**1.0 Introduction**

**a) Goals and objectives:**

The main objective objective of the Data Scrape Bot will be to aide researchers cut the time they spend on looking for the data, and help them concentrate on analyzing the data. The users will be able to gather data from one or a set of websites. By applying flags, the data found will be refined to suit the user’s desires better. The user can also decided how much data to gather, as well as the type of data being collected.

**b) Statement of scope:**

There are four main areas that the program will try to fulfil.

1. A search that is highly customizable by the user, with options to add multiple key-words and places to look for the data. The option of what kind of data is also given from text, images, and sound files.
2. The scrape will often try to becomes better by ways of machine learning techniques. This shall make the program more efficient as the user uses it more.

c) Data will be saved to a file, text will be neatly arranged, the links for images and sounds will be saved in their own lines, if possible with a line of description to their side.

d) The data will be displayed in a neat format, easy to be reviewed in order to cut time and find the most useful information.

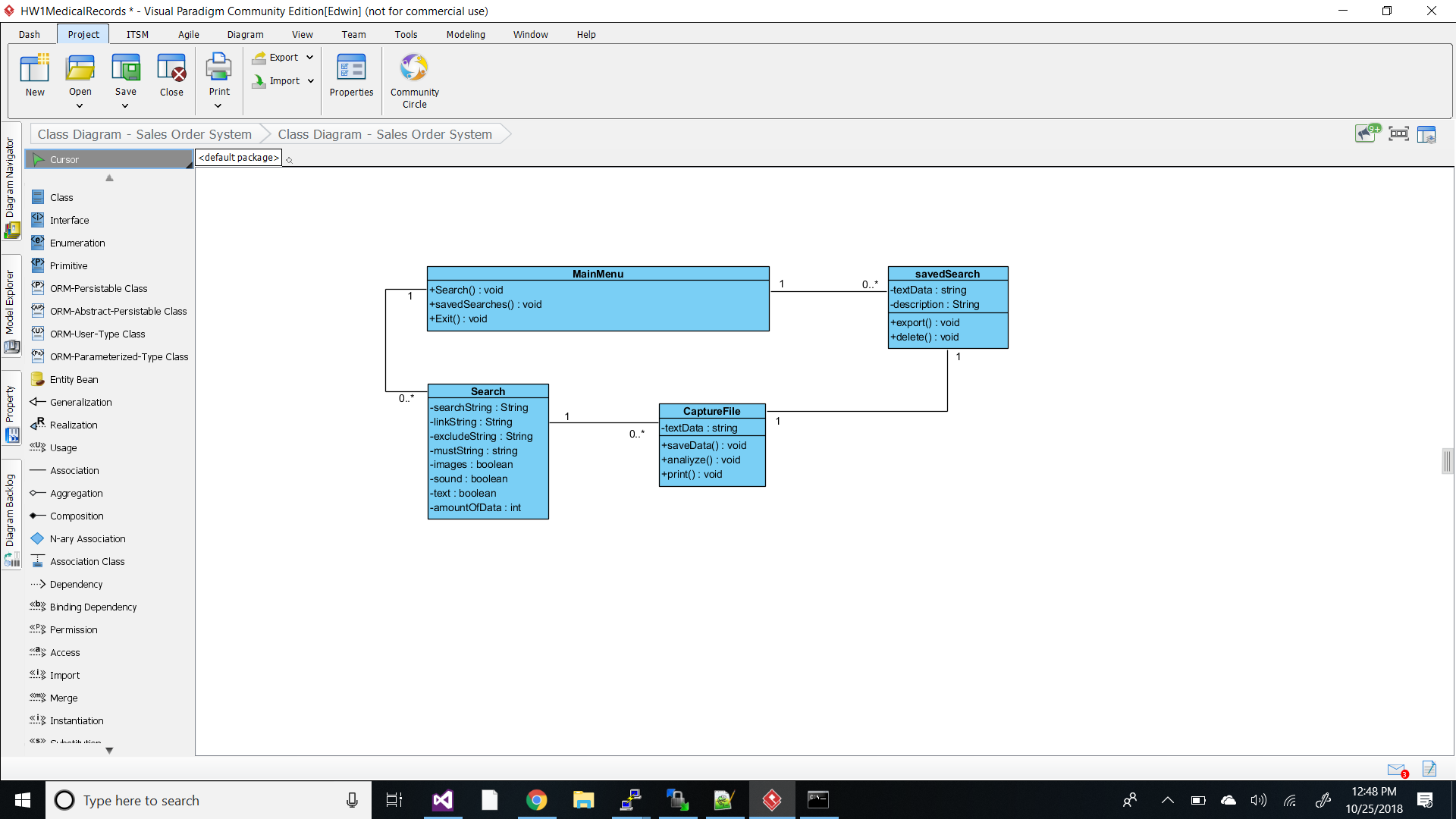
**2.0 Data Design**

**Main Menu:** Where all the functions that will be used are going to be displayed.

**Search:** Uses the url from the user to go scrape data from the web browser.

**CaptureFile:** The user will be given the choice to save his search onto a document.

**SavedSearch:** The saved search can be stored in data, one can later visit this storage to review previous searches and finds.

