

Enhancing Social Media Experience by Usage of User-Defined Nicknames as Additional Identifiers for Online Interaction

Akshay Aggarwal*

Department of Information Technology, Vellore Institute of Technology, Vellore, India

*Corresponding author: akshyantimatter@gmail.com

Abstract Recent years have seen the rise of various social media platforms that allow for content sharing and social networking. Though some social media platforms allow identifying group tags or channel tagging using ‘@’ or ‘#’ symbols respectively, there has been, however, a lack of user customization in the field since a user has to remember a person by the name they give on their social media during sign up. In this paper, we seek to propose an alternative basis by which users would be able to save and then search for their friends by the names with which they identify them in real life (nicknames). We also discuss how the proposed system can be used in chat systems for identifying users and while tagging friends using the similar concept of ‘@’ symbol, as they find application on social media like Facebook, Twitter, etc.

Keywords: Social Media Engineering, TAO API for Facebook, Facebook name aliasing, Interpersonal interaction on social media, Human Computer Interaction, Nicknames on Social Media, Neo4j, Graph based database

Cite This Article: Akshay Aggarwal, “Enhancing Social Media Experience by Usage of User-Defined Nicknames as Additional Identifiers for Online Interaction.” *Journal of Computer Sciences and Applications*, vol. 4, no. 1 (2016): 1-8. doi: 10.12691/jcsa-4-1-1.

1. Introduction

Interpersonal interaction has always been a very active field of research. Numerous experiments have been done in fields related to psychology, anthropology, etc. that seek to understand human behaviour in detail. There have been experiments that try to model how humans interact with each other. Social Cognitive Theory [1], for example, argues that people are driven not by inner forces, but by external factors. According to the theory, human interaction can be explained by a triadic interaction of behavior, personal and environmental factors.

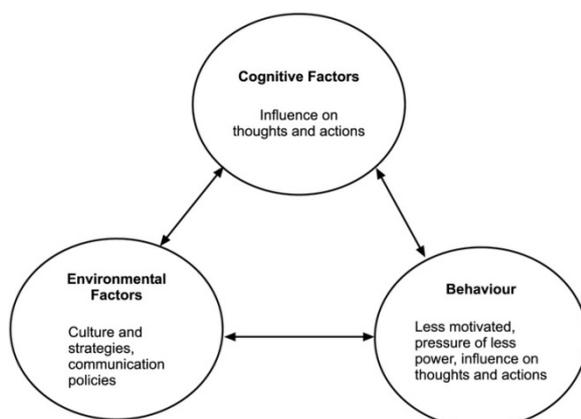


Figure 1. Bandura's Triadic Reciprocal Determinism

However, this is not the only accepted model of human interaction. A number of other studies also note the external influence of society on human behavior. A common feature of such research articles that can be starkly seen is that external factors in particular, their interaction with people around themselves, play a crucial role in defining their personality [2,3,4,5,6].

An essential aspect of human identity, their name and often their nickname, provides us with valuable insights into how they are being perceived by society and also how they perceive themselves. A nickname, as defined by Wikipedia, stands for a shortened substitute for the proper name of a familiar name, person, place or thing, for affection or ridicule. Although nicknames in real life scenario were given analytical detail in 1998 by B.Fortado [7] for a management journal, they had earlier been studied in context of online systems, especially IRC (Internet Relay Chat) by Bechar-Israeli [8] in as early as 1995. As Fortado noted in his work, “these monikers are found to have a wide variety of purposes: including among others, furthering social control, contributing to socialization, marking group boundaries, building camaraderie, catalyzing joking, conveying discontent, cathartically venting frustrations, equalizing social exchanges and adjusting to labeling.”. Bechar-Israeli went as far as to realize importance of nicknames, drawing parallels from a perspective of nicknames as identities, stage names, nicknames and performances, among others. In the studies done by Bechar-Israeli, it was found that 45% of users preferred to use a nickname on IRC that were related to self in some way. In a very recently done

study on prison inmates and their nicknames [9], Black et al. observe “knowing and using nicknames (1) gives a sense of unity among prison peers, while (2) representing their individuality, and (3) facilitating communication among them. Nicknames can be friendly, showing peer approval and in-group unity.” It is worth noting that while some nicknames are self-allotted (users can select nicknames for themselves, as in case of online game handles and anonymous chat groups), some nicknames are peers-allotted too (users are given nicknames by their friends, often as way of shortening their given names. For example- Rob for Robert, etc.). While referring to nickname, we shall include both categories, for the sake of completeness.

