PlasmaGraph

Software Design Description

By: Plasma Visualization Group

Polytechnic University of Puerto Rico

Electrical & Computer Engineering and Computer Science Department

Fall – Winter 2013

CS 4200: Computer Science Senior Project

Professor: Luis A. Ortiz

Version: SDD r2

Members:

Daniel E. Quintini Greco	[CS - # #####]
Gerardo A. Navas Morales	[CS - # #####]

Revision Chart

- Tuesday, January 21, 2014 – r1 – First version of document. No diagrams, just Introduction. Using SPMP as template.
- Thursday, January 23, 2014 – r2 – Second version of document; added diagrams made up to this date. Provided better structure and removed all mentions of SPMP, including Table of Contents.

Preface

Table of Contents

I. Introduction

1. Purpose

   This document will detail the design of the product in terms of programming internals. It will take advantage of various UML 2.5-specified diagrams in order to present the program flow in a graphical manner.

2. Scope

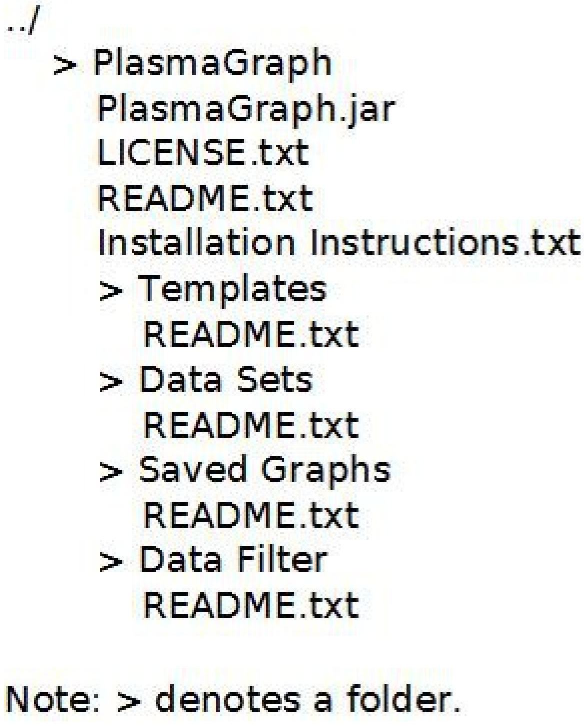
   PlasmaGraph is a data graphing program made in Java, and is designed to take files of certain formats (See the SRS for the specific formats.), turn them into usable data, and graph them based on certain user-provided criteria. To that end, this document describes how the product will be designed; it details the organization of the individual files that contain the programing code, the various classes that exist in the program, and how the functions of each of these classes act and interact to create the functionality requested by the Client. The graphical nature of this document makes it a valuable tool in understanding the processes

3. Notes

   This document is a work in progress; as such, the order of the diagrams is not indicative of its order on the final product.

