

Course name: Software Engineering 1

TA: Eng. Mohamed Samir

Project name: Social Networks

Leader name: Heba Sayed Shamardy Mohamed

SRS Doc title: The requirements of an API of Social networks

Contact of leader:

ID: 20120476

Email: [hesashmoel@gmail.com](mailto:hesashmoel@gmail.com)

Mobile:

Group: 4-CS

**Team:**

Note: red text represents feedback

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | E mail | Phone |
| 20120430 | Noha Ashraf Ibrahim | [nohaashraf8838@hotmail.com](mailto:nohaashraf8838@hotmail.com) |  |
| 20120426 | Neama Fouad Mohamed | [neama\_fouad55@hotmail.com](mailto:neama_fouad55@hotmail.com) |  |
| 20120431 | Noha Magdy Saber | noha.magdy.99@gmail.com |  |
| 20120428 | Nehal Khaled Hafez | nehal.khaled2012@hotmail.com |  |
| 20120476 | Heba  sayed Shmardy | [hesashmoel@gmail.com](mailto:hesashmoel@gmail.com) |  |

**Document Purpose and Audience :**

* What is this document?

This document is a Software Requirements Specification,

Whish describe the requirements to build software like API for social media network.

* Who is expected to read it?

1-The developer who want to apply some work in my project to get it better than now.

2-The project manager to make sure the project is walk in the right track .

3-The team of developers and designer who will make this project for the customer.

4-The customer to check his requirements and let the developers and analysis and designers begin their work.

**Introduction:**

1-The Purpose:

The purpose of this document is to define the requirements for

“Small social network API “to the customers, another developers and the project manager in a simple way that can anyone read this document can understand well the requirements of the project.

**In this document we are defining:**

1-The scope of our software

Who are interested in it, the functions it will be in, and so on..

2-The functional, non-functional requirements

The description of what the system should do, how the system works, the operation s and the system quality.

3-Examples for use case model.

The expression of the system actors & operations and the system UML .

4- Examples for use case table.

Show for the user/ the system interaction in the system or the software.

2-Software Scope

-As a social communication network (Like Facebook graph) to use in small scale (for example .. for college students and professors ) .

We have too many components to implement starting from the user account and his privacy to the way of sending messages and communicate with the other on the program.

But here in our project we just consider some parts of all of that.

We have 6 main parts of our software:

* The user account and his privacy.
* The pages functionality and what can the user do with it.
* Something called “Hash tag” which the user can see all the posts related to specific topic in few minutes.
* The groups which can block of people can be together in a place which they can talk and share thing no body outside the group can see, or a group of public people who are interested about something can talk about it and share their experience on the group.
* The sending messages functionality, which let the user to send private message to another user or to more than one user.
* The post that the user shares they can be on: (his own page, in a group, in a page, or in his friend page).

And we have 4 main functions for the developer who will use this API:

- The developer can create page

- The developer can write post (a paragraph about something) on any user’s page.

- The developer can manage the posts of the software users, he can get all the posts of specific user by using the date he want (to get the post by the date it was published on).

- The developer have function that let him to get the user data. It is let him to get the main data about specific user like (his name, email, gender, Profile picture, his list of friends).

3-Definitions, acronyms, and abbreviations:

|  |  |
| --- | --- |
| FB | Express the “Facebook “word. |
| Msg | Express the messages text. |
| HshTg | Express the “Hash Tag “class and its specification. |
| User | Express the user who uses the software |
| Priv | Express the private group or message or post. |
| Pub | Express the public group or message or post. |
| pst | Express the block of words that the user writes (Posts). |
| grb | Express the group (the group of people in one private place). |
| API | An API is the standard code or block of classes that any developer or programmer can use to make his own program with it, And implement more functions or classes he need to develop his own program. |

**Functional Requirements:**

## Creating new user account

User should create a new account with unique user name and has a password so that he can use the system.

### Saving user data

To make an account user enters his/her name, email, gender, profile picture, user   
name and password. The system saves these data in database for later usage.

### Choosing user type

After entering user data, each user chooses a user type (normal user or premium user). Premium user has advanced features such as their post can be shown on home page for any normal user.

## User login

A user must login to interact with the system.

## Send friend request

User can send a friend request to another user in the system, this request will be pending request until the anther user accepts this request

## Add friend

User can accept any of the pending requests he receives, user can receive friend requests from the other users and if user accept a request then becomes a friend with the sender of this request

## Signout

User can sign out from the system anytime and leave the system in current time

## Creating new grp

Any user can create a group of users that has a certain goal known through the group name and any other user can ask to join it. Groups can be private (only users who join the group can post in it and see other users posts) or public (Everyone can see posts on the group but only users joined the group can post in it).

