Software Requirements Specification

for

PlasmaGraph

Version 2-2014-05-01

|  |  |  |
| --- | --- | --- |
| Daniel E. Quintin |  | Gerardo A. Navas |
| Computer Science Undergraduate Program |  | Computer Science Undergraduate Program |

Polytechnic University of Puerto Rico

May 1, 2014

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Product Scope 1

1.3 References 3

2. Overall Description 6

2.1 Product Perspective 6

2.2 Product Functions 14

2.3 User Characteristics 25

2.4 Design and Implementation Constraints 25

2.5 User Documentation 26

2.6 Assumptions and Dependencies 26

2.7 Apportioning of Requirements 26

3. Specific Requirements 27

3.1 External Interfaces 27

3.2 Functions 30

3.3 Performance Requirements 43

3.4 Design Constraints 43

3.5 Software System Attributes 44

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Daniel E. Quintini | 2014/5/1 | New requirements needed to be added | 1-2014-05-01 |
|  |  |  |  |

# Introduction

## 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” (Colizeeum Technology). Also, it explains the necessity of developing a software tool that can help in the analyzing of experiments data for the plasma research team at the Polytechnic University of Puerto Rico as well as specifying what that software is to do.

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

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

## 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® “a high-level language and interactive environment for numerical computation, visualization, and programming” (Mathworks.com, MATLAB: The Language of Technical Computing) 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.

We the suppliers have come up with our solution. A software product named “PlasmaGraph” which 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 is what we will call a “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 the data analyst.
* Provide a graphical user interface that simplifies making graph charts. This means that the data analyst must be able accomplish 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).
  + Produce any of the graph charts described above but grouping (X, Y) points using a third set of values. This third set can’t 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. Also, thanks to template files (TEM), the data analyst only needs to tell the system once how to handle the data being graphed. This way, the user has the option to only 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.

## References

Colizeeum Technology. “What are the benefits of a Great SRS?” Web. 21 Jan 2014. <<http://www.colizeeumtechnology.com/OurMethodology/SRS.aspx>>

Mathworks.com. "MATLAB: The Language Of Technical Computing". The MathWorks, Inc., Web. 21 Jan 2014. <<http://www.mathworks.com/products/matlab/>>

Tableausoftware.com. "Business Intelligence and Analytics”. Tableau Software, Web. 21 Jan 2014. <<http://www.tableausoftware.com/>>

Visual.ly. “Infographics & Data Visualization”. Visually, Inc. Web 21 Jan 2014. <<http://visual.ly>>

Oracle.com. “Java SE at a Glance”. Oracle Corporation. Web 21 Jan 2014. <<http://oracle.com/technetwork/java/javase/overview/index.html>>

Opencsv.ourceforge.net. “What is Opencsv?”. Opencsv Project. Web 21 Jan 2014. <<http://opencsv.sourceforge.net>>

Jfree.org. “JFreeChart 1.0.17 API Documentation”. Object Refinery Limited. Web 21 Jan 2014 <<http://www.jfree.org/jfreechart/api/javadoc/index.html>>

Mathworks.com. “Overview of Java Interfaces”. The MathWorks, Inc. Web 21 Jan 2014 <<http://www.mathworks.com/help/matlab/matlab_external/product-overview.html>>

Docs.oracle.com. “Package javax.swing”. Oracle Corporation. Web 21 Jab 2014. <<http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>>

Docs.oracle.com “Java™ Platform, Standard Edition 7

API Specification”. Oracle Corporation. Web 21 Jan 2014. <<http://docs.oracle.com/javase/7/docs/api/>>

Docs.oracle.com. “Initial Heap Size and Maximum Heap Size Changed for Parallel Garbage Collector” Oracle Corporation. Web 21 Jan 2014. <<http://docs.oracle.com/javase/7/docs/technotes/guides/vm/gc-ergonomics.html>>

Docs.oracle.com. “JDK 7 and JRE 7 Installation Guide”. Oracle Corporation. Web 21 Jan 2014 <<http://docs.oracle.com/javase/7/docs/webnotes/install>>

