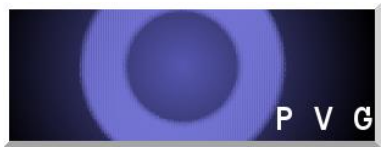

Software Requirements Specification

Version 3-2014-05-14

For

PlasmaGraph

Daniel E. Quintini

Computer Science Undergraduate Program

Gerardo A. Navas

Computer Science Undergraduate Program

Polytechnic University of Puerto Rico

Electrical & Computer Engineering and Computer Science Department

May 14, 2014

Revision History

Name	Date	Reason For Changes	Version
Daniel E. Quintini	2014/5/1	New requirements needed to be added.	1-2014-05-01
Daniel E. Quintini	2014/5/14	Document was revised by a third party and some errors were found.	2-2014-05-14
Daniel E. Quintini	2014/5/15	The system can't make graphs with spline interpolation for non-injective functions.	2-2014-05-14
Daniel E. Quintini	2014/5/29	Remove TBD from user manual and installation manual references	3-2014-05-14
Daniel E. Quintini	2014/6/4	Added Overview and rearranged section 1	3-2014-05-14

Table of Contents

Table of Figures

List of Tables

1. Introduction

1.1 Purpose

The production of a Software Requirements Specification (SRS) offers the following benefits: "Reduces the development effort, provides a basis for estimating costs and schedules, provides a baseline for validation and verification, facilitate transfer and serve as basis for enhancement" *[1]*.

Audiences for this document are the developers Daniel E. Quintini & Gerardo A. Navas, the client Angel E. Gonzalez-Lizardo, Ph. D. and anyone interested in the product delivered as a result of this specification.

In addition this document is the basis of agreement between the developers Daniel E. Quintini & Gerardo A. Navas and the client Angel E. Gonzalez-Lizardo, Ph. D.

1.2 Product Scope

The plasma research team at the Polytechnic University of Puerto Rico is currently undergoing research in several subjects under the topic of plasma. A big part of this research is done using a mirror and cusp plasma machine whose output is interpreted using MATLAB® *[2]* the problem is that even though MATLAB is able to produce visualizations from the experiments performed by the team, it lacks from a proper graphical user interface to do so. This results in an inefficient process where only people with MATLAB programming skills are able to produce graphs from the mirrors and cusp plasma machine output.

The solution is a software product named "PlasmaGraph". It is a tool capable of analyzing the team's data and produces a visual representation of the numbers. In essence, a program with a graphical user interface that makes chart graphs using as input a MATLAB LEVEL 5 MAT-File

(MAT) with a set of variables and values. This input will be called "data file" ***[specifics on data files are discussed in section 2.2.1.1]***.

Because there are many ways of making a graph with the same data; PlasmaGraph is going to collect additional information from the data analyst so it can validate the data file (MAT) and make the graph. This information can be stored by the system as a template file (TEM) so the data analyst is able to apply common settings to certain data ***[this is explained in more depth on section 2.2.2.1]***. PlasmaGraph also provides the option to save the produced graph as an image in portable network graphics format (PNG). In short, this product does the following things:

- Detect invalid values in data files (MAT) and inform to the data analyst ***[see section 2.3.1]*** that there is a problem with the input file provided.
- Provide a graphical user interface that simplifies making graph charts. This means that the data analyst can select the following functions with this user interface:
    - Import a data file (MAT).
    - Produce a graph chart for a set of X and Y values retrieved from a data file (MAT).
    - Produce a graph chart with a linear interpolation for a set of X and Y values retrieved from a data file (MAT).
    - Produce a graph chart with a quadratic interpolation for a set of X and Y values retrieved from a data file (MAT).
    - Produce a graph chart with a spline interpolation for a set of X and Y values retrieved from a data file (MAT). The spline interpolation only works for injective functions.
    - Produce any of the graph charts described above but grouping (X, Y) points using a third set of values. This third set cannot contain more than twenty different values.
    - Determine the confidence level for the data represented in the graph produced.
    - Edit the graph's title and axis labels.
    - Save the graph chart produced as a portable network graphic image (PNG).
    - Save template files (TEM).

PlasmaGraph can highly improve productivity at the laboratory. Team members will no longer have to worry with coding new experiment results into MATLAB plotting functions or finding ways to save the graphs produced by MATLAB in other formats like *.jpeg or *.png. In addition, thanks to template files (TEM), the data analyst only needs to tell the system once how to handle the data. In this way, the user only has to select a template file to apply to the data instead of going through the whole process of telling the system how to make the graph. PlasmaGraph can also be modified to accomplish new requirements that benefit the research team. This can be done by anyone with programming skills in both Java and MATLAB by using the Development Kit ***[see section 2.1.1.2]*** delivered with this product.

1.3 Definitions, acronyms and abbreviations

Term	Definition
Alternating Current	Electric current of which magnitude and direction vary.
Array	Systematic arrangement of objects, usually in rows and columns.
Bit	The basic unit of information in computing and digital communications.
Buffer	Region of a physical memory storage used to temporarily store data.
Byte	Unit of digital information that consists of eight bits.
Codomain	In mathematics, the codomain or target set of a function is the set Y into which all of the output of the function is constrained to fall.
Data Type	Classification identifying one of various types of data, such as real, integer or Boolean, that determines the possible values for that type.
Distributed Version Control System	Piece of software that keeps track of software revisions and allows many developers to work on a given project without requiring that they maintain a connection to a common network.
Domain	In mathematics, the domain of definition or simply the domain of a function is the set of "input" or argument values for which the function is defined.
End User	Person who uses a product.
File System	System used to control how data is stored and retrieved from a computer.
GHz	Unit of alternating current (AC) equal to one

Term	Definition
	thousand million hertz (1,000,000,000 Hz).
HDD	A hard disk drive (HDD) is a data storage device used for storing and retrieving digital information.
Hertz	Unit of frequency.
Injective Function	In mathematics, an injective function or injection or one-to-one function is a function that preserves distinctness: it never maps distinct elements of its domain to the same element of its codomain.
Java Package	In the Java programming language, a package is a mechanism for grouping parts of the source code that are related between each other.
KB	The kilobyte (KB) is a multiple of the unit byte for digital information. Its value is 1024 bytes.
Mb	The Megabyte (Mb) is a multiple of the unit byte for digital information. Its value is 1024 * (2^10) bytes.
Microprocessors	Hardware within a computer that carries out the instructions of a computer program.
Personal Computer	General-purpose computer, whose size, capabilities and original sale price makes it useful for individuals, and which is intended to be operated directly by an end-user with no intervening computer operator.
Pixel	Smallest controllable element of a picture represented on the screen.
Plain Text File	Computer file whose contents can be red without much processing as opposed to “binary files” in which some portions must be interpreted as

Term	Definition
	binary objects.
Programming Language	Artificial language designed to communicate instructions to a machine, particularly a computer.
Pros and Cons	The phrase "pros and cons" means "advantages and disadvantages". It derives from the Latin prefixes "pro-", meaning on behalf of or for, and contra-, meaning against.
RAM	Random-access memory (RAM) is a form of computer data storage.
Refresh Rate	Number of times in a second that display hardware updates its buffer.
Screen Resolution	Number of distinct pixels in each dimension that can be displayed in a computer monitor.
Software	Non-tangible component of computers
Variable	Symbolic name associated with a value and whose associated value may be changed
Virtual Machine	Software-based emulation of a computer

Table 1: Definitions

1.4 References

[1] Colizeeum Technology, "Colizeeum Technology," [Online]. Available: http://www.colizeeumtechnology.com/OurMethodology/SRS.aspx. [Accessed 21 01 2014].

[2] The MathWorks, Inc., "MathWorks," [Online]. Available: http://www.mathworks.com/products/matlab/. [Accessed 21 1 2014].

[3] Tableau Software Inc, "Tableau Software," [Online]. Available: http://www.tableausoftware.com/. [Accessed 21 1 2014].

[4] "Visual.ly," [Online]. Available: http://visual.ly. [Accessed 21 1 2014].

[5] Oracle Corporation, "Learn About Java Technology," [Online]. Available: https://www.java.com/en/about/. [Accessed 21 1 2014].

[6] Oracle Corporation, "Java SE Runtime Environment 7 Downloads," [Online]. Available: http://www.oracle.com/technetwork/java/javase/downloads/java-se-jre-7-download-432155.html. [Accessed 13 5 2014].

[7] opencsv, "Frequently Asked Questions," 28 07 2011. [Online]. Available: http://opencsv.sourceforge.net/#what-is-opencsv. [Accessed 13 5 2014].

[8] jfree.org, "JFreeChart 1.0.17 API Documentation," [Online]. Available: http://www.jfree.org/jfreechart/api/javadoc/index.html. [Accessed 21 1 2014].

[9] The Apache Software Foundation., "Commons Lang," 09 4 2014. [Online]. Available: http://commons.apache.org/proper/commons-lang/. [Accessed 13 5 2014].

[10] The Apache Software Foundation., "Commons Math: The Apache Commons Mathematics Library," 03 11 2013. [Online]. Available: http://commons.apache.org/proper/commons-math/. [Accessed 13 5 2014].

[11] The Apache Software Foundation., "Commons Validator," 28 2 2014. [Online]. Available: http://commons.apache.org/proper/commons-validator/. [Accessed 13 5 2014].

[12] Google, "hamcrest," [Online]. Available: http://code.google.com/p/hamcrest/. [Accessed 13 5 2014].

[13] Junit, "JUnit 4," [Online]. Available: http://junit.org/. [Accessed 13 5 2014].

[14] W. Gradkowski, "JMatIO - Matlab's MAT-file I/O in JAVA," 7 10 2007. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/10759-jmatio-matlab-s-mat-file-i-o-in-java. [Accessed 13 5 2014].

[15] Oracle Corporation, "Package javax.swing," [Online]. Available: http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html. [Accessed 24 1 2014].

[16] Oracle Corporation, "Java™ Platform, Standard Edition 7," [Online]. Available: http://docs.oracle.com/javase/7/docs/api/.

[17] Oracle Corporation, "Initial Heap Size and Maximum Heap Size Changed for Parallel Garbage Collector," [Online]. Available: http://docs.oracle.com/javase/7/docs/technotes/guides/vm/gc-ergonomics.html. [Accessed 21 1 2014].

[18] Oracle Corporation, "JDK 7 and JRE 7 Installation Guide," [Online]. Available: http://docs.oracle.com/javase/7/docs/webnotes/install. [Accessed 21 1 2014].

[19] The MathWorks, Inc., "MATLAB MAT-File Format R2013b Figure 1-1," [Online]. Available: http://www.mathworks.com/help/pdf_doc/matlab/matfile_format.pdf. [Accessed 21 1 2014].

[20] The MathWorks Inc., "Cell Arrays," [Online]. Available: http://www.mathworks.com/help/matlab/cell-arrays.html. [Accessed 24 1 2014].

[21] The MathWorks Inc., "Create Variables," [Online]. Available: http://www.mathworks.com/help/matlab/matlab_prog/create-variables.html. [Accessed 24 1 2014].

[22] The MathWorks Inc., "Data Types," [Online]. Available: http://www.mathworks.com/help/matlab/data-types_data-types.html. [Accessed 24 1 2014].

[23] The MathWorks Inc., "NaN," [Online]. Available: http://www.mathworks.com/help/matlab/ref/nan.html. [Accessed 24 1 2014].

[24] Concordia University Human Resources Department, "Computer Skills: Levels of Proficiency," [Online]. Available: http://web2.concordia.ca/hr/eed/online_guides/pdf/hr_guide_CompSkills.pdf. [Accessed 24 1 2014].

[25] jfree.org, "Welcome to JFreeChart!," [Online]. Available: http://www.jfree.org/jfreechart/.

[Accessed 24 1 2014].