## Creating page

Any user or developer can create a fan page for a certain product, entity or cause that has an audience of users interested in.

### Choosing page type

To create a page, the user/developer must choose a type for the page from these types (company, product, public figure, cause, entertainment). The type of the page identifies what the page’s posts should be about.

### Saving page data

The number of likes of a page, its admin and its posts are saved to the database for later usage.

## Writing pst

Any user can write post and publish it to other users.

### Pst on normal user account

In this case when normal user publishes post, it appears only on his friends of users’ home page.

### Pst on premium user account

When a premium user writes a post, it is published to any normal user on the system.

### Pst on fan page

A user can write a post on his/her page related to the page type.

### Pst on a group

A user can post on any group he/she has joined.

### Saving Pst data

Any post is saved to the database with the user name who wrote it, the number of likes to the post and the number of sharing of the post.

## Liking Pst

A user can like a post on his/her home page. The number of likes to the post increases by one.

## Sharing Pst

A user can share a post from his/her home page to his/her own account so that his friends of users can see the post shared.

## HshTg posts

### Writing hashtag post

Any user can write a post which includes # in it. This post is categorized based on a word begins with hash tag. These posts are added to database for later usage.

### Retrieving posts with certain hashtag ordered by its importance

When a user searches for a certain hash tag word, the system retrieves the posts which have this word from the database and show to the user.

## Sending Msg

A user can send a private to a user of his friends by choosing the name of that friend. Also the user can send to a group of his friends by writing their user names and he must name the conversation.

## Getting Msg to show

A user can open his messages with certain friend/friends by choosing the user name of his/her friend or if a group message, the user enters the name of the conversation.

1. **Developer’s Pst (This is not type of post, developer can use createPost function to make post, you addressed this function before)**

A developer can post on a user’s/users’ home page.

1. **Manage posts**

A developer can get and show any posts for certain user by their date duration (posts that were posted by the user between 2 dates) by writing the user’s user name and the date of the post.

1. **Get user data**

A developer can get any user’s data -by writing the user’s user name - to use in developed applications.

(This data such as, list of friends, list of liked pages, list of subscribed groups, list of all posts in newsfeed (profile) , current user type and current privacy settings)