******

**3.0 Architecture design**

**Architecture Overview**

The main utilities that our program will utilize would be internet access and the computer itself. The storage of the items would also be stored inside the computer that is being used.

**Storage Layer**

The storage layer will be a text file that is generated by the program to store the retrieved information inside the computer.

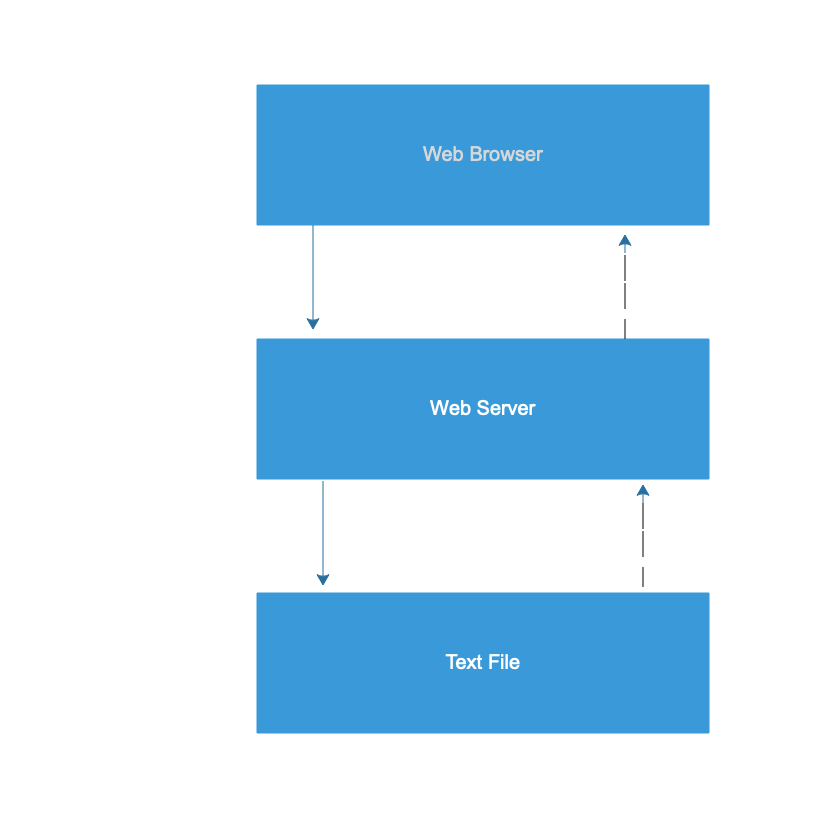
**Application Layer**

Our application layer is mainly a terminal window, this is what our program will always be used to execute. Our program can be executed in any computer that has our program.

**Access Layer**

The requirements needed to access our program and have it run well would be a computer with internet access.

