Job Matching Platform Using Latent Semantic Indexing and Location Mapping Algorithms

Francis G. Balazon[1], Albert A. Vinluan[2], Shaneth C. Ambat[3]
AMA University, Quezon City
fbalazon@yahoo.com[1], albert.vinluan@gmail.com[2], shaneth_ambat@yahoo.com[3]

Asia Pacific Journal of Multidisciplinary Research
Vol. 6 No.4, 1-8
November 2018 Part II
P-ISSN 2350-7756
E-ISSN 2350-8442
www.apjmr.com
CHED Recognized Journal
ASEAN Citation Index

Date Received: March 3, 2018; Date Revised: October 25, 2018

Abstract *– Nowadays, there are existing online job matching platform. Though these existing online jobs matching offer increased computation speed of servers and more convenient online job matching, they still strive to improve their job matching applications to achieve accuracy and relevancy. This will further improve the hiring process and reduce the hiring time and cost for companies.*

However, since most of these job matching methods are still based on the basic and traditional approach, the job qualification information of the job seeker is not effectively compared to the basic job requirements provided by the employer. Thus, when the job seeker uses these traditional online recruitment applications which normally use only simple Boolean operations to compare information, irrelevant jobs would be matched, or a number of job descriptions will be obtained.

This research endeavour proposed algorithms to recommend suitable jobs to job seekers based on the collection and analysis of information, and applicants on a specific position to employers using latent semantic indexing and location mapping algorithms. Latent Semantic Indexing algorithm extract and represent the contextual use and meaning of words thru statistical computation applied to document list. The location of jobs input of employers/recruiters are connected to the location mapping module. Likewise, the requested work locations input of job seekers is connected into "geocodes".

Based on the results and analysis of the developed job matching platform, it can discover similar jobs from a query job and employer can easily evaluate candidate job seeker without the needs of human intervention.

The testing results show that using latent semantic indexing and location mapping algorithms performed the best in matching the similar jobs.

Keywords *– corpus, job matching, job seeker, latent semantic indexing, location mapping.*

INTRODUCTION

Considering that every job seeker wants to find a job that best suits his skills and education, walking independently to a company or institution to try his luck to get hired on a desired position would be ineffective. With this dilemma, a number of recruitment agencies offer different recruitment applications that match the profile of the job applicant and the basic job requirement. However, since most of these job matching methods are still based on the basic and traditional approach, the job qualification information of the job seeker is not effectively compared to the basic job requirements provided by the employer. Thus, when the job seeker uses these traditional online recruitment applications which normally use only simple Boolean operations to compare information, irrelevant jobs would be matched, or a number of job descriptions will be obtained.

Evidently, without prior information on how to use online recruitment applications, the job seeker may feel drowsy with the numerous job recommendations offered by these applications based on the data inputs. In addition, through Boolean operations and other traditional matching techniques, these applications provide a long list of job advertisements and queries; thus, the job seeker himself needs to fill up massive form-based basic information which includes salary, job name and position level, experience and education background, skills and other related details. Considering that some of these queries are too broad or not clear, inputs about his job qualification may not be accurate.

Hence, this inaccuracy of data inputs perhaps causes problems which lead to irrelevant job recommendation.

Besides, a number of companies have different definitions for a single job position. Since every individual may have different understanding on a single term, the job seeker, by mainly and solely resting on his own understanding, may provide inaccurate input. In addition, since the traditional job matching applications are based on the basic approach, they hardly distinguish polysemy, synonymy and other meaning-related concepts. With this, it is quite challenging for the job seeker to determine the best job that matches his profile.

With the resurgence of interest towards effective employment, this research endeavour aims to introduce a job matching mechanism through latent semantic indexing and location mapping algorithms. Statistical classifier was used to categorized and ranked job seekers by linking the job description and skill requirements with the qualifications provided by employer.

OBJECTIVES OF THE STUDY

This research aims to develop improved algorithms to recommend suitable jobs to job seekers based on the collection and analysis of information, and applicants on a specific position to employers. Specifically, this research focuses on the following objectives: to design improved algorithm for job matching, to design a job-matching platform that can better propose suitable jobs for job seekers and employers to find prospective applicants for a specific job, to develop and implement the job matching algorithm using latent semantic indexing and location mapping algorithms and to test the application quality and usability.

