

CS251: SOFTWARE ENGINEERING-1- 2014

SOCIAL NETWORK

TA-MOHAMED SAMIR

SRS Doc TITLE: SN-API

LEADER: LOBNA MOSTAFA EL-SHAZLY

CONTACT OF LEADER: 01146312043

TEAM:

<u>ID</u>	<u>NAME</u>	<u>EMAIL</u>	<u>MOBILE</u>
20120280	Lobna Mostafa El-Shazly	lobna.elshazly@hotmail.com	01146312043
20120179	Raghda Mohamed Ahmed		01220888180
20120420	Nada Abdelkader Khaled		01281263296

Document Purpose and Audience

- This document has a description of a back-end API software, which is able to manage interactions between users, create and manage groups and pages, manage different type of posts and retrieve current posts in specific hashtag and sort these posts according to the most important post
- Audience of this document are developers-TAs-, and project managers.

Introduction

Software Purpose

- The purpose of this project is to develop a back-end API for social network. Something like Facebook API.
- This is to able you develop on a back-end API that is able to manage interactions between users, create and manage groups and pages, manage different type of posts and retrieve current posts in specific hashtag and sort these posts according to the most important post

Software Scope

- This API software has Users, which use a social network, who creates accounts, connect with their friend users, by messages, posts, gathering in groups
- Users have types as Normal, premium, aslo have the ability to publish posts create messages, make groups.
- Groups have types as Public, Private, which can be controlled by the user, and Premium user has extra features as he/she can create a group and change its privacy.
- Posts can be controlled to be public, private by the user, but the Premium user has extra features as adding more information to the post about his/her status.
- Hashtags for categorization of posts, can be seen if the user searched for the hashtagged word preceded by '#'.
 - Messages are to create chat between users, or group chat, by typing message in the chatbox, either to one friend at a time, or many.

Definitions, acronyms, and abbreviations

<u>API</u>	In computer programming, an application programming interface (API) specifies
------------	--

	a software component in terms of its operations, their inputs and outputs and underlying types.
<u>HASHTAG(#)</u>	Users use hashtags to categorize the content of specific post.
<u>DEVELOPER</u>	the user dealing with this document, to handle the API.

Requirements

Functional Requirements

- **Installing the API: (complexity of level one)**
 - API is downloaded from the GitHub for the first time by copying the link from the clone from the Repository, and using the command of git clone (link).
 - The API will be saved in the folder in the local system, which the user had chosen.
- **User Operation: (complexity of level two)**
 - The developer can create two types of users (Normal user, Premium user), they will have common attributes such as name, email, gender, email, P.P(profile picture)...etc., but the Premium user will have extra attributes such as having the ability to login or create his own account on the social Network with his/her mobile number, Credit serial number.
 - Social network user can sign up or login - in case signed up already- by his/her own username, password -in case login for Normal user -, extra email, mobile number is needed if signing up for both types of users. Email, and mobile number is an extra features for premium user to login.
 - Social Network user will sign up by writing his username, password,
- **Groups Operation: (complexity of level two)**
 - The developer should create and manage different types of groups (public, private), only premium users can create secret groups, for his own privacy with his friend users, while Normal user can create public groups.
- **Pages Operation:(complexity of level three)**
 - The developer should be able to create user-fan page-, with this component he/she should be able to create and manage user pages and maintain some important attributes (eg: Number of users likes, Number of views), only premium user can create the Verified pages.

- Only premium users can be admins for the verified pages.
- Pages are created, where posts of the admin -premium user- are published to fans, to be liked, and shared.
- **Posts Operation:(complexity of level five)**
 - The social Network user will want to share his/her status in Social Network by writing posts. A post is a text-content written by the website user to express something and share it with other users, and the user has the ability to control who will see his/her posts. Post's privacy can be controlled to be public or private.
 - Premium users can add additional information like his/her location using the foursquare.
 - Social Network user can write a post, in which he/she can tag her friends, including images.
- **Hashtags Operations:(complexity of level five)**
 - This operation is to categorize the content of specific posts, there must be a link between the hashing and its post
 - Posts should be sorted according to its importance and date.
 - Hashtags are written down by the users on their posts to be easily found on search engine.
- **Messages Operations:(complexity of level four)**
 - Social Network user can send messages to other user(s) in his friend list, be able to create message from user to user, get messages between two users or get messages from a specific group
 - Normal user can send messages and create a chat between his/her friend, send images, links, smileys.

Non Functional Requirements

- **Performance**
The response time must be short and take few seconds
Creating an account, a group, a new message or a new post should take 5 seconds.
- **Usability**

There must be using hints when the user creates his first account and hints during using the site to make everything clear and easy.

- **Security**

The social network site should be well secured because it contains data and information of users so the privacy of these details must be high.

- **Implementation**

The API must be implemented in java.

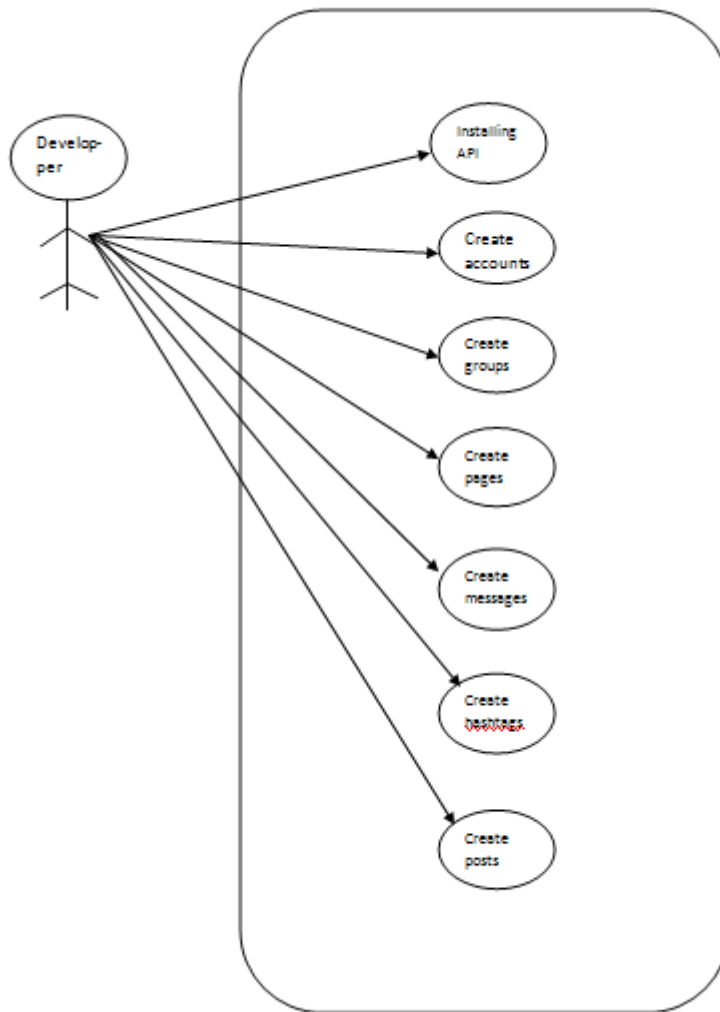
- **Interface**

The interface should be simple to be easy for the user to create a post, send a message, create a group and so on.

- **Maintainability**

The site must be maintained every week to meet new requirements, make future maintenance easier and cope with a change environment.

USE CASE MODEL



Use Case Tables

Use Case ID:	SN-API-1	
Use Case Name:	Installing API	
Actors:	The original user of the API (developer)	
Pre-conditions:	API on the Github, needs to be installed from its repository.	
Post-conditions:	SN-API in on the local system of the user ready to be used.	
Flow of events:	User Action	System Action

	1-User copy the clone URL of the SN-API for the first time, and pasting it in the command of git clone in the git terminal.	
		2- The command is done and the API is pulled on the local system from the master.
	3- SN-API is ready to be used by the user.	
Exceptions:	If the SN-API is already cloned before, and has already been downloaded on the Local system, so the command to be operated is "git pull origin master", this will pull the API from the Github, with no need to clone it again.	
Includes:	Logging into Github account, and copying clone URL, for terminating the command on the console.	
Notes and Issues:		

Use Case ID:	SN-API-2	
Use Case Name:	User Actions	
Actosr:	The original user of the API, and the virtual user using the software itself.	
Pre-conditions:	Allowing the virtual user to create and logging into his/her account.	
Post-conditions:	Allowing the virtual user to use his/her accessible options on this account.	
Flow of events:	User Action	System Action

	1- The developer allows the virtual user opens the webpage of the SN, and choose if he/she has a ready-made account, or needs to sign up.	
		2-a) In case of new user, he/she has the ability to sign up, using his/her name, email address, gender b) Premium user has the ability to create the account by his mobile number, or the Credit card serial number.
	3- In case the User has already a made account, and he/she needs to login only. in case of Normal user, he/she can login with his/her username, and password, else if a Premium user, he/she can login with the mobile number or credit card serial number	
		4-a) The system creates the account for the signed up user. b) In case signing in, the system login the account.
Exceptions:	In case of invalid password or account, the system has the option to have another email address for the user, in order to recreates a new password and the user can verify it on the another email address.	
Includes:	Accessing the SN webpage to login, writing the username and password, or filling the whole information in case signing up.	
Notes and Issues:		

Use Case ID:	SN-API-3
Use Case Name:	Groups operations
Actosr:	The original user of the API, and the virtual user using the software itself.
Pre-conditions:	Allowing the user to create a group, and maintain some attributes.

