**Automated Data Frame Filter**

**&**

**Query Generator**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared By** | **Reviewed by** | **Approved By** |
| **Name** | Rajat Kumar Singh  Kunal Tyagi  Sudhanshu Sharma | Mr. Pankaj Anadure |  |
| **Role** |  |  |  |
| **Date** |  |  |  |

Table of Contents

**Chapter No. Topics Page No.**

Declaration 3

Certificate from the Supervisor 4

Acknowledgement 5

**Chapter-1 Introduction** 6 - 8

1.1 General Introduction

1.2 Problem Statement

1.3 Brief Description of the Solution Approach

1.4 Comparison of existing approaches to the problem framed

**Chapter 2: Requirement Analysis and Solution Approach** 9 – 22

2.1 Description of the project

2.2 Empirical Study of tools and techniques

**Chapter 3:** **Literature Survey** 23- 32

3.1 Requirement Analysis and Solution Approach

3.2 Overall architecture

**Chapter-4 Modelling and Implementation Details** 33 - 42

4.1 Design Diagrams

4.2 Implementation details and issues

**Chapter-5 Appendix and Conclusion** 43-45

5.1 Appendix

5.2 Conclusion

**References** 46

**DECLARATION**

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Bengaluru, Karnataka

Date: 15/04/2019

Name: Rajat Kumar Singh Kunal Tyagi Sudhanshu Sharma

Candidate Id: 768949 768970 768992

## CERTIFICATE

This is to certify that the work titled **“Automated Data Frame Filter And Query Generated”** submitted by “**Rajat Kumar Singh, Kunal Tyagi and Sudhanshu Sharma**” in partial fulfilment for the award of Intership of Cognizant Techonology Solutions, Bengaluru has been carried out under our supervision. This work has not been submitted partially or wholly to any other Company or Institute for the award of this or any other degree or diploma.

Signature of Supervisor:

Name of Supervisor:

Designation:

Date:

## ACKNOWLEDGEMENT

We want to express our special thanks of gratitude to our supervisor **Mr Pankaj Anadure** who gave us the golden opportunity to do this wonderful project of **Automated** **Data Frame Filter And Query Generator.** He whole heartedly supported us at all the stages of the project and offered his invaluable experience and knowledge to guide us throughout the project and accomplish the project goals.

Further, we would like to thank our parents and friends who helped a lot in finalizing this project within the limited time frame.

Name of Interns

Candidate Id

Date

***Chapter-1***

***Introduction***

1. **INTRODUCTION**

## About this document

### **Purpose & Scope of the document**

The purpose of the software requirements document is to systematically capture requirements for the project and the system “Automated Data Frame Filter And Query Generator.” to be developed. Functional requirements of this system are captured in this document. It also serves as the input for the project scoping,

The scope of this document is limited to addressing the requirements from a user, quality, and non-functional perspective. It is recommended that design aspects are not added in this document

### 

### **Intended Audience**

### Project Team

## About the Software System

The primary objective of the project is to read the metadata of the schema given and the end user should be able to select the columns from different tables and form the report on their own. The tool will form the query on its own.

### **Exclusions**

### The system will operate only on the modules discussed above and will not include any

### additional functionality.

**1.1 GENERAL INTRODUCTION**

The continually evolving world of software development means the ability to transfer your knowledge from one skill set to another is becoming of paramount importance.  As technologies rapidly develop and evolve, it is an almost impossible task to retrain from scratch, and so the ability to build on existing skills is the key to keeping at the forefront of technological evolution.

One of the most important aspects of application development is managing the modules that make up an application. Large applications can consist of literally thousands of modules, and millions of lines of code. In addition, modules which are important to the project as a whole but which are not compiled into the application itself (such as design specifications, test scripts, and documentation) must also be tracked and maintained.

Oracle Forms includes an Application Programming Interface (API) that enables Java programming language to read, edit, save, and compile forms module files (.txt, .csv, .json, .xml files) from self-written Java programmers. Form module files are normally created and edited using Form Builder, the design-time component of Oracle Forms. The Forms API gives you access to almost all the Form Builder functionally, and give programmatic interface and not an interactive development environment.

**1.2 PROBLEM STATEMENT**

“Automated Data Frame Filter And Query Generator” proposes to develop a Reporting tool which can read the metadata of the schema given and the end user should be able to select the columns from different tables and form the report on their own. The tool will form the query on its own.

**1.3 Brief Description of the Solution Approach**

Consider the following example problem that we might face using Form Builder. Let us assume that our Application contains 100 page form module files, each having different type of modules files (.txt, .csv, .xml, .json ). It is more complicated when we open different type of file in different GUI Form. Additionally, we have different tables in the database, each having different columns and none of the tables have any common key columns between them. The solution is to give the user an option to fetch the different columns from different tables as per required and form a report table as a resultant of combination of all the columns selected. Even more important than understanding the process for developing an effective GUI is understanding the people who will use it. In fact, our success is directly related to how well we understand our users. Therefore, we are creating a GUI to form a report using different tables as well as querying the data accordingly and to select different types of files that can be opened in a single Application and we fetch the data from the file and show that in the form of table.

# **1.4 Comparison of Existing Approaches to the Problem Framed**

Existing approaches have worked mainly on same type of File using Form Builder. A little work is done on some other module. skipped. Our model combines all type of files (.csv, .txt, .xml, .json) in single Application and show in the form of table.

***Chapter-2***

***Literature Survey***

**Description of the project**

If you have been part of the IT industry, you would know the challenge of working with different data types as well tables . Different formats, different compression, different parsing on different systems.

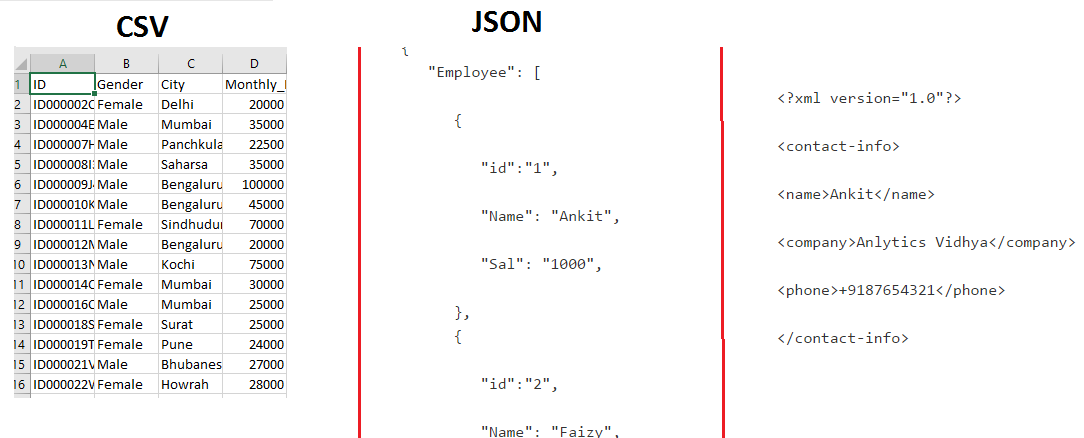
For any data scientist or data engineer, dealing with different formats can become a tedious task. In real-world, people rarely get neat tabular data. Thus, it is mandatory for any engineer to be aware of different file formats, common challenges in handling them and the best / efficient ways to handle this data in real life.

This project provides common formats an engineer must be aware of. I will first introduce you to different common file formats used in the industry. Later, we’ll see how to read these file formats using JAVA

## What is a file format?

A file format is a standard way in which information is encoded for storage in a file. First, the file format specifies whether the file is a binary or ASCII file. Second, it shows how the information is organized. For example, comma-separated values (CSV) file format stores tabular data in plain text.

To identify a file format, you can usually look at the file extension to get an idea. For example, a file saved with name “Data” in “CSV” format will appear as “Data.csv”. By noticing “.csv” extension we can clearly identify that it is a “CSV” file and data is stored in a tabular format.

[](https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/03/01103613/CSV.png)

**Why should we understand different file formats**

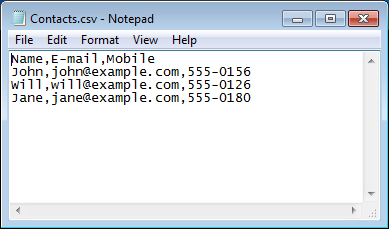
The files you will come across will depend on the application you are building. For example, in an image processing system, you need image files as input and output. So you will mostly see files in jpeg, gif or png format.

you need to understand the underlying structure of various file formats, their advantages and dis-advantages. Unless you understand the underlying structure of the data, you will not be able to explore it. Also, at times you need to make decisions about how to store data.

Choosing the optimal file format for storing data can improve the performance of your models in data processing.

Different file formats and how to read them

* XLSX
* ZIP
* Plain Text (txt)
* JSON
* XML



**Reading the data from CSV**

how to read a CSV. For loading the data you can use the “pandas” library in python.

import pandas as pd

df = pd.read\_csv(“/home/Loan\_Prediction/train.csv”)

**Reading a CSV file in Java**

OpenCSV is a CSV parser library for Java. OpenCSV supports all the basic CSV-type operations you are want to do. Java 7 is currently the minimum supported version for OpenCSV. Java language does not provide any native support for effectively handling CSV files so we are using OpenCSV for handling CSV files in Java.