Mathworks.com. “MATLAB MAT-File Format R2013b Figure 1-1”. The MathWorks, Inc. Web 21 Jab 2014. <<http://www.mathworks.com/help/pdf_doc/matlab/matfile_format.pdf>>

java.com. “Learn About Java Technology”. Oracle Corporation. Web 21 Jab 2014. <https://www.java.com/en/about/>

(Code Conventions for the Java Programming Language) http://www.oracle.com/technetwork/java/codeconv-138413.html

# Overall Description

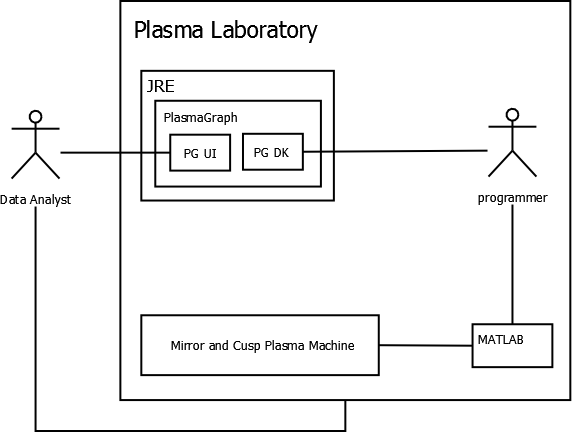
## Product Perspective

There are several data visualization software products in the market like Tableau (Tableausoftware.com) and Visua.ly (Visual.ly) 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. Also, because this software is made by students of the Polytechnic University of Puerto Rico as part of their curriculum, the resulting code of the product can be maintained and improved by other students. This means that PlasmaGraph’s value has the potential to increase at the same time that it also help students gain better skills and graduate with a deeper understanding of how to work with software. The following table sheds light on the pros and cons of the tools considered and our own solution.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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 (if its installed it can be used) * Developed with Java to ensure portability. 89% of Desktops in the U.S. Run Java (www.java.com/en/about) * Source code available for future improvements by other students * 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 takes 6 months to be finished |

### 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 (java.com, Learn About Java Technology) and is comprised itself by another two systems that work together in order to accomplish the requirements stated in this document. These systems are PlasmaGraph User Interface (PG UI) and PlasmaGraph Development Kit (PG DK).



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 the data analyst can ask the system to retrieve the converted file from his or her computer and generate a graph with the data contained inside said file.

The programmer can also communicate with PlasmaGraph through PG DK 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.

#### 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 stated in section 2.2.1 of this document.

#### 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.

### 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 | *(Oracle.com, Java SE at a Glance)* |
| **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 | 2.3 | (sourceforge.net, What is Opencsv?) |
| **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 |  | 1.0.17 | (jfree.org, JFreeChart 1.0.17 API Documentation) |
| **commons-lang3-3.2.1** |  |  |  |  |
| **commons-math3-3.2** |  |  |  |  |
| **commons-validator-1.4.0** |  |  |  |  |
| **Hamcrest-core** |  |  |  |  |
| **jcommon** |  |  |  |  |
| **jmatio** |  |  |  |  |
| **junit** |  |  |  |  |

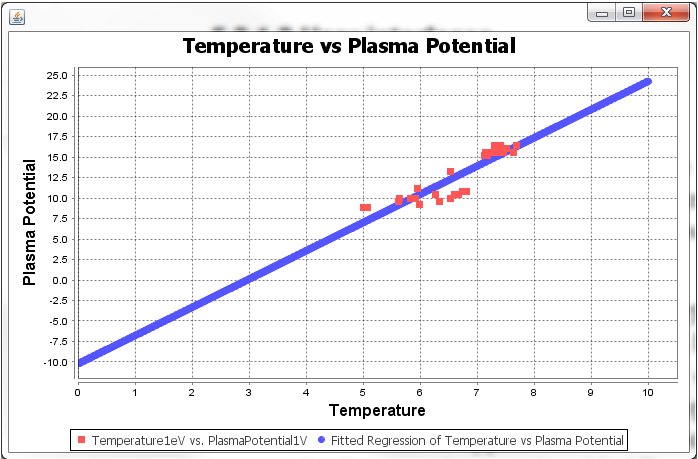
***PlasmaGraph is compatible with any system that can run a Java Runtime Environment; which means that it doesn’t depend on a specific operating system. Also, it can interact with MATLAB directly through the development kit described in section 2.1.1.2 provided that the programmer includes the correct libraries in the code (mathworks.com, Java libraries).***

