Software Design Document

for

PlasmaGraph

Version 0-2014-05-10

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

Polytechnic University of Puerto Rico

May 10, 2014

Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Reason For Changes** |
| January 21, 2014 | 0-01-21-2014 | First version of document. No diagrams, just Introduction. Using SPMP as template. |
| January 23, 2014 | 1-02-23-2014 | Second version of document; added diagrams made up to this date. Provided better structure and removed all mentions of SPMP, including Table of Contents. |
| May 22, 2014 | 1-05-22-2014 |  |

Table of Contents

[1. INTRODUCTION 2](#_Toc388528912)

[1.1 Purpose 2](#_Toc388528913)

[1.2 Scope 2](#_Toc388528914)

[1.3 Overview 2](#_Toc388528915)

[1.4. Reference Materials 2](#_Toc388528916)

[1.5 Definitions and Acronyms 2](#_Toc388528917)

[2. SYSTEM OVERVIEW 2](#_Toc388528918)

[3. SYSTEM ARCHITECTURE 2](#_Toc388528919)

[3.1 Architectural Design 2](#_Toc388528920)

[3.2 Decomposition Description 2](#_Toc388528921)

[A. Package Contents 2](#_Toc388528922)

[B. Requirement Fulfillment 2](#_Toc388528923)

[C. Composition Diagrams 2](#_Toc388528924)

[3.3 Design Rationale 2](#_Toc388528925)

[4. DATA DESIGN 2](#_Toc388528926)

[4.1 Data Description 2](#_Toc388528927)

[4.2 Data Dictionary 2](#_Toc388528928)

[5. COMPONENT DESIGN 2](#_Toc388528929)

[6. HUMAN INTERFACE DESIGN 2](#_Toc388528930)

[6.1 Overview of User Interface 2](#_Toc388528931)

[6.2 Screen Images 2](#_Toc388528932)

[6.3 Screen Objects and Actions 2](#_Toc388528933)

[7. REQUIREMENTS MATRIX 2](#_Toc388528934)

[8. APPENDICES 2](#_Toc388528935)

List of Figures

[Figure 1: “org.pvg.plasmagraph" Package Contents 2](#_Toc388528879)

[Figure 2: "”org.pvg.plasmagraph.controllers” Package Contents 2](#_Toc388528880)

[Figure 3: "”org.pvg.plasmagraph.models” Package Contents 2](#_Toc388528881)

[Figure 4: "”org.pvg.plasmagraph.tests” Package Contents 2](#_Toc388528882)

[Figure 5: "”org.pvg.plasmagraph.utils” Package Contents 2](#_Toc388528883)

[Figure 6: "”org.pvg.plasmagraph.utils.data” Package Contents 2](#_Toc388528884)

[Figure 7: "”org.pvg.plasmagraph.utils.data.readers” Package Contents 2](#_Toc388528885)

[Figure 8: "”org.pvg.plasmagraph.utils.graphs” Package Contents 2](#_Toc388528886)

[Figure 9: “org.pvg.plasmagraph.utils.template” Package Contents 2](#_Toc388528887)

[Figure 10: "”org.pvg.plasmagraph.utils.tools” Package Contents 2](#_Toc388528888)

[Figure 11: “org.pvg.plasmagraph.utils.interpolation” Package Contents 2](#_Toc388528889)

[Figure 12: “org.pvg.plasmagraph.utils.tools.outlierscan” Package Contents 2](#_Toc388528890)

[Figure 13: “org.pvg.plasmagraph.utils.tools.outlierscan.distances” Package Contents 2](#_Toc388528891)

[Figure 14: “org.pvg.plasmagraph.utils.tools.outlierscan.methods” Package Contents 2](#_Toc388528892)

[Figure 15: “org.pvg.plasmagraph.utils.tools.types” Package Contents 2](#_Toc388528893)

[Figure 16: “org.pvg.plasmagraph.utils.views” Package Contents 2](#_Toc388528894)

[Figure 17: “HelpManual” Package Contents 2](#_Toc388528895)

[Figure 18: “Import Data” Sequence Diagram 2](#_Toc388528896)

[Figure 19: “Validate Data” Sequence Diagram 2](#_Toc388528897)

[Figure 20: “Choose Graph Options” Sequence Diagram 2](#_Toc388528898)

[Figure 21: “Create Graph” Sequence Diagram 2](#_Toc388528899)

[Figure 22: “Save Graph” Sequence Diagram 2](#_Toc388528900)

[Figure 23: “Save Template” Sequence Diagram 2](#_Toc388528901)

[Figure 24: “Import Template” Sequence Diagram 2](#_Toc388528902)

[Figure 25: “Inspect Data” Sequence Diagram 2](#_Toc388528903)

[Figure 26: “Display Help” Sequence Diagram 2](#_Toc388528904)

[Figure 27: “Choose Location” Sequence Diagram 2](#_Toc388528905)

[Figure 28: “Elicit Graph Options” Sequence Diagram 2](#_Toc388528906)

[Figure 29: Package Usage Flow Diagram 2](#_Toc388528907)

[Figure 30: Main MVC Relationship Diagram 2](#_Toc388528908)

[Figure 31: Data Set MVC Relationship Diagram 2](#_Toc388528909)

[Figure 32: Tool MVC Relationship Diagram 2](#_Toc388528910)

[Figure 33: Graph MVC Relationship Diagram 2](#_Toc388528911)

List of Tables

[Table 1: Document Definitions 2](#_Toc388528873)

[Table 2: Document Acronyms 2](#_Toc388528874)

[Table 3: Requirements Matrix 2](#_Toc388528875)

# 1. INTRODUCTION

## 1.1 Purpose

This document, the Software Design Document, is designed to describe the architectural and system details of the PlasmaGraph product to all whom are interested in understanding the design decisions made in the creation of this product; specifically, this document is made for the students working at the Polytechnic University of Puerto Rico's Plasma Laboratory and any future contributors to the PlasmaGraph project. It explains what the program does and why it does it in order to provide an accurate synopsis of how the functional and non-functional requirements are achieved. Diagrams and pseudo-code will be both used in order to appropriately and concisely detail how PlasmaGraph will satisfy the requirements listed in the System Requirement Specifcation (SRS) document. In doing so, readers will obtain a close-to-exact idea of how the program will function without having to look at the code itself.

## 1.2 Scope

The project, currently by the Plasma Visualization Group, is designed to create a product called PlasmaGraph. PlasmaGraph is a graphing solution specifically made to create graphs from the PUPR Plasma Laboratory's Matlab-encoded data files. It allows the user to select the Matlab-encoded data file to use, select the graphs columns and other visual settings, generate regressions based on one or more groups of data, and save the resulting graphs. PlasmaGraph is to be used at the end of the experiment analysis pipeline of the Plasma Laboratory in order to visualize experiment data tendencies without resorting to tools that require programming knowledge, such as Matlab. This product would, therefore, be indispensable to both new members of the Plasma Laboratory work team and older members who want to quickly view and understand experiment data.

## 1.3 Overview

The SDD is divided into various sections, each emphasizing an aspect of PlasmaGraph. Section 2 will give a broad explanation of the project’s background, providing the historical backdrop, before continuing on to explain the product’s architectural design and necessary functionality. From there, the next sections will discuss the architectural details and rationale for the final design over other potential choices that were thought of before the final design was developed. Once the architecture has been thoroughly explained, this document will discuss what data structures exist in this product, and how they further the requirements as presented in the Software Requirements Specification (SRS), and the primary components of the program that perform the vital functions necessary for its proper functioning. The program’s Graphical User Interface’s design will then be explained, describing the tools it provides any potential users interested in creating graphs. Finally, the Requirements Matrix Section will provide a summary of what sections of the program’s code further certain requirements.

## 1.4. Reference Materials

|  |  |
| --- | --- |
| [1] | Object Refinery Limited, "JFreeChart," Object Refinery Limited, 25 November 2013. [Online]. Available: http://www.jfree.org/jfreechart/. [Accessed 13 May 2014]. |
| [2] | The MathWorks, Inc., "MATLAB - The Language of Technical Computing," [Online]. Available: http://www.mathworks.com/products/matlab/?s\_tid=hp\_fp\_ml. [Accessed 12 May 2014]. |
| [3] | Oracle Corporation, "What is Java and why do I need it?," 25 March 2014. [Online]. Available: http://www.java.com/en/download/faq/whatis\_java.xml. [Accessed 13 May 2014]. |
| [4] | Merriam-Webster, "Dictionary and Thesaurus - Merriam-Webster Online," [Online]. Available: http://www.merriam-webster.com/. [Accessed 12 May 2014]. |
| [5] | B. Goines, S. Chacon and M. McCullough, "Git - About Version Control," [Online]. Available: http://git-scm.com/book/en/Getting-Started-About-Version-Control. [Accessed 13 May 2014]. |
| [6] | PC.net, "Definition of IDE," [Online]. Available: http://pc.net/glossary/definition/ide. [Accessed 13 May 2014]. |
| [7] | Polytechnic University of Puerto Rico, "Polytechnic University of Puerto Rico - Main Page," [Online]. Available: http://www.pupr.edu/. [Accessed 13 May 2014]. |
| [8] | Institute of Electrical and Electronics Engineers, "IEEE Standard for Software Project Management Plans," 22 December 1998. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=741937&isnumber=16012. [Accessed 8 May 2014]. |
| [9] | Institute of Electrical and Electronics Engineers, "IEEE Recommended Practice for Software Requirements Specifications," 22 December 1998. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720574&isnumber=15571. [Accessed 13 May 2014]. |
| [10] | Institute of Electrical and Electronics Engineers, "IEEE Recommended Practice for Software Design Descriptions," 22 December 1998. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=741934&isnumber=16019. [Accessed 13 May 2014]. |
| [11] | Institute of Electrical and Electronics Engineers, "IEEE Standard for Software Test Documentation," 22 December 1998. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=741968&isnumber=16010. [Accessed 13 May 2014]. |
| [12] | GitHub, Inc., "About - GitHub," [Online]. Available: https://github.com/about. [Accessed 12 May 2014]. |
| [13] | "How to Write Doc Comments for the Javadoc Tool," 16 November 2012. [Online]. Available: http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html. [Accessed 13 May 2014]. |
| [15] | Git SCM, "Git - GUI Clients," [Online]. Available: http://git-scm.com/downloads/guis. [Accessed 14 May 2014]. |
| [16] | R. C. Martin, "UML Java Programmers," 10 December 2008. [Online]. Available: http://www.csd.uoc.gr/~hy252/references/UML\_for\_Java\_Programmers-Book.pdf. [Accessed 24 04 2014]. |
| [17] | t. t. w. cbwatcham, "JMatIO - Matlab's MAT-file I/O in JAVA," 04 December 2012. [Online]. Available: http://sourceforge.net/projects/jmatio/. [Accessed 14 May 2014]. |
| [18] | Apache Software Foundation, "Math - Commons Math: The Apache Commons Mathematics Library," 02 November 2013. [Online]. Available: http://commons.apache.org/proper/commons-math/index.html. [Accessed 12 February 2014]. |
| [19] | Oracle Corporation, "Java SE Application Design With MVC," March 2007. [Online]. Available: http://www.oracle.com/technetwork/articles/javase/index-142890.html. [Accessed 15 May 2014]. |
| [20] | R. C. Martin, UML for Java Programmers, Upper Saddle River, NJ: Prentice Hall PTR, 2003. |
| [21] | S. Ramanujan, "Software Design Document (SDD) Template," 7 May 2014. [Online]. Available: http://cmsu2.ucmo.edu/public/classes/sam/Advanced%20Systems%20Project/sdd\_template.pdf. |

## 1.5 Definitions and Acronyms

Unless otherwise specified, all definitions and acronyms are specific to this project’s scope.