With advent of social media networks like Facebook, Twitter, Instagram, etc., socialization is not just restricted to friends and neighbors anymore. Anyone you know can be in your social media connection. Also, it is worth noting that many people we know in real life, we might not know them by their actual name. This is especially the case with people who know each other through gaming portals, gaming lounges and even the friends who are referred to by their nicknames rather than their actual name. The basic idea of this paper is to bring close such people by not forcing them to remember each other’s proper name, but by the nicknames they know each other with.

In this paper, we shall look at different existing technologies and base our implementation to resemble that of Facebook’s database structure. The reasons for the same being- (1) Facebook attracts a wide variety of users to itself¹, so working on a database structure with a huge read-request set should, in general, cover other media with larger or smaller user statistics, and (2) because Facebook uses Graph based database for its functioning, which is already highly optimized for read requests².

The rest of the paper is organized as follows. In the next Section, we focus on the existing work related to the project idea, discussing in brief about already existing aliases features on various social media, the database structure of Facebook and the concepts of tagging using ‘#’ and ‘@’ symbols. We then proceed to describe the proposed functionality with a demo implementation of the same using Neo4j as base database and giving an algorithm for data retrieval to make efficient use of proposed model in Section 3. Section 4 analyses in detail the proposed method, in terms of performance and scalability issues, using the data and results. In Section 5, we explain in brief about how the proposed functionality can be made use of by the users in different test cases. Finally, we conclude the paper in Section 6, followed by the References section listing the sources and references used throughout the length of the paper.

2. Related Work

In this section, we build base to show existing work and technology in relevance with the proposed idea. We start with discussion of aliases features that currently exist on various social media in Section 2.1. We then proceed to

work with special case of Facebook. In Section 2.2, we discuss about database structure of Facebook. We then proceed in Section 2.3 to discuss already existing concept of tagging on various social media using ‘@’ symbol.

2.1. Aliases feature on Social Media

As Johansson et al. [10] notice in their work, users tend to make multiple accounts of themselves on various forums in absence of multiple aliases being supported. In another work by Johansson et al. [11], the reason for the multiple aliases being created is attributed as follows- “Many people who discuss sensitive or private issues on social media services are using pseudonyms or aliases in order to not reveal their true identity, while using their usual, non-private accounts when posting messages on less sensitive issues.” In this news article³, dated October 2, 2014, it has been reported that Facebook has re-allowed the use of aliases for user accounts. Also, Google+ changed its policies related to alias names back in July 2014⁴. It is worth noting that the aliases being talked about in the previous passage refers to self-attributed nicknames, as mentioned in Section 1 of this paper.

There have been no known instances of incorporating peer-defined nicknames in any social media yet. There have been posts saying they work using alternate schemes⁵. However, the question of defining user-defined nicknames comes up often, as in here⁶. The footnote reference also talks about a Chrome extension that might solve the purpose but in the current age of technology, where people access social media through all kinds of devices (Desktops, Laptops, Mobile Phones, PDAs, etc.), on different browsers (Google Chrome, Mozilla Firefox. Thunderbird, Iceweasel, etc.), the provided option of using a Chrome extension is highly biased and thus, not available to everyone. Also, a user using the method defined earlier⁷ would have to find out their friends’ social media links and then proceed further. With social media access being provisioned on different platforms and devices, this might not always be possible. Added to the previous point is the weight that the mentioned feature is nothing less than a headache and thus is a poor design that does not fit well with Fitts’s Law [12] or Welford’s Law [13].

2.2. Database Structure of Facebook

In the 2013 USENIX Annual Technical Conference held in San Jose, CA on June 26-28, 2013, Facebook presented a paper [14] in which they explained in detail the TAO API and how TAO API is used for database management in social structure, as it is used at Facebook. To quote directly, “We introduce a simple data model and API tailored for serving the social graph, and TAO, an implementation of this model. TAO is a geographically distributed data store that provides efficient and timely access to the social graph for Facebook’s demanding

¹ <http://newsroom.fb.com/company-info/Section-Stats>

² <https://m.facebook.com/notes/facebook-engineering/tao-the-power-of-the-graph/10151525983993920>

³ <http://www.reuters.com/article/2014/10/02/us-usa-drag-california-idUSKCN0HR19220141002>

⁴ <https://support.google.com/plus/answer/1228271?hl=en>

⁵ <https://www.facebook.com/notes/update-status-via-blackberryiphoneipad/how-to-tag-someone-on-facebook-with-different-names-not-working/320619914631725>

⁶ <http://www.facebook.com/help/community/question/?id=525000937558538>

workload using a fixed set of queries. It is deployed at Facebook, replacing memcache for many data types that fit its model. The system runs on thousands of machines,

is widely distributed, and provides access to many petabytes of data. TAO can process a billion reads and millions of writes each second.”

