Course Name : Software Engineering 1

TA : Mohamed Samir

Project Name : Social Network

Project Year : 2014

Leader Name : Mahmoud Fathi Gamal El-Dien

Contact : mahmoudfathi2016@gmail.com

Contents

Team

ID	Name	Email	Mobile
20120358	Mahmoud Fathi Gamal El-Dien	mahmoudfathi2016@gmail	
20120357	Mahmoud Adel Abd El-Aal		
20120370	Mahmoud Nasser Mahmoud		
20120369	Mahmoud Morsy Ahmed		

Document Purpose and Audience

- **This document describe the project of social networking and its purpose and scope**
- **This document specifies all the features required in the software including technical requirements and its purpose and scope**
- **The audience of this document is the clients that wants to perform domain analysis or an organization that wants to develop a social network .**

Introduction

Software Purpose

- **The purpose of the system is to develop a backend API of a social network that connect its users by sharing their statuses and messaging each other**

Software Scope

- **The scope of our software is to create a social network that connects people in local area like a faculty and restricted number of users of 1000 person.**
- **The major features of this software that the user can create pages of a products or for entertainment and create groups of other users to share something between them only , the user can communicate with others by messaging or sharing photos .**

Definitions, acronyms, and abbreviations

Admin	Admin is the person who manages a group or a fan page
Hashtag	Technology used to specify a post or categorize it
Like	User gives admiration of a post of another user's
Share	User makes the same post of another user's like a copy of it
Post	Can be a text that a user writes or a photo he uploads

Requirements

Functional Requirements

- **Developer Control**

  The developer can deactivate any user's account or page or group . He can remove any post , he can get all users data .

- **Signing up**
  - The user must signing in the system to use it
  - He must enter his name, email, gender, birthdate, and his password to protect his account privacy.
- **Loginig in**
  - The user enter his account by entering his username and his password to use the software .
- **Loginig out**
  - The user logs out of the system after he ends using it .

- **Editing information**

  - User can edit his account settings like editing his username, password, profile picture, gender, and his mail and so on.
- **Friend request list / Follower list**
  - To show the friend requests or follows that you have and you have the full right to accept them or not .
- **Show friend list**
  - The user can see his friend list and have the right to chat with any of them or unfriend or unfollow any one of them .
- **Show photo list**
  - User can see the photos he posted or the photos that any friend posted to him or the photos he was tagged in .

- **Search in the bar**
  - The user can search about a page or a group or about another he wants to follow or wants to send him a request to be a friend
- **Creating a group**
  - The user of the social network can create a group that consists of another users
- **Managing the group**
  - The user how created the group can manage it by making it public or closed or secret group
  - He can accept how will be a member of the group or how will be admin to manage the group and he can ban a member of the group
- **Creating a fan page**
  - A social network user can create a fan page user for specific entity or product of any type such he can create a page specified in sports or entertainment or medical page and so on.

- **Managing a page**
  - The admin of the page can control how can be another admin and how will be banned from the page and maintain the number of likes (number of users like the page).
- **Adding a post**
  - A user can post on his personal page or in a joined group and if he has a page he can post on it .
  - The post can be of different types post of normal user on his timeline or a page post that it has the page profile picture and the admin of the page he is the only person can post and premium user post like admin of group that can’t be removed or edited .
  - Any othe friend can like or share his post.

    - The post can have many likes and shares.