Table : Document Definitions

| Term | Definition |
| --- | --- |
| PlasmaGraph / Product | A Matlab-file error-checking graphing program made for the PUPR Plasma Laboratory. |
| Client | Professor Angel Gonzales-Lizardo; the person requesting the project. |
| Advisor | University-designated overseer of the project's progress. |
| JFreeChart [1] | A set of tools written in the “Java” programming language that create graphical representations of data provided to it. |
| Matlab [2] | Data-manipulation and graphing IDE made by The Mathworks, Inc. |
| Java [3] | Object-oriented, interpreted programming language known for its portability between multiple operating systems and general-purpose capabilities. |
| Operating System [4] | Software that controls the operation of a computer and directs the processing of programs (as by assigning storage space in memory and controlling input and output functions). |
| Project Team / Team | The creators and maintainers of the project and all of its end products. Composed of Gerardo A. Navas Morales and Daniel E. Quintini Greco. |
| Version Control System [5] | Manages how multiple users can access and change the same files without losing data. Also known as a Revision Control System or Source Control System |
| Integrated Development Environment [6] | Software development program that keeps track of all files related to a project and provides a central interface for writing source code, linking files together, and debugging the software. |
| Vetting work packages | The process of reviewing a work package for semantical errors, documentation requirements (if a document), or programming errors (if a program). |
| Program Documentation | All Javadoc files and the User Manual. |
| Project Documentation | The SRS, SDD, STD, and SPMP documents. |
| Polytechnic University  of Puerto Rico [7] | University where the Plasma Laboratory is located, as well as where the group's members study. |
| Software Project Management Plan / SPMP [8] | This document; One of the four IEEE project documents being created as part of this project. |
| Software Requirements Specification / SRS [9] | One of the four IEEE project documents being created as part of this project. |
| Software Design Descriptions / SDD [10] | One of the four IEEE project documents being created as part of this project. |
| Software Test Documentation / STD [11] | One of the four IEEE project documents being created as part of this project. |
| Graphical User Interface / GUI [4] | A computer program designed to allow a computer user to interact easily with the computer typically by making choices from menus or groups of icons. |
| Development | Creation of product code. |
| PNG / .png / JPG / JPEG /  .jpg / .jpeg | Types of computer files that hold image data. |
| MAT / .mat | Matlab data file format. |
| Contributor | A GitHub user status; allows the user to make changes to the project. |
| Incomplete document | A document with one or more sections labelled [TBD]. |
| Complete document | A document with no sections labelled [TBD]. |
| Document verification | The process an Advisor performs on a document to check for errors. |
| GitHub [12] | File repository system that uses the Git Version Control System. [12] |
| Class | A file that serves to encapsulate functionality within it. |

Table : Document Acronyms

|  |  |
| --- | --- |
| Term | Acronym |
| PUPR | Polytechnic University of Puerto Rico. |
| SPMP | Software Project Management Plan. |
| SRS | Software Requirement Specification. |
| SDD | Software Design Descriptions. |
| STD | Software Test Documentation. |
| GUI | Graphical User Interface. |
| VCS | Version Control System |
| IDE | Integrated Development Environment. |

# 2. SYSTEM OVERVIEW

## 2.1. System Context and Background

This project is designed to supplant the usage of Matlab [2] in the PUPR Plasma Laboratory for new and old students. A recurring problem in the laboratory is that newer students find themselves unable to easily create graphs of some types of data due to their lack of Matlab knowledge. Similarly, older students find that Matlab is a less effective graphing solution than they should have available. PlasmaGraph is designed to be both easy to use and quick to graph to resolve both of these problems and, thus, aid laboratory students in analyzing data.

## 2.2. System Functionality

The capabilities of the PlasmaGraph program, as defined by the requirements presented by the client in this project’s SRS document, are as follows:

1. Import Matlab Level-5-encoded data files.
2. Validate data files for uneven column sizes or invalid data points.
3. Provide a Graphical User Interface for the user to select options.
4. Create graphs based on user-provided data and options.
5. Save graphs made by the user.
6. Save and import graph’s options for later use.
7. Allow user to view data manually.
8. Provide a help interface to explain the purpose of some functions.

## 2.3. System Design Overview

PlasmaGraph is designed with Object-Oriented principles in mind, and will use the Model-View-Controller (MVC) pattern to implement user interface management. Data will be pulled from files and placed within various objects (listed in Section 4). The GUI will be handled via MVC objects, which will manage the visual positioning of interface objects as well as the signals sent to and from the interface and will form a part of the user settings for the graph. These settings, alongside the previously-mentioned data, will then be used to create graphs. The design of the program will be discussed with further detail in Section 3.

# 3. SYSTEM ARCHITECTURE

## 3.1 Architectural Design

As listed in Section 2.2, PlasmaGraph has a number of features it must contain in order to be considered a full product. These features are based on the functional and non-functional requirements stipulated in PlasmaGraph’s SRS document. To implement these features, the program will group their implementation into three categories:

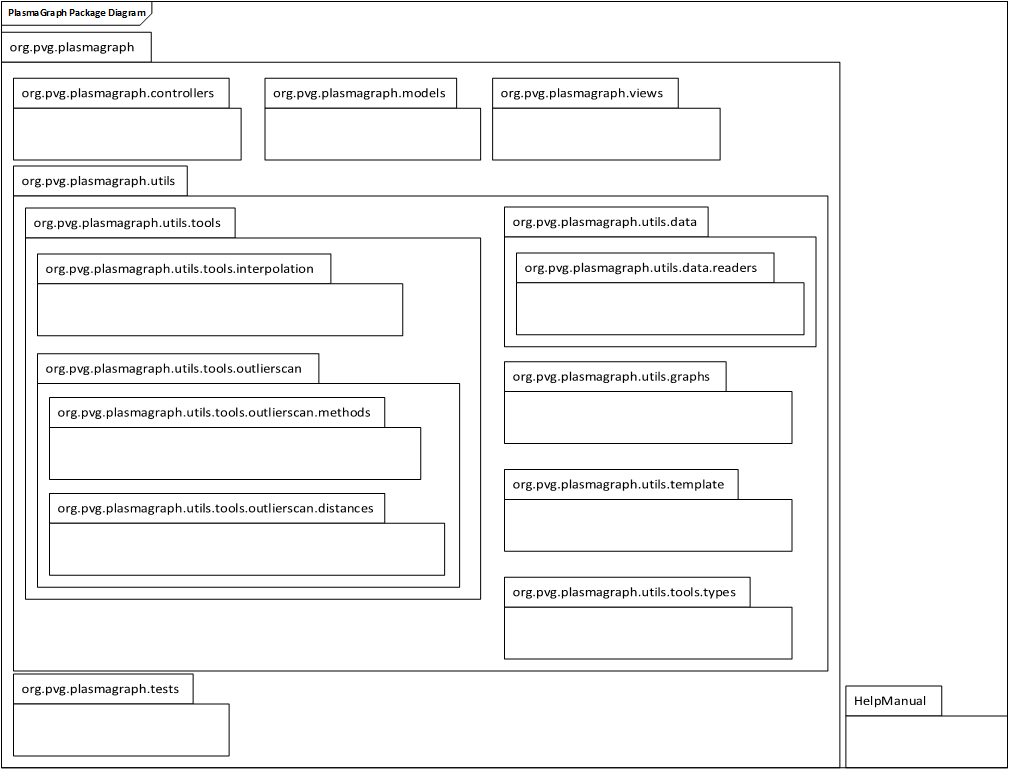
* Graphical User Interface.
* Data-reading and data-storing.
* Data-graphing and optional tools.

In following with the object-oriented design pattern, PlasmaGraph’s features will be divided into objects whose specifications are located in various packages:

* All Class files related to the GUI are divided between the “org.pvg.plasmagraph.controllers“ (Controller classes), “org.pvg.plasmagraph.models“ (Model classes), and “org.pvg.plasmagraph.views“ (View classes) packages.
* Data-Containing and Data-Reading Class files are located in the “org.pvg.plasmagraph.utils.data“ and “org.pvg.plasmagraph.utils.data.readers“ packages.
* Class files involving the Interpolation and Outlier Scanning tools are found in the “org.pvg.plasmagraph.utils.tools.interpolation“ and “org.pvg.plasmagraph.utils.tools.outlierscan“ packages or their sub-packages (“org.pvg.plasmagraph.utils.tools.outlierscan.distances“ and “org.pvg.plasmagraph.utils.tools.outlierscan.methods“).
* Automated test files are found in the “org.pvg.plasmagraph.tests“ package.
* Files containing graph types and their procedures are found under “org.pvg.plasmagraph.utils.graphs“.
* Template files are located under “org.pvg.plasmagraph.utils.template“.
* Class files denoting unique types of options are found under “org.pvg.plasmagraph.utils.types“.
* Files regarding the Help Manual are found in the “HelpManual“ package.
* Most importantly, the file containing the program’s initialization routine is found under “org.pvg.plasmagraph“.

A diagram depicting PlasmaGraph’s package structure is found in the next page.

Figure : PlasmaGraph Package Diagram



The table in Section 7 explains how each of the packages mentioned above help further each of the SRS requirements.

## 

## 3.2 Decomposition Description

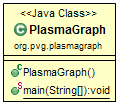
The following sections depict the code contained in the packages mentioned in Section 3.1 by way of describing the classes and how they’re used in implementing PlasmaGraph’s requirements.

### Package Contents

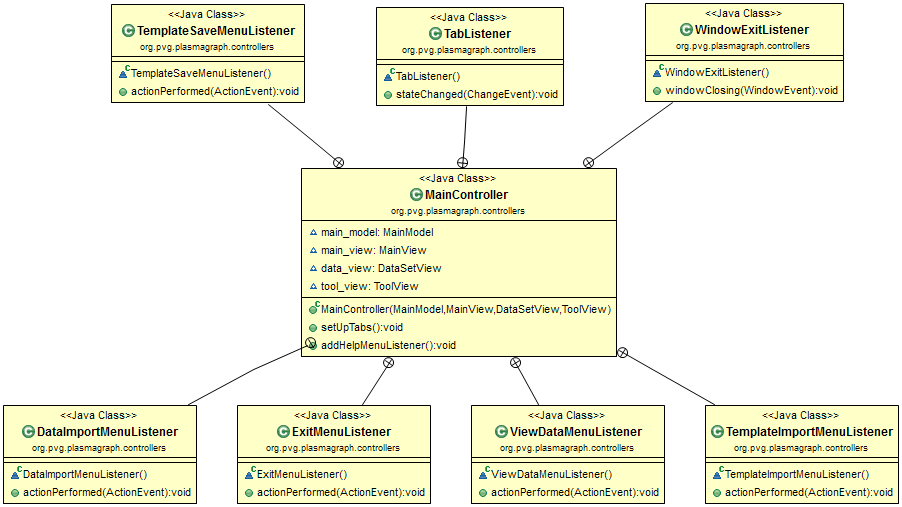
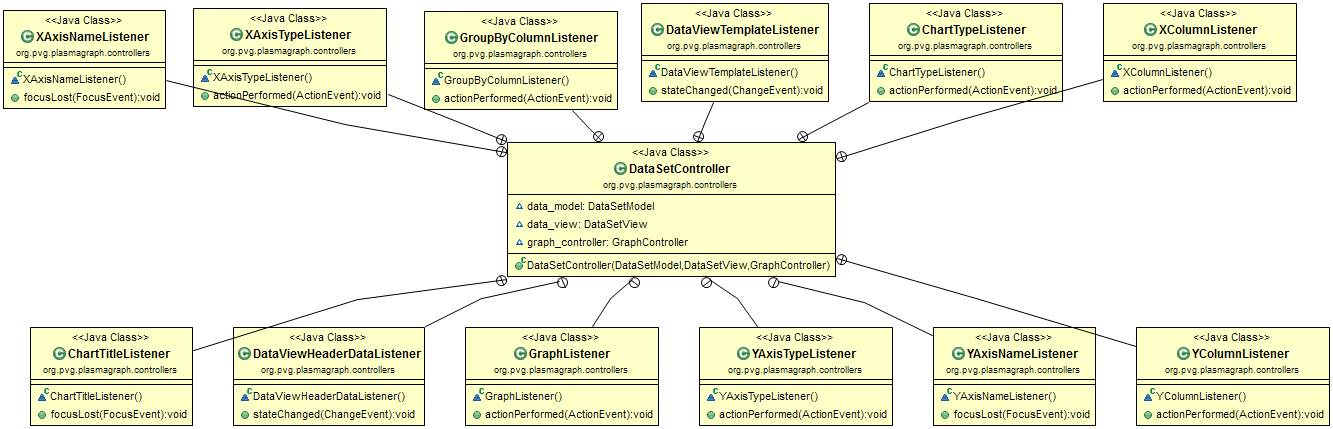
[TBD – Possibly a Legend for the Class Diagrams?]