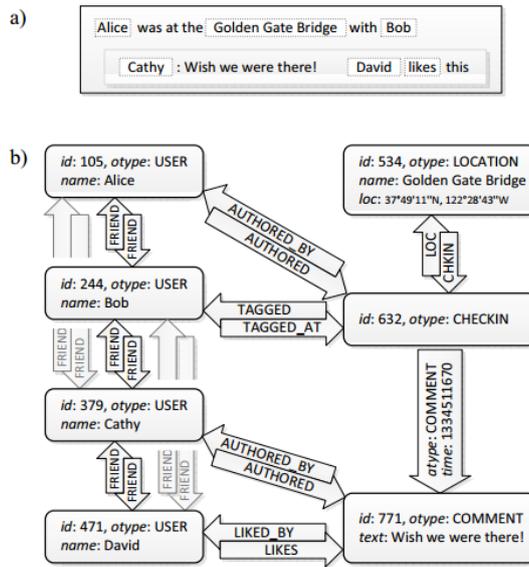


Figure 2. Example used by Bronson N. et al. in their work [14]

In their paper [14], Bronson N. et al. have gone in great detail explaining how the whole system works. They have illustrated the same using an example which showcases how the different objects work in symphony with each

other, elaborating on how objects and associations work. The example they used is depicted in Figure 2, which is taken directly from the paper, without any modifications.

