CLASH Product Description CS410 Lab 1

Blue/Purple Teams

Ming Jiang

CS 411W

Professor Price and Professor Brunelle

March 4, 2015

Version 2

Contents

[**1.** **INTRODUCTION** 4](#_Toc414277112)

[2.2 Major Hardware and Software Components 6](#_Toc414277113)

[**3 IDENTIFICATION OF CASE STUDY** 7](#_Toc414277114)

[**4 CLASH Product PROTOTYPE DESCRIPTION** 11](#_Toc414277115)

[4.1 Hardware and Software Prototype Architecture 13](#_Toc414277116)

[4.2 Prototype Features and Capabilities 14](#_Toc414277117)

[4.3 Prototype Development Challenges 14](#_Toc414277118)

[**Glossary** 15](#_Toc414277119)

[**References** 16](#_Toc414277120)

**List of Figures**

*Figure 1.* CLASH Teaching Process ……………………………………………………………5

*Figure 3.* Major Function Component Diagram ………………………………………………....6

*Figure 3.* Prototype Major Function Component Diagram ……………………………………..12

**List of Tables**

*Table 1.* Real world VS Prototype Features …………………………………………………….10

# INTRODUCTION

According to the Institute of International Education, there were almost 900,000

International students who attended a United States college or university during the 2013-2014 academic year.1 While some of the International students come to the colleges with fluent language skills, many of them have limited proficiency in English. For those English Language Learners, or ELL, they need to pass the English as Second Language (ESL) program before they can sign up for other classes. The main problem with the International students is the time restriction, so they can only spend limited time in the ESL program. Due to the time limitation and different levels of reading proficiency among ESL students, it becomes even harder for ESL teachers to prepare the teaching materials for those students. If a software can be designed to streamline the teaching processes in ESL program, it will improve the success rate of those International students.

Color Lexical Analysis algorithm and Slash Handler, or CLASH, is a senior project that is designed to help the ESL teachers to improve their teaching efficiency. It will also provide parts of speech tagging and speed reader functionalities to ESL students that aims to improve their reading speed and reading comprehension.

*(This Space Intentionally Left Blank)*

1 Institute of International Education. (2014). "International Student Enrollment Trends, 1948/49-2013/14." Open Doors Report on International Educational Exchange. Retrieved from http://www.iie.org/opendoors