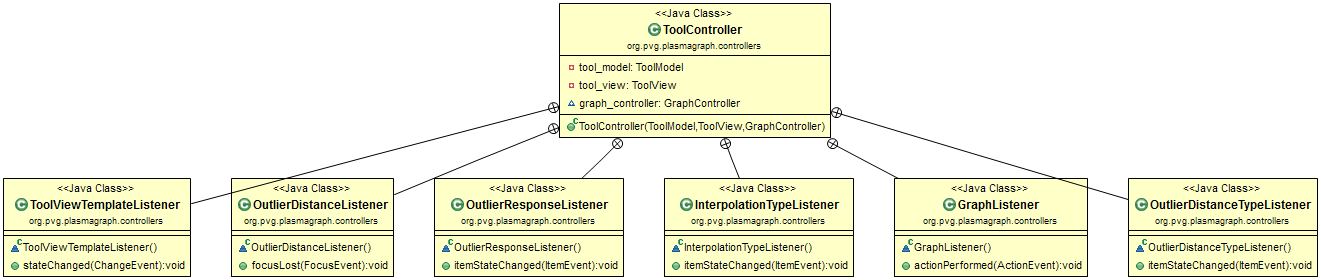
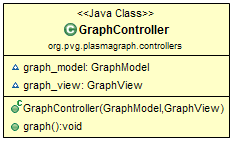
* 1. Package "org.pvg.plasmagraph"

Figure : “org.pvg.plasmagraph" Package Contents

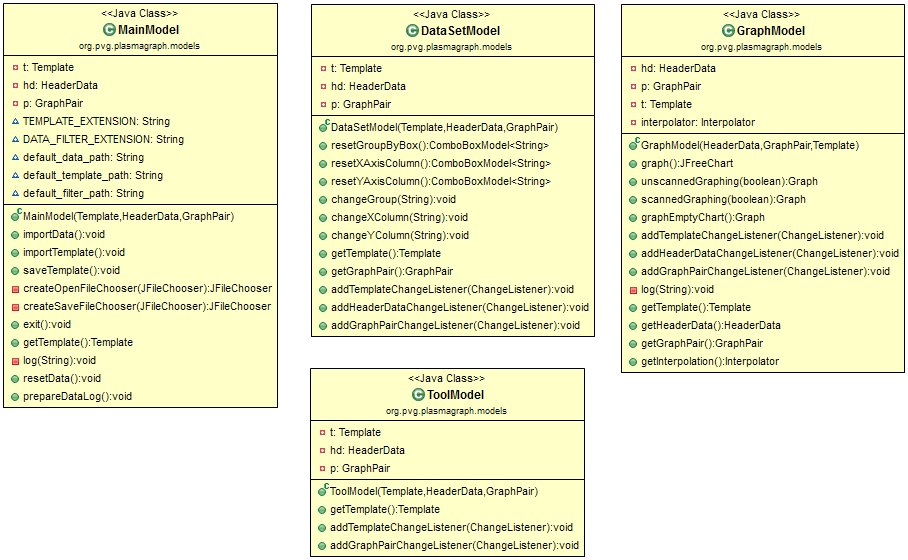


* 1. Package "org.pvg.plasmagraph.controllers"

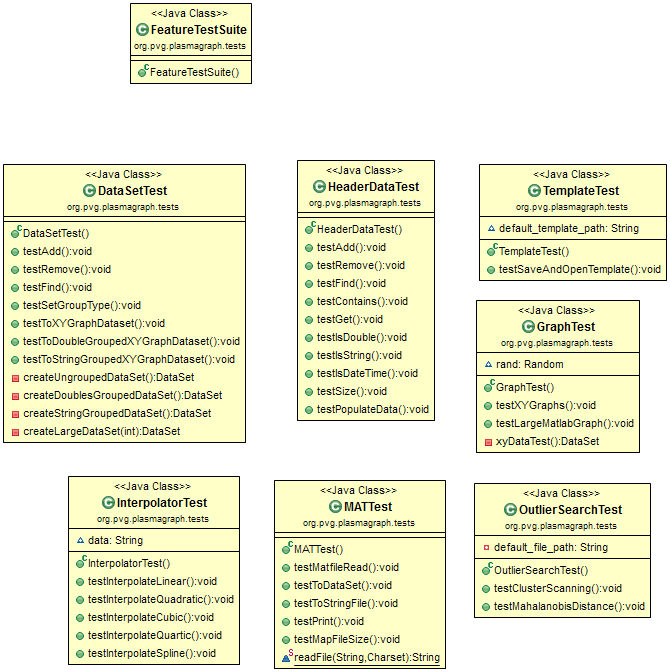
Figure : "”org.pvg.plasmagraph.controllers” Package Contents



* 1. Package "org.pvg.plasmagraph.models"

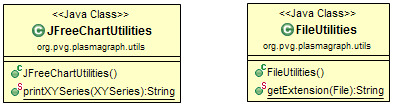
Figure : "”org.pvg.plasmagraph.models” Package Contents

* 1. Package "org.pvg.plasmagraph.tests"

Figure : "”org.pvg.plasmagraph.tests” Package Contents

* 1. Package "org.pvg.plasmagraph.utils"

Figure : "”org.pvg.plasmagraph.utils” Package Contents



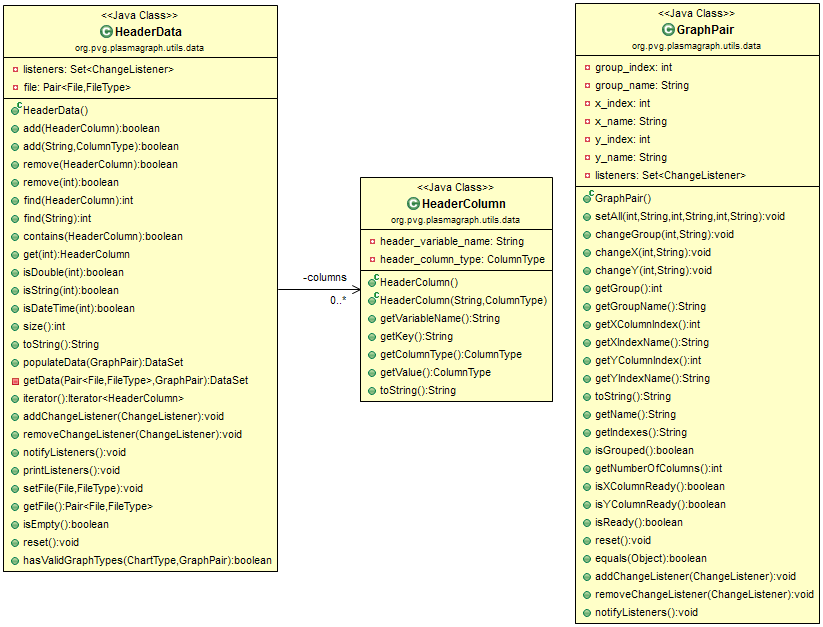
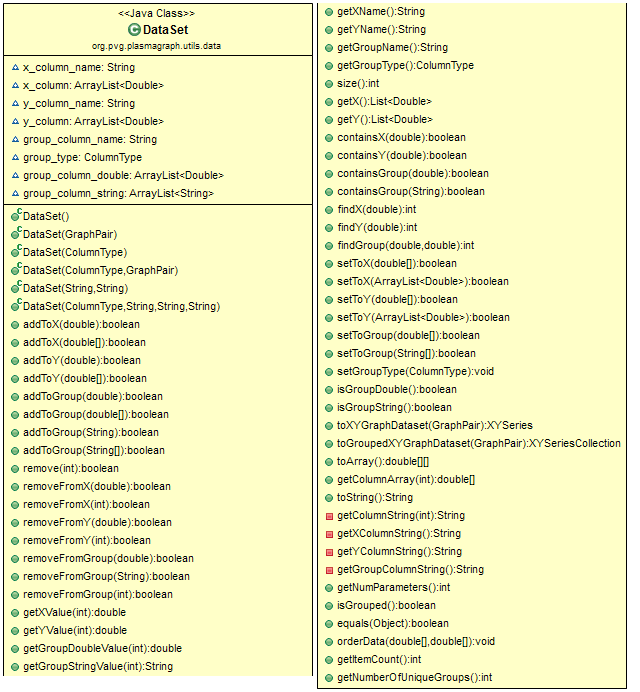
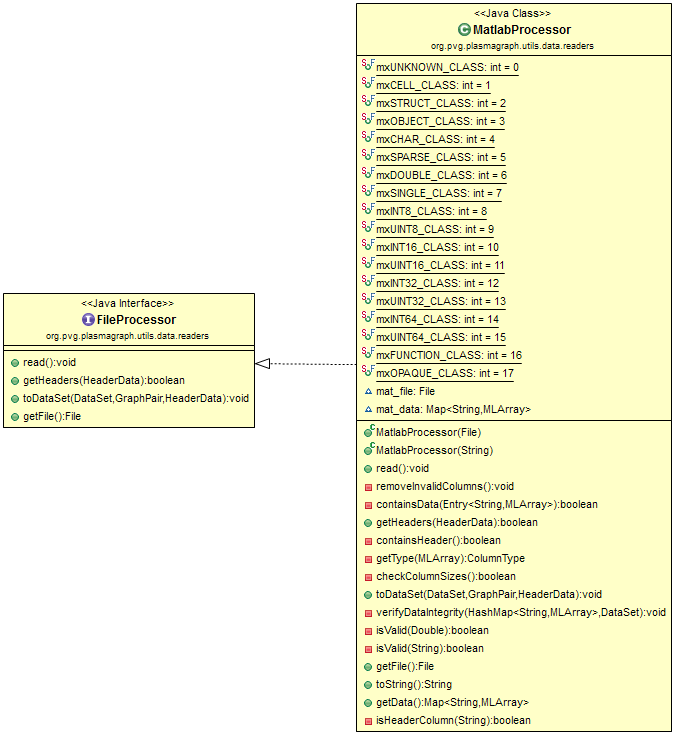
* 1. Package "org.pvg.plasmagraph.utils.data"

Figure : "”org.pvg.plasmagraph.utils.data” Package Contents

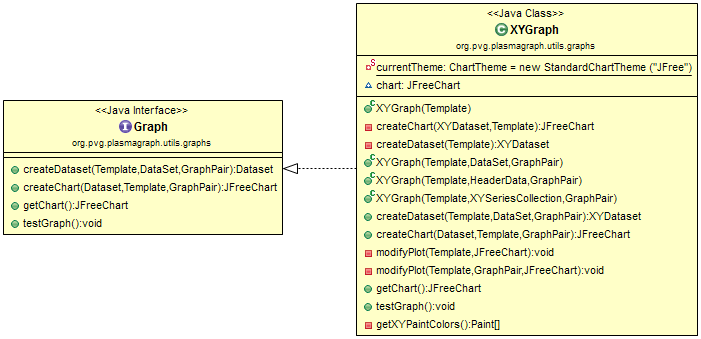


* 1. Package "org.pvg.plasmagraph.utils.data.readers"

