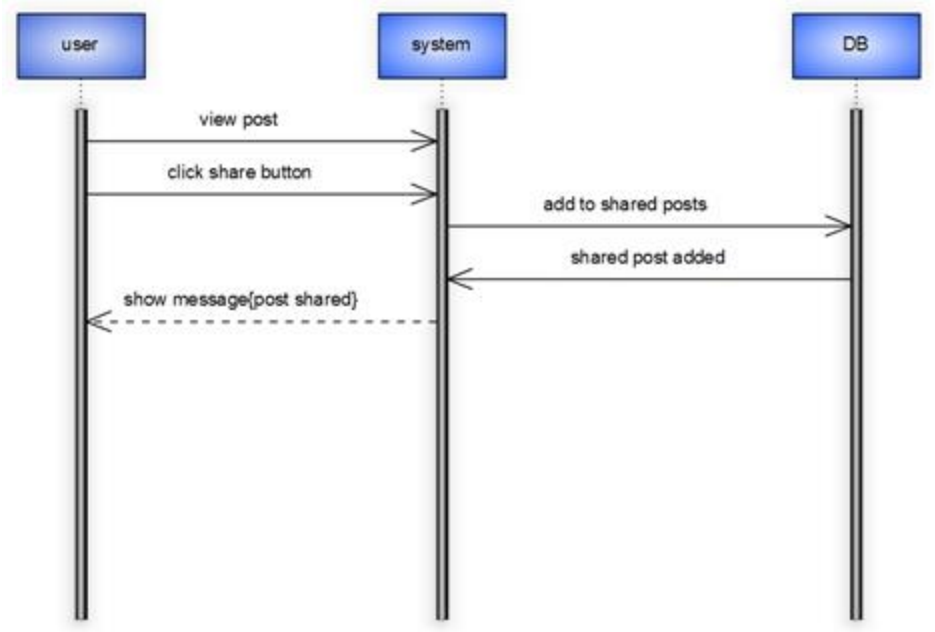




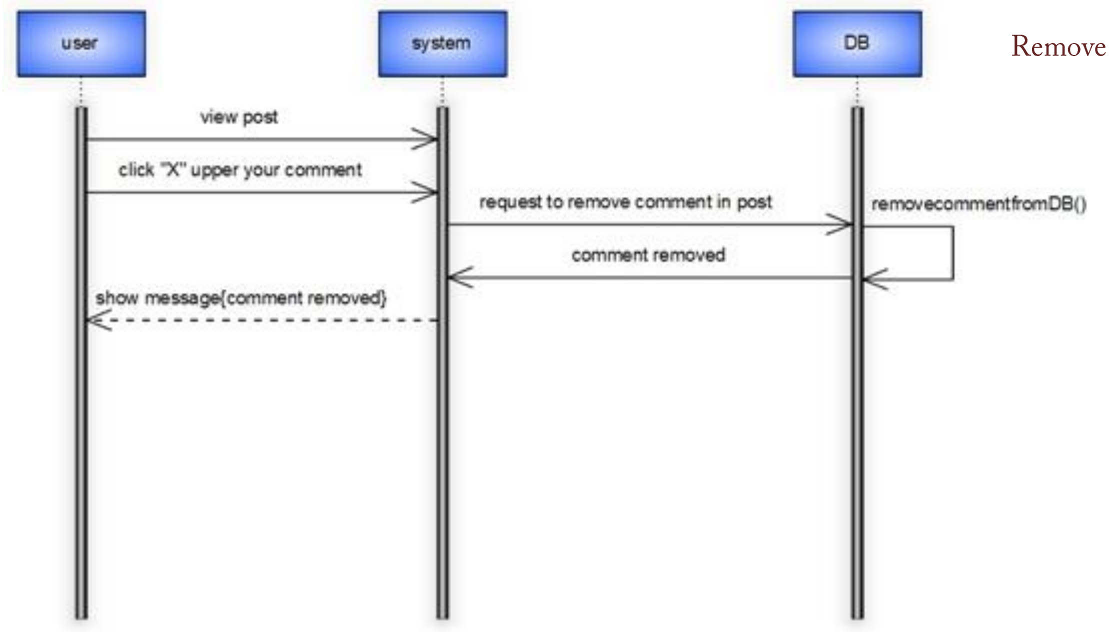
Update profile



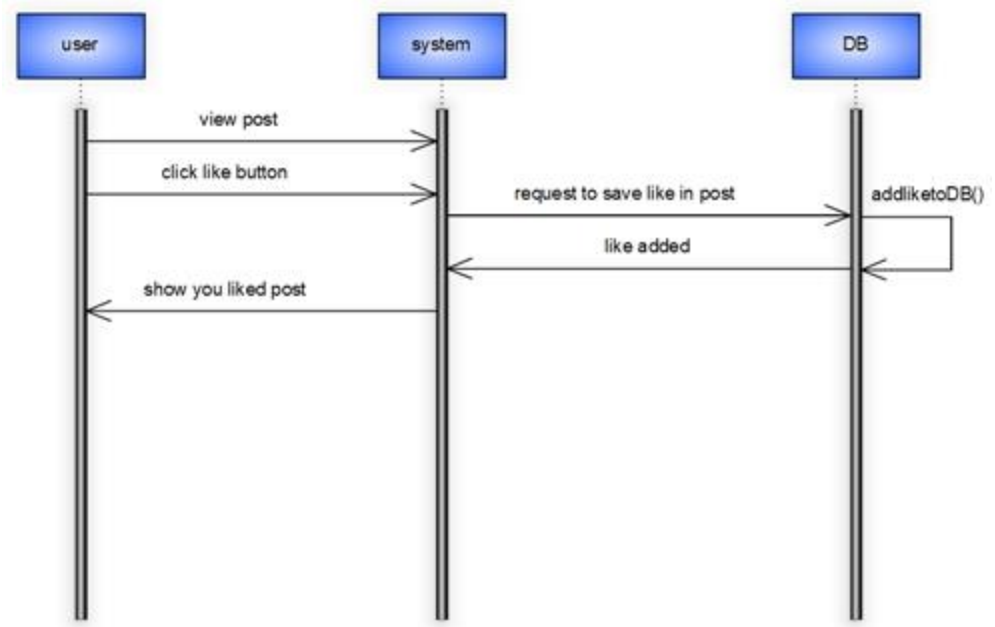
Retrieve password



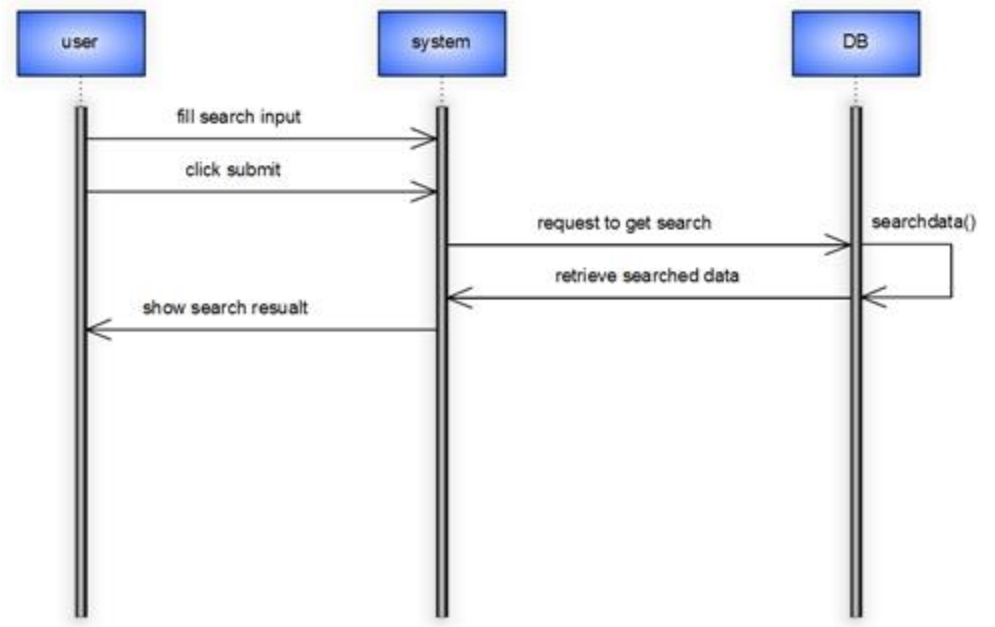
“Share”


user
system
DB
view post
click "X" upper your comment
request to remove comment in post
removecommentfromDB()
comment removed
show message{comment removed}