**Some useful classes of open csv**

1. **CSVReader –**This class provides the operations to read the CSV file as a list of String array.
2. **CSVWriter –**This class allows us to write the data to a CSV file.
3. **CsvToBean –**This class will be used when you want to populate your java beans from a CSV file content.
4. **BeanToCsv –**This class helps to export data to CSV file from your java application.

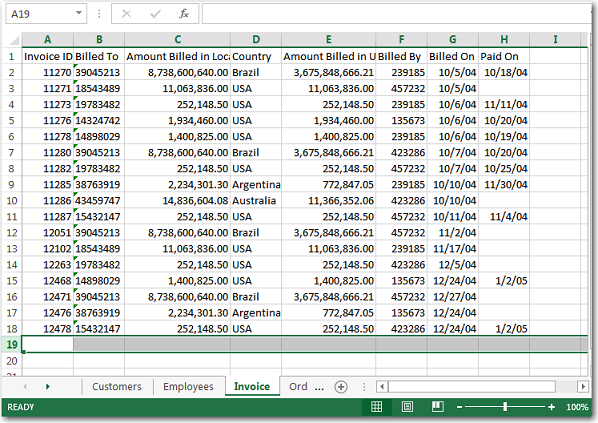
**Text File**

A **text file** (sometimes spelled **textfile**; an old alternative name is **flatfile**) is a kind of [computer file](https://en.wikipedia.org/wiki/Computer_file) that is structured as a sequence of [lines](https://en.wikipedia.org/wiki/Line_(text_file)) of [electronic text](https://en.wikipedia.org/wiki/Electronic_text). A text file exists [stored as data](https://en.wikipedia.org/wiki/Data_storage) within a [computer file system](https://en.wikipedia.org/wiki/Computer_file_system). In operating systems such as [CP/M](https://en.wikipedia.org/wiki/CP/M) and [MS-DOS](https://en.wikipedia.org/wiki/MS-DOS), where the operating system does not keep track of the file size in bytes, the end of a text file is denoted by placing one or more special characters, known as an [end-of-file](https://en.wikipedia.org/wiki/End-of-file)marker, as padding after the last line in a text file. On modern operating systems such as [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems, text files do not contain any special EOF character, because file systems on those operating systems keep track of the file size in bytes. There are for most text files a need to have [end-of-line](https://en.wikipedia.org/wiki/Newline) [delimiters](https://en.wikipedia.org/wiki/Delimiter), which are done in a few different ways depending on operating system. Some operating systems with [record-orientated file systems](https://en.wikipedia.org/wiki/Record-oriented_filesystem) may not use new line delimiters and will primarily store text files with lines separated as fixed or variable length records.

### **XLSX files**

XLSX is a Microsoft Excel Open XML file format. It also comes under the Spreadsheet file format. It is an XML-based file format created by Microsoft Excel. The XLSX format was introduced with Microsoft Office 2007.

In XLSX data is organized under the cells and columns in a sheet. Each XLSX file may contain one or more sheets. So a workbook can contain multiple sheets.



In above image, you can see that there are multiple sheets present (bottom left) in this file, which are Customers, Employees, Invoice, Order. The image shows the data of only one sheet – “Invoice”.

### **JSON file format**

JavaScript Object Notation(JSON) is a text-based open standard designed for exchanging the data over web. JSON format is used for transmitting structured data over the web. The JSON file format can be easily read in any programming language because it is language-independent data format.

{

  "Employee": [

      {

         "id":"1",

         "Name": "Ankit",

         "Sal": "1000",

      },

      {

         "id":"2",

         "Name": "Faizy",

         "Sal": "2000",

      }

   ]

}

JSON is a simple file format that is very easy for any programming language to read.

simplicity means that it is generally easier for computers to process than others, such as XML

Working with JSON in Python is almost the same such as working with a python dictionary.

import json

json\_data = open("file root")

data = json.load(json\_data)

Then data[“key”] prints the data for the json

**Parsing JSON in Java/JSON Processing in Java**

**JSON is parsed through** the [org.json](https://github.com/stleary/JSON-java) library.

To use org.json to parse JSON in Java, we need to add the library as a dependency.

import org.json.JSONArray;

import org.json.JSONObject;

public class ParseJSON {

static String json = "...";

public static void main(String[] args) {

JSONObject obj = new JSONObject(json);

String pageName = obj.getJSONObject("pageInfo").getString("pageName");

System.out.println(pageName);

JSONArray arr = obj.getJSONArray("posts");

for (int i = 0; i < arr.length(); i++) {

String post\_id = arr.getJSONObject(i).getString("post\_id");

System.out.println(post\_id);

}

}

}

First, we need to convert the JSON string into a JSON Object, using JSONObject class.

Also, note that “pageInfo” is a JSON Object, so we use the getJSONObject method.

Likewise, “posts” is a JSON Array, so we need to use the getJSONArray method.

### **XML file format**

XML is also known as Extensible Markup Language. As the name suggests, it is a markup language. It has certain rules for encoding data. XML file format is a human-readable and machine-readable file format. XML is a self-descriptive language designed for sending information over the internet. XML is very similar to HTML, but has some differences. For example, XML does not use predefined tags as HTML.

Let’s take the simple example of XML File format.

The following example shows an xml document that contains the information of an employee.

<?xml version="1.0"?>

<contact-info>

<name>Ankit</name>

<company>Anlytics Vidhya</company>

<phone>+9187654321</phone>

</contact-info>

The “<?xml version=”1.0″?>” is  a  XML declaration at the start of the file (it is optional). In this deceleration, version specifies the XML version and encoding specifies the character encoding used in the document. <contact-info> is a tag in this document. Each XML-tag needs to be closed.

**XML Sample**

<?xml version="1.0"?>

<catalog>

<book id="bk101">

<author>Gambardella, Matthew</author>

<title>XML Developer's Guide</title>

<genre>Computer</genre>

<price>44.95</price>

<publish\_date>2000-10-01</publish\_date>

<description>An in-depth look at creating applications

with XML.</description>

</book>

<book id="bk102">

<author>Ralls, Kim</author>

<title>Midnight Rain</title>

<genre>Fantasy</genre>

<price>5.95</price>

<publish\_date>2000-12-16</publish\_date>

<description>A former architect battles corporate zombies,

an evil sorceress, and her own childhood to become queen

of the world. </description>

</book>

<book id="bk103">

<author>Corets, Eva</author>

<title>Maeve Ascendant</title>

<genre>Fantasy</genre>

<price>5.95</price>

<publish\_date>2000-11-17</publish\_date>

<description>After the collapse of a nanotechnology

society in England, the young survivors lay the

foundation for a new society.</description>

</book>

</catalog>

**Parsing XML files in java**

There are a few ways to parse XML in Java:

SAX parser: An event-based sequential access parser API that only operates on portions of the XML document at any one time.

DOM parser: The Document Object Model parser is a hierarchy-based parser that creates an object model of the entire XML document, then hands that model to you to work with.

JAXB: The Java Architecture for XML Binding maps Java classes to XML documents and allows you to operate on the XML in a more natural way.

The most popular way to work with XML is to use the DOM parser

**JAVA DOM Parser**

The Java DOM API for XML parsing is intended for working with XML as an object graph in memory - a "Document Object Model (DOM)". The parser traverses the XML file and creates the corresponding DOM objects. These DOM objects are linked together in a tree structure. Once the parser is done, you get this DOM object structure back from it. Then you can traverse the DOM structure back and forth as you see fit.

Following are the steps used while parsing a document using JDOM Parser.

* Import XML-related packages.
* Create a SAX Builder.
* Create a Document from a file or stream
* Extract the root element
* Examine attributes
* Examine sub-elements

**Creating A Java DOM XML Parser**

Creating a Java DOM XML parser is done using the javax.xml.parsers.DocumentBuilderFactoryclass

For example:

DocumentBuilderFactory builderFactory =

DocumentBuilderFactory.newInstance();

DocumentBuilder builder = null;

try {

builder = builderFactory.newDocumentBuilder();

} catch (ParserConfigurationException e) {

e.printStackTrace();

}

It is the DocumentBuilder instance that is the DOM parser. Using this DOM parser you can parse XML files into DOM objects, as we will see in the next section.

**Parsing XML with a Java DOM Parser**

Parsing an XML file into a DOM tree using the DocumentBuilder is done like this:

try {

Document document = builder.parse(

new FileInputStream("data\\text.xml"));

} catch (SAXException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

**2.2 Empirical Study of tools and techniques**

* Java
* Multithreading
* File Handling
* JDBC – for login page
* AWT
* Swing
* Event Handling
* DOM Parser
* Text File
* JSON File
* CSV File
* XML File

**2.2.1 JAVA**

Java a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) [computer-programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is intended to let [application developers](https://en.wikipedia.org/wiki/Application_developer) "[write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)", meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to ["bytecode"](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The language derives much of its original features from [SmallTalk](https://en.wikipedia.org/wiki/SmallTalk), with a [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. As of 2018, Java was according to [Github](https://en.wikipedia.org/wiki/Github) one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity), particularly for [client-server](https://en.wikipedia.org/wiki/Client%E2%80%93server) [web applications](https://en.wikipedia.org/wiki/Web_applications), with a reported 9 million developers.

**2.2.2 Multithreading**

**Multithreading in java** is a process of executing multiple threads simultaneously.

A thread is a lightweight sub-process, the smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.