II. Main System

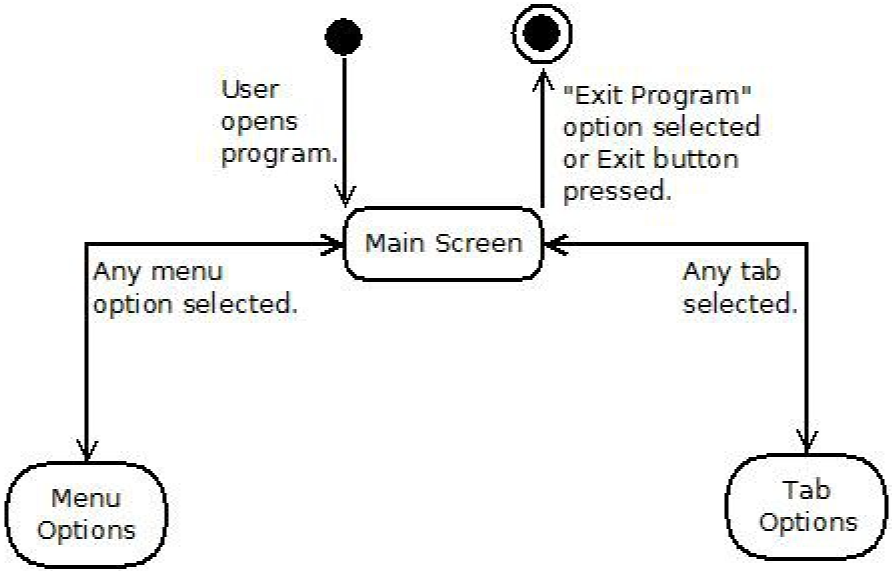
1. Product Directory Organization



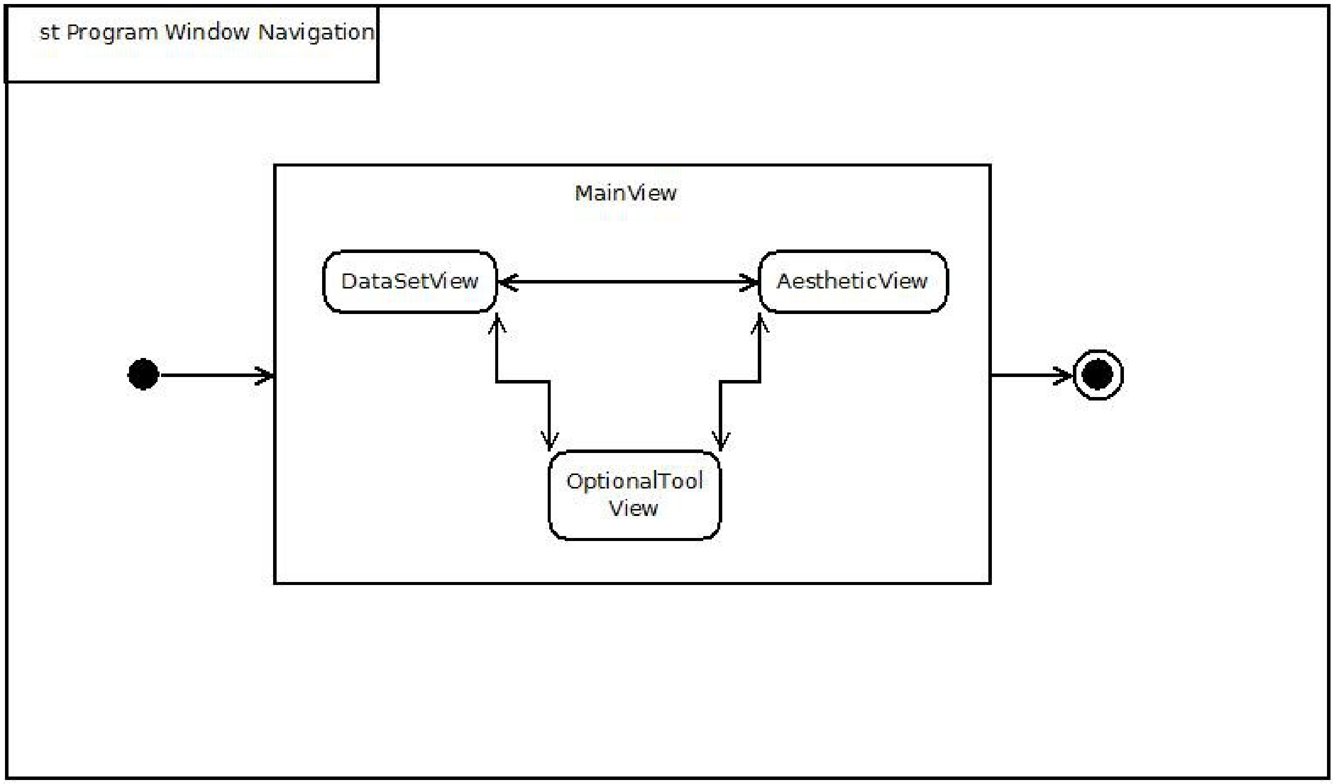
```
../
  > PlasmaGraph
    PlasmaGraph.jar
    LICENSE.txt
    README.txt
    Installation Instructions.txt
    > Templates
      README.txt
    > Data Sets
      README.txt
    > Saved Graphs
      README.txt
    > Data Filter
      README.txt
```

Note: > denotes a folder.

