|  |
| --- |
| THE GEoRGE WASHINGTON UNIVERSITY  miniteMinerLogo2-1024.png        Minute Miner |
| Capstone Project & Senior Thesis II Spring 2014 |
| Seung Kwon An  Shichiji Ishiyama |
| 4/27/2014 |
|  |
|  |

# Abstract

In almost any industry, the use of data has increased as the technology to aid data collection has been developed. The term “big data” can be seen as a major role in a business such as the IT world and the Healthcare world. Data in business refers to any material or knowledge in a particular field that will benefit the business with improved efficiency or an increase in revenue. With so much data to process and analyze, the final product of researched data will prove to be a very valuable resource for a variety of tasks in a business. The goal of this project is to create and combine open source software projects that intend to collect or “mine” the data, analyze the data through computer coded algorithms, and create an export of the data as a readable report that will contain useful information for businesses to utilize. The project will use Java as the fundamental backbone of the software and will integrate other successful software to create a very user friendly data analyzer.

**Table of Contents**

[Abstract 1](#_Toc383966881)

[Brief Overview 3](#_Toc383966882)

[Title: 3](#_Toc383966883)

[Concept: 3](#_Toc383966884)

[Objectives and Goals 3](#_Toc383966885)

[Target Users 4](#_Toc383966886)

[Review of Related Literature 4](#_Toc383966887)

[Methodology 6](#_Toc383966888)

[Conclusion 6](#_Toc383966889)

[Design Process 7](#_Toc383966890)

[Algorithm 7](#_Toc383966891)

[Activity Diagram 10](#_Toc383966892)

[Context Diagram 11](#_Toc383966893)

[Level 0 data flow diagram 12](#_Toc383966894)

[Decomposition Diagram of Process 1 13](#_Toc383966895)

[Decomposition Diagram of Process 2 14](#_Toc383966896)

[Decomposition Diagram of Process 3 15](#_Toc383966897)

[User Manual 17](#_Toc383966898)

[Resources 20](#_Toc383966899)

[Appendix 21](#_Toc383966902)

# Brief Overview

## Title:

　 Minute Miner

## Concept:

　　 Our project is to develop a user-friendly data science application. Most data mining and science tools are difficult to use and highly focused on the professional use. Today, the amount of data is exponentially increasing every year but the increase in the number of people who can explore and discover something out of big data are not adequately proportionate to the growth speed of data amount. We considered this problem and came up with the idea of making a user-friendly and easy to use but certainly functional data mining and science tool. Minutes Miner’s core concepts are stated as:

* Minute Miner is an application that enables users to perform simple data mining
* Process large amounts of text-based data on website
* Different perspective data analysis
* Increase individual users productivity rate regarding the use of data

## Objectives and Goals

The objectives and goals of our project are:

* Ease of use
* To make graphically and methodologically user-friendly interface and process for data science
* To change the preconception among people that data science is way beyond average users’ mean
* To design adequate algorithm of this application development and make it into practice
* To enable non-professional users to explore data in depth

## Target Users

People who are interested in data mining and science but do not know what to do or how to use the tools for them.

# Review of Related Literature

The software selected for review is a data mining software called WEKA. The Waikato Environment for Knowledge Analysis (WEKA) was first drafted in 1992 by the University of Waikato located in New Zealand1. This software primarily focuses on analyzing data sets using different algorithms pre-programmed into WEKA3. The algorithms are compiled to be available in WEKA at any time and the ability to add or use the algorithms without restrictions is one of its key features3. Another reason for selecting this software to review is the fact that WEKA was written in Java. From algorithms to the interface of the software all were developed in Java.

WEKA is an open source project that allows for other Java code to be integrated into the software or reversely, implement the algorithms in WEKA to another project or software that is compatible with Java6. WEKA contains unique features that attempt to allow the users to have a better understanding of what data analysis process is going on. WEKA consists of a section called the “workbench”, which contains numerous algorithms for users to try and study the results all in one area2. This feature proves to be very useful for scientists or data analysts that are attempting to create multiple results.

The workbench feature is one of the interesting options that WEKA has for experimenting with data. Another feature is called “experiment mode”, which allows the user to modify a series of conditions that might bring very different results when compared to the traditional data analysis methods4. This is a useful feature if a particular project requires varying algorithms or to simply test how new modified algorithms might be beneficial. From altering a single variable to implementing different conditions to an existing algorithm, the experiment feature is a great tool to have when discovering new alternate ways of analyzing data, even if it seems to be an unconventional way.

In general, WEKA is used primarily for data analysis that involves large amounts of numerical data5. It is widely used by scientists and data researchers. Our project will be focused on a smaller scale of numbers and data to be analyzed. Our goal through this project is to analyze business data, which might be financial reports, sales invoice, monthly inventory, etc. Also, our project will be catered to small local businesses, which generally do not have extensive amounts of data. We chose to take this route on the project, in order to create a reachable goal. Without extensive knowledge of the data analysis field, it will be impossible to create complex algorithms at this time.

Our hope through the project is that more business related data mining and analysis algorithms will be available for people to use as an open source. We will also add the option of making our algorithms easily accessible for use with the WEKA software to show our support of the open source mission WEKA is striving towards. When it comes to business data analysis there are software and companies that already offer the service, but at a very high cost. We want our project to be free of charge and create software that everyone will be able to understand and use. Currently, even for WEKA, the interface is not very straightforward, and it is difficult for a regular user to make use of the program even if he/she desires to do so. The goal of the capstone project is to create a small business data analyzer, but at the same time create software anyone can use at any experience level.

# Methodology

There were two major methods that were carried out to complete this project. For the background, we researched texts, books, manuals, and any information related to the topic of data mining/analysis to gain an in depth knowledge of the subject. Due to the fact that data analytics is a large field pertaining to a variety of industries, the focus was shifted to business analysis in general. The second portion of the methodology consisted of actually running a variety of software that was similar to the objectives in the capstone project. Because the project focused on developing open source software, we researched and downloaded open source projects that carried out data analysis. The process of learning how to operate the individual software proved to be a challenge due to each of the software having its own methodology of carrying out the data analysis.

Most of the research regarding texts was carried out through the George Washington University’s library. Through research we were able to identify scholarly journals, business related articles, eBooks on the data mining software themselves, and texts on methodology of data analysis methodology. Through text material we were able to gain basic fundamental knowledge of how the data mining industry operated, along with basic concepts of how algorithms are constructed.

The second half of the research as mentioned before consisted of a hands on approach to learning the actual software themselves. This gave the much needed insight to how simple applications accepted data files and what steps were necessary to allow the software to attempt to analyze the data. Although there were variances from one software to another, the basic requirements for analyzing data stayed similar in terms of concepts.