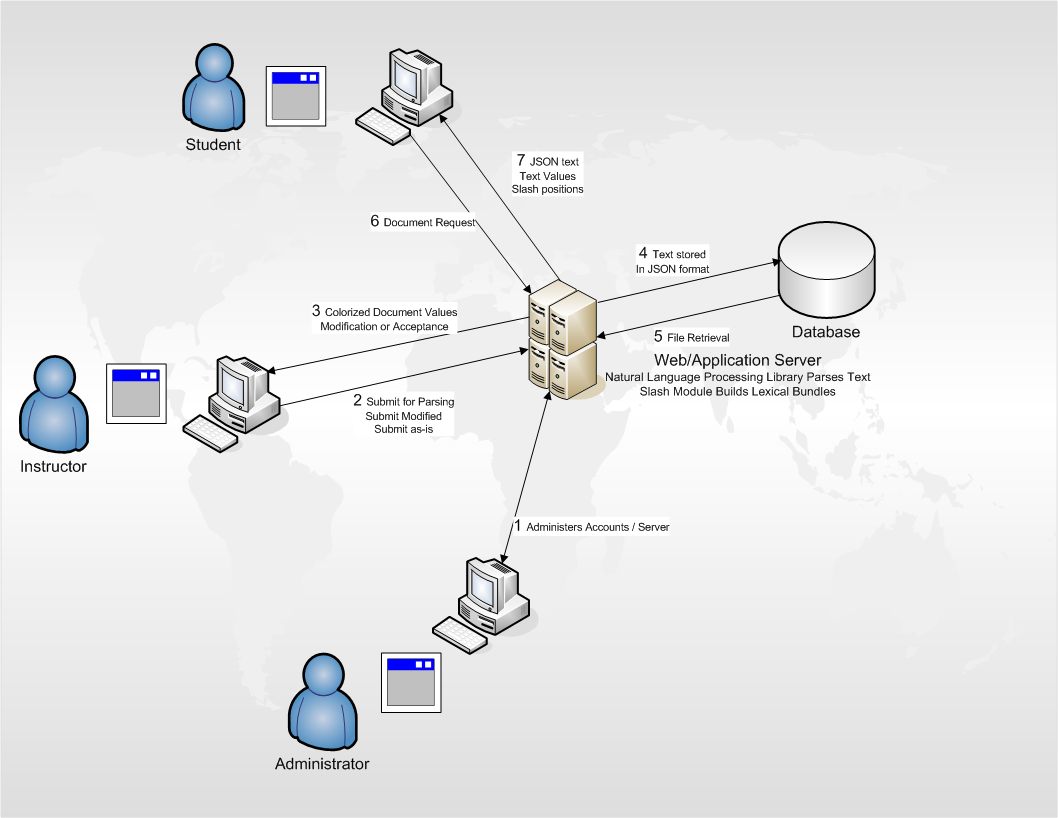
# CLASH PRODUCTION DESCRIPTION

CLASH is software designed to help improve the ESL teachers’ teaching efficiency and ESL students’ reading performance. CLASH is a web-based software that can be accessed from any computer with an Internet connection. It will be a virtual classroom for instructors to provide reading materials to students. CLASH will also present a Graphic User Interface for users to interact with CLASH server.

The product will reduce the time instructors need to manually mark each documents, and it will allow students to access the reading materials from home. The product will provide parts-of-speech-tagging that displays different grammatical forms in different colors. It will also provide lexical bundles reading experience for users to improve their reading performance; and help to turn readers from word by word reader into lexical bundle reader. The product will also provide report functionality for instructors to track students’ reading speed and overall reading performances, which can help identify students’ participation and improvement throughout the course period.

The instructor will be able to upload a document to server for processing, and the result file will be sent back to the instructor for inspection. The instructor can modify the error grammar tagger and slash position before saving file on the server. After the instructor edits the file, students will be able to download and read the document from CLASH user interface.

*(This Space Intentionally Left Blank)*



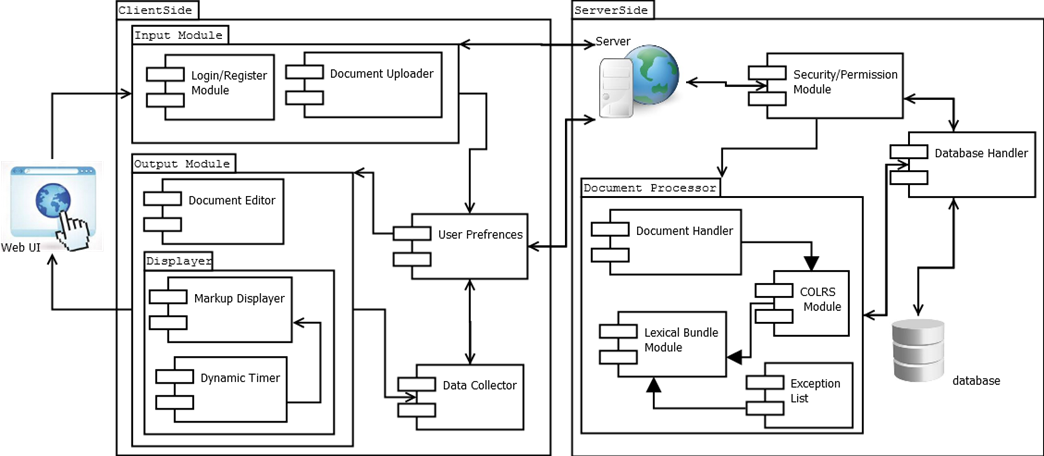
*Figure 1.* CLASH Teaching Process

*(This Space Intentionally Left Blank)*

## Key Product Features and Capabilities

CLASH will be a web-based application that consist of multiply modules. It aims to improve teaching efficiency and a better learning experience for ESL students. The key features of this product are the ability to analyze the upload document and generate part of speech taggers for each word, and the ability to break sentences into lexical bundles.