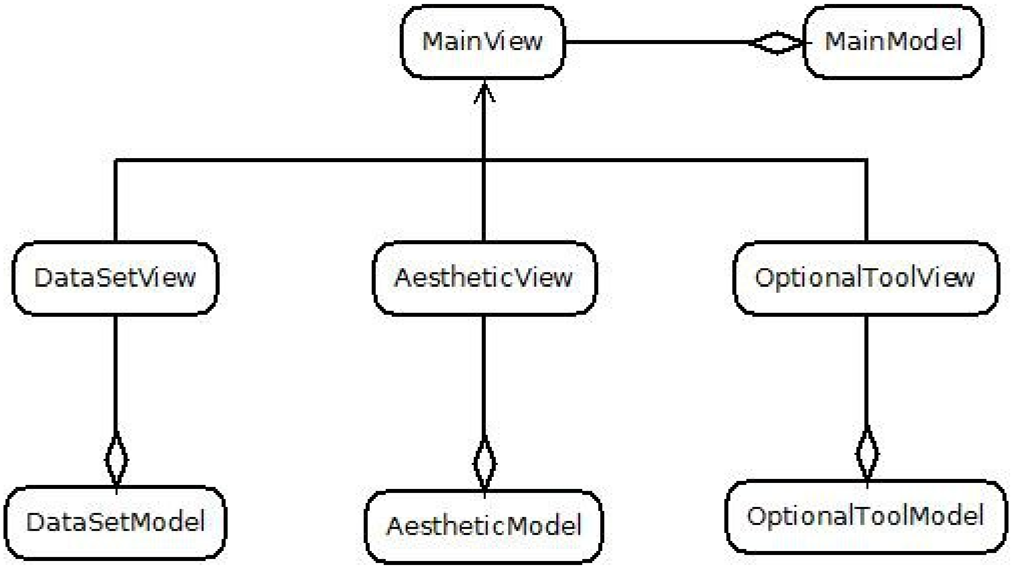
2. Program Flow


User opens program.
"Exit Program" option selected or Exit button pressed.
Main Screen
Any menu option selected.
Any tab selected.
Menu Options
Tab Options