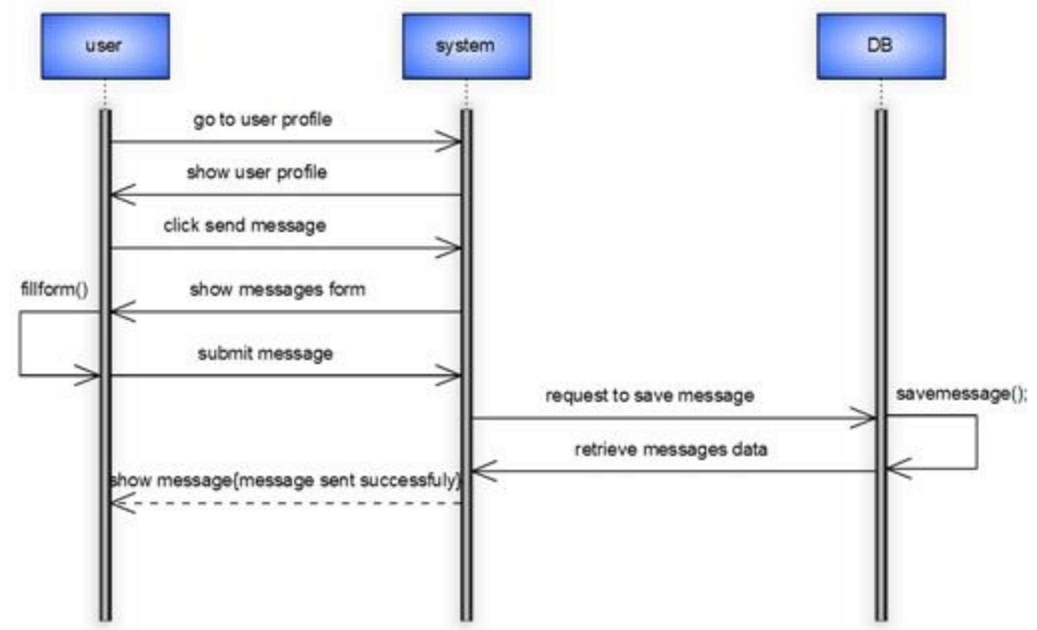
Remove comment



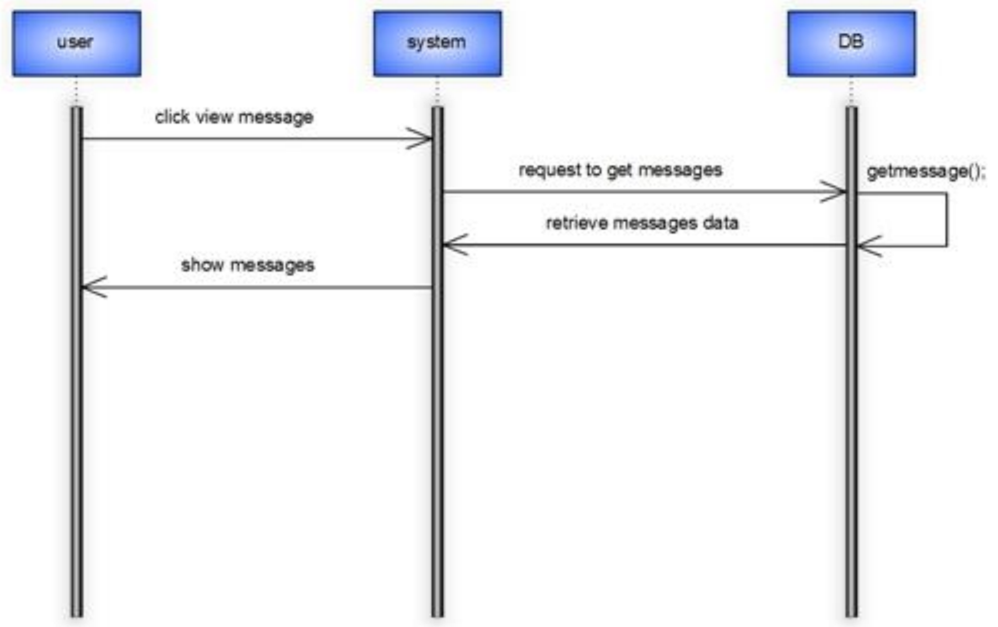
Like post



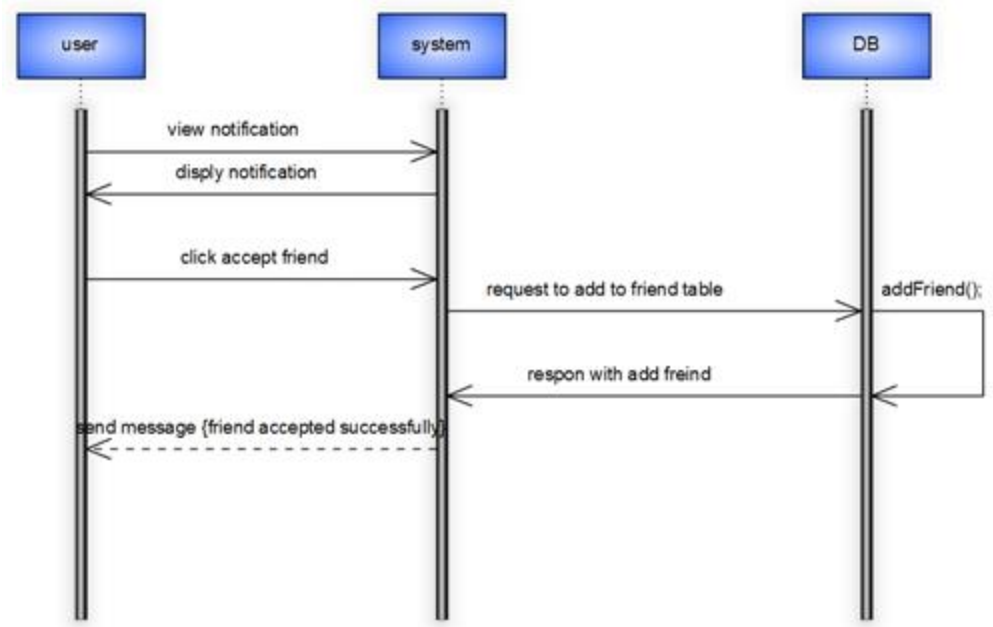
search



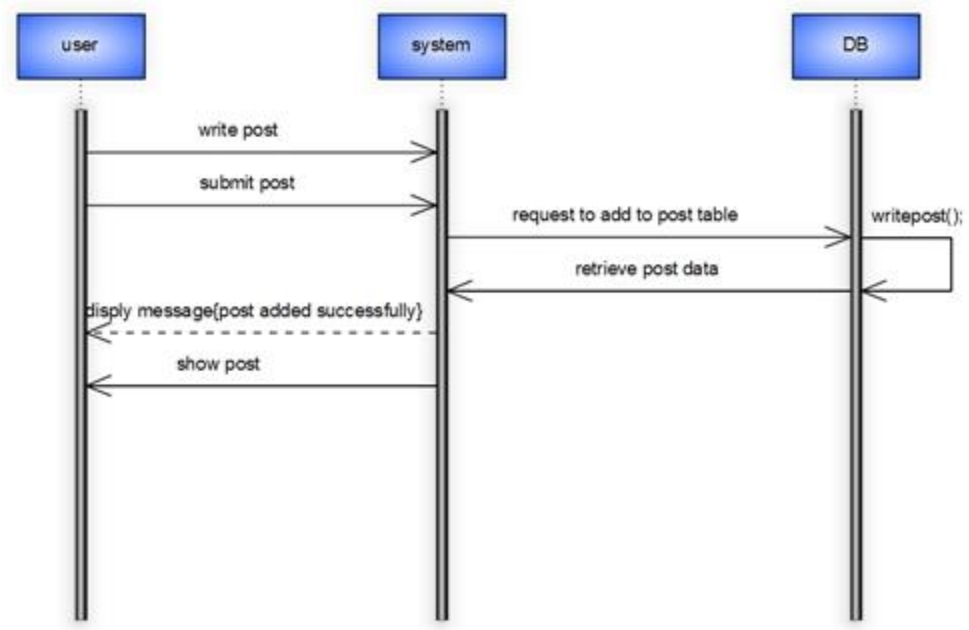
