CSCI651 – ALGORITHM CONCEPTS
C-NYIT

Chintan Gokhale
Department of Computer Science
New York, USA
Student Id – 11025476
cgokhale@nyit.edu

Deep Desai
Department of Computer Science
New York, USA
Student Id – 1059666
ddesai11@nyit.edu

Megha Vasudeva Rao
Department of Computer Science
New York, USA
Student Id – 1080315
mvasudev@nyit.edu

Ravina Varhadi
Department of Computer Science
New York, USA
Student Id – 1074914
rvarhadi@nyit.edu

Shannon Mascarenhas
Department of Computer Science
New York, USA
Student Id – 1076866
smascare@nyit.edu

Abstract— **In today's fast paced world all the people are professionally connected through Linked-In and all the other social networks to stay aware of what's happening in the rapidly growing world. When it comes to academic world, club involvement on campuses is a good way to develop social skills and professional relationships. To unite both academic and professional worlds we have come up with an idea to make a combine application where in we can see recommendations of various clubs based on your skill-sets, endorsements, volunteering experience and 1 st connections. So we are creating a platform to design clubs, events based on user's interests and recommendations based on their connections interest.**

Keywords—LinkedIn, Mobile Application Development, Academic clubs, Porter Stemmer, Keyword Extraction, Mean Weightage

I. Introduction

LinkedIn is the one of the most popular professional networking site. Among the various benefits that linkedIn provides, everyone likes staying connected through LinkedIn to find the best suited jobs, join groups and get in touch with industry experts. In academic institutions, there are variety of clubs and events taking place around the year. Participating in such events expands one's network and gives an opportunity for overall personality development.

The insight is to combine both the academic and professional platforms. Our approach is to create an application to display club recommendations in an institution based on one's LinkedIn skill-sets and endorsements, volunteering experience and 1st connections. In other words, we intend to recommend a user a list of groups or clubs based on his interests.

We have successfully signed in with LinkedIn and used its API to get the user's information such as its skills. We then, have created a technique to recommend clubs to that user.

II. Related Work

A. *Sign in with LinkedIn*

When creating a account on a website, the available options is to create a account or to sign in with LinkedIn or Sign in with Google+, etc. Creating a new account would require the new user to enter all his details. But if the new user selects, "Sign in with LinkedIn", the user details are taken from his/her LinkedIn profile and used for populating and filling up the user information for an account at the website. This latter option internally uses a LinkedIn API to fetch all the user's information. The option for signing in with LinkedIn is widely used in various websites. One example is ProInsights.com. It is a LinkedIn Visualization & Offline Contact Management app that profiles your LinkedIn network and provides meaningful insights.[1]

B. *Meetup*

The format for our application is taken from Meetup. Meetup is an online social networking portal that facilitates offline group meetings in various localities around the world. [2] Meetup allows members to find and join groups unified by a common interest, such as politics, books, games, movies, health, pets, careers or hobbies. Meetup was designed as a way for organizers to manage the many functions associated with in-person meetings and for individuals to find groups that fit their interests.[3]

C. *Natural Language Processing*

Natural language processing(NLP) refers to the use and ability of systems to process sentences in a natural language such as English.[4]. There are three types of machine learning: Supervised, semi supervised and unsupervised. Generally, the problems of machine learning may be considered variations on function estimation for classification, prediction or modeling.[5]

In supervised learning one is furnished with input (x1, x2, . .,) and output (y1, y2, . .,) and are challenged with finding a function that approximates this behavior in a generalizable fashion. The output could be a class label (in classification) or a real number (in regression)-- these are the "supervision" in supervised learning. In the case of unsupervised learning, in the base case, you receives inputs x1, x2, . ., but neither target outputs, nor rewards from its environment are provided. Based on the problem (classify, or predict) and your background knowledge of the space sampled, you may use various methods: density estimation (estimating some underlying PDF for prediction), k-means clustering (classifying unlabeled real valued data), k-modes clustering (classifying unlabeled categorical data), etc. Semi-supervised learning involves function estimation on labeled and unlabeled data. This approach is motivated by the fact that labeled data is often costly to generate, whereas unlabeled data is generally not. The challenge here mostly involves the technical question of how to treat data mixed in this fashion. [5]

1) *Porter Stemmer*:

Stemming is a pre-processing step in Text Mining applications as well as a very common requirement of Natural Language processing functions. It is widely used in most of the Information Retrieval systems. The main purpose of stemming is to reduce different grammatical forms/ word forms of a word like its noun, adjective, verb, adverb etc. to its root form.[6]

Stemming has many approaches like Brute force look up, Suffix, affix stripping, Part-of-speech recognition, Statistical algorithms (n-grams),etc.[7] Porter Stemmer utilizes suffix stripping. It doesn't address suffixes.

2) *Keyword Extraction*:

Keyword extraction (KE) is defined as the task that automatically identifies a set of the terms that best describe the subject of document. Term frequency, Inverse document Frequency (TFIDF) weighting has been widely used for keyword or key phrase extraction. The idea is to identify words that appear frequently in a document, but do not occur frequently in the entire document collection. Much work has shown that TFIDF is very effective in extracting keywords for scientific journals, e.g., (Frank et al., 1999 [8]; Hulth, 2003[9]; Kerner et al.,2005[10])

Web information has also been used as an additional knowledge source for keyword extraction. Turney(2002) [12] selected a set of keywords first and then determined whether to add another keyword hypothesis based on its PMI (point-wise mutual information) score to the current selected keywords. [11]

Keyword extraction has also been treated as a classification task and solved using supervised machine learning approaches. In these approaches, the learning algorithm needs to learn to classify candidate words in the documents into positive or negative examples using a set of features.[11]

Another line of research for keyword extraction has adopted graph-based methods similar to Google's PageRank algorithm [13]. In particular, Wan et al.,(2007) attempted to use
a reinforcement approach to do keyword extraction and summarization simultaneously, on the assumption that important sentences usually contain keywords and keywords are usually seen in important sentences.[14]

A lot of work has been done on Keyword extraction, we intend to use the unsupervised keyword extraction algorithm and further implement it.