### 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 the primary Java GUI widget toolkit (docs.oracle.com, Package javax.swing) and its components are:

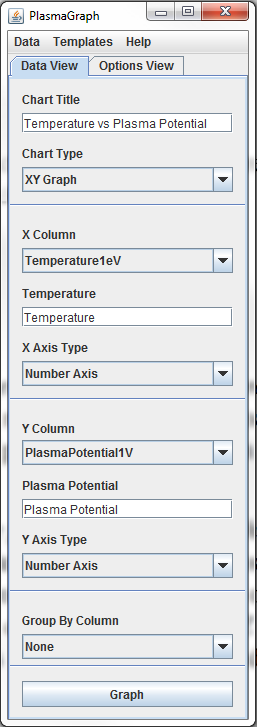
#### Graph Pane

This is the portion of the program where the graph is displayed. It can also be used to save said graph in the user’s computer.



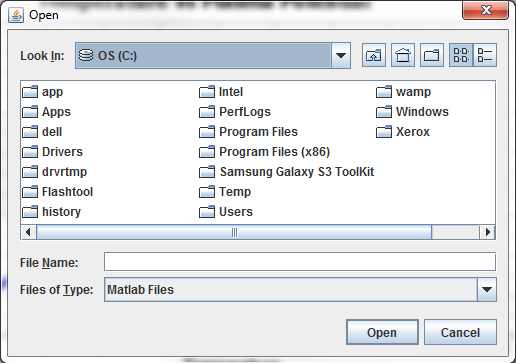
#### Tools Pane

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 or not. Also, here the user can ask 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 or not.

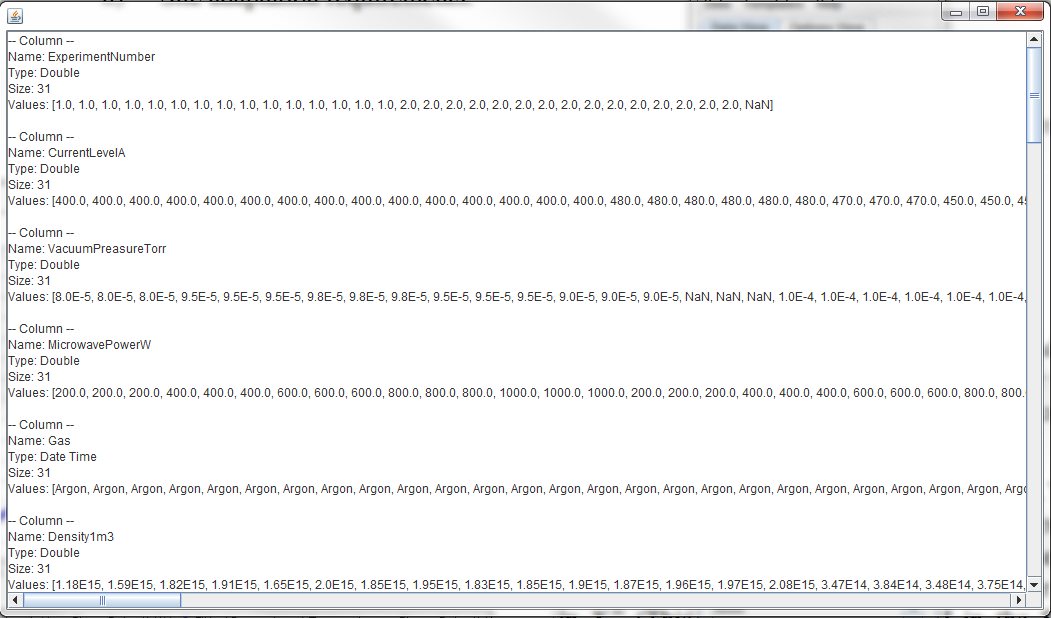
#### File Chooser

It is used to navigate the file system of the computer in which the java runtime environment is executing PlasmaGraph, and then either choosing a file or directory from said system.