## Major Hardware and Software Components



*Figure 2.* Major Functional Component Diagram

**Overview.** The CLASH will be a web-based application that hosts on server, and it can be easily accessed through any major web browsers. There is no special hardware requirement other than an active server, a database on the server end, and an Internet-enabled device for the client to access the application. There are three major software components for this product, COLRS Module, Slash Module, and Client-side Reader.

**COLRS module** use open source Natural Language Processing (NLP) library to parse and tokenize the input document, and it will generalize the parts of speech into nine grammar categories. The output will be a tokenized stream with token and tagger pairs.

**Slash module** will accept tokenized stream from the COLRS module and insert slashes to break sentences into lexical bundles base on the preset rules. The instructors can provide a list of common lexical bundles, or exception list, to avoid any slash within those phrases, thus improving the accuracy of the final output.

**Client-side reader** will receive data from the server and parse each individual token according to its part of speech tagger. There will be two display modes, color mode and speed mode. In color mode, the user can choose which of the nine grammar categories to be display in color, and user can choose to show the slash in between lexical bundles and highlight the exception list provided by the instructor. Under speed mode, the reader will display a single lexical bundle at a time and change the lexical bundle according to preset reading speed.

*(This Space Intentionally Left Blank)*

# IDENTIFICATION OF CASE STUDY

Professor Greg Raver-Lampman, an instructor in English Language Center at Old Dominion University, proposed this project aim to improve the teaching efficiency of the ESL program. With different levels of reading proficiency in a single classroom, it became harder for instructors to prepare reading material and reading time slots in class. So there is a need for a software that can automate the process of prepare the document, such as identifying part of speech taggers and breaking sentence into lexical bundles. He also wants a tool that can display document in lexical bundle and force user to become faster reader. Both functionalities will be included in the CLASH program to help improve ESL students’ reading performance.

*(This Space Intentionally Left Blank)*

# CLASH Product PROTOTYPE DESCRIPTION

The goal of this project is to create an interactive website that can help instructor improve the teaching efficiency and also increase the students’ reading performance. The CLASH prototype will be built as a Single Page Application, or SPA. It will be a highly responsive single webpage that does not reload as webpage content change, and it will be built completely by JavaScript, which is supported by all major browsers. The traditional SPA program includes a JavaScript user interface, a JavaScript web server, and a NoSQL database such like MongoDB. The CLASH application will be slightly different than traditional SPA, in which we will be using traditional relational database, MySQL, instead of the NoSQL database. The CLASH will still be built with a pure JavaScript user interface and a JavaScript web server to increase overall performance.

SPA can offer a desktop application look and feel, and users can access the tool without any complex installation. The JavaScript server is also known for its highly responsive feature, it can reduce the waiting time even with high volume of Internet traffic. Also, the entire application will be built as SPA, so it will be easy for users to see all the functionalities.

The prototype will be built with most of the functionalities as the real world product. However, due to the time limitation and university restriction, there will be few differences in the CLASH prototype, as listed in Table 1.

The first difference will be the limited reporting functionality. Since all user accounts and user data will be pre-populated fake data, there will be no need for complex reporting function in the prototype. With the time limitation, it became unpractical to include support for various file formats. So the second difference is the prototype will not support input format other than copy and paste.

|  |  |  |
| --- | --- | --- |
| **Features** | **Real World Product** | **Prototype** |
| 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 Features*