D. *Weighted Mean*

Weighted Mean has many applications. It refers to calculating the mean based on the weights of the objects. Weighted Average is used to with Application to Biomedical Signals. They found that the noise almost destroys the image. The most promising approaches used for noise reduction problem and the one that gave the best results was Weighted Mean. [30]

III. Implementation

A. *System Architecture*

Our system architecture includes LinkedIn API, Mobile Application (client), Service API (C# Web Service) and Database as shown as in Fig. 1

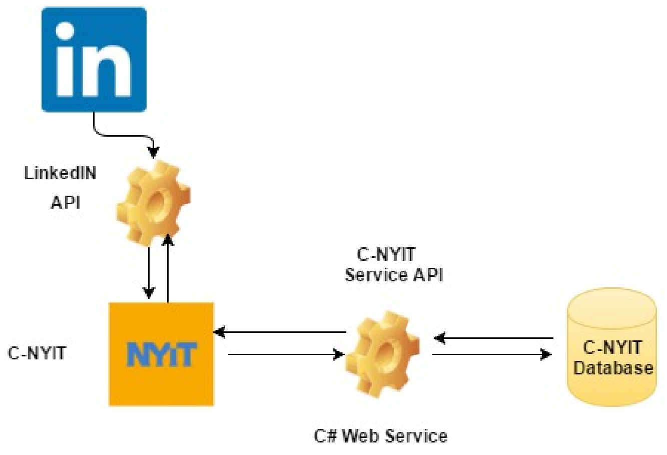


Fig.1. System Architecture Diagram

Firstly, the mobile app will ask for "Sign In with LinkedIn". Once the user has selected the option, internally a LinkedIn API is used to get the user's profile and skills. The clubs and events information is stored in the database. After the LinkedIn skills are returned, they are compared with the Clubs and events information. This club and event related information is retrieved from the database.

We have used HTML, CSS and AngularJS to design the system. MySQL is used for backend. We have implemented RESTful web-services in .NET framework to receive request and response through LinkedIn and User Interface.

B. *Workflows*

1) *Entity Relationship Diagram*

Fig. 2 shows the ER diagram for our application. User can be a current student at NYIT or an Alumni and must have a Linked-In account. This is an end user of the application. The user's personal details, NYIT account details and Linked-In user name is stored in UserDetail table.

The objective of our application is to recommend relevant club to the users. The club related information such as ClubName, ClubID, ClubDescription, ClubLogo, ClubLocation, Advisor, Meeting dates are stored in the ClubDetail table. Information such as whether the group is for International students, or if the group is currently active or if the group is technical is also stored in the ClubDetail table. Every club will have on-going and future events. All the event related information such as the EventName, EventID, EventDate, EventLocation, Co-ordinator name, and date will be stored in a EventDetail table. One of the must have field for an event is the ClubId field. Every event created should belong to a club.

We have Globally Unique Identifiers (GUIDs) as the primary key for the all the tables i.e UserGUID, ClubGUID and EventGUID as primary keys for UserDetail, ClubDetail and EventDetail tables respectively. UserID, ClubID and EventID defined as unique. The events will to tied to a particular club and hence, EventID will reference to ClubID as foreign key.

User may belong to more than one club. To maintain a record of the clubs the user belongs to, we have UserClubDetail table. It contains UserID and its respective ClubID. The field status is to determine the status of the user in the club. Active or not active. We have RoleDetail which will contain the three roles Administrator, Advisor and Member (or user). The Admin will have the highest privileges. An Admin can create clubs and create events, approve student request to join the clubs. For the sake of simplicity, in our initial phase the admin will accept all students wishing to join the club. Lastly, to maintain the user and its role and the club it belongs to in sync, we have created UserClubRole table.

RoleId and UserClubRoleId serve as the primary keys to tables RoleDetail and UserClubRoleDetail respectively. ClubId of the EventDetail table serves as foreign key, it references to the ClubId of the ClubDetail table.

