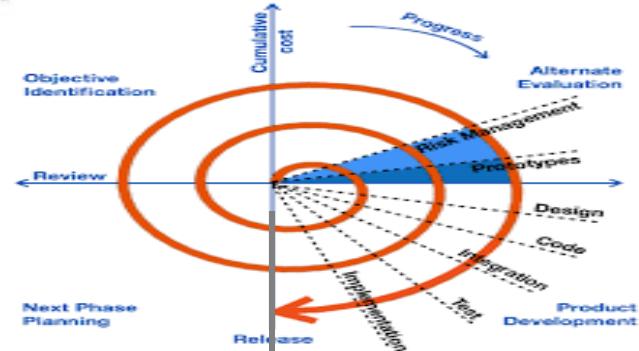




A PROJECT ON

SOCIAL NETWORKING SYSTEM

FOR FULFILLMENT OF 3rd sem UNDER OOD.



November 18
2016

We define social networking system as web based services that allow individuals to construct a public profile within a bounded system , articulate a list of other users with whom they share a connection , and view and traverse their list of connection and those made by others within the system the nature and nomenclature of these connections may vary from site to site

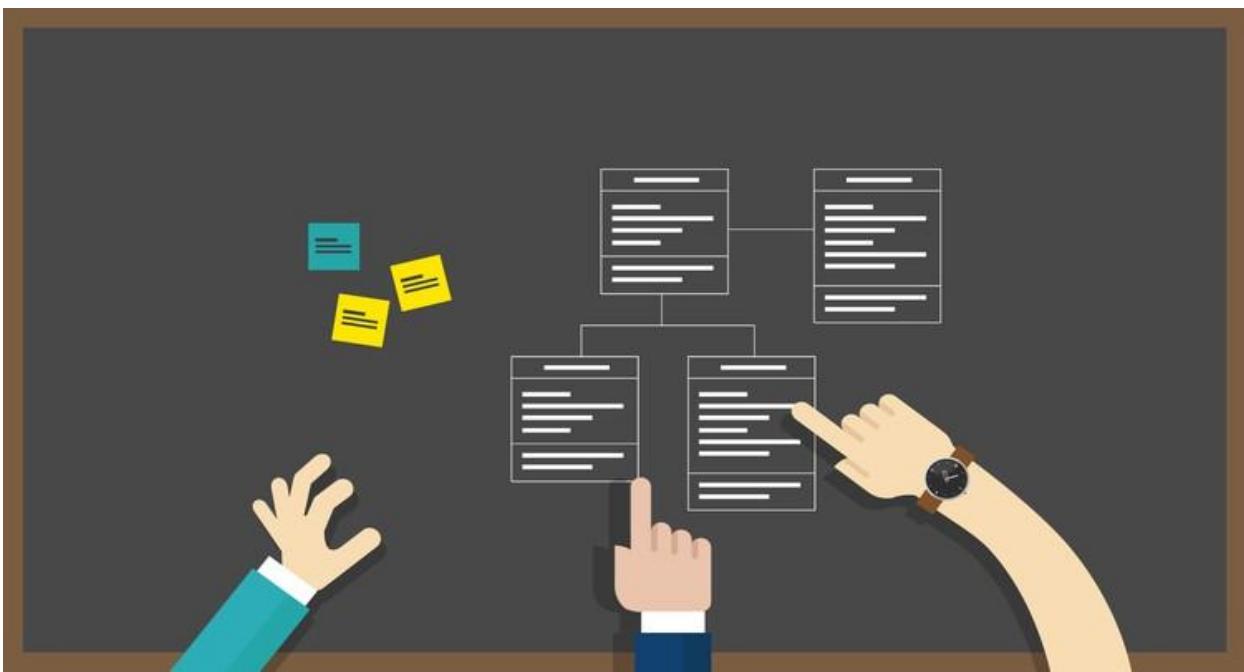
**OBJECT
ORIENTED
DESIGN**

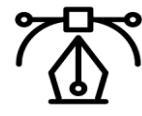
SUBMITTED BY:-

NIJWMSAN NARZARY(15-1-5-053)
PURUSHOTTAM GAUDEL(15-1-5-055)
SUBHAM KAMALAPURI(15-1-5-059)
APURBAJYOTI DAS(15-1-5-065)
SACHIN VUTUKURI(15-1-5-103)

PREFACE

A **social networking service** is a platform to build social networks or social relations among people who, for example, share interests, activities, backgrounds, or real-life connections. A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. Most social network services are web-based and provide means for users to interact over the Internet, such as e-mail and instant messaging. Online community services are sometimes considered as a social network service, though in a broader sense, social network service usually means an individual-centered service whereas online community services are group-centered. Social networking sites allow users to share ideas, pictures, posts, activities, events, and interests with people in their network.





Acknowledgement

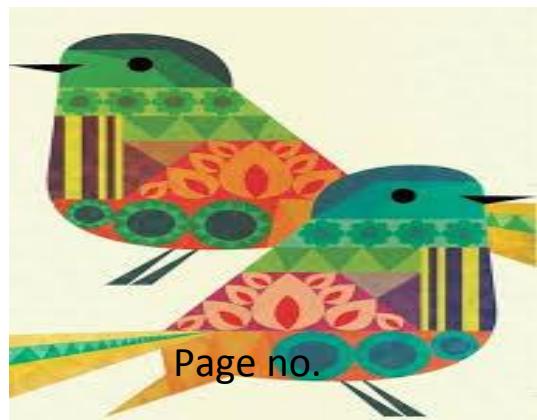


We take this opportunity to express our gratitude to our most respected sir, **Mr. Dalton Meitei** . Associate professor of **COMPUTER SCIENCE AND ENGINEERING DEPARTMENT , NIT SILCHAR** for his support and encouragement in bringing up the successful completion of this project named “SOCIAL NETWORKING SYSTEM ”. Under object oriented design.

Also we would like extend our thanks to our valuable classmate for theirs co-operation and queries for which this project came out beautifully.

Finally we would like to thanks to all who are directly or indirectly related to this project.

CONTENT



Sl no.

Topic

Page no.