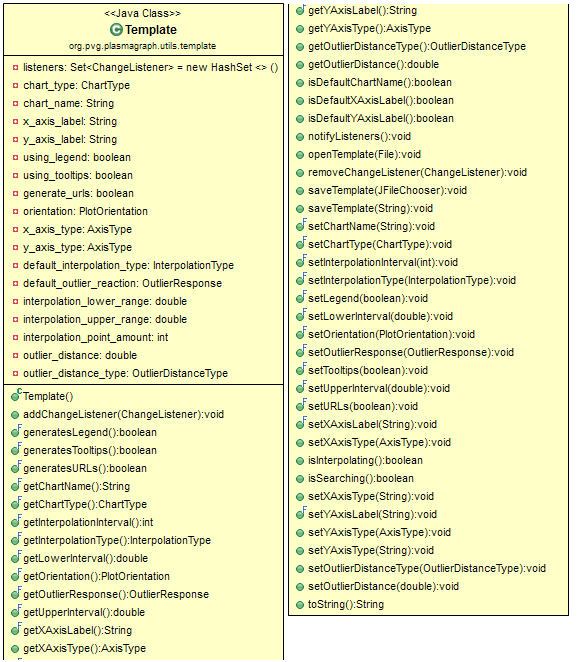
Figure : "”org.pvg.plasmagraph.utils.data.readers” Package Contents

* 1. Package "org.pvg.plasmagraph.utils.graphs"

Figure : "”org.pvg.plasmagraph.utils.graphs” Package Contents

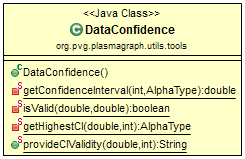


* 1. Package "org.pvg.plasmagraph.utils.template"

Figure : “org.pvg.plasmagraph.utils.template” Package Contents

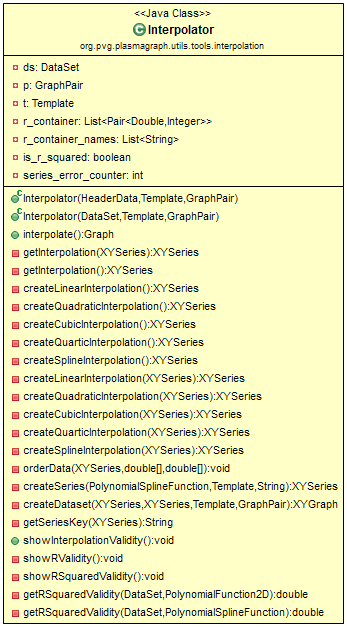
* 1. Package "org.pvg.plasmagraph.utils.tools"

Figure : "”org.pvg.plasmagraph.utils.tools” Package Contents



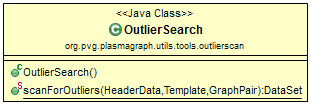
* 1. Package "org.pvg.plasmagraph.utils.tools.interpolation"

Figure : “org.pvg.plasmagraph.utils.interpolation” Package Contents

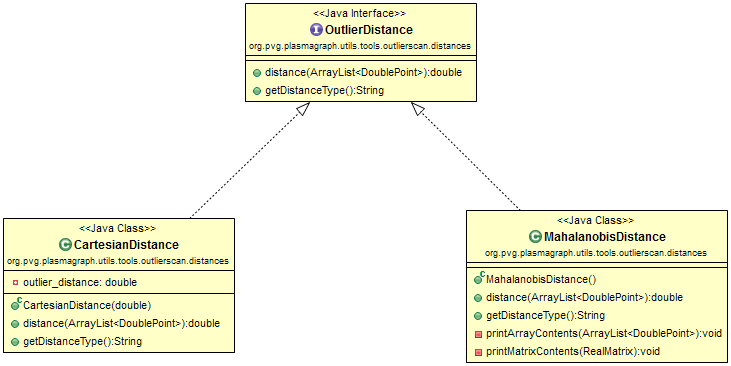


* 1. Package "org.pvg.plasmagraph.utils.tools.outlierscan"

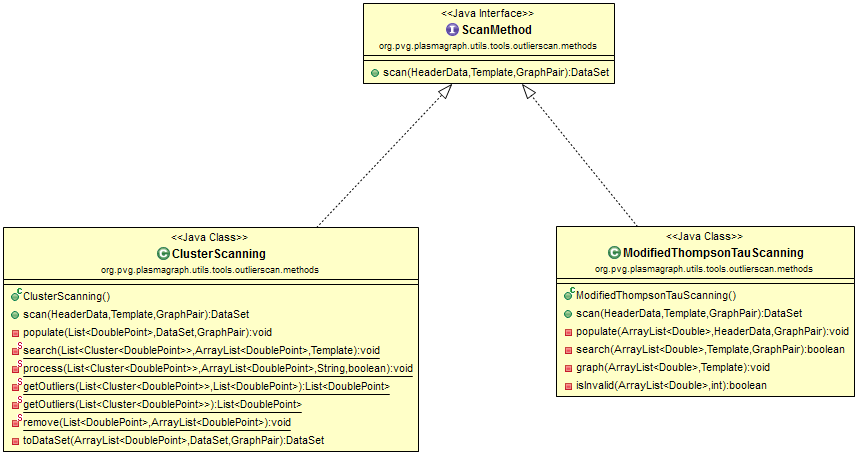
Figure : “org.pvg.plasmagraph.utils.tools.outlierscan” Package Contents



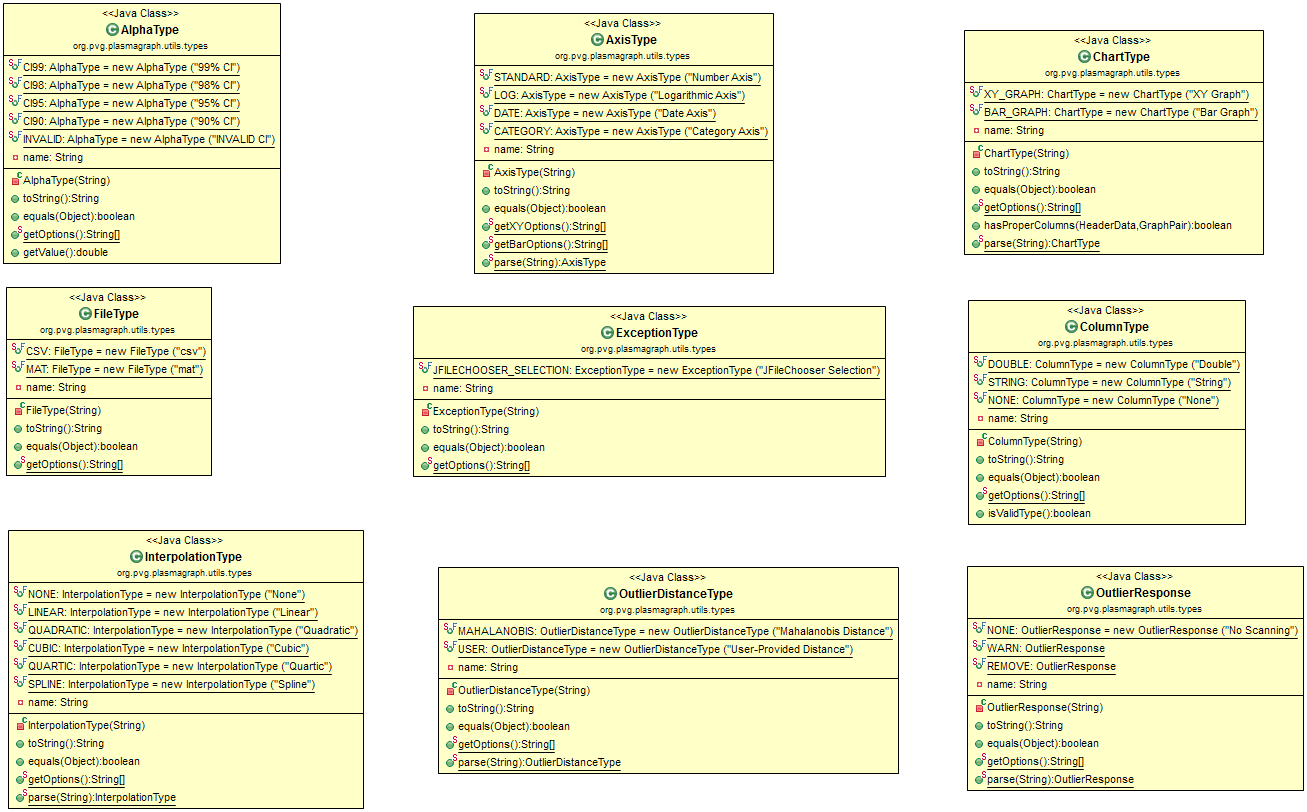
* 1. Package "org.pvg.plasmagraph.utils.tools.outlierscan.distances"

Figure : “org.pvg.plasmagraph.utils.tools.outlierscan.distances” Package Contents

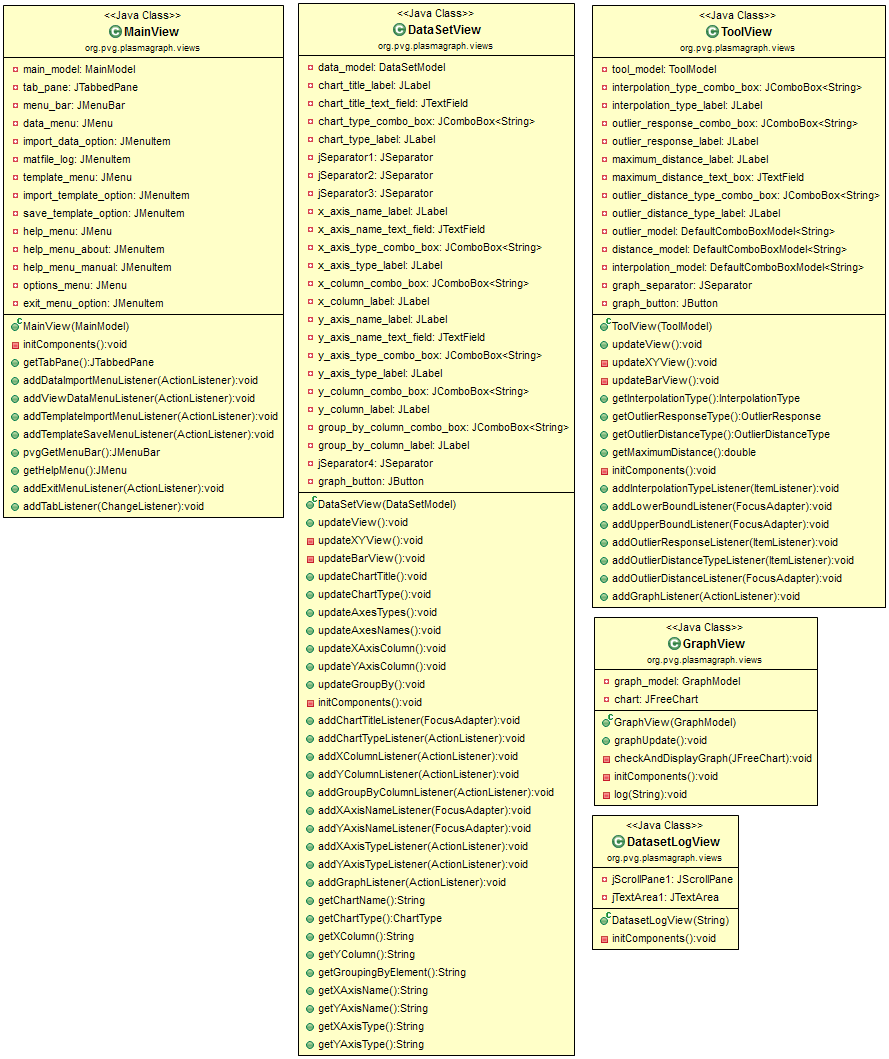
* 1. Package "org.pvg.plasmagraph.utils.tools.outlierscan.methods"

Figure : “org.pvg.plasmagraph.utils.tools.outlierscan.methods” Package Contents

* 1. Package "org.pvg.plasmagraph.utils.tools.types"

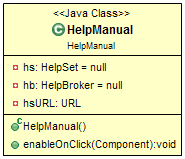
Figure : “org.pvg.plasmagraph.utils.tools.types” Package Contents