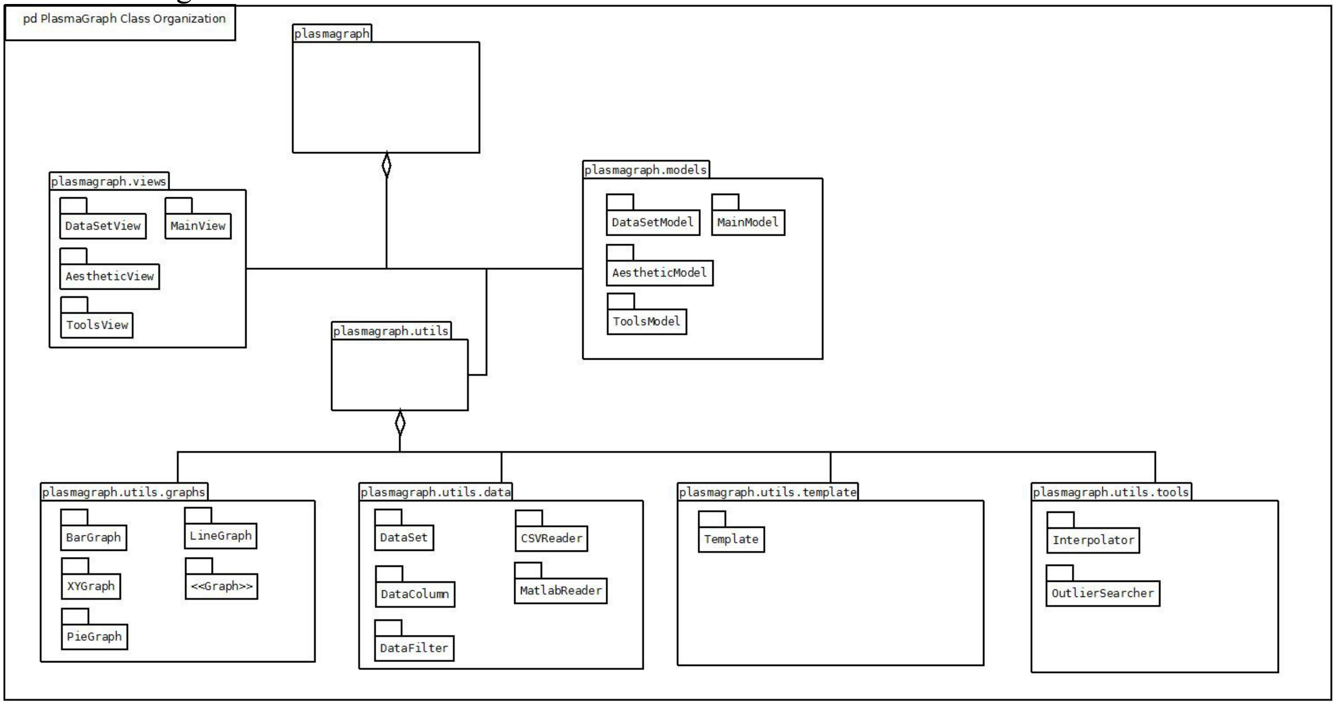
3. Program Window States



4. View – Model Organizational Diagram



5. Packages and Classes



MainView

- menu_bar: JMenuBar
- data_menu: JMenu
- import_data_option: JMenuItem
- template_menu: JMenu
- import_template_option: JMenuItem
- save_template_option: JMenuItem
- data_filter_menu: JMenu
- import_data_filter_option: JMenuItem
- modify_data_filter_option: JMenuItem
- graph_menu: JMenu
- create_graph_option: JMenuItem
- tab_pane: JTabbedPane

+ MainView (): MainView
- initComponents (): void

AestheticView

- chart_title_label: JLabel
- chart_title_text_box: JTextField
- x_axis_label: JLabel
- x_axis_text_box: JTextField
- y_axis_label: JLabel
- y_axis_text_box: JTextField
- label_orientation_separator: JSeparator
- plot_orientation_label: JLabel
- plot_orientation_button_group: ButtonGroup
- horizontal_orientation: JRadioButton
- vertical_orientation: JRadioButton

