

Lab – 1 C.L.A.S.H. Product Description

Blue/Purple Teams

Cory Morewitz

CS 411W

Janet Brunelle, Hill Price

Old Dominion University

February 9<sup>th</sup>, 2015

Version 1

### **Table of Contents**

|  |    |
|--|----|
| <b>1 INTRODUCTION</b> .....                            | 3  |
| <b>2 PRODUCT DESCRIPTION</b> .....                     | 4  |
| 2.1 Key Product Features and Capabilities .....        | 5  |
| 2.2 Major Components (Hardware/Software) .....         | 5  |
| <b>3 IDENTIFICATION OF CASE STUDY</b> .....            | 7  |
| <b>4 CLASH PROTOTYPE DESCRIPTION</b> .....             | 9  |
| 4.1 Hardware and Software Prototype Architecture ..... | 11 |
| 4.2 Prototype Features and Capabilities .....          | 12 |
| 4.3 Prototype Development Challenges .....             | 13 |
| <b>GLOSSARY</b> .....                                  | 14 |
| <b>REFERENCES</b> .....                                | 15 |

### **List of Figures**

|  |    |
|--|----|
| Figure 1. Major Functional Component Diagram.....        | 7  |
| Figure 2. Current Process .....                          | 8  |
| Figure 3. Future Process .....                           | 9  |
| Figure 4. Prototype Hardware/Software Architecture ..... | 12 |

### **List of Tables**

|   |    |
|---|----|
| Table 1. Real World vs. Prototype ..... | 11 |
|---|----|

## Lab 1 – CLASH Product Description

**1 INTRODUCTION**

C.L.A.S.H, or Color Lexical Analysis algorithm and Slash Handler is a web interface that includes two modules, 'COLRS' and 'Slash'. The 'COLRS' module 'colorizes' each part of speech (P.O.S) in a text document with a particular color to help increase comprehension of sentence structure and grammar. The 'Slash' module parses text into lexical bundles (a group of words that occur repeatedly together, or present one single thought), or thought groups to help increase reading speed and comprehension. A lexical bundle is a group of words that occur repeatedly together within the same register. Though our program could be useful in a number of settings, we are primarily concerned with English as a second language (ESL) students.

Based on state-reported data, in 2004, it is estimated that 4,999,481 ESL students were enrolled in public schools. Despite those numbers 15% of these ESL students had no special resources or programs to help them learn the language. This problem is systemic. In 2001, in the states that tested ESL students in reading comprehension, only 18.7 percent of ESLs were assessed as being at or above the norm. That same year, 10 percent of ESLs in grades 7-12 were retained. In February 2001, it was reported that ESLs had dropout rates up to four times that of their native English-speaking peers. Here at ODU, we have an entire department focused on addressing this problem, but while they work diligently to help ESL students, the processes of teaching reading and grammar are outdated. (McKeon)

Currently, the process is simple, for grammar, the professor writes a sentence on the board, then circles, or marks in some way each part of speech. Not only is this time consuming, it limits the amount and size of examples that can be given. Psychologically, it has been proven that color impacts learning, they relieve eye fatigue, increases information retention, increase

productivity and accuracy, and support developmental processes.

(<http://sdpl.coe.uga.edu/HTML/W305.pdf>) Extrapolating from that information the 'COLRS'

module should help students identify P.O.S. and increase their potential for learning grammar.

For increasing reading speed and comprehension, reading assignments are given, or students are

directed to sites like Spreeder(<http://www.spreeder.com>). These sites don't help comprehension

due to their focus on speed; they teach students to read word for word vs. in lexical bundles. It

has been shown that those who learn lexical bundles read faster and perform better in word and

sentence recall experiments.(Tremblay, Derwing, Libben, Westbury)

## **2 PRODUCT DESCRIPTION**

CLASH is a single page web application that will help ESL students improve reading speed and comprehension by parsing text into an easier to read format. The graphical user interface will be simple, and easy to use and contain three modules. These modules: Slash, Slash Playback, and COLR to delineate and mark parts of speech (P.O.S.) and lexical bundles.

The 'Slash' module will parse documents into 'lexical bundles'. This will help non-native speakers/readers transition from reading word for word to reading in thought groups. The slash module will also feed into a playback function that will allow for students to slowly increase the speed of their reading.

