

**CS251 – Software Engineering I**

|  |  |
| --- | --- |
| **Staff Memebers** | **Email** |
| **Dr Mohamed El Ramly** | **m.elramly@fci-cu.edu.eg** |
| Eng Catherine Bedrossian | c.bedrossian@fci-cu.edu.eg |
| Eng Sara Tarek Ali Othman | s.tarek@fci-cu.edu.eg |
| Eng Mohamed Samir | m.samir@fci-cu.edu.eg |
| Eng Marwa Nabil | m.nabil@fci-cu.edu.eg |
| Eng Yomna Magdy Mohamed | yomna@fci-cu.edu.eg |
| Eng Desoky Abd El-qawy | d.abdelqawy@fci-cu.edu.eg |
| Eng Omar Khaled Ali Ragab | o.khaled@fci-cu.edu.eg |
| Eng Mohamed Elarnaoty | m.elarnaoty@fci-cu.edu.eg |
| Eng Mostafa Saad | m.saad@fci-cu.edu.eg |

Contents

[Team 3](#_Toc402452669)

[Document Purpose and Audience 3](#_Toc402452670)

[Introduction 3](#_Toc402452671)

[Software Purpose 3](#_Toc402452672)

[Software Scope 4](#_Toc402452673)

[Definitions, acronyms, and abbreviations 4](#_Toc402452674)

[Requirements 4](#_Toc402452675)

[Functional Requirements 4](#_Toc402452676)

[Non Functional Requirements 4](#_Toc402452677)

[System Models 5](#_Toc402452678)

[Use Case Model 5](#_Toc402452679)

[Use Case Tables 5](#_Toc402452680)

[Ownership Report 6](#_Toc402452681)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
|  | Hazem Ashraf | Hazem.ashraf.mongd@hotmail.com | *01281330058* |
|  | Aya Alaa | aya\_yoyo1471@hotmail.com | 01008151924 |
|  | Ibrahim Desouky | ibrahim.desouky44@yahoo.com | *01141205894* |
|  | *Ahmed Tarek El-Kordy* | ahmedelkordy25@yahoo.com | 01146424382 |

# Document Purpose and Audience

* **What's this Document?**

- This document is to present a detailed description to the related readers of this document about "Social Network back-end API". It will explain the purpose and features of the system, It’s interfaces, all functionalities of the system, constraints and restrictions that's agreed among stakeholders and developers and How the system actors interact with the system, and It will be proposed to the product owner for its approval.

* **Audience list:**

-Product owner

-Project manager

-Editor

-Customer

# Introduction

## Software Purpose

## - Making a back-end API Social Network to make people all over the world to interact and telecommunicate together in different types of the interactions and telecommunications.

## Software Scope

## - This software will be a back-end API social network.

## - You can sign up to this social network system and you will be registered to sign in later.

## - It will make people's to stay connected simply and share every moment together.

## - You can easily catch up with friends with their photos, videos and their posts.

## - You can categorize as a (normal, premium user) to specify your authorities to the system.

## - Making posts with It's hashtag to make the post categorized by its importance.

## - Like and sharing your friends posts who are in your friend list.

## - send messages to your friends and displaying them if you need it , It can be between two or more.

## - You can make a fan pages to your product to make your business.

## - User can make a group and be admin on it and give any other users to be administrator on it too.

## - making posts on your page and specify it's privacy (to who will see it), Or make posts in you profile page and your friend profile page.