Post-conditions:		
Flow of events:	User Action	System Action
	1- The developer allows the user to create a group either it is a public, or private.	
		2- System creates the group as a plain public group and leave a choice for only the premium users to make it secret.
	3- The user is allowed to choose the group privacy of his/her own group. a) In case of normal user, the group will be only public, and he/she can add his/her own friends, and others out of his/her friendlist can join. b) in case of premium user, the group can be public or private, private can enable him/her to add only his/her chosen friends, and enables him to make the group hidden or seen to other friends.	
		4- group is created and posts by admin can be seen, and shared.
Exceptions:	Only the admin, who created the group, have the right to see all the users in his/her group.	
Includes:	logging in the account as an admin, or a user on the group.	
Notes and Issues:		

Use Case ID:	SN-API-4	
Use Case Name:	Pages Operations	
Actosr:	The original user of the API, and the virtual user using the software itself.	
Pre-conditions:	Allowing the user to create a page on th SN.	
Post-conditions:	Allowing the user to be the admin of his/her own page, and enables him/her to view likes, and number of shares, and publishing posts.	
Flow of events:	User Action	System Action
	1- The developer enables the user to create a fan page. a) The user chooses "creating Page"	
		2- The system checks if the user is normal, or premium. a) in case normal user, the options to be allowed for him/her is to post, and view the number of likes in the page, and number of views. b) in case if premium user, he/she can create a verified page, where he/she can be an admin, and can control the page with all its features, and block reported users.
	3- The user is allowed to choose, and establish his/her page.	
		4- Page is created.
Exceptions:	Only premium users can be admins for the verified pages.	
Includes:	Logging in the account as an admin, or a user on the group.	
Notes and Issues:		

Use Case ID:	SN-API-5	
Use Case Name:	Posts Operations.	
Actors:	The original user of the API, and the virtual user using the software itself.	
Pre-conditions:	The virtual user is enabled to publish posts, wether he/she is a premium or normal	
Post-conditions:	Posts are published and its privacy can be controlled by him/her.	
Flow of events:	User Action	System Action
	1- The developer enables the user to write the post, and click "post" button.	
		2- The system publish the post to the public. a)in case the user is Normal user, he/she has the ability to control the privacy, tag friends, and post pictures. b)in case the user is Premium user, he/she has the same abilities of the normal user, but can add his/her location as an extra information.
	3- The user is allowed to write, post, publish the post with pictures.Only Premium users can add their location.	
Exceptions:	No exceptions.	
Includes:	logging into the account, and Filling the post plain text board.	
Notes and Issues:		

Use Case ID:	SN-API-6
--------------	----------

Use Case Name:	Hashtags Operations	
Actors:	The original user of the API, and the virtual user using the software itself.	
Pre-conditions:	The user is allowed to search for a word, preceded by '#', the word is found in any public post.	
Post-conditions:	the user can find out all the available results.	
Flow of events:	User Action	System Action
	1- The user is enabled to search for the word, which is found in any public post, by writing down the word in the search textbox.	
		2- System starts searching for the word, by scanning all the public posts other users have published. a) posts having the word are sorted according to importance and date
Exceptions:	If the post is private, the hashtag results can't be shown.	
Includes:	logging in and typing the word the user is searching for.	
Notes and Issues:		

Use Case ID:	SN-API-7	
Use Case Name:	Messages Operations	
Actors:	The original user of the API, and the virtual user using the software itself.	
Pre-conditions:	The virtual user is allowed to send messages to his/her friends	
Post-conditions:	The user messages his/her friends and send images, links, smileys.	
Flow of events:	User Action	System Action

	1- The user is enabled to write message in the chat box between his/her friends, and ending it by clicking "enter"	
		2- The systems sends the message, to the other friend, by saving it in a buffer, and linking it to the other side where the other friend is reciving.
Exceptions:	No exceptions	
Includes:	logging into the account, and typing the message in the chatbox.	
Notes and Issues:		

Ownership Report

<u>ITEMS</u>	<u>OWNERS</u>
introduction	Nada Abdelkader
requirements	The whole team
System modelling	Lobna El-shazly and Raghda Mohamed