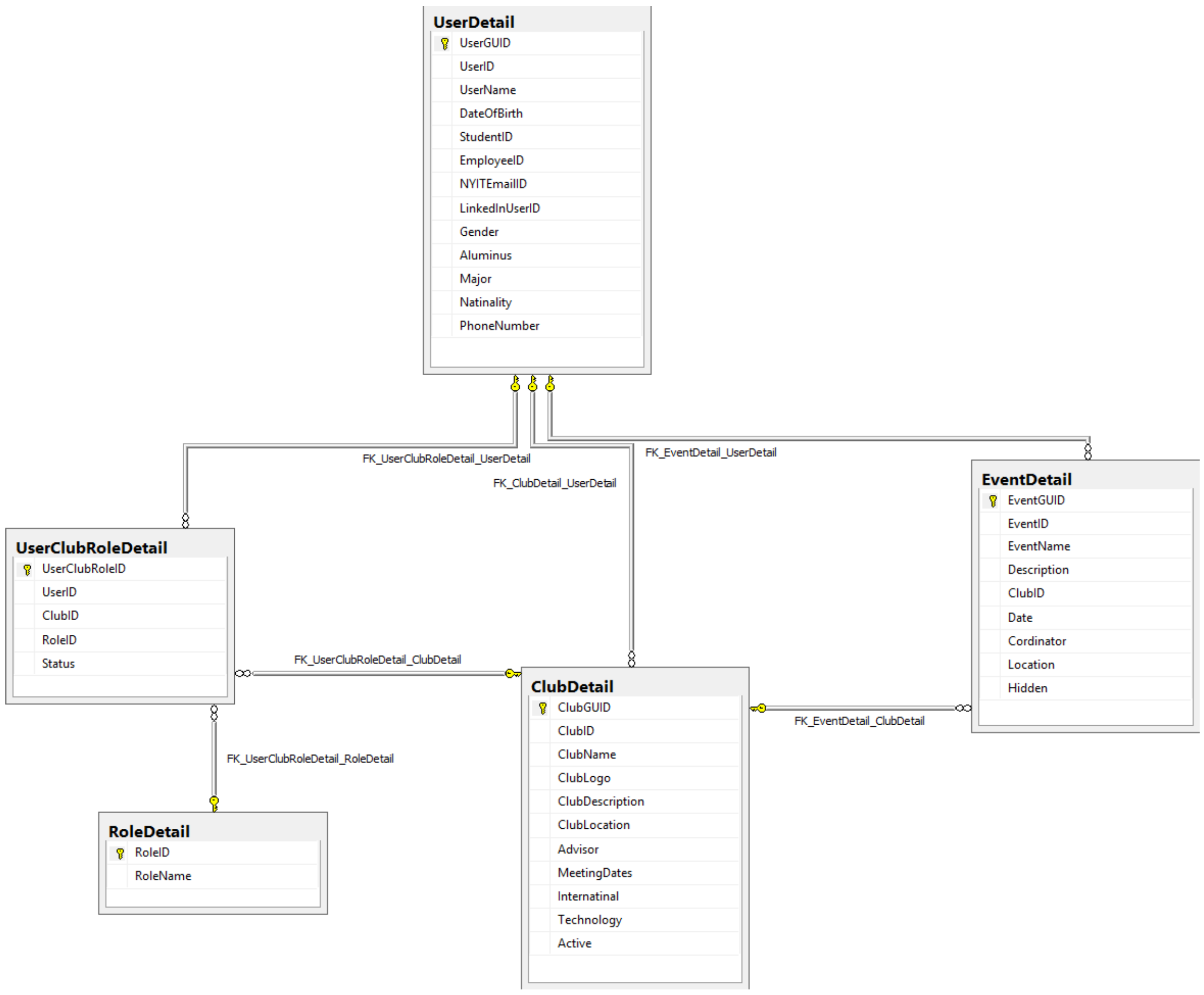


Fig. 2. ER Diagram

2) 2) User Workflow

Fig. 3 describes the workflow of the end user. The user must create a valid account in the application. The user will then login using his/her Linked-in credentials. If the user does not have an account with LinkedIn, then he/she must create a valid account and add skills and other details.

Once the user is logged in,he/she will be able to see their profile. This profile will contain your details from LinkedIn. You name and title is picked up from LinkedIn. You will be able tp browse through the different clubs present in an institution. As the LinkedIn API is used to get your skills and process them. One list knows as 'Recommended clubs" will be added to each user profile. There will be two lists – All clubs and recommended clubs. The user can look for any clubs that interests him. He can see the Club logo, club name, club description and decide if he would like to join the club. Any club thats interests the user, he/she can join the group by clicking the "Join" club.

Once you join the club, you will be able to see their on-going and future events. He/she can look through all the events, register for them. The events will have their name, description , meeting data, location, time and some other details. The user can register by clicking on the Register icon and he/she will be registered for the event.

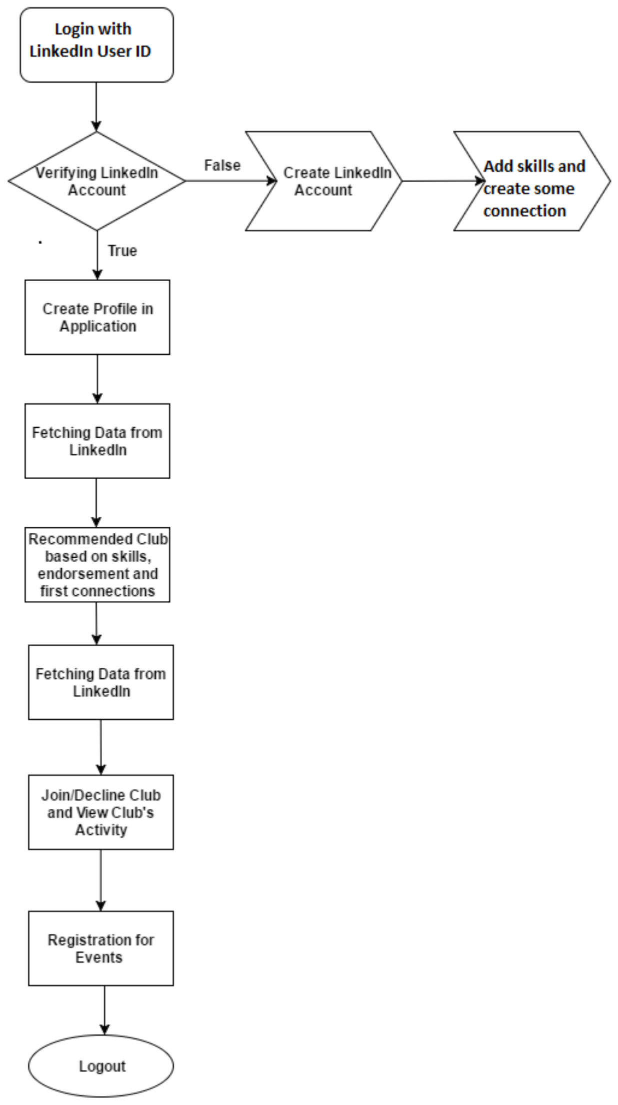

Login with LinkedIn User ID
Verifying LinkedIn Account
False
Create LinkedIn Account
Add skills and create some connection
True
Create Profile in Application
Fetching Data from LinkedIn
Recommended Club based on skills, endorsement and first connections
Fetching Data from LinkedIn
Join/Decline Club and View Club's Activity
Registration for Events
Logout