[26] Oracle Corporation, "Code Conventions for the Java Programming Language," [Online]. Available: http://www.oracle.com/technetwork/java/codeconv-138413.html. [Accessed 21 1 2014].

[27] IEEE, "IEEE Recommended Practice for Software Requirements Specifications," 1998.

[28] IEEE, "IEEE Recommended Practice for Software Design Descriptions," 1998.

[29] IEEE, "IEEE Standard for Software Project Management Plans," 1998.

[30] IEEE, "IEEE Standard for Software Test Documentation," 1998.

[31] "Git FAQ," Linux Kernel Organization, Inc, 9 3 2013. [Online]. Available: https://git.wiki.kernel.org/index.php/GitFaq#What_is_Git.3F. [Accessed 24 1 2014].

[32] Oracle Corporation, "Oracle," [Online]. Available: http://oracle.com/technetwork/java/javase/overview/index.html. [Accessed 21 1 2014].

[33] Opencsv, "Opencsv Project," [Online]. Available: http://opencsv.sourceforge.net. [Accessed 21 1 2014].

[34] Oracle Corporation, "Overview of Java Interfaces," [Online]. Available: http://www.mathworks.com/help/matlab/matlab_external/product-overview.html. [Accessed 21 1 2014].

[35] Oracle Corporation, "Package javax.swing," [Online]. Available: http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html. [Accessed 21 1 2014].

[36] The MathWorks Inc., "double," [Online]. Available: http://www.mathworks.com/help/symbolic/double.html. [Accessed 24 1 2014].

[37] JFree.org, "JFreeChart," [Online]. Available: http://www.jfree.org/jfreechart/api/javadoc/index.html. [Accessed 13 5 2014].

1.5 Overview

The rest of this document is divided in two sections which are overall description and specific requirements. The first describes the general factors that affect PlasmaGraph and its requirements. The second contains the requirements in a level of detail sufficient enough to enable the developers to design and test a system that meets the objectives defined.
The overall description of the product is divided into seven subsections that defines the product's perspective and functions, user characteristics, constrains, design, implementation, user documentation assumptions and dependencies and apportioning of requirements. Meanwhile the specific requirements section spawns five subsections covering external interfaces, functions, performance requirements, design constrains and software systems attributes.

2. Overall Description

2.1 Product Perspective

There are several data visualization software products in the market like Tableau *[3]* and Visua.ly *[4]* but they are expensive and come with too many options which results in a steeper learning curve for the person using the product. PlasmaGraph in the other hand is specialized to meet the client's requirements making it simpler and easier to use. The following table sheds light on the pros and cons of the tools considered and PlasmaGraph.

	Tableau	***Visual.ly***	***PlasmaGraph***
Pros	• *Graphical user interface* • *Data sharing through Tableau server* • *Graphs from *.csv and many other file formats.* • *Ready product*	• *Graphical user interface (beta)* • *Easy to share content* • *Free (single user)* • *Portable (runs on a web browser)* • *Ready product*	• *Graphical user interface* • *No license required* • *Developed with Java to ensure portability. 89% of Desktops in the U.S. Run Java [5]* • *Source code available for future improvements by the client* • *Easy to use (only the options needed by the lab's team)*
Cons	• *Expensive $999-$1,999* • *Lacks portability (windows only)*	• *One account per user* • *GUI still in beta* • *More oriented toward info graphics and sharing rather than scientific research*	• *Product is delivered June 1st*

Table 2: PlasmaGraph vs. Others

2.1.1 System Interfaces

PlasmaGraph is a new addition for the plasma laboratory data collection and analysis system which aims to help with the task of analyzing data produced by the mirror and cusp plasma machine. It runs on the Java Runtime Environment (JRE) to guarantee portability across 89% of desktops *[5]* and it is divided in two modules that work together in order to accomplish the requirements stated in this document. These modules are PlasmaGraph User Interface (PG UI) and PlasmaGraph Development Kit (PG DK) ***[see figure 1]***.

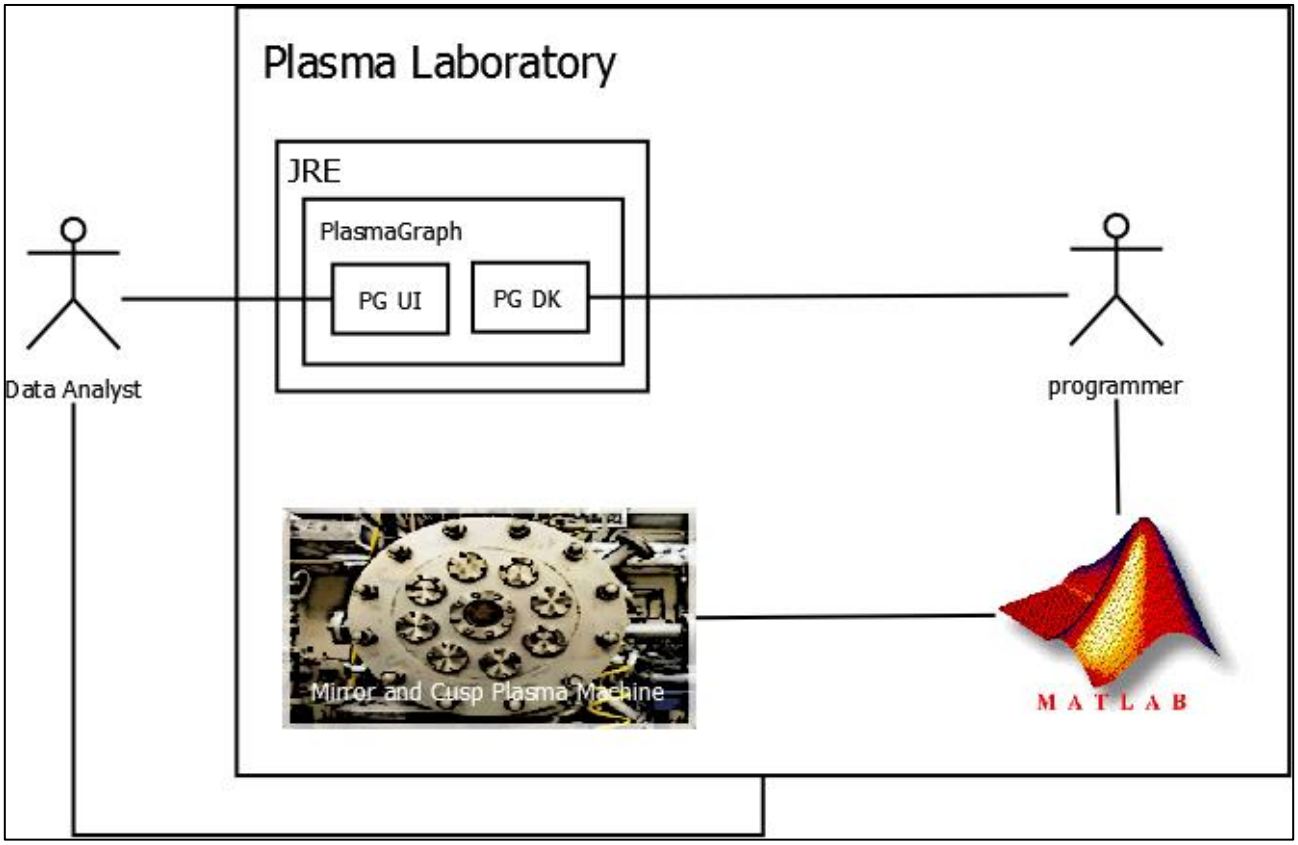

Figure 1: System Interfaces

PlasmaGraph has two active actors; programmers and data analysts. First, the programmer elicits raw data from the mirror and cusp plasma machine and converts it into a file that PlasmaGraph understands ***[see section 2.2.1.1]***. Then, the data analyst can get this file from the plasma laboratory and into his personal computer.

Using PG UI ***[see section 2.1.1.1]*** the data analyst can order the system to retrieve the converted file from his or her computer and generate a graph with the data contained inside the selected file. The programmer can also communicate with PlasmaGraph through PG DK ***[see section 2.1.1.2]*** which gives its user's the ability to add, remove, or update any functionality delivered by this software. This includes customizing how the program interprets data files.

2.1.1.1 PlasmaGraph User Interface (PG UI)

Its primary function is to enable data analysts with no programming skills to produce graphs from data files created by programmers. This is accomplished by presenting the user with a series of windows, buttons, and input fields that he or she can use to tell PlasmaGraph what to graph. In short; a graphical user interface (GUI) that can be used to accomplish every functional requirement of PlasmaGraph ***[see section 2.2.1].***

2.1.1.2 PlasmaGraph Development Kit (PG DK)

It encapsulates the entire program that is PlasmaGraph into a Java package that can be manipulated by a programmer. This is useful for adding, updating, or removing functionality from the product. But for now, it will be used to configure how the system is going to interpret data files.

The development kit delivered with the product includes the following software items:

- PlasmaGraph Java Package.
- Eclipse Standard/SDK Version: Kepler Service Release 1.
- Java Runtime Environment 7.

2.1.2 Software interfaces

Name	Description	Mnemonic	Version	Source
Java Runtime Environment	*Standard environment used to develop applications written in the Java programming language. It is used for both producing and executing PlasmaGraph's source code*	JRE 7	JRE 7u51	[6]
Opencsv	A comma-separated value parser library for Java programming language. It is used by PlasmaGraph's programmers so they can integrate comma separated value (CSV) files into the code with more ease.	opencsv	2.3	[7]
JFreeChart	A free chart library for Java that can generate a wide variety of charts for use in applications, applets and servlets. It is used by the programmers to produce the code that draws the graph charts which are the visual representations of the data provided by the data analyst	jfree	1.0.17	[8]
commons-lang3-3.2.1	Lang provides a host of helper utilities for the java.lang API, notably String manipulation methods, basic numerical methods, object reflection, concurrency, creation and serialization and System properties. Additionally it contains basic enhancements to	java.lang	3.2.1	[9]

Name	Description	Mnemonic	Version	Source
	java.util.Date and a series of utilities dedicated to help with building methods, such as hashCode, toString and equals.			
commons-math3-3.2	Commons Math is a library of lightweight, self-contained mathematics and statistics components addressing the most common problems not available in the Java programming language or Commons Lang.	Commons.math3	3.2	[10]
commons-validator-1.4.0	A common issue when receiving data either electronically or from user input is verifying the integrity of the data. This work is repetitive and becomes even more complicated when different sets of validation rules need to be applied to the same set of data based on locale. Error messages may also vary by locale. This package addresses some of these issues to speed development and maintenance of validation rules.		1.4	[11]
Hamcrest-core	Provides a library of matcher objects (also known as constraints or predicates) allowing 'match' rules to be defined declaratively, to be used in other frameworks. Typical scenarios include testing frameworks, mocking	hamcrest	1.3	[12]

Name	Description	Mnemonic	Version	Source
	libraries and UI validation rules.			
junit	JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.	junit	4	[13]
jmatio	JMatIO is a JAVA library to read/write/manipulate with Matlab's binary MAT-files.	jmatio	1	[14]

Table 3: Software interfaces

PlasmaGraph is compatible with any system that can run a Java Runtime Environment; which means that it doesn't depend on a specific operating system. It also can interact with MATLAB directly through the development kit described in section 2.1.1.2.

2.1.3 User Interfaces

As stated in section 2.1.1.1, PlasmaGraph has a graphical user interface which can be used to accomplish all the functional requirements defined by this document in section 2.2.1. It is made using Swing *[15]* and its components are described in the following sub-section:

2.1.3.1 Graph Pane

This is the portion of the program where the graph is displayed. It can also be used to save the graph in the user's computer ***[see figure 2]***.