## - Anyone can tag their friends to notify them to see his post.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| Terms & Abbreviations | Definition |
| Social Network | Individuals or groups that interacts together on an interconnecting network ( e.g. catch up friend's photo, videos or status and send messages..etc). |
| Back-end API | It's a term in computer programming that's abbreviated "Application Programming Interface(API) " , that's specifies a software component in terms of it's operation , Allowing both definition and implementation to vary without comprising each other. It deals with the system and allows programmers to deal with it. |
| Post | It's a text that describes the status of the post writer. |
| Normal User | A user that has a limited rights or authorities on the system. |
| Premium User | A user that has some authorities more than the normal user. |
| Administrator | Anyone can manage and admin the system. |
| Profile | It's a page that contains all information about the profile owner. |
| Fan page | A web page that describes a product or a popular thing which has fans. |
| Group | Some of people who interact together. |
| Hashtag(#) | It categorizes the posts that’s written (e.g. write post about "#Sport") it will put this post with posts that has the same hashtag #Sport. |
| Tag (@) | Notifying friends by write tag symbol (@) and his name (e.g. @Hazem Ashraf). |
| Friend List | A list of your friends that you are choosing. |
| Hashtag importance | It will be leveled by its user (e.g. Premium user has posts with more important than Normal User). |
| Sharing a post | An action that allows you to duplicate a post to anywhere authorized you want. |
| Like a post | An action that describes the state of like to a certain post. |
| Sign up | Procedures to make an account on the system with your information needed. |
| Sign in | Procedures to enter to the system with your ID and Password. |

# 

# Requirements

## Functional Requirements

Introduction –

This document contains the requirements for the Social Network Back End API. These requirements are organized by the features discussed in the project problem statement. Features from project problem statement are then refined into use case diagrams and to sequence diagram to best capture the functional requirements of the system.

## 1.1 Signing up for the server

**Description:**

The user should be able to sign up to our social network through any device connected to the internet. The user must provide user-name, password and e-mail address. The user can choose to provide a regularly used phone number, gender, birthday …. Etc.

## 1.2 Signing in on the server

**Description**:

Given that a user has registered, then the user should be able to log in to the server by entering user name and password.

## 1.3 Creating customer profile.

**Description:**

Given that a user has logged in, then he/her should be able to create personal profile.

## 1.4 Create fan page

**Description:**

Users may need to create fan page for specific entity or product. With this component user should be able to create and manage user pages and maintain some important attributes (e.g. number of users like specific page)

## 1.5 Create a group

**Description:**

User should be able to create and manage different types of groups (for example: public group, private groups, … ) each group is managed by set of users (including user creates the group).

## 1.6 Writing a post

**Description:**

Users may want to share their status in social network by writing **posts**, Posts is one of the building blocks of social media content, Post (in our social network) defined as text content written by specific user, any other **friend user** can "**like**" this text content or **share** it on his/her personal profile page.

User can write post in his/her personal page or in joined groups. If user owns a page, user can write a post in his/her page. With this component you should be able to create and manage different types of posts (user post, page post, post written by premium users, …). Also you should maintain some important attributes for post (number of likes, number of shares)

## 1.7 Hashtag

**Description:**

Users use hashtags to categorize the content of specific post, if user wants to write a post about "education", user may put the word #education (# the symbol of hashtag) in the post to categorize this post as education post. If another user wants to read some posts about education, so user may request all posts contains #education phrase.

All posts containing hashtags will be sort according to post importance (for examples, posts written by sponsored users should be more important than posts written by normal users).

## 1.8 Messages

**Description:**

User can send message to another user(s) in his/her friends list.

Using this component, you should be able to create message from user to any other user(s), or get messages between 2 users, or get messages in specific message group (which contains more than 2 users)

## Non Functional Requirements

**Definition**: Non-functional requirements are not non-functional at all.

Rather, they describe various quality factors, or attributes, which affect the functionality's effectiveness. They do not exist in the abstract but only with respect to relevant functionality.

They are often called "ilities," because many end in "ility," such as, usability, reliability, and maintainability..

2.1 **Designing**: Better component design to get better performance at peak time.

2.2: **Performance: The** performance constraints specify the timing characteristics of the software.

Certain tasks or features are more time-sensitive than others.

2.3 **Scalability**: System should be able to handle a number of users. For e.g.

Handling around thousand users at the same time..