However, we use multithreading than multiprocessing because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

**2.2.3 File Handling**

After studying, [**Singleton Class in Java**](https://data-flair.training/blogs/singleton-class-in-java/), we are ready to discuss File Handling in Java. Moreover, we will discuss Java Filewriter and Java File reader with their [**constructor s**](https://data-flair.training/blogs/constructor-in-java/)and methods used.

File Handling in Java programming Language: FileWriter and FileReader classes are very frequently used to write and read data from text files (they are Character Stream classes). For any Byte stream classes, if you want to read and write them it is not wise and recommended to use FileInputStream.

**Java FileWriter**

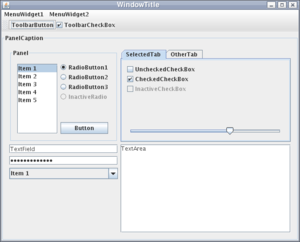
* **FileWriter(File file) –** This constructor constructs a FileWriter object when a file object is given.
* **FileWriter (File file, boolean append) –** This constructor constructs a FileWriter object.
* **FileWriter (FileDescriptor fd) –** This constructor constructs a FileWriter object associated using a file descriptor.
* **FileWriter (String fileName) –**This constructor constructs a FileWriter object when a file name is given.
* **FileWriter (String fileName, Boolean append) –** This constructor constructs a FileWriter object when a file name is given with a Boolean to decide whether it append or not.
* **public void write (int c) throws IOException –** This constructor writes a single character.
* **public void write (char [] stir) throws IOException –** This constructor writes an array of characters.
* **public void write(String str)throws IOException –** This constructor writes a string.
* **public void write(String str**,**int off, int len)throws IOException –** This constructor writes a portion of a string.

Reading and writing are done in an accordance that is character by character, which increases the count of I/O operations and effects performance of the system. BufferedWriter can use with FileWriter to improve the speed of execution of the operatio

**2.2.4 JDBC**

Java Database Connectivity (JDBC) is an [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), which defines how a client may access a [database](https://en.wikipedia.org/wiki/Database). It is a Java-based data access technology used for Java database connectivity. It is part of the [Java Standard Edition](https://en.wikipedia.org/wiki/Java_Standard_Edition)platform, from [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). It provides methods to query and update data in a database, and is oriented towards [relational databases](https://en.wikipedia.org/wiki/Relational_database). A JDBC-to-[ODBC](https://en.wikipedia.org/wiki/ODBC) bridge enables connections to any ODBC-accessible data source in the [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) host environment.

**2.2.5 Swing**

[](https://en.wikipedia.org/wiki/File:Gui-widgets.png)

Example Swing widgets in Java

**Swing** is a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) [widget toolkit](https://en.wikipedia.org/wiki/Widget_toolkit) for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)).[[1]](https://en.wikipedia.org/wiki/Swing_(Java)#cite_note-1) It is part of [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation)'s [Java Foundation Classes](https://en.wikipedia.org/wiki/Java_Foundation_Classes) (JFC) – an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for providing a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI [components](https://en.wikipedia.org/wiki/Software_component) than the earlier [Abstract Window Toolkit (AWT)](https://en.wikipedia.org/wiki/Abstract_Window_Toolkit). Swing provides a [look and feel](https://en.wikipedia.org/wiki/Look_and_feel) that emulates the look and feel of several platforms, and also supports a [pluggable look and feel](https://en.wikipedia.org/wiki/Pluggable_look_and_feel) that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element.

***Chapter-3***

***Literature Survey***

**3.Requirement Analysis and Solution Approach**

**3.1 Overall Description**

**3.1.1 User Interface**

1. The user interface is made with the help of Swing.

2. The data is stored in txt file, CSV file, json file and xml file.

3. The results are displayed using GUI.

**3.1.2 Software Requirement**

1. Operating system: Windows

2. Language: Java

3. Tools: Oracle, Eclipse

4. Java Libraries Required: JDBC, Multithreading, AWT, Swing, Event Handling, Util Package,

XML parser, DOM parser

**3.1.3 Hardware requirement**

1.CPU: 500 MHz processor (Minimum)

2. Computer Processor: Intel i5 or Intel i3 core

3. Computer Memory: 8 Gb or more

4. Graphics hardware: Not required

5. Network: Internet connection required

**3.2 Functional Requirements**

* **Requirement ID R1.1**

Title: Fetch csv File

Description: This action is done to fetch File from data set.

* **Requirement ID R1.2**

Title: Fetch Xml File

Description: This action is done to fetch Xml File from data set.

* **Requirement ID R1.3**

Title: Fetch Json File

Description: This action is done to fetch Json from data set.

* **Requirement ID R1.4**

Title: Fetch Txt

Description: This action is done to fetch Txt from data set.

* **Requirement ID R1.7**

Title: OS requirement

Description: Anomaly detection system must run on Windows- like platform using JAVA

* **Requirement ID R1.8**

Title: Prerequisites

Description: All essential libraries must be installed/imported.

* **Requirement ID R1.9**

Title: Get Trust

Description: This action is done to get the trust value of a node.

**3.2 Non-Functional Requirements**

**A. Reliability**

i. The system should be reliable while delivering the data to/from the user.

ii.The system must collect data from reliable source.

**B. Availability**

i. The system should be active while the user uses the service to get the correct results.

**C. Safety**

i. The ability of the system to operate without catastrophic failure and not causing harm.

**D. Security**

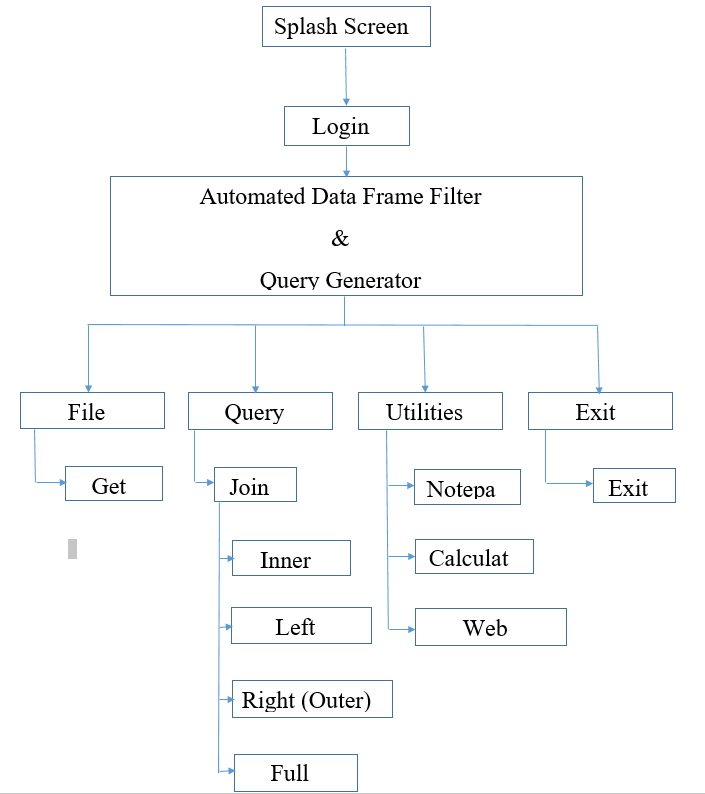
i. The data sent to user has to be protected against data thefts.

ii. The privacy of the data has to be maintained

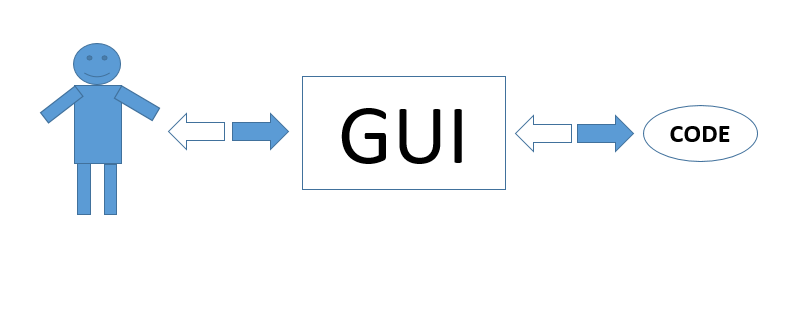
**3.3 Overall architecture**

# **Flow Charts**

Figure 1. Work flow Diagram

****

**Fig – 2 : Flow Chart**



**Fig – 3**

**3.4 Test Plan**

The testing plan consists of various steps which include various types of testing

such as requirement testing, integration testing, performance testing, stress testing

etc.