METHODOLOGY

A. Software Development Methodology

In this study, the researchers trailed the test driven development (TDD) approach. Development of code in this approach requires automated test case encrypting and encoding codes to fulfill the test and refactoring. Its mantra defines red/green/refactor which enlists the order of programming tasks [1].

For red, a small automated unit test which shall not pass and compile at first shall be written. On the other hand, to pass the failing test a code is necessary for Green. Likewise, ensuring other tests has also pass (if present) in the suite is also needed. Checking in the code shall also be a pre-requisite. Meanwhile for refactor, in a gradual step without altering the intent, existing code is necessary to look organized.

B. Job Matching Platform Methodology

Given a set of job resumes, mapping similar ones is really a difficult task. There are lot of ways to determine similar concepts within a pile of resumes. Manual document evaluation, document classifying, and grouping are among the many ways to do. Identifying common concepts among the resumes/documents is a way to job matching.

For a concept-based approach, Latent Semantic Indexing (LSI) is frequently integrated. It consists of words and jobs forming relationships between words and jobs exported whenever there is a searching [2]. Mathematical properties of a job matrix and identification of concepts by matrix computation were used by the researchers in the conduct of this study.

For job matching concept-based approach is a more fitting method than the conventional method. On the other hand, to determine the similarities of two entities where job seekers input data will be utilized as an input query to find similar jobs from all job contents based on certain keywords that are grouped/matched, keyword matching method is applied. However, this approach is based on input query, content which have their different perspective use on the job position.

Context-based job matching is advantageous for employers who determine similar jobs. Moreover, a wider array of job can be looked into compared to that of the keyboard matching. Using different words and naming, employers can now provide a job description. This is one of the reasons why concept-based approach fits job matching.

Figure 1. Latent Semantic Indexing Method

In the middle of input query and targeted jobs, a concept is undefined and additional layer. This introduces job context rather than job content which is operationalized in mapping a query to jobs and vice versa. Concept are produced based on the semantic relationship between them and are not preset and fixed.

Figure 1 shows the latent semantic indexing method. This involves processes that have non-ordinary operations and objectives. In general, the collected data will be stored in jo database. Job database is a platform for input of datasets for supplementary processing whereas through succeeding processes, data are passed. The output of every process shall be considered as input for the further process. In figure 1, "pre-processed text for jobs" output pre-processor is an input to make a document vector. Similarly, one or more sub process like Tokenization, Lemmatization and Stopwords in pre-processor is contain in each process. A blogroll of similar jobs recommended to jobs seekers and applicants recommended for employers are the final output of this algorithm.

Pre-processor

Before the job database can be used in the job matching process, the raw datasets need to be pre-processed as raw datasets containing unanalyzed data. Pre-processor is a program that allows data cleaning and data filtering so that the irrelevant and duplicate datasets can be screened and purged before running an analysis. It also transforms the data sets into more representative and easy access format. This is the earliest stage that helps system to capture and manipulate datasets into the proper forms so that the job matching platform computation will be carried out smoothly.

Lemmatization

Lemmatization is one of the steps of pre-processing. Oftentimes, user refers item as the root word. According to Popovic, root word is the portion of a word that is left after the removal of its affixes [3] Lemmatization considers the structure and forms of words and reduces each word to its root form. Through lemmatization retrieval performance can be improved as variants of words are summarized to root words. Meanwhile, the size of indexing structure is relevantly diminished as the number of certain index words has reduced. In this research, lemmatization practical application is Porter Stemmer developed by Martin Porter in the late 1970s considered as one of the popular lemmatization algorithms [4].

Stop Words

Another part of pre-processing are the stopwords. This is used to filter words not important to overall context and text comprehension during pre-processing stage. Function words prevent a good search since they are less useful in searching as whole using search engine lens. These also apply in job matching field. Nevertheless, words considered with no significant relationship with job matching are removed. Stopwords list multiply through time since there is no fix set of it, about usage, good stopwords are tested by context and field [5]. These are also used to improve the searching accuracy and matching. Performance-wise, stopwords makes efficiency as it reduces total words of the overall context.

Tokenization

Tokenization is the last step in pre-processing. Through this stage specific patterns in raw datasets broken into stream of text or terms making it more manageable are discovered. Raw datasets contained in the job database create tokenization patterns. Illustration of the real implementation of this appears in the design and implementation section. In this study, datasets from job database were preprocesses using lemmatization, stopwords and tokenization. The output thereof will be send to matrix parser for further processing.