The COLRS module will parse text, identifying parts of speech and coloring them. This will allow for instructors to easily display and therefore teach grammar. This will lead to increased comprehension, and recognition of syntax.

## 2.1 Key Product Features and Capabilities

CLASH will be a single page web-based application that will provide an easy to use interface. The CLASH site will be accessible through any standard browser with access to the internet. Using the intuitive user interface, users will be able to display parsed text that has been COLR'd, Slashed, or both. The Slash module will feed into a Slash Playback module that will display the lexical bundles. The COLR module will allow users to single out particular part(s) of speech to display so they can easily study one, or multiple concepts.

CLASH will have three different user roles, delineated by login. Roles will be administrator, instructor, and student. The student will be able to view parsed documents, and interact with the Slash Playback module. The instructor will be able to input text, view the parsed output, edit the parsed output in case of error, as well as save and delete documents from the server. The prototype will attempt to provide reporting function for the instructor to view student activity and progress. The administrator will have all the previously mentioned functionality plus the ability to add or remove instructor permissions to particular users.

## 2.2 Major Components (Hardware/Software)

CLASH is a web application hosted on a virtual machine and accessed through a browser. There are no special hardware requirements, as CLASH is mostly a software product. The only hardware necessary is the machine the virtual environment hosting the CLASH site and backend is running on.

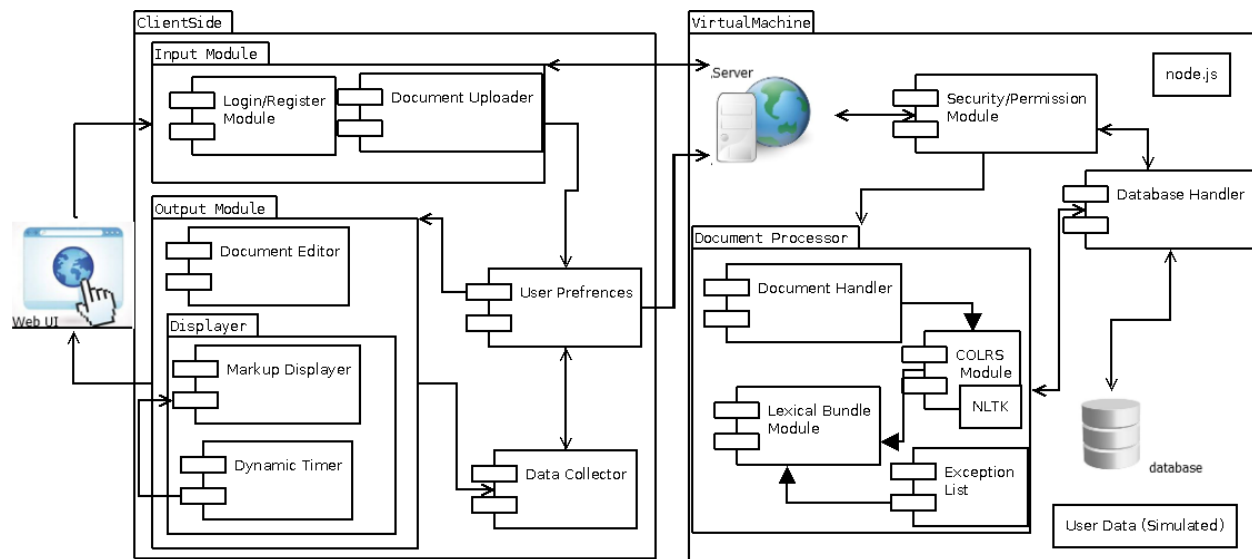
The COLRS module will make use of a natural language processor via python to parse text. This text will then be inserted into a .json file. The .json file will then be parsed by attribute, one of which is the part of speech. A javascript will run against the .json file and color each word with the same part of speech attribute tag and color it a matching color.

The SLASH module will take the parsed COLR .json file, and using the same attributes, will parse the text into lexical bundles. It will do this by following a few basic rules. First, the module will compare the document looking for “exceptions” set by the professor. This will ensure those lexical bundles on the “exception” list will not be split up. Then, it starts at the beginning of the document and adds a slash after each period, coma, semicolon, colon, or question mark. Next, the module will add a slash before each proposition and each conjunction.

On the client side, there will be a website which receives text from the user, and then parses it according to what the user requests. In reality, the entire thing is parsed completely, but what is shown to the user varies depending on what the user selects. The user will select from the options: COLR, SLASH, SLASH PLAYBACK. This will show them colored text delineating parts of speech, the text parsed into lexical bundles, or a playback module. The playback module will be similar to a speed reader.