Fig. 3 . User Workflow

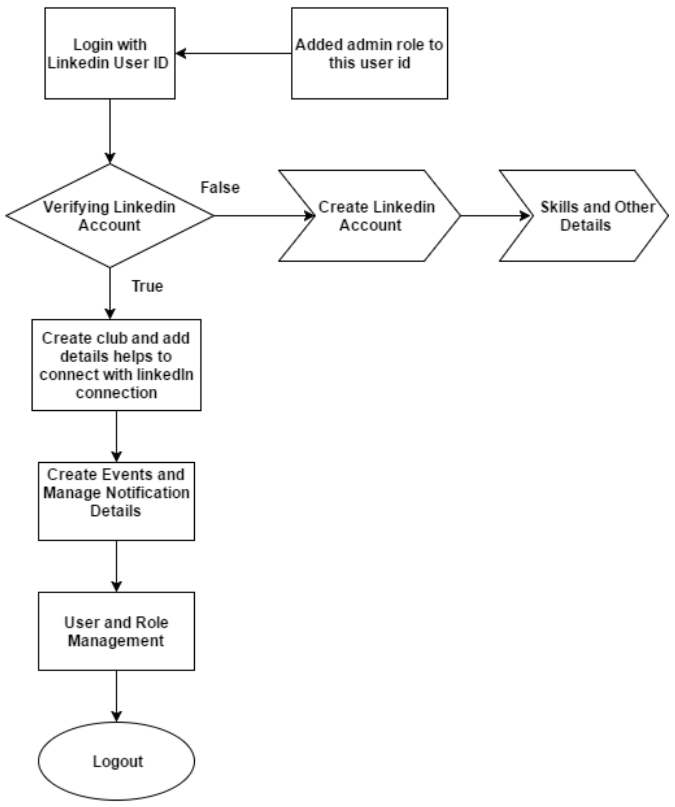

Login with Linkedin User ID
Added admin role to this user id
Verifying Linkedin Account
False
Create Linkedin Account
Skills and Other Details
True
Create club and add details helps to connect with linkedIn connection
Create Events and Manage Notification Details
User and Role Management
Logout

Fig. 4 . Admin Workflow

3) Admin Workflow

Fig. 4 describes the Admin's roles and responsibilities. The Admin will his Linkedin credentials just like any other member. This user will be given an Admin role in the database. Once this user(Admin) has logged in, he/she will get a option for creating clubs and events for them. Admin will also manage roles for different users.

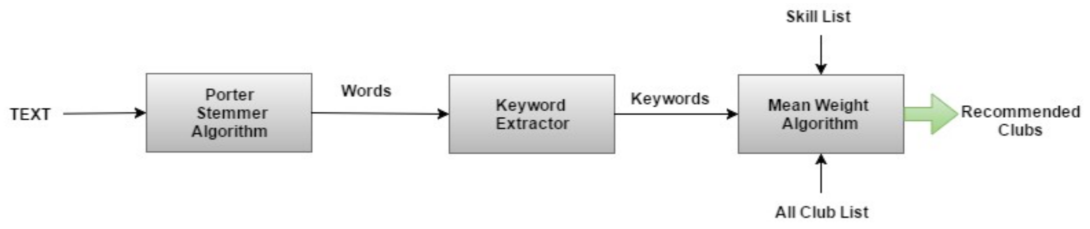


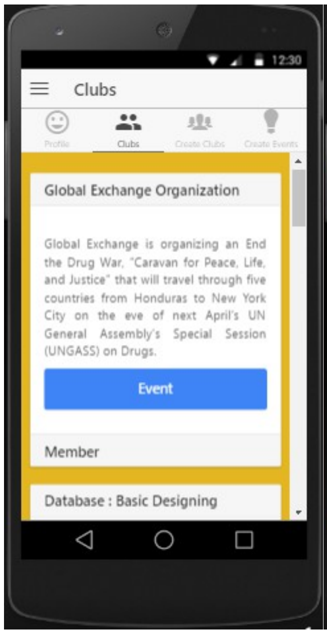
Fig. 5 . Recommendation Workflow

3) Recommendation Workflow

We have used three algorithms in our project, namely, Porter Stemmer, Keyword Extraction and Weighted Mean. The Porter Stemmer is used to stem the words. We have used Suffix stripping that address the suffixes of the words and return them. Then the keyword extractor takes this words and intelligently picks up words that are frequently used and determines keywords. Finally the keywords are given as an input to Mean Weight Algorithm with the LinkedIn skill set and the club list from database. It then calculates the format to recommend clubs to the user. Fig. 5 shows the recommendation workflow.

4) User Interface Design

Fig. 6- 15 show the user interface created for our mobile application. Fig. 6 show the page that contains all clubs information. Fig. 7and 8 describe the admin facility to create a club. It contains options like Club name, club logo, club description, location, etc. There is "Create club" button at the bottom of the page. On clicking a club will be created. Once a club is created. We can add events to it. Fig. 9 and 10 again, is the admin facility to create an event in a club. Event information like Event name, event description, date, time, location, etc. There is "Create Event" button as the bottom of the page. On clicking that, an event will be created. Fig. 11 demonstrates Menu and the different options like user's profile, Events, clubs and Log In. On clicking "Profile", we get a page displaying user information. (Fig. 12). On clicking "Login", we get "Login in LinkedIn" option. (Fig. 13). Once the user selects that, we get it profile page. (Fig. 12). Fig 14 is the LinkedIn application developer key returns the client id and password.[28] We have used PhoneGap to make the App cross platform [29]



Global Exchange Organization

Global Exchange is organizing an End the Drug War, "Caravan for Peace, Life, and Justice" that will travel through five countries from Honduras to New York City on the eve of next April's UN General Assembly's Special Session (UNGASS) on Drugs.