- **Editing / Deleting**
    - The user can delete or edit his post

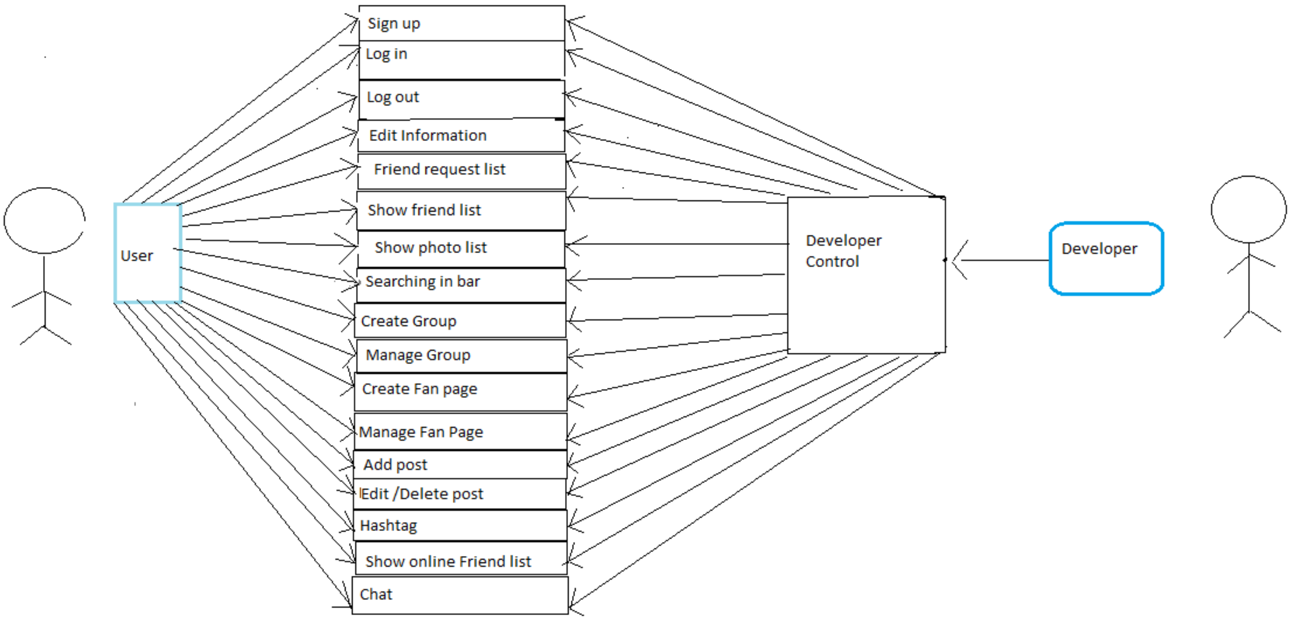
- **Hashtag**
    - Users use hashtags to categorize the content of specific post, if the user wants to write a post about specific thing , he puts the '#' before the matter he wants .
    - He can use the hashtag to read posts about something specific , and to get all posts that contains specific hashtags and sort these posts according to its importance like posts that have many likes.
- **Show online friend list**
    - The user can show his online friend list to see the friend he wants to chat with him
    - He can create a message to any othe user so the chat can be between 2 users or a group of users
    - He can clear the message box between him and any other users

Non Functional Requirements

Requirement :	**Details:**
Quality Requirement	• **Availability , that the server of software shouldn't be broken down and it will operating six-nine 99.9999% in its life cycle determined by the client.** • **Backup, the data stored on two different databases to recover the data if the main database broke down so we call the alternative database.** • **Recovery, the software has several servers so if the main server broke down we can activate the software on another server.** • **Response time, the system take 5 seconds in logging operation** • **Resource Usage, the software takes 100 MB of the memory** • **Reliability, the time between failures of the servers.** • **Usability, there will be a search bar to search for specific user, group or page . There will be a list of groups that the users involved in and pages he likes. There will be a list of friends who are online . There will be a button that refreshes the timeline and a button for settings if the user wants to changes his personal information.** • **Interface, there will be a simple user interface . there will be some buttons and menu lists that helps the user using the software easily.** • **Reusabilty, this API can be used to be the backend of any social network.**
Platform Requirement	• **Supported on Windows platform** • **Supported on Linux Platform**
Process Requirement	• **Depends on the plan of the client** • **Depends on the schedule time organized by the client**

System Models

Use Case Model



Use Case Tables

Use Case ID:	1	
Use Case Name:	Signinig in	
Actosr:	User	
Pre-conditions:	The user trying to create an account	
Post-conditions:	The user will have an account	
Flow of events:	**User Action**	**System Action**
	1- User Enter his mail	1-System verify the mail
	2- User enter his name	2- System Verify user data
	3- User Enter his password	3-System authenticate the password
	4- User Select the gender	4-System verify the gender wether the user is a male or female.

Exceptions:	User Enters an invalid email or enters invalid name 1-System returns to step1 waiting entering valid mail and name
Includes:	No use cases involved here
Notes and Issues:	The password must start with capital characters mixed with numbers

<table>
<tr><td>Use Case ID:</td><td colspan="2">2</td></tr>
<tr><td>Use Case Name:</td><td colspan="2">Login</td></tr>
<tr><td>Actorsr:</td><td colspan="2">User</td></tr>
<tr><td>Pre-conditions:</td><td colspan="2">User try to enter his account</td></tr>
<tr><td>Post-conditions:</td><td colspan="2">User enters his account</td></tr>
<tr><td rowspan="5">Flow of events:</td><td><b>User Action</b></td><td><b>System Action</b></td></tr>
<tr><td>1- User Enter username and Password.</td><td></td></tr>
<tr><td></td><td>2- System Verify user data</td></tr>
<tr><td>3- User enters his profile</td><td></td></tr>
<tr><td>4- User interacts with the system by the functions available to him</td><td></td></tr>
<tr><td>Exceptions:</td><td colspan="2">User enters an invalid username or invalid password
1-user will not be loged on the system
2-use case resumes to step1</td></tr>
<tr><td>Includes:</td><td colspan="2">No use cases included in this use case</td></tr>
<tr><td>Notes and Issues:</td><td colspan="2">User must enter the username and password correctly</td></tr>
</table>