+ AestheticView (): AestheticView
- initComponents (): void

DataSetView

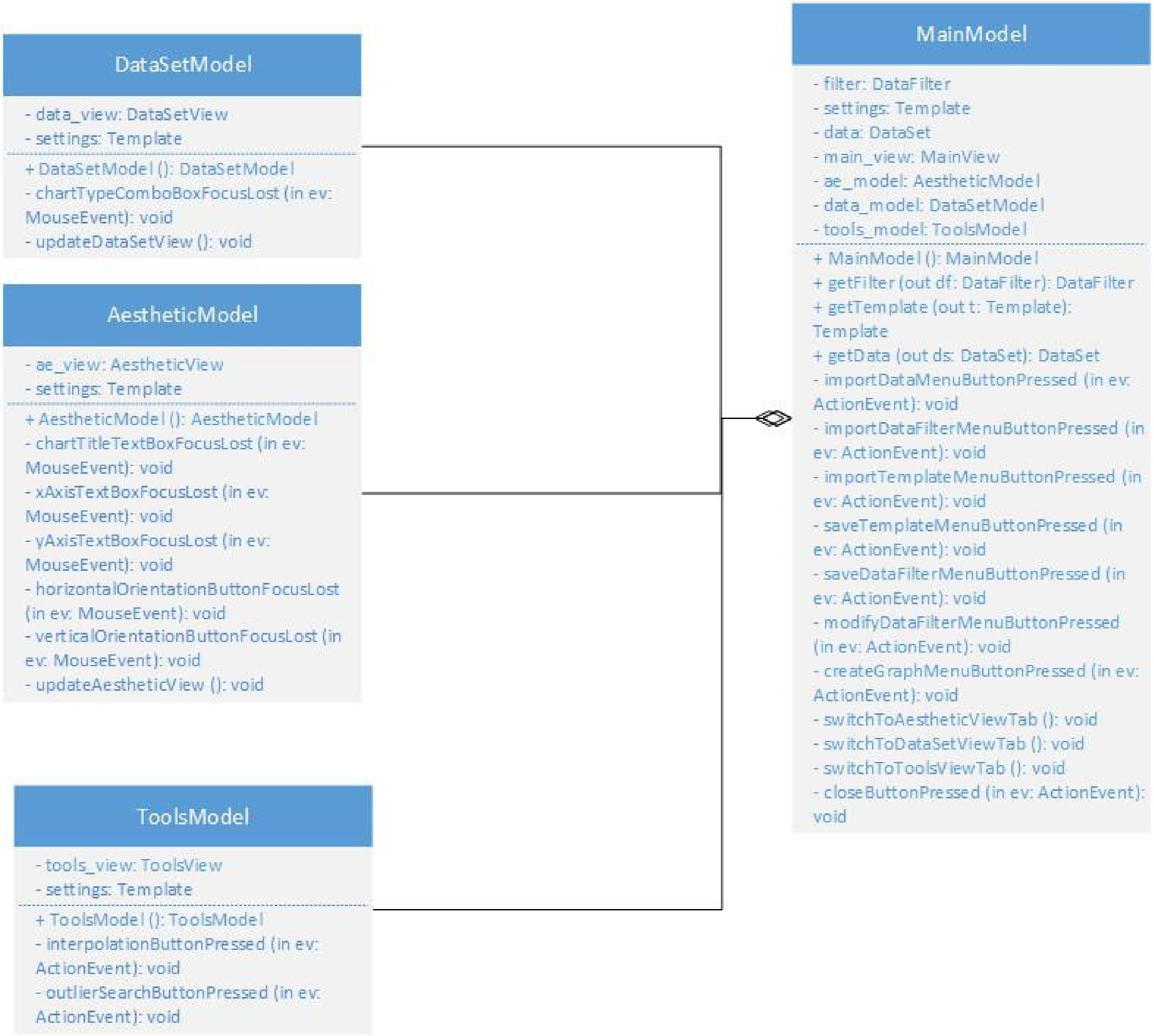
- chart_type_label: JLabel
- chart_type_combo_box: JComboBox
- available_datasets_list: JList
- available_datasets_pane: JScrollPane
- selected_datasets_list: JList
- selected_datasets_pane: JScrollPane
- add_button: JButton
- pair_button: JButton
- remove_button: JButton

+ DataSetView (): DataSetView
- initComponents (): void

ToolsView

- interpolation_label: JLabel
- interpolation_type_combo_box: JComboBox
- interpolation_button: JButton
- interpolation_outlier_separator: JSeparator
- outlier_search_label: JLabel
- outlier_action_combo_box: JComboBox
- outlier_button: JButton

+ ToolsView (): ToolsView
- initComponents (): void



DataSetModel

- data_view: DataSetView
- settings: Template

+ DataSetModel (): DataSetModel
- chartTypeComboBoxFocusLost (in ev: MouseEvent): void
- updateDataSetView (): void

AestheticModel

- ae_view: AestheticView
- settings: Template

+ AestheticModel (): AestheticModel
- chartTitleTextBoxFocusLost (in ev: MouseEvent): void
- xAxisTextBoxFocusLost (in ev: MouseEvent): void
- yAxisTextBoxFocusLost (in ev: MouseEvent): void
- horizontalOrientationButtonFocusLost (in ev: MouseEvent): void
- verticalOrientationButtonFocusLost (in ev: MouseEvent): void
- updateAestheticView (): void