# Conclusion

We understand the difficulties in achieving success in our application development since we haven’t had enough software development skills and knowledge. Thus, we finally decided to development the application in different method from what we proclaimed at the beginning of the semester. Our current plan focuses on the better use of data mining application and enabling the users to explore further the information of the data that are produced via Weka by highly sophisticated calculations and methodology. In other words, our task is to design user-friendly GUIs and some simple processes that use the data-mined data from Weka and keep the data mining function. We have already started designing the GUI and this process should not be the difficult part. Some difficult tasks are calling functions and class files from Weka to our original GUI. Also, data transfer and variable conversions within every section requiring converting variables is also difficult part.   
 In conclusion, it could be said that our application development plan is improved since the virtual presentation. We refined our objectives, goals, application size, and algorithm; we made a project size much smaller than the first draft idea. We are quite excited in completing our project as we proposed in this rough draft.

# Design Process

In this section, the design process of Minute Miner is divided in several diagrams consisting of an activity diagram and data flow diagrams. Also, the core algorithms of Minute Miner are described in the beginning of the section.

## Algorithm

Algorithms used in the program processes are:

1. Extract HTML data (filtered and unfiltered HTMLdata)
2. Store the extracted HTML text data into txt.file and two dimensional arrays
3. Visualize unstructured text data stored in arrays
4. Make a separate array for HTML header & metadata information
5. Determine criteria (Count & Highlight frequently used terms)
6. Outlier Examination
7. Determine keywords/terms from extracted data
8. Provide data matching sources using keywords
9. **Extract HTML data (filtered and unfiltered HTMLdata)**

To extract HTML data, we have written a class file that simply extracts website’s HTML elements inspection page. Our class file extracts both filtered and unfiltered HTML data. The filtered HTML data means text data pulled out from HTML tags such as <head>, </body>, and <p>. Also, the reason why unfiltered HTML data is needed is because our data mining process uses HTML tags as index in data classification. The class file consequently brings data to client’s device in a text format.

1. **Store the extracted HTML text data into txt.file and two dimensional arrays**

Our program stores the extracted data in two data types: .txt file and two dimensional arrays. The purpose of storing the data into .txt file is to make a backup for the extracted data. Arrays are passed and the values constantly change, so there must be an original data that is not required to be altered its data value. On the other hand, the data stored into arrays are largely changeable through the data mining process: data classifying and data clustering.

1. **Visualize unstructured text data stored in arrays**

All the visual representations and statistical calculations are processed based on the text data stored into the arrays. The extracted data should be stored into two dimensional arrays before the visualizing process and visualizations are executed with unstructured but filtered text/string data. The purpose of visualizing unstructured data is to demonstrate how information is originally used in a website. The process definitely shows some data concentrations and the outliers. This data will be quite significant in achieving effective differentiation between structured and unstructured data.

1. **Make a separate array for HTML header & metadata information**

In HTML, header information and metadata are significant information about the schema and substance of websites and therefore, the information can be highly useful in comprehending essence, dimension, and emphasis of the website. To utilize HTML header information and metadata, our program file parses the header and metadata out of unfiltered HTML data previously stored into an array. After this process, header and metadata analysis will be able to be executed.

1. **Determine criteria (count & highlight frequently used terms)**

To analyze big amounts of data, criteria are requireds that can effectively value the existing data inside arrays, thus, determining criteria by using header and metadata is said to be the adequate utilization of the information. This process is part of header and metadata analysis and counting and highlighting frequently used terms are the main tasks done in here. The terms used are sorted and restored into an array in a quantity-based manner (if there are words having the same quantity then the order will be ascending). After the restoration, the pivot table created by the restored array will examine outliers, mean, median, mode, range, variance, and standard deviation, and these components are used as criteria.

1. **Examine outliers**

The outliers examined in the previous step will be stored in the outlier array and outliner examination in depth will be performed. Here, our program file attempts to find correlations between outliers and non-outliers. Some outliers can be important criteria to find some comparable information to the result produced from structure data.

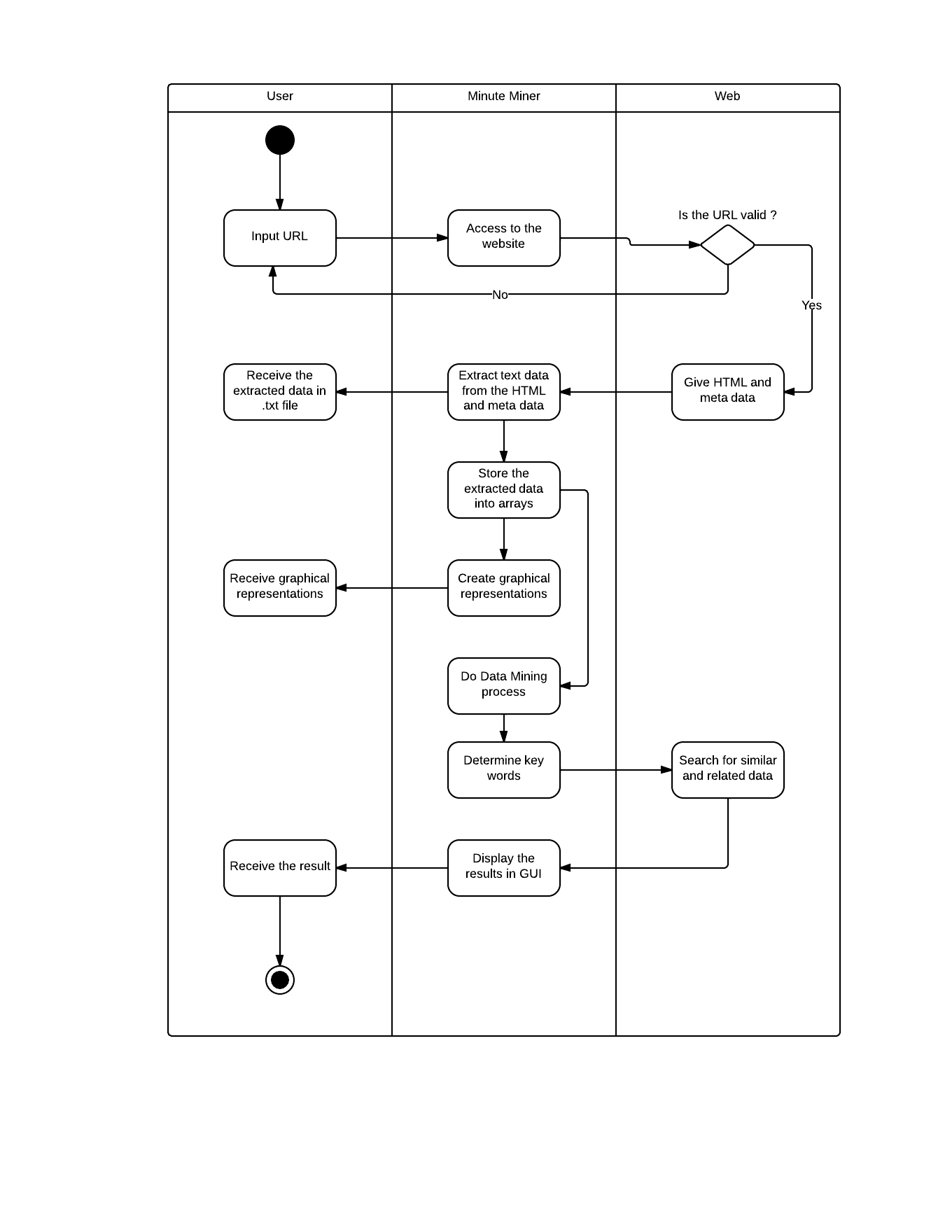
1. **Determine keywords/terms from extracted data**