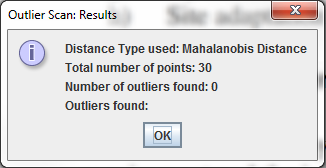
#### Data Pane

This component is used to display the contents of a data file in a way that is readable by humans.



#### Options Pane

They are windows meant for transmitting messages to the user and in some cases elicit an option from him or her.

### 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 (docs.oracle.com, Java™ Platform, Standard Edition 7 API Specification).

### Communications Interfaces

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

### 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” (docs.orale.com, Initial Heap Size and Maximum Heap Size Changed for Parallel Garbage Collector).

### 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 (docs.oracle.com, JDK 7 and JRE 7 Installation Guide) 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 HD
* 512Mb of RAM

## Product Functions

PlasmaGraph is a software tool design 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 from a graphical interface thus cutting down to zero the programming skills needed for this task. 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.

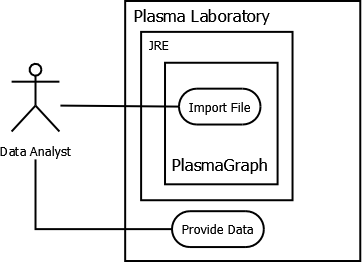
### Functional Requirements

#### 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. Atthe time of delivery PlasmaGraph is only able to import files that meet the following format and structure:

1. The file must be formatted as a binary MATLAB Level 5 MAT-File (http://www.mathworks.com/help/pdf\_doc/matlab/matfile\_format.pdf).
2. 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. (http://www.mathworks.com/help/matlab/cell-arrays.html).
3. 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. (http://www.mathworks.com/help/matlab/matlab\_prog/create-variables.html)
4. Variables or constants contained by arrays must be of the data type double, char or cell. (<http://www.mathworks.com/help/symbolic/double.html>), (<http://www.mathworks.com/help/symbolic/char.html>), (<http://www.mathworks.com/help/matlab/cell-arrays.html>).
5. The file must occupy 56300kb or less.

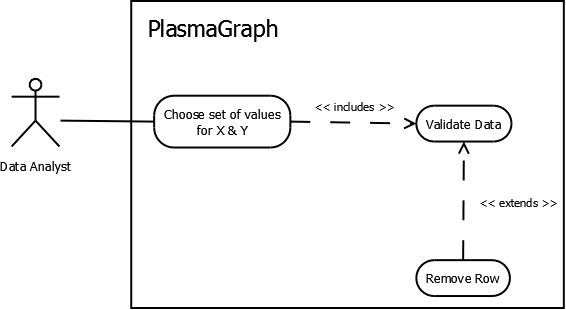
Once the data analyst has this file him or her, can tell PlasmaGraph where to import the file from.



#### 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 (http://www.mathworks.com/help/matlab/ref/nan.html) 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.



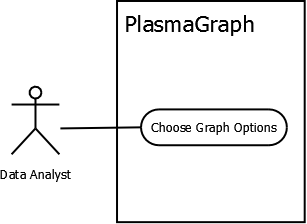
***It should be noted that removing rows from a chosen set of X and Y values won’t remove anything from the original data file imported.***

#### 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:

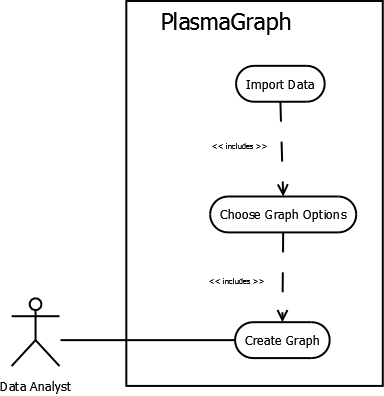
1. Which variable from the data file corresponds to the X axis of the graph? \*
2. Which variable from the data file corresponds to the Y axis of the graph? \*
3. Which type of scale shall be used in the X axis of the graph, numerical or logarithmic?
4. Which type of scale shall be used in the Y axis of the graph, numerical or logarithmic?
5. What is the title of the graph?
6. What kind of data is represented in the X axis of the graph?
7. What kind of data is represented in the Y axis of the graph?
8. Which variable from the data file, if any, shall be used to group points defined by any (X,Y) coordinate?
9. Should the system warn its user if there’re any outliers in the graph?
10. Should the system eliminate outliers from the graph?
11. 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 that of the above items only the ones that end with a \* are required for generating the graph.