Send message



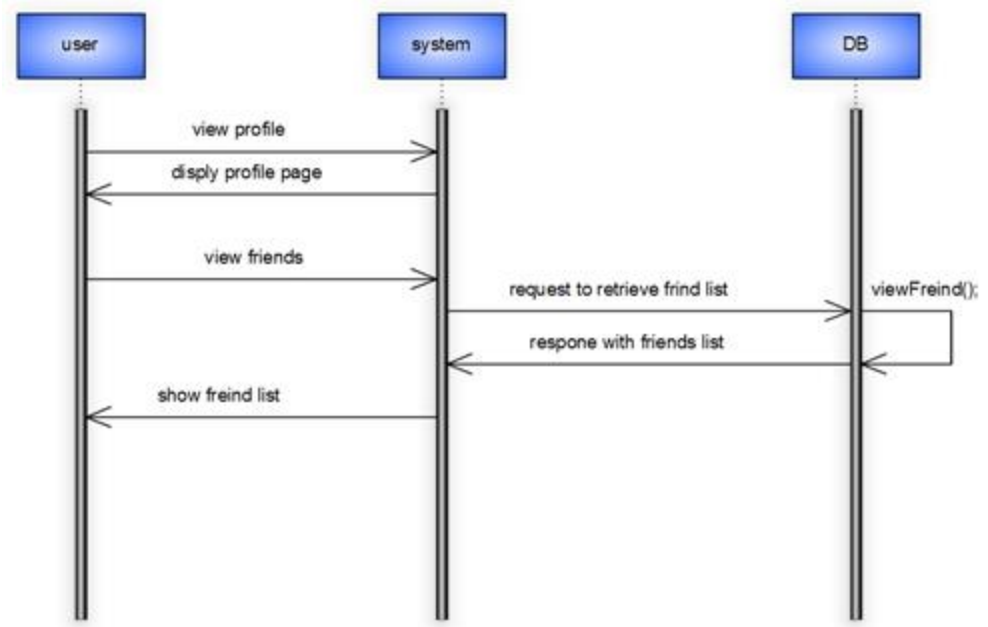
View message



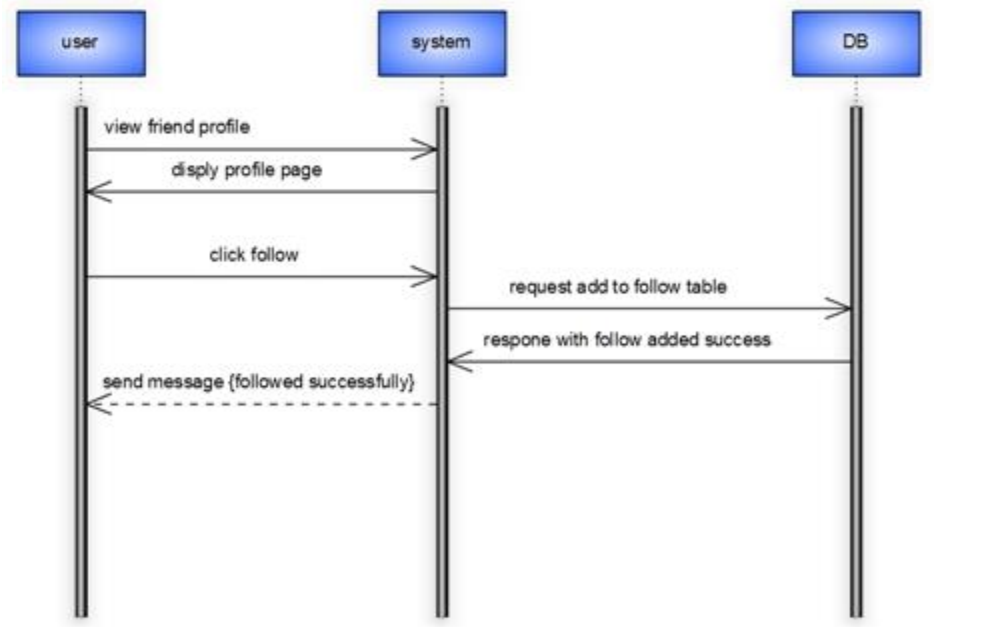
Accept friend



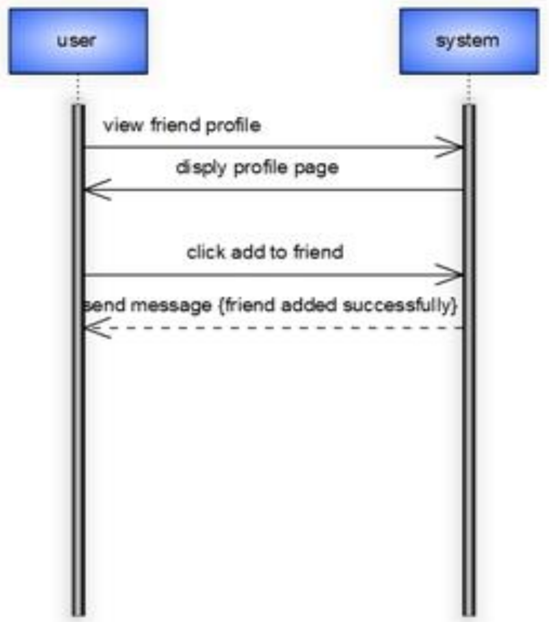
Write post