Determining keywords/terms is the most important step in discovering related, similar, or comparable data for the website that the client have chosen. To determine keywords/terms, Minute Miner program file will make unstructured data to structured data by classifying and clustering data. In the classifying process, all unstructured data will be sorted and stored in a statistical procedure. In the clustering process, groups the classified data based on the data similarity. To consolidate similar data, the criteria key terms that are determined in the previous step will used as index of cluster arrays and the classified data will be stored based on the index. After completing structuring the data, the structured-data arrays are examined in a statistical procedure again, and then produce final processed data as keywords/terms with a correlation table.

1. **Provide data matching sources using keywords**

In this step, the correlation table created in the previous step will be used to produce outputs to the client. The correlation table contains the percentages of the correlation between each keyword. Our program file chooses the correlation with two keywords having the highest correlation percentage and put them into a web search engine. The results are shown in a separate interface. Visual representations with the structured data will be output in this process as well.

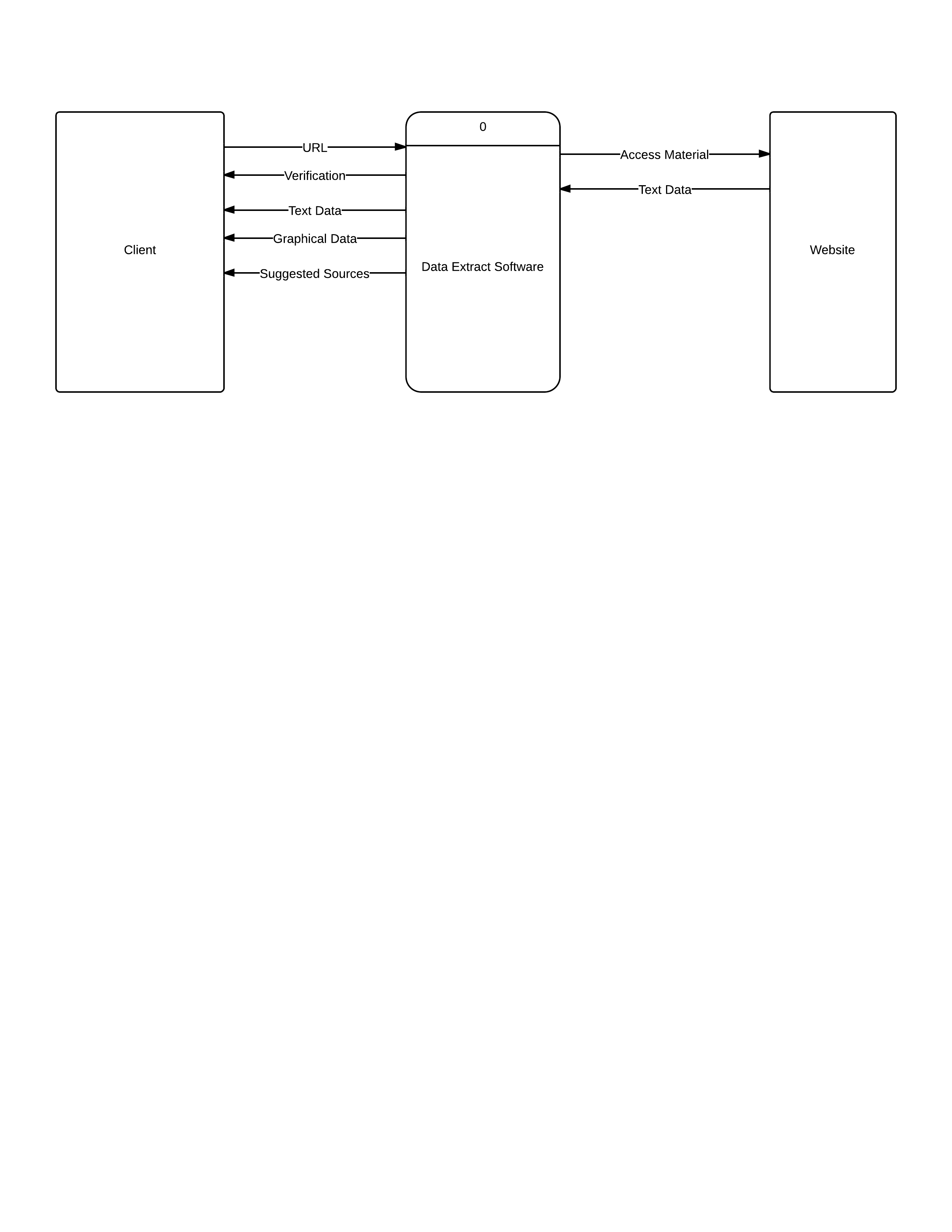
## Activity Diagram

 The basic activities executed by a client, Minute Miner, and Web are described in the activity diagram shown below.

The only thing a client needs to do in order to proceed from the beginning to the end is to input the website’s URL that the client demands to explore. The websites that require authentications such as passwords and user ID cannot be accessed by Minute Miner.

