# Introduction

## Purpose

This document specifies and explains the necessity of developing a software product hereinafter referred to as “PlasmaGraph” or “the product”. It also specifies the features and functions that the product delivers as well as providing a general explanation of what PlasmaGraph can and can’t do. The information contained in this document is intended to help Angel E. González-Lizard, Ph. D hereinafter referred to as “the client” to understand the product. Another use for this document can be as a starting point and guide of reference along the development process for Daniel E. Quintini and Gerardo A Navas hereinafter referred to as “the developers”.

## Scope

PlasmaGraph is intended to assists in the job of analyzing figures collected in experiments by providing a tool capable of processing a file with tabular data and produce a visual representation of the numbers within the file. The product has a graphical user interface that the operative system user (hereinafter referred to as “the user”) can employ to interact with PlasmaGraph. Users can load files, specify how PlasmaGraph will interpret the data in that file (this can be saved as a “view”), tell PlasmaGraph to make a visual representation of the file loaded and save the resulting image in the operative system.

What the product does:

* Show a visual representation of a data set.
* Provide a graphical interface so the user can interact with the product.
* Create views to interpret data sets in particular ways.
* Fix any error found on the data set uploaded.
* Provide a graphical interface to configure the way a data set is visualized.
* Organize the data by column and criteria (alphabetical, numerical, ascending, and descending).
* Save all the configurations used when creating the visualization for any particular data set. This information will be used as a “view”

What the product does not do:

* Process files that are not formatted as comma-separated values.
* Modify the original file.
* Catch 100% of possible errors in the file provided.
* Fix 100% of detected errors in the file provided.

## Definitions, acronyms and abbreviations

\*.png, GUI, software, \*.csv

## References

## 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 five subsections that defines the product’s perspective and functions, user characteristics, constrains, assumptions and dependencies. Meanwhile the specific requirements section spawns eight subsections covering external interfaces, functions, performance requirements, logical database requirements, design constrains, software systems attributes, organization of the specific requirements and any additional comments.

# Overall description

## Product perspective

PlasmaGraph is a software tool design to make visual representations of data sets. The product provides its user with a GUI to select the data file to be processed, specify how it is going to graph the data and whether or not to save these specifications to use later. PlasmaGraph will also ask the user where to save the image file generated with the provided data.

Because this software is intended to be used by Angel E. González-Lizard, Ph. D when analyzing experiment data from the PUPR plasma laboratory. The specific chart types provided by PlasmaGraph are TBD and they won’t be determined until Angel E. González-Lizard, Ph. D states which charts he uses most but for now the developers will use bar charts, line charts and pie charts.

There are several data visualization software products in the market like Tableau and Visua.ly but they are expensive and come with too many options which results in a steeper learning curve for the person using the software. PlasmaGraph in the other hand is specialized to meet the client’s requirements making it simpler yet easier to use.

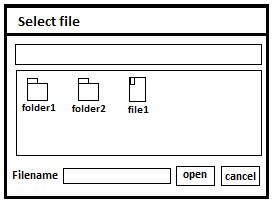
### System Interfaces

PlasmaGraph is a composite of interfaces working together in order to transform a set of data into an image that is a graphical representation of that data. The two predominant interfaces of the user interface and the software interface:

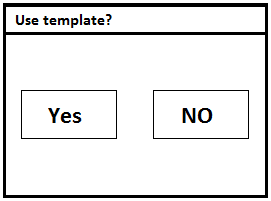
The user interface (also referred as GUI) is the one that interacts with the person using the program. It is made of a series of windows populated with fields and buttons that enable the user to tell the program what to do.

The software interface won’t be accessible to the person using the program. It is made by a collection of functions that receive data from or send data to the GUI in order to perform the operations necessary to generate the graph or report the problem.

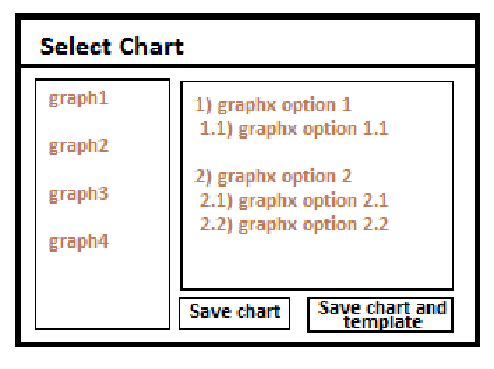
### User Interfaces



When PlasmaGraph starts, the user is going to see a window asking him/her to select a file. Using this window, the user can select the file containing the data that needs to be graphed.



Once the user selects the file, another window is displayed asking if a view is going to be used. When the selection is made, the system proceeds to check the file for problems. If a problem is found, a message with a brief description of the problem is shown to the user. If the option selected was ‘Yes’, the system will provide another window with a list of charts to select from and once the chart is chosen, a set of options is provided inside the window so the user can control how the data is going to be displayed in the chart (charts and their configurations are TBD until a meeting with Angel E. González-Lizard, Ph. D is scheduled to resolve this problem). If the option selected when asked to use a view is ‘No’, then the same window with a list of charts will appear but the chart and all its options will be already selected. This last window also has two buttons, one to save the chart and another to save the chart and the options used as new view.



When the user proceeds to save the chart/view, the system will use the data provided to make an image file and display a window asking the user where to save the image. If a problem occurs, then a window with a brief description of the problem is shown and the PlasmaGraph program is terminated.

### Hardware Interfaces

The minimum hardware requirements for PlasmaGraph are a monitor, keyboard and mouse and of course a computer with enough memory (specified in section 2.1.5).

Because PlasmaGraph runs in the Java VM, the hardware interface is defined by Java

### Software Interfaces

PlasmaGraph is going to be interacting with other systems in order to accomplish its functional requirements. These interactions are handled by the following interfaces:

* Operative System Interface
* Database Abstraction Interface

### Memory

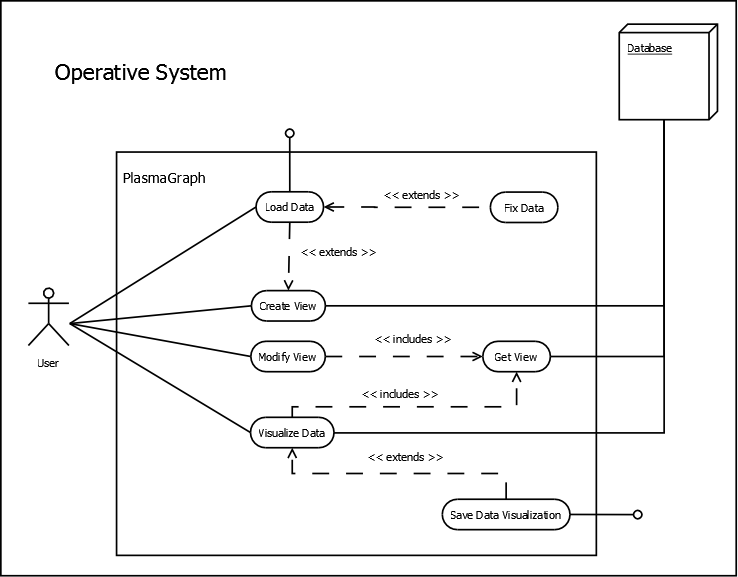
TBD (for this section to be determined, a prototype of PlasmaGraph should be evaluated first)

### Site Adaptation Requirements

For PlasmaGraph to function, Java must be installed in the current operative system.

## Product functions

PlasmaGraph is able to load a file so it can make visual representations of its data, it can fix invalid data and modify how the data is organized and interpreted by creating a “view”. PlasmaGraph can also store and retrieve views from a Database and save the graphics generated into the operative system.



## User characteristics

The product is used by Angel E. González-Lizard, Ph. D as a tool to analyze experiment’s data from the plasma laboratory at the Polytechnic University of Puerto Rico. The user needs to know how to interact with a window based interface and how to read the chart types provided.

## Constraints

The product will only recognize a handful of problems with the data provided by the user and even then it won’t be able to fix all the problems (specifics of these problems are TDB until a meeting is scheduled with Angel E. González-Lizard, Ph. D to resolve the issue).