Database : Basic Designing

Fig. 6. Clubs

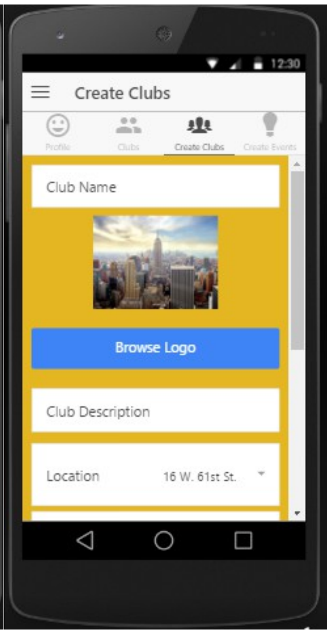


Fig.7. Createclubs1

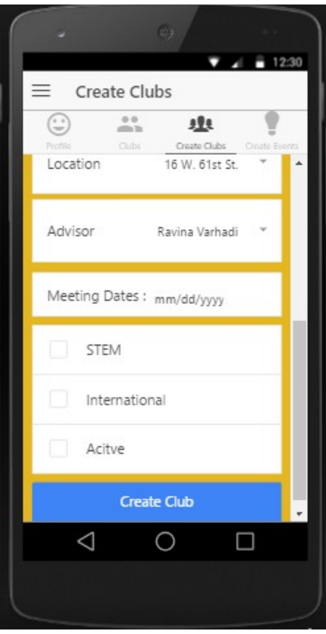


Fig. 8. CreateClubs

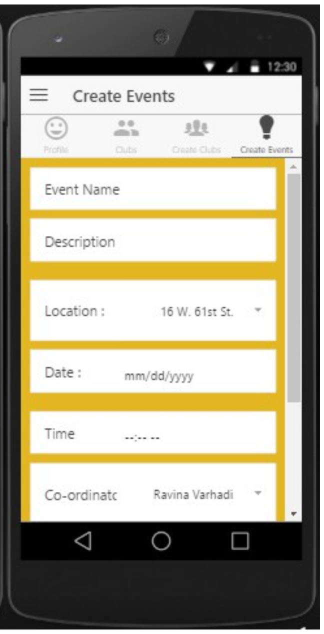


Fig. 9. CreateEvents1

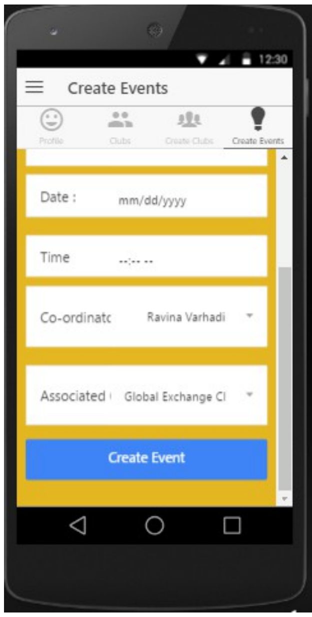


Fig.10. CreateEvents2

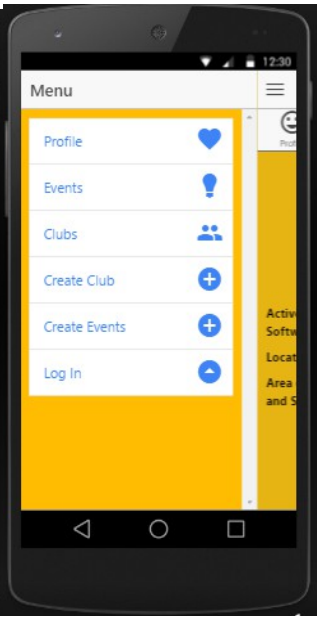
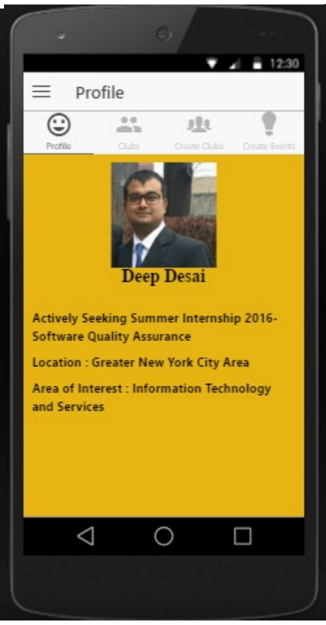


Fig. 11. Menu



Deep Desai

Actively Seeking Summer Internship 2016- Software Quality Assurance

Location : Greater New York City Area

Area of Interest : Information Technology and Services

Fig . 12. Profile

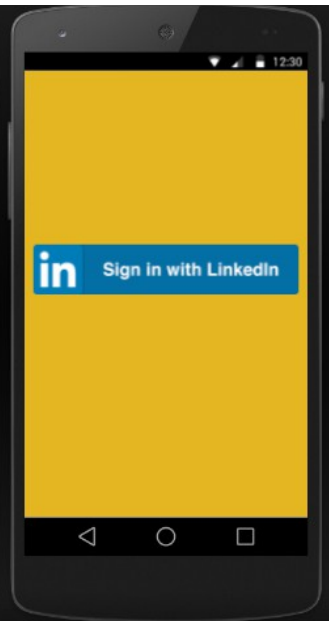
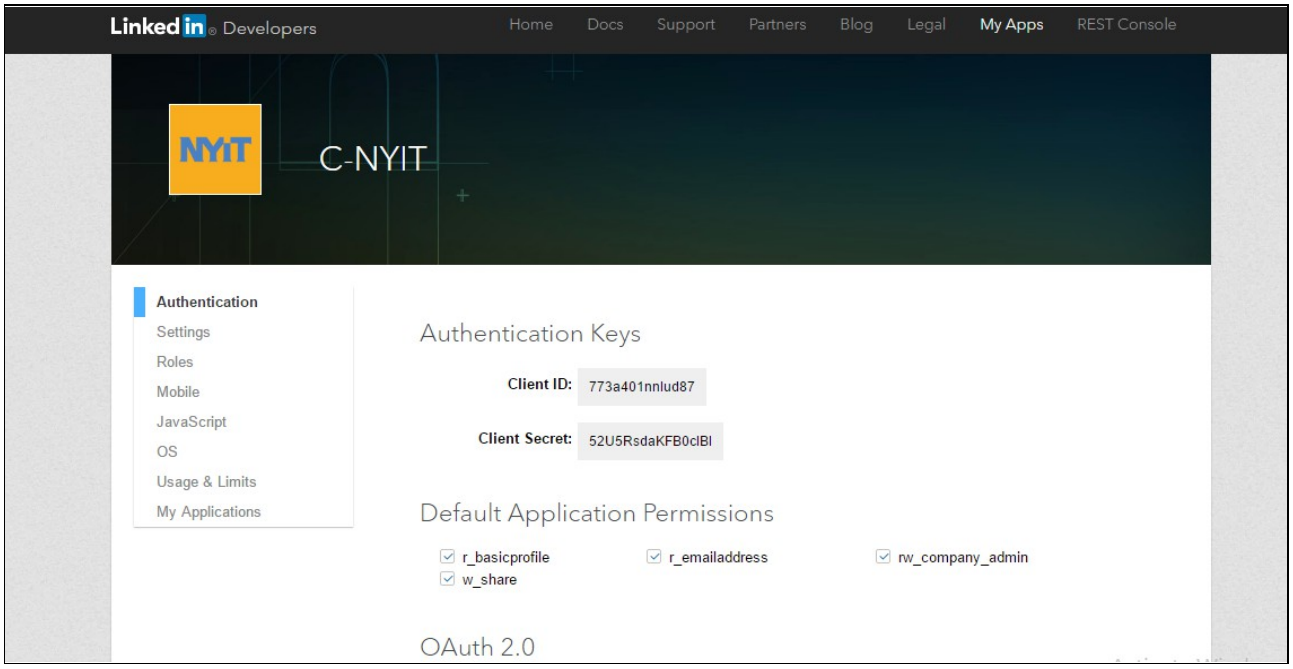