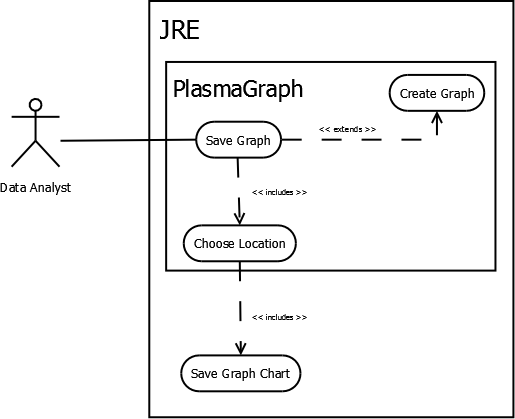
#### Create Graph (FR-04)

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.



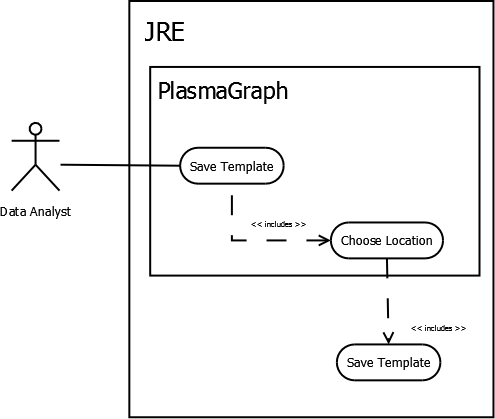
#### Save Graph (FR-05)

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.

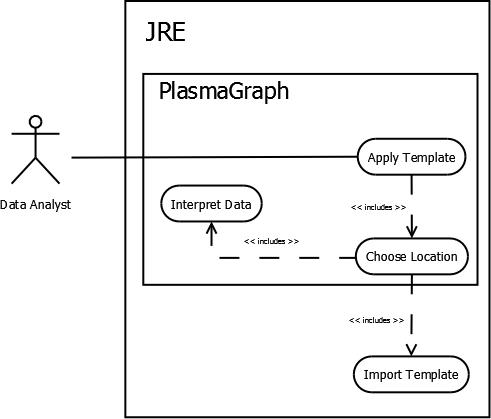


### Non-Functional Requirements

#### 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 data analyst to choose a save location and finally proceed to save the file.

#### Import Template (NR-02)

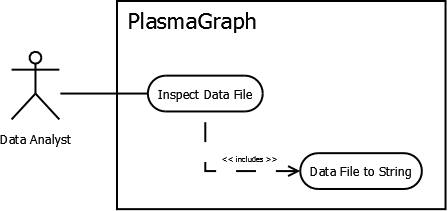
In order to import a template file, the data analyst needs to tell PlasmaGraph where the file is. Once this information has been identified, JRE will proceed to import the file to PlasmaGraph who is then responsible for interpreting the file and setting all graph options accordingly.

#### Inspect Data (NR-03)

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

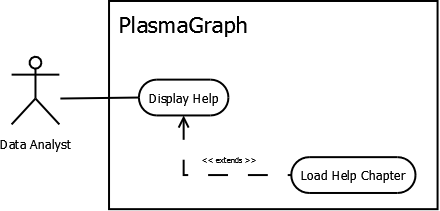
The information presented includes:

1. Name of the data file.
2. Name of each variable contained within the data file
3. Data type of each variable contained within the data file.
4. Total amount of values contained in each variable.
5. Every value contained in each variable.



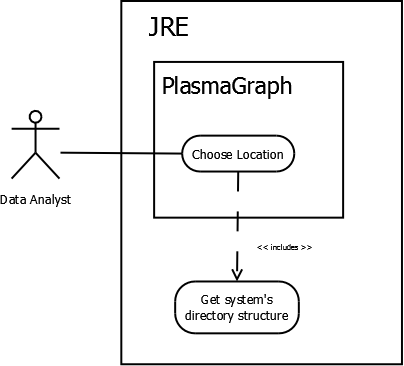
#### Display Help (NR-04)

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.