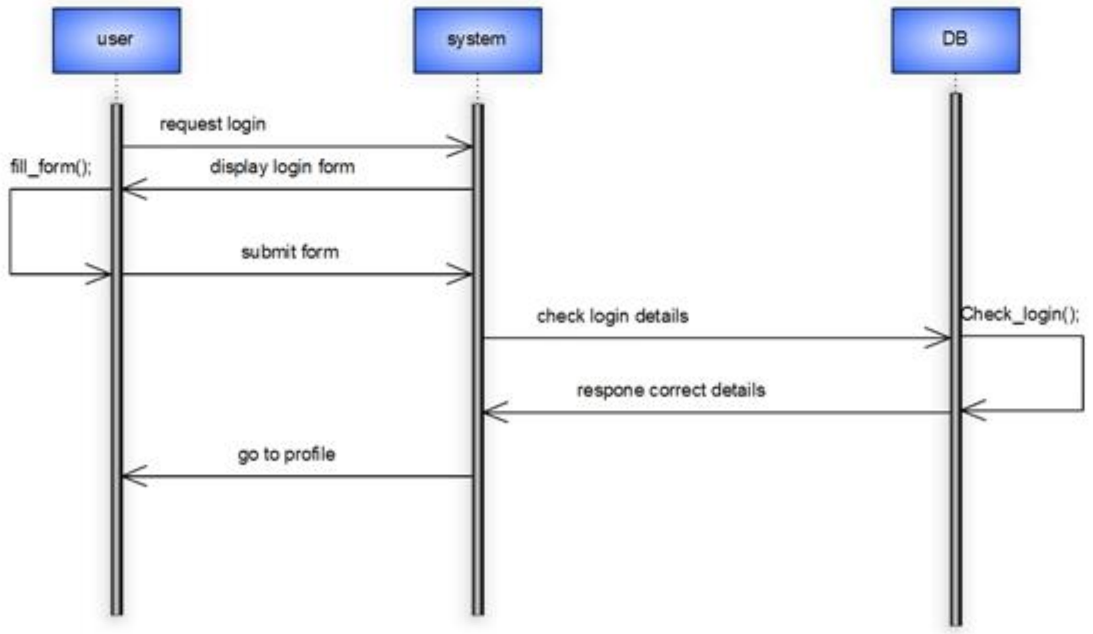
View friends


user
system
DB
view friend profile
disply profile page
click follow
request add to follow table
respone with follow added success
send message {followed successfully}