.

(This space intentionally left blank.)

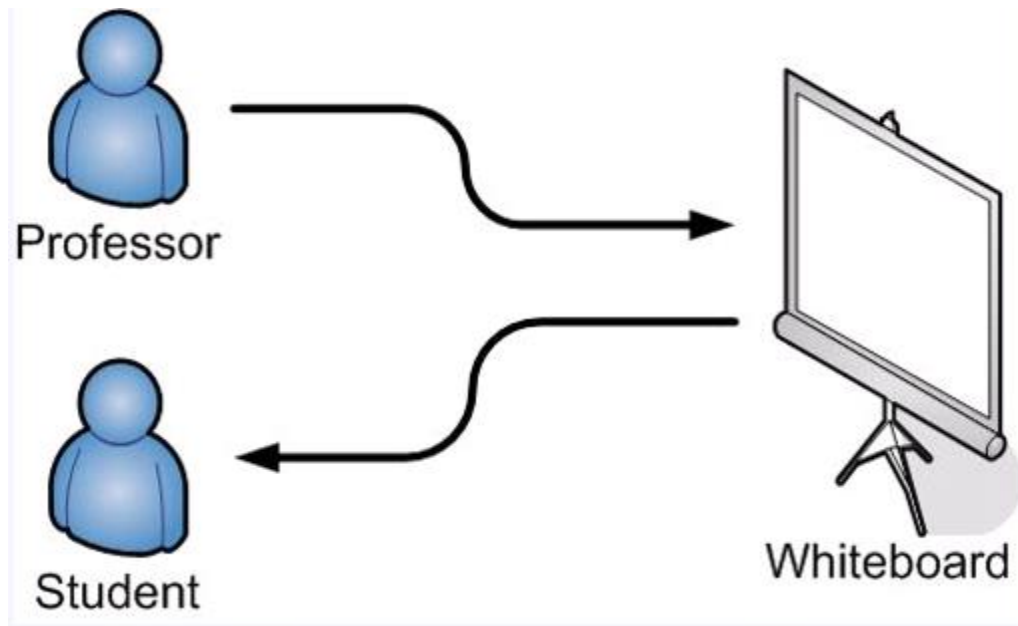


*Figure 1. Major Functional Component Diagram*

### 3 IDENTIFICATION OF CASE STUDY

Professor Greg Raver-Lampman; an instructor at the Old Dominion University English Language Center hypothesized that coloring parts of speech and defining and displaying lexical bundles would increase the reading speed and comprehension of ESL students. Therefore, he tasked the CLASH team with making a product that would help prove out his hypothesis. While this is simply a hypothesis, it is an informed one. Currently, he colors parts of speech and displays lexical bundles manually.

(This space intentionally left blank.)