ToolsModel

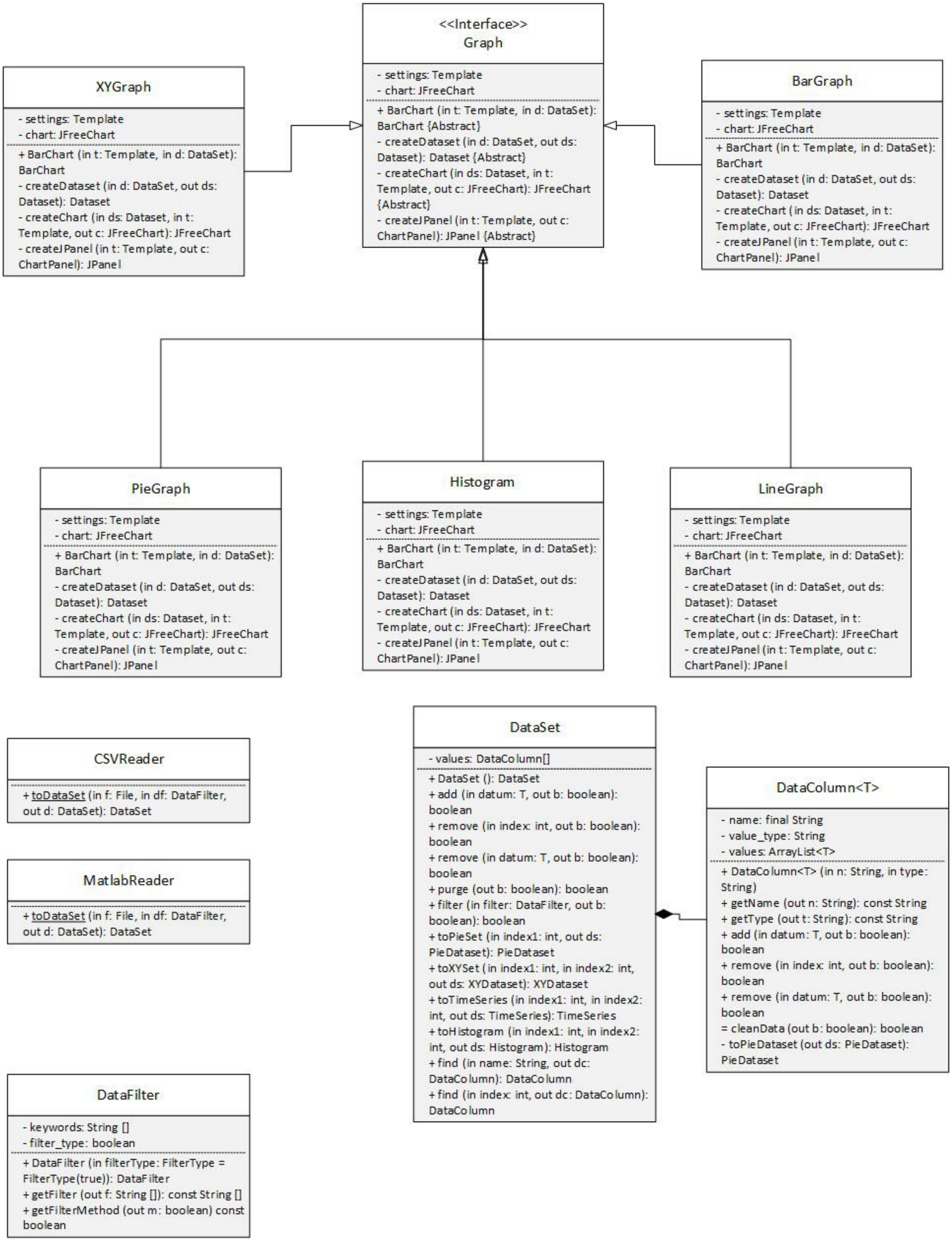
- tools_view: ToolsView
- settings: Template