Fig.13. SignIn with LinkedIn



Authentication Keys

Client ID: 773a401nnlud87

Client Secret: 52U5RsdaKFB0clBl

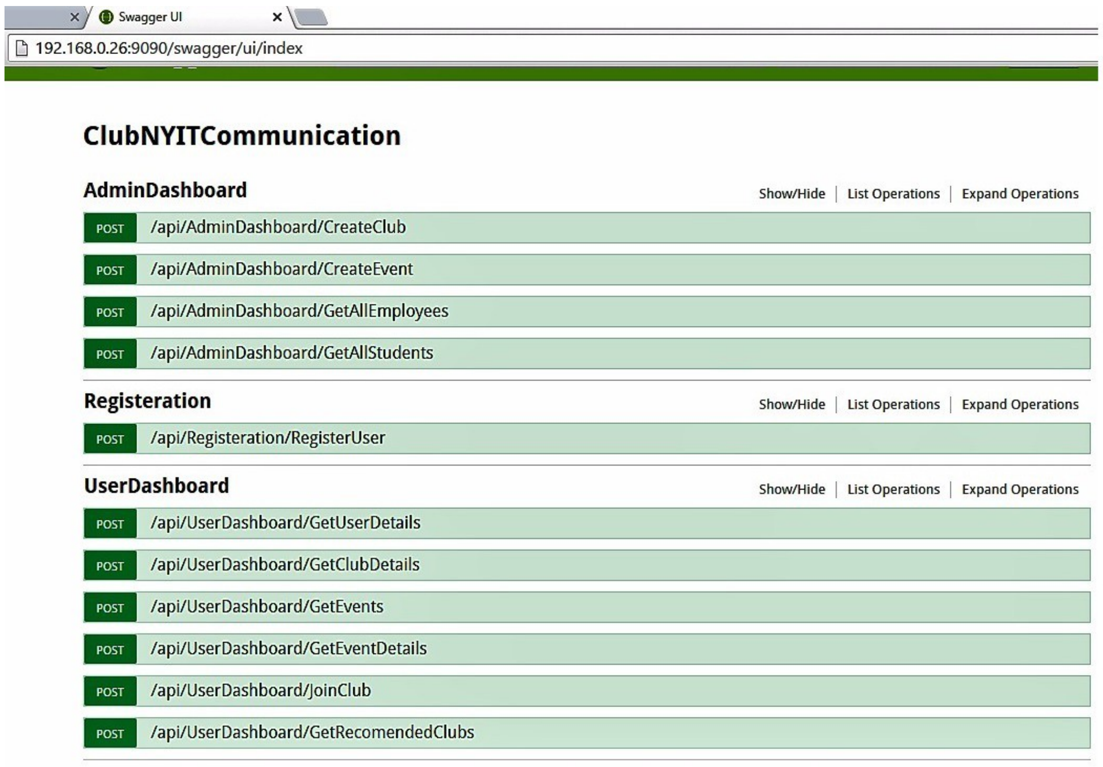
Default Application Permissions

✓ r_basicprofile ✓ r_emailaddress ✓ rw_company_admin
✓ w_share

OAuth 2.0

Fig. 14. LinkedIn Developer Application Key

Fig. 14 shows the LinkedIn developer Application key. LinkedIn relies in Oauth 2.0 [28] protocol for granting access. Firstly, we need to congigure our application with linkedIn. To do this, we have to go to the linkedIn developer website and then to MyApps. There is an option called "CreateApplication". On clicking this option, we get a page to page to enter the App name, logo , site and other details. Once the configuration is done, we get a clientID and client secret key as shown as in Fig 14. We further request for authorization code. Once the authorization is complete, we get access_token and expires_in parameter.



ClubNYITCommunication

AdminDashboard

POST /api/AdminDashboard/CreateClub
POST /api/AdminDashboard/CreateEvent
POST /api/AdminDashboard/GetAllEmployees
POST /api/AdminDashboard/GetAllStudents

Registeration

POST /api/Registeration/RegisterUser

UserDashboard

POST /api/UserDashboard/GetUserDetails
POST /api/UserDashboard/GetClubDetails
POST /api/UserDashboard/GetEvents
POST /api/UserDashboard/GetEventDetails
POST /api/UserDashboard/JoinClub
POST /api/UserDashboard/GetRecomendedClubs

Fig .15 Service APIs

7) Web Service API

Fig. 15. shows the different service API built for our application. We have divided the service API into three modules namely AdminDashboard, Registration and userDashboard. The calls serve as API endpoints. When the Admin user selects the create club from the UI. A web API will be submitted containing all the information like club name, logo, description, location, time, etc. This data will be stored in the database. CreateEvent API will be submitted when the Admin user will click on the "Create Event". The data such as event name, description, date, time, location etc, will be stored in the database. GetAllEmployees and GetAllStudents API are used to get the total number and information of students/employees.

RegisterUser API will be used to register a new user to use our application. GetUserDetails API is used to get all the user details from the database. GetClubDetails returns all the club related information from the database. GetEvents returns all the events in a club from the database. GetEventDetails returns all the event related information from the database. When the application student/alumni user clicks "Join club", the JoinClub API will be used. And the student's name and information will be entered in club. GetRecommendedClubs API is used to get a list of recommended clubs to the user.

C. Algorithm

1) Porter Stemmer

Stemming is widely used in to retrieve information. It is technique used to reduce words to their root form, by removing derivational and inflectional affixes.[17] Porter Stemmer was developed by Martin Porter in 1980 at the University of Cambridge[16]. Porter's algorithm is applied as a pre - processing step for the indexing task; its main use is as part of a term normalization process that is usually done when setting up an Information retrieval system. In terms of retrieval performance, Porter Stemmer is better than classification, clustering, spam filtering… The stemmer is based on the idea that the suffixes in the English language are mostly built of a combination of smaller and simpler suffixes; for instance, the suffix "fullness" is composed of two suffixes "full" and "ness ". Thus, Porter stemmer is a linear step stemmer; it applies morphological rules sequentially allowing removing affixes in stages.[17]

The algorithm has five steps. Words' suffixes are removed step by step.