Sl no.	Topic	Page no.
01	Introduction	01
02	Problem statement	02
03	Project scope	03
04	Functional specification	04-06
	Server object	04
	Account object	04
	Privacy setting object	05
	Message object	05
	Wall post object	06
05	Account creation	07
06	Privacy option	07
07	Chatting	08
08	Messaging	09
09	Events	10
10	Media uploading	11
11	Class & object diagram	12
12	Non functional specification	13
13	Use-case diagram	14-19
14	State transition diagram	20
15	Sequential diagram	21
16	Collaboration diagram	23
17	Activity diagram	24
18	Conclusion	25

INTRODUCTION

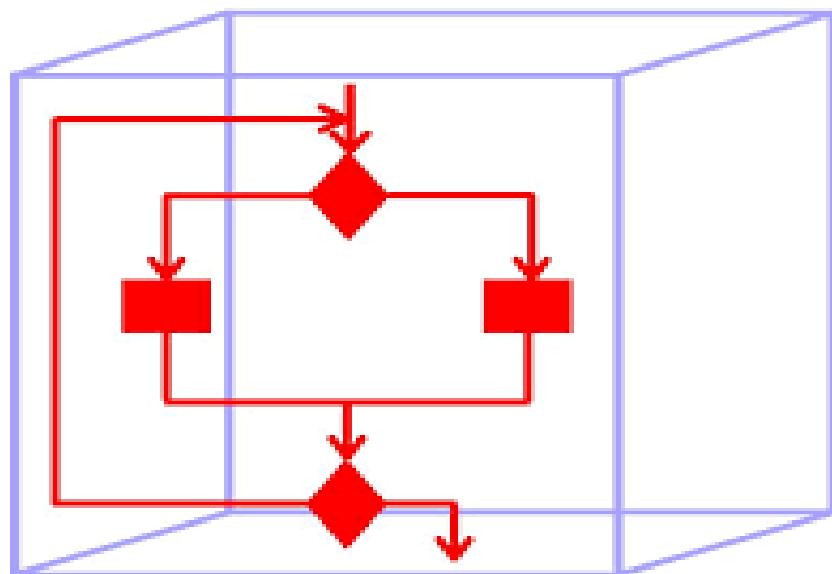
Social Networking – It's the way the 21st century communicate now . Social networking is the grouping of individual into specific group, like small rural communities or neighborhood subdivision . Although social networking is possible in person equally in the workplace , Universities and High school , it is most popular in online . This is because unlike most high schools , colleges or workplaces , the internet is filled with millions of individual who are looking to meet other people .

Social networking is the mapping and measuring of relationships and flows between people , groups, organization ,computers , URLs and other connected information/knowledge entities . The nodes in the network are the people and groups while the links shows relationships or flow between the nodes. Social network provides both visual and mathematical analysis of human relationship.

Social networking websites project itself is a huge project comprising various features like profile updating friends list organization and various other application to enhance the overall look and feel of the website.

PROBLEM STATEMENT:-

We define social networking sites as web based services that allow individuals to construct a public profile within a bounded system , articulate a list of other users with whom they share a connection , and view and traverse their list of connection and those made by others within the system the nature and nomenclature of these connections may vary from site to site .



PROJECT SCOPE :-

- This system provide users to register their various types of profile like social , personal , general , professional.
- This system provide users to send a scrape message images and data file to their friends.
- User can maintain the scrap book whatever scrap he has send to users .
- This system provide the user to maintain their friend list and user can update their friend list .

Functional Specifications :-

- Server Object**

The Server class acts as a wrapper for all server functions for our social networking site. It essentially act as a link between all of the information such as accounts, account details, pages, notes, etc to our database. When any other model object such as a page is pulled from the server, a temporary copy is made. If that temporary copy is changed in any way. The new version must be sent to the server in order to update the permanent copy. The reason behind local copies is that all the necessary information for the object is sent over in one easy-to-use package. Then the update to the database can be done all at once by sending back that single object. There is no need for multiple functions or a function that takes a large number of parameters.

- Account Object**

Each user who wants to use the site must create an account. This is the head class that all other objects use to determine what a user does and when the user did it. The account's information has four purposes: hold the login information, hold friend information, hold profile information, and hold privacy information with such a large amount of information to keep track of, the Account class would be very large and difficult to work with. Therefore to ease the load, the Account class was broken up into three different classes. There is the actual account class which keeps track of login information and friend information. It also holds the other two classes within it. Profile information was outsourced to the Account Details class, and privacy settings were outsourced to the Privacy Settings class. The only time

the Account class needs to be updated is when the user changes his/her username and/or password. All other settings are handled by the Account Details and Privacy Settings classes.

- **Account Details Object**

An Account Details object is a helper class created whenever a new Account object is created. The object contains all the information that shows up in the user's profile. The user can edit this by modifying his/her profile. Overall, this class has no other purpose but to be a helper class to its account object.

- **Privacy Settings Object**

A Privacy Settings object is the other helper class created whenever a new Account object is created. This object contains all the privacy settings that a user has, such as who can view his/her media or custom pages. This class is called any time a user visits a profile or content created by another user. However, it does not directly interact with the other model classes, only the view.

- **Chat Session Object, Event Object, Link Object, Note Object and Page Object**

These objects contain unique information for a particular type of action a user performs. All of these contain a reference to the account that owns them. Each object is a "working-copy" of an object in the Server. Anytime one of these objects is created on the Server, an entry of its creation is added to the news feed database.

- **Message Object**

A Message object is created when a user composes a new message to be sent to a friend. After it is confirmed that the friend is located in the database, the Message object adds its information to the database. When a user checks his or her inbox, a list of messages that were sent to the user will be shown in descending order of when they were received.

• Wall Post Object

A Wall Post object works very similar to a Message object. The only difference is that the Wall Post objects are viewable to all friends of the user. Another difference is that a Wall Post object can hold comments. If a user adds a wall post to his or her own account, their status will change to the new wall post.

• Friends

The most important feature is being able to add and remove friends. In our Social Networking site, making friends is a fairly straightforward process. Users can type in the name of a friend in the search bar at the top of their home page. The database is queried for an account that has the search term contained in the full name, any media files with the search terms in the description, any pages with the search terms in the titles, any links with the search terms

in the titles, and any notes with the search terms in the title. For example, User A could search for User B in the search bar. After clicking on User B's profile, User A will see a button that says Send Friend Request. Clicking on it will send an alert to User B that User A wants to be a friend. The friend request will now be in the

friends list of User B, where he/she can either accept it or ignore it, letting it sit there indefinitely. If user B accepts the request, User A will be added to User B's friends list and vice versa. Being friends has its advantages. For example, only friends can chat to each other. Also, friends can view any part of a profile that is marked as friends only. Finally, for a user to view his/her friends and incoming friend requests, he/she just clicks on the friends tab which brings up a frame.