* 1. Package "org.pvg.plasmagraph.views"

Figure : “org.pvg.plasmagraph.utils.views” Package Contents

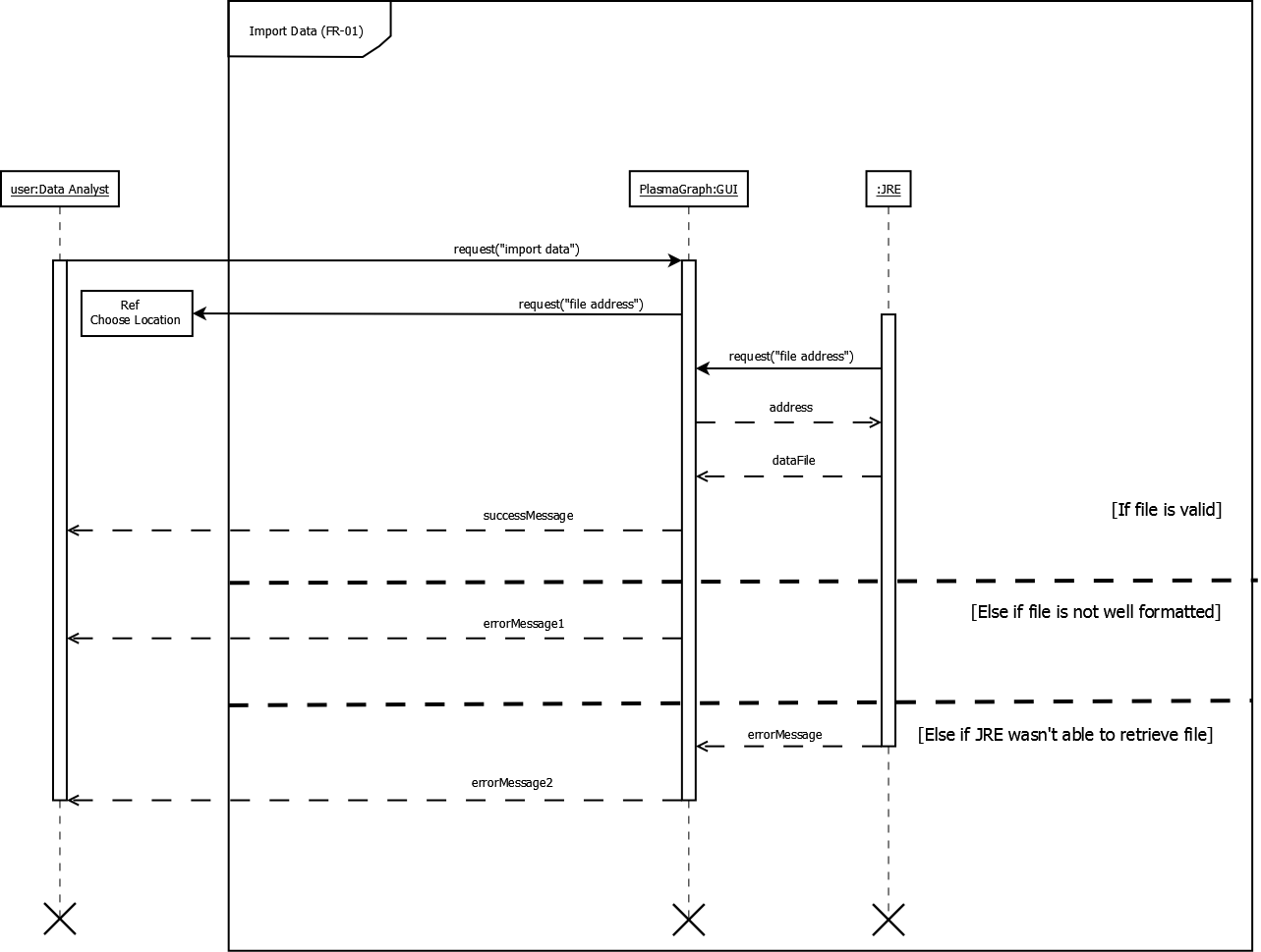
* 1. Package "HelpManual"

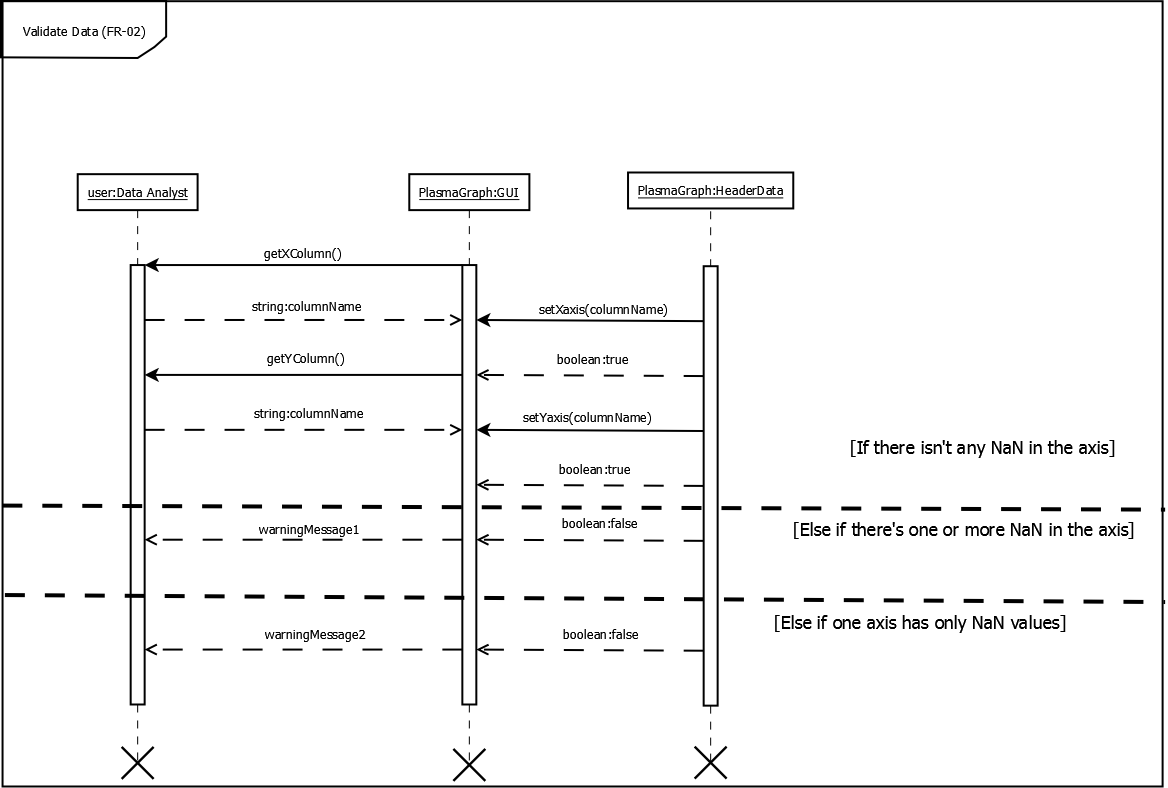
Figure : “HelpManual” Package Contents



### Requirement Fulfillment

#### Functional Requirements

Figure : “Import Data” Sequence Diagram

Figure : “Validate Data” Sequence Diagram

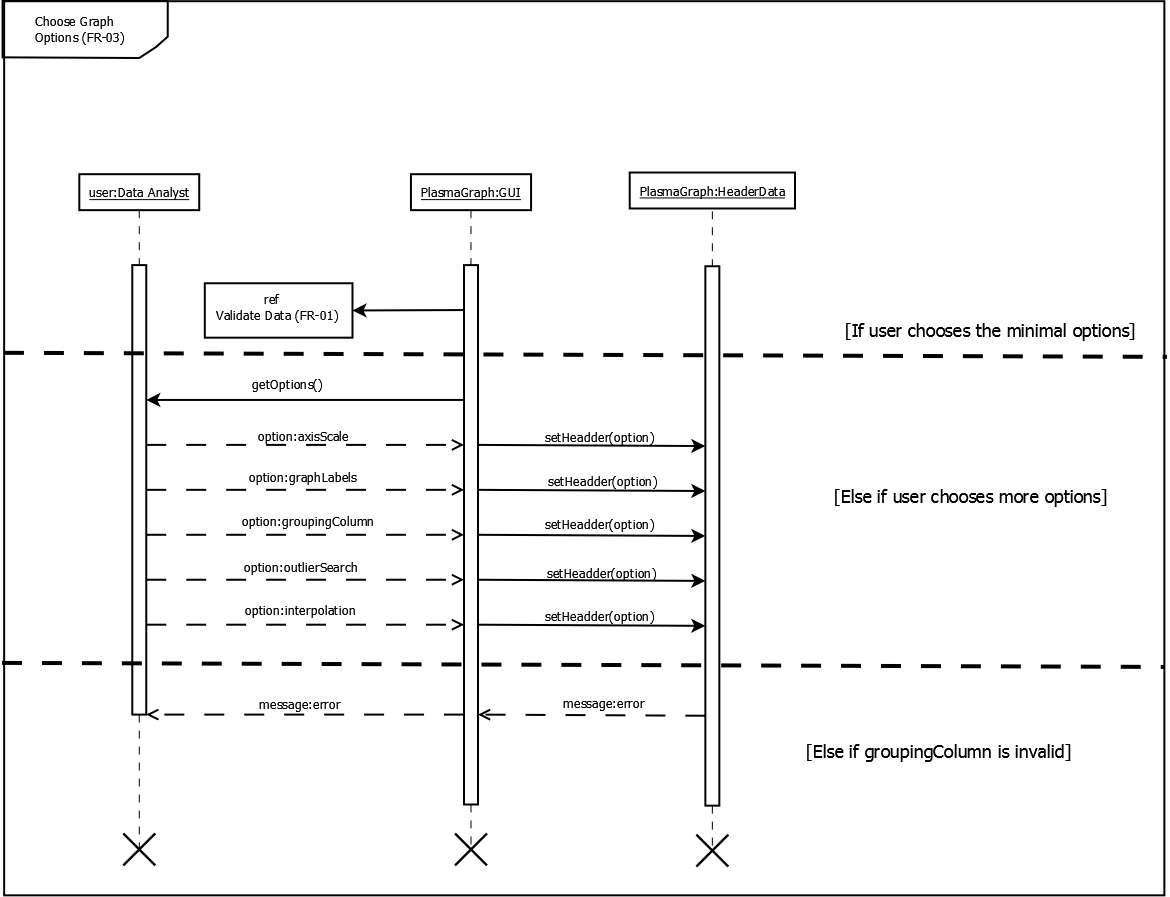
Figure : “Choose Graph Options” Sequence Diagram