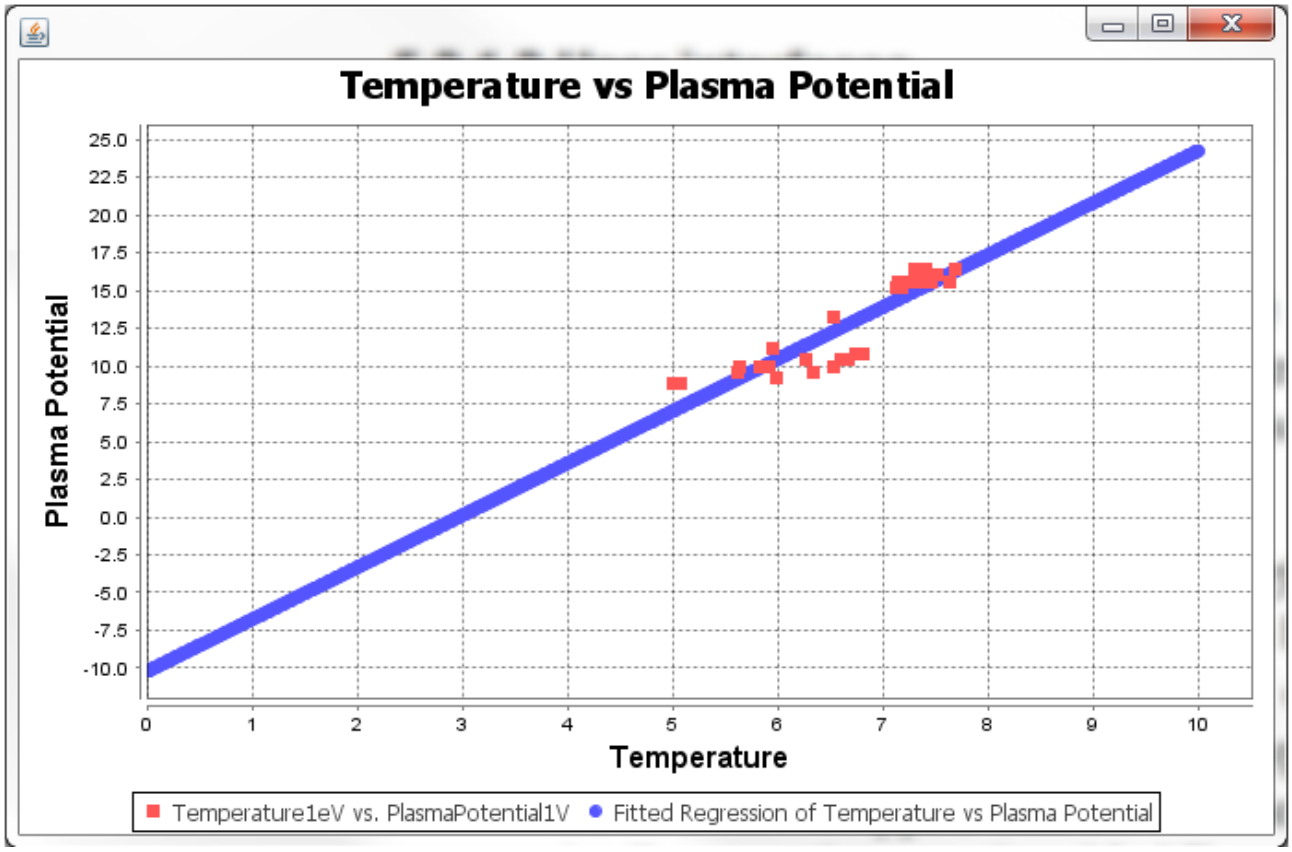

Figure 2: Graph Pane

2.1.3.2 Tools Pane

This is where the user will do most of the input. The tools pane is used to import data or template files into the system, save template files, view data files, set the graphs type, graph title and axis's labels, and determine which pair of data columns is to be graphed and if the values should be grouped using a third column as reference. In addition, the user can order the system to produce the graph using one of three different kinds of interpolations (linear, quadratic, or spline) and whether it should scan for outliers before producing the graph ***[see figure 3]***.

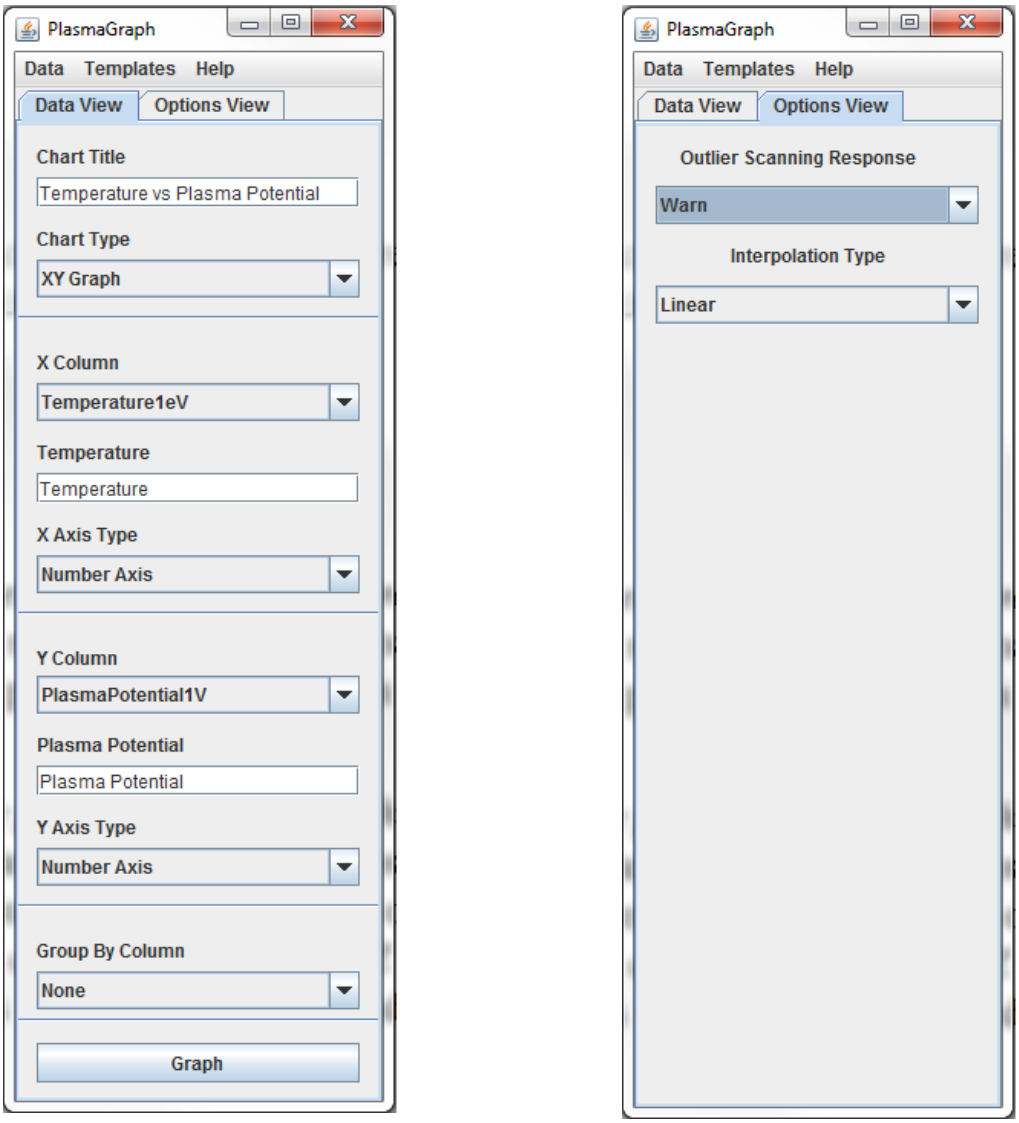

Figure 3: Tools Pane

2.1.3.3 File Chooser

It is used to navigate and select the file system of the computer in which the java runtime environment is executing PlasmaGraph ***[see figure 4]***.

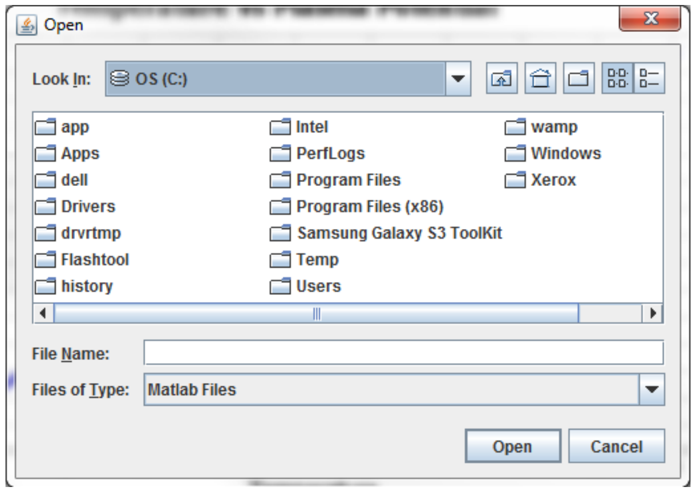

Figure 4: File Chooser

2.1.3.4 Data Pane

This component is used to display the contents of a data file in a way that is readable by humans ***[see figure 5]***.

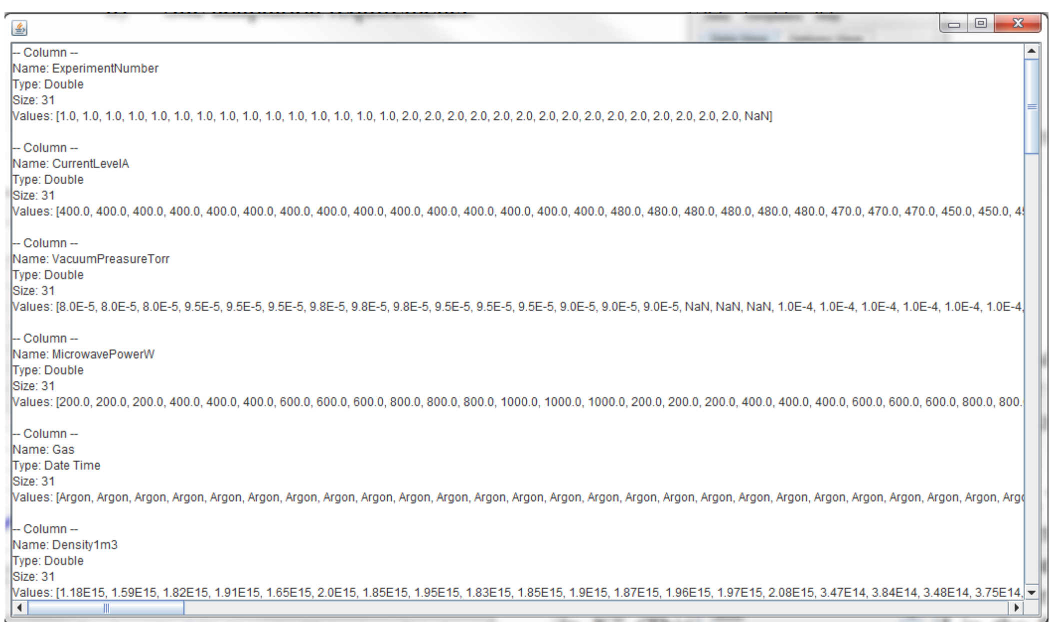

Figure 5: Data Pane

2.1.3.5 Options Pane

They are windows for transmitting messages to the user and in some cases elicit an option from him or her ***[see figure 6]***.