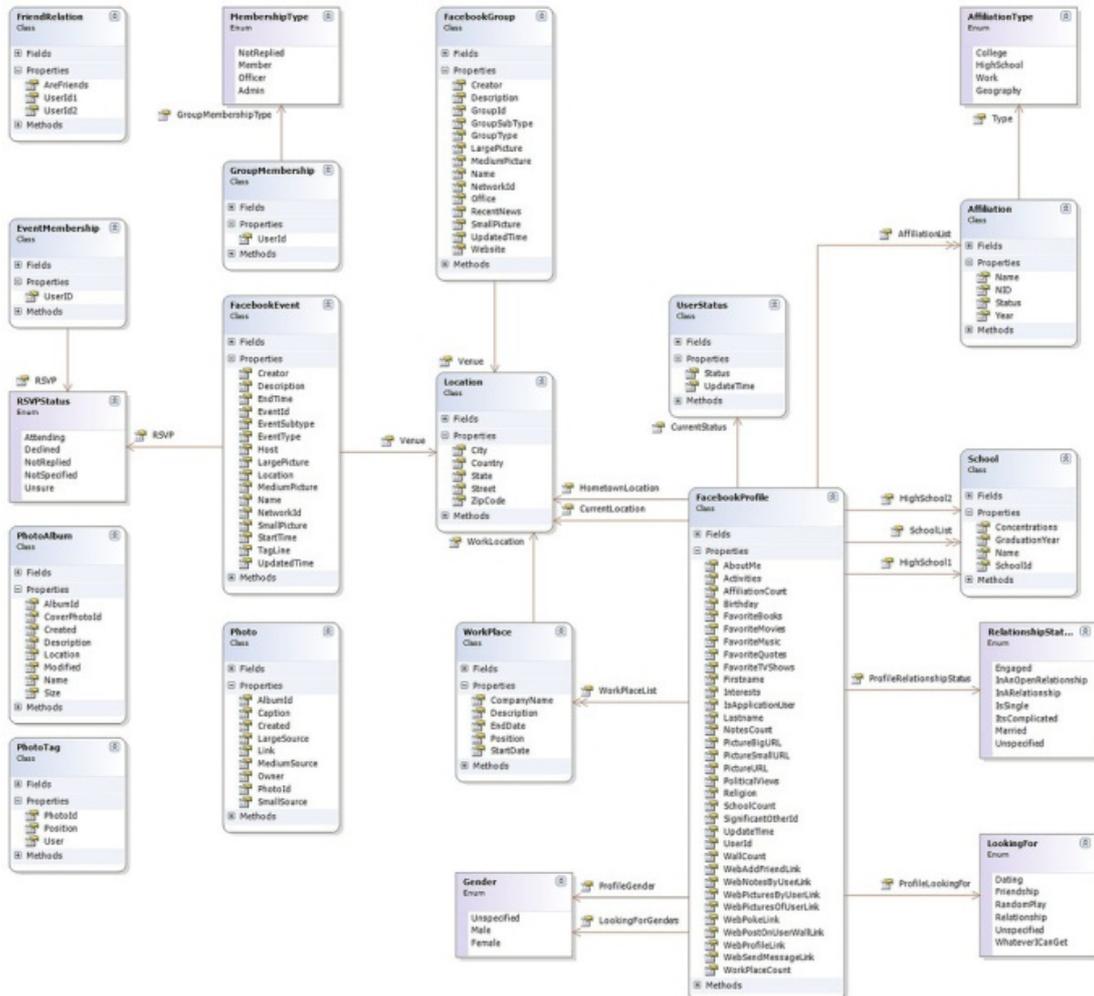


Figure 3. Most widely-agreed upon Class Diagram of Facebook database structure

The paper shows in great detail how the tagging works on Facebook. A presentation video of the same⁷ helps make things even clearer.

Facebook uses TAO API coupled with MySQL database. There are many threads on stackoverflow⁸ seeking the likely relational schema of the database Facebook might use. On a popular vote, Figure 3 seems to be the most logical structure (class diagram) of how database is stored on Facebook.

Figure 3, used above, is not claimed by author and the image being used on stackoverflow has unknown origins.

2.3. Concept of Tagging using '@' symbol

This article⁹ by Wall Street Journal dated March 14, 2013 discusses how Facebook was planning on launching hashtags service to its users as well, following the pattern of Twitter. The article also says- "Facebook has now increasingly moved onto Twitter's turf. The Menlo Park, Calif., social network is prodding users to share more content with the public. In recent years it has mirrored some of Twitter's features by creating "subscriber" lists for users, and allowing people to tag celebrities and brands with the '@' sign."

This patent by Zuckerberg et al. [15] and instructions¹⁰ on how to use '@' symbol for tagging on Facebook explain in detail how the process of tagging works on Facebook. The patent deals with tagging a particular region in a digital media.

3. Implementation

We start with discussion of proposal of database modification in Section 3.1 and then we discuss the accessing of system by users in Section 3.2.

3.1. Modification of Database

We implement the whole system using Neo4j, a graph database. The reasons for choosing a graph database over a relational or an object oriented database are (a) the former are more optimized databases for scenarios where number of read queries far exceed write queries, as is the case with social networks, (b) the read and write times can be easily kept track of and often are in much smaller order than those in relational or object oriented databases, and (c) Facebook uses graph based database and thus we want our implementation to be as close as possible to the industrial scenario. However, we have simplified the implementation a bit. We have not considered interaction with digital media at all. Also, as we do not essentially need a lot of fields defined by user to work with, we use a minimal user structure that involves their first name (fname), last name (lname), sex and age of user. To further simplify the matters, we work with 5 users and build relations between them. We then define these relationships using a friendship matrix and then assign nicknames between users. The user names were generated randomly using an online resource¹¹.

Table 1. Defining users for test set

| S.No. | fname | lname | sex | age |
|-------|---------|---------|-----|-----|
| 1 | Delena | Bowens | F | 24 |
| 2 | Gwenn | Varnado | F | 25 |
| 3 | Genna | Royall | F | 16 |
| 4 | Arnoldo | Dieter | M | 24 |
| 5 | Taylor | Cody | M | 22 |

We next define friendship matrix for the 5 users. A friendship between any two users is marked by '1' and absence of a friendship between users is marked by '0'. We use dash (-) to represent the interaction of a user with themselves.

Table 2. Defining friendship matrix

| | Delena | Gwenn | Genna | Arnoldo | Taylor |
|---------|--------|-------|-------|---------|--------|
| Delena | - | 1 | 0 | 1 | 1 |
| Gwenn | 1 | - | 1 | 1 | 0 |
| Genna | 0 | 1 | - | 0 | 1 |
| Arnoldo | 1 | 1 | 0 | - | 1 |
| Taylor | 1 | 0 | 1 | 1 | - |

Next, we assume the following are the nicknames being used by users to refer to each other in real life. Note that a max of 2 names are allowed for a particular friend and so while some use 2 names for identifying a person, others may not use even a single one.