**Architecture Diagram**

****

**4.0 Detail design**

**### main file ###**

**# project: obtaining and making sense of natural data**

**# library list (for your personal files, any standard libraries)**

**# getters (as in get data from web to computer)**

**# transformers (transforming natural data into a mutable format)**

**Pull out the text that the user would read from a website**

**def htmlToText(htmlFile, textFile) :**

**return 'stub'**

**Pull out pictures from website**

**def htmlToPictures(htmlFile, images):**

**return 'stub'**

**fileToStr [file\_name] Turns a file into a large string. Good for testing small files.**

**def fileToStr (fileName) :**

**f = open(fileName, 'r')**

**lines = f.readlines()**

**f.close()**

**return str(lines)**

**turn a picture into workable data**

**def pictureToMatrix(picture) :**

**return 'stub'**

**# selectors (picking out, or compiling, bits of data from large groups of data)**

**count [search string] [string] return the number of occurrences of a string in a string**

**def strCount (searchstr, string) :**

**return 'stub'**

**strLocate [search string] [string] (nth occurrence instead of all occurrences)**

**def strLocate (searchstr, count='', string) :**

**return 'stub'**

**count letters in string**

**def letterCount (string) :**

**return 'stub'**

**count tuples of letters in string for example occurences of 'ab' vs 'gh'**

**def nletterCount (n, string) :**

**return 'stub'**

**count the number of image files in image folder**

**def imageCount (folder) :**

**return 'stub'**

**# putters (put data back into a file for use by another program)**

**make data into an excel file format**

**def datTosheet (dat, xlName) :**

**make a small amount of data into a powerpoint slide**

**def datToPP (dat, PPname) :**

**put data into a visual format, such as a graph. Python comes with many libraries for visualizing data.**

**def datTovis (dat, fileName) :**

**personal favorite" -jw :**

**Put data into a elisp list. Very useful for working with data while coding or doing homework.**

**def datToLisp (dat, fileName) :**

**# tests**

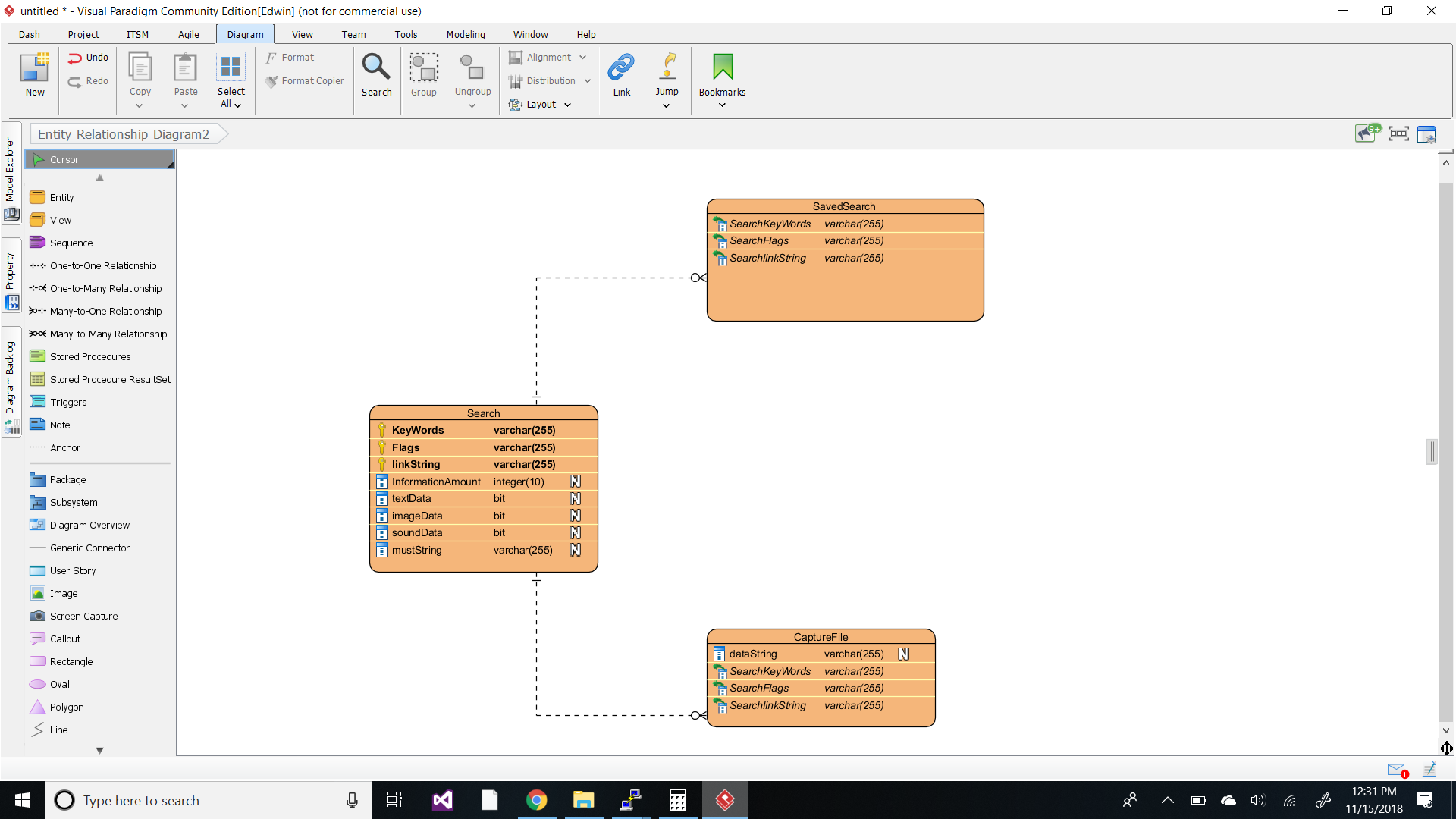
**fileName = "Model\_theory.txt"**

**searchtxt = "math"**

**#print(fileToStr(fileName))**

**5.0 Database design**

The database design is very simple since the program is simple. There are only 3 entities and two of the communicate with the primary one which is the search.

****

**6.0 Software interface design**

**External Machine Interfaces**

Our program is intended to be run through any computer that is capable to connect to a web browser. This means the computer must have internet access.

**External system Interfaces**

The system interface required for this would be a web browser. The program will be going into the user’s url address and extract the information desired.

**Human Interfaces**

The human interface will be mainly a terminal window open with a set of description of what they can do. The main target of the program is for programmers and having a terminal window open would be efficient for programmers to use.

**7.0 Appendices**

**Assumptions**

In our program we are under the assumption that the program can extract images and sound recordings as well.

**Limitations**

One of the main restrictions for non programmers is that it only works on terminal, so it might not look attractive to some users.

**Restrictions**

Internet access would be a restriction from allowing the user to extract data.