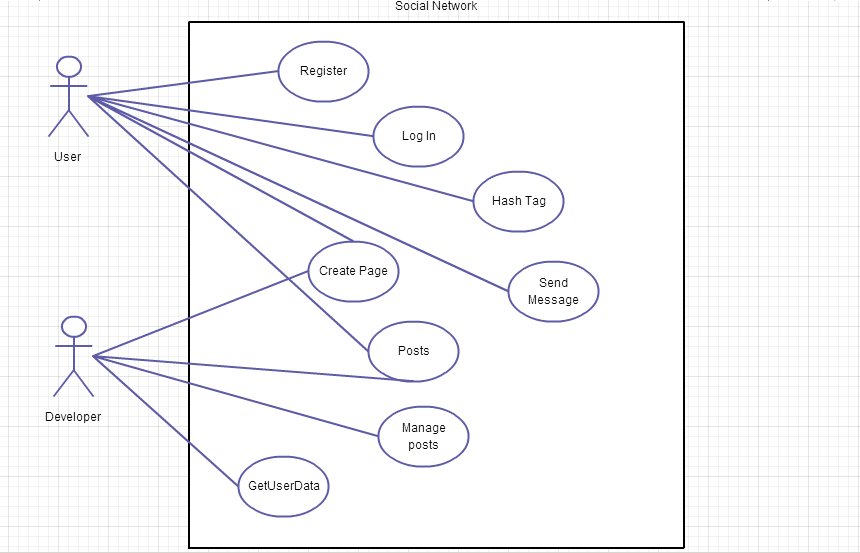
**Non Functional Requirements:**

* + - * **Quality Requirements:**
* Quality requirements ensure the system possesses quality working fine according to usability, efficiency, reliability, maintainability and reusability. These requirements *constrain* the design to meet specified levels of quality.
* **Response time:** is to require that the system gives results or feedback To the User in a certain minimum time. And this requirement is related to software systems that process a lot of data. The fewer response times the more efficient software system.
* **Example:** 30 seconds.
* **Throughput**: for software systems which concerned with computations or transactions per minute, and this is a good way to specify throughput. It’s better to the social media software’s throughput to be few minutes as the average of operations per minute.
* **Example:** 50 computations/minute
* **Resource usage**: For software systems like social media that use a huge amount of memory and network bandwidth, as these are considered as resources, you should specify the maximum amount of each resource that the software system will use. This provides an ability to plan hardware upgrades.
* **Example:** one terabyte for memory.
* **Availability:** is a measure to the amount of time that a server is running and available to respond to users. Telecommunications systems have very rigorous availability criteria in this requirement specifically. Social media software must have a high availability to many users anywhere and anytime. And it's like response time.
* **Example:** 30 seconds
* **Recovery from failure:** is the ability of the system to recover the error/failure in few minutes and with a certain minimal loss of data if a failure is happened to the system so, the social media software should be able to recover any error.
* **Example:** if server failure happened, thesoftware triesto recover itin short time.
* **Allowances for maintainability and enhancement:** in order to achieve this, we should make sure that system can be adapted in the future, and then you should describe the expected changes for the next versions, this constraints design and improves its quality and improve its performance too, without adding new obvious functional requirements.
* **Example:** improve the users' functionalities in API social network.
* **Allowances for reusability:** this is desirable in many cases to specify that a certain percentage of the system measured in terms of lines of code.
* **Example:** change code of premium user option in API social network.
* **Platform requirements:**
* This type of requirement constrains the environment and technology of the

System:

* **Computing platform:** is to know the hardware and operating system, to know that it will be able to work on it or not, and this is important to protect the social software from breaking down suddenly and specify any powerful or bad performance components. Then we should choose the best operating system and hardware to work on.
* **Example:** API network may not work on Linux, but it may work on windows perfectly.
* **Technology to be used:** unlike it’s better to give the software designer chance to choose the technology he want to work on, but technology sometimes has an important effect in software, then we decided the best technology for social media software, we need to train all the team members on that technology. To avoid technology problems later.
* **Example:** the 2G RAM may not help to deal with the API, but 8G RAM will definitely deal with it.
* **Process requirements**
* **Development process to be used:** to get a high quality social media software you should do a lot of development steps which assure to you the quality of the software. Such as particular approaches to testing. After every testing step you should get the customer opinion to save your time.
* **Example:** doing user login part and testing it on a lot of machines.
* **Cost and delivery date:** it’s important to agree about the suitable date for you, to finish the software development perfectly, if you didn’t get the suitable date it may cause software crises as it happened before.
* **Example:** 3 months is enough to make a perfect API if you work with team, and 6 months if you work alone.

**System Models:**

1-Use Case Model: 

2-Use Case Tables:

2.1 SN\_1 Register

The purpose of this use case is having account in social network.

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_1 | |
| Use Case Name: | Register | |
| Actor: | User | |
| Pre-conditions: | The user register his/her Data (Name , password ,e-mail , Age , Gender) | |
| Post-conditions: | The user already has a account in social network | |
| Flow of events: | **User Action** | **System Action** |
| 1-User enters Name. (This operation done by front-end not by you) |  |
|  | 1-System enters user name. |
| 2- User enters Email. (This operation done by front-end not by you) |  |
|  | 2- System enters user email. |
| 3- User enters Age. (This operation done by front-end not by you) |  |
|  | 3- System enters user age. |
| 4- User enters Name, email, Age, Gender. (You should start from flow of events from here, you have now user personal data to register) |  |
|  | 4- System Verify user data. |
| 5- User Select Gender from the list |  |
|  | 5- System Verify user data. |
| 6-press Sign-up |  |
|  | 6- The system has a new data stream |
|  | |
| Exceptions: | The user must to write all data; else the system shows a warning message. | |
| Includes: |  | |
| Notes and Issues: |  | |

Table 1: Register use case

2.2 SN\_2 Log in

The purpose of this use case is entering to your account.

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_2 | |
| Use Case Name: | Log in | |
| Actor: | User-Owner | |
| Pre-conditions: | The user writes his/her Data (Name, password). | |
| Post-conditions: | The user enters to his/her account. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enters Name. |  |
|  | 1-System enters user name. |
| 2- User enters password. |  |
|  | 2- System enters user password. |
| 3- User Enter Name, password then log in. |  |
|  | 3- System Verify user data. |
| Exceptions: | The user must to write data correctly; else the system shows a warning message. | |
| Includes: | SN\_1 | |
| Notes and Issues: |  | |

Table 2: Log in use case

2.3 SN\_3 Posts

The purpose of this use case is writing a post to share it with friends.