Figure : “Create Graph” Sequence Diagram

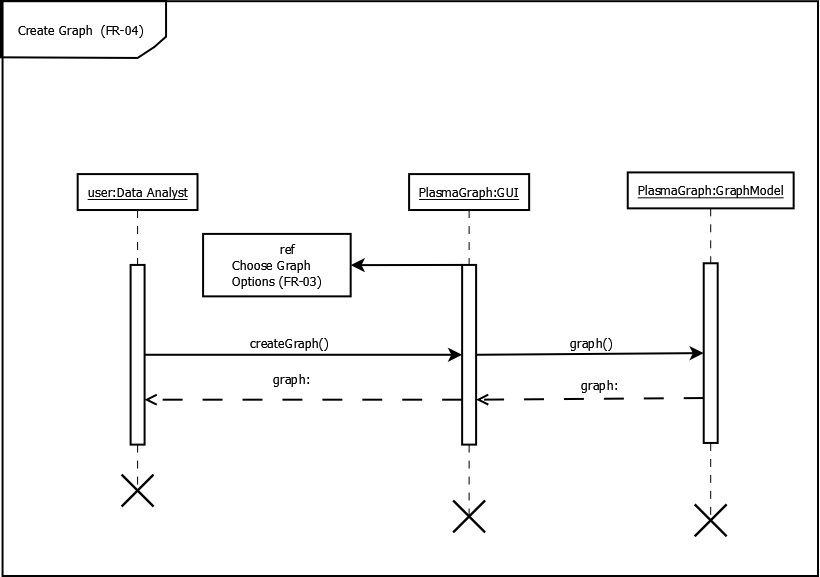
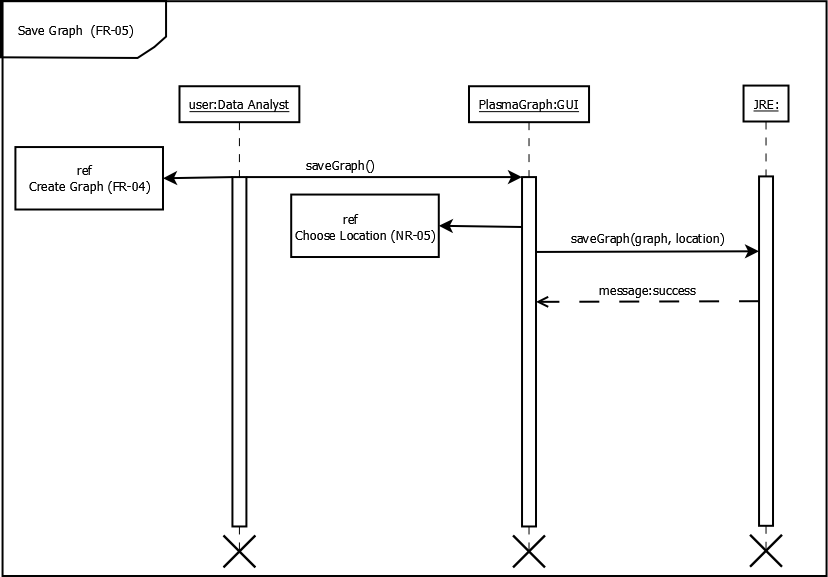


Figure : “Save Graph” Sequence Diagram

#### Non-Functional Requirements

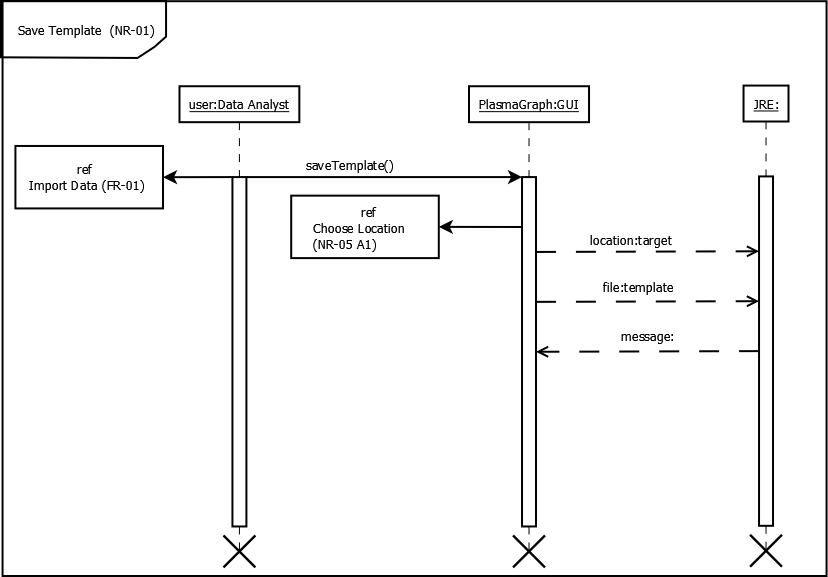
Figure : “Save Template” Sequence Diagram

Figure : “Import Template” Sequence Diagram

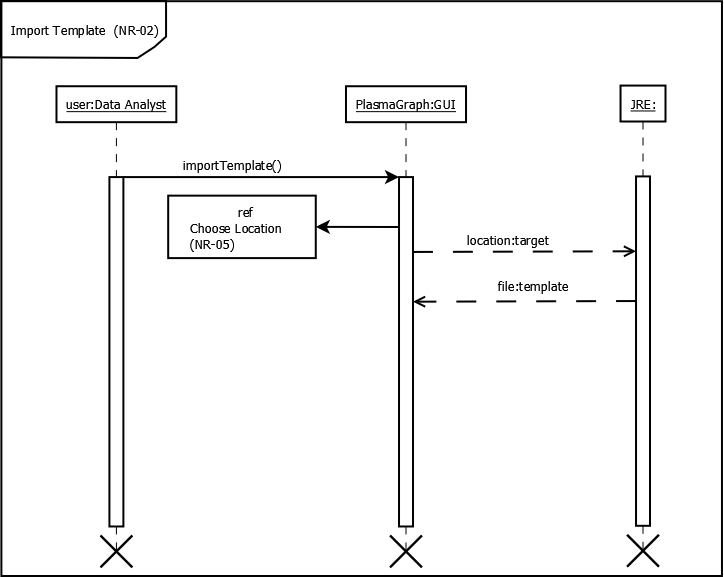


Figure : “Inspect Data” Sequence Diagram

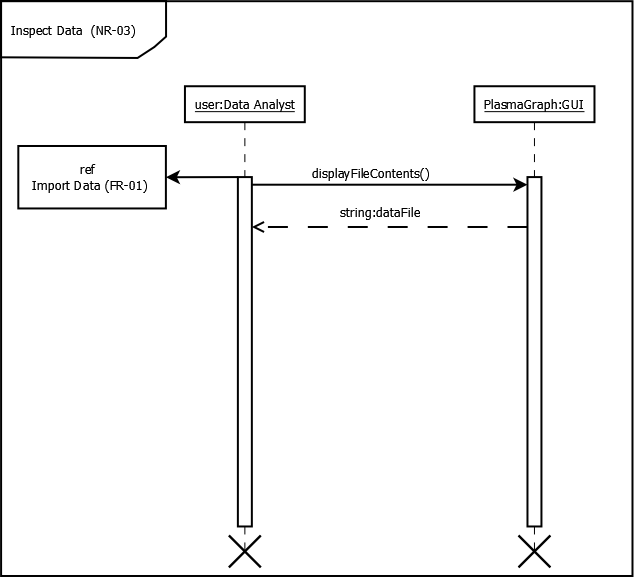


Figure : “Display Help” Sequence Diagram

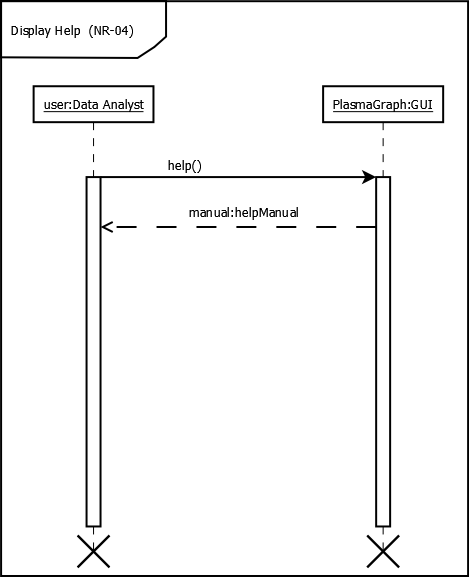


Figure : “Choose Location” Sequence Diagram

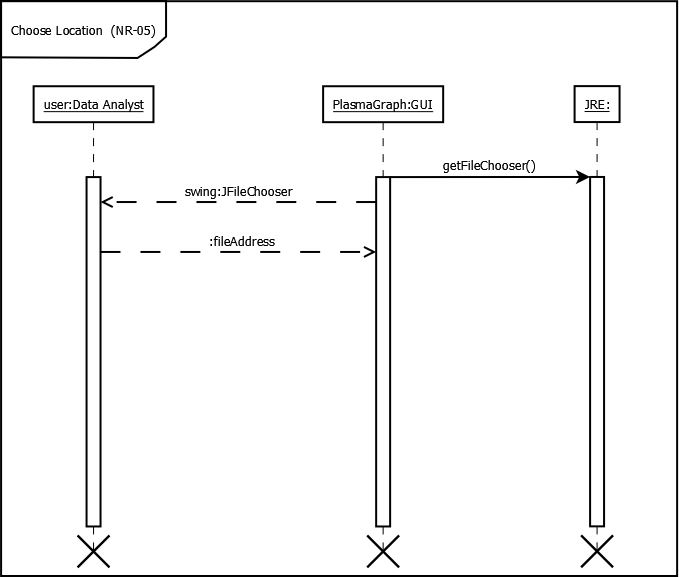
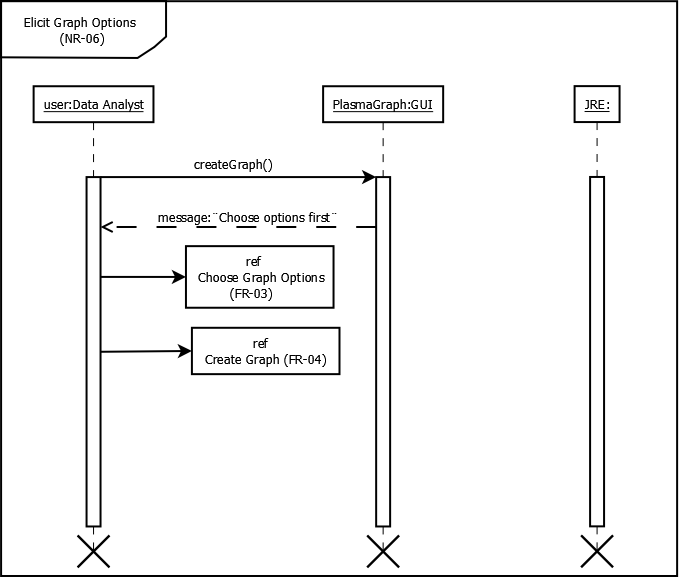


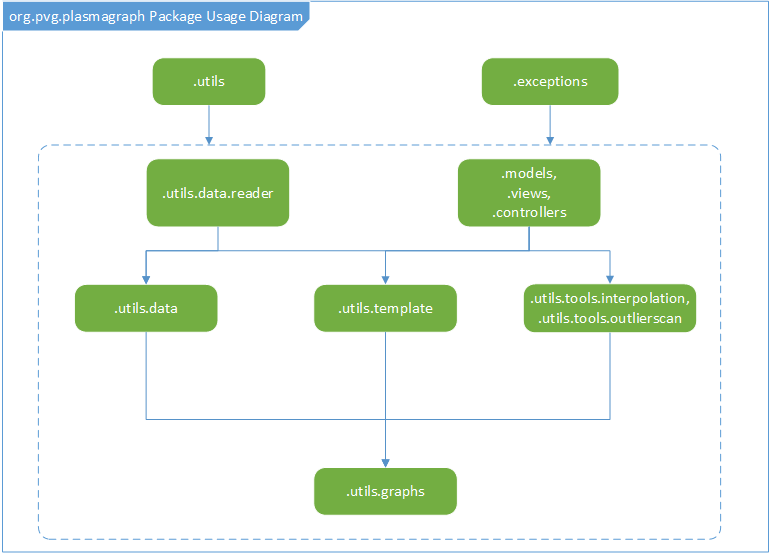
Figure : “Elicit Graph Options” Sequence Diagram