## Context Diagram

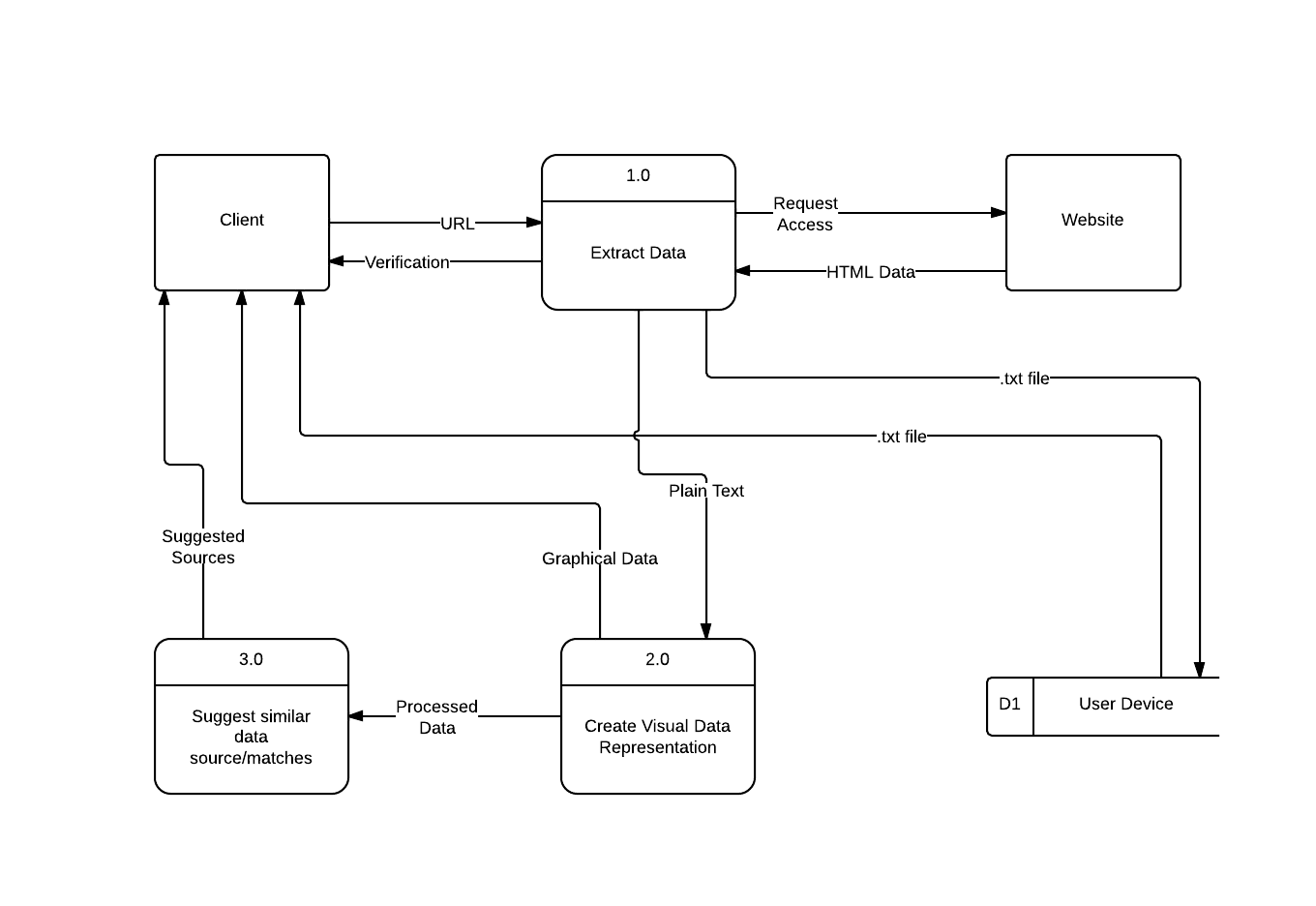
Here is the context diagram for the Minute Miner system with two stake holders: clients and website.



The data flow of the Minute Miner system starts from the client’s input which is website URL. Once a client inputs web URL, the system automatically access to the demanded website and brings back the client the verification whether the website was accessible or not. If the access to the website is verified, the Minute Miner System starts the extracting process and offers the client text data and graphical data. After a while, the result of web data mining will be displayed for the clients as suggested sources and these sources includes several types of data format such as visual representations, spread sheets, and websites.

## Level 0 data flow diagram

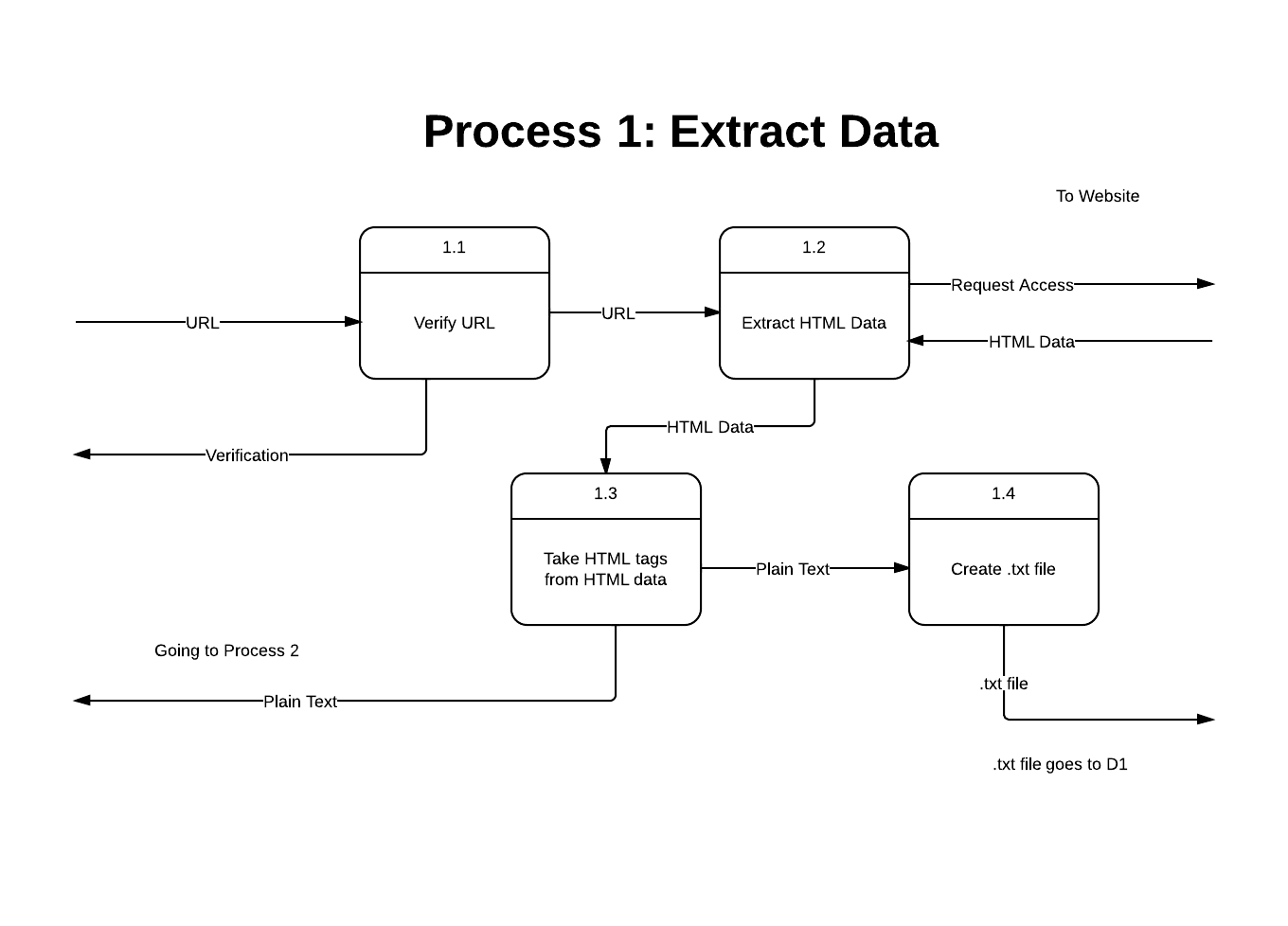
This diagram shows inside the Minute Miner system mentioned in the previous context diagram.