2.4 **Usability**: Simple user interfaces that a layman can understand.

2.4 **Speed**: Speed of the system should be responsive i.e. response to a particular action should be available in short period of time. For e.g. on registering as a new user the

Notification about the approval is sent immediately to the respective user through email.

Updating the account takes few seconds for the changes if the entry is not starred.

2.5 **Availability**: System should be available for 24hours a day.

A system’s availability, or “uptime,” is the amount of time that it is operational and available for use.

This is specified because some systems are designed with expected downtime for activities like database upgrades and backups .

2.6 **Efficiency**: Specifies how well the software utilizes scarce resources: CPU cycles, disk space, memory, bandwidth, etc

2.7 **Reliability**: Reliability specifies the capability of the software to maintain its performance over time.

Unreliable software fails frequently, and certain tasks are more sensitive to failure .

2.8 **Robustness**: A robust system is able to handle error conditions gracefully, without failure.

This includes a tolerance of invalid data, software defects, and unexpected operating conditions.

2.9 **Maintainability**: There should always be regular system maintenance. You should think about how much time

You need to do maintenance, and how and when you will notify use .

2.10 **Good user interface**: Any social networking site must provide an easy handling of its tools and other features of the system.

If the features or services are difficult to understand then there are more chances of users leaving such social networking site and looking for better ones. Any user interface requirement will demand application navigation and good user assistance all throughout the application. Overall the look and feel of the online social networking site must be good enough for the user to access it in an easier way.

2.11 **Safety**: Safety is the state of being safe. The condition of being protected against consequences of failure, damage, another event which could be considered non-desirable. Some kind of precautions must be kept in hand in order to face

Such failures like system crash, profile deleted, photo album being removed or hacking your credentials.

2.12 **Maintenance** **cost**: The maintenance cost for such online social network must also be considered. If developing such social network requires more money then only way to gain it back is only through the advertisements or by selling user data.

2.13 **security**: Security requirements are included in a system to ensure:

-Unauthorized access to the system and its data is not allowed

-Ensure the integrity of the system from accidental or malicious

Damage.

2.14 **Portability**: Portability specifies the ease with which the software can be installed on all necessary platforms, and the platforms on which it is expected to run.

2.15 **Reusability**: Many systems are developed with the ability to leverage common components across multiple products.

Reusability indicates the extent to which software components should be designed in such a way that they can be used in applications other than the ones for which they were initially developed.

2.16 **Flexibility**

If the organization intends to increase or extend the functionality of the software after it is deployed,

That should be planned from the beginning; it influences choices made during the design,

Development, testing, and deployment of the system

Flexibility is the ease with which the system can be reused, deployed, and tested.

2.17 **Audit**-**ability**

When something goes wrong, you need to understand the root cause of it.

The problem is that you hardly remember to have all the checkpoints in the process, all the exceptions logged, and to ensure that the subsystem to support it does not interfere with your application performance.