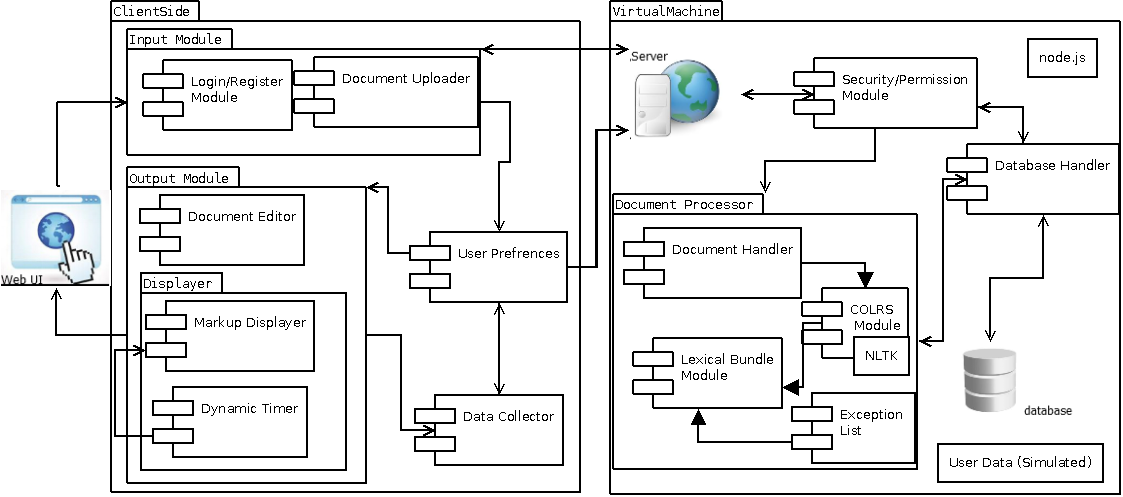
The CLASH prototype will not be able to access any student enrollment data from Old Dominion, so instead, it will be simulated the students and instructors data. The instructor will be allowed to add students from his/her class into the prototype as test cases, and they will be allowed to access most of the functionalities CLASH provided.

## Hardware and Software Prototype Architecture

The CLASH prototype will build the application on a Virtual Machine that simulates a server. On the user’s side, the only hardware need will be an Internet capable computer.

There will be some software requirement for the server, but all of the software will be open source project or liberally licensed software. Our server will be running Ubuntu 14.04 LTS operating system, and the main http server will be the Nginx and the Nodejs, which will handle incoming network requests from students and instructors. The COLRS module will be written in Python with the open source NLTK language processing toolkit, and Slash module will be a pure Java application that breaks documents into lexical bundles according to the input from COLRS. It will be using the MySQL database on the server to manage user authentications and document storage, the database will also include miscellaneous data like tracking data and session information.

*(This Space Intentionally Left Blank)*



*Figure 3. Prototype Major Function Component Diagram*

## Prototype Features and Capabilities

The CLASH prototype application will have the ability to identify parts of speech and break the document into phrase groups. Research has found that reading speed and comprehension increase as students, including native English speakers, break the habit of word-by-word reading and learn to read text in phrase groups and common clusters of words known as lexical bundles.1 In addition, the prototype will provide a speed reader to display one phrase group at a time. The prototype will provide functionality to move forward or backward lexical through the phrases and adjust the reading speed. Also, users will be able to send feedback about the accuracy of colorization and lexical bundles.

A working prototype will allow the instructor to test the product. There will be some student test group from the English Language Center at Old Dominion University who will use this product for a few weeks, and they could compare their reading performance before and after using the CLASH. The result can serve as a proof of concept of the prototype.

1 Raver-Lampman, Greg. (2014. August). Personal Interview.

There will be some risk associated with the Slash and COLRS module. Due to the complexity of Natural Language Processing, it is possible for those two modules to provide incorrect result. In order to migrate this risk, CLASH will provide an instructor edit mode to help correct those errors.

## Prototype Development Challenges

There are several challenges during CLASH development. The first challenge was the language complexity, the CLASH is running JavaScript as backbone on the server, but it has to use NLTK, which is a Python library, to identify the parts of speech; also, it requires Java to actually break document into phrase groups, and putting multiple programming languages together and trying to integrate them will be harder than using a single programming language overall. The second challenge will be the deadline; the project needs to be built and workable before most of other team, so there will be time for Professor Raver-Lampman to actually test the tool and see how lexical bundle reading helps students’ reading performance.