* **1.0: Extract Data**This process 1.0 receives URL from the client and requests access to the target website. After sending the client verification, the process starts extracting HTML data, makes the HTML data to plain text data, and then sent them to the process 2.0.
* **2.0: Create Visual Representation**This process receives plain text data and store them into arrays. After storing, graphical data are sent to the client and the processed data with arrays are passed to the process 3.0.
* **3.0: Suggest similar data source/matches**The process 3.0 receives the processed data from the process 2.0 and then perform data mining process. Finally, 3.0 sends suggested sources to the client.
* **D1: User Device**The process 1.0 makes .txt file consisting of the extracted HTML data and plain texts and send it to the data store which is a hard drive in the client’s device. Then, the client becomes able to open the file anytime and also use it as a backup file.

## Decomposition Diagram of Process 1

This diagram shows the inside of the process 1 described in the level 0 diagram and there are total four processes. Every single process requires at least one class file.



* **1.1: Verify URL**

1.1 checks whether the URL is verified or not. Some websites require users’ authentications but Minute Miner is not able to access to such websites. The URL given by the client goes through the process and is sent to 1.2. If URL is verified then the1.2 starts extracting process. If URL access is denied then 1.1 sends the client a message as “failed”.

* **1.2: Extract HTML Data**1.2 receives URL from the process 1.1, access to the website, and reads HTML data of the website. This HTML data is imported to a class file. If the access to the website is denied then 1.1 sends the client a message as “failed”.
* **1.3: Take HTML tags from HTML data**Process 1.3 receives the HTML data extracted in 1.2 and starts taking HTML tags off. As a result, the extracted HTML data becomes plain text data and it is sent to the process 1.4 and 2.0.
* **1.4: Create .txt file**1.4 receives plain text data from 1.3, makes it into .txt file, and send it to D1 data store.

## Decomposition Diagram of Process 2

This diagram shows the inside of the process 2 described in the level 0 diagram and there are total five processes. Every single process requires at least one class file.

## process2 - New Page-.png

* **2.1: Store plain texts into arrays**

The process 2.0 receives plaintext data form 1.4 and store it into arrays. The arrays are sent to 2.2, 2.3, and 2.4.

* **2.2: Create a pie chart**

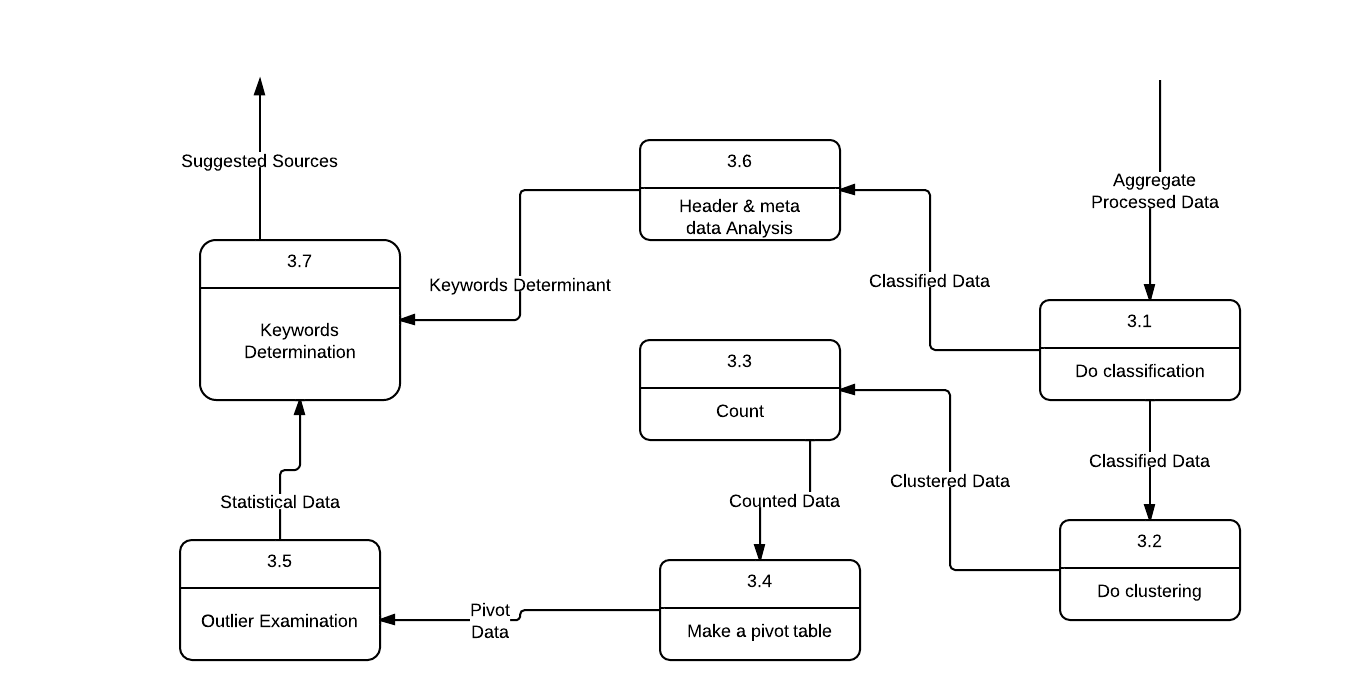
2.2 creates a pie chart based on the array data sent from 2.1.

* **2.3: Create a bar chart**

2.3 creates a bar chart based on the array data sent from 2.1.

* **2.4: Create a Time series chart**2.4 creates a Time Series chart based on the array data sent from 2.1.
* **2.5: Collect Outputs**2.5 collects all the visual representations generated in the previous three processes and send them to the client. In addition, the visual representations are show in an integrated interface. After sending the graphical data to the client, the new array data resulted from statistical calculations in each chart creation process is passed as aggregate processed data (unstructured) to the process 3.0.