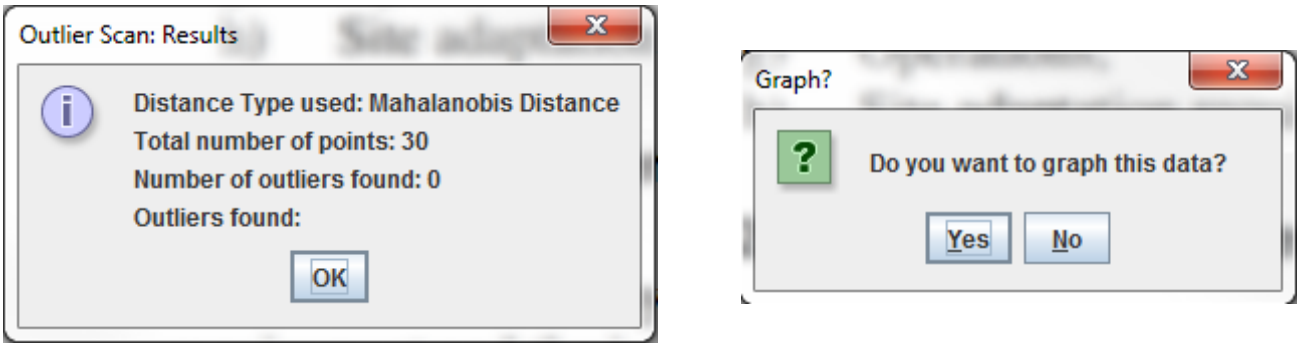

Figure 6: Options Pane

2.1.4 Hardware Interfaces

Because this product is executed under Java Runtime Environment, this particular kind of interface (hardware) is entirely delivered by the programming language itself *[16]*.

2.1.5 Communications Interfaces

PlasmaGraph doesn't make use of the internet or any other network.

2.1.6 Memory Constraints

Because PlasmaGraph runs in the Java Runtime Environment (JRE), the actual memory limit will be established by this virtual machine. According to Oracle's documentation, JRE determines a default initial memory size of "1/64th of the machine's physical memory on the machine or some reasonable minimum… [and maximum size]... 1/4th of the physical memory or 1GB" [17].

2.1.7 Site Adaptation Requirements

To be able to use this product's user interface, the data analyst has to install Oracle's Java Runtime Environment version 7 [18] in any personal computer that meets the following specifications:

- 15'' Monitor with a resolution of 1200x700 and refresh rate of 30Hz
- A standard alphanumeric American (QWERTY) keyboard
- Two 3.20Ghz microprocessors
- 500Mb of available HDD
- 512Mb of RAM

2.2 Product Functions

PlasmaGraph is a software tool designed for optimizing the process of gathering and analyzing output data produced by the mirror and cusp plasma machine at the PUPR by letting the data analyst generate graphs with this data. In other words, it uses data files produced by the plasma machine in order to generate graphs that can be saved as images in the data analyst's computer. The following sub-sections will explain in detail all the functional and non-functional requirements met by PlasmaGraph.

2.2.1 Functional Requirements

2.2.1.1 Import Data (FR-01)

Before the data analyst can make a graph, it needs to tell someone at the plasma laboratory with MATLAB programming skills to provide the data file produced by the plasma machine. At the time of delivery, PlasmaGraph is only able to import files that meet the following format and structure:

a. The file must be formatted as a binary MATLAB Level 5 MAT-File [19].
b. All objects contained in the file must be MATLAB arrays of 1 column and any amount of rows. Every array needs to have the same amount of rows [20].
c. All arrays contained in the file must be populated with at least one MATLAB variable or constant and all variables within an array have to be of the same data type [21].
d. Variables or constants contained by arrays must be of the data type double, char or cell [22].
e. The file size must be 56300kb or less.

Once the data analyst has selected the file, he or she can tell PlasmaGraph where to import the file.

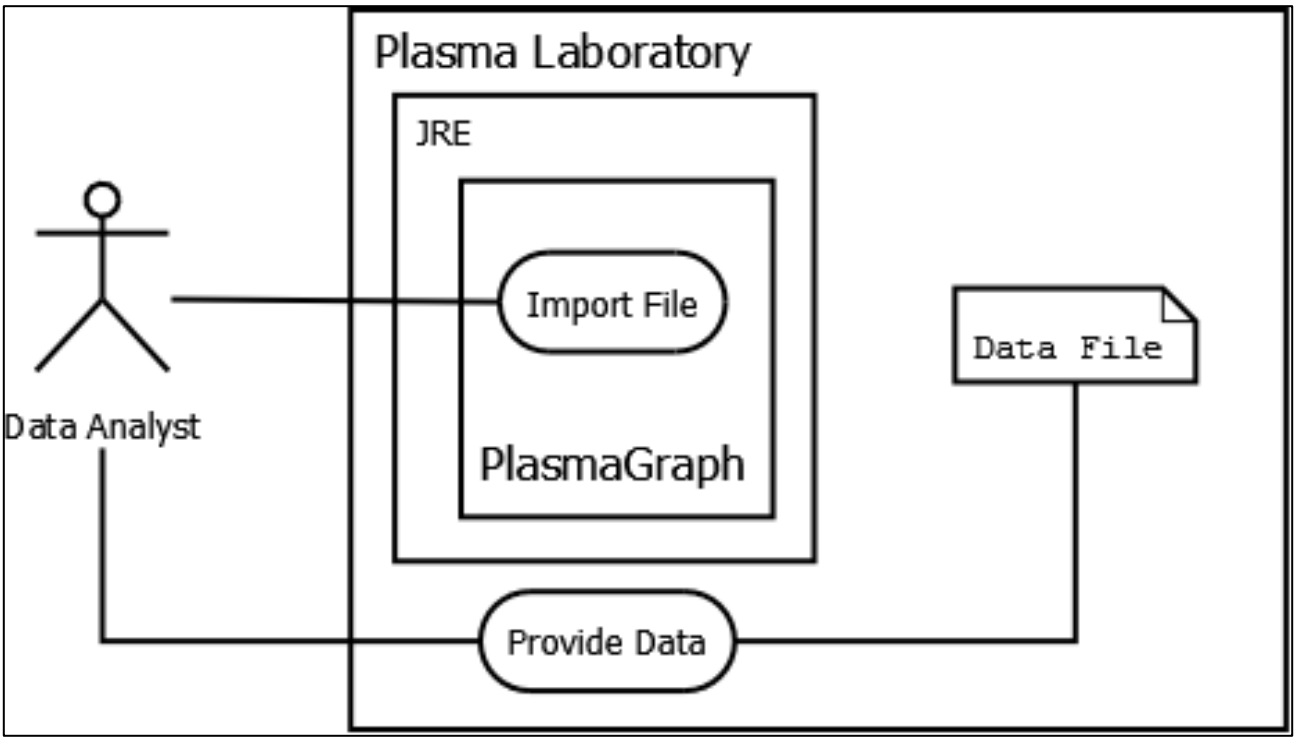

Figure 7: FR-01

2.2.1.2 Validate Data (FR-02)

Sometimes, the mirror and cusp plasma machine produces output were one or more of the measurements scheduled didn't take place. In this case, the output file should have a NaN value [23] in the place where the scheduled measurement was supposed to be.

Once the data analyst imports the data file and tells PlasmaGraph which of its variables corresponds to the X axis and which to the Y axis, the system will proceed to check every value within those two variables. Because PlasmaGraph uses pairs of (X,Y) values to draw the graph, if there's a NaN at some point, PlasmaGraph will eliminate the entire row. That is, when a NaN is found at some point in the X axis the system removes that value in X and its corresponding value in the Y axis even if the last one isn't NaN. The same applies when NaN is found at some point in the Y axis.

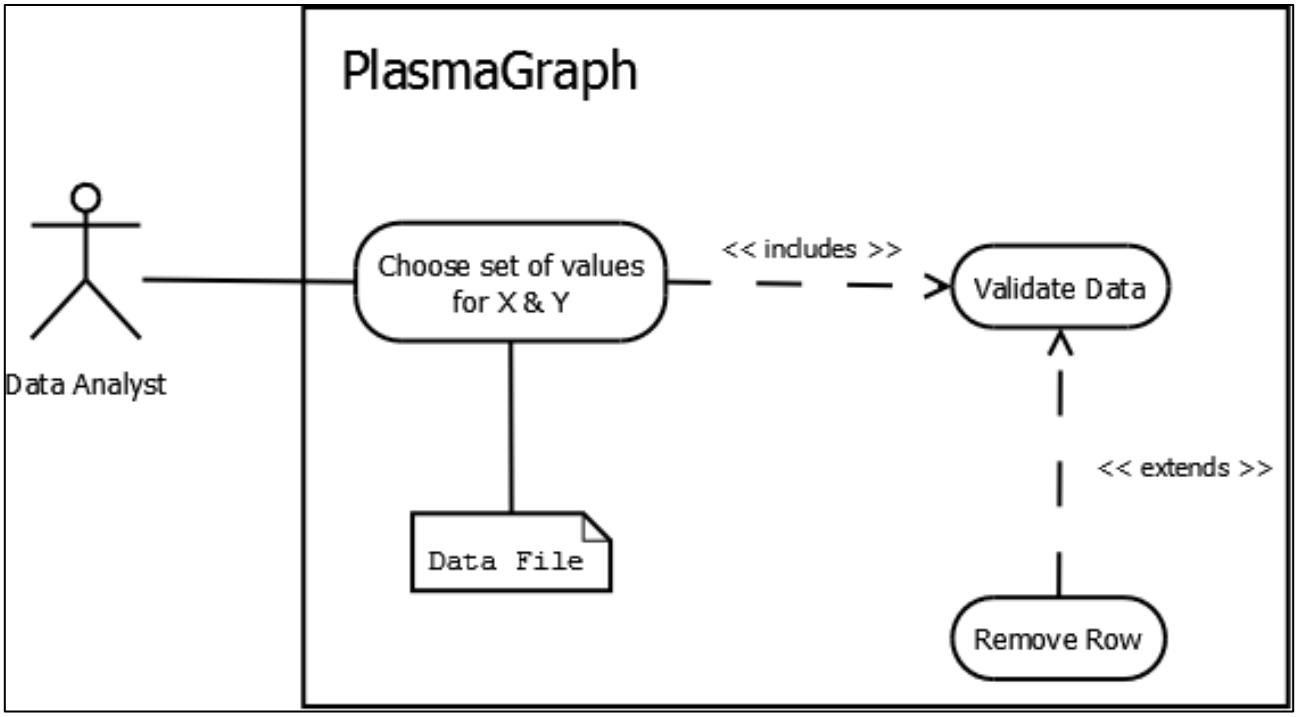

Figure 8: FR-02

It should be noted that removing rows from a chosen set of X and Y values won't remove anything from the original data file imported.

2.2.1.3 Choose Graph Options (FR-03)

Before this system can create a graph chart with the data file imported, the data analyst needs to tell PlasmaGraph the information that results from answering the following questions:

a. Which variable from the data file corresponds to the X axis of the graph? *
b. Which variable from the data file corresponds to the Y axis of the graph? *
c. Which type of scale shall be used in the X axis of the graph, numerical or logarithmic?
d. Which type of scale shall be used in the Y axis of the graph, numerical or logarithmic?
e. What is the title of the graph?
f. What kind of data is represented in the X axis of the graph?
g. What kind of data is represented in the Y axis of the graph?
h. Which variable from the data file, if any, shall be used to group points defined by any (X,Y) coordinate?
i. Should the system warn its user if there're any outliers in the graph?
j. Should the system eliminate outliers from the graph?
k. Should the points in the graph be interpolated? If the answer is yes, which of this kind of interpolation shall the system use: linear, quadratic, or spline?

Note: the above items that end with a * are required for generating the graph.