*(This Space Intentionally Left Blank)*

# Glossary

**CLASH:** Color Lexical Analysis algorithm and Slash Handler.  
  
**Client Side:** The user-interface of CLASH.  
  
**COLRS:** Colored Organized Lexical Recognition Software.  
  
**Document Processor:** A Server Side component responsible for processing the text entered by an Instructor user type.   
  
**ELC:** English Learning Center at Old Dominion University.  
  
**ESL:** English as second language.  
  
**ELL:** English Language Learner.

**GUI:** Graphic User Interface

**HTML:** HyperText Markup Language

**IBT:** International Benchmark Test

**Intensive English Program:** A short and intensive English language training program offered by US colleges and universities to improve the English language skills of international students who did meet the minimum TOEFL scores for typical enrollment.

**JS:** JavaScript

**JSON:** JavaScript Object Notation. A nested data structure commonly used to pass data between a server and a client.  
  
**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).  
  
**Node.js:** Open source, cross-platform run-time environment for server-side and networking   
applications.  
  
**POS:** Part-of-Speech such as noun, adjective, verb, etc….  
  
**Server Side:** The back-end of the CLASH system responsible text processing, the database, user-authentication, and web-hosting.   
  
**SPA:** Single page application. A highly responsive web application that fits on a single page  
and does not reload as the web page changes states.

**Spreeder:** Speed reading tool; [www.spreeder.com](http://www.spreeder.com/)

**Software as a Service (SaaS):** Software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet.

**TOEFL:** English language proficiency test required by universities for enrollment for internationally based students.  
  
**Token:** Text that has been processed into individual words by the Document Processor.

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

**VM**: Virtual Machine.

# References

"College Enrollment by Sex, Age, Race, and Hispanic Origin: 1980 to 2009." U.S. Census Bureau, 1 Jan. 2012. Web. 2 Feb. 2015. <http://www.census.gov/compendia/statab/2012/tables/12s0282.pdf>.

McKeon, Denise. "Research Talking Points on English Language Learners." National Education Association, 1 June 2005. Web. 2 Feb. 2015. <http://www.nea.org/home/13598.htm>

Choudaha, R., and L. Chang. "Trends in International Student Mobility." World Education Services, 1 Feb. 2012. <http://www.wes.org/ras/TrendsInInternationalStudentMobility.pdf>

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.

Educational Testing Service. (2015). About the TOEFL iBT® Test. Retrieved February 7, 2015, from ETS.org: <<http://www.ets.org/toefl/ibt/about?WT.ac=toeflhome_ibtabout2_121127>>

Hanel, S. D. (2011, March 5). Lexical Bundles. Retrieved February 7, 2015, from Communicating in English - An Internet-based, English Language Resource : <<http://sdhanel.com/corpuslinguitics/lexicalbundles.html>>

Haynie, D. (2014, Novemeber 14). Number of International College Students Continues to Climb. Retrieved February 6, 2015, from US News: <http://www.usnews.com/education/best-colleges/articles/2014/11/17/number-of-international-college-students-continues-to-climb>

Institute for International Education. (2014). Intensive English Programs: Leading Places of Origin. Retrieved February 7, 2015, from Institute for International Education: <http://www.iie.org/Research-and-Publications/Open-Doors/Data/Intensive-English-Programs/Leading-Places-of-Origin/2012-13>

Institute of International Education. (2014). "International Student Enrollment Trends, 1948/49-2013/14." Open Doors Report on International Educational Exchange. Retrieved from <http://www.iie.org/opendoors>

Institute of International Education. (2014). Open Doors Fact Sheet: Virginia. IIE with support from the U.S. Department of State's Bureau of Educational and Cultural Affairs.

Open Doors. (2014). Open Doors 2014: A 15-Year Snapshot. Washington DC: Institute of International Education.

Powell, M. &. (2014). Single Page Web Applications. Shelter Island: Manning Publications Co.

Raver-Lampman, Greg. (2014. August). Personal Interview.