### Composition Diagrams

* 1. Package Usage Flow

Figure : Package Usage Flow Diagram



* 1. Model-View-Controller Connections

Figure : Main MVC Relationship Diagram

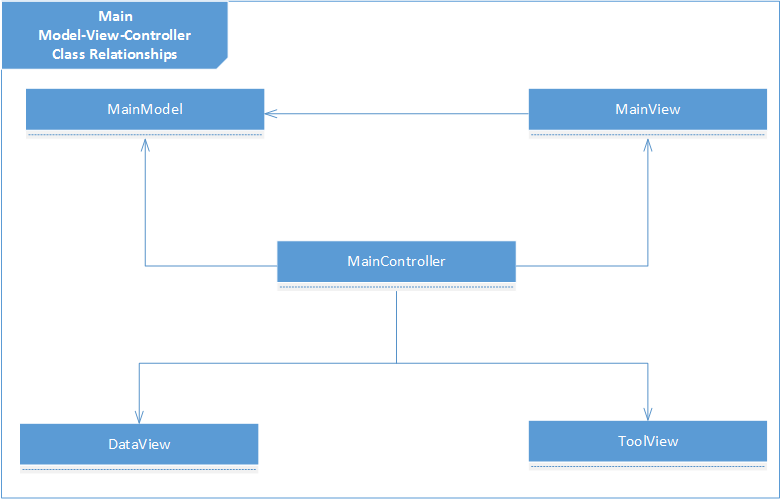


Figure : Data Set MVC Relationship Diagram

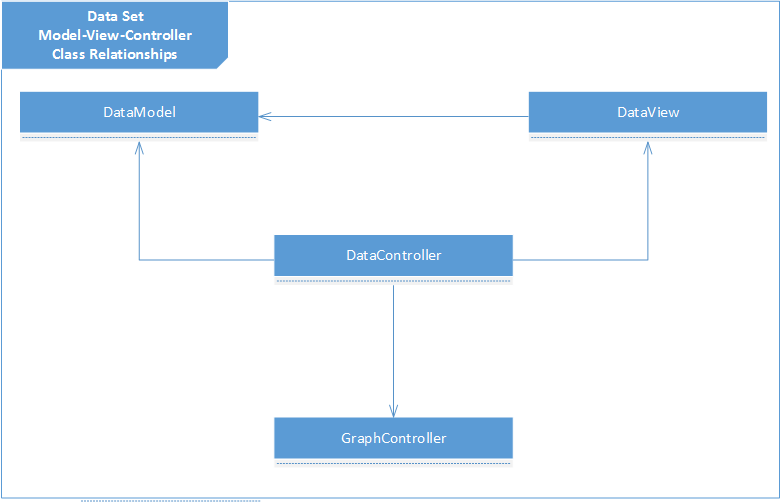


Figure : Tool MVC Relationship Diagram

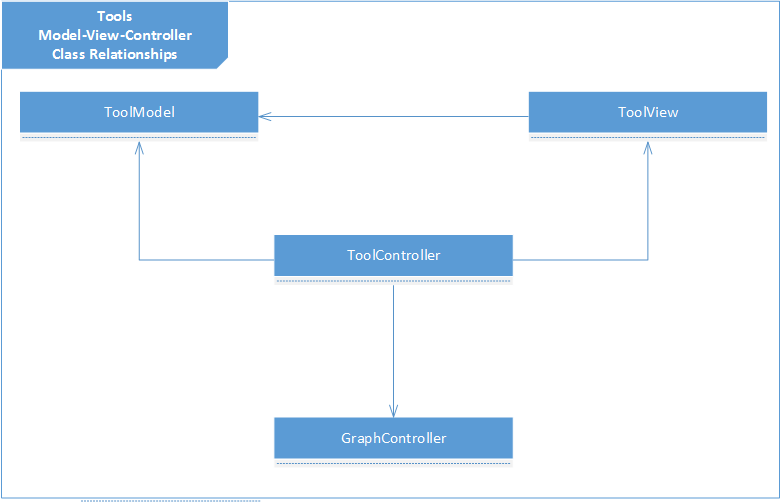
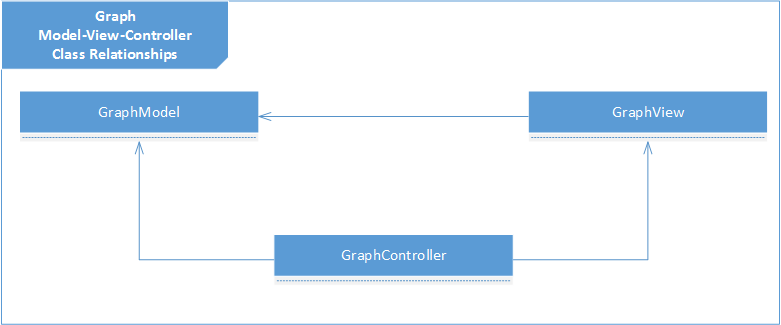


Figure : Graph MVC Relationship Diagram



## 

## 3.3 Design Rationale

PlasmaGraph has undergone many small changes from its initial design. The initial design’s key visual features and their issues are as follows:

* The GUI separated visual column settings and the selection of columns to graph into different windows, forcing the user to switch between both windows to modify parts of the same graph.
* A user could not modify the same graph; instead, a new graph would be produced. This created confusion between the users regarding which was the most recent version of the graph.

These key design features were supported by various internal design decisions. The previous iteration’s structure and the changes made are:

* The GUI was separated into multiple sections based on the different classes of settings available. Instead, all the settings were consolidated into one main window, and all tools (Interpolation and Outlier Scanning) were placed into another. This provided a clear separation between cosmetic and tool settings.
* The graphing procedure would create a new window for the graph each time the “Graph” button was pressed. Instead, a new window was made to contain the graph, and the “Graph” button was made to update the window with a new graph.

These changes allow the program to be focused on editing individual graphs at a time and provide clear instructions on how a user can modify a graph with the options available.

# 

# 4. DATA DESIGN

The data objects that will form a part of PlasmaGraph are:

* HeaderData (Located in: org.pvg.plasmagraph.utils.data)
* GraphPair (Located in: org.pvg.plasmagraph.utils.data)
* DataSet (Located in: org.pvg.plasmagraph.utils.data)
* Template (Located in: org.pvg.plasmagraph.utils.template)

These objects contain the data and settings required by PlasmaGraph in order to produce a graph. The following describes the purpose of each object in relation to furthering the program’s requirements as stated in the SRS.

* HeaderData: This object holds the user-provided data file’s location, as well as a list of the data file’s column names and column data types. This component can be combined with a GraphPair object to create a DataSet. The HeaderData object participates in furthering the functional requirement FR-01.
* GraphPair: This object contains the index references of the X, Y, and Group Columns, as well as their names. The GraphPair is a main component of all graphs, and allows for the easy creation of DataSets via its HeaderData column references. The GraphPair object participates in furthering the functional requirement FR-03 and FR-04.
* DataSet: The DataSet object is the synthesis of the HeaderData and GraphPair objects, containing the X, Y, and (if selected) Group Columns’ data. This object is used in conjunction with the Template to produce graphs. The DataSet object participates in furthering the functional requirement FR-03 and FR-04.
* Template: The Template object is a container of any graph settings not derived from the data file itself. It is one of the two indispensable components (along with the GraphPair) of any PlasmaGraph graph. The Template object participates in furthering the functional requirement FR-03.

Section 3.2 of this document details the underlying composition of these objects, and Section 5 details the processes by which these objects are combined to create graphs.

# 5. COMPONENT DESIGN

The following section will describe all the functions participating in PlasmaGraph’s Use Case Sequence Diagrams (Section 3.2, Sub-section B) via pseudocode.

### Functional Requirements

* 1. FR-01
     1. org.pvg.plasmagraph.models.MainModel – importData () [TBD]
     2. org.pvg.plasmagraph.utils.data.reader.MatlabProcessor – getHeaders ()

boolean getHeaders (HeaderData hd):

if file hasn't been read yet {

store the data file's contents into data;

}

for each column in data {

if the column contains only null, NaN, or empty values

store the column's variable name in an array;

remove the column from the data

}

if the array contains at least one value

print all values in the array;

if the data's size is greater than or equal to 2

empty the contents of hd;

if the size of every column in the data is the same

for each column in data {

create a new HeaderColumn containing column's name and column's data type;

add the new HeaderColumn to hd;

}

set hd.file to file;

return true;

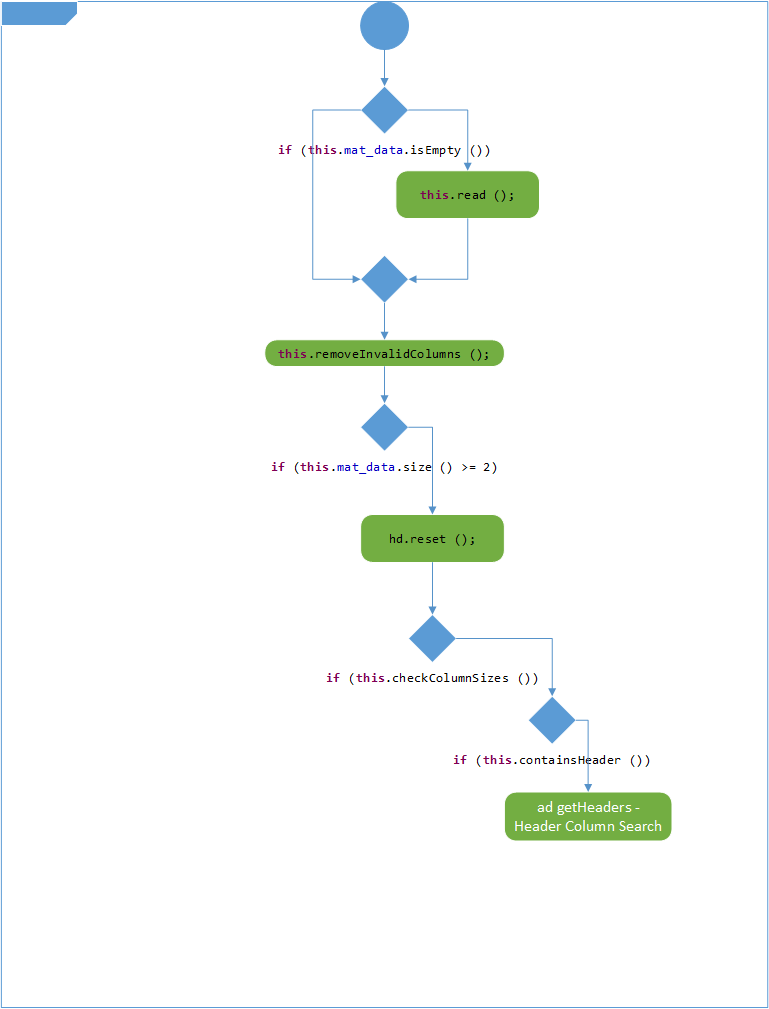
else

throw a new InvalidDataSizeException;

else

throw a new InvalidFileException;

Figure : org.pvg.plasmagraph.utils.data.reader.MatlabProcessor – getHeaders



[TBD – finish diagram, add “Adding Columns to “hd”” diagram]

* 1. FR-02
     1. [TBD]
  2. FR-03
     1. [TBD]
  3. FR-04
     1. [TBD]
  4. FR-05
     1. [TBD]

### Non-Functional Requirements

* 1. NF-01
     1. [TBD]
  2. NF-02
     1. [TBD]
  3. NF-03
     1. [TBD]
  4. NF-04
     1. [TBD]
  5. NF-05
     1. [TBD]
  6. NF-06
     1. [TBD]

# 6. HUMAN INTERFACE DESIGN

## 6.1 Overview of User Interface

PlasmaGraph’s Graphical User Interface is designed to provide the user the tools to modify a single graph at a time. The program (which can be seen in Section 6.2) is divided into two windows:

* The Settings Window: Located on the left side of the screen, this window is responsible for showing the available options to the user. This window also contains a Menu Bar that provides various data-related functions and a “Graph” button that triggers an update in the Graph Window. The Menu Bar contains the following:
  + “Data” Menu Bar Item: This menu provides options related to the data currently in use by PlasmaGraph.
    - “Import Data” Menu Item: Allows the user to select a new data file to use.
    - “View Data” Menu Item: Allows the user to view the data file currently in use.
  + “Templates” Menu Bar Item: This menu provides options related to the settings currently in use by PlasmaGraph.
    - “Import Template” Menu Item: Allows the user to import settings they had saved previously.
    - “Save Template” Menu Item: Allows the user to save settings they have selected for the current graph for later use.
  + “Help” Menu Bar Item: This menu provides options related to the User Guide manual included with PlasmaGraph.
    - “User Guide” Menu Item: Displays a manual that shows the various ways to use PlasmaGraph.