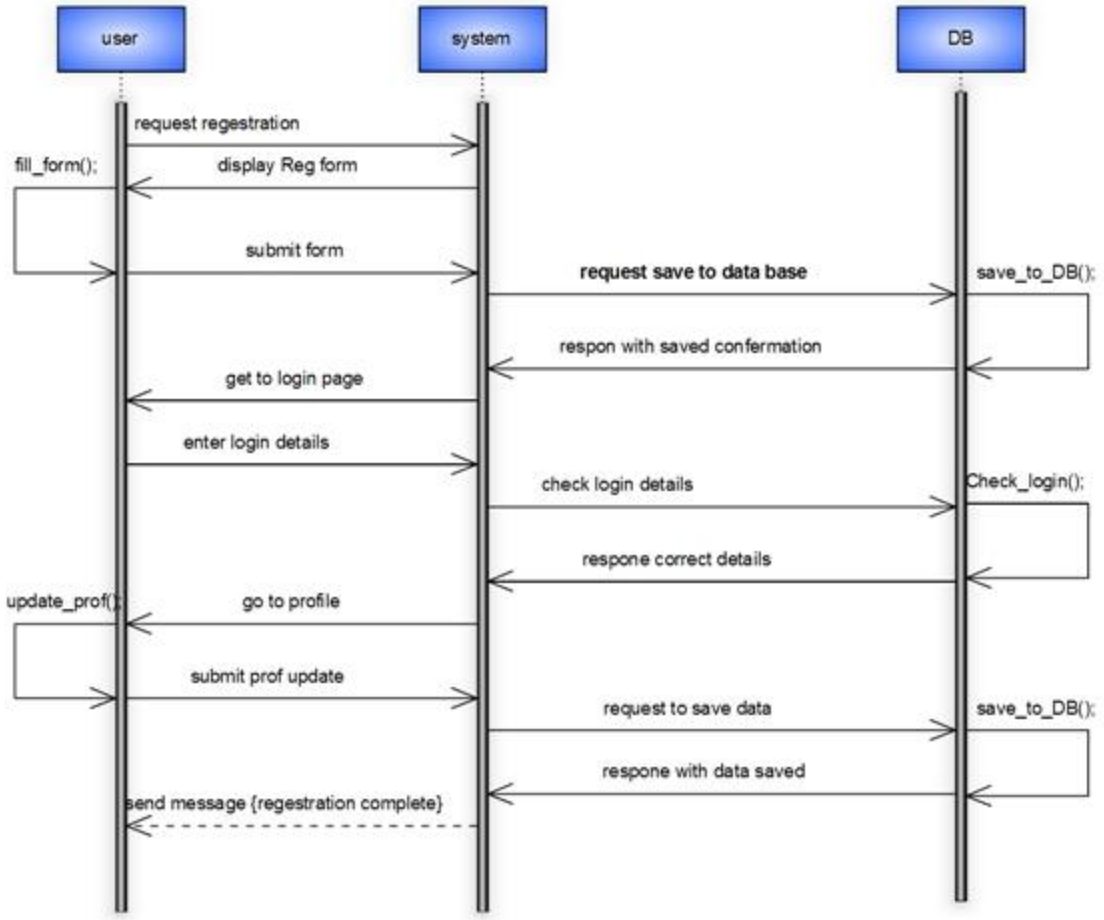
Follow



Add friend

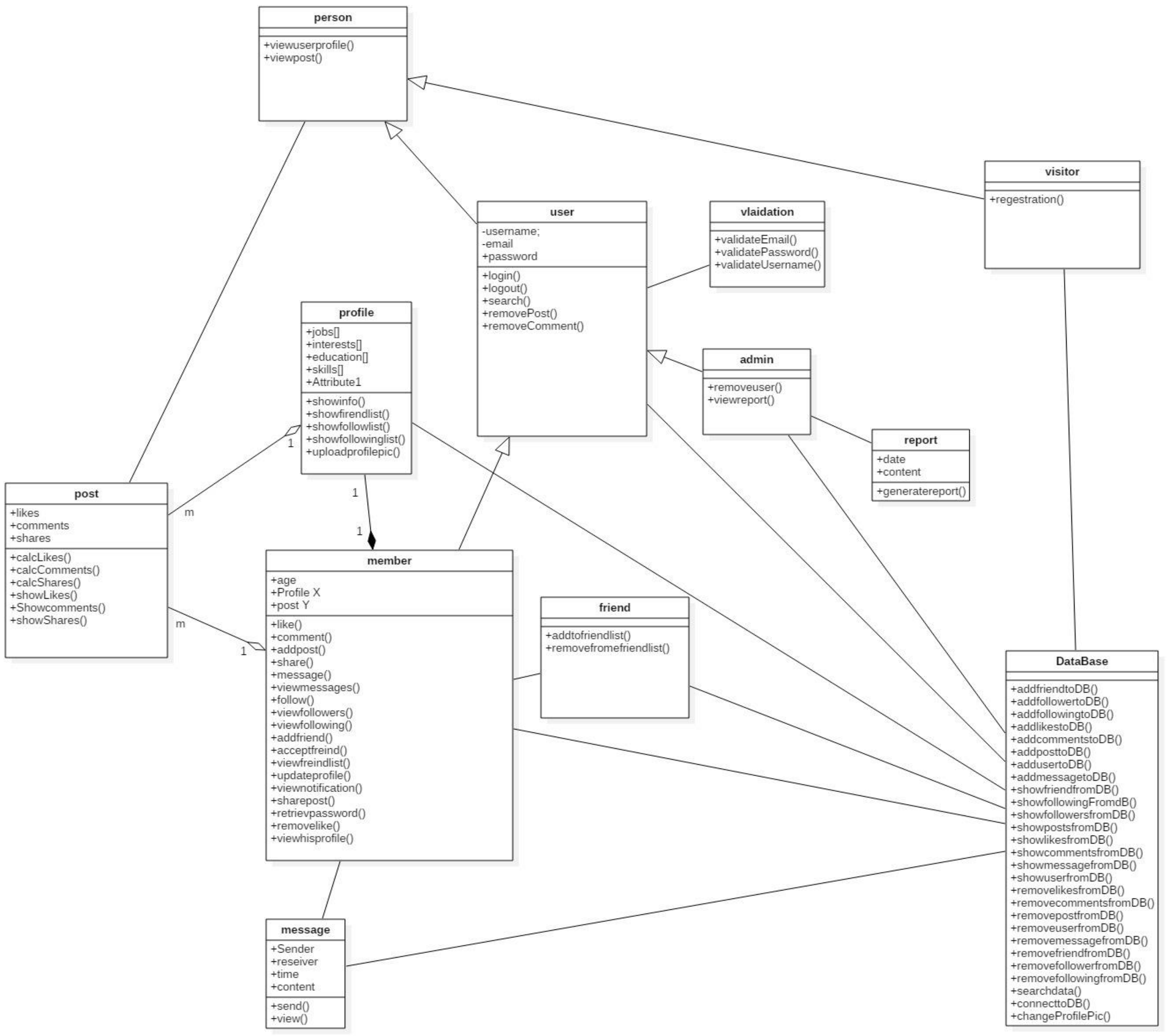


login



registration

Class diagram

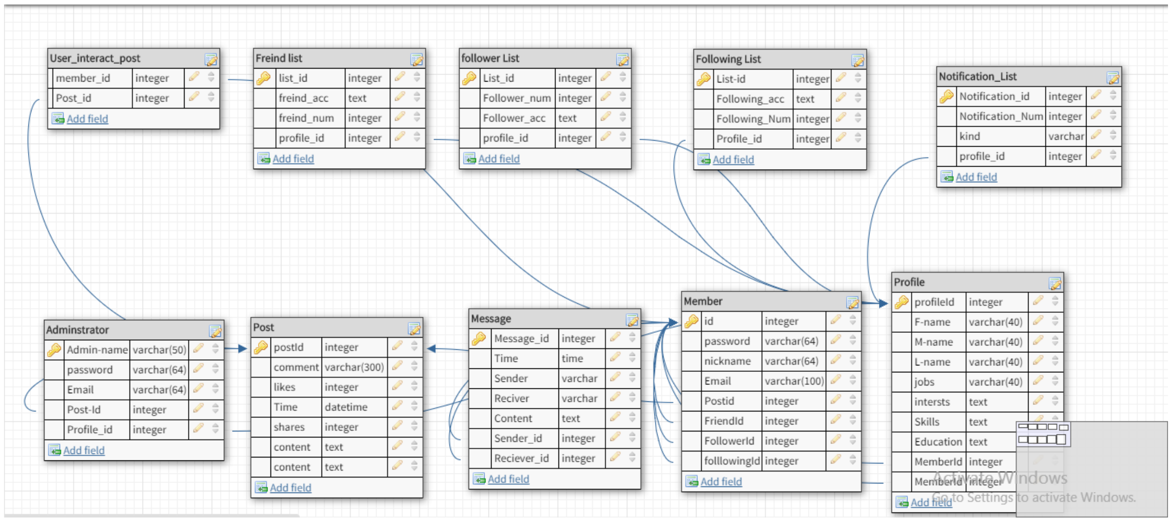


Data Design

Data Description

Data stored in database of the system, the stored data in the database like username of users, their password, skills, and their activities.