Table 3. Associated nicknames amongst users

| User | Friend | Nickname1 | Nickname2 |
|---------|---------|-----------|-----------|
| Delena | Gwenn | DeGw1 | DeGw2 |
| Delena | Arnoldo | DeAr1 | - |
| Delena | Taylor | - | - |
| Gwenn | Delena | GwDe1 | - |
| Gwenn | Genna | - | - |
| Gwenn | Arnoldo | GwAr1 | - |
| Genna | Gwenn | - | - |
| Genna | Taylor | GeTa1 | - |
| Arnoldo | Delena | ArDe1 | ArDe2 |
| Arnoldo | Gwenn | ArGw1 | ArGw2 |
| Arnoldo | Taylor | ArTa1 | ArTa2 |
| Taylor | Delena | - | - |
| Taylor | Genna | TaGe1 | - |
| Taylor | Arnoldo | TaAr1 | - |

It is noteworthy that we have associated some assumptions upon which the nicknames mentioned above are based. The following are the assumptions-

- The nicknames are peer-defined and not user self-defined.
- The nicknames are not part of the user's first or last name. Example- Rob as nickname for Robert won't be counted as a nickname.
- It is assumed that the first few characters in nickname(s) do not match the first few characters of the user's first or last name. Example- Robby for Robert does not count as a nickname.
- The nicknames need not be essentially unique. Example- 2 separate people can be nicknamed as 'gunner'.

The reasons for the aforementioned assumptions, being multiple, are as follows-

- Self-defined nicknames by users can be incorporated as 'aliases' in various social networking platforms and thus the paper does not need to address that issue.

⁷ <https://www.usenix.org/conference/atc13/technical-sessions/presentation/bronson>

⁸ www.stackoverflow.com

⁹ <http://www.wsj.com/articles/SB100014241278873233933045783606513453733>

¹⁰ <https://www.facebook.com/help/218027134882349>

¹¹ <http://listofrandomnames.com/>

- If the nickname is just a shortened version of real name, it will automatically suggest the friend whom the user intended, and thus is already implemented and hence not needed to be addressed by the paper.
- If nickname(s) share common first few characters from the real name, while typing the nickname, the system shall automatically recommend intended friend until the characters start to differ.
- A friend can be a part of different social groups, and different friends can be assigned the same nickname in different groups.

TAO API differs from Neo4j in a way that when an association is defined between 2 entities, the reverse association, if needed, is defined simultaneously. However, in Neo4j, we have to define the reverse association manually. We suggest modifying this functionality of TAO API for a single [friends] relationship so as to utilize the functionality in a more optimized manner. Alternatively, if the reverse association generation property is kept, the individual association can be edited at a time to adapt the proposed method. Having assigned nicknames, we proceed to define attributed relationships amongst users by defining [friends] relation between users who are friends of each other. The relationship has attribute(s) that determine the nicknames being used.

Having defined the relationships with attributes as nicknames, the graph looks like Figure 4.

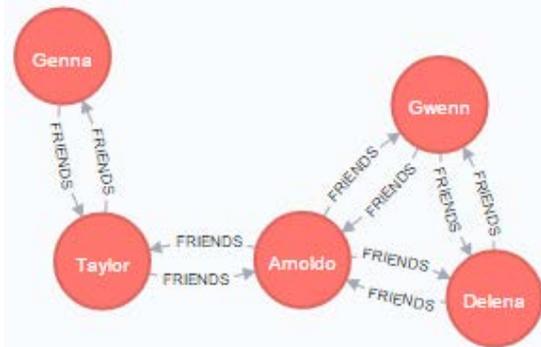


Figure 4. Friendship graph with nicknames as attributes

A sample relation [FRIENDS] between Arnoldo and Gwenn with attributes being their nickname is defined in Cypher, the query language for Neo4j (tutorial available¹²), as-

```
START n=node(*), m=node(*) WHERE
n.fname="Arnoldo" AND n.lname="Dieter" AND
m.fname="Gwenn" AND m.lname="Vardnado" CREATE
n-[r:FRIENDS{name1:"ArGw1",name2:"ArGw2"}]->m;
```

In case a user wants to remove an already existing nickname, the following query shall be made to run in Cypher-

```
MATCH (n {name1: "ArGw1"}) SET n.name1 =
NULL SET n.name2=NULL
```

This query can be used to remove already set nicknames or add nicknames to a friend whose at least 1 nickname has been specified. To specify first nickname in a relation, the following query needs to be run-

```
MATCH (m:User{fname:"Arnoldo", lname:"Dieter"}),
(n:User {fname:"Gwenn", lname:"Vardnado"}) MERGE
(m)-[r:FRIENDS{name1:"ArGw1",name2:"ArGw2"}]->(n)
```

In the queries used above, we have used first and last names as the unique identifiers for the users among which the relationship needs to be defined. In the system, user-ids can be used as unique identifiers.