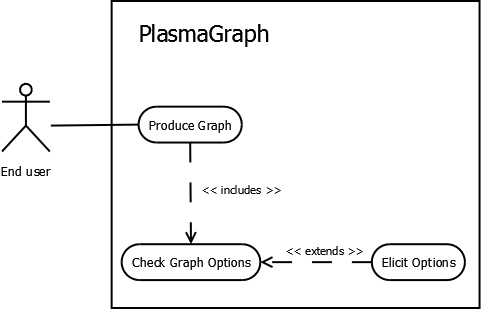
#### Choose Location (NR-05)

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, tell the data analyst to choose one of the addresses specified in said structure and then tell JRE where to save the file based on this selection.



#### 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 said graph are selected. Otherwise PlasmaGraph will tell the data analyst to choose all necessary options before attempting to produce the graph.



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.

## User Characteristics

### Data Analyst

This type of user will be interacting 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 (http://web2.concordia.ca/hr/eed/online\_guides).
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.

### Programmer

This kind of user can interact with PlasmaGraph using the system development kit PG DK defined in section 2.1.1.2. It needs to be someone with programming skills in both MATLAB and Java.

## Design and Implementation Constraints

There are some items that can limit the developer’s options when designing and implementing PlasmaGraph. These items can be subdivided as follows:

### Hardware Limitations

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

1. 15’’ Monitor with a resolution of 1200x700 and refresh rate of 30Hz
2. A standard alphanumeric American (QWERTY) keyboard
3. Two 3.20Ghz microprocessors
4. 500Mb of available HD
5. 512Mb of RAM

Not that this doesn’t mean that the system can’t function in computers with less than the minimum hardware requirements but rather that executing PlasmaGraph in hardware with less resources than the stated above is not guarantee to provide all functionality stated in this document.

### 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. But because PlasmaGraph is written using Java you could also use MATLAB’s Java interface (mathworks.com, Overview of Java Interfaces) and PG API interface to integrate both systems.

### Control functions

The global scope of this system must have a function that interprets data and template files; one that elicit options, one that applies filters to the data, one that produces graph charts and another that saves graph charts.

## User Documentation

This product is delivered with a User’s Manual document (TBD) which teaches how to interact with PlasmaGraph’s PG UI in order to accomplish every functional requirement stated in section 2.2.1 of this document.

## Assumptions and Dependencies

TBD

PlasmaGraph runs on top of Java Runtime Enviroment 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” (jfree.org, JFreeChart) and OppenCSV “a very simple csv (comma-separated values) parser library for Java” ([opencsv.sourceforge.net](http://opencsv.sourceforge.net), what is opencsv?).

## Apportioning of Requirements

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

1. Importing data files with variables that don’t have the same amount of values.
2. Plot values for a third axis (Z).
3. Make bar charts.

# Specific Requirements

## 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 that relate to each of these particular interfaces:

### 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.

#### Sources of input and destination of output

On the surface, PG UI has only one sources of input which is the data analyst. However, this type of user can also interact with the plasma laboratory and ask for a data file which he can then imported into the system. This data file becomes the second source of input.

As for output destination; through PG UI the data analyst can save into his computer both template files and graph charts generated by the system.

#### Validity of inputs

PG UI restrains 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 and lists of predefined options which are always valid.

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

1. The file must be formatted as a binary MATLAB Level 5 MAT-File (http://www.mathworks.com/help/pdf\_doc/matlab/matfile\_format.pdf).
2. 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. (http://www.mathworks.com/help/matlab/cell-arrays.html).
3. 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. (http://www.mathworks.com/help/matlab/matlab\_prog/create-variables.html)
4. Variables or constants contained by arrays must be of the data type double, char or cell. (<http://www.mathworks.com/help/symbolic/double.html>), (<http://www.mathworks.com/help/symbolic/char.html>), (<http://www.mathworks.com/help/matlab/cell-arrays.html>).
5. The file must occupy 56300kb or less.

Finally, because this is the first version of the system, we can say that template files are valid if they were produced by PlasmaGraph and haven’t been altered.