Furthermore, the Settings window is divided into two tabs:

* + “Data View” Tab: This tab contains all the cosmetic settings as well as the X, Y, and Group Column settings. Refer to Section 6.3 for information regarding the individual settings in this tab.
  + “Options View” Tab: This tab contains all the optional tool settings for the Interpolation and Outlier Scanning features. Refer to Section 6.3 for information regarding the individual settings in this tab.
* The Graph Window: Located on the right side of the screen, this window is responsible for showing the graph and allowing the user to save the graph via a right click on the graph itself. This window’s graph updates itself whenever the “Graph” button is pressed on the Settings Window.

The program’s basic use flow is as follows:

1. User opens program.
2. User selects a file to use via the Import Data file-choosing window that automatically appears when the program is started.
3. User selects any number of options in the Data View or Options View tabs.
4. User presses the “Graph” button on the bottom of either tab.

## 6.2 Screen Images

Figure : PlasmaGraph Graphical User Interface Windows

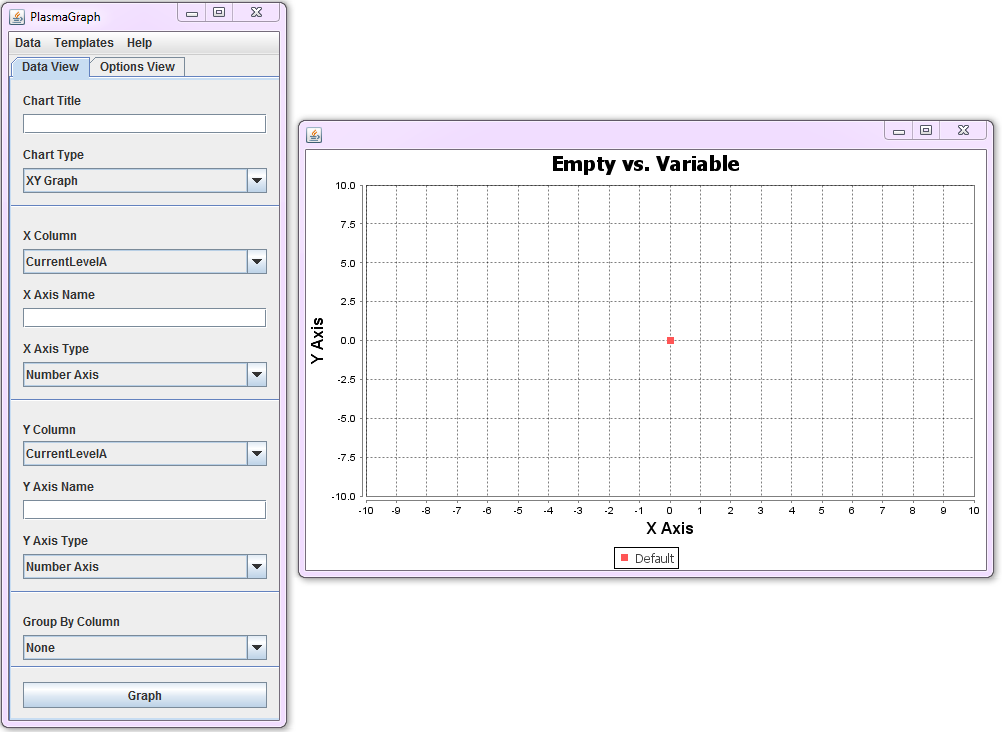


Figure : PlasmaGraph Settings Window Tabs

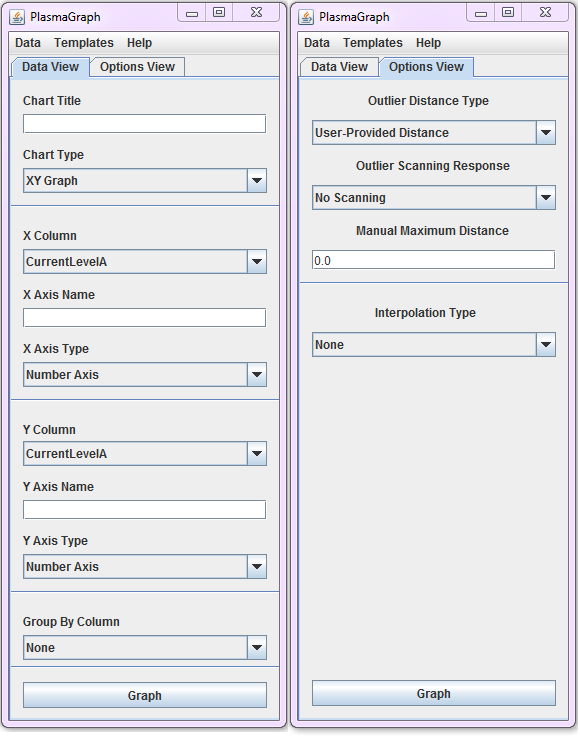


Figure : Open File / Template Window

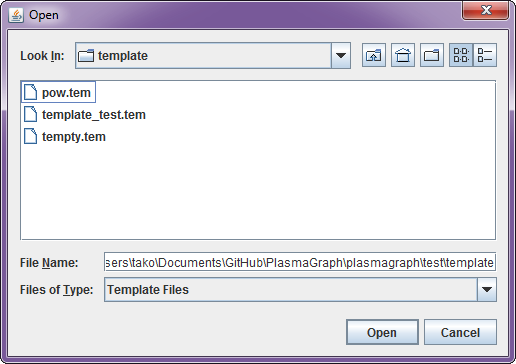


Figure : Save Template Window

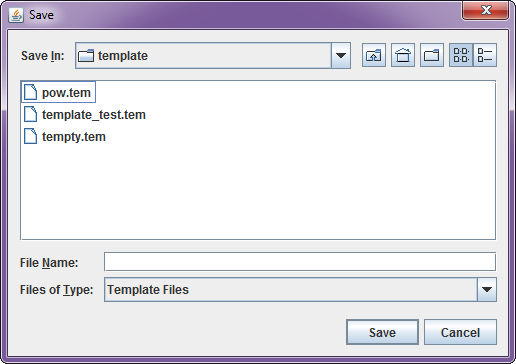


Figure : View Data Window

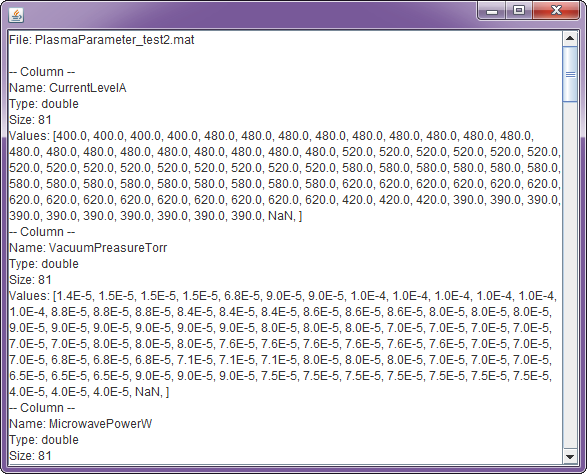


Figure : User Manual Window

[TBD – Image]

## 6.3 Screen Objects and Actions

This section will discuss the various components of PlasmaGraph’s GUI and what functionality they provide to the user.

* Chart Title Text Box: Allows the user to provide a name to the graph. If left empty, PlasmaGraph will automatically populate the graph’s Chart Title with the names of the X Column and Y Column Selection Box selections.
* Chart Type Selection Box: Allows the user to select between various types of graphs. Currently, the only selection available to users is the “XY Graph” option, which provides a XY Scatter Plot, but, in the interest of providing visibility to this setting and showcasing the possibility of future selections, the selection box has been allowed to remain.
* X Column / Y Column Selection Box: Allows the user to select between columns in the imported data file. Currently, the options in these selection boxes are restricted to numerical values because the only chart type available is the XY Graph option. [TBD – Spin it positively? “The Selection Box only displays columns graphable by the selected Chart Type”?]
* X Axis Name / Y Axis Name Text Box: Allows the user to provide a name to the X or Y Axes. If left empty, PlasmaGraph will automatically populate the graph’s X and Y Axes Labels with the names of the X Column and Y Column Selection Box selections, respectively.
* X Axis Type / Y Axis Type Selection Box: Allows the user to select between the available Axis scales for the graph. Currently, the options are:
  + Number Axis: The standard numerical scale.
  + Logarithmic Axis: The standard logarithmic (base-10) scale.
* Group By Column Selection Box: Allows the user to select between columns in the imported data file for a third column to separate the data by. Selecting any option besides “None” groups the X and Y Column values into different groups based on the different types of values in the Group By Column.
* Outlier Distance Type Selection Box: Allows the user to select the type of measuring mechanism to be used when scanning for outliers. Currently, the options for this Selection Box are:
  + User-Provided Distance: Allows the user to provide the Cartesian distance between points where points are statistically invalid.
  + Mahalanobis Distance: Utilizes a modified version of the standard deviation to determine whether points are statistically invalid.
* Outlier Scanning Response Selection Box: Allows the user to select whether Outlier Scanning will be performed. The “Warn” and “Remove” options both provide a window detailing the different values removed from the graph, but “Remove” will have PlasmaGraph automatically remove them, whereas “Warn” will have PlasmaGraph provide the user the option to remove the potential outliers.
* Manual Maximum Distance Text Box: Allows the user to select the maximum Cartesian distance between points for the purposes of Outlier Scanning. Only used when the Outlier Distance Type Selection Box is set to “User-Provided Distance”.
* Interpolation Type Text Box: Allows the user to select the type of interpolation they wish to perform on all the data. The available options are:
  + None: No interpolation will occur.
  + Linear: The data will be interpolated linearly.
  + Quadratic: The data will be interpolated with a quadratic line.
  + Cubic: The data will be interpolated with a cubic line.
  + Spline: The data will be interpolated with a curve passing through each data point. For this option, the data must pass the Vertical Line Theorem (no two points may have the same X axis value).

# 7. REQUIREMENTS MATRIX

Table : Requirements Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Requirements | HelpManual | plasmagraph | controllers | models | tests | | utils | | data | | readers | | exceptions | |
| FR-01 |  |  |  | x |  | | x | | x | | x | | x | |
| FR-02 |  |  |  |  |  | |  | |  | | x | | x | |
| FR-03 |  |  | x | x |  | |  | | x | |  | | x | |
| FR-04 |  |  | x | x |  | |  | | x | |  | | x | |
| FR-05 |  |  |  |  |  | |  | |  | |  | |  | |
| NR-01 |  |  | x | x |  | |  | |  | |  | |  | |
| NR-02 |  |  | x | x |  | |  | |  | |  | |  | |
| NR-03 |  |  | x | x |  | |  | | x | |  | |  | |
| NR-04 | x |  | x | x |  | |  | |  | |  | |  | |
| NR-05 |  |  | x | x |  | |  | |  | |  | |  | |
| NR-06 |  |  | x | x |  | |  | |  | |  | | x | |
| Requirements | graphs | template | tools | interpolation | | outlierscan | | distances | | methods | | types | | views | |
| FR-01 |  |  |  |  | |  | |  | |  | | x | |  | |
| FR-02 |  |  |  |  | |  | |  | |  | | x | |  | |
| FR-03 |  | x |  |  | |  | |  | |  | | x | | x | |
| FR-04 | x |  | x | x | | x | | x | | x | | x | | x | |
| FR-05 | x |  |  |  | |  | |  | |  | |  | |  | |
| NR-01 |  | x |  |  | |  | |  | |  | |  | | x | |
| NR-02 |  | x |  |  | |  | |  | |  | |  | | x | |
| NR-03 |  |  |  |  | |  | |  | |  | |  | | x | |
| NR-04 |  |  |  |  | |  | |  | |  | |  | | x | |
| NR-05 |  |  |  |  | |  | |  | |  | |  | | x | |
| NR-06 |  |  |  |  | |  | |  | |  | |  | | x | |

# 8. APPENDICES

This section is optional