Use Case ID:	3	
Use Case Name:	Log out	
Actors:	user	
Pre-conditions:	User wants to quit his account	
Post-conditions:	User account is already quited	
Flow of events:	**User Action**	**System Action**
	1- User clicks on the log out button	
		2- System ends the user session
Exceptions:	No exceptions here	
Includes:	No use cases included here	
Notes and Issues:	No notes	

Use Case ID:	4	
Use Case Name:	Editing Information	
Actors:	User	
Pre-conditions:	User trying to update/change his personal info	
Post-conditions:	The info are already updated	
Flow of events:	**User Action**	**System Action**
	1- User enter the field he wants to update .	
		2- System Verify user data
	3- user click save	
		4- System replace the old info with the new info
Exceptions:	User enters invalid field or incorrect field 1-System returns to its current state before editing operation	

Includes:	Log in use case
Notes and Issues:	User must enter a valid field and must enters it correctly

Use Case ID:	5	
Use Case Name:	Friend request list	
Actosr:	User	
Pre-conditions:	The user wants to know the other users who want to be friend with him	
Post-conditions:	The user accepts them or not	
Flow of events:	**User Action**	**System Action**
	1- User Enter the button of the friend list	
		2- System shows the list of user that sent friend request
	3- User has the right to accept them or not .	
	4- if the user accept	
		5-System adds the new friend to his friend list
Exceptions:	No exceptions here	
Includes:	Log in use case	
Notes and Issues:	User has the right to accept or not the users that want to make friendship with him	

Use Case ID:	6
Use Case Name:	Show friend list
Actors:	User

Pre-conditions:	User want to show his current friend list	
Post-conditions:	User know the friends in his friend list and their number	
Flow of events:	**User Action**	**System Action**
	1- User click on friend list button	
		2- System show his current friend list
	3- user select one friend to chat with him	
		4- System shows the message box between them
	5- user wants to unfriend someone	
		6- system remove this user from the friend list
Exceptions:	No exceptions here	
Includes:	Log in use case	
Notes and Issues:	The user has the right to start chat with a friend if he is online He has the right to remove a friend from his friend list	

Use Case ID:	7	
Use Case Name:	Show photo list	
Actors:	User	
Pre-conditions:	User trying to see what is in his photo list	
Post-conditions:	User knows his photo list	
Flow of events:	**User Action**	**System Action**

	1- User click on the button	
		2- System views the photos that he posted or the photos posted to him by a friend
	3- User can select any photo to delete	
		4- System removes this photo from his list of photos
Exceptions:	No exceptions	
Includes:	Log in use case	
Notes and Issues:	User can see his photos he posted or shared He can remove any photo or add to his photo list	

Use Case ID:	8	
Use Case Name:	Search using the search bar	
Actors:	User	
Pre-conditions:	User wants to search about a page, group or user	
Post-conditions:	User gets the desired output	
Flow of events:	**User Action**	**System Action**
	1- User Enter the name of the group, page or the friend he wants to search for	
		2- System views all possible outputs
	3- User Select the desired entity you want from the list	
		4- System takes you to the desired input
Exceptions:	User searches for invalid entity	

	1-system will not show any thing
Includes:	Log in use case
Notes and Issues:	User must enters valid or real thing related to the website so the system shows any possible entities related to what the user entered in the search bar.

<table>
<tr><td>Use Case ID:</td><td colspan="2">9</td></tr>
<tr><td>Use Case Name:</td><td colspan="2">Create group</td></tr>
<tr><td>Actors:</td><td colspan="2">User</td></tr>
<tr><td>Pre-conditions:</td><td colspan="2">User wants to make a group of some interest</td></tr>
<tr><td>Post-conditions:</td><td colspan="2">The group is already done</td></tr>
<tr><td>Flow of events:</td><td><b>User Action</b></td><td><b>System Action</b></td></tr>
<tr><td></td><td>1- User click on the button to creat a group</td><td></td></tr>
<tr><td></td><td></td><td>2- System asks the user to name the group</td></tr>
<tr><td></td><td></td><td>3-the system shows the message of creating the group successfully</td></tr>
<tr><td>Exceptions:</td><td colspan="2">User doesn’t complete the all the required from the system to create the group
1-group will not be created
2-the system will asks the user to complete all the required</td></tr>
<tr><td>Includes:</td><td colspan="2">Log in use case</td></tr>
<tr><td>Notes and Issues:</td><td colspan="2">User must enter good name in good way</td></tr>
</table>