# System Models – Use Case Model C:\Users\Hazem\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Untitled-2.jpg

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.1 | |
| Use Case Name: | Sign Up | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User will interact with the server interface. | |
| Post-conditions: | Account is created | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters personal information (name, gender, age,…) |  |
|  | 2- System Verify user data |
|  | 3-System Requires some information for privacy |
| 4-User Enter these information |  |
|  | 5-System Creates The Account |
| Exceptions: | Users Enter un Valid Data Or There is an Existing Account | |
| Includes: | Log in | |
| Notes and Issues: | User should make sure that the Data he enters is Valid | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.2 | |
| Use Case Name: | Log in | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User must create an account first by signing up | |
| Post-conditions: | User is logged in | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters his user name or e-mail and password |  |
|  | 2- System check the validity of this data |
| 3-User logs in to the Social network |  |
|  |  |
|  |  |
| Exceptions: | Users Enter un Valid Data Or un legal user is logging in "Hacker" | |
| Includes: | Creating a fan page , creating a group , writing a post , writing a hash tag and Sending messages | |
| Notes and Issues: | Make sure that you Enters the Right data to log in or your account will be blocked until you answer several questions | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.3 | |
| Use Case Name: | Create your profile | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User must Log in first | |
| Post-conditions: | Profile is created | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters Some Personal information (school , Languages , Woks , … ) |  |
|  | 2- System Saves this Information |
| 3-User Sets a profile picture |  |
|  | 4-System check the validity of the picture |
| 5-Use can update his Profile from time to another |  |
| Exceptions: | Un legal user is logged in | |
| Includes: | nothing | |
| Notes and Issues: | Make sure that you Write a legal information about yourself | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.4 | |
| Use Case Name: | Create a Fan Page | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User must Log in First | |
| Post-conditions: | Page is Created | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters the type of the page and its Purpose |  |
|  | 2- System Requires some information (name , Cover photo , Description ) |
| 3-User Invites People to like his Page |  |
|  | 4-System allows him to invite his friends only |
|  |  |
| Exceptions: | This page is un-appropriate , the system won't allow you to create it | |
| Includes: | none | |
| Notes and Issues: | none | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.5 | |
| Use Case Name: | Create a Group | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User must Log in First | |
| Post-conditions: | Group is Created | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enters Some Description about the group and its purpose |  |
|  | 2- System Requires some information (name , Cover photo , Privacy ) |
| 3-User invites people to join group |  |
|  | 4-System allows him to invite his friends only |
|  |  |
| Exceptions: | None | |
| Includes: | None | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.6 | |
| Use Case Name: | Write a Post | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User must Log in First | |
| Post-conditions: | Post is Written | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Writes the post |  |
|  | 2- System check type of the post ( normal user post, Page, post, group post, Premium user post) |
| 3-user choose the privacy of the post |  |
|  | 4-The system will publish this post on the user’s profile |
|  | 5-The system should allow another users (just only friends of the user wrote the post) to like and share and also comment on the published post an d count number of likes |
| Exceptions: | There is no Internet Connection, Or you are not member in the group | |
| Includes: | You have to be member in the group , Liked that specific page | |
| Notes and Issues: | none | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.7 | |
| Use Case Name: | Hash Tag | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User must Log in First | |
| Post-conditions: | Hash tag is Written or Found | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Write a Hash tag Post by including this Symbol "# " or Search for Specific Hash tag |  |
|  | 2- System check type of the post ( normal user post, Page, post, group post, Premium user post) |
| 3-user choose the privacy of the post |  |
|  | 4-The system will publish this post on the user’s profile |
|  | 5-The system should display all the posts about That hash tag categorized by the published user as the premium user’s post will Preceded the ordinary users’ post |
| Exceptions: | There is No Hash tag For your Statement | |
| Includes: | Searching For Specific Situation in Education as an example | |
| Notes and Issues: | none | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1.8 | |
| Use Case Name: | Writing a message | |
| Actors: | Normal User, Premium User | |
| Pre-conditions: | User must Log in First | |
| Post-conditions: | Message is Sent | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Write a message or get a message |  |
|  | 2- System checks for the type of message (message between 2 users or message from a user in a group) |
| 3-user can respond to the message |  |
|  | 4-system sends the Respond |
|  |  |
| Exceptions: | The other user is not in your friend list, Or you are not member in the group, or the user Deactivated his account | |
| Includes: | none | |
| Notes and Issues: | User can't Send messages to someone he doesn't know, he must be in his friend list | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Document Purpose and Audience + Introduction  ---------------------------------------------------------------------------Functional Requirements | *Hazem Ashraf*  *------------------------------------------*  *Ahmed Tarek ElKordy* |
| Non- Functional Requirements  -------------------------------------------------------------------------- Use Case Model + Table | *Ibrahim Desouky*  *------------------------------------------*  *Aya Alaa* |