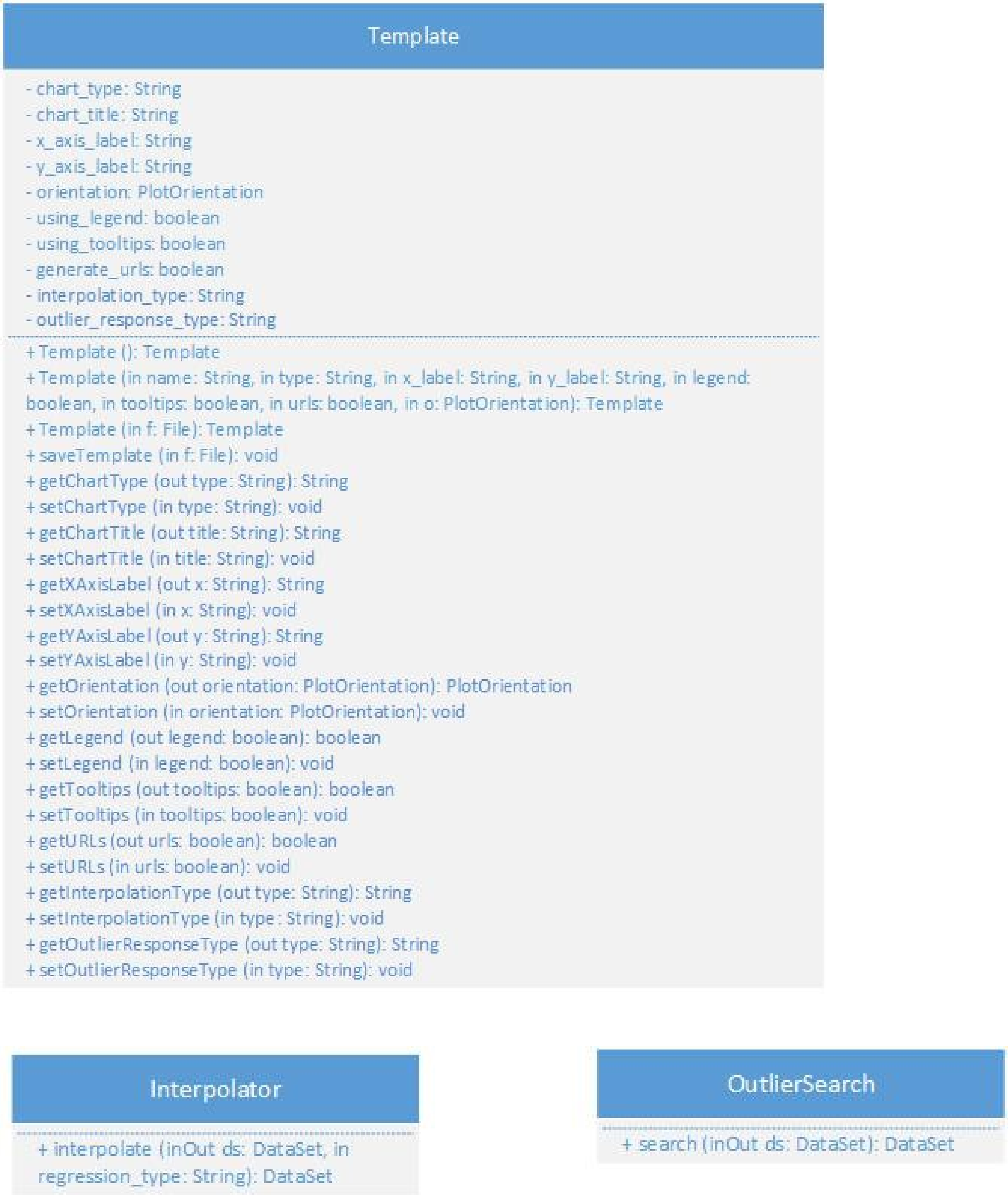
+ ToolsModel (): ToolsModel
- interpolationButtonPressed (in ev: ActionEvent): void
- outlierSearchButtonPressed (in ev: ActionEvent): void

MainModel

- filter: DataFilter
- settings: Template
- data: DataSet
- main_view: MainView
- ae_model: AestheticModel
- data_model: DataSetModel
- tools_model: ToolsModel