DB schema



User_interact_post	
member_id	integer
Post_id	integer

Freind list	
list_id	integer
freind_acc	text
freind_num	integer
profile_id	integer

follower List	
List_id	integer
Follower_num	integer
Follower_acc	text
profile_id	integer

Following List	
List-id	integer
Following_acc	text
Following_Num	integer
Profile_id	integer

Notification_List	
Notification_id	integer
Notification_Num	integer
kind	varchar
profile_id	integer

Adminstrator	
Admin-name	varchar(50)
password	varchar(64)
Email	varchar(64)
Post-Id	integer
Profile_id	integer

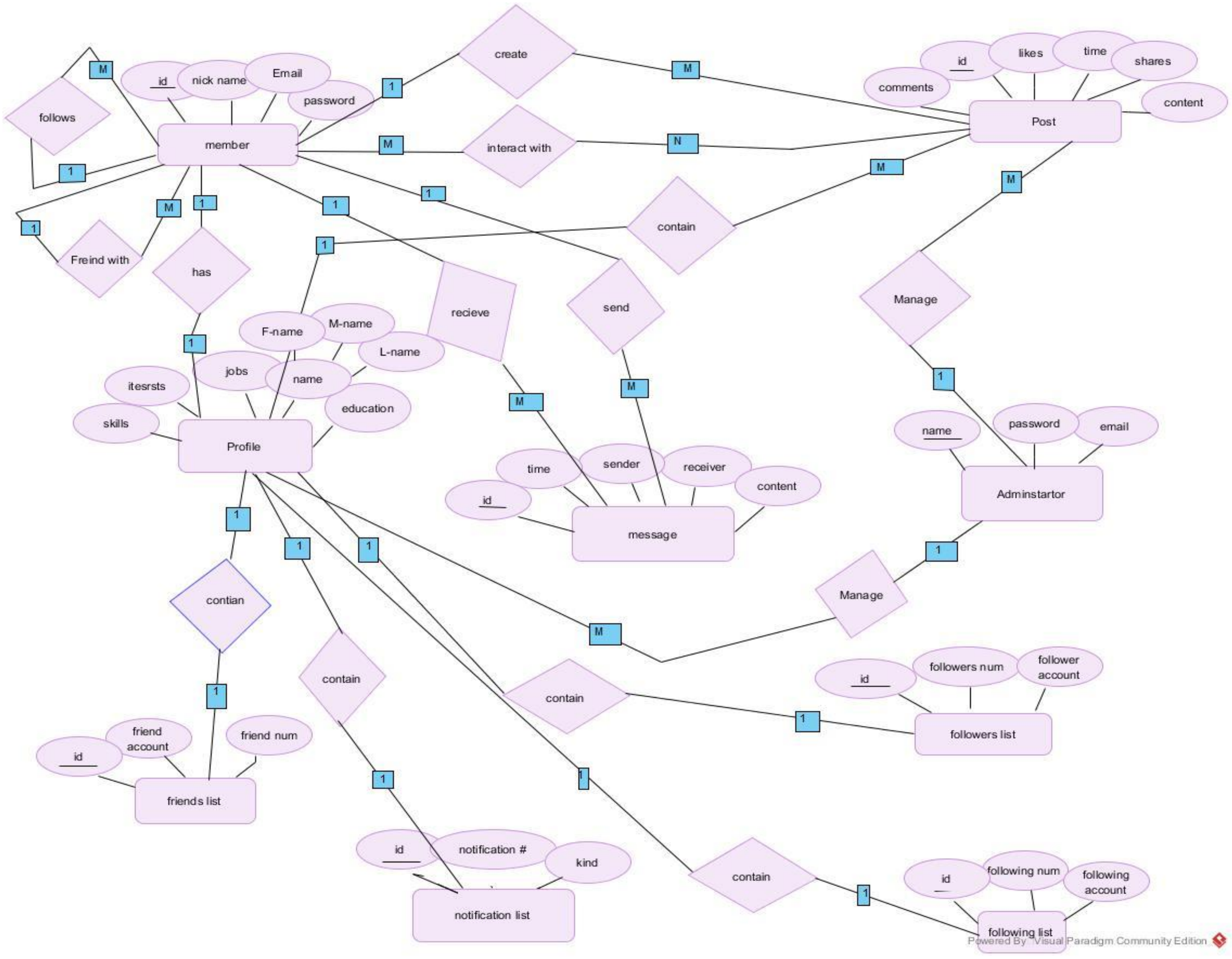
Post	
postId	integer
comment	varchar(300)
likes	integer
Time	datetime
shares	integer
content	text
content	text

Message	
Message_id	integer
Time	time
Sender	varchar
Reciver	varchar
Content	text
Sender_id	integer
Reciever_id	integer

Member	
id	integer
password	varchar(64)
nickname	varchar(64)
Email	varchar(100)
PostId	integer
FriendId	integer
FollowerId	integer
folllowingId	integer

Profile	
profileId	integer
F-name	varchar(40)
M-name	varchar(40)
L-name	varchar(40)
jobs	varchar(40)
intersts	text
Skills	text
Education	text
MemberId	integer
MemberId	integer