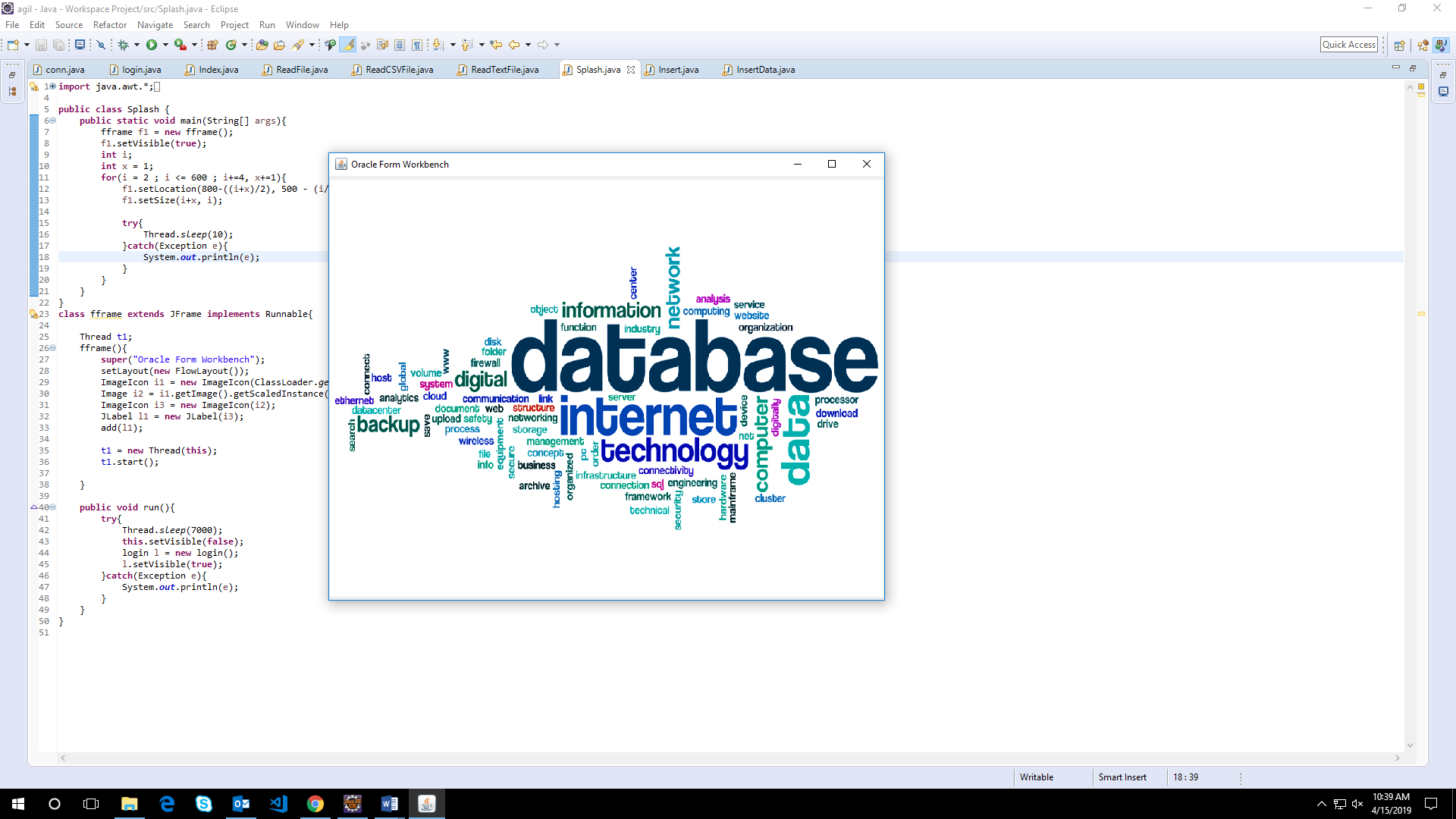
|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Test** | **Performed?** | **Explanation** | **Software component** |
| Requirements Testing | yes | It is done to check the  requirements for our project such as system requirements, platform requirements,  technology  requirements which will help us develop our project in a better  way. The time  requirements are also tested. | Java  Jdbc  package |
| Unit Testing | yes | Unit testing being most important of all testing levels. It helps in finding the bugs during  the progress of the project, at the module or unit level which is  much more  economical to fix as compared to after the  completion of the  project. | Dataset collection  Outlier detection  Trust calculation |
| Integration Testing | yes | Integration testing is being performed to  verify functional performance and  reliability requirements placed on  major design items. It tests that all components within the  module of units interacts completely. | This is applied to all the libraries such as swing, awt, DOM parser. |
| Volume Testing | yes | Volume Testing tests the group of non-functional  tests i.e. for  non-functional  requirements. It means testing a software application with a certain amount of data. This generally involves  testing the software | Outlier Detection  Trust Calculation |
| Load Testing | yes | Load Testing is done to convert the information in  compatible format. | Dataset Collection |

***Chapter-4***

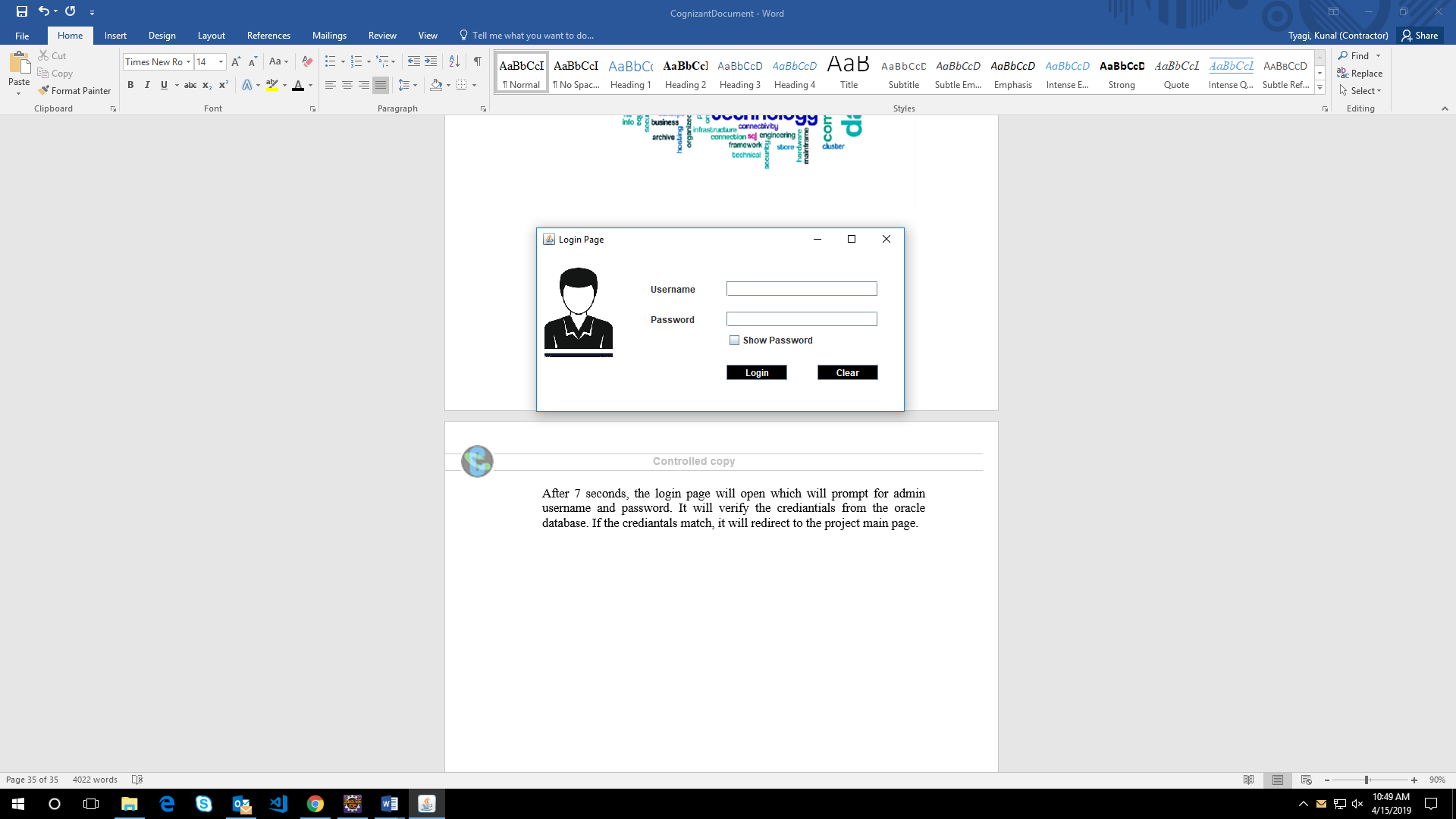
***Modelling and implementation details***