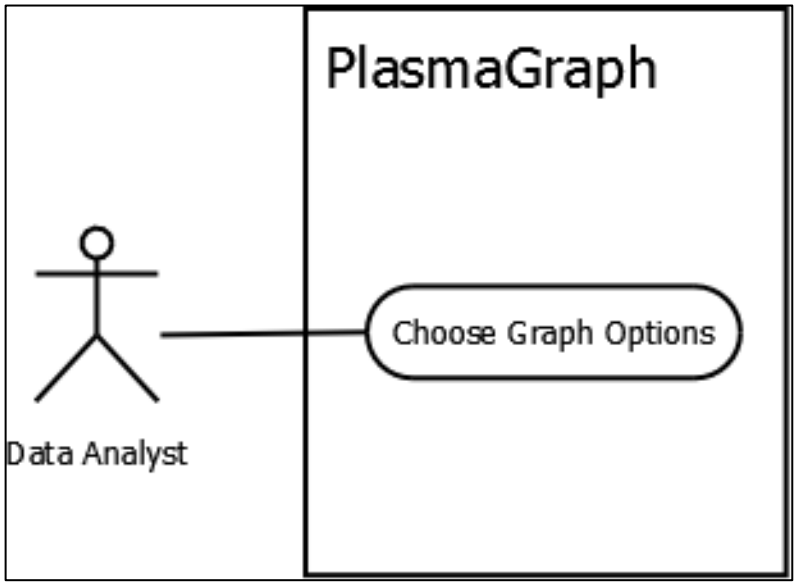

Figure 9: FR-03

2.2.1.4 Create Graph (FR-04)

To produce a graph chart, the data analyst needs to import a data file into the system and choose at least the minimal options needed to produce a graph which are described in section 2.2.1.3 of this document.

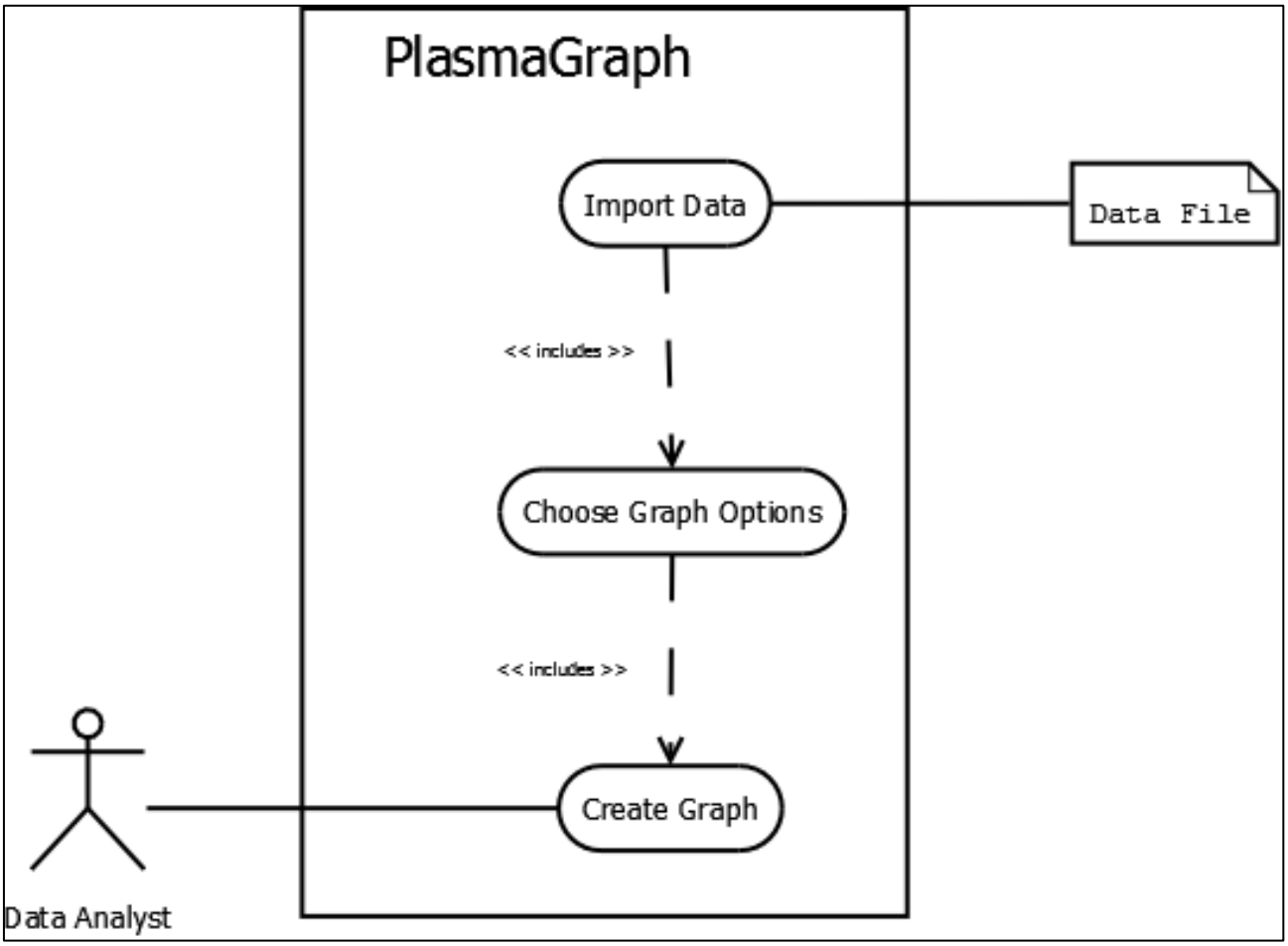

Figure 10: FR-04

2.2.1.5 Save Graph (FR-05)

Any graph chart produced by PlasmaGraph can be exported to the data analyst's computer system as an image in PNG format. To accomplish this, PlasmaGraph tells the data analyst to choose a directory on which to save the file and the JRE will take care of saving the final image of the graph.

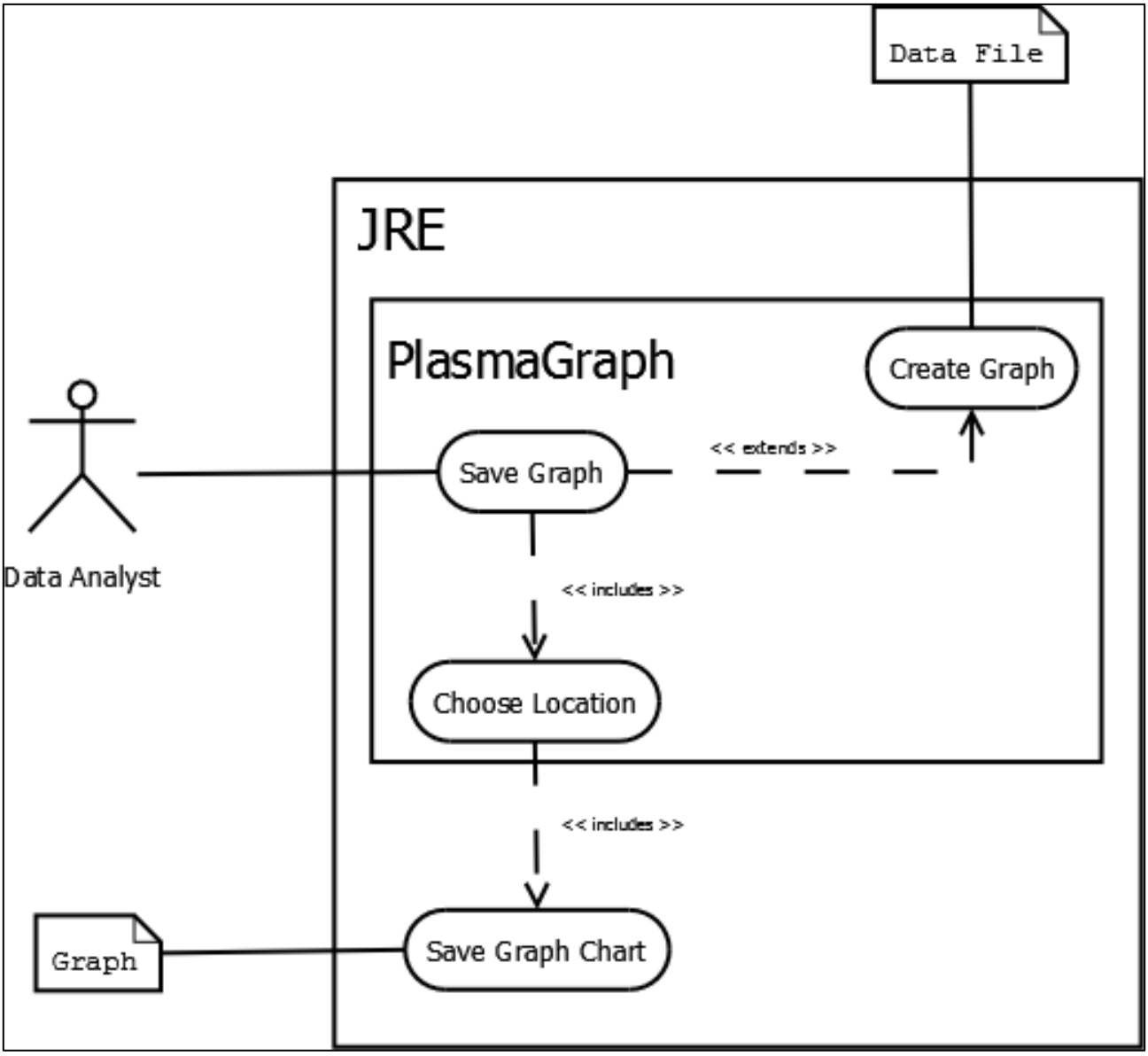

Figure 11: FR-05

2.2.2 Non-Functional Requirements

2.2.2.1 Save Template (NR-01)

As mentioned in section 1.2, PlasmaGraph can also create a file with all the values for the set of options chosen by the data analyst. As soon as the system creates the file, it will ask the where to place the file and finally, it proceeds to save the file.