## Decomposition Diagram of Process 3

This diagram shows the inside of the process 3 described in the level 0 diagram and there are total seven processes. Every single process requires at least one class file.

* **3.1: Do classification**Aggregate processed data from the process 2.5 is passed into the process 3.1 and this process classifies the aggregate data. The classified data is passed to 3.2.
* **3.2: Do clustering**In 3.2, data clustering is performed and the clustered data is passed to 3.3.
* **3.3: Count**The process 3.3 counts the clustered data passed from 3.2. There are several types of counting methods inside the class file. After counting all the data in arrays, these are sent to 3.4.
* **3.4: Make a pivot table**3.4 places every element inside the arrays sent from 3.3 to make a pivot table. Once a pivot table is created, some statistical components are automatically generated such as outliers, mean, median, mode, range, variance, and standard deviation, and then these pivot data is passed to 3.5.
* **3.5: Outlier examination**In 3.5, the pivot data is received and outlier examination is performed. After all the tasks are done, the finalized statistical data is sent to 3.7.
* **3.6: Header & metadata analysis**The process 3.6 acquires classified data from 3.1 and extract header and metadata information. The data are sorted and processed in a statistical method and sent to keyword determinant information to 3.7.
* **3.7: Keywords Determination**In this process, finalized structured statistical data and keyword determinants produced from header and metadata are received and the data matching percentages are calculated. Related sources are automatically searched online based on the result of the calculations and the search results are sent to the client as suggested sources.

# User Manual

**1.1 System Overview**

This data mining software shall gather text based information from websites though Java. The software shall obtain HTML information from specific websites, and precede to convent into basic texts files for analysis. The gathered information is then processed into a graphical representation easily accessible and readable to all users.

**1.2 User Access**

Minute Miner will be accessible and usable through a well-defined graphical user interface. There is no coding involved from the user and no tasks will be command line based.

**1.3 Operational Status**

Minute Miner is currently under development and is released as alpha stage software. Limited support will be provided under current conditions.

**1.4 Authorized Use Permission**

Minute Miner is provided to the public free of charge and is accessible by anyone for use. However, the software might not be used for servicing any illegal activity or with intention to cause harm to others and/or systems. Minute Miner may not be reverse engineered and may not be altered without written permission and consent from the original developers. Minute Miner may not be sold, resold, or used as a marketing tool under any circumstance without proper consent from its original developers.

**2.0 System Summary**

Minute Miner is a free and easy to use software for collecting data from any website on the Internet. It uses basic and complex algorithms to collect information from a website to convert into easily readable graphs and visual representations of data. No prior training in data mining is required to use the program. Basic computer skills are all that is needed to benefit from using this software.

**2.1 System Configuration**

Minute Miner utilizes Java at its core function, and all aspects of the program have been implemented with Java, including its graphical user interface. The software will run on any Windows laptop or desktop. Macs have not yet been tested and are unsupported at this time. Tablets and smartphones are also not supported at this time, but mobile versions of this software might be introduced in the future. At minimum, the computer should have the following hardware specifications to run the software smoothly:

-Windows XP or higher (Win Vista, Win7, Win8)

-Processor with 2.0Ghz or above for optimal calculations (Dual-Core, Quad-core recommended)

-2GB of RAM (DDR2 and up recommended)

-At least 200mb of free hard drive space (Software itself is a small program, but caching and temp files for graphics will accumulate fast.)

-Integrated graphics with at least 64mb of discreet memory

-Latest version of Java recommended for optical performance.

**2.2 User Access**

Any user is allowed to install and use the program without limitations or restrictions except where prohibited by law and Authorized Use Permission on section 1.4.

**3.0 Getting Started**

To start using Minute Miner, not much experience is required. Few simple steps will start the program to start collecting data from a specific webpage of the user’s choice. There will be a variety of parameters a user will be able to alter to fit the desired functions when collecting data, and will be implemented in future releases.

**3.1 Logging On**

Minute Miner does not need any specific log in to function as of this alpha release. In future releases, there will be a login feature to save specific profile settings to collect data and the user will not have to calibrate setting each time the software is run.

**3.2 System Menu**

A splash screen will display a logo with the software name and its current version running on the computer.

**3.2.1 Start**

This icon will allow the user to start the data gathering process and asks for an url of a website to begin.

**3.2.2 Upload**

If not using a website, the user has the ability to upload excel files to Minute Miner to analyze and display data in a graphical representation.

**3.3 Exit System**

A user can exit the program at any time by pressing the exit button or simply closing out of the program by press the X icon on the upper right of the window of the program.

* **Some screen shots of Minute Miner application developments are attached as appendix.**

# Resources

1Araki, K.; Furukawa, Z.; Jingde Cheng, "A general framework for debugging," Software, IEEE ,

vol.8, no.3, pp.14,20.

2Chou, Chen-Huei (2013). Design and Implementation of jAHP: A Java-based Analytic

Hierarchy Process Application. International journal of computer applications, 62 (15), p. 35 - 41

# 3Corbitt, T. (2003). Business intelligence and data mining. *Management Services, 47*(11), 18. Retrieved from http://search.proquest.com/docview/234332816?accountid=11243

# 4Frank, E., Holmes, G., Kirkby, R., and Hall, M. (2002). Racing committees for large datasets. In Proceedings of the International Conference on Discovery Science, pages 153–164. Springer-Verlag.

5Froelich, J., Ananyan, S., & Olson, D. L. (2005, Winter). Business intelligence through text mining. *Business Intelligence Journal, 10*, 43-50. Retrieved from http://search.proquest.com/docview/222638608?accountid=11243

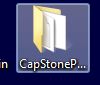
6Hornik, K., Buchta, C., & Zeileis, A. (2009). Open-source machine learning: R meets weka. *Computational Statistics, 24*(2), 225-232. doi:http://dx.doi.org/10.1007/s00180-008-0119-7

7Sauban, M. and Pfahringer, B. (2003). Text categorisation using document profiling. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 411–122. Springer.