#### Relationship between inputs and outputs

All data collected by PG UI serves to produce a specific graph chart. Data coming from data files is used in conjunction with data collected from the user to tell the whole system how to produce the graph.

To be more specific, data files are used to determine which points can be drawn in the data pane described in section 2.1.3.4 while the data collected from the user tells PlasmaGraph which specific points to actually draw, what labels to use and whether to group, interpolate or find outliers with those points.

Template files are used to act as if it was an data analyst who gave all necessary information for generating a graph chart except a data file.

#### Timing

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

#### Screen format

PG UI is design 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.

### *PlasmaGraph Development Kit (PG DK)*

As described in section 2.1.1.2 of this document, this is the interface that programmers will be using to tell PlasmaGraph how to interpret files produced by the mirror and cusp plasma machine.

This development kit includes all the tools necessary for creating new versions of PlasmaGraph which can come with new features, updated features or even less features

## Functions

### 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  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 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. | |

### Use Case: Validate Data (FR-02)

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

### 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 corresponds 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 combination 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:   1. Title of the graph 2. Title of X axis 3. 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 and don’t include them in the graph. | The system asks DA if it should interpolate the points in the graph using any of these methods:   1. Linear interpolation. 2. Quadratic interpolation. 3. Spline interpolation. | A4 |
| 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:   1. Title of the graph 2. Title of X axis 3. 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:   1. The grouping variable needs to have less than 25 different values 2. The grouping variable needs to have at least two repeated values. |

### Use Case: Create Graph (FR-04)

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

### 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 | | Produce Graph Chart (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. | |  | |  |  |
| 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. |  |
| 8 |  | | The system tells JRE to save the graph in the address provided by DA even if it results in overwriting another file. | |  |  |

### Use Case: Save Template (NR-01)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Use case: Save Template (NR-01) | | | | | | | |
| Principal Actor | | Secondary Actor | | | Pre-conditions | | Post-conditions |
| Data Analyst (DA) | | JRE | | | 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. | |
| 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. | |  | |

### Use Case: Import Template (NR-02)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use case: Import Template (NR-02) | | | | | | |
| Principal Actor | | Secondary Actor | | Pre-conditions | | Post-conditions |
| Data Analyst (DA) | | JRE | | 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. | |

### Use Case: Inspect Data (NR-03)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case: Inspect Data (NR-03) | | | | |
| Principal Actor | | Secondary Actor | Pre-conditions | Post-conditions |
| Data Analyst (DA) | | N/A | 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:   1. Data type. 2. Amount of values contained. 3. Each value contained in the variable | | | |

### Use Case: Display Help (NR-04)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case: Apply Template (NR-02) | | | | |
| Principal Actor | | Secondary Actor | Pre-conditions | Post-conditions |
| Data Analyst (DA) | | N/A | N/A | N/A |
| SCENARIOS | | | | |
| # | Principal | | | |
| 1 | DA asks the system for help. | | | |
| 2 | The system gives DA a manual which explains how to use the system’s user interface. | | | |

### Use Case: Choose Location (NR-05)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case: Choose Location (NR-05) | | | | | |
| Principal Actor | | Secondary Actor | | Pre-conditions | Post-conditions |
| Data Analyst (DA) | | JRE | | 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. | | |

### 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) | | N/A | 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) | | | |

## Performance Requirements

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

## Design Constraints

This software product is 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” (oracle.com, Code Conventions for the Java Programming Language).

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

* SRS – Made using document IEEE 830 SRS as guide
* SDD – Made using document IEEE 1016 SDD as guide
* STD – Made using document IEEE 829 STD as guide
* SPMP – Made using document IEEE 1058 SPMP as guide

All these documents are stored in a subfolder named “specs” and have the following naming conventions: The document’s name (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 “-“. For example SRS-0-2014-01-20 reefers to the SRS document version 0 released on January 20, 2014

### 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% of its whole.

Also, changes made to PlasmaGraph need to be updated in their respective specification document as well.

## Software System Attributes

### 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 required to make a graph couldn’t be red. 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.

### 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.

### 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 can’t be altered so the original documentation of each one can stay as true to its software as possible.

PlasmaGraph uses git 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.

### 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.