Term Vector Model

An algebraic model used to depict text documents and other objects in general as vectors of identified is the term vector model or connectively known as Vector Space Model.

Latent Semantic Indexing

A theory and method used to extract and represent the contextual use and meaning of words thru statistical computation applied to document list is latent semantic indexing on analysis [6]. The reason for the word Latent (hidden) is that the method doesn't use any sematic process. It is a mathematical process that enhances results semantically. However, the kind of semantic relation that is constructed during the modification process is not identified, but it could be determined by observing the results.

Singular Value Decomposition

Latent Sematic Indexing alters the term-document matrix with a linear algebraic method called Singular Value Decomposition.

Term Frequency Normalizer

To equalize the infrequent and common terms, value of frequency matrix is normalized for job matching platform. In other words, important terms

normally influence the similarity measurement and usually unimportant terms usually bring less or no values to the overall context. An infrequent or different term should weigh more heavily than common term in the comparison of job responsibilities and requirements as it helps to differentiate the job responsibilities and requirements.

Location Mapping

The location of jobs input of employers/recruiters are connected by the location mapping module. Likewise, the requested work locations input of job seekers is connected into "geocodes". These geocodes are latitude and longitudinal coordinates which are distances between locations and the positions between jobs and job seekers can be manipulated to determine relative distances between locations.

User Evaluation Method

The researchers utilized the graded relevance approach in the evaluation of this research. This evaluation technique is widely used in job matching studies. This is also referred to as normalized discounted cumulative gain (nDCG) [7]. The respondents of this study were the employee of government agency engage in recruitment and selection and their applicants and experts in the field of information technology. The total number of the respondents is 130 government employees and applicants and 10 information technology experts. The quota sampling was used in accordance to a given quota which represented the major characteristics of the population by sampling a proportional amount of each. It is a non-probability sampling wherein the selection of the sample is made by the researchers who has given quota to fill from specified sub-groups of the population [11].

A 4-point scale in nDCG evaluation method was used by the researchers for unit testing. These are Strongly Agree", "Agree", "Disagree" or "Strongly Disagree". Meanwhile, experts in research also helped in the evaluation process to critically retrieve and evaluate the study. The following scales and its equivalency used as follows:

Table 1. Numerical Rating

Range	Verbal Interpretation
3.5 – 4	Strongly Agree
2.5 - 3.49	Agree
1.5 - 2.49	Disagree
1 - 1.49	Strongly Disagree

RESULTS AND DISCUSSION

Improvement of Existing Job Matching Platform

There is multi-phase process in choosing the best job seeker or applicants for a job. The job is to be published first. Initial screening is the next process which is done based on rigid qualifications which are considered required like degree and major. Last, in choosing a candidate is based on knowledge, professional skills and interview.

Currently, some notable software for job matching are Sovren, Daxtra, TextKernel. Samples of snapshot of these mobile apps are shown in Appendix A. Most of these and other similar software's use the process known as resume parsing.

Resume parsing is used to convert free-form resume into structured information appropriate for storing, reporting and manipulating by a computer, it is a smaller part of the discipline called Natural Language Processing, which is constructing a structured document from an unstructured text.

Observing this software's and the algorithms shows a common difficulty. To extract semantic information from a text, they need hand-written rules about the structure of text [8]. In the case of a resume parser for example, they need to determine what type of information a body of text is referring to, whether it is referring to a course, or a company or address. To solve this problem, one needs a large database containing names of company address, work experiences and job description. This is known as an ontology. This can be a very tiresome task that needs constant updating. In addition, one needs a set of patterns that will represent key information on a resume.

In relation, the researchers developed an algorithm for job matching. The developed application introduces an algorithm for both job matching (matching jobs to candidates) and candidate matching (matching candidates to jobs). The developed job matching platform algorithm used statistical classifier trained on selecting applicants based on the applicant's profile matched with the job information and description provided by employer. Job matching platform is a mobile-application developed using Apache Cordova technologies.

To manage the business logic and to handle the users request to the server the developed mobile applications used PHP MVC framework. To access the MYSQL database the researchers utilized the XAMPP technologies. Because the application needs dynamic mobile web environment the developer used HTML 5

and JavaScript in designing the user interface. Posting a job, applying for a job, searching for a job or applicants are some of the major features of the developed mobile applications wherein users can access over the internet and perform activities which he/she is authorized.

Design of a job matching platform

The researchers used Unified Modelling Language (UML). This UML uses cases and classes diagrams to present the functional requirements of the developed job matching platform. To visualize, specify and document the behaviour of the system use-case diagrams was used by the researchers. According to Sparks, the building blocks of Object-Oriented programming are class diagrams [9]. These class diagrams are described by classes, attributes, operations and the relationships between them. Conceptualization, specification and implementation are some of the important aspects to consider for the critically and cohesively design the system.

The Case Diagrams are used to perceive the behavioural requirements of a system. This can also be used to present relationship among different actors in the system. The system and its user are the examples of actors. In utilizing the system, every actor has an important goal. Anything that the actor aims to achieve by using the system is a goal. All the unique goals that the various actors have in utilizing the system is gathered using the Use-Cases. Use-Cases are commonly found in the specifications of the requirement. Unified Modelling Language diagrams is the visual table of contents to written use cases [10]. Job seeker, job generator and system administrator are the main or key actors in the developed job matching platform. Their interaction is shown in Figure 2. Using the developed mobile applications, job seeker can search and can apply for a job

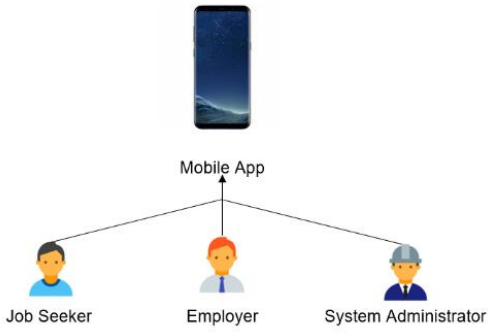

Figure 2: Actors in Job Matching Platform

posted by the employer likewise, the employer can post a job and select among the applicants the best fitted for the job posted. Moreover, the system administrator is responsible in maintaining the system.

The job seeker can create his/her profile, view job posted and apply for the jobs. Meanwhile, the employer could publish a job and review the profile of the candidates. On the other hand, system administrator typically maintains the system and troubleshoot the problems.

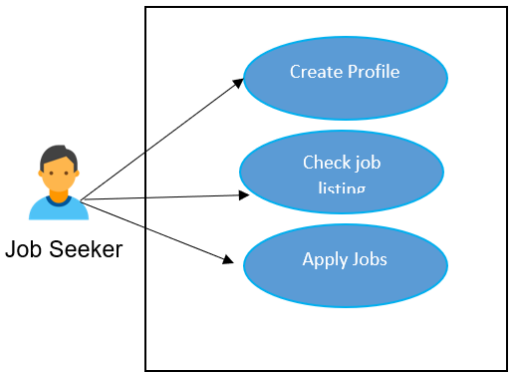

Figure 3: Use-case diagram of the job seeker

The use-case diagram in Figure 3 captures the goals of the user. The job seekers user creates his/her profile by filling the form with his/her personal information, academic history consisting of his/her major of study as well as his/her work experiences. The job seeker user checks the job listing available in the mobile applications and applies for a job by clicking the apply button.

The use-case diagram in Figure 4. describes the goals of the user employer. Employer user creates the job by filling a form with details like job title, job summary with skills required and employer details. The employer user checks the list of applicants and gets a list of applicants by running the job matching platform algorithm.

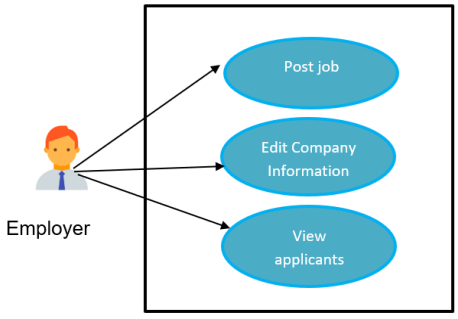

Figure 4. Use case of the employer

Application of latent semantic indexing and location mapping algorithms

Collecting the Documents

First step of the project was to collect job seeker and employer profile. Job seeker and employer profile was put in the system. All the job seeker profile samples provided in the system came from the different job seekers in Lipa City while the employer information particularly the job descriptions were obtained from a private agency.

Creating the Dictionary

Dictionary is the list of words contained in the corpus. To create an efficient dictionary, three main operations were made: tokenization, stop word removal and lemmatization.

Creating the Document Vectors and Term Document Matrix

Second task was to construct the document vectors. If the dictionary has n terms, then each job description was transformed into an n-dimensional vector.

Term Vector Model

Below is an example of term vector model application in the developed mobile app.

Document from job seeker profile: *Programmer Bachelor of Science in Information Technology Management Information System Data Analyst responsible for the management of systems and maintaining the data integrity and correctness.*

Document from employer's profile: *Senior Programmer Will program system for daily transaction Graduate of any Information Technology course*

To transform these documents into vectors, a dictionary of words will be created first. Let T be the list of terms(words).

T = *{Programmer, Bachelor, of, Science, in, Information, Technology, Management, Information, System, Data, Analyst, responsible, for, the, management, of, system, and, maintaining, data, integrity, and, correctness}*

T = *Senior, Programmer, Will, program, system, for, daily, transaction, Graduate, of, any, Information, Technology, course*

After applying lemmatization and stopwords removal, shortened list is identified:

T = {Programmer, Bachelor, Science, Information, Technology, Management, Information, System, Data, Analyst, responsible, management, system, maintain, data, integrity, correctness.}

T = {Senior, Programmer, Will, program, system, daily, transaction, Graduate, Information, Technology, course}

Now that the list is identified, vectors can now be constructed from the documents. The documents will be transformed into vectors where each dimension represents the number times the term occurs in that document.

Document from Job Seeker→ V1 = {1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1 }
Document from Employer→ V1 = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}

Now that the documents are transformed into vectors, certain vectors operations can be done to analyze the documents.

Cosine Similarity

In Term Vector model, this formula indicates the similarities of the documents. It can be a value between -1 and 1. For two documents from Job Seeker and Employer, if all the terms of Job Seeker and Employer are common, then cosine similarity will be 1. If no two terms are common, cosine similarity will be 0.

The Fig. 6 and 7 shows sample screenshots of implementing Latent Semantic Indexing algorithm.

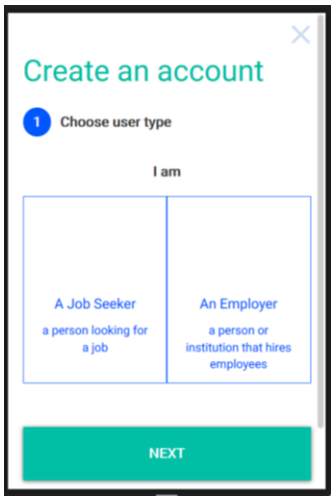

Fig. 6. Job Seeker and Employer account creation module

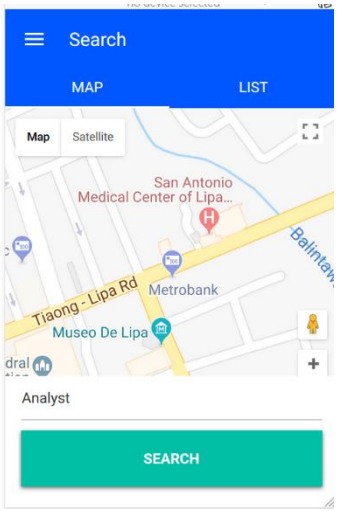

Fig. 7. Sample result of job matching in map

Evaluation Results

The system was evaluated on its acceptability in terms of user interface, functionality, compatibility and speed using *ISO*/IEC 25010:2011.

Table 2. Survey Result

No.	Criteria	Mean	Verbal Interpretation
1	User Interface	3.55	Strongly Agree
2	Functionality	3.43	Agree
3	Compatibility	3.55	Strongly Agree
4	Speed	3.50	Strongly Agree

Table 2 shows the result of the evaluation from the respondents. In terms of user interface respondents were tasked to assess the user-friendliness in the design of the developed mobile application. Respondents gave assessment of strongly agree on the user-friendliness of the user interface, given by the composite mean of 3.55. Functional testing was used by the researchers in the software testing process. In this case, software is tested to ensure that it conforms to all the requirements during the software development. Based on the respondent's assessment, the developed mobile application is functional with the composite mean of 3.43 and verbal interpretation of agree. This proves that this application is noteworthy for a job seeker and meet their requirements. However, compatibility was tested using the statements of whether the flow of the application is organized logically, reports and ease of use even without user manual help. This section got 3.55 composite mean with verbal interpretation of strongly agree which means that the study is compatible to the aspects. Another factor to consider in evaluating an application is its response time or speed. Meaning, it is on how fast or slow the mobile application interacts with the user. In this developed mobile application, it is operational, and speed is dependent on the user's internet connectivity. Overall, the result shows that the respondents strongly agree that the mobile applications is easy and fast to use and with composite mean of 3.5.

CONCLUSION

This section contains the conclusions formulated by the researchers after the conduct of this study. Moreover, these conclusions were rooted mainly from the answers on the statement of the problems.

Two algorithms were developed in this platform. The researchers critically looked into factors considered in the job matching. With the existence of different job matching platforms. The researchers intertwined the algorithm for job matching and algorithm for location mapping and be its unique feature and innovation in job matching applications.

In this job matching platform, statistical classifier was used to categorized and ranked job seekers by linking the job description and skill requirements with the qualifications provided by employer.

The innovation in this developed job matching platform is the infusion of Latent Semantic Indexing (LSI) and Location Mapping Algorithms. LSI is an information retrieval method. This enables the optimized analysis and identification of semantic relationships, patterns and commonality using a linear algebraic technique called Singular Value Decomposition (SVD). Almost all the essential information was derived using this technique.

The results and analysis of this matching method revealed that it is a functional platform in connecting job and employers without the need of human intervention along the line. Moreover, the integration of different approaches like tokenization, stopwords and lemmatization, the performance is successful with many numbers of returns of similar jobs. On the other hand, another facture this job matching platform is its flexibility for use on mobile.

The results presented in this research demonstrates the effectiveness of the proposed job matching methods. However, it could be further enhanced in few ways: developed mobile application job matching platform is a content-based recommendation system that is mostly focused on comparing the similarities between the job seeker profile and a relevant job information and qualifications. In future work, the

researcher could introduce a hybrid recommendation system that would take advantage of other recommendation algorithms such as Collaborative Filtering. Future work on this system would place greater consideration on job seeker's personal preference like job location, career development plan, and company background.

Acknowledgement

This endeavor is made possible through the help and support of all the contributors including mentors, professors, students, family and friends.

REFERENCES

[1] Beck, K. (2002) *Test-Driven Development: By Example*, Addison-Wesley Professional.

[2] Landauer, T.K. (2007). The Traditional Antilearning Argument. In Thomas K. Landauer, D.S.M.S.D.W.K. Handbook of Latent Semantic Analysis. Mahwah, New Jersey, London: Lawrence Erlbaum Associations, Inc. Ch. 1., 5–8.

[3] Popović, M. (2009). Machine Translation: Statistical Approach with Additional Linguistic Knowledge, Belgrad, Serbien: RWTH Aachen University.

[4] Porter, M.F. (1997). *Readings in information retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[5] Landauer, T.K. (2007). *The Traditional Antilearning Argument. In Thomas K. Landauer, D.S.M.S.D.W.K. Handbook of Latent Semantic Analysis*. Mahwah, New Jersey, London: Lawrence Erlbaum Associations, Inc. Ch. 1. pp.5 – 8.

[6] Price, R. J.; Zukas, A. E. (2005). Application of Latent Semantic Indexing to Processing of Noisy Text, Intelligence and Security Informatics. Lecture Notes in Computer Science. 3495.p. 602. *ISBN 978-3-540-25999 2*. *doi*:*10.1007/11427995_68*.

[7] Kekäläinen, J. (2005). Binary and graded relevance in IR evaluations: comparison of the effects on ranking of IR systems. Information Processing and Management: an International Journal", 41(5), 1019 – 1033.

[8] Prakash M. N., Lucila O. M., Wendy W. C. (2011). *Natural language processing: an introduction*, J. Am. Med. Inform. Assoc., vol. 18, pp.1.

[9] Sparks, G. (nd). *Database Modelling in UML"*, Retrieved 8 September, 2011

[10] *Unified Modeling Language User Guide, The (2 ed.).* Addison-Wesley. *p. 496.* ISBN 0321267974. 2005,

[11] *Quota Sampling*. Retrieved May 25, 2018 from Explorable.com: https://explorable.com/quota-sampling

COPYRIGHTS

Copyright of this article is retained by the author/s, with first publication rights granted to APJMR. This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creative commons.org/licenses/by/4.