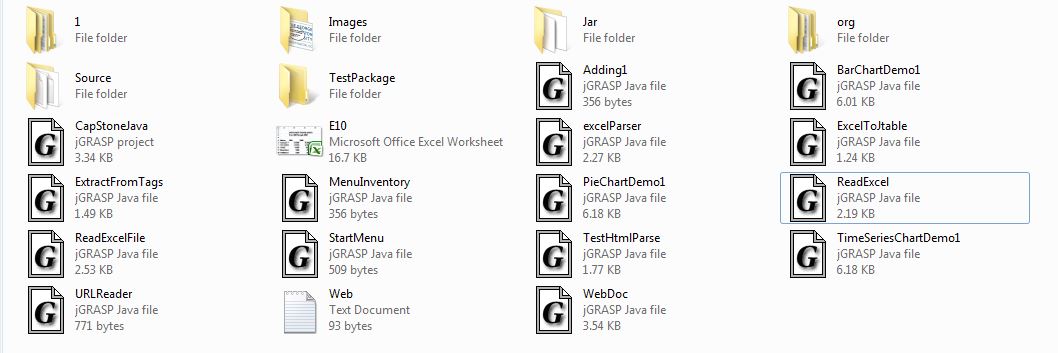
# Appendix

The figures below are some screenshots from the current application development process.