**4.Implementation and findings**

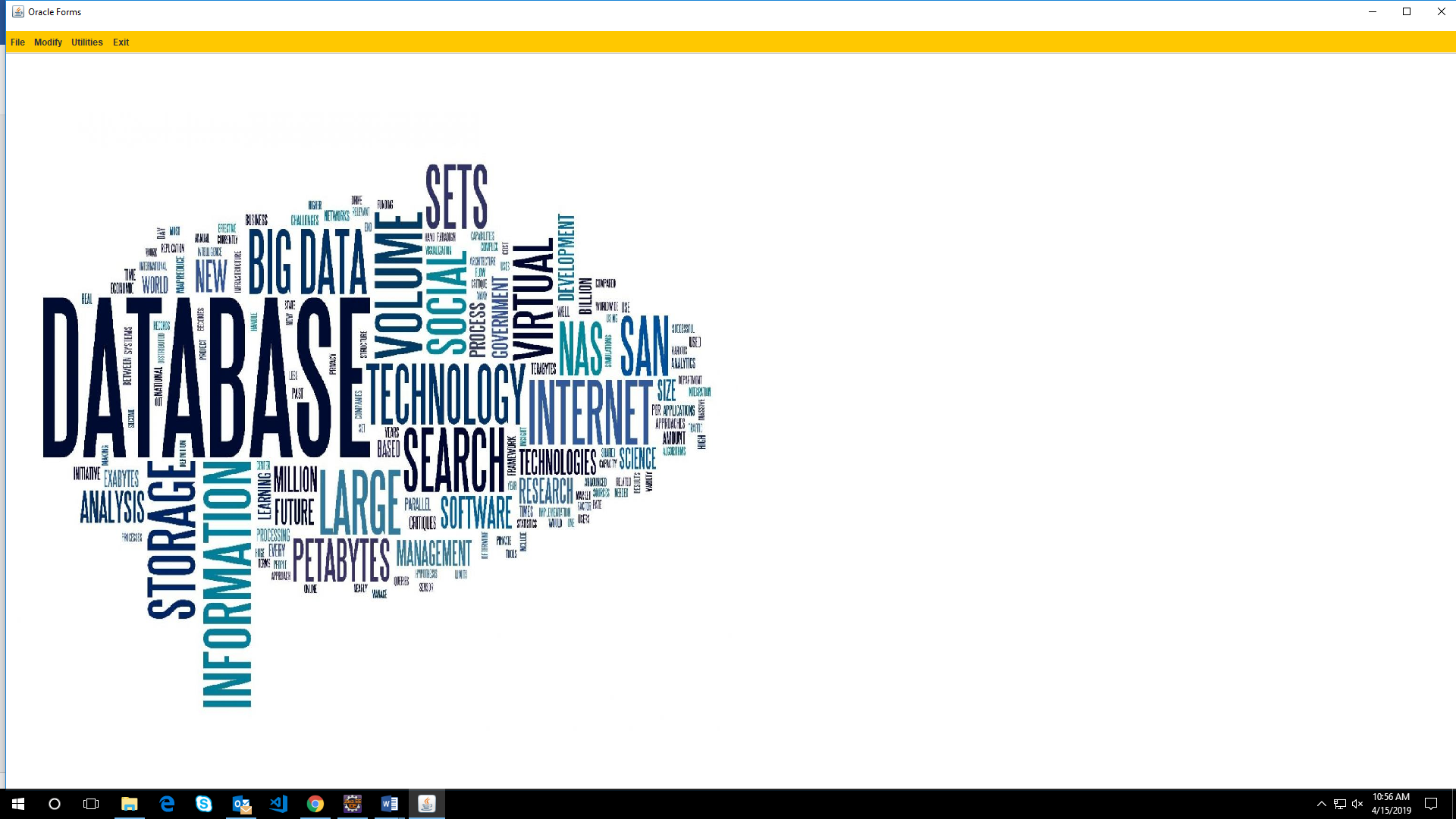
Initially, a frame will open which will display a splash screen for 7 seconds



After 7 seconds, the login page will open which will prompt for admin username and password. It will verify the crediantials from the oracle database. If the crediantals match, it will redirect to the project main page.



After verifying the credentials, the login page will redirect to the page below.



The Main page contains four drop down menus

1 - File

1. Read File

2 - Modify

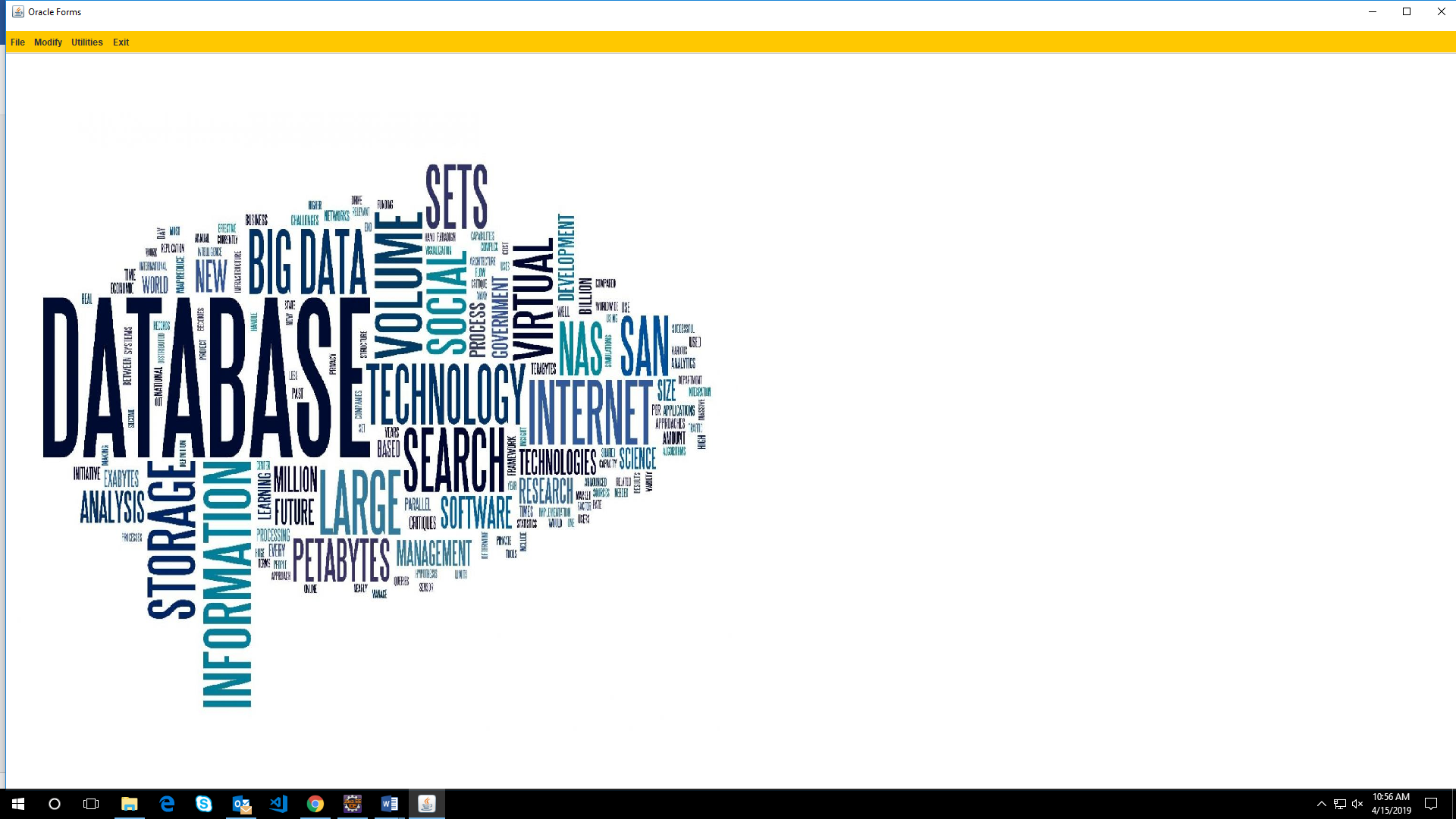
1. Left Join
2. Right Join
3. Inner Join
4. Full Outer Join