3.2. Using Nicknames to Access User Details

Having defined nicknames for user interaction, in this Section, we'll see on how to retrieve nicknames using the modified database so that the user can use either parts of real name (fname or lname) or assigned nickname(s).

Consider the following example between 2 users, A1 and B1. In format of User (fname, lname, sex, age), we can define A1 and B1 as User (A1, A2, M, 24) and User (B1, B2, F, 24). For sake of simplicity, we consider one-directional friendship relation, i.e. A1->B1 as shown in Figure 3. The nicknames given to B1 by A1 are AB1 and AB2.



Figure 5. Friendship graph between 2 users

When A1 wants to access data of B1 by using their nicknames, they can access the same using the Generic Case¹³, in which the predicates are evaluated in order until a true value is found, and the result value is used. If no match is found the expression in the ELSE clause is used, or null, if no ELSE case exists.

```
CASE
WHEN predicate THEN result
[WHEN ...]
[ELSE default]
END
```

Arguments:

- predicate: A predicate that is tested to find a valid alternative.
- result: This is the result expression used if the predicate matches.
- default: The expression to use if no match is found.

In addition to the query above, some extra work needs to be done. The nicknames should also be indexed alongside the user's friend names for faster retrieval. Also, we recommend usage of a special character, like tilde character (~) in order to signal system that the nickname is going to be used and thus reading through appropriate algorithm as mentioned in Figure 6. However, the same can be incorporated using already existing functionalities associated with '@' symbol, thus adding new functionality without forcing user to use another key for analogous purpose. Using indexing along with special

¹² <http://neo4j.com/docs/stable/preface.html>

¹³ <http://neo4j.com/docs/stable/cypher-expressions.html#syntax-generic-case>

character as mentioned above, we can use nicknames to identify intended friend as per algorithm as stated in Figure 6.

The algorithm can easily incorporate the existing functionality as well. Consider the example we picked at the start of this Section. If A1 wants to lookup (the term lookup is used interchangeably to mention both usage scenarios mentioned as mentioned later in Section 5) B1 via AB1 (the associated nickname), we propose the lookup be performed by means of traversing the directed relation edge between A1 and B1 so as to include the nicknames in the form of attributes of the directed relationship.

When a user starts to input the nickname of a friend, the lookup being performed using the directed edge will automatically link to the name of friend as stated in their social media profile, thus including the functionality.

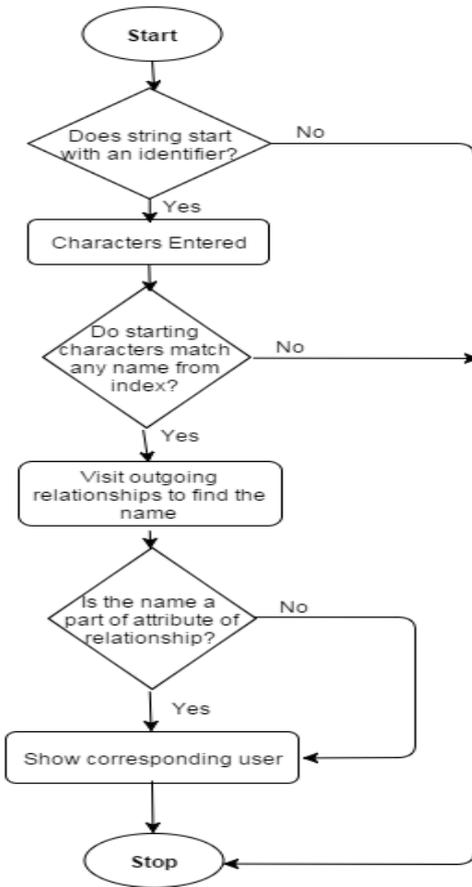


Figure 6. Algorithm for reading intended user data by incorporating nicknames as identifier alongside the usage of real name

If the user has not setup the nickname for their friend and want to link to their profile using the real name of their friend, the directed edge, devoid of any attributes, shall access the name from the directed node and thus shall work as effectively as present system.