• Account Creation

When a user accesses the site for the first time, he/she must create an account before using any of the site features. The account creation process is broken into three sections. The first section deals with the login information and is required for the user to fill out. This includes the email, password, and password confirmation. The purpose behind the password confirmation is to ensure that the user didn't accidentally mistype when creating a password. The second section deals with information about who you are such as name, location, and gender. Most of these fields are optional except for your name and gender. It wouldn't be much of a social network if everyone was named anonymous. The final section deals with information about the users likes and dislikes, such as interests and activities. Unlike the other two sections, this section is completely optional. Once the user clicks create account, a new account, account details, and privacy settings are added to the server, and the user is brought back to the login page

• Privacy Options

Privacy is very important feature for some people, and social networking is no exception to this. Our system provides three levels of privacy: open, friends only, and closed. A feature with an open privacy level is public and may be viewed by

anyone. By contrast, a feature with a closed privacy level is completely private and can only be viewed by the account owner. A feature with a friends only privacy level is fairly self explanatory. By default, all privacy levels are set to open when a new account is created. To change privacy levels, the user can click on the options tab on the top menu bar. The current features with privacy levels are media share, account wall, user pages, notes, and the overall profile privacy.

The implementation of privacy is a fairly straightforward process. When the user makes changes to the privacy level, the Privacy Settings object in the account is updated with the new privacy settings. That working copy is then passed into the update method of the server, and that queries the database to set the record for that privacy settings object to the new values. When a user visits another profile, the profile owner is pulled from the database. A function called canViewPage() is and is passed the profile owner, the viewer profile and the page type. In the function the privacy settings object is pulled from the viewer profile and the page type determines which privacy setting is pulled from the object. If the setting is open, the function automatically returns true. If the setting is friends only, the function only returns true if the viewer profile is a friend of the page owner or if the viewer profile and the page owner are the same. Finally, if the setting is closed, the function only returns true if the viewer profile is the page owner. Once the function returns its answer the page will either do one of two things. If the function returns true, the viewer is allowed to view this page, and it will load normally. If the function returns false, the viewer is not supposed to view the page, and it will redirect to an error page.

• Chatting

One of the advantages of having friends is the ability to have live communications with them via chat. When a friend is online, he/she will appear in the chat tab as an available friend to chat with. To start a chat session, simply click on the name of the friend which will begin a new chat session. The little bar at the bottom will change from chat disabled to chatting with friend name as seen in the figure below. Also, the friend receiving the chat will get a notification that a new chat session has started. Then the two friends can chat with each other until one or both log out.

The chat feature is a little more complex than others as it requires a combination of JavaScript and JQuery to work. When a user clicks on a friend name to begin a chat, a chat request is sent to the database. On every page there is a JQuery function that queries the server for any new chat requests once a second. It needs to be JQuery, so the client can request for the server to run a check and return any relevant information. If it was pure PHP, the page would never load completely, because the page cannot finish loading until the server is done all preprocessing. The JavaScript portion of JQuery allows the client to ask the server to do more processing after the page has been loaded.[3] If there is a new chat request, a pop under is generated, notifying the receiver of the new chat that is starting or a new pending chat if the user is in another chat. When a user send a message, another JQuery function is called, to alert the server that it needs to update the chat session with a new message. At the same time, another JQuery function is running to ask the server once a second if the chat session has been updated. Overall, it is a lot of the client asking the server to send over any changes that exist.

• Messaging

Unlike with chatting, users can send a message to any other user. For user A to send a message to user B, he/she simply goes to the message center tab and clicks compose message. He then fills out the form as shown in the figure below with the email of the recipient, the message title, and the message content. The new message will then be in User B's inbox in the message center.

Messaging is implemented by storing a record of the message in the server. First a message object is constructed that takes in the to and from accounts, the message body, and the date it was sent. The message body is retrieved from a user submitted form, the from account is retrieved from the user session, and the receiver account is retrieved from the recipient email address. The receiver account is validated, to ensure the sender didn't try to send a message to a non-existent receiver. Finally, the new message is stored in the server. Retrieving messages sent to the user is a simple method of querying the server for any messages that have been sent to the user, and this is done by checking the receiver id of each message against the account id of the user. Any matches are returned to the inbox of the user as shown in the figure below.

• Events

In addition to messaging your friends, social networking sites are a great place to alert your friends of important upcoming events. For example, if a user is throwing a birthday party, he/she could create an event and invite friends he/she wants to attend or simply make it a public for anyone. The process of creating an event is as simple as filling out a form which looks like this.

After filling out the form, the user will be brought to the standard confirmation page. He/she can then view the new event by clicking on the Events tab and clicking on the new event. If the user wishes to edit some information about the event, he/she can click on the edit link next to the event where a similar form to creating an event will be displayed with all the fields populated with the current event information.

Storing the event is a fairly straightforward process. Once the form data is submitted, a new Event object is created storing the Account that created the event as well as all the information sent along with the form. That new event is then sent to the server, which creates a database query and stores the fields of the event object into equivalent fields of the event database table. Retrieving events is also a fairly straightforward process. A MySQL query is called to pull the contents of the event into an array, and that array is used to instantiate a new working copy of the event in an Event object.

• Media Uploading

As people use their social networking account, they will want to be able to upload funny or interesting images, video, music, etc., to share with their friends. The media upload section will be located at the media tab, where users are able to specify a file to upload as well as provide a short description of the file to be uploaded. In order to prevent users from uploading potentially malicious files such as executables, only certain file extensions are supported. These allowed extensions cover popular image extensions such as png, jpg, gif, and bitmap, video

extensions wmv and avi, audio extensions mp3, wma, and wav, and some document files including txt, rtf, doc, and pdf.

The inner workings of this uploading process are surprisingly simple, since the HTTP server takes care of requesting the file from the client automatically. After the user clicks upload, the file is sent to the server and stored in a temporary location. Through PHP all information about these temporary files can be accessed through the \$ FILES variable. The temporary location is stored in that variable and can be used to pull the file name and file extension.[1] That extension is then checked against an array of allowed extensions. If the extension is in the list, the file is then moved to a permanent location in the media folder under a subfolder for the user's account. If the extension is not on the list, it is left in the temporary folder where it will be automatically deleted by the server once the php script completes.