+ MainModel (): MainModel
+ getFilter (out df: DataFilter): DataFilter
+ getTemplate (out t: Template): Template
+ getData (out ds: DataSet): DataSet
- importDataMenuButtonPressed (in ev: ActionEvent): void
- importDataFilterMenuButtonPressed (in ev: ActionEvent): void
- importTemplateMenuButtonPressed (in ev: ActionEvent): void
- saveTemplateMenuButtonPressed (in ev: ActionEvent): void
- saveDataFilterMenuButtonPressed (in ev: ActionEvent): void
- modifyDataFilterMenuButtonPressed (in ev: ActionEvent): void
- createGraphMenuButtonPressed (in ev: ActionEvent): void
- switchToAestheticViewTab (): void
- switchToDataSetViewTab (): void
- switchToToolsViewTab (): void
- closeButtonPressed (in ev: ActionEvent): void


<<Interface>>
Graph
- settings: Template
- chart: JFreeChart
+ BarChart (in t: Template, in d: DataSet): BarChart {Abstract}
- createDataset (in d: DataSet, out ds: Dataset): Dataset {Abstract}
- createChart (in ds: Dataset, in t: Template, out c: JFreeChart): JFreeChart {Abstract}
- createJPanel (in t: Template, out c: ChartPanel): JPanel {Abstract}
XYGraph
- settings: Template
- chart: JFreeChart
+ BarChart (in t: Template, in d: DataSet): BarChart
- createDataset (in d: DataSet, out ds: Dataset): Dataset
- createChart (in ds: Dataset, in t: Template, out c: JFreeChart): JFreeChart
- createJPanel (in t: Template, out c: ChartPanel): JPanel
BarGraph
- settings: Template
- chart: JFreeChart
+ BarChart (in t: Template, in d: DataSet): BarChart
- createDataset (in d: DataSet, out ds: Dataset): Dataset
- createChart (in ds: Dataset, in t: Template, out c: JFreeChart): JFreeChart
- createJPanel (in t: Template, out c: ChartPanel): JPanel
PieGraph
- settings: Template
- chart: JFreeChart
+ BarChart (in t: Template, in d: DataSet): BarChart
- createDataset (in d: DataSet, out ds: Dataset): Dataset
- createChart (in ds: Dataset, in t: Template, out c: JFreeChart): JFreeChart
- createJPanel (in t: Template, out c: ChartPanel): JPanel
Histogram
- settings: Template
- chart: JFreeChart
+ BarChart (in t: Template, in d: DataSet): BarChart
- createDataset (in d: DataSet, out ds: Dataset): Dataset
- createChart (in ds: Dataset, in t: Template, out c: JFreeChart): JFreeChart
- createJPanel (in t: Template, out c: ChartPanel): JPanel
LineGraph
- settings: Template
- chart: JFreeChart
+ BarChart (in t: Template, in d: DataSet): BarChart
- createDataset (in d: DataSet, out ds: Dataset): Dataset
- createChart (in ds: Dataset, in t: Template, out c: JFreeChart): JFreeChart
- createJPanel (in t: Template, out c: ChartPanel): JPanel
CSVReader
+ toDataSet (in f: File, in df: DataFilter, out d: DataSet): DataSet
MatlabReader
+ toDataSet (in f: File, in df: DataFilter, out d: DataSet): DataSet
DataFilter
- keywords: String []
- filter_type: boolean
+ DataFilter (in filterType: FilterType = FilterType(true)): DataFilter
+ getFilter (out f: String []): const String []
+ getFilterMethod (out m: boolean) const boolean
DataSet
- values: DataColumn[]
+ DataSet (): DataSet
+ add (in datum: T, out b: boolean): boolean
+ remove (in index: int, out b: boolean): boolean
+ remove (in datum: T, out b: boolean): boolean
+ purge (out b: boolean): boolean
+ filter (in filter: DataFilter, out b: boolean): boolean
+ toPieSet (in index1: int, out ds: PieDataset): PieDataset
+ toXYSet (in index1: int, in index2: int, out ds: XYDataset): XYDataset
+ toTimeSeries (in index1: int, in index2: int, out ds: TimeSeries): TimeSeries
+ toHistogram (in index1