4. Analysis of Proposed System

The system cannot be analysed without analysing the storage and time complexity of the same. Figure 7 shows the size of database before declaring properties to ‘FRIENDS’ relation while Figure 8 shows the size of database after adding properties to the relationship.

| Store Sizes | |
|-------------------------|-------------------|
| Array Store | 8.00 KiB |
| Logical Log | 9.59 KiB |
| Node Store | 8.00 KiB |
| Property Store | 7.97 KiB |
| Relationship Store | 7.97 KiB |
| String Store Size | 8.00 KiB |
| Total Store Size | 180.74 KiB |

Figure 7. Database size before adding properties to the relationships

| Store Sizes | |
|-------------------------|-------------------|
| Array Store | 8.00 KiB |
| Logical Log | 20.01 KiB |
| Node Store | 8.00 KiB |
| Property Store | 7.97 KiB |
| Relationship Store | 7.97 KiB |
| String Store Size | 8.00 KiB |
| Total Store Size | 200.23 KiB |

Figure 8. Database size after adding properties to the relationships

While most of the store sizes did not increase in value, the logical log store swelled by approximately 108% of its initial value practically doubling itself. However, overall database size showcased a marginal growth of 11% from 181 KiB to 200 KiB of data. Although it is agreed that 11% growth of data size can be quite hefty, but companies having server farms to hold and manage petabytes of data, it can be deemed safe to not pay much heed to this small problem. However, any further developments that may outdo this shortcoming are always welcome.



Figure 9. Time taken to access all the users with their relationships before adding properties to the relationships

Figure 9 shows the time taken the complete database, with all defined users and the relationships between them, without any properties defined on any relationship. Figure 10 shows the time taken after the properties have been defined for the relationships. The time to access the

complete database went up by approximately 13% from 24ms to 31ms. However, it's worth noting that the stats are for accessing the whole database, not for individual relations. Also, the request was not optimised in any manner, which is not the case with industrial implementations. The current database has not been optimised for read requests, unlike TAO [14]. It is expected that in the industrial implementation, the increase in time to access shall be trimmed down to approximately 5% as compared to the current increase of 13% in the current non-industrial, non-optimised version of the database.

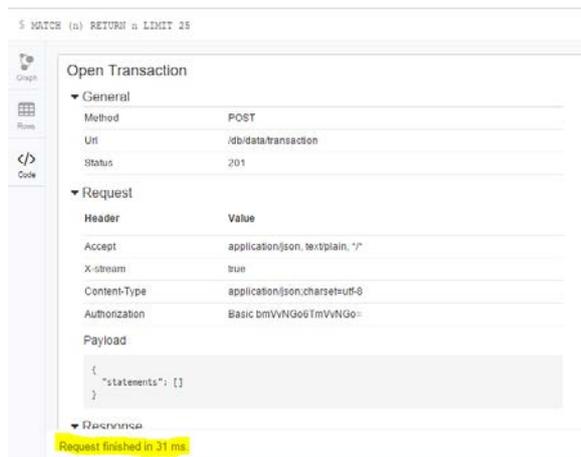


Figure 10. Time taken to access all the users with their relationships after adding properties

5. Proposed Usage

We propose the above mentioned system can be used in following cases-

- As search identifier, in chat systems or search terms
- During tagging a friend as participant in social media activity in form of status or comments

In order to use the system in either of mentioned scenarios, the modus operandi remains the same. The intended friend data is accessed not by the connection or unique user ID but by means of accessing the same through outward edge of a defined relationship.

We have proposed the idea of using multiple nicknames as some people would like to store multiple nicknames for a single person. As mentioned in Section 3.2.3 of this paper by Vosecky et al. [16], the authors point out that multiple nicknames can vary owing to small variations in spelling. For example- Lionel might be referred to as Leo or Leon. Although saving any one might work, people however tend to type in multiple alphabets in a single stretch before waiting for the matching results to show up. Thus, the need for storing multiple nicknames.

There is another example when people might want to store multiple nicknames. Consider this example of a guy called Shinichi, a Japanese name. Most often than not, his user id would have his name in Japanese (written as 新 in Kanji script of Japanese language). Shinichi moves to some other country, and adapts another name for himself for the local people who might have trouble pronouncing his name (usually the case with names in Oriental cultures like those of Chinese, Korean, Japanese, among others). However, some people might still call him by his real

name, but unknown of the script as to how to write his name, they might Romanise it, thus preferring to use both names interchangeably, but stored as nicknames.