CLASS AND OBJECT DIAGRAM:-

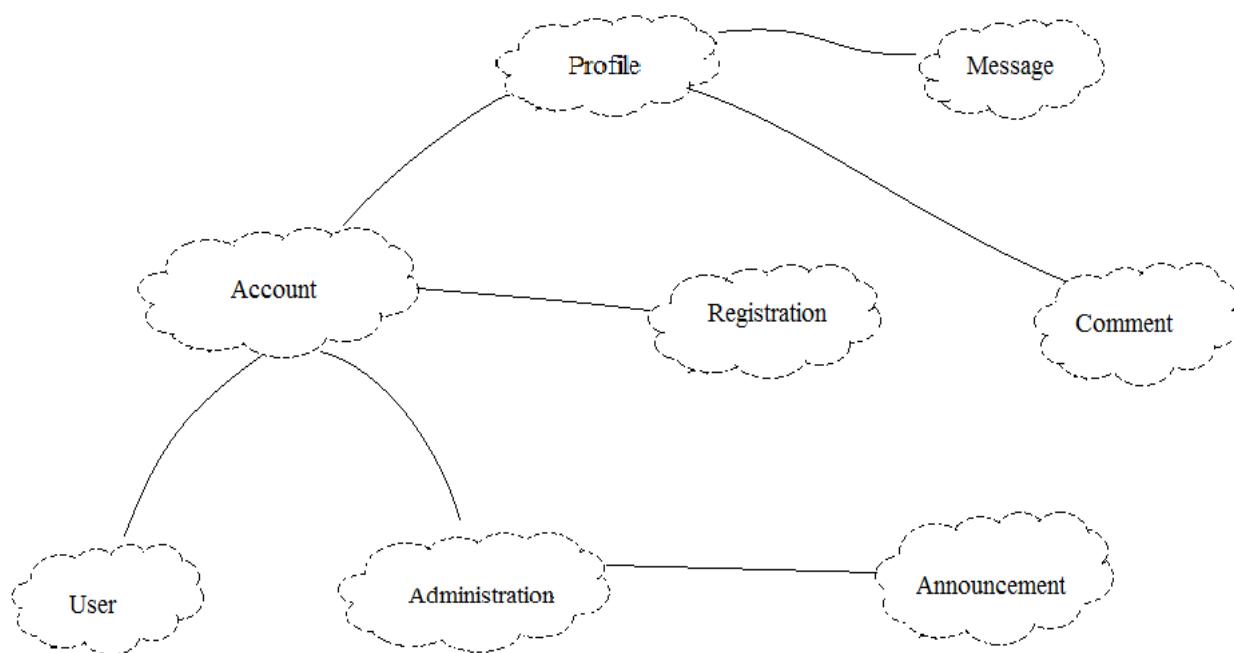


Fig:-class diagram of social networking system

Non-Functional Specification:-

- Secure access of confidential data by user name and password. This application is secure for every kind of its users, because if any user logout from any session then nobody will be able to access his profile without knowing his confidential password.
- 24 X 7 availability
- Better component design to get better performance at peak time.
- The database used here is robust, reliable & fast. So users will have to wait for the output very short time.
- This application can be accessed from any type of platform.
- There is no case of redundancy in the database so it will not take extra memory space.
- Username & password are sent to the users via email after registration.
- Password recovery system is also provided in case of forgetting the password.

Use case diagrams:-

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components. The boundary, which defines the system of interest in relation to the world around it.

- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

• Login/Registration

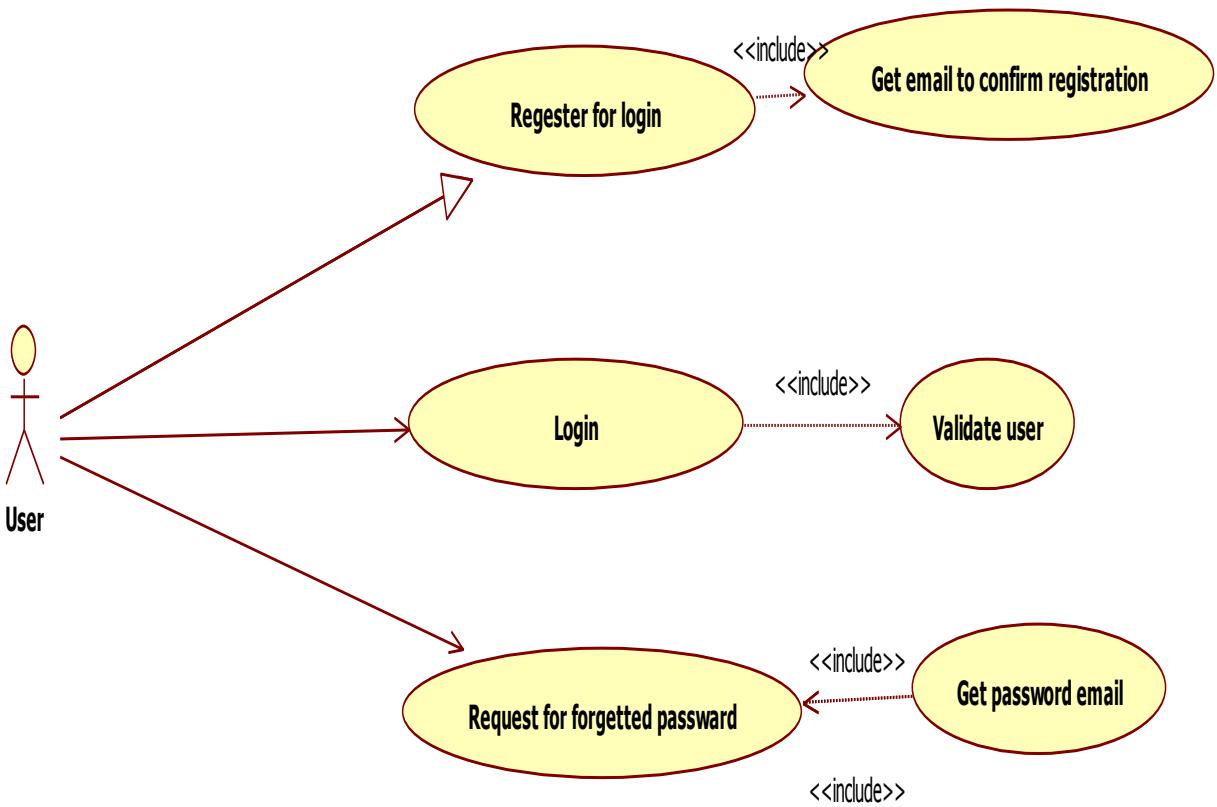


Fig :- Login/Registration

• Home Page

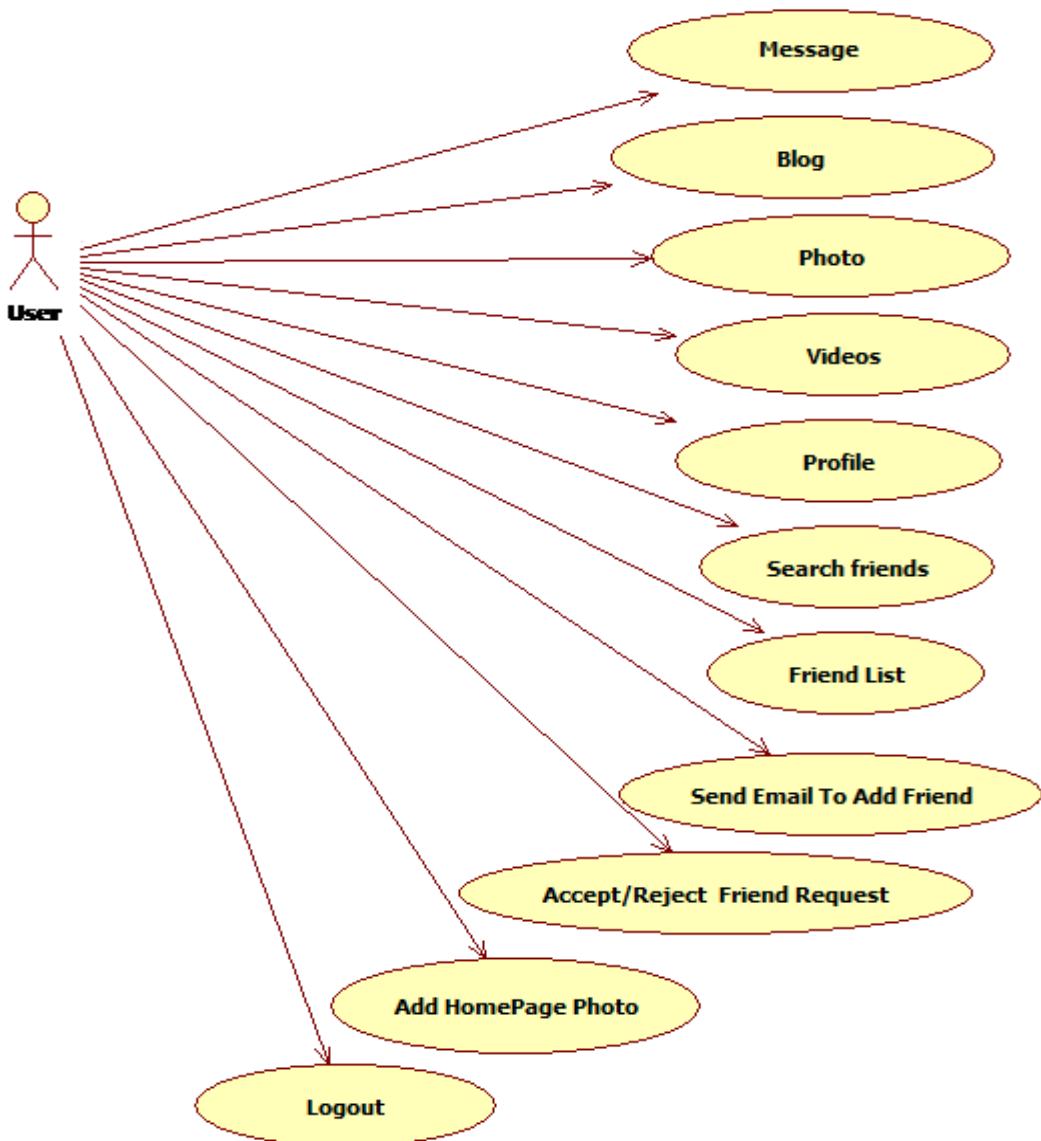


Fig :- homepage

• Wall Page

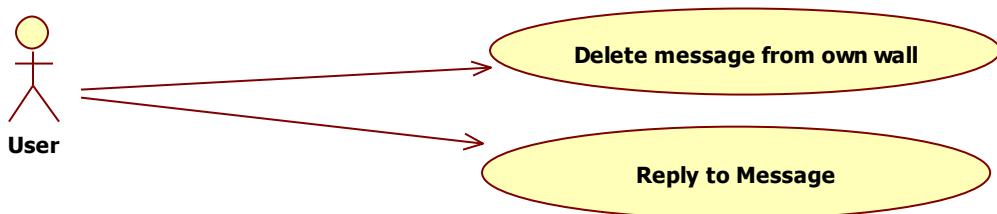


Fig :- Wall page

• Blog Page

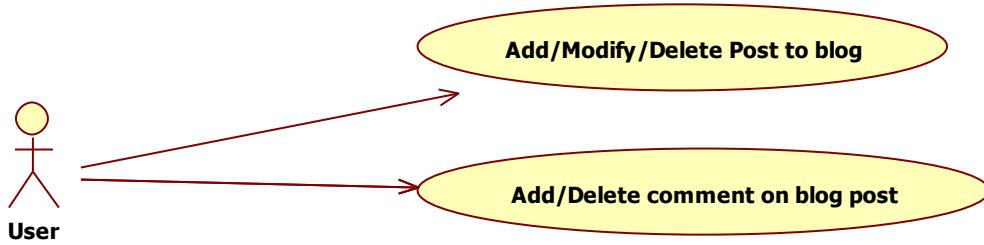


Fig :- Blog page

• Profile Page

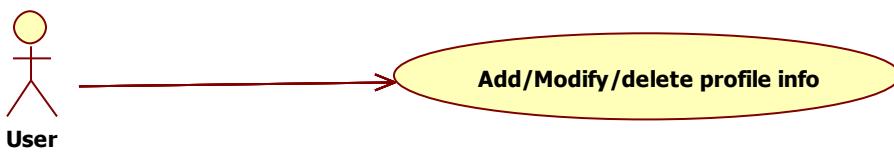


Fig :- Profile page

• Search Friend Page

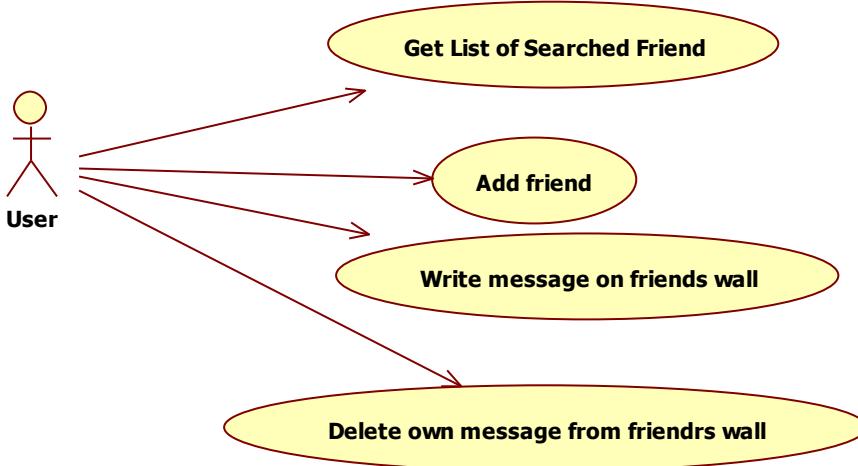


Fig :- Search friend page

- **Friend List Page**

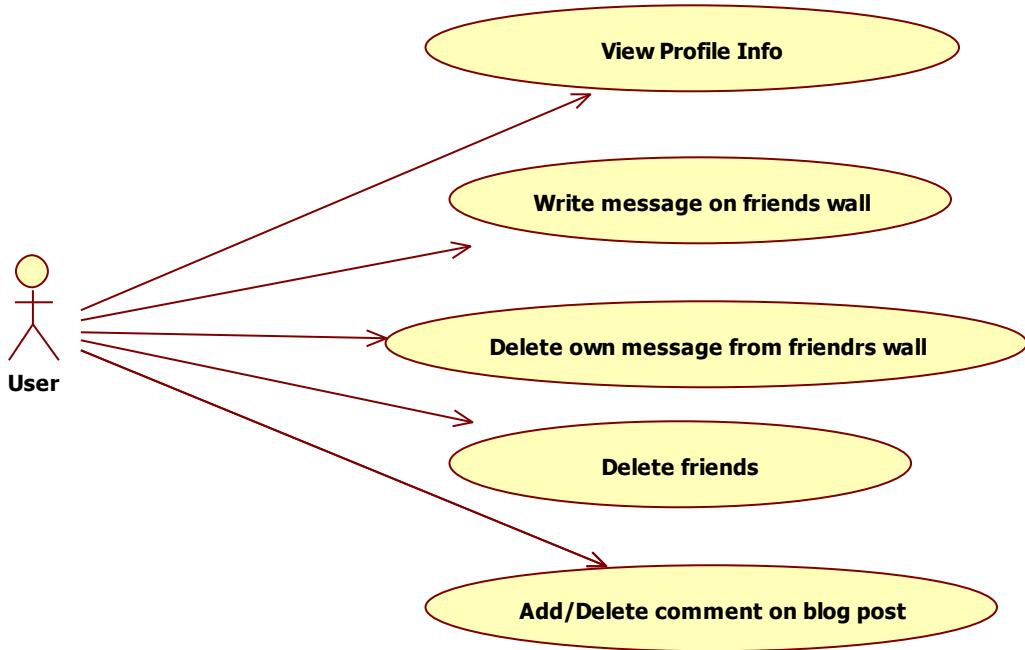


Fig :- Friend List Page

- **Photo Page**

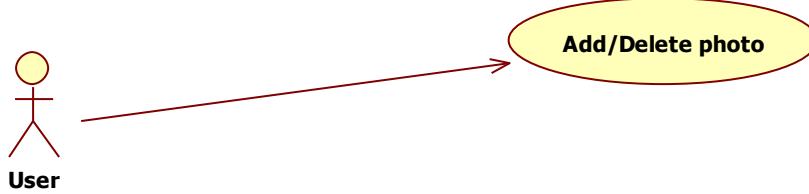


Fig:- Photo page

STATE TRANSITION DIAGRAM :-

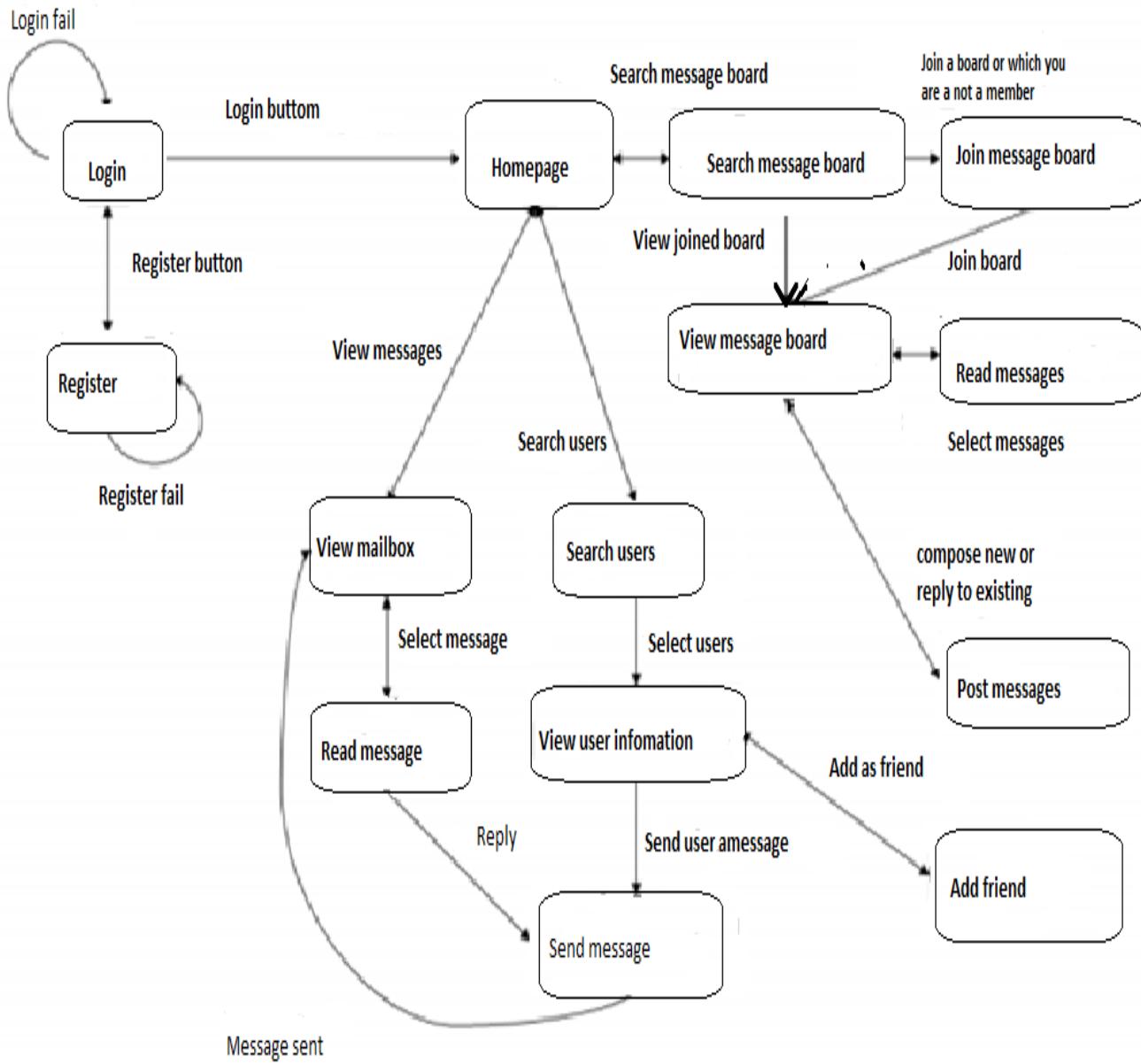


Fig:-State transition diagram of facebook

State diagram shows the behaviour of the classes in response to external stimuli .