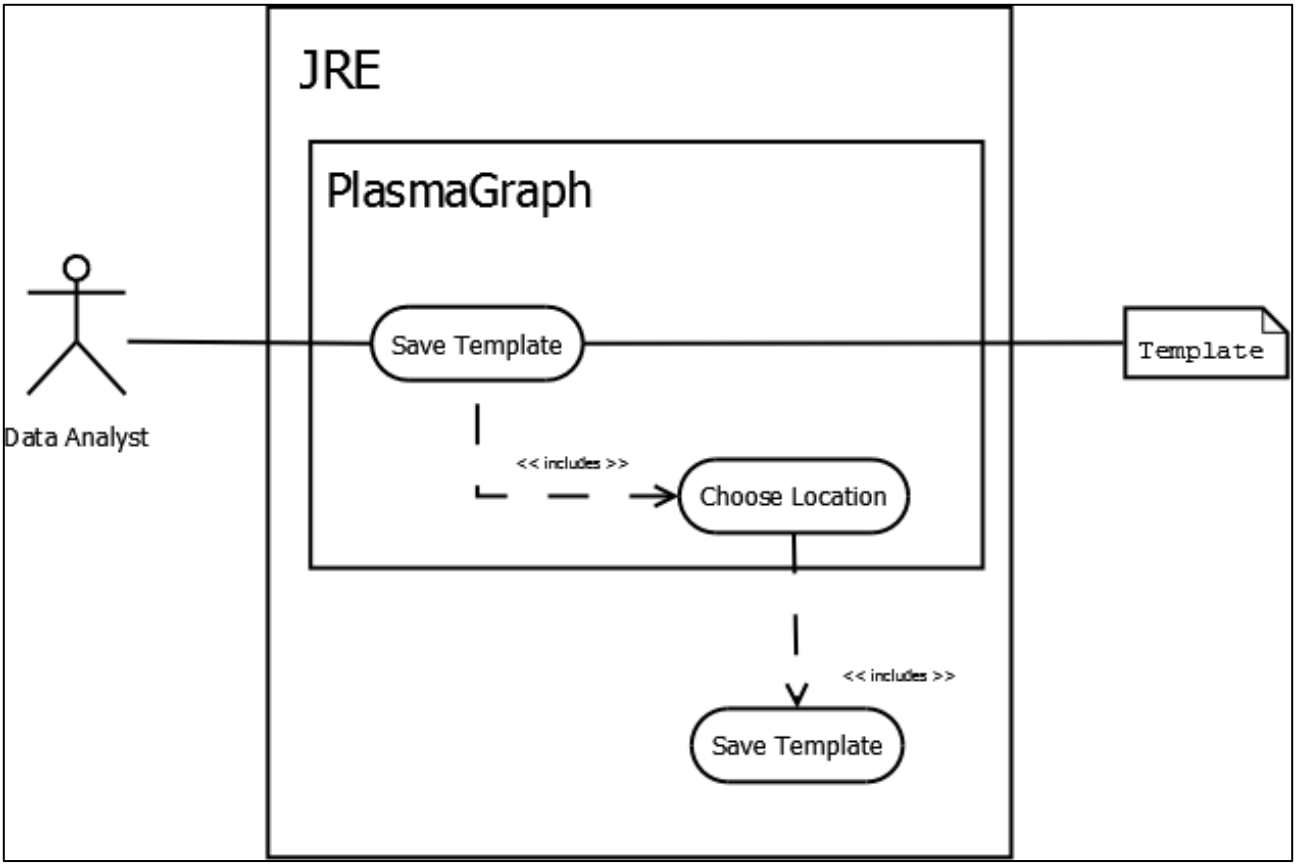

Figure 12: NR-01

2.2.2.2 Import Template (NR-02)

In order to import a template file, the data analyst needs to tell PlasmaGraph where the file is. Once this information has been identified, JRE will proceed to import the file to PlasmaGraph who is responsible for interpreting the file and setting all graph options accordingly.

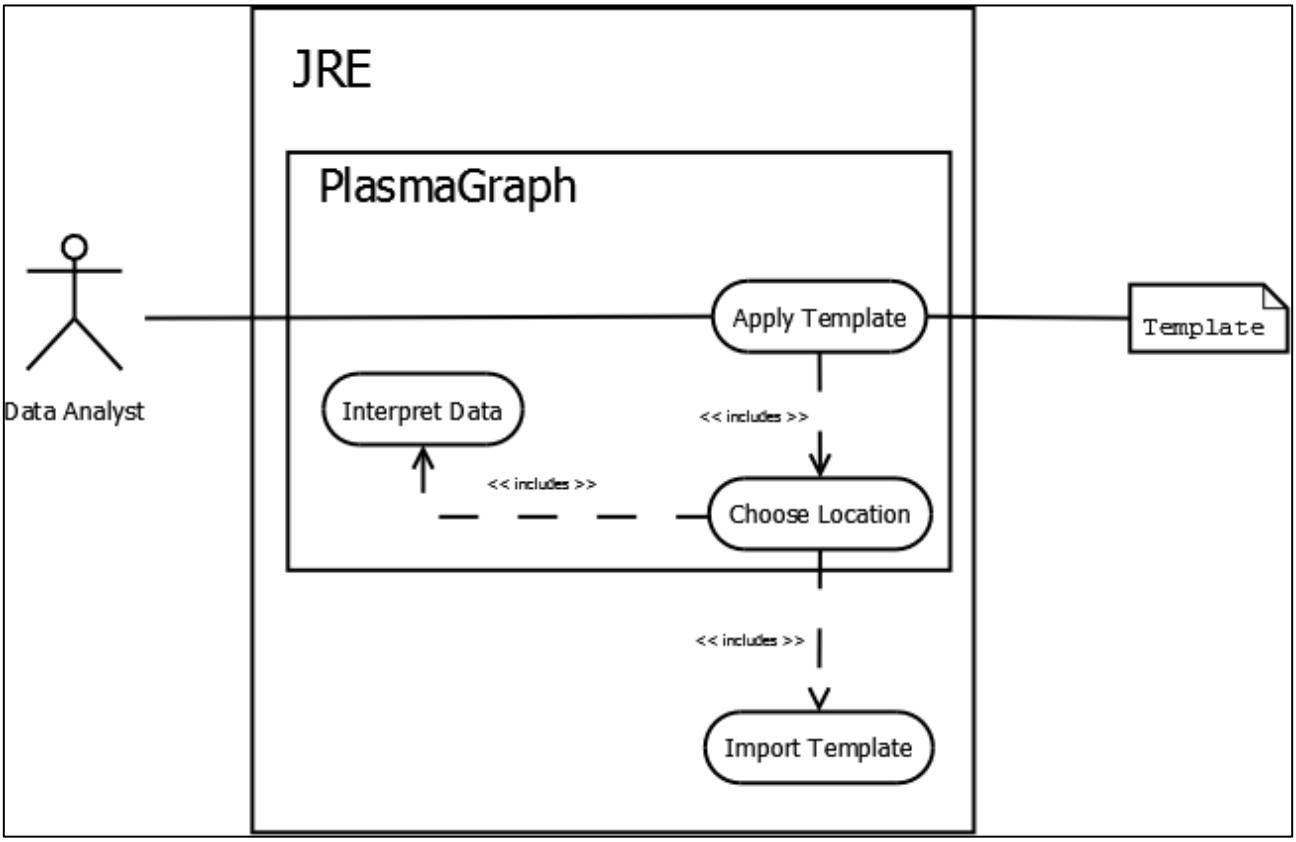

Figure 13: NR-02

2.2.2.3 Inspect Data (NR-03)

A data analyst can ask PlasmaGraph to show him or her, a human readable version of the data file imported.

The information presented includes:

a. Name of the data file.
b. Name of each variable contained within the data file
c. Data type of each variable contained within the data file.
d. Total amount of values contained in each variable.
e. Every value contained in each variable.

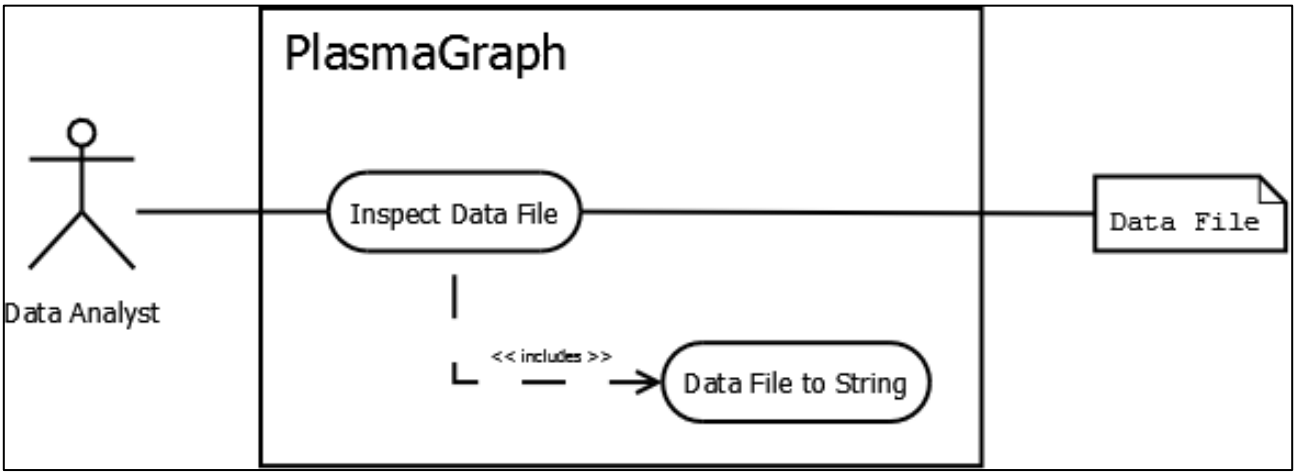

Figure 14: NR-03

2.2.2.4 Display Help (NR-04)

Provide a written guide describing how to use PlasmaGraph's user interface. The guide is composed of an index and several sections that can be loaded on demand by the user.

2.2.2.5

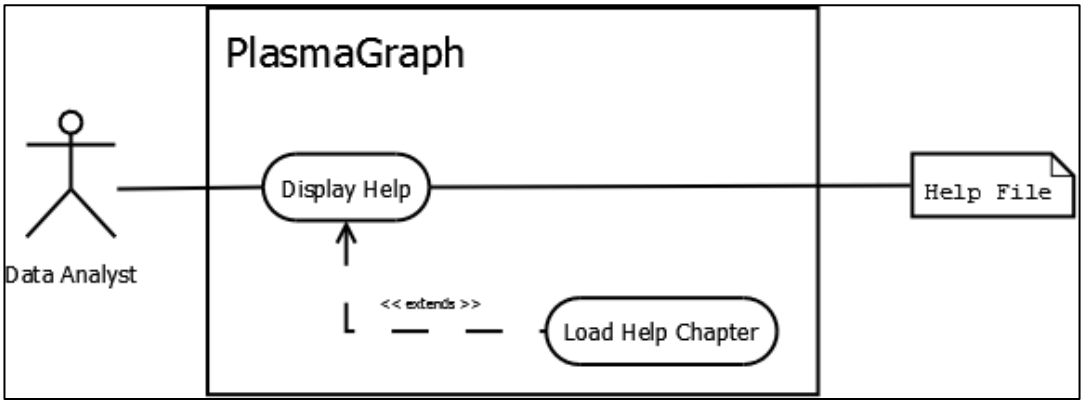

Figure 15: NR-04

Choose Location (NR-05)

In order to save any type of file into the data analyst's computer, PlasmaGraph needs to interact with both JRE and the data analyst. It needs to retrieve the directory structure that describes the data analyst's computer, ask the data analyst to choose one of the addresses specified and then order JRE where to save the file based on the selection.