Also, multiple nicknames can be valid depending on the context of the conversation. This is especially true in case when the addressee has street credentials. Consider the example of this guy called Jonathan. His street name can be l'il Johnny and in the classroom, he might be known as John. A person who knows Jonathan in both the scenarios would store both the nicknames as while tagging in a post that concerns their school, he is more likely to refer to him as John than l'il Johnny. Thus the need for storing multiple nicknames.

6. Conclusion

In this paper, we have worked with a small, yet functionally similar implementation as that of TAO API used by Facebook. We, however, modified it a bit by adding an extra node at each user to allow added functionality of usage of nicknames in social media.

We have not discussed the performance metrics in terms of access time for a single result using the proposed methodology as the system being closely associated with TAO, shall optimize the read and write queries using sharding of cache as discussed in the paper that gives an overview of TAO [14].

The proposed system shall be highly advantageous to the users of social media. The current methods do not allow the users to remember their friends by using nicknames. However, with the proposed system, the user can tag their friends using their nicknames. It takes the load off the users to remember each other using their actual names and allow better bonding amongst people as they would communicate in real life.

We hope that this system would see light of day in near future to allow users yet another functionality to connect to their contacts on social media better.

References

- [1] Bandura, A. (1986). *Social Foundations of Thought and Action*. Englewood Cliffs, New Jersey: Prentice-Hall
- [2] Bell, R. Q., & Chapman, M. (1986). Child effects in studies using experimental or brief longitudinal approaches to socialization. *Developmental Psychology*, 22(5), 595-603.
- [3] Hill, K. R., Walker, R. S., Božičević, M., Eder, J., Headland, T., Hewlett, B., Wood, B. (2011). Co-Residence Patterns in Hunter-Gatherer Societies Show Unique Human Social Structure. *Science*, 331 (6022), 1286-1289.
- [4] Bartholomew, K. (1990). Avoidance of Intimacy: An Attachment Perspective. *Journal of Social and Personal Relationships*, 7 (2), 147-178.
- [5] Greene, A., Spellman, B., Levy, W., Dusek, J., & Eichenbaum, H. (2001). Relational learning with and without awareness: Transitive inference using nonverbal stimuli in humans. *Memory & Cognition*, 29(6), 893-902.
- [6] Shweder, R., & Bourne, E. (1982). Does the Concept of the Person Vary Cross-Culturally? In A. Marsella & G. White (Eds.), *Cultural Conceptions of Mental Health and Therapy SE - 4* (Vol. 4, pp. 97-137). Springer Netherlands.
- [7] Fortado, B. (1998). Interpreting Nicknames: A Micropolitical Portal. *Journal of Management Studies*, 35: 13-34.
- [8] Bechar-Israeli, H. (1995). FROM <Bonehead> TO <cLoNehEAd>: NICKNAMES, PLAY, AND IDENTITY ON INTERNET RELAY CHAT. *Journal of Computer-Mediated Communication*, 1: 0.

- [9] Sharon Black Brad Wilcox and Brad Platt (2014). "Nicknames in Prison: Meaning and Manipulation in Inmate Monikers". *Names*, 62(3), 127-136.
- [10] Johansson, F., Kaati, L., & Shrestha, A. (2013). Detecting multiple aliases in social media. *Advances in Social Networks Analysis and Mining (ASONAM)*, 2013 IEEE/ACM International Conference on.
- [11] Johansson, F., Kaati, L., & Shrestha, A. (2015). Timeprints for identifying social media users with multiple aliases. *Security Informatics*, 4(1), 7.
- [12] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. US: American Psychological Association.
- [13] WELFORD, A. T. (1952), THE 'PSYCHOLOGICAL REFRACTORY PERIOD' AND THE TIMING OF HIGH-SPEED PERFORMANCE—A REVIEW AND A THEORY. *British Journal of Psychology General Section*, 43: 2-19.
- [14] Bronson, N., Amsden, Z., Cabrera, G., Chakka, P., Dimov, P., Ding, H., ... Venkataramani, V. (2013). TAO: Facebook's distributed data store for the social graph. *Usenix Atc'13*, 49-60.
- [15] Paper, T., Data, P. P., Stremel, J., & Juan, Y. (2011). (12) United States Patent, (12).
- [16] Vosecky, J. Hong, D., & Shen, V. Y. (2009). User identification across multiple social networks. *Networked Digital Technologies*, 2009. NDT '09. First International Conference on.