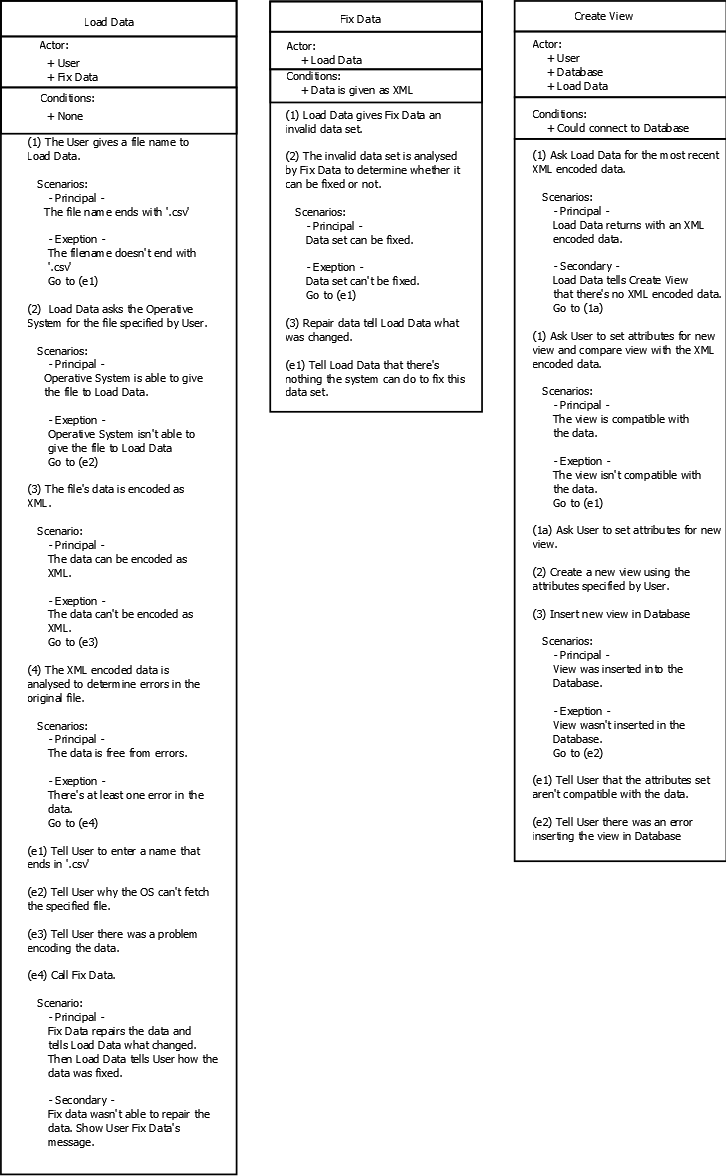
## Assumptions and dependencies

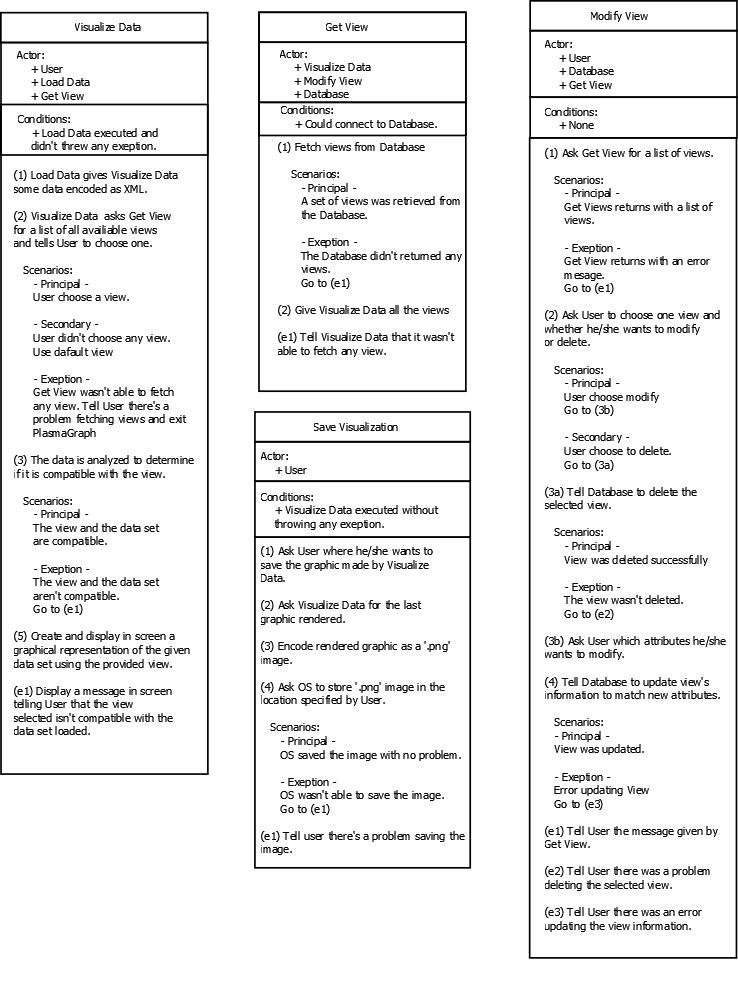
For PlasmaGraph to execute, Java VM must be running

# Specific requirements

## External interfaces

## Functions





## Performance requirements

## Logical database requirements

## Design constraints

### Standards compliance

## Software system attributes

### Reliability

### Availability

### Security

### Maintainability

### Portability

## Organizing the specific requirements

### System mode

### User class

### Objects

### Feature

### Stimulus

### Response

### Functional hierarchy

## Additional comments