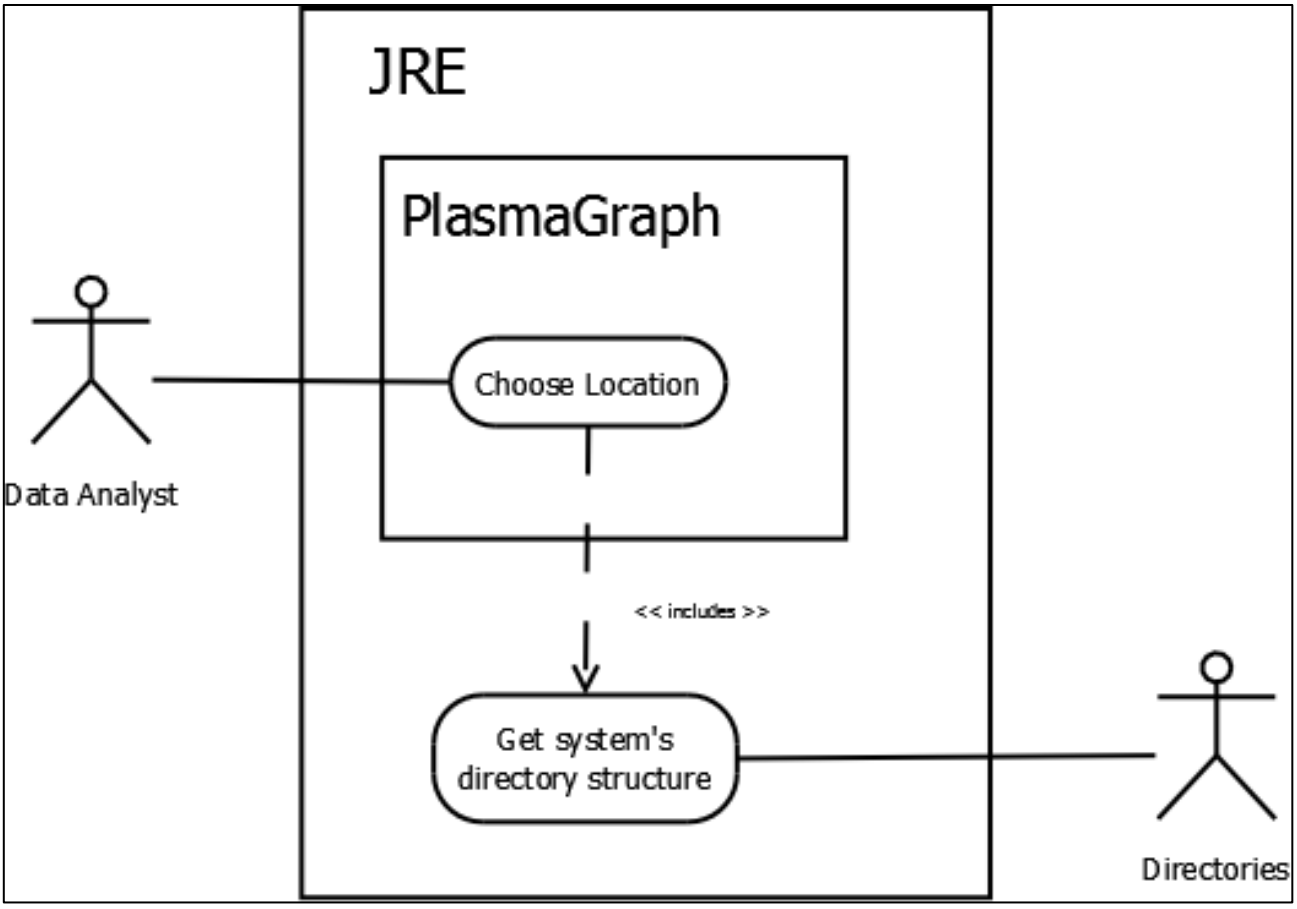

Figure 16: NR-05

2.2.2.6 Elicit Graph Options (NR-06)

When the data analyst tells the system to create a graph, it will check that all options necessary for creating the graph are selected. Otherwise, PlasmaGraph will inform the data analyst to choose all necessary options before attempting to produce the graph.

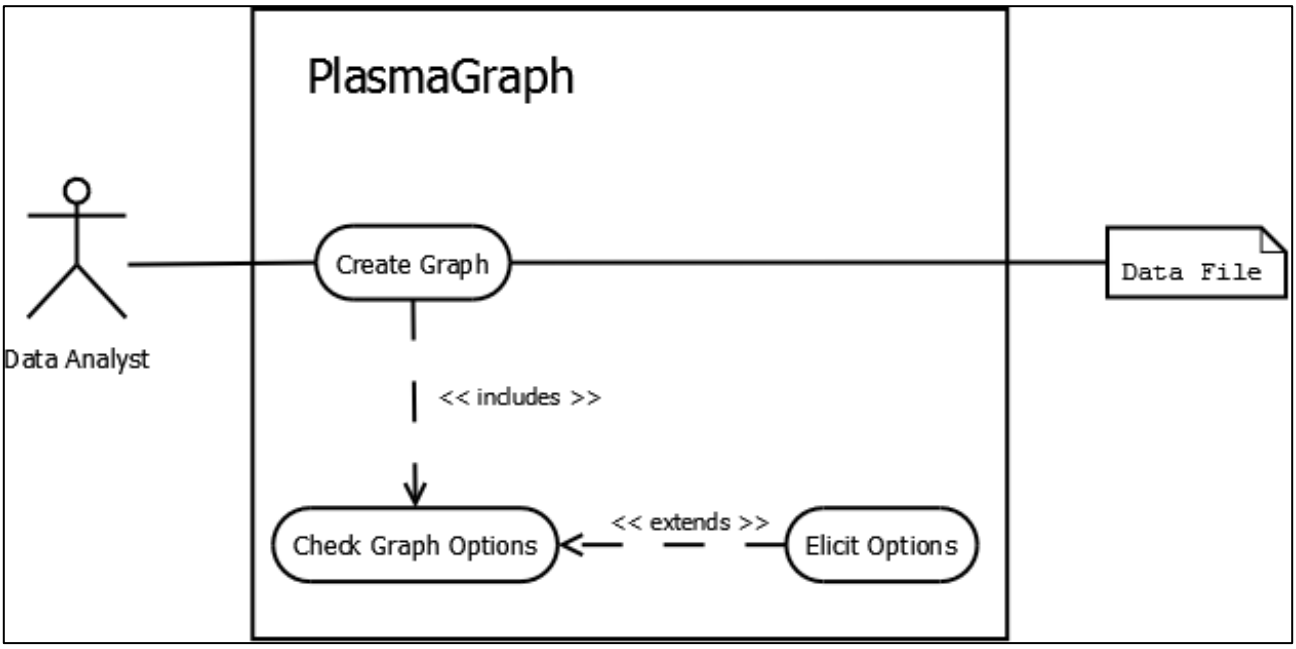

Figure 17: NR-06

The system can't produce any graph unless the data analyst imports a data file and choses a set of values for X and a set of values for Y. Other options available to the data analyst for configuring how this system draw graph charts are described in more detail in section 2.2.1.1 of this document.

2.3 User Characteristics

2.3.1 Data Analyst

This type of user will interact with PlasmaGraph through the system's graphical user interface PG UI which is defined in section 2.1.1.1.

A data analyst has to comply with the following two requirements:

1. He or she has at least five of the nine skills that determine a basic level of competency using Microsoft Word according to Concordia University's human resources department [24].
2. He or she is at least a senior student at a mathematics, engineering or science university program. In the case of the Polytechnic University of Puerto Rico, the data analysts are the students, professors, and mentors that do research.

2.3.2 Programmer

This kind of user can interact with PlasmaGraph using the system development kit PG DK defined in section 2.1.1.2. The person must have programming skills in both MATLAB and Java.

2.4 Design and Implementation Constraints

There are some items that can limit the developer's options to design and implement PlasmaGraph. These items can be subdivided as follows:

2.4.1 Hardware Limitations

This system is designed to function in a personal computer with the following minimum specifications:

a. 15'' Monitor with a resolution of 1200x700 and refresh rate of 30Hz
b. A standard alphanumeric American (QWERTY) keyboard
c. Two 3.20Ghz microprocessors
d. 500Mb of available HDD
e. 512Mb of RAM

2.4.2 Interfaces to other applications

PlasmaGraph was originally made as a tool to make graph charts from experiment results obtained using MATLAB so it is logical to consider compatibility between these two systems. At the time of

delivery, the product can only create graph charts from data in files that are in either comma separated value format *.csv or MATLAB's LEVEL 5 MAT-File format.

2.5 User Documentation

This product is delivered with a User's Manual (UM-2-2014-05-28) which describes how to use PlasmaGraph and an Installation Manual (IM-2-2014-05-28) which describes how to install PlasmaGraph.

2.6 Assumptions and Dependencies

PlasmaGraph runs over Java Runtime Environment version 7 or above. It also makes use of JFreeChart "a Java chart library that makes it easy for developers to display professional quality charts in their applications" *[25]*.

2.7 Apportioning of Requirements

The following requirements won't be delivered with this version of the system:

a. Importing data files with variables that don't have the same amount of values.
b. Plot values for a third axis (Z).
c. Make bar charts.

3. Specific Requirements

3.1 External Interfaces

This section provides a detailed description of all inputs into and outputs from PlasmaGraph. Because this system has two points of interaction with the outside, the following two subsections will explain in detail inputs and outputs:

3.1.1 PlasmaGraph User Interface (PG UI)

As described in section 2.1.1.1 of this document, this is the interface that the data analyst is going to utilize in order to produce graph charts and interact with the system in general.

3.1.1.1 Sources of input and destination of output

Input	Output
Users	Template Files
Data Files	Graph Charts
Template Files	

Table 4: Input & Output

3.1.1.2 Validity of inputs

PG UI limits considerably the input that its users can enter into the system. Most input is gathered by means of text fields which can contain 100 ASCII characters or less.

Data files on the other hand have to comply with the following format and structure:

a. The file must be formatted as a binary MATLAB Level 5 MAT-File [19].
b. All objects contained in the file must be MATLAB arrays of 1 column and any amount of rows. Every array needs to have the same amount of rows.
c. All arrays contained in the file must be populated with at least one MATLAB variable or constant and all variables within an array have to be of the same data type.
d. Variables or constants contained by arrays must be of the data type double, char or cell [22].
e. The file size must be 56300kb or less.

Template files have to comply with the following format and structure:

a. The file must be formatted as a plain text file with ASCII characters only.
b. Each line of text has to identify exactly one option from the Tools Pane ***[see section 2.1.3.2]***.

3.1.1.3 Relationship between inputs and outputs

All data collected by PG UI is used to produce a specific graph chart. Data coming from data files is used in conjunction with data collected from the user to order the system how to produce the graph. Data files are used to determine which points can be drawn in the data pane described in section 2.1.3.4, and data collected from the user indicates to PlasmaGraph the specific points for the drawing, the axis labels, and how to group, interpolate or find outliers.
Template files are used to inform the system how to generate particular types of graph charts.

3.1.1.4 Timing

Importing a data file, generating a graph chart, and saving that graph chart can be done in less than 10 minutes.

3.1.1.5 Screen format

PG UI is designed to function on a 15'' monitor with a resolution of 1200x700 pixels and refresh rate of 30Hz. Although bigger monitors with more resolution and refresh rate won't affect any functional requirement of this product.

3.1.2 *PlasmaGraph Development Kit (PG DK)*

As described in section 2.1.1.2 of this document, this is the development kit that programmers will be using to order PlasmaGraph how to interpret files produced by the mirror and cusp plasma machine.
The development kit includes all the tools necessary for creating new versions of PlasmaGraph (add new features or update old features).

3.2 Functions

3.2.1 Use Case: Import Data (FR-01)

Use case: Import Data (FR-01)			
Principal Actor	Secondary Actors	Pre-conditions	Post-conditions
Data Analyst (DA)	Plasma Laboratory Data File JRE	Choose Location (NR-05)	N/A

SCENARIOS			
#	Principal	Exception: (E1) “Original file isn’t well formatted”	Exception: (E2) "Error Loading file"
1	DA asks the system to import a data file.		
2	Choose Location (NR-05)		
3	JRE asks the system for a file’s address.		
4	The system gives the address chosen in step 2 by DA to JRE.		
5	JRE gives the system the data file contained in the address provided.		
6	The system tells DA that the file was loaded successfully.	The system tells DA that the file is not formatted correctly.	
7			JRE gives an error message to the system.
8			The system tells DA that an error occurred while loading the file.

Table 5: Use Case (FR-01)

3.2.2 Use Case: Validate Data (FR-02)

Use case: Validate Data (FR-02)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	Data File	Import Data (FR-01)	N/A

SCENARIOS			
#	Principal	Alternate: (A1) "Data file has NaN values"	Alternate: (A2) "All values are NaN"
1	The system asks DA to choose a variable from the data file imported to assign to axis X and a variable to assign to axis Y.		
2	DA tells the system which variable to use for the X axis and which to use for the Y axis.		
3		The system tells DA that one or more values weren't included in the selected axis.	The system tells DA that it can't make a graph with the specified data because there isn't any valid value in one of the selected variables.