Use Case ID:	10	
Use Case Name:	Managing the group	
Actors:	User	
Pre-conditions:	User wants to edit the settings of the group	
Post-conditions:	The settings are updated	
Flow of events:	**User Action**	**System Action**
	1- User click on the settings button	
		2- System views the list of settings of the group
	3- User Select the settings that he wants to edit like making it closed or public group	
		4- System update this setting
	5- the admin user can show list of users that wants to join this group and he can accept them or not	
		6-if the user accepted the member, the system adds this user to the list of members of the group
	7-The admin can ban a user from the group	
		8-the system removes this user from the member list of the group
Exceptions:	No exceptions	
Includes:	Log in case and create group use case	
Notes and Issues:	The admin user can control the group He can promote any user to be admin	

	He can accept members and can ban members and control the type of the group He can remove any posts of any user

Use Case ID:	11	
Use Case Name:	Creating a fan page	
Actors:	User	
Pre-conditions:	User wants to create a fan a page of products or for entertainment	
Post-conditions:	The page is created for specific entity as the admin of it wants	
Flow of events:	**User Action**	**System Action**
	1- User clicks on the button to create a page	
		2- System asks the user about the type of the page
	3- The user specifies the type	
		4- The page is created with the type that the user chose
Exceptions:	that the page notify the user that the info needed are not complete 1-The system returns the user to step2 to complete all the info	
Includes:	Log in use case	
Notes and Issues:	User must complete all the info needed	

Use Case ID:	12
Use Case Name:	Managing the page
Actors:	User
Pre-conditions:	User wants to edit the settings of the page
Post-conditions:	The settings of the page is updated

Flow of events:	**User Action**	**System Action**
	1- User can ban a user from the page	
		2- System removes the user from the the list of users who likes the page
Exceptions:	No exception	
Includes:	Log in and create page	
Notes and Issues:	User can control the page by banning a user who likes the page He manages the number of likes of the page	

Use Case ID:	13	
Use Case Name:	Adding post	
Actors:	User	
Pre-conditions:	User wants to share his status with his friend or with specific group	
Post-conditions:	His status shared in the conditions he specified	
Flow of events:	**User Action**	**System Action**
	1- User Enter text or photo to be posted	
		2- System shows the post to the selected friends`
Exceptions:	That the status of the user is not shared among his friends 1-user should try again by posting what he wanted to share again If the same problem appear again he must inform the developer by filling evaluation form available to the user to report about a problem	
Includes:	Log in use case	

Notes and Issues:	User must enters the text in the area specified for adding a post User can add post on group or page

Use Case ID:	14	
Use Case Name:	Editing / Deleting post	
Actors:	User	
Pre-conditions:	User wants to edit the text of the post or delete it	
Post-conditions:	His status will be edited or deleted	
Flow of events:	**User Action**	**System Action**
	1- User select the post he wants to edit or delete	
		2- System edit the post if the user wants to edit or delete if he wants to delete
Exceptions:	No exceptions	
Includes:	Log in and adding post	
Notes and Issues:	User select the post of him he wants delete or edit and repost the new one if he edited	

Use Case ID:	15
Use Case Name:	Hashtag
Actors:	User
Pre-conditions:	User wants to categorized his status (post) or search for selective category

Post-conditions:	System shows the result of this category to User	
Flow of events:	**User Action**	**System Action**
	1- User Enter # before his status	
		2- System categorized the post
	3- User Enter # in search bar to search for specific status	
		4- System show the categorized status
Exceptions:	No exceptions	
Includes:	Log in and adding post	
Notes and Issues:	User must add # sign before writing a post He can search for a categorized post by using the # .	

Use Case ID:	16	
Use Case Name:	Show online friend list	
Actors:	User	
Pre-conditions:	User wants to show the online friend list	
Post-conditions:	Online friend list will be shown in user‘s Home	
Flow of events:	**User Action**	**System Action**
	1- User press on the chat button	
		2- System show the online friend list
	3- User Select one of the online friend	
		4- System show the message box of this friend
	5- User write text for his friend and press enter	

		6- System deliver the message to User’s friend
Exceptions:	That user can’t see the users that are already online He can report the system	
Includes:	Log in	
Notes and Issues:	User choose a friend he wants to chat with and write the message in the dialog box that appear once he choose the friend	

Use Case ID:	17	
Use Case Name:	Developer Control	
Actors:	Developer	
Pre-conditions:	Developer wants to control and collect user‘s data	
Post-conditions:	System enabled him to Control the information	
Flow of events:	**User Action**	**System Action**
	1- Developer wants to collect the data	
		2- System enables him to collect data
	3- - Developer wants to control the data	
		4- System enables him to control data
	and so on	
Exceptions:	No exceptions	
Includes:	No included use cases	
Notes and Issues:	The developer wants to collect data from the users to use it in another application	

Ownership Report

Item	Owners
Requirements	*The Team*
System Models	*The Team*

Policy Regarding Plagiarism:

Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.