Step 1	: Gets rid of plurals and -ed or -ing suffixes
Step 2	: Turns terminal y to i when there is another vowel in the stem
Step 3	: Maps double suffixes to single ones: -ization, -ational, etc.
Step 4	: Deals with suffixes, -full, -ness etc.
Step 5	: Takes off -ant, -ence, etc.
Step 6	: Removes a final –e

Helper Functions:-

- m() - Returns the number of "consonant sequences" in the current stem:
  gives 0 (cry, cry-ing) vc
  gives 1 (care, car-ing, scare, scar-ing) vcvc
  gives 2 (probab-ility)

- r(String str) -If m-function is > 0 then set the current suffix to "str"

- cvc(int pos) - Checks whether the previous 3 characters before pos were consonant, vowel, consonant

Step1: Gets rid of plurals and -ed or -ing suffixes. Pseudo Code:-

```
if b[k] == 's'
    if ends("sses")
        k -= 2
    else if ends("ies")
        setto("i")
    else if b[k-1] != 's'
        k--
if ends("eed")
    if m() > 0
        k--
else if ends("ed") || ends("ing") &&
vowelinstem()
    k = j
    if ends("at")
        setto("ate")
    else if ends("bl")
        setto("ble")
    else if ends("iz")
        setto("ize")
    else if doublec(k)
        k--
        int ch = b[k]
        if ch=='l' || ch=='s' || ch=='z'
            k++
        else if (m() == 1 && cvc(k))
            setto("e")
```

Examples:-

- Possesses -> possess
- Ponies -> poni

Step 2: Turns terminal y to i when there is another vowel in the stem. Pseudo Code:-

```
if ends("y") && vowelinstem()
    b[k] = 'i'
```

Examples:-

- Coolly -> coolli
- Furry -> furri

Step 3: Maps double suffixes to single ones.
Pseudo Code:-

```
switch b[k-1]
    case 'a':
        if ends("ational") -> r("ate")
        if ends("tional") -> r("tion")
    case 'c':
        if ends("enci")) -> r("ence")
        if ends("anci")) -> r("ance")
    case 'e':
        if ends("izer") -> r("ize")
    case 'l':
        if ends("bli") -> r("ble")
        if ends("alli") -> r("al")
        if ends("entli") -> r("ent")
        if ends("eli") -> r("e")
        if ends("ousli") -> r("ous")
    case 'o':
        if ends("ization") -> r("ize")
        if ends("ation") -> r("ate")
        if ends("ator") -> r("ate")
    case 's':
        if ends("alism") -> r("al")
        if ends("iveness") -> r("ive")
        if ends("fulness") -> r("ful")
        if ends("ousness") -> r("ous")
    case 't':
        if ends("aliti") -> r("al")
        if ends("iviti") -> r("ive")
        if ends("biliti") -> r("ble")
    case 'g':
        if ends("logi") -> r("log")
```

Examples:-

- Rational -> rational
- Optional -> option

Step 4: Deals with suffixes, -full, -ness etc.
Pseudo Code:-

```
switch b[k]
    case 'e':
        if ends("icate") -> r("ic")
        if ends("ative") -> r("")
        if ends("alize") -> r("al")
    case 'i':
        if ends("iciti") -> r("ic")
    case 'l':
        if ends("ical") -> r("ic")
        if ends("ful") -> r("")
    case 's':
        if ends("ness") -> r("")
```

Examples-

- Authenticate ->authentic
- Predicate -> predic

Step 5: Takes off -ant, -ence, etc.
Pseudo Code:-

```
switch b[k-1]
    case 'a':
        if ends("al") -> break
    case 'c':
        if ends("ance") -> break
        if ends("ence") -> break
    case 'e':
        if ends("er") -> break
    case 'i':
        if ends("ic") -> break
    case 'l':
        if ends("able") -> break
        if ends("ible") -> break
 … more rules here...
    default: return
if m() > 1
    k = j;
```

Examples:-

- Precedent -> preced
- Operational -> operate -> oper

Step 6: Removes a final –e.
Pseudo Code:-

```
if b[k] == 'e'
    int a = m()
    if a > 1 || a == 1 && !cvc(k-1)
        k--
    if b[k]=='l' && doublec(k) && m() > 1
        k--
```

Examples:-

- Parable -> parabl
- Fate -> fate (cvc)

[16- 17]. After the Porter Stemmer algorithm is applied, the output is words which is given as an input to Keyword Extractor.

2) Keyword Extraction

Keyword extraction enriches a document with keywords that are explicitly mentioned in text. Words that occurred in the document are analyzed in order to identify the most representative ones, usually exploring the source properties (i.e. Frequency, length) commonly, keyword extraction does not use a predefined thesaurus to determine the keywords. Existing methods for automatic keyword extraction can be divided by Ping-I and Shi-Jen into[25]: 1) Statistics Approaches and 2) Machine Learning Approaches, or slightly more detailed in the four categories proposed by Zahang et al. [22]: 1) Simple Statistics Approaches, 2) Linguistics Approaches ,3) Machine Learning Approaches and 4) Other Approaches [18 – 25]We have used the unsupervised machine learning approach. The keyword extraction problem can be divided into:

- Individual keyword extraction – individual words with a special and important meaning, generally in the form of a noun or named entity.
- Keyphrase extraction and derivation – phrases contain two or more keywords and other information in a human-readable form. This phrase can contain verbs and stop words for better readability. Short phrases can be joined by their co-occurrence percentage number.

A keyword extraction algorithm has three main components:

- Candidate selection: Here, we extract all possible words, phrases, terms or concepts (depending on the task) that can potentially be keywords.
- Properties calculation: For each candidate, we need to calculate properties that indicate that it may be a keyword. For example, a candidate appearing in the title of a book is a likely keyword.
- Scoring and selecting keywords: All candidates can be scored by either combining the properties into a formula, or using a machine learning technique to determine probability of a candidate being a keyword. A score or probability threshold, or a limit on the number of keywords is then used to select the final set of keywords.