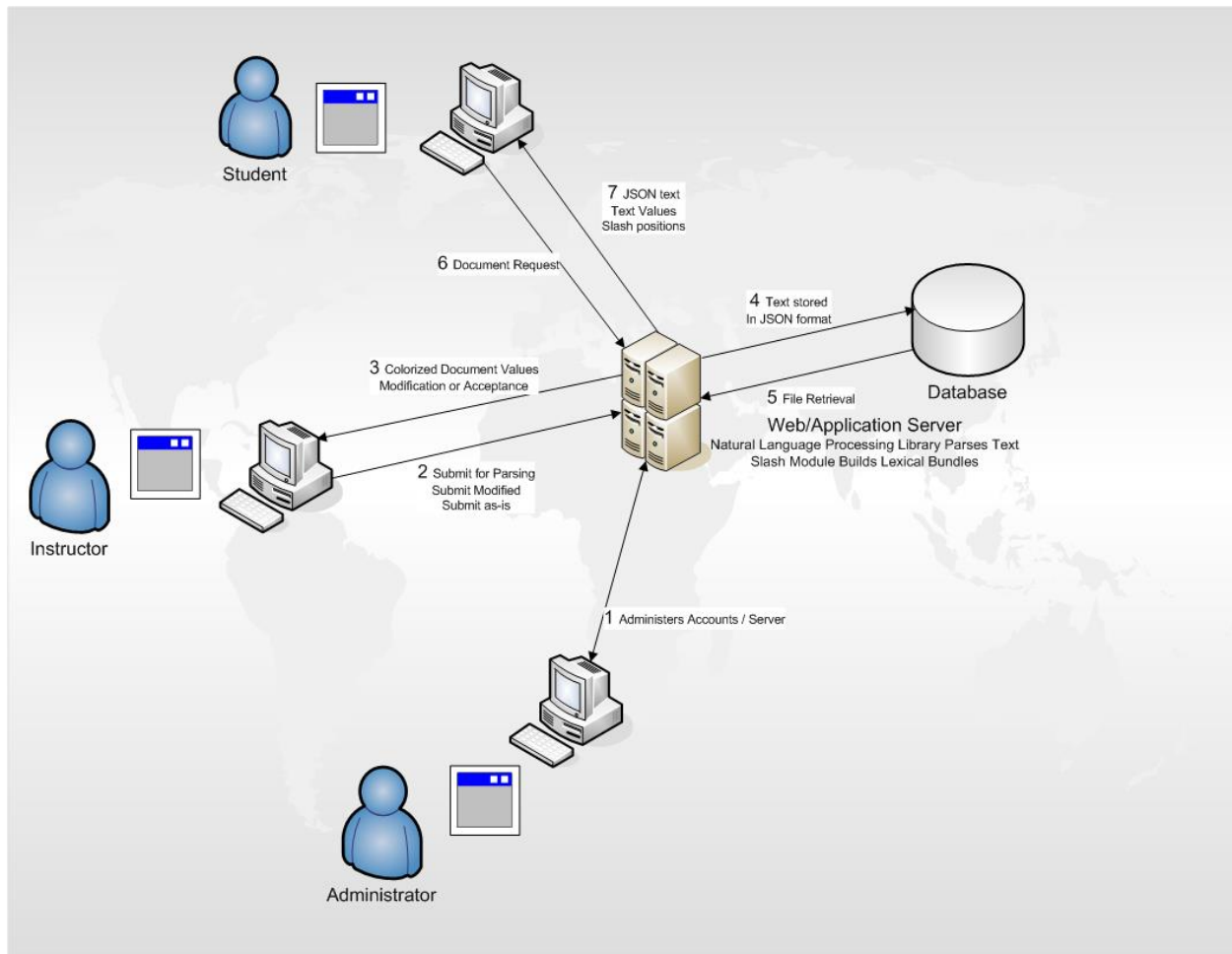
*Figure 2. Current Process*

This process, while effective is tedious, time consuming and prone to error.

The CLASH web application will expedite this process. It will also allow the instructor to save his documents and reuse them from year to year. Eventually, reporting and tracking algorithms could be added to allow the instructor to track his student's progress and increases in reading speed. It will also better allow students to study on their own time via the website.

(This space intentionally left blank.)





*Figure 3. Future Process*

#### 4 CLASH PROTOTYPE DESCRIPTION

The CLASH prototype will be built as a Single Page Application. This form of web application first on a single page and does not reload due to state changes. It will rely heavily on JavaScript as the primary programming language, as will most of the modules interacting with it. We will be using Node.js, .json file formats, and a JavaScript interface. There will also be a python shell that interacts with the natural language processor and the javascript. Where CLASH will differ from traditional SPAs is it will use a relational database, as opposed to a NoSQL database.

There are many benefits to using a SPA. It is easy for the end user to access and use, no matter how limited their technical background may be. It is also beneficial because there is no software for the user to install. As there is no software to install, it also means the user can access CLASH anywhere he has an internet connection. Finally, being a single page means that the user won't get 'lost' in a path of web pages. All functionality is accessible through the menus on the main page.

The prototype will share many features with the real world product. However, the prototype is merely a proof of concept. As such, some functionality may be simulated, faked, or non-existent. For example, the COLRS module will not have homework modes for the student, the POS tagging will rely on a pre-built library as opposed to learning from the instructor editing the mistaken tagging. The prototype will also only be built for Old Dominion University. It will not provide functionality to expand the user base beyond campus.

