# 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 analyzing a file with tabular data and produce a visual representation of the numbers within that file. In other words, a program with a graphical user interface that makes chart graphs.

Chart graphs help people spot patterns easier and in less time. But because there are many ways of making a graph with the same data, PlasmaGraph is going to need an additional file that specifies the following missing information:

* Name of the graphic
* Horizontal headers
* Vertical headers
* Columns
* Data order (ascending /descending)
* Type of chart (bar, line, pie)

This file will be referred to as “template” and each time a new one is created, it will be saved in PlasmaGraph’s data base so it can be used later (note that templates can also be modified or deleted). Also, once the chart graph is created, the person using this product will be able to save it in his or her local file system.

In addition to all this, PlasmaGraph is able to catch and fix certain errors in the data that needs to be represented visually.

What the product does:

* Create a chart graph.
* Provide a graphical interface so the user can interact with the product.
* Create and save templates.
* Modify or delete templates.
* Alert about possible errors in the data that has to be graphed.
* Fix error or suggest solutions.

What the product does not do:

* Process files that are not formatted as comma-separated values.
* Catch 100% of possible errors.
* Fix 100% of detected errors.

## 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

PlasmaGraph is a software tool that can read a file with tabular data and create a chart graph. Because the tabular data alone won’t be enough information to make the chart graph; PlasmaGraph also uses other files called templates. Templates have this additional information needed to make the graph.

## Product perspective

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 product. PlasmaGraph in the other hand is specialized to meet the client’s requirements making it simpler and easier to use.

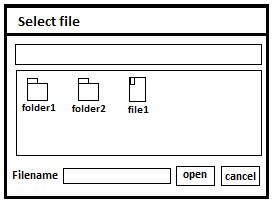
### System Interfaces

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 are the user interface and the software interface:

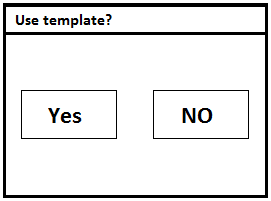
The user interface (also referred as GUI) is the one that interacts with the person using the program. It is made of a series of windows populated with fields and buttons that enable the user to tell the program what to do.

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.

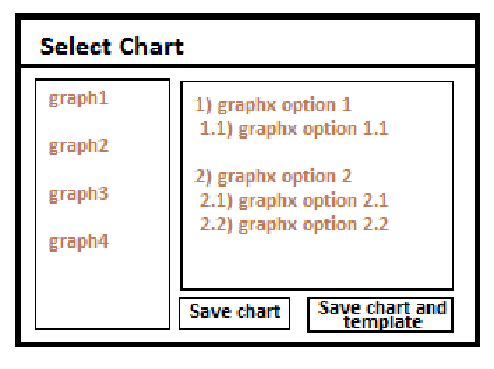
### User Interfaces



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 template 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 template 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 template.



When the user proceeds to save the chart/template, 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).

### Software Interfaces

PlasmaGraph runs on Java 7 and uses its standard class library as well as OpenCSV, JavaX.xml and JFreeChart.

### Memory

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

### Site Adaptation Requirements

For PlasmaGraph to function, Java 7 must be installed and running.

## Product functions

This product performs the following functions:

* Load a file
* Create chart graph
* Create template
* Save chart graph
* Save template
* Modify/Delete template

[use case dia]

## 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 errors in the data provided by the user and even then it won’t be able to fix all the problems (specifics of these problems are TBD until a meeting is scheduled with Angel E. González-Lizard, Ph. D to resolve the issue). It will also limit the type of charts to bar, line and pie and the amount of rows and columns to TBD

## Assumptions and dependencies

For PlasmaGraph to function, Java 7 must be installed and running.

# Specific requirements

## External interfaces

## Functions

[Use case scenarios here]

## 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