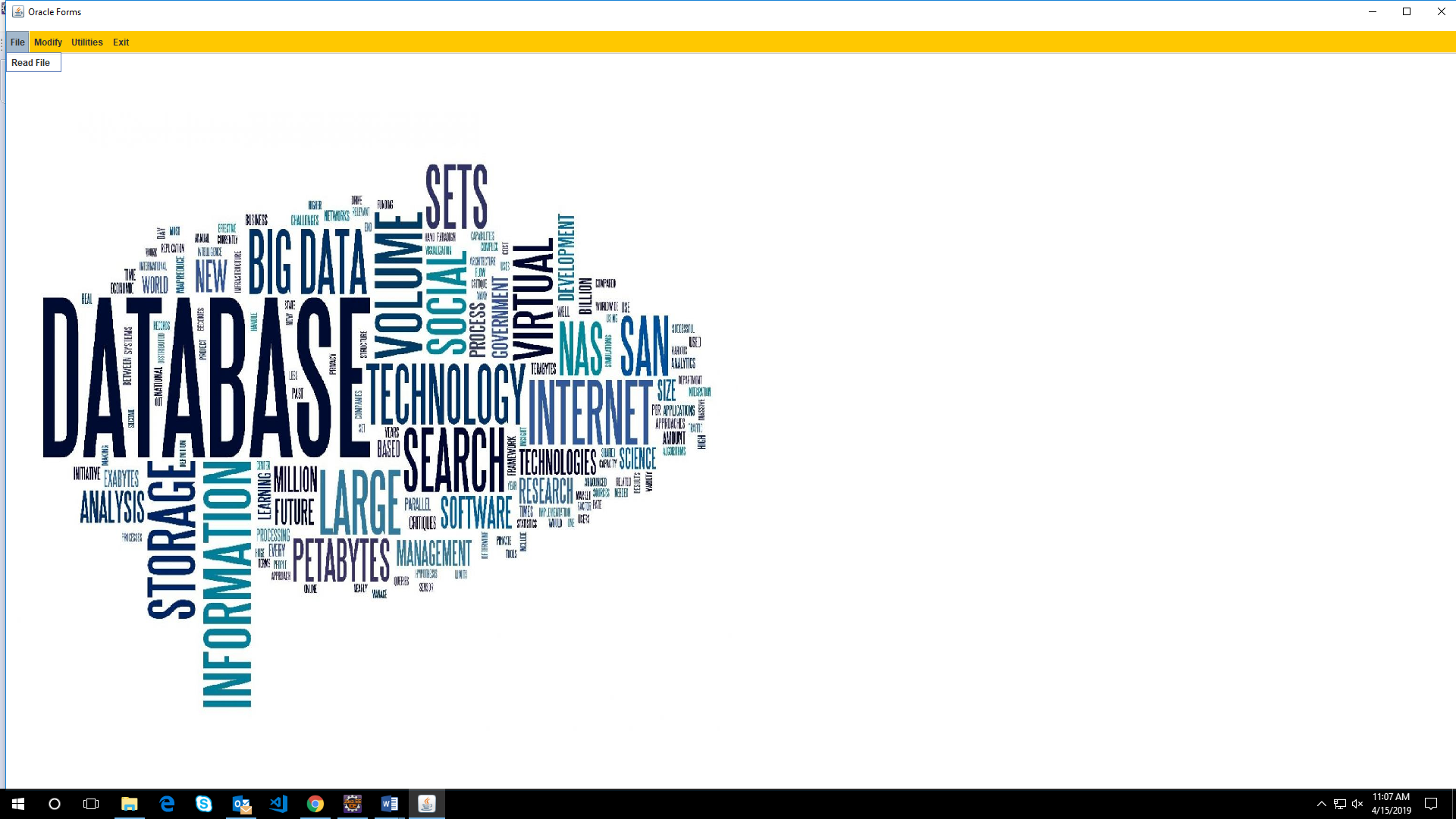
3 - Utilities

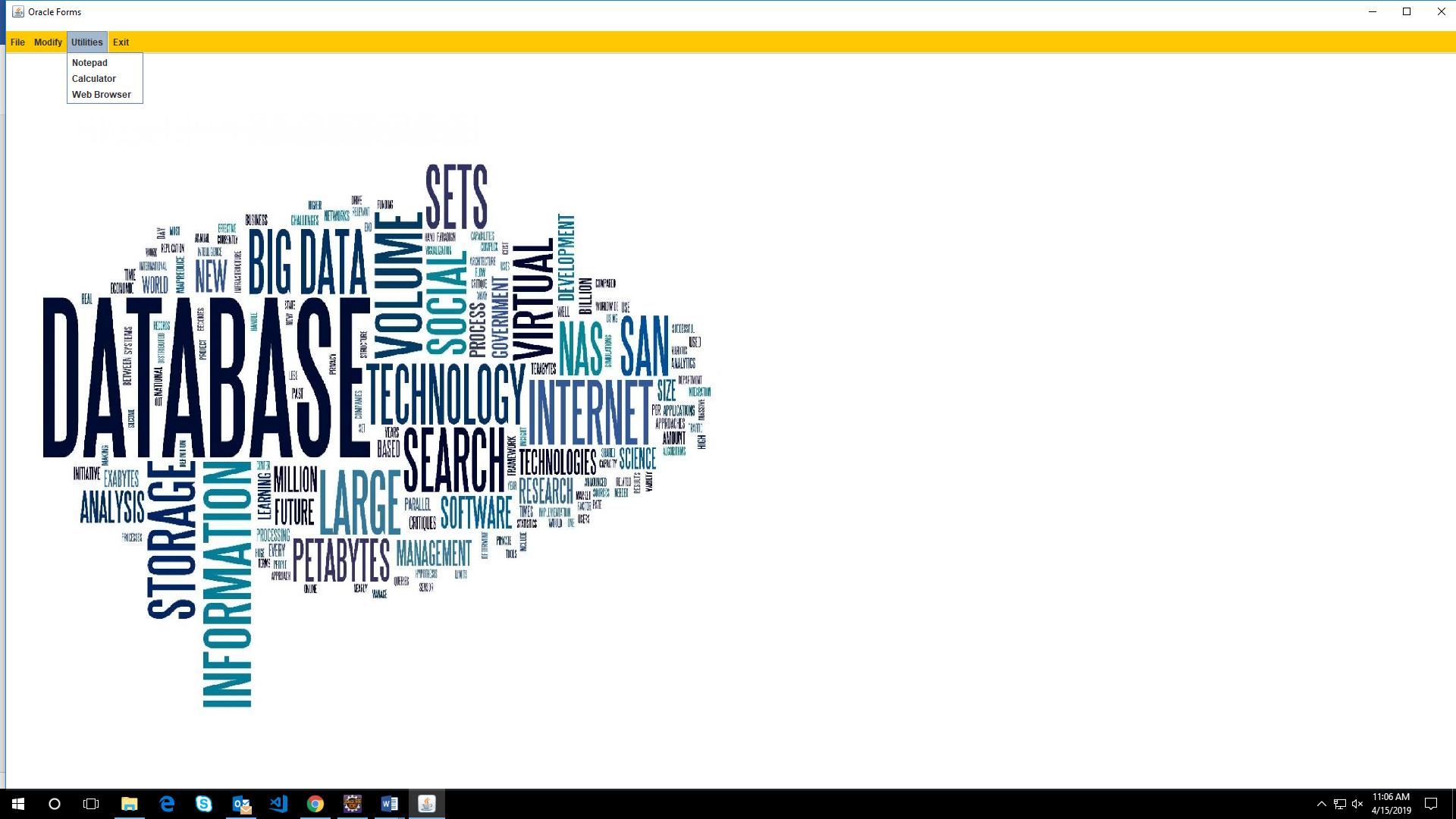
1. Notepad
2. Calculator
3. Web Browser

4 - Exit

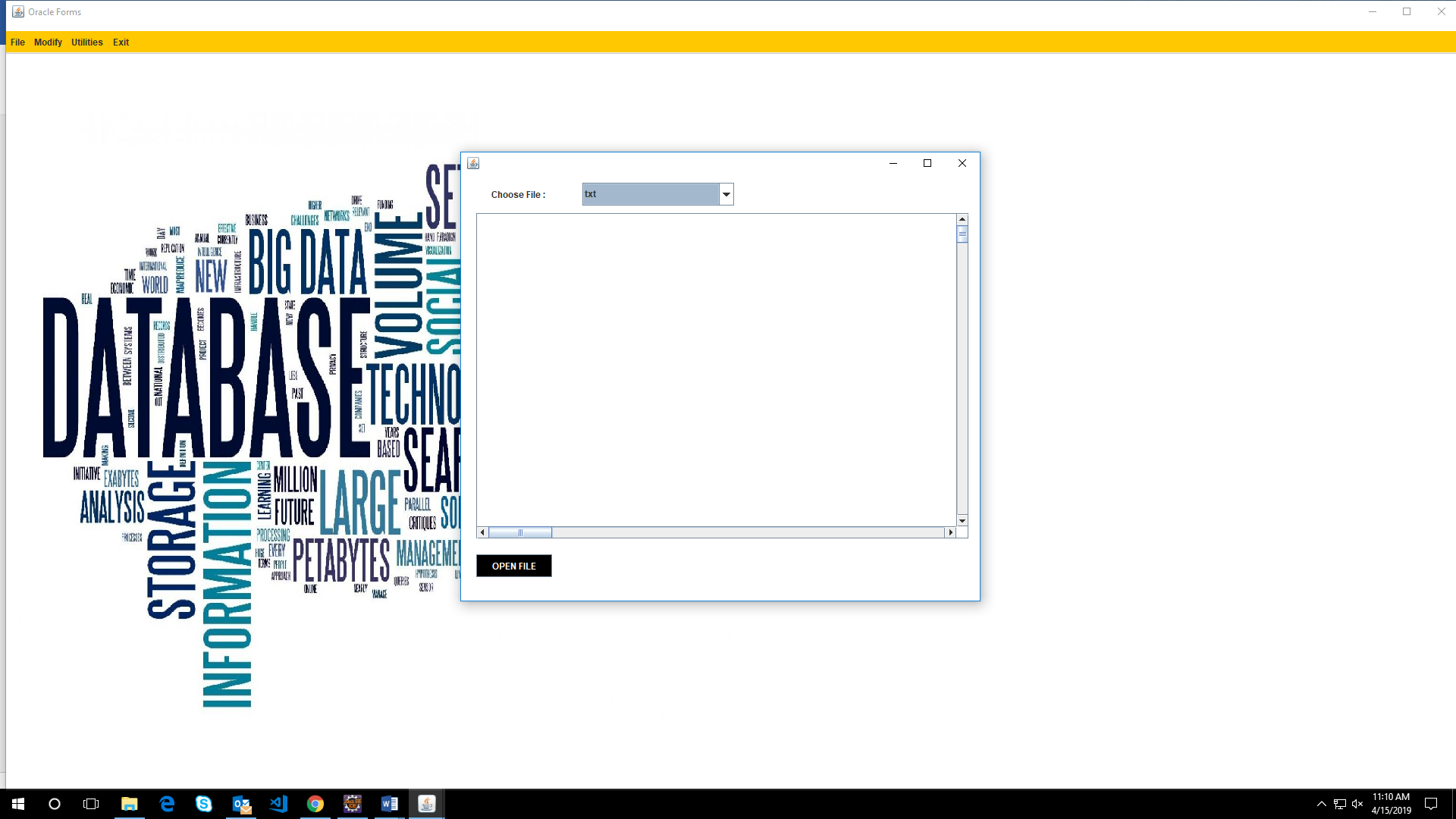
1. Exit



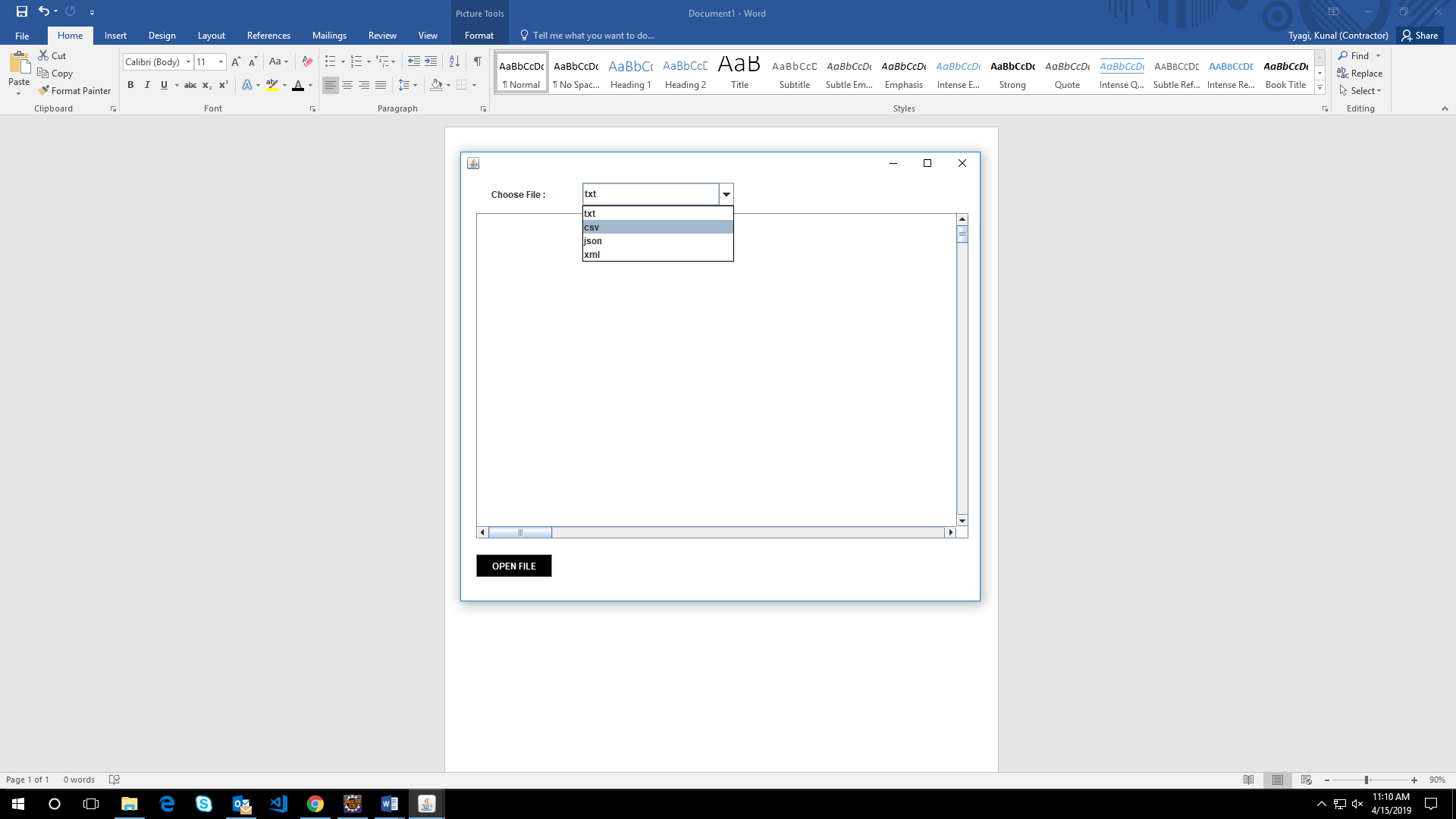




If we click on the option “Read File”, it will open a frame which will ask for the file to be read.