| <b>Features</b>                                    | <b>Real World Product</b>  | <b>Prototype</b>  |
|--|--|---|
| Parsing Capabilities                               | Ability to Parse different kinds of documents  | Ability to parse text copy and pasted in text block                                   |
| Text Modification                                  | Ability to modify and store previously parsed documents  | Ability to modify and store previously parsed documents                               |
| Color Capabilities                                 | Ability to color chosen parts of speech using a JSON format and JavaScript functions.                  | Ability to color chosen parts of speech using a JSON format and JavaScript functions. |
| Slashing Capabilities                              | Ability to identify Lexical Bundles through the inserting of slashes.                                  | Ability to identify Lexical Bundles through the inserting of slashes.                 |
| Displaying Lexical Bundles in a single bundle form | Ability to speed up, slow down and pause Lexical Bundles being displayed.                              | Ability to speed up, slow down and pause Lexical Bundles being displayed.             |
| Exception list                                     | Lists of commonly used expressions that would otherwise be incorrectly handled by the SLASH Algorithm. | Lists of commonly used expressions that would otherwise be incorrectly                |

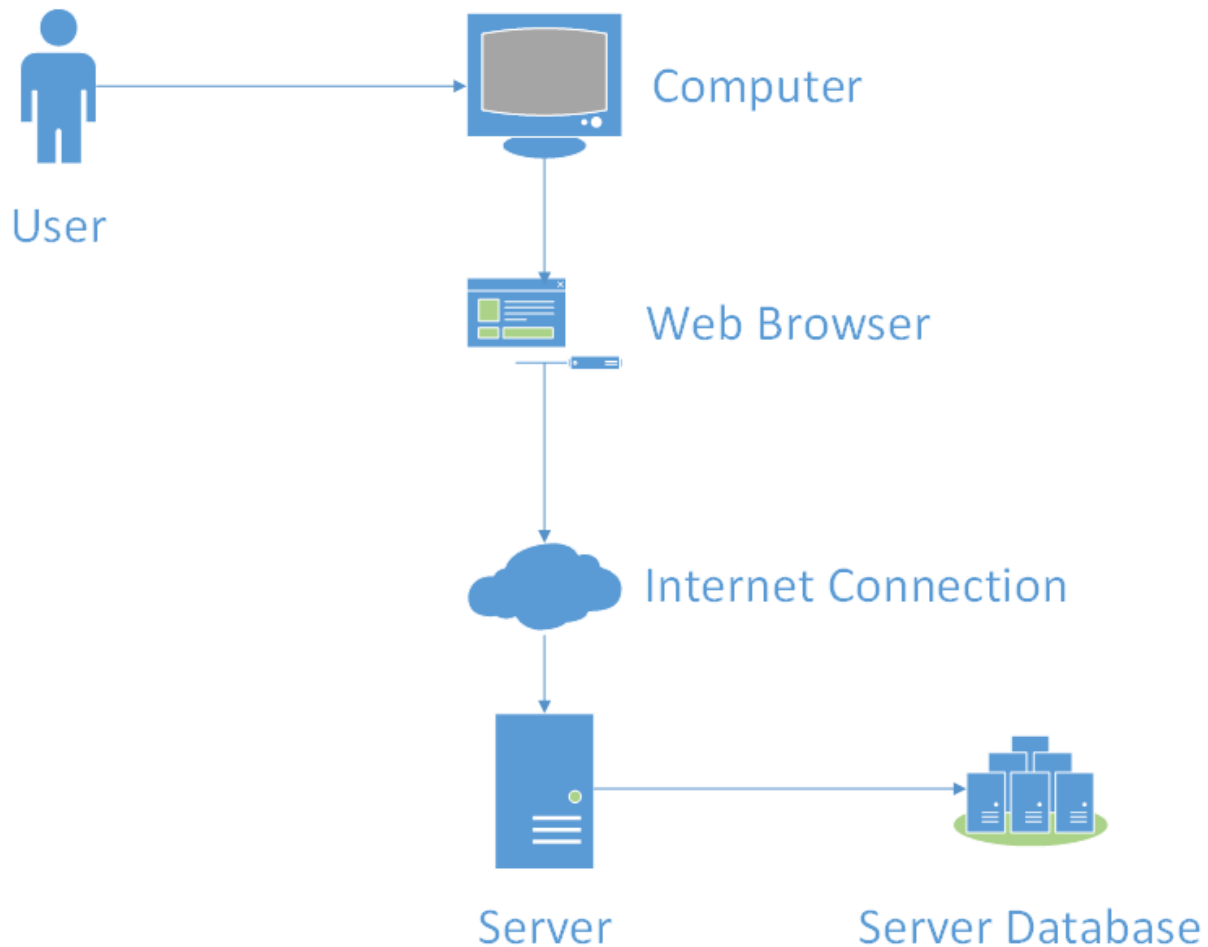
|                             |  |  |
|-----------------------------|--|--|
|                             |  | handled by the SLASH Algorithm.  |
| Login interface             | User Authentication in a stand-alone environment   | User Authentication in a stand-alone environment   |
| Student Data Reporting      | Tracks individual and collective student progress. To include words per minute, total time and total Lexical Bundles. Data to be stored in database. Displayed in graphs and statistics. | Limited basic student metrics will be available such as Lexical Bundles per Minute.      |
| Homework Mode               | Instructors have the ability to remove coloring of words and have students correctly identify the part of speech.  | Not Included.  |
| Administrative Privileges   | Administrators are able to edit, add, or remove users and saved documents in the system.   | Administrators are able to edit, add, or remove users and saved documents in the system. |
| SLASH Document Viewing Mode | Ability to view documents with slashes inserted and SLASH Reader.  | Ability to view documents with slashes inserted and SLASH Reader.                        |