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_3 | |
| Use Case Name: | Posts | |
| Actor: | User | |
| Pre-conditions: | The user writes a status and presses enter. | |
| Post-conditions: | The status is published in home and other friends can see it | |
| Flow of events: | **User Action** | **System Action** |
| 1- Click in write something window. |  |
|  | 1-The system allows to write. |
| 2-the user click enter |  |
|  | 2- System Publishes post in home. |
| Exceptions: |  | |
| Includes: | SN\_2 | |
| Notes and Issues: |  | |

Table 3: Posts use case

2.4 SN\_4 Create a page

The purpose of this use case is creating a page to post a status the friends and other can see it.

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_4 | |
| Use Case Name: | Create a page | |
| Actor: | User or Developer | |
| Pre-conditions: | The user enter Create a page. | |
| Post-conditions: | The user already has a page and he can publish anything he wants on it. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Click in Create a page. |  |
|  | 1-The system print types of page. |
| 2-the user choose type of page. |  |
|  | 2- System shows the windows should be filled by data. |
| 3-Enter Started |  |
|  | 3-The system show a message that now you have a page. |
| Exceptions: | The user must to write all data; else the system shows a warning message. | |
| Includes: | SN\_2 | |
| Notes and Issues: |  | |

Table 4: Create a page use case

2.5 SN\_5 hash tag

The purpose of this use case is writing a something to share with others

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_5 | |
| Use Case Name: | Hash Tag | |
| Actor: | User | |
| Pre-conditions: | The user writes hash then the word. | |
| Post-conditions: | The hash tag is published and all can participate by like or comment. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Click in write something window. |  |
|  | 1-The system allow to write. |
| 2-the user press enter |  |
|  | 2- System publishes hash tag in home and all can see it. |
| Exceptions: |  | |
| Includes: | SN\_2 | |
| Notes and Issues: |  | |

Table 5: Hash tag use case

2.6 SN\_6 Send a message

The purpose of this use case is sending a message to another friend.

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_6 | |
| Use Case Name: | Send a massage. | |
| Actor: | User | |
| Pre-conditions: | The user writes a message. | |
| Post-conditions: | Another user reply on it. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Click in message window. |  |
|  | 1-The system show the friends that I can send to them a message. |
| 2-Choose a friend. |  |
|  | 2-the system show a window to write to your friend. |
| 3-Press enter |  |
|  | 3-The system send it. |
| Exceptions: | The message is sent to a friend who registers in your list friends. | |
| Includes: | SN\_2 | |
| Notes and Issues: |  | |

Table 6: Send a message use case

2.7 SN\_7 Manage posts

The purpose of this use case is managing posts.

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_7 | |
| Use Case Name: | Manage posts | |
| Actor: | Developer | |
| Pre-conditions: | The developer asks for user posts. | |
| Post-conditions: | The developer shows posts between two dates. | |
| Flow of events: | **User Action** | **System Action** |
| 1- The developer enters username. |  |
|  | 1-the system searches username in database. |
| 2- Developer enters two dates for posts. |  |
|  | 2- This system gives off all posts at this time. |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

Table 7: Manage posts use case

2.8 SN\_8 Posts

The purpose of this use case is developer is able to write in user’ pages.

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_8 | |
| Use Case Name: | Posts | |
| Actor: | Developer | |
| Pre-conditions: | The developer is able to see , manage user’ page | |
| Post-conditions: | The status is published in user’ page | |
| Flow of events: | **User Action** | **System Action** |
| 1. The developer writes a user name |  |
|  | 1. The system finds the name in the database. |
| 2- the developer writes a post in user’ page |  |
|  | 1. The system publishes the post to user’ page. |
| Exceptions: | If the user name is not in database, the system shows a message that there is no user name found | |
| Includes: |  | |
| Notes and Issues: |  | |

Table 8: Posts use case

2.9SN\_9 Get User Data

The purpose of this use case is getting user data.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | SN\_9 | | |
| Use Case Name: | GetUserData | | |
| Actor: | Developer | | |
| Pre-conditions: | Developer asks for certain user data. | | |
| Post-conditions: | Data will be presented to the developer. | | |
| Flow of events: | **User Action** | **System Action** | |
| 1- The developer enters username. |  | |
|  | 1-The system displays the data that belong to the user of the developer. | |
|  |  | |
|  | |  |
| Exceptions: | If user name do not exist the system will show that there is no data in the database. | | |
| Includes: |  | | |
| Notes and Issues: |  | | |

Table 9: Get User Data use case

3-Ownership Report:

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Introduction | **Noha Ashraf Ibrahim** |
| Functional requirements | **Heba Sayed shmardy** |
| Nonfunctional requirements | **Noha Magdy Saber** |
| Use case model | **Neama Fouad Mohamed** |
| use case table + ownership report | **Nehal Khaled Hafez** |