The “Read File” frame contains a drop down for different file formats.



After selecting a particular file format from the drop down and clicking on the “Open File”

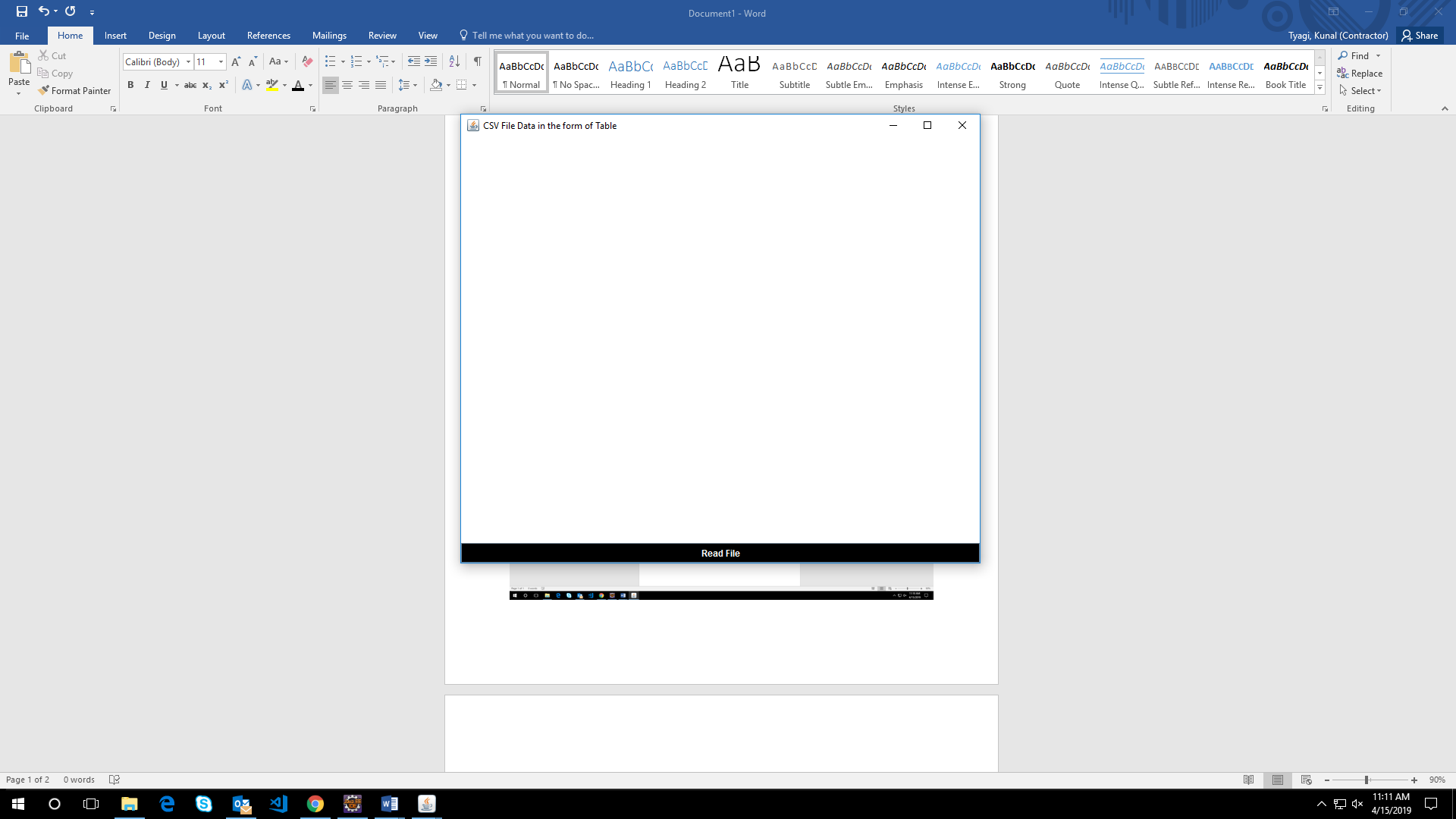
button, a new frame will open which will let the user browse the required file from the file