Table 6: Use Case (FR-02)

3.2.3 Use Case: Choose Graph Options (FR-03)

Use case: Choose Graph Options (FR-03)								
Principal Actor			Secondary Actor		Pre-conditions		Post-conditions	
Data Analyst (DA)			N/A		Import Data (FR-01)		N/A	
SCENARIOS								
#	Principal	Alternate: (A1) "DA choose more than the minimal options"	Alternate: (A2) "Setting Scales"	Alternate: (A3) "Setting Labels"	Alternate: (A4) "Grouping by a third variable"	Alternate : (A5) "Outlier search"	Alternate: (A6) "Interpolating"	Exceptional : (E1) "Invalid Grouping"
1	The system gives DA a list of names. Each name correspond s to exactly one of the variables in the data imported.							
2	DA looks at the list and tells the system which of those variables should be associated to the X axis of the graph and which to the Y axis of the graph.							
3		Any possible combinatio n of any of the alternate scenarios from A2 to A6.	The system asks DA if the graph should use a numerical scale for the X axis or a logarithmic scale.	The systems asks DA to provide the following information in 3 different text fields: a. Title of the graph b. Title of X axis c. Title of Y axis	DA looks at the list and tells the system which one of those variables shall be used to group points defined by any (X,Y) coordinate .	The system asks DA if it should issue a warning message in case of finding outliers when creating the graph or if it should simply ignore outliers	The system asks DA if it should interpolate the points in the graph using any of these methods: a. Linear interpolation. b. Quadratic interpolation. c. Spline interpolation.	A4

Use case: Choose Graph Options (FR-03)								
Principal Actor			Secondary Actor		Pre-conditions		Post-conditions	
Data Analyst (DA)			N/A		Import Data (FR-01)		N/A	
SCENARIOS								
#	Principal	Alternate: (A1) "DA choose more than the minimal options"	Alternate: (A2) "Setting Scales"	Alternate: (A3) "Setting Labels"	Alternate: (A4) "Grouping by a third variable"	Alternate : (A5) "Outlier search"	Alternate: (A6) "Interpolating"	Exceptional : (E1) "Invalid Grouping"
						and don't include them in the graph.		
4			DA tells the system which scale to use for the X axis (numerical or logarithmic) .	DA gives the system the following information in the form of 3 different strings of characters, one for each of the text fields provided: d. Title of the graph e. Title of X axis f. Title of Y axis		DA tells the system if it should issue a warning message in case of finding outliers when creating the graph or if it should simply ignore outliers and don't include them in the graph.	DA tells the system if it should use a linear, quadratic or spline interpolation method for the points plotted in the graph.	The system tells DA one of the following messages: a. The grouping variable needs to have less than 25 different values b. The grouping variable needs to have at least two repeated values.

Table 7: Use Case (FR-03)

3.2.4 Use Case: Create Graph (FR-04)

Use case: Create Graph (FR-04)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	Data File	Import Data File	N/A

SCENARIOS			
#	Principal	Alternate: (A1) "Graph with interpolation"	Alternate: (A2) "Graph with outlier warning"
1	Import Data (FR-01)		
2	Choose Graph Options (FR-03)		
3	DA tells the system to create a graph.		
4	The system tells DA that the graph was created.	The system tells DA that the graph was created and the confidence level of the data plotted.	The system tells DA that the graph was created and the amount of outliers it has

Table 8: Use Case (FR-04)

3.2.5 Use Case: Save Graph (FR-05)

Use case: Save Graph Chart (FR-05)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	JRE Graph Data File	Create Graph (FR-04)	N/A

SCENARIOS				
#	Principal	Alternate: (A1) "Writing over an existing file"	Exception: (E1) "Not Writing over an existing file"	Exception: (E2) "JRE wasn't able to save the graph"
1	DA tells the system to save a graph			
2	Choose Location (NR-05)			
3	DA gives the system the address of a location where it can save the graph			
4	The system tells JRE to save the graph in the location described by the address provided by DA.			

Use case: Save Graph Chart (FR-05)				
Principal Actor	Secondary Actor	Pre-conditions		Post-conditions
Data Analyst (DA)	JRE Graph Data File	Create Graph (FR-04)		N/A
SCENARIOS				
#	Principal	Alternate: (A1) "Writing over an existing file"	Exception: (E1) "Not Writing over an existing file"	Exception: (E2) "JRE wasn't able to save the graph"
5	JRE saves the graph in the location provided.	JRE tells the system that saving the graph in the address provided will result in overwriting another file.	JRE tells the system that saving the graph in the address provided will result in overwriting another file.	JRE tells the system that it wasn't able to save the graph in the location provided.
6	The system tells DA that the graph has been saved.	The system tells DA that saving the graph in the address provided will result in overwriting another file	The system tells DA that saving the graph in the address provided will result in overwriting another file	The system tells DA that it can't save the graph in the location provided.
7		DA tells the system to continue and save the graph anyways.	DA tells the system not to save the graph.	

Use case: Save Graph Chart (FR-05)				
Principal Actor	Secondary Actor	Pre-conditions		Post-conditions
Data Analyst (DA)	JRE Graph Data File	Create Graph (FR-04)		N/A
SCENARIOS				
#	Principal	Alternate: (A1) "Writing over an existing file"	Exception: (E1) "Not Writing over an existing file"	Exception: (E2) "JRE wasn't able to save the graph"
8		The system tells JRE to save the graph in the address provided by DA even if it results in overwriting another file.		

Table 9: Use Case (FR-05)

3.2.6 Use Case: Save Template (NR-01)

Use case: Save Template (NR-01)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	JRE Template	Import Data (FR-01)	N/A

SCENARIOS				
#	Principal	Alternate: (A1) "Writing over an existing file"	Exception: (E1) "Not Writing over an existing file"	Exception: (E2) "JRE wasn't able to save the template"
1	DA asks the system to save a template.			
2	Choose Location (NR-05 A1)			
3	The system gives JRA the address and name provided by DA in step 2.	JRE tells the system that saving a template file in the location provided with the name provided will result in overwriting another file.	JRE tells the system that saving a template file in the location provided with the name provided will result in overwriting another file.	JRE tells the system that it wasn't able to save the template in the location provided.
4	JRE saves the template file in the address provided using the name provided.	The system tells DA that saving the template in the location specified will result in overwriting another file.	The system tells DA that saving the template in the location specified will result in overwriting another file.	The system tells DA that it wasn't able to save the template in the address provided.

Use case: Save Template (NR-01)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	JRE Template	Import Data (FR-01)	N/A

SCENARIOS				
#	Principal	Alternate: (A1) "Writing over an existing file"	Exception: (E1) "Not Writing over an existing file"	Exception: (E2) "JRE wasn't able to save the template"
5		DA tells the system to proceed anyways and save the template	DA tells the system not to save the template.	
6		The system tells JRE to save the template file in the address provided and with the name provided even if it means overwriting another file.	The system tells JRE not to save the template.	

Table 10: Use Case (NR-01)

3.2.7 Use Case: Import Template (NR-02)

Use case: Import Template (NR-02)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	JRE Template	N/A	N/A
SCENARIOS			
#	Principal	Exception: (E1) "Invalid address"	Exception: (E2) "Invalid template file"
1	DA asks the system to import a template.		
2	Choose Location (NR-05)		
3	The system gives JRE the address specified by DA in step 2.		
4	JRE gives the system the data stored in the address specified.	JRE tells the system that there isn't any file in the address specified.	JRE gives the system the data stored in the address specified.
5			The system tells DA that the file selected isn't a valid template file.

Table 11: Use Case (NR-02)

3.2.8 Use Case: Inspect Data (NR-03)

Use case: Inspect Data (NR-03)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	Data File	Import Data (FR-01)	N/A
SCENARIOS			
#	Principal		
1	DA asks the system to show him the contents of the data file imported.		
2	The system gives the user a list naming each of the variables in the data file and providing the following information for each variable: a. Data type. b. Amount of values contained. c. Each value contained in the variable		

Table 12: Use Case (NR-03)

3.2.9 Use Case: Display Help (NR-04)

Use case: Display Help (NR-04)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	Help File	N/A	N/A
SCENARIOS			
#	Principal		
1	DA asks the system for help.		
2	The system gives DA a help file which explains how to use the system's user interface.		

Table 13: Use Case (NR-04)

3.2.10 Use Case: Choose Location (NR-05)

Use case: Choose Location (NR-05)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	JRE Directories	N/A	N/A
SCENARIOS			
#	Principal	Alternate: (A1) “Choose location and file name”	
1	The system asks JRE to provide a hierarchical tree structure with the specific addresses of all directories available.		
2	The system gives DA a set of addresses organized in the form of a hierarchical tree structure.		
3	DA gives the system one of the addresses contained in the hierarchical tree structure given.	DA gives the system one of the addresses contained in the hierarchical tree structure given and a name that isn’t in the location determined by that address.	

Table 14: Use Case (NR-05)

3.2.11 Use Case: Elicit Graph Options (NR-06)

Use case: Elicit Graph Options (NR-06)			
Principal Actor	Secondary Actor	Pre-conditions	Post-conditions
Data Analyst (DA)	Data File	N/A	N/A
SCENARIOS			
#	Principal		
1	DA tells the system to create a graph.		
2	The system tells DA that it can't make a graph until he or she selects the minimal options for creating graphs.		
3	Choose Graph Options (FR-03)		
4	Create Graph (FR-04)		

Table 15: Use Case (NR-06)

3.3 Performance Requirements

PlasmaGraph can only be used by one person at a time and can only process one file at a time. The size of the data file loaded for generating the graph must be smaller than 56300kb. The time the system takes to create the graph is no more than 5 minutes.

3.4 Design Constraints

This software product will be made using Java programming language and following Oracle's code conventions which contemplates "filenames, file organization, indentation, comments, declarations, statements, white space, naming conventions,[and] programming practices" [26].

Additionally, PlasmaGraph has four different specification documents that describe the product which are:

- Software Requirements Specifications [27].
- Software Design Description [28].
- Software Project Management Plan [29].
- Software Test Documentation [30].

All these documents are stored in a subfolder named "specs" and have the following naming conventions: The document's name acronyms (SRS, SDD, STD or SPMP) followed by a number from 0 to 3 and the release date in the format yyyy-mm-dd all separated by "-".

The number that goes after the acronym of the document's name provides the following information:

- The number **'0'** indicates that information contained in the document is incomplete and does not provide a coherent explanation of the subject. This is called "pre-alpha".
- The number **'1'** indicates that the document contains sufficient information to be coherent and informative, but still lacks detail. This is called "alpha";

- The number **'2'** indicates that the document is complete. This means that it is coherent, informative, and covers all the details required by the standard on which it is based on. This is called "beta".
- The number **'3'** indicates the same as '2' and the document was revised by a third party (not the developers or client). This document has all its information in an organized format and is free from errors. It is called "release candidate".

For example: SRS-0-2014-01-20 reefers to the Software Requirements Specifications document version pre-alpha released on January 20, 2014.

3.4.1 Standards Compliance

All code submitted must be tested for compliance against Oracle's code conventions before inclusion in PlasmaGraph's source. Any arbitrary piece of code being tested has to comply with these conventions in at least 50%.
Changes made to PlasmaGraph need to be updated in their respective specification document as well.

3.5 Software System Attributes

3.5.1 Reliability

The following are scenarios that could disrupt PlasmaGraph and explanations on how the system deals with them:

- Read errors: either the data file or the template file is incompatible with the system. In this case, the system displays a message reporting the issue to the user.
- Data storing errors: the system wasn't able to store the template or graph produced. In this case the system displays a message reporting the issue to the user.

3.5.2 Security

Because this system won't deal with any particularly sensitive information, there were no security considerations taken beside those provided by the platform on which this application runs which is the Java Runtime Environment version 7.

3.5.3 Maintainability

This product will use only java standard libraries with the exeptions of JFreeChart 1.0.16 and Opencsv 2.3. A library's source code cannot be altered so the original documentation of each one can stay as true to its software as possible.
PlasmaGraph uses git [31] which is a distributed version control system that helps to keep track of code revisions and allows many developers to work on the same project.

3.5.4 Portability

PlasmaGraph runs in the Java Runtime Environment which means that it doesn't depend on a specific OS. The JRE can be executed in Windows, Mac, and Linux as well as many other operative systems [5].