In Candidates step, we break at stop words & punctuation, normalize, Map to vocabulary and disambiguate. In the Properties step, we calculate the frequency of the occurrences, position of the element, phrase length, etc. In the scoring step,there is Heuristic formula that combines the most powerful properties or there is unsupervised Machine learning that determines the importance of properties from the keywords. [26]

3) *Mean Weightage*

The weighted mean is defined as follows: [27]

$$\mu = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

where x represents the skill and w it weight. The skill is compared to club description. Hence, the weight represents the skill matching the keywords in the club description.

IV. Privacy

Privacy of our Mobile App user and his linkedIn profile is maintained. We use the LinkedIn API to get the user's skills. These skills are not stored anywhere (not in our DB or anywhere else). This LinkedIn API is submitted every time a user request's it I.e when the user sign's in with linkedIn. The skills retrieved are used to compute the recommended clubs. Hence, the privacy of the user is intact.

Conclusion

Thus, we have successfully implemented a front end to sign in with linkedIn and we then get user's skills from LinkedIn. We have completed the backend with web service API to store and retrieve the club/ event and related information from the database. Continuing, we have come up with a technique to recommend clubs based on the algorithms described earlier.

Future Work

Future work will include integrating the front end and the backend. Furthermore, we intend to send email notifications to users about the new club additions, recommending clubs based on endorsement weights and first connection's skillsets, adding Eventbrite API to get information of the events happening around and enabling white labeling/branding service to our application.

Learning Process

Initially, we researched on the “Sign In with LinkedIn” functionality and learnt how it worked. It uses oauth for authentication. We used that for authenticating user in our application. We did a detailed study of the different algorithms that can be used for our application. I.e for a best technique to recommend clubs. From the different stemming approaches, Porter stemmer uses suffix stemming and seemed suitable for initial processing of the club description. Among the many approaches for Keyword extraction, though the linguistic approach is promising, the unsupervised machine learning approach seemed suitable. We did a study of the light weight algorithms that can be used to develop a Mobile Application. We learnt the different exception handling methods. Also did a thorough study while creating a Web APIs.

Acknowledgment

We thank Professor Wenjia Li for his valuable guidance during the course of the project. His insight and suggestions has helped and motivated us in the successful completion of a Mobile Application.

References

[1] [Online]. Available http://www.proinsights.me/

[2] Meetup [Online]. Available https://en.wikipedia.org/wiki/Meetup_%28website%29

[3] Meetup [Online]. Available http://www.meetup.com/

[4] Winfred Phillips , Introduction to Natural Language Processing (2006) [Online]. Available http://www.mind.ilstu.edu/curriculum/protothinker/natural_language_processing.php

[5] John L. Taylor Availabe [Online] http://stats.stackexchange.com/questions/517/unsupervised-supervised-and-semi-supervised-learning

[6] Ms. Anjali Ganesh Jivani, A Comparative Study of Stemming Algorithms. Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938 ISSN:2229-6093

[7] Daniel Waegel, [Online] Available https://www.eecis.udel.edu/~trnka/CISC889-11S/lectures/dan-porters.pdf

[8] Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G.Nevill-Manning. 1999. Domain-specific keyphrase extraction. In Proceedings of IJCAI, pages 688–673

[9] Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of EMNLP, pages 216–223

[10] Y.H. Kerner, Z. Gross, and A. Masa. 2005. Automatic extraction and learning of keyphrases from scientific articles. In Computational Linguistics and Intelligent Text Processing, pages 657–669

[11] Feifan Liu, Deana Pennell, Fei Liu and Yang Liu, Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts. Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 620–628, Boulder, Colorado, June 2009.2009 Association for Computational Linguistics

[12] P.D. Turney. 2000. Learning algorithms for keyphrase extraction. Information Retrieval, 2:303–336

[13] S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 30.

[14] X. Wan, J. Yang, and J. Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In Proceedings of ACL, pages 552–559

[15] Porter, M.F. (2006). An algorithm for suffix stripping. Program: electronic library and information systems, Vol. 40 Iss: 3, pp.211–218

[16] Wahiba Ben Abdessalem Karaa (2013). A New Stemmer to improve Information Retrieval International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.4, July 2013

[17] Daniel Waegel (2011). The Porter Stemmer. Available [Online] https://www.eecis.udel.edu/~trnka/CISC889-11S/lectures/dan-porters.pdf

[18] Brian Lott. 012. Survey of Keyword Extraction Techniques

[19] Martin Dostal and Karel Jezek. Automatic Keyphrase Extraction based on NLP and Statistical Methods

[20] Feifan Liu. Deana Pennell. Fei Liu and Yang Liu. Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts

[21] FangJiang. GuoheLi. XueYun and XiangYue . Semantic-based Keyword Extraction Method for Document. International Journal of u-and e-Service, Science and Technology Vol.8, No.5(2015), pp.37-46 http://dx.doi.org/10.14257/ijunesst.2015.8.5.04

[22] Chengzhi ZHANG . Huilin WANG. Yao LIU. Dan WU. Yi LIAO.Bo WANG. Automatic Keyword Extraction from Documents Using Conditional Random Fields.Journal of Computational Information Systems4:3(2008) 1169-1180 Available at http://www.JofCI.org

[23] Slobodan Beliga. Keyword extraction: a review of methods and approaches

[24] Kazi Saidul Hasan and Vincent Ng. Automatic Keyphrase Extraction: A Survey of the State of the Art. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1262–1273,Baltimore, Maryland, USA, June 23-25 2014. 2014 Association for Computational Linguistics

[25] P. Chen, S. Lin, "Automatic keyword prediction using Google similarity distance", presented at Expert Syst. Appl., pp.1928- 1938, 2010

[26] Alyona Medelyan , “NLP keyword extraction tutorial with RAKE and Maui” [Online]. Available https://www.airpair.com/nlp/keyword-extraction-tutorial

[27] Tony Finch. 2009. Incremental calculation of weighted mean and variance

[28] LinkedIn developer site , [Online] Available https://developer.linkedin.com/

[29] PhoneGap, [Online] Available http://docs.build.phonegap.com/en_US/#googtrans(en)

[30] Alina Momot. Methods of Weighted Averaging with Application to Biomedical Signals.