ER diagram

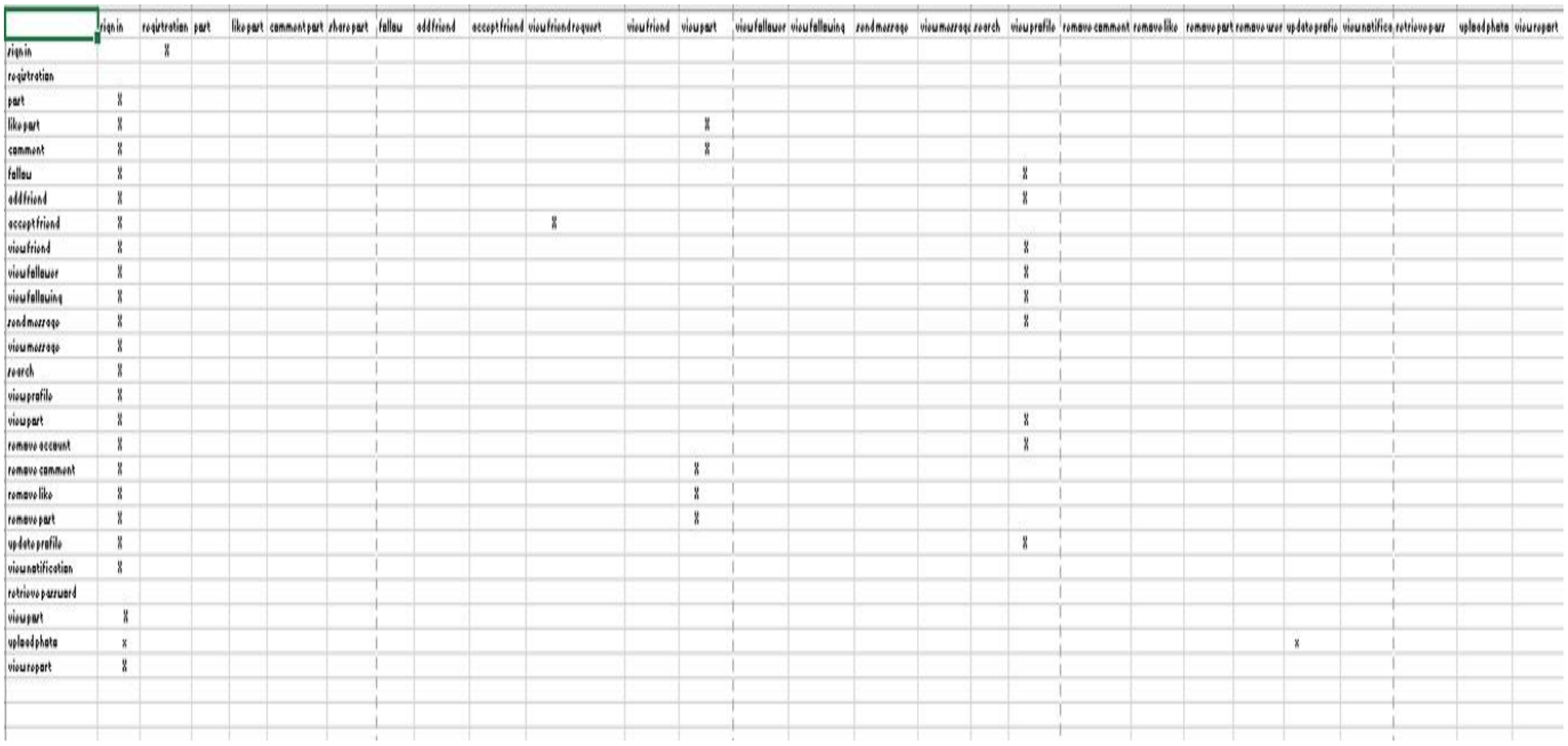

member
id
nick name
Email
password
follows
Freind with
has
create
interact with
contain
recieve
send
Post
id
likes
time
shares
comments
content
Manage
Adminstartor
name
password
email
message
time
sender
receiver
content
id
Profile
F-name
M-name
L-name
name
jobs
itesrsts
skills
education
contian
contain
contain
contain
Manage
friends list
id
friend account
friend num
notification list
id
notification #
kind
followers list
id
followers num
follower account
following list
id
following num
following account
M
N
1
Powered By Visual Paradigm Community Edition

Human Interface Design

Overview of User Interface

Any user in the system can do many functions by using the application like communication with other, searching for jobs, add more experience to him, and collect a lot of information about his career. user can easily do all of this function by making an account on the application and add his skills and career to make searching and communication mare easy to him.

Requirements Matrix



	signin	registration	post	likepost	commentpost	sharepost	follow	addfriend	acceptfriend	viewfriendrequest	viewfriend	viewpost	viewfollower	viewfollowing	sendmessage	viewmessage	search	viewprofile	remove comment	remove like	remove post	remove user	update profile	viewnotifica	retrieve pass	uploadphoto	viewreport
signin		X																									
registration																											
post	X																										
likepost	X											X															
comment	X											X															
follow	X																	X									
addfriend	X																	X									
acceptfriend	X									X																	
viewfriend	X																	X									
viewfollower	X																	X									
viewfollowing	X																	X									
sendmessage	X																	X									
viewmessage	X																										
search	X																										
viewprofile	X																										
viewpost	X																	X									
remove account	X																	X									
remove comment	X											X															
remove like	X											X															
remove post	X											X															
update profile	X																	X									
viewnotification	X																										
retrieve password																											
viewpost	X																										
uploadphoto	x																						x				
viewreport	X																										