Software Requirement Analysis



**COMS 3002**

Group 9

Tracking Interconnected Facebook Links: Using Graph Database Neo4j

28-August-2017

# Introduction

## Purpose

Social media plays a vital role in everyday people lives. There is enormous information about individuals and how they relate to one another. This information is useful to individuals, advertisers, politicians and many other organisations. The purpose of our software is to provide means to get links between individuals in social media. Our first focus will be analysing Facebook links using Neo4j database via the website.

## Scope

* Create a database using Neo4j to store our social media data (Clifford).
* Find out about the legal implications of using Facebook data (Lindiwe).
* Register with Facebook to enable them to give us access to their data (Lindiwe).
* Create a beautiful easy to use web interface (Thomas/Lindiwe).
* Testing of the software (Clifford/Thomas/Lindiwe).

## Definitions

SDLC: Software Development Life Cycle.

UC: use case.

## Overview

For this program we will be using agile SDLC.

The idea is to get a user to log in to our web site with their Facebook credentials and after doing the necessary legal verifications, take the data from their Facebook account to our database and do some data analysis of the various links they have on Facebook and then display this in a friendly manner.

# Overall description

## Product Perspective

Since this is the first software we are producing, there no other programs to interface with. With this program you will be able to understand your interaction in Facebook as an individual.

The program will be expected to deliver on the following:

* Find the person who like your post the most.
* Find the people who unfriended you within the given period.
* A Facebook friend who commented the most in that given period.

## Product functions

* This program will be designed in Client-Server model.
* The front end will be through the web and the back end will be done by Neo4j.
* The program is expected to have a response time of no more 15 seconds 90% of the time.
* The program is expected to do at least three different data analysis.

## User characteristics

* The user will have to be willing to share their Facebook data with our program.
* The program should be able to handle at least five (5) users without affecting the user experience.

## General Constraint

* Obtaining permission from Facebook to use their data.

## Assumptions and dependencies

* Facebook will grant us access to the data.
* There will be a server where we can run the program.

# Detail requirements

## External Interface Requirements

### User interfaces (Thomas)

### Hardware interfaces (Clifford)

The only Hardware interface that will be required for this project is the computer that will be used to program and save the database for this project.

### Software Interfaces (Clifford)

Neo4j will be used as the database for this project and the browser afforded by Neo4j will be the first choice if feasible as the user interface. At the moment Java or Python are the front runners to be used for the back-end programming. Java because Neo4j supports native Java and Python because many data science libraries are written in Python.

## Function requirements

### Back-end requirement (Clifford)

#### UC1: find the person who like your post the most.

Primary actor: The account holder.

Precondition: The account holder has logged in.

Main success scenario:

The account holder click on the link and the system tells his/her the top 3 people who liked his status.

Exception scenario:

The system prints some sad emoji indicating that no one has liked his/her status within that given period.

#### UC2: Find the people who unfriended you within the given period.

Primary actor: The account holder.

Precondition: The account holder has logged in.

Main success scenario:

The account holder click on the link and the system tells him/her the people who have unfriended him in Facebook.

Exception scenario:

The system prints some smiley emoji indicating that no one has unfriended them in the given period.

#### UC3: A Facebook friend who commented the most in that given period.

Primary actor: The account holder.

Precondition: The account holder has logged in.

Main success scenario:

The account holder click on the link and the system tells him/her the top 3 friends that commented the most in the given period

Exception scenario:

The system prints some sad emoji indicating that no one has commented on their status in the given period.

### Front-end requirement (Thomas)

The front interface will be through the Neo4j browser.

Different interaction, can be represented by colour connections, where repeating an action could be represented by the thickness of the line.

#### UC1:Likes

I’m not sure if the change in thickness will be obvious enough, need to think of something perhaps more noticeable. Perhaps different style lines or different colour

#### UC2:Unfriends

Show newly established friendships in a colour and recent unfriends as a different colour, maybe a broken line.

#### UC3:Posts

Similar to UC1 represent frequency through thickness if obvious enough

## Performance Requirements (Lindiwe)

## Design constraints (Lindiwe)

## Attributes (Lindiwe)

## Other Requirements