specifically state diagram describe the behaviour of a single object in response to a series of events in system .

sometimes it is also known as a state machine diagram . Create state diagram when the business logic for a particular flow is very complex , or needs greater

understanding by the developer to be coded perfectly . Arrange the state and transition to

ensure that the lines that cross each other are minimal .Criss crossing lines a potential sources of confusion in conveying the diagrams meaning .

SEQUENTIAL DIAGRAM:-

An example of sequence diagram which shows how facebook user could be authenticated in a web application to allow access to his/her facebook resources .

if web application obtain the access token for facebook user it can perform authorized request on behalf of that

Facebook user by including the access token in the facebook graph API requests .if the user did not authorize web application facebook issues redirect request to the URI specified before and add the errors reason parameter to notify the web application that authorization request was denied.

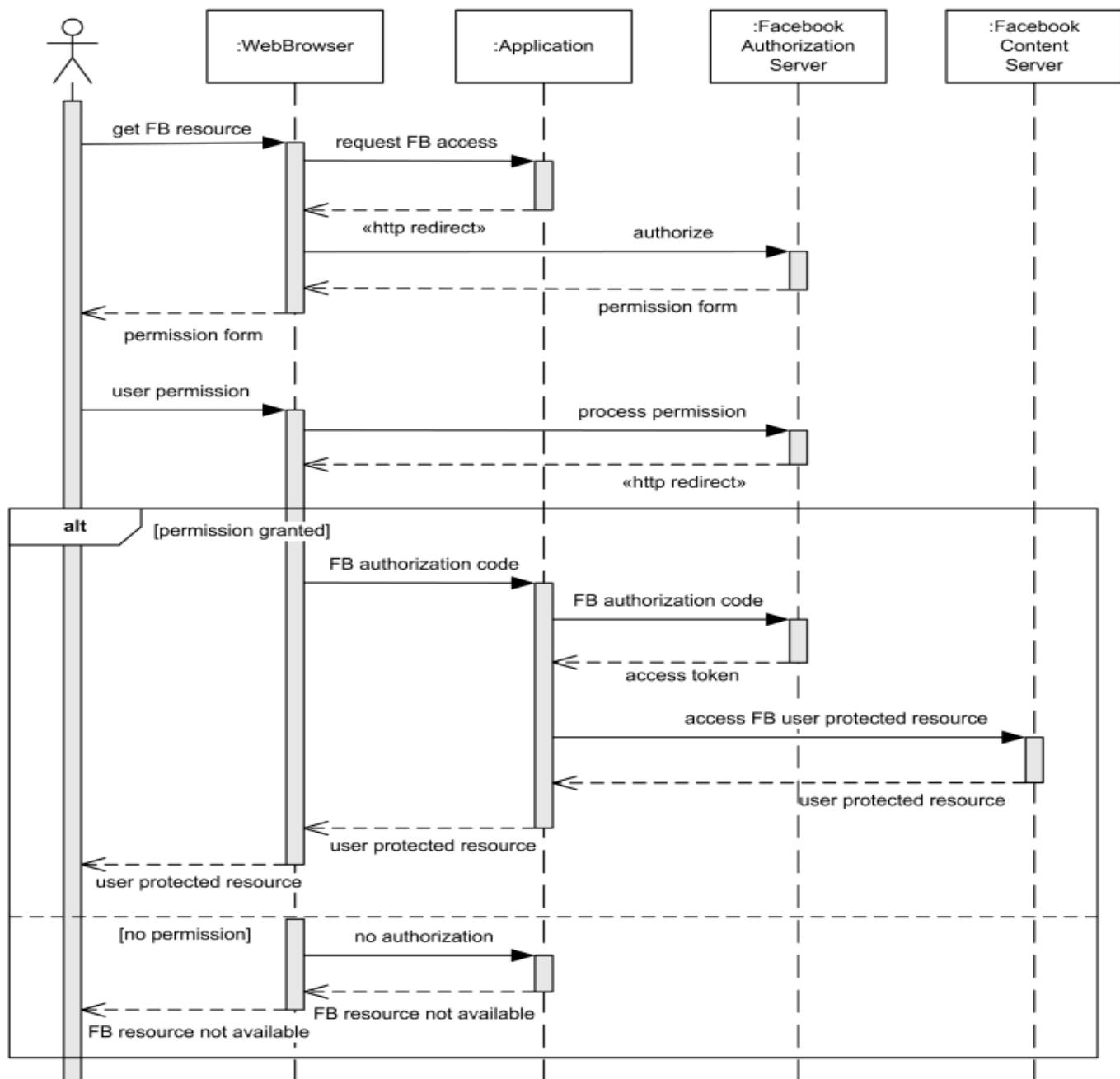


Fig:-sequential diagram of facebook login.

COLLABORATION DIAGRAM:-

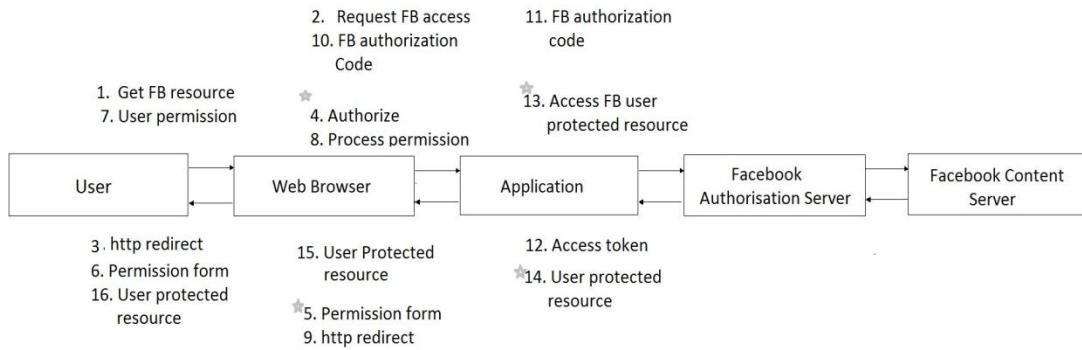


Fig: Colaboration Diagram for Facebook Login (PERMISSION GRANTED)

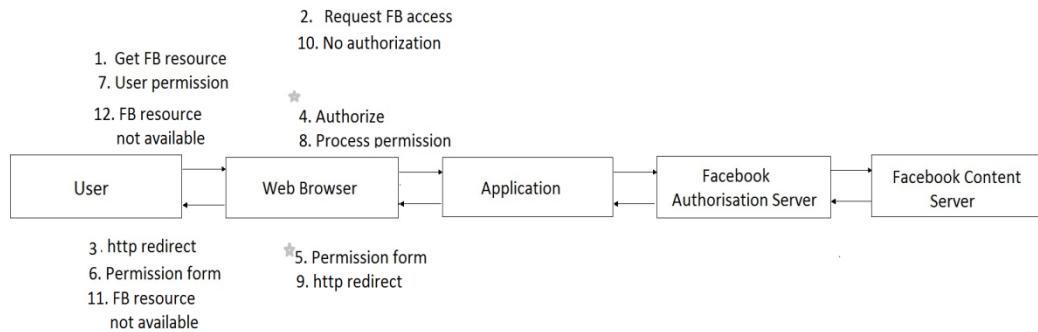


Fig: Colaboration Diagram for Facebook Login (NO PERMISSION)

ACTIVITY DIAGRAM:-

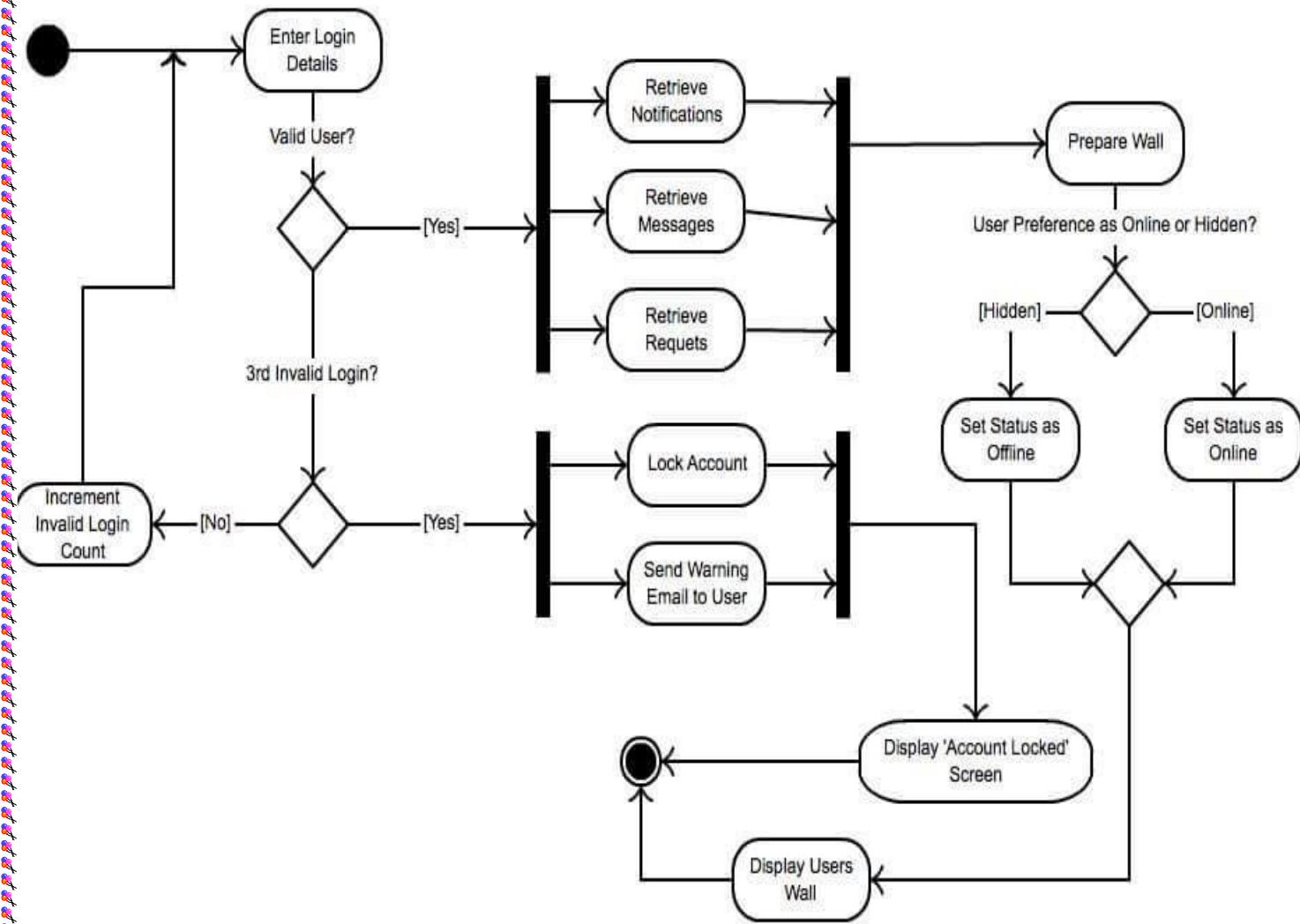


Fig:-Activity diagram of facebook

Activity diagram is another important diagram in booch notation to describe dynamic aspect of the system .

Activity diagram is basically a flow chart to

represent the flow from one activity to another activity . the following diagram shows the activity diagram of a social networking system which is like facebook.The following diagram is drawn

with the three main activities 1)Login 2) see notification feed 3) Do a post.

CONCLUSION:-

After going through this project we came to know about the real time class and object and their relation . What does a class and an object mean , literally how to identify class and object. Also we came to know about various type of diagram such as class diagram , object diagram , sequence diagram , state transition diagram , activity diagram , etc.



BIBLIOGRAPHY:-

BOOKs:-

Object oriented design & analysis

By- GRADY BOOCHE.

Wikipedia.(internet)

https://en.wikipedia.org/wiki/Object-oriented_design

https://en.wikipedia.org/wiki/Object-oriented_design

<http://www.objectorienteddesign.org/>

<http://www.slideshare.net/maaano786/srs-of-social-networking>