**Software Requirements Specifications**



Course Name : Software Engineering 1

DR : Mohamed El-Ramly

TA : Mohamed Samir

Year : 2014-2015

Leader Name : Ahmed Mamoud Fahmy

Leader Mail : [a7med.fahmy94@gmail.com](mailto:a7med.fahmy94@gmail.com)

Leader Phone : 01121466814

Project On Github : [GitHub](https://github.com/a7medfahmy94/SE_Project)

Contents

Team 1

Document Purpose and Audience 2

Introduction 2

Software Purpose 2

Software Scope 2

Definitions, acronyms, and abbreviations 2

Requirements 2

Functional Requirements 3

Non Functional Requirements 3

System Models 3

Use Case Model 3

Use Case Tables 3

Ownership Report 4

Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20120065 | Ahmed Mahmoud Fahmy | a7med.fahmy94@gmail.com | 01121466814 |
| 20120441 | Hala Mohamed Osman | hala.mohamed199@gmail.com |  |
| 20120282 | Leila Saeed ahmed | leilasaeed2013@gmail.com |  |

Document Purpose and Audience  
  
This document is for the purpose of analyzing the Social Network project.  
It introduces the SRS for the proposed system.  
This document is for developers to read and discuss, it will be a good start for any   
new developer joining the team.  
Also , any technically experienced person can read,discuss and review this document.

Introduction

Software Purpose

the purpose of the system is to develop an API for social network that allow users to stay connected and interact with each other.

it is simple and clear program and you can use it easily and don't require a big effort to use

it is free and available for all people

it'll offer basic services :create account , add friends,make groups and pages,message friends and receive messages ,etc..

it'll offer some features for premium users.

Software Scope

This is an API for any social network to depend on.

This software focuses on the basic operations that a user of a social network can perform, such features include:

1. create a new account (normal , premium)
2. add,view,delete a friend
3. make,edit,remove,like,share a post
4. upload some photos
5. create,join a group (private , public)
6. create,like,share a page and post to it (if administrator)
7. like,share,comment to friends' posts or pages
8. add a new hashtag to his posts or view posts with specific hashtag
9. message his friends and receive messages from his friends

it will not go through the complicated matters of advertising or any other monetary features (except for premium users)  
groups can have more than one administrator, but pages only have one.

Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| Acronyms,abbreviations | Definition |
| **URL** | **Uniform Resource Locator**  **(internet address)** |
| **API** | **An application programming interface** |
| **chat** | **chat is a place where you can talk to other people online** |
| **pp** | **Pp is a profile picture** |
| **hashtag** | **It’s symbol (#) to categorize the content of specific post** |
| **share** | **Display the user’s posts,pages**  **To the other users** |
| **post** | **It’s statements which user write them to express something specific** |
| **page** | **Page is a public profile created by businesses, organizations and celebrities** |
| **group** | **Group is a page created for an organization or business to promote activities** |

Requirements

Functional Requirements

* The system can create new users, normal or premium , if given the appropriate parameters => (user name,email,gender,premium?)
* the system stores the information of the user and allows for updating/deleting/inserting any additional information => (profile picture , birth date , address)
* the system offers different features for different types of users (to be discussed with PO)
* A user can create groups (given that he is authorized to do so), there are different types of groups (public groups, private groups)
* The group owner (who created the group) can add new members and promote any number of them to manage the group.
* Users can create pages for general purposes or for specific purposes.
* A page is managed by only the creator (to be further discussed)
* anyone can view the number of likes for any page.
* Anyone can like/share/comment on any page's post or activity.
* Any user can make a post , share it In a group , like his/her friends' posts and share and comment to them
* Premium users can mark their posts as global , which means not only his/her friends view them , but it goes on the main stream of the whole system
* A page can make posts with it's name by the owner
* Any post is stored along with its number of likes , shares and comments
* Any post can have a Hashtag in its body , which serves as a reference for all other related posts of the same topic (contains the same Hashtag)
* Any user can retrieve a list of all posts associated with a specific Hashtag, sorted in order of importance (sponsored posts goes before normal posts)
* Any user send/receive a message to/from any one of his friends
* Any group of users can create a group chat at which all of them can take part in the conversation
* The system stores conversations between users , but doesn't supply them to any one besides the participants of the conversation

Non Functional Requirements

* **Non-functional requirements describe how the system works**
  + **E.g. Withdraw operation will be done within 20 second. Network is using secured protocols. System allows up to 30,000 withdrawal per minute.**
  + **Think about the operation / system quality**
* **There are too many non functional requirements. Read in [wiki](http://en.wikipedia.org/wiki/Non-functional_requirement) and Chapter 4. Pick the suitable ones for your system. Non-functional requirements must be VERIFIABLE, i.e., MEASURABLE.**
  + **Some Types as just examples: Usability, Reliability, Performance, Security, Scalability, Portability, Maintainability**
  + **Select the sutiable ones, for each one write the details**
  + **Be realistic **

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * **Withdraw operation will be done within 20 second** |
| **Scalability** | * **System could be horizontally ....** * **And** * **In addition** |

System Models

Use Case Model

* **Using UML, write the use case model expressing the system actors & operations**

Use Case Tables

* **Using below table template, for each requirement write a use case table that shows user/system interaction**
  + **If one requirement is so big, you could divide it to more than table**
  + **If some requirements are not major, you could plugin them in other senario**
    - **E.g. you may not do LogIn Usecase table as it is simple functionality**
* **Flow of events should be very detailed**

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: |  | |
| Actosr: |  | |
| Pre-conditions: |  | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter Card and Password. |  |
|  | 2- System Verify user data |
| 3- User Select Vodafone from the list |  |
|  | 4- System retrieves Vodafone bills |
| and so on |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

Ownership Report

* **Remove the following notes and any red notes**
* **For every item in this document, write the owners. If someone is owner of something, s/he understands it 100.%**
* **Team leader must verify the table with the team members.**

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |