

**Technology**

**(Web Applications and relational database)**

**Student Name :** Ong Qi Yong

**Admin no. :** 1004142D  
**MP Group :** SOLY  
**Project ID :** FBI1215

**Project Title :** nTrader

Contents

1. **Introduction ----------------------------------------------------------------------------------------------- 1**
   1. **Background ------------------------------------------------------------------------------------------ 1**
   2. **Scope of review -------------------------------------------------------------------------------------- 1**
   3. **Constraints and limitations ----------------------------------------------------------------------- 1**
      1. **Time constraint ------------------------------------------------------------------------------1**
      2. **Financial constraint ------------------------------------------------------------------------ 1**
      3. **Skill Limitation (Manpower Constraint) -----------------------------------------------1**
2. **Programming Languages -----------------------------------------------------------------------------1 - 2**
   1. **PHP ---------------------------------------------------------------------------------------------------- 2**
   2. **Java --------------------------------------------------------------------------------------------------2 - 3**
   3. **Python ------------------------------------------------------------------------------------------------ 3**
   4. **Analysis and decision ----------------------------------------------------------------------------- 3**
3. **Cloud Computing ---------------------------------------------------------------------------------------- 3**
   1. **Cloud Server ---------------------------------------------------------------------------------------- 4**
   2. **Dedicated Private Server ------------------------------------------------------------------------- 4**
   3. **Comparison of cloud server and dedicated server ---------------------------------------- 4 - 7**
4. **Lexical Databases --------------------------------------------------------------------------------------- 7**
   1. **The Dictionary database model --------------------------------------------------------------- 7 - 8**
   2. **The Morphological database model ---------------------------------------------------------- 8 - 9**
   3. **The MorDebe Model --------------------------------------------------------------------------- 9 - 10**
   4. **SOLY’s choice – WordNet database ---------------------------------------------------------- 10**
5. **Conclusion ----------------------------------------------------------------------------------------------- 10**

[**Bibliography**](#_Toc341926460)

**Appendix**

1. **INTRODUCTION**
   1. **BACKGROUND**

Technology has revolutionised many ways on how people do things and one big benefit that technology has brought to us is automation of processes. Project nTrader aims to leverage on the power of automation to analyse news rapidly to facilitate faster and more informed decision making for investors. In order to do so, many technologies had been used and this document will facilitate the understanding of the various technologies that are available as well as the technologies that Project nTrader has adopted.

* 1. **SCOPE OF REVIEW**

This document will serve as a comparison of various technologies however not all technologies used in the nTrader project will be discussed in the document. This document will focus on the main components that Project nTrader relies on and provide a detailed analysis on these components as well as other technologies that the industry in general uses. The following are the main components that will be discussed in this document: Cloud computing, Lexical Database Models and Programing languages.

* 1. **CONSTRAINTS AND LIMITATIONS**

Suggestions that are shown in this document are subjected to various constraints and limitations that are placed on SOLY group and thus it is important to first understand the various constraints that SOLY group face to better understand how decisions are made.

* + 1. **Time Constraint**

Project nTrader is a 10 week project and thus not all technologies that are available can be implemented despite it being the better solution. Time constraint also accounts for the time that SOLY group will require to learn new skills and understanding the controls of technologies that are unknown.

* + 1. **Financial Constraint**

Certain technologies require SOLY obtaining certain hardware or software before being able to use them. As SOLY group has limited funding, many of these paid technologies would not be available for us. Financial constraint can also affect decision making as SOLY group would take careful consideration to emphasize on efficiency by avoiding technologies that takes up significantly larger amount of resources despite it being more effective. This is mainly due to the higher cost that arises due to high resource usage.

* + 1. **Skill Limitation (Manpower Constraint)**

SOLY group members have limited skills set and thus not all technologies that are used in the industry can be successfully implemented into the nTrader project. Simple technologies that are easy to learn and adopt will however be taken into consideration for implementation.

1. **PROGRAMMING LANGUAGES**

nTrader is a web-based application and thus web application development skills are essential in this project. There are a wide variety of programming languages that are used in web application development however this section will focus on the most suitable web application development technologies (Programming Languages) that are used in the industry for web-based analysis of large amount of text.

* 1. **PHP**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML (The PHP Group, 2012). PHP is a very simple language to learn and it has fewer restrictions compared to languages such as Java and C++ which provide a high level of ease of use. PHP is most often used in the development of dynamic websites due to its ability to integrate with HTML, CSS and JavaScript. Having a high level of compatibility with these web design codes allow web developers to use PHP to create dynamic web pages by simply placing the codes in the same file and run it through a web processor.

PHP is a server sided language and as such, no processing is done on the user’s system. Server sided means that the server will process the PHP codes and generate HTML codes as output to the client (User). The client would thus be unable to configure the codes used and interfere with the webpage settings and output content directly. This provides more security for developers as developers can have better control on what content is to be shown to the client by using PHP to control the HTML output.

Besides its high compatibility with web design codes, PHP is also highly compatible with many operating systems and platforms such as Windows, Unix and Linux (Sanskruti, 2007). It also supports a large amount of database such as MySQL and Oracle which makes it undoubtedly the best programming language for front end web development such as web page developments (jotorres, 2012). In a web application project such as nTrader, PHP will definitely be an essential skill to have in order to develop a dynamic and interactive website that has a good user interface.

* 1. **JAVA**

Java is a programming language and computing platform that is first released by Sun Microsystems and it is the underlying technology that powers satate-of-the-art programs, utilities, business applications and games (Oracle, 2012). Java is an object-oriented language that can be run on any java virtual machine regardless of computer architecture which makes it a very widely used language. Unlike PHP, Java is client-sided which means that the program is run on the client’s system. This means that the user will be required to download a java virtual machine before it is able to run the program.

Being a client-sided programming language also means that people can interfere with the codes to create unexpected results and exploit unexpected security loopholes. Although Java is not as convenient and simple to use as PHP, Java does have certain benefits over PHP and one of these benefits is its efficiency compared to PHP. As shown in the figure (Refer to Figure A), Java performs up to 30 times faster than PHP which makes it highly suitable for text analysis. SOLY group has adopted the natural language processing (NLP) algorithm to perform text analysis for project nTrader and NLP deals with large amount of data and text to obtain reliable results. Therefore performance has become a very important factor when deciding which programming language to adopt for the project and Java triumph over PHP in the performance aspects.

* 1. **PYTHON**

Python is an open-source object-oriented high level programming language that offers two to ten-fold of programmer productivity over languages such as C, Java and Visual Basic (Wingware, 2007). Similar to PHP and Java, Python is compatible with many operating systems and platforms such as Unix, Mac and Windows. Performance wise, Python can potentially be slower than PHP (Refer to Figure B) however Python possess a large standard library which greatly helps in Natural language processing and parsing of text. Using these libraries, many complicated logic can be easily performed which indirectly improves performance.

Python is designed in a way to write codes quickly and thus is useful for hypothesis testing and quick implementation of certain features. It is also highly compatible with many other programming languages such as Java and C++ which helps to facilitate development work when using other languages as well.

* 1. **ANALYSIS AND DECISION**

SOLY group has decided to pick up PHP as the only programming language that will be used in project nTrader. Although Java has a superior advantage over PHP in natural language processing, SOLY group members have no experience with the Java and Python language. Furthermore, nTrader is highly reliant on a database to process and identify words and being a web page, PHP has a natural advantage over Java and SOLY group members could also use PHP to code simple NLP algorithms using PHP. Due to the constraints that were placed on SOLY group, the scope of the project is also smaller and thus compared to industry standard NLP algorithms, SOLY will perform algorithms targeted on a smaller sample set which can reduce complexities that derive from large amount of text. To add on, Java uses a larger memory (Refer to Figure A) and that may add on greater stress on the cloud server which may incur higher cost. Therefore, SOLY group members preferred programming language is PHP however future improvement opportunities can be to explore the usage of Java in more complicated NLP algorithms for text analysis.

1. **CLOUD COMPUTING**

SOLY group has decided to use the internet as a platform for nTrader due to its ability to distribute content to users rapidly regardless of physical location. In order to do so, SOLY will require a web host server to store and process its content. As technology improves, more and more options become available for web application developers and cloud computing has quickly become one of the most popular platform for web application hosting. This section will discuss on the traditional physical dedicated server versus a cloud virtual server for web hosting as well as the pros and cons that each type of server will have.

* 1. **CLOUD SERVERS**

Cloud computing is a general for anything that involves delivering hosted services over the internet (Rouse, Cloud Computing, 2010). As such content are not delivered physically and instead they are delivered through a virtual environment. Cloud server is one major component of the cloud computing technology and it is essential during web application development. As said earlier, more and more businesses is increasing rapidly as more and more business shift towards cloud computing. Cloud server demand has surge greatly since the past two years and has since doubled since then (Refer to Figure C). Due to technological advancement, high speed internet has become more available to the public and businesses seek new technological solution such as cloud computing to reduce IT expenses which will continue to drive cloud servers demand in the future. It is predicted that this trend of cloud computing will keep going and will double again by 2015 (Johnson, 2012).

* 1. **DEDICATED PRIVATE SERVER**

A dedicated server is a web server that is leased or owned outright which provides the owner exclusive use of the web server resources (Kyrnin, 2012). A dedicated server provide many benefits however the key benefit of owning a dedicated server is the level of control the owner has. With a high level of control, owners have more flexibility as they can customize the dedicated server to suit their needs. This is very beneficial for web application development as not all technologies can be run on all systems and with a high level of control, the dedicated server can be tuned to use certain operating systems and software that can best suit user needs in development. However dedicated server has high overheads and maintenance cost thus not everyone have access to a dedicated server.

* 1. **COMPARISON OF CLOUD SERVER AND DEDICATED SERVER**

Both cloud server and dedicated server have their own pros and cons therefore careful comparison and analysis need to be done before committing to use these technologies. Cloud server generally provides key benefits such as elasticity, simplicity and flexible pricing models while dedicated server provides key benefits such as security, performance and control. Each of these key benefit is essential in web application development and their level of importance is difficult to measure thus SOLY need to analyse which of these will be more suitable for the nTrader project and a detailed comparison have to be done. The table below will highlight and elaborate key differences of a cloud server and dedicated server for comparison.

|  |  |  |
| --- | --- | --- |
| **Comparison factor** | **Cloud Servers** | **Dedicated Private Servers** |
| **Operating System** | Operates on a dedicated operating system and support a wide range of operating systems. Switching of software and operating systems can be done by changing a few settings. | Operates on a dedicated operating system however hardware may not support other operating and legacy systems. |
| **Installation** | Installation of software is very simple and can be done remotely. | Installation of software may be difficult and gap analysis may be required. The installation process may also take up significant amount of time. |
| **Configuration and management** | Configuration may be difficult as server settings cannot be changed directly. Workarounds may be required to achieve the same effect as a dedicated server which may result in a reduction of system performance. | Configuration and management is easier compared to a cloud server and there are fewer limitations imposed on the owner which provides more control over the server. |
| **Access** | Highly accessible to users and users can connect to the server remotely to perform task. | Users are unable to remotely access the server and remote access is unavailable. |
| **Server Snapshot** | Server snapshots can be taken to save the state of the server in the case of unexpected events. Server can load the last saved snapshot to revert to its original last working state. This will be useful when testing if certain codes are compatible with the server during development. | Server snapshots cannot be taken and any changes may affect the system permanently. However a backup can be done which will significantly increase redundancy and reverting process will be longer than using a server snapshot. |
| **Performance** | A cloud server performs slightly better than low end and average dedicated servers. However internet speeds will be a limitation and bottleneck situations may happen when there is high traffic incoming. | Large scale and high end dedicated servers perform much better than cloud servers and have better traffic control compared to cloud servers. However the costs of such servers are significantly greater. |
| **Server Redundancy** | If a physical server fails, cloud servers will automatically be switched to other live server which lowers risk of failure as well as reduce redundancy of having to prepare a backup. | If a dedicated server fails, a new server will need to be acquired and software will need to be reinstalled and reconfigured. Having a backup server will increase redundancy and cost. |
| **Power Redundancy** | Cloud servers run on a shared resource pool which is served on redundant power sources. Thus risk of power failure is reduced as there are other redundant power sources to continue to power the resource pool. | Dedicated servers usually use a single power supply and a backup power supply may be required in the situation of power failure. |
| **Hardware Resource Scaling** | CPU power, memory and disk space can be added or removed while the server is running. This scaling process can also be automated by code and thus there is no need to estimate traffic flow and prepare sufficient resources to handle the traffic. Server can also be optimized this way and all hardware resource can be fully utilized and reduce resource redundancy. | Dedicated server has to be shut down before hardware resources can be modified. This creates a downtime as the server will be unavailable while hardware resources are changed. Traffic flow has to be estimated to ensure that the server has sufficient resources to handle the traffic. This reduces efficiency and increase redundancy as hardware resources are not fully optimized. |
| **System Upgrades** | Reinstallation of operating systems and drivers are not required when upgrading to a more powerful server. | Reinstallation of operating systems and drivers are required and server has to be reconfigured when upgrading to a more powerful server. |
| **Disaster Recovery** | If the cloud server fails, the backup cloud image can be used while the recovery process is on-going which negates server downtime. | If the cloud server fails, system files and database needs to be restored which will take a few hours. This creates a server downtime. |
| **Pricing Structure** | Cloud server providers generally have a pay-for-usage basis pricing structure which means that user only pay for the resources that they use and any redundant resources that are not used does not require payment. | Dedicated server providers generally have a fixed pricing structure in the form of plans. Users pay a fixed cost for the plan and any upgrades or changes may incur additional cost. This creates redundancy as not all resources will be utilized and the user has to pay for it as well. |
| **Cost** | Cloud servers cost slightly more expensive than a dedicated server on a per hour basis for the same amount of resource usage. | Dedicated server may be cheaper than cloud servers however pricing structure limitations may result in an overall higher cost compared to cloud servers. Thus it is only beneficial when the server has to be run 24/7 for a long period of time. |
| **Security** | Cloud servers are a shared group of computers and as such, users have no control on the security of the content that is in the cloud server. Furthermore, cloud servers can be access globally via internet connection and this makes it easier for hackers to breach. | Dedicated server is a computer that is fully owned by the user which allows the user to configure user settings and privileges which can provide better security by managing all the user access rights. |
| **Mobility** | Cloud servers are managed in a virtual environment thus everything is stored virtually. This makes shifting of server location easy as user will just need to transfer the data over to another cloud server located at another region to host webpages on that region. | Dedicated servers are harder to move as hardware has to be physically shipped to the new location. |
| **Dependency** | Cloud servers are usually controlled by the service provider and thus by using cloud servers, users will be dependent on the service provider and changes made by the service provider may affect the user. Etc. Change of pricing structure of cloud servers or change in security policies. | Dedicated servers are usually fully owned by the user while the service provider is only responsible to setting up the server and performing system upgrades and changes on demand. Thus user is less dependent on the service provider as other service providers can perform the same task and user will not be always dependent on the same service provider. |

Based on the detailed comparison above, SOLY eventually decided to commit to using a cloud server. The main reason for choosing a cloud server instead of a dedicated server is the cost. As compared above, a cloud option has a more flexible pricing structure which is very essential as SOLY will only have to pay for what we use and do not need to fully commit into spending money for a dedicated server. This not only gave us the flexibility to start up multiple cloud servers for testing but also gave us more options such as exploring into technologies such as Hadoop (cloudera, 2012) which will not be possible if we are not committed to spend large amount of money on hardware. Server security is an important factor but SOLY feels that the data that we manage is open to public and as such there is no need for high levels of security. Although security would not be an issue, server efficiency is a greater concern for SOLY as the NLP algorithm that is used in nTrader will require large amount of system resources. To address this issue, SOLY has decided to combine the cloud server and database by directly installing the database into the cloud server and by doing so, SOLY will be positioning the database in an internal environment rather than an external environment which will reduce network connection bottlenecks and improve data processing efficiency. As a result of doing so, SOLY will be trading off system resources for efficiency and cloud server scalability would be very advantageous for SOLY in this situation as we would be able to increase system resources when required. In conclusion, a cloud server would be a better solution for SOLY after considering the constraints that SOLY have.

1. **LEXICAL DATABASES**

Databases are a key component in many current technologies and for a project like nTrader, a database is very essential to provide a storage place for data that will be used in the algorithms. The database used in the nTrader project are relational databases which means that data items are organised as a set of formally-described tables from which data can be accessed or reassembled in many ways without having to reorganise the database tables (Rouse, relational database, 2006). Relational databases can be used for many purpose in web application development however for project nTrader, SOLY has decided to structure it as a lexical database to facilitate the NLP algorithm used. A lexical database is a database of information about words (thefreedictionary, 2012) and it is chosen for nTrader due to its ability to facilitate efficient processing in NLP. There are many types of lexical databases however I will focus on three types of lexical databases that are used for NLP in this document.

* 1. **THE DICTIONARY DATABASE MODEL**

A dictionary database model can be described as a database whose main function is the compilation of lexicographic products (Janssen, 2005). The main characteristic of a dictionary database is that it compiles words based on the lemma. Thus when querying through the database for results, the headword is set to match the lemma. A headword can be described as a word that serves as the heading for an entry in a dictionary, encyclopaedia or similar reference work (thefreedictionary, 2009). In the context of this document, a headword is used as a search criterion in the database to obtain all information of the word such as related words (Synonyms), lexeme and word senses (context of word).

As a dictionary database is based around the lemma, all the lemma in the English language will be stored under a single table while its relation such as lexeme, sematic and various information are stored in separate tables and are tagged onto the main table that stores the lemma (Refer to Figure D for sample diagram of structure). Since the headword is set to search the lemma table, the queries will need to pull data from other tables in order to obtain more information of the lemma for processing. For example: the lemma, “run”, can mean a fast movement by using one’s feet when used as a verb while it can also mean the continuous period of time a machine is in operation when used as a noun. The database will have to extract all available information of the headword by using the tags of the lemma to refer to the proper relevant information in other tables.

Based on the above example, a dictionary database model can be thus seen as the most basic form of a lexical database design as it provides the basic skeleton on how a lexical database can be developed as seen by the database structure. By basing on the lemma, a dictionary database will have the advantage of using the relation between each table to add words easily. A new lemma can be easily managed and its related information such as lexeme can be stored in the various respective tables. This also provides the flexibility for a dictionary database to add new relation to a lemma by simply creating new tables and tagging it to the main lemma table. Although using such a design can prove to be very efficient in terms of data insertion and management, the same cannot be said for data retrieval. The issue with data retrieval through a dictionary database is that users will not be able to directly retrieve lexemes and various word-forms as the headword would not be able to directly match it from the lemma table. Since the lemma table strictly stores only the lemma, lexemes would naturally not exist in the lemma table and in order to obtain information of lexeme like “running”, the database would need to search across multiple tables which not only is difficult but also inefficient. This problem could however be solve by creating database views to combine the lexeme and lemma under the same table and using the headword to search the view.

* 1. **THE MORPHOLOGICAL DATABASE MODEL**

If a dictionary database is based on the lemma, the morphological database can be described as a database that is based upon the morpheme. According to language development organisation SIL International, a morpheme can be defined as the smallest meaningful unit in the grammar of a language (SIL International, 2004). In a morphological database, the headword is set to match the morphemes or inflectional and derivational paradigms of a word. Morphemes are stored in a similar way as a word in a table however there are significant difference between a morpheme and a word. A morpheme is a standalone meaning while a word can be made up of multiple morphemes. A free morpheme can function as an independent word while a bound morpheme appears as part of words in the form of affixes.

Bound morphemes can be derivational or inflectional. Derivational morphemes change part of a speech or the basic meaning of the word by acting as affixes in the English language while inflectional morphemes vary the form of words in order to express the grammatical features and do not change basic syntax category (Part of speech) (Lieberman, 2012). An example of a derivational morpheme is “re-”. When it is combined with a word “activate”, it change its meaning to activate again. An inflectional morpheme like “-s” when added to the word “tree” does not change its original meaning or part of speech. However it changes the word-form of the word and turns it to a plural form.

Compared to a dictionary database, morphological database can be more effective in data retrieval and large amount of more relevant data can be extracted using the morpheme. By using the morpheme, user will be able to easily extract all inflectional paradigms and differentiate them with derivational paradigms of words. This will aid NLP algorithms greatly as NLP algorithm can not only identify the lemma through the morphemes but it is also able to easily identify the part of speech and sematic of a word as these information are fixed onto the morpheme table.

* 1. **THE MORDEBE MODEL**

The MorDebe model is one of the most comprehensive database designs for a lexical database. MorDebe is a Portuguese lexical database that aims explicitly at the use of a single set of lexical data in a wide range of application, including both NLP systems and human consultation (Janssen, 2005). Although MorDebe is a Portuguese database system, the concept of MorDebe is largely language independent and can be applied to English lexical databases as well. MorDebe is designed to be based around full-form words instead of the lemma. Full-form words will include the various lexemes, inflected paradigms of a lemma, part of speech and all available details of the word. The headword will therefore be set to match the full-form words instead of the lemma.

By creating a table with full-form words, the database becomes difficult to manage. Information in a full-form word database may not be relevant and related however in the MorDebe model, a secondary table is used to capture all the lemmas. This allows the full-form words to be organised around the lemma and to relate back to them. This structure will also allow inflectional relationship to be shown through the tagging of full-form words to the lemma (Refer to figure E for diagram). Although inflectional relationship and various lexeme relationships to a lemma can be explicitly shown through this method, the MorDebe database faces one key weakness which is the large amount of space it requires to hold the entire list of full-form words in a single table.

The MorDebe model will bring about many advantages over the standard dictionary database design and the key advantage will be of course the data retrieval. As explained earlier, a dictionary database relies on lemma to retrieve data and if a user specifies the headword as an inflected form, the database would not be able to provide information directly as compared to the word-form driven MorDebe database. One example would be the headword “mice”. A dictionary database would not be able to identify the word as it does not possess the capabilities to directly recognise that “mice” is actually the inflected form of “mouse”. A MorDebe database however treats “mice” and “mouse” as two words and thus it would be able to identify the word “mice”. Through the relation to the lemma, the MorDebe database could then obtain all the inflected relationship information of the word “mice” and recognise it as a plural form of the word “mouse”

The second major advantage of the MorDebe model will be its ability to separate lemma over the dictionary database. As a MorDebe database stores full-form words, it is able to clearly identify one or multiple lemma that the word belongs to and thus words that has multiple word-sense would not be classified under dubious lemmas. An example would be the word “ring” when in the inflected form can either be “ringed (bird)” or “rang (phone). In a dictionary database, this cannot be separated into two lemmas as the lemmas are used to match the headword and thus they cannot be duplicated. However in a MorDebe database, full-form words are used to match the headword and thus the lemma can be duplicated to represent two lexical entries.

* 1. **SOLY’S CHOICE – WORDNET DATABASE**

Although, all three design discussed earlier are feasible designs to use in the nTrader project, SOLY is limited by the skill and time constraints to build such databases. Although the scope of the project could be reduced and a sample set of data could be used to create these database designs for NLP algorithms, SOLY found a better option which can not only allow SOLY to create a complete lexical database but also create NLP algorithms that are reliable within the time and skill constraints. Thus said solution would be to adopt the WordNet database.

WordNet is a free and public lexical database that is started by George A. Miller in the mid-1980s (WordNet, 2012). WordNet is a lemma based lexical database as such it highly resembles a dictionary database. However WordNet incorporated certain aspects of a Morphological database by grouping similar words that have the same meaning together as “Synsets (Cognitive synonym sets)”. In order to do so, WordNet linked the word-forms, lemmas, word-sense together into a view and based on the sense of each word-form, perform grouping into “Synsets”.

WordNet may not be the best solution for nTrader project however it’s lexical database capabilities is sufficient for SOLY to perform NLP algorithm development for news. Although WordNet provided a quick solution for SOLY, it is not catered towards the financial keywords and instead catered towards general English language. Thus SOLY would also be required to modify certain areas of the WordNet database before it is usable in any actual NLP algorithm that SOLY creates.

As WordNet is more similar to a dictionary database, it is also not easy for SOLY to modify the database and gear it towards financial terms and context. A word-form based database such as the MorDebe model would be more ideal as words can be freely added and tagged to lemmas rather than adding lemma for financial terms. However, with the flexibility of a dictionary database, SOLY would still be able to indirectly incorporate financial keywords tagging to the current WordNet design through other tables.

1. **CONCLUSION**

This document has covered many areas of technologies that SOLY can use in the development of nTrader. Due to many constraints and limitations, SOLY may not be able to incorporate all the industry level techniques or use industry level technology on project nTrader. However this document has also highlighted various alternatives that SOLY can consider and many technologies such as the cloud server and WordNet has allowed us to eliminate the difficulties and limitations that we may face during the development process.

# Bibliography

cloudera. (2012). *Hadoop and Big Data*. Retrieved from cloudera: http://www.cloudera.com/content/cloudera/en/why-cloudera/hadoop-and-big-data.html

Janssen, M. (2005). *Lexical vs. Dictionary Database.* Lisboa, Portugal: Instituto de Linguística Teórica Computational.

Johnson, R. C. (19 1, 2012). *Cloud Server Market Exploding*. Retrieved from Smarter technology: http://www.smartertechnology.com/c/a/Cloud-Computing/Cloud-Server-Market-Exploding/

jotorres. (23 3, 2012). *My top 9 reasons why I use PHP*. Retrieved from joTORRES: http://www.jotorres.com/2012/03/my-top-9-reasons-why-i-use-php/

Kyrnin, J. (2012). *dedicated server*. Retrieved from about.com webdesign: http://webdesign.about.com/od/webhosting/g/dedicated\_server\_definition.htm

Lieberman, P. M. (2012). *Morphology*. Retrieved from Linguistics 001: http://www.ling.upenn.edu/courses/Fall\_2007/ling001/morphology.html

Oracle. (2012). *What is Java technology and why do I need it?* . Retrieved from Java: https://www.java.com/en/download/faq/whatis\_java.xml

Rouse, M. (4, 2006). *relational database*. Retrieved from SearchSQLServer: http://searchsqlserver.techtarget.com/definition/relational-database

Rouse, M. (12, 2010). *Cloud Computing*. Retrieved from searchcloudcomputing: http://searchcloudcomputing.techtarget.com/definition/cloud-computing

Sanskruti. (21 5, 2007). *Why use PHP for server-side scripting*. Retrieved from go4expert: http://www.go4expert.com/forums/showthread.php?t=4385

shootout. (25 11, 2012). *Computer Language Benchmarks Game* . Retrieved from http://shootout.alioth.debian.org: http://shootout.alioth.debian.org/u32/benchmark.php?test=all&lang=java&lang2=php

SIL International. (5 1, 2004). *What is a morpheme?* Retrieved from SIL: http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsAMorpheme.htm

The PHP Group. (23 11, 2012). *What is PHP?* Retrieved from php.net: http://php.net/manual/en/intro-whatis.php

thefreedictionary. (2009). *headword*. Retrieved from thefreedictionary: http://www.thefreedictionary.com/headword

thefreedictionary. (2012). *lexical database*. Retrieved from thefreedictionary: http://www.thefreedictionary.com/lexical+database

Wingware. (30 7, 2007). *What is Python?* Retrieved from pythoncard: http://pythoncard.sourceforge.net/what\_is\_python.html

WordNet. (9 11, 2012). *What is WordNet?* Retrieved from WordNet: http://wordnet.princeton.edu/

**APPENDIX**

|  |
| --- |
| **Figure A** |

|  |
| --- |
| **Figure B** |

|  |
| --- |
| http://www.smartertechnology.com/images/stories/cloudservers.JPG  **Figure C** |

|  |
| --- |
| **Figure D** |

|  |
| --- |
| **Figure E** |