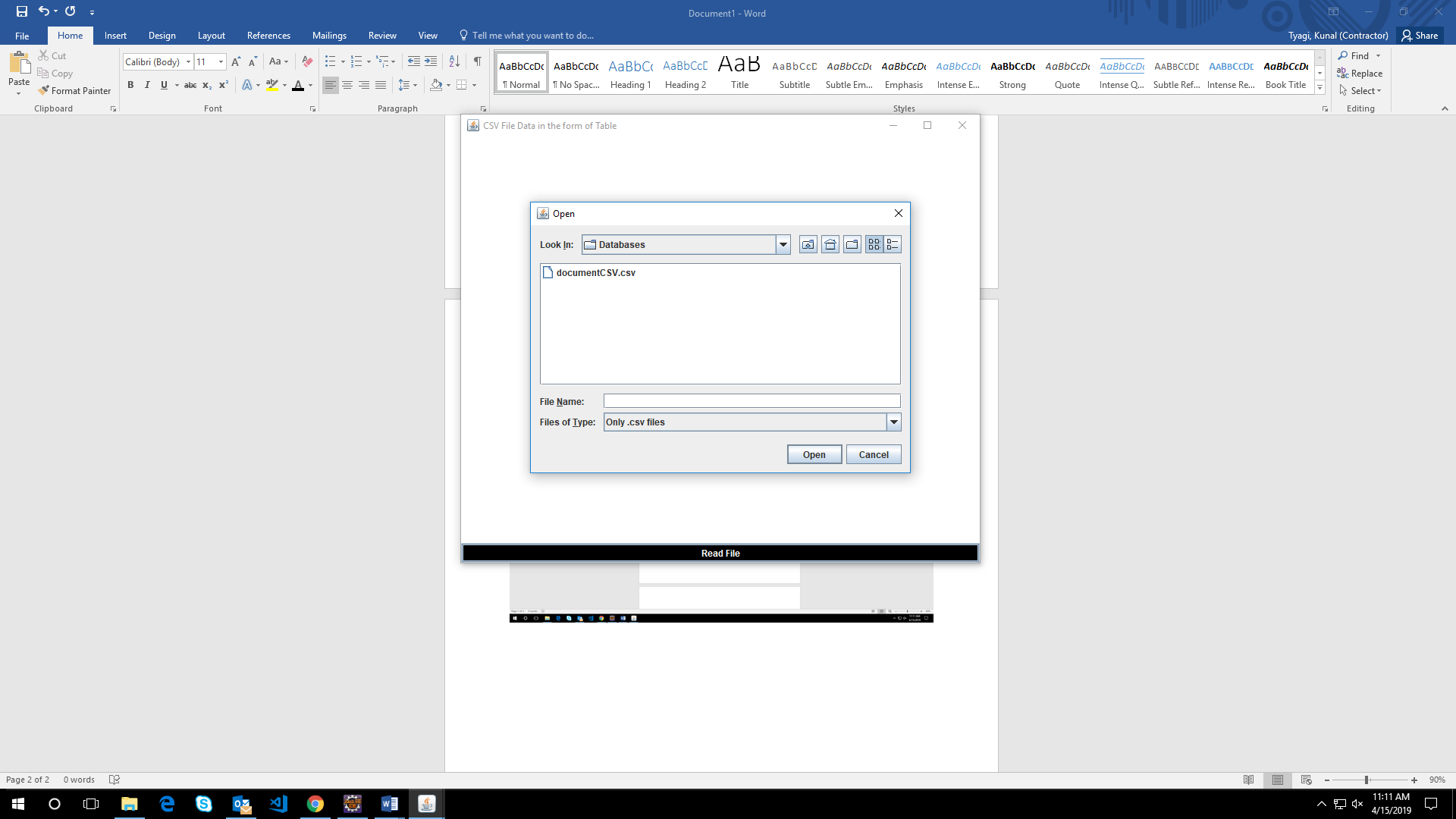
explorer.

It will restrict all the other file formats except the one selected.

Suppose we have selected the CSV file format. Then, after clicking “Open File” button the new

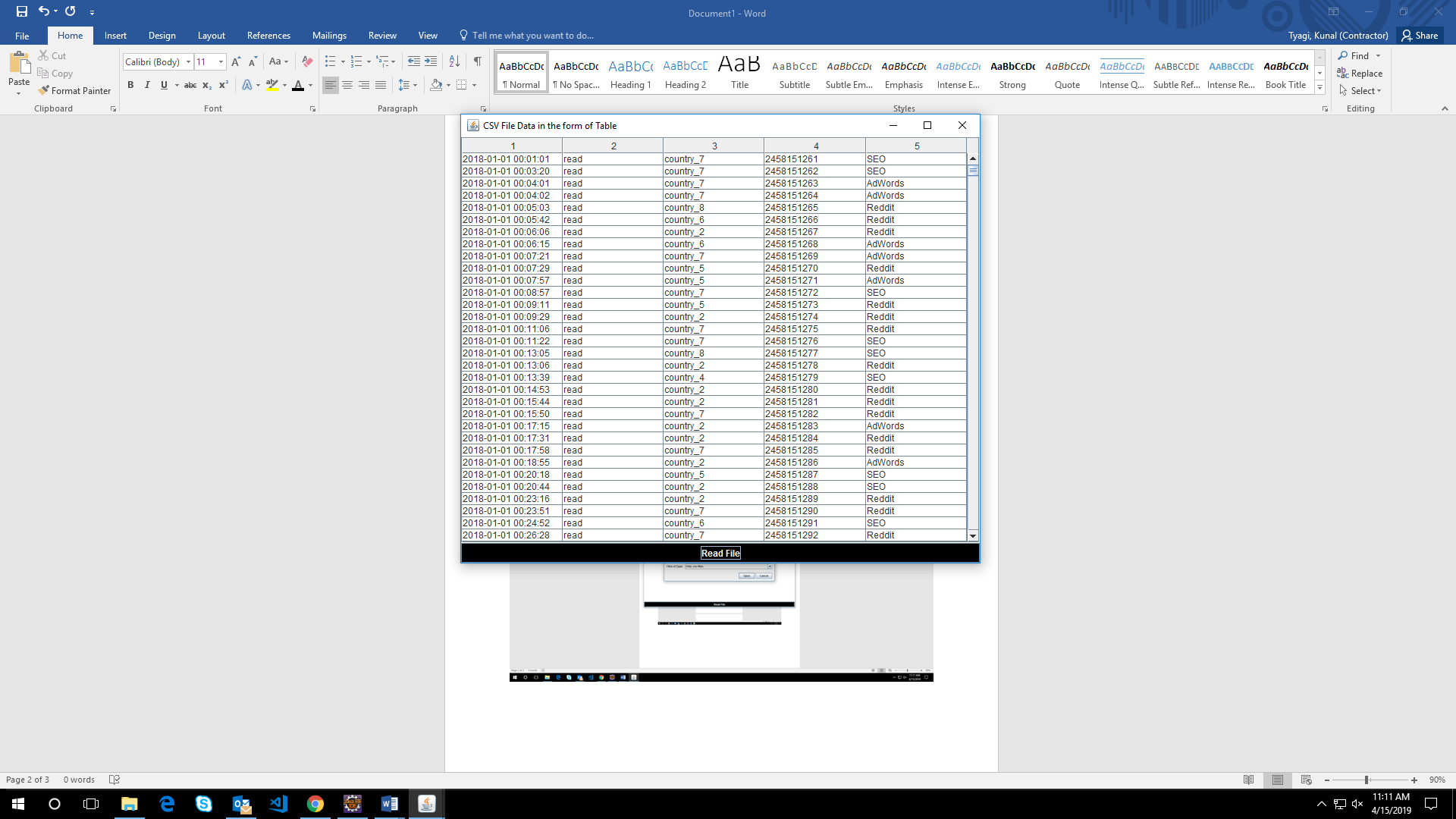
frame will ask the user to read the file.





After selecting the CSV File and clicking on the “Open” button, DOM parser reads the CSV

File, parses the file and displays the data in the form of Table (JTable) in a new frame as shown in the figure below



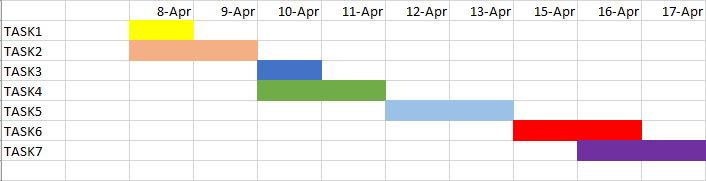
***Chapter-5***

***Appendix and***

***Conclusion***

**APPENDIX**

**A. Project Plan as Gantt chart or WBS**

  
  
  
**TASK1**: - Finding Research Papers relevant to the topic.

**TASK2**: - Implementation of the research paper

**TASK3**: - Analysis of data parsing methods in JAVA

**TASK4:** - Implementation of front end using GUI

**TASK5**: - Finding more relevant research papers and implementation

                 Of DOM parser.

**TASK6**: - Integrating the different file formats to a single GUI

application

**TASK7**: - Writing Final Report

# **Conclusion**

As a result of reading the different data frame formats (.json, .csv, .xml, .txt) and then parsing of that data and displaying the result, the user can read or access different format of files and can even select and modify data from a large set.

The user can select a particular attribute from a large set of data and modify accordingly and can even perform the join operations on the two tables.

The access is not granted to any unauthorized person. Only the admin can access and modify the files and perform the operations. Hence maintaining the security.

The user can also make use of the utilities provided i.e. Notepad, Calculator and Web Browser

**Reference**

[1] A. Abouzied, D. J. Abadi, and A. Silberschatz. Invisible loading: Access-driven data transfer from raw files into database systems. In EDBT, 2013

[2] A. Alexandrov et al. The Stratosphere platform for big data analytics. The VLDB Journal, 23(6):939–964, Dec. 2014.

[3] G. J. Bex, F. Neven, and S. Vansummeren. Inferring XML schema definitions from XML data. In VLDB, 2007.

[4] J. Hegewald, F. Naumann, and M. Weis. XStruct: efficient schema extraction from multiple and large XML documents. In ICDE Workshops, 2006.

[5] Spark SQL: Relational Data Processing in Spark Michael Armbrust† , Reynold S. Xin† , Cheng Lian† , Yin Huai† , Davies Liu† , Joseph K. Bradley† , Xiangrui Meng† , Tomer Kaftan‡ , Michael J. Franklin†‡, Ali Ghodsi† , Matei Zaharia†∗ †Databricks Inc. ∗MIT CSAIL ‡AMPLab, UC Berkeley

[6] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica. GraphX: Graph processing in a distributed dataflow framework. In OSDI, 2014.

[7] Different ways of Reading a text file in Java <https://www.geeksforgeeks.org/different-ways-reading-text-file-java/>