*Table 1. Real World vs. Prototype*

#### 4.1 Hardware and Software Prototype Architecture

From a hardware perspective, as mentioned earlier, there will only be a machine which will run a virtual machine to hold the server, database, and website files. The software used will be a collection of open source, free to use, and custom written. The virtual machine will run Ubuntu 14.04 LTS, the web and application servers will be Node.js. The Node.js server will interact with the MySQL database and the Natural Language Toolkit.

(This space intentionally left blank.)



*Figure 4. Prototype Hardware/Software Architecture*

#### **4.2 Prototype Features and Capabilities**

The CLASH prototype will demonstrate the ability to identify specific parts of speech by coloring the text also identify lexical bundles using slashes. The prototype will also allow users to display lexical bundles in a ‘speed reader’ type fashion. In addition the prototype will allow users to display bundles one at a time, increasing, decreasing or pausing the lexical bundle stream. Success of the prototype will be determined by the accuracy of the tagged parts of speech and lexical bundles.

Determined risks are to be mitigated through agile development, the involvement of the mentor throughout development of the product, and early and constant beta testing by potential users. Upon completion of the working prototype containing core functionality will result in the customer being able to test the product in a real world situation against a control group, further assisting the development of the product. By completing the proof of concept the program prototype we will be able to demonstrate the feasibility of such a system and together with the prototype testing group, the utility of using the product in an academic environment thus giving the customer the desired outcomes.

#### **4.3 Prototype Development Challenges**

There are many challenges being faced during the development of CLASH. The main challenges are scope creep, project management, and learning new software. Professor Greg Raver-Lampman has very high aspirations and great ideas for what CLASH could and should do. As such, we have a very definite problem of managing scope creep. Beyond scope creep, the team is comprised of many people. Attempting to manage a large group with little to know authority is almost impossible. Keeping everyone busy and on task is a major undertaking. The team also faces the challenge of learning new programming languages and concepts such as .json and SPA. The program itself faces all the challenges inherent with anything concerning the English language. It is a very complex language, and we are merely scratching the surface with CLASH.

**GLOSSARY**

**CLASH** - Color Lexical Analysis algorithm and Slash Handler

**COLRS** – Colored Organized Lexical Recognition Software

**ELC** – English Learning Center

**ESL** – English as second language

**IBT** – International benchmark test

**JSON** – JavaScript Object Notation

**Lexical Bundle** – a group of words that occur repeatedly together within the same register

**MFCD** – Major Functional Component Diagram

**NLTK** – a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language.

**Node.js** – an open source, cross-platform runtime environment for server-side and networking applications.

**POS** – Parts of Speech

**SPA** – single page application, is a highly responsive web application that fits on a single page and does not reload as the web page changes states.

**TOEFL** – Test of English as a Foreign Language

**Ubuntu** – a Debian-based Linux operating system.

**VM** – Virtual Machine

## **REFERENCES**

McKeon, D. (n.d.). Research Talking Points on English Language Learners. Retrieved December 11, 2014.

Tremblay, A., Derwing, B., Libben, G., & Westbury, C. (2011, January 15). Processing Advantages of Lexical Bundles: Evidence From Self-Paced Reading and Sentence Recall Tasks. Retrieved December 10, 2014.

Mikowski, M., & Powell, J. Single Page Applications. Manning Publications 2014.