**Screen Shot 1: an application folder**

****

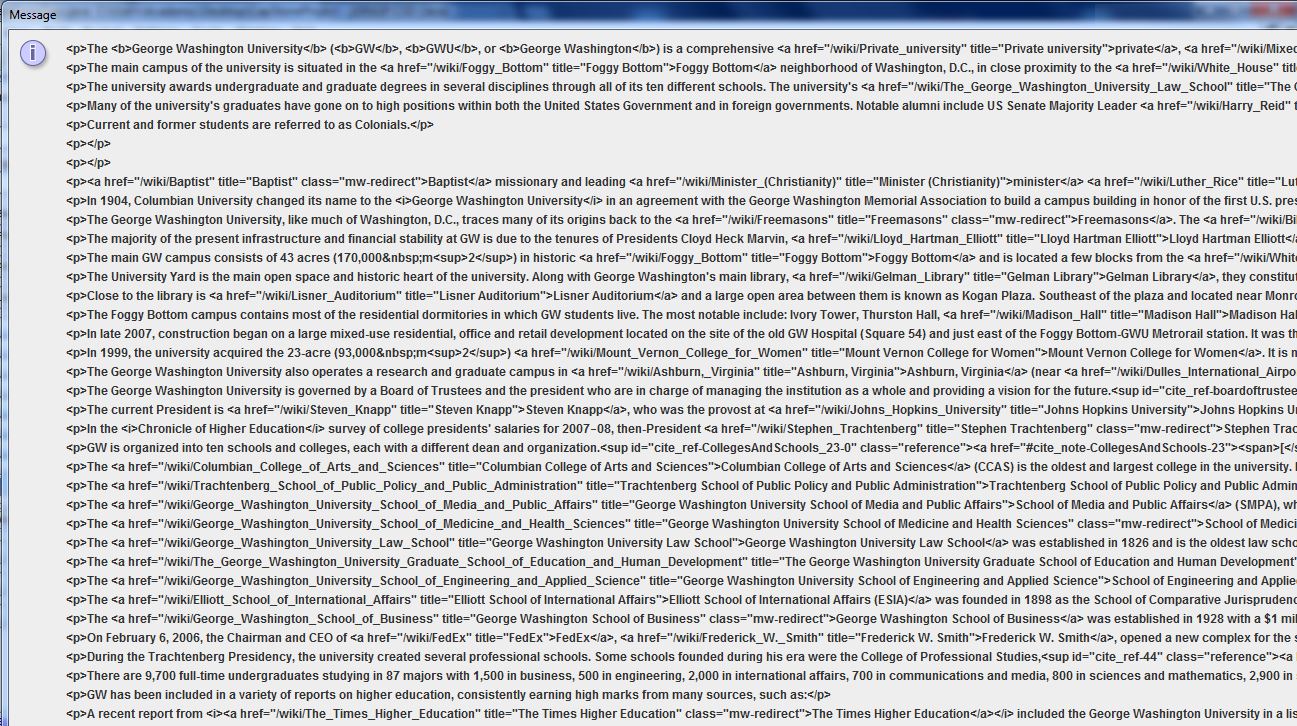
**Screen Shot 2: class files**

****

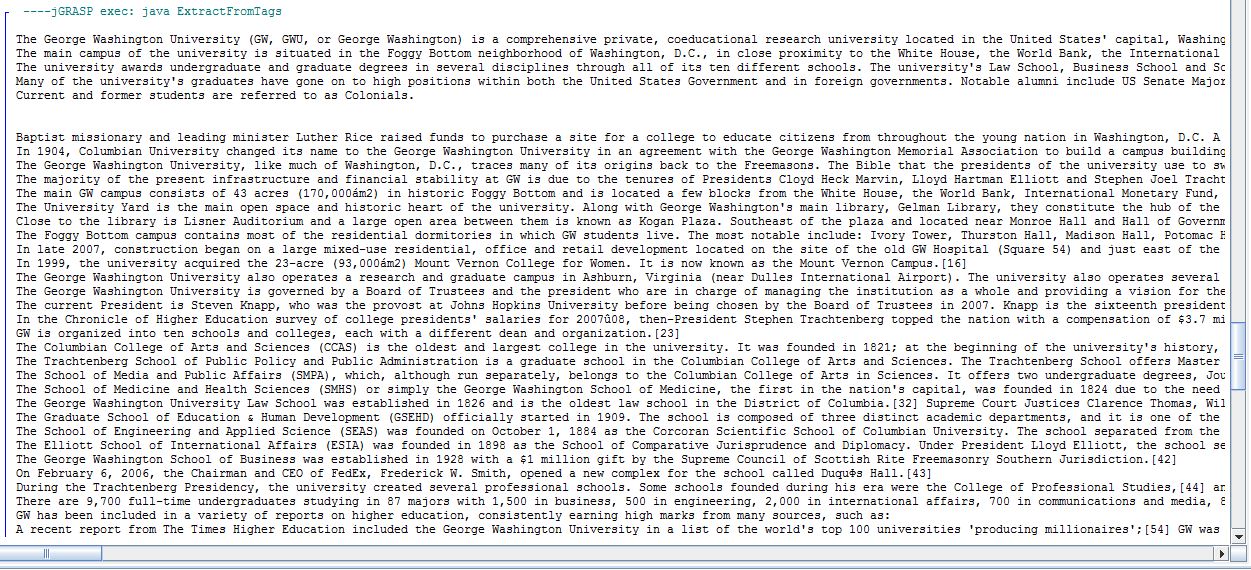
**Screen Shot 3: user input box**

****

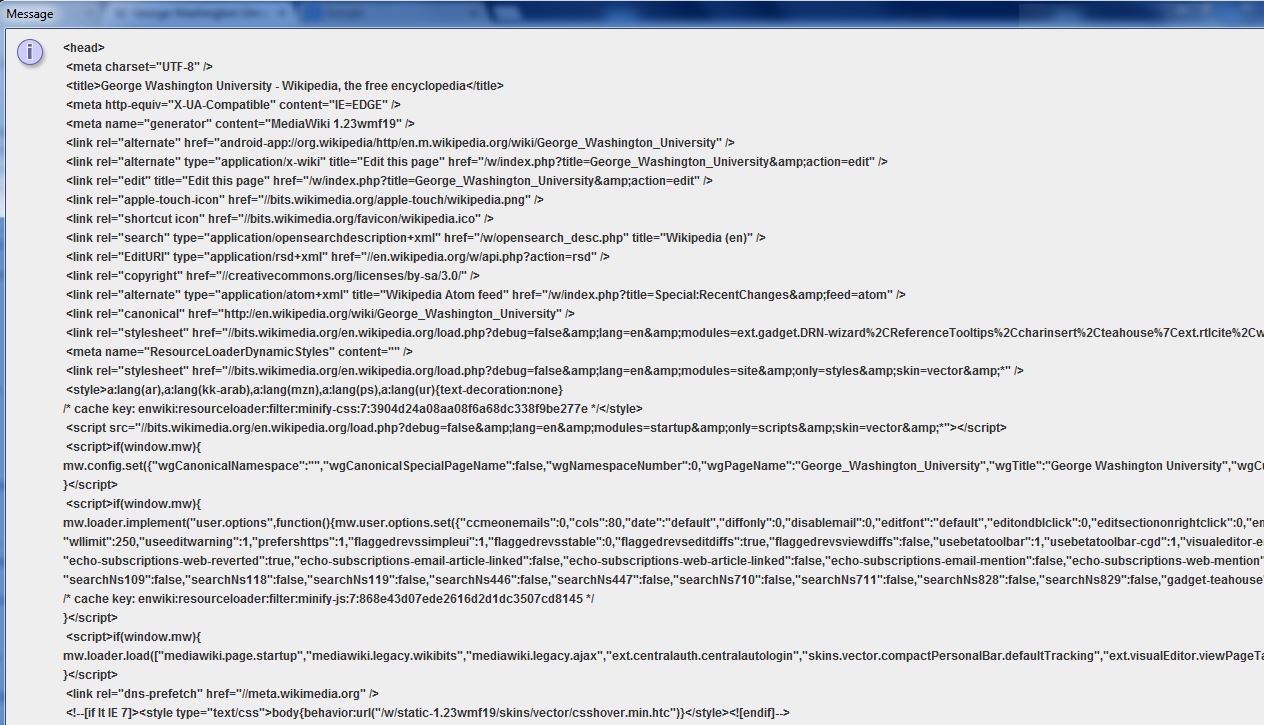
**Screen Shot 4: unfiltered HTML data extracted by Minute Miner**

****

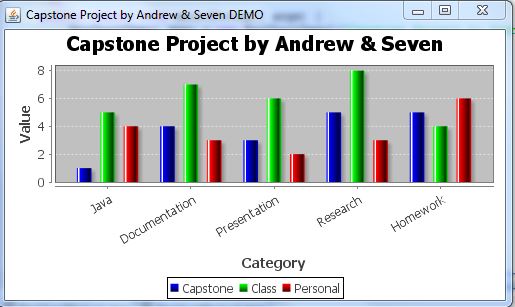
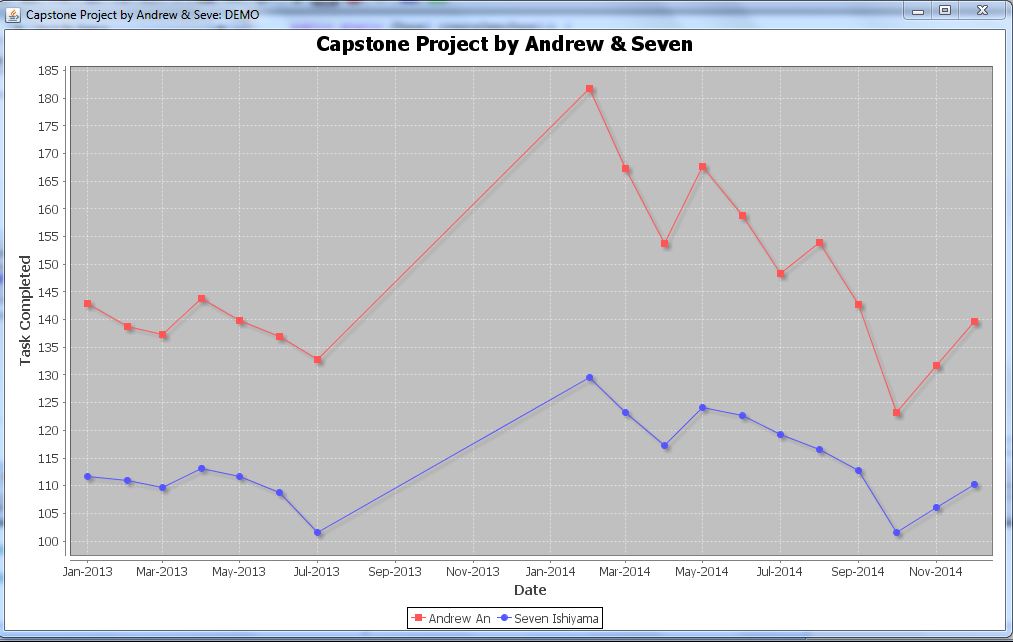
**Screen Shot 5: Filtered HTML data (by HTML tag removal)**

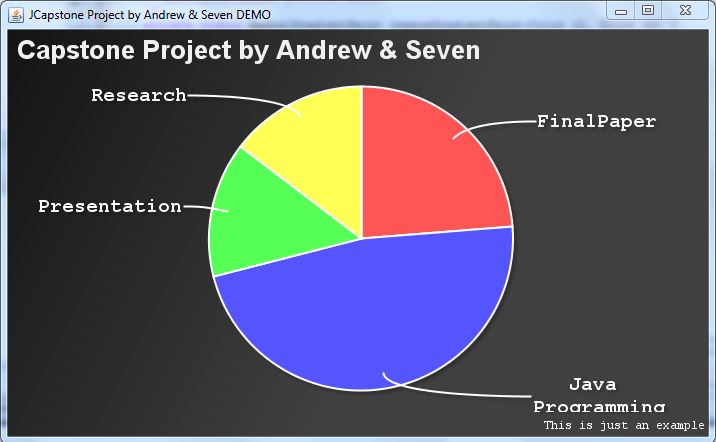
****

**Screen Shot 6: Header and metadata extracted by Minute Miner**

****

**Screen Shot 7: a bar, pie, Time Series chart created based on the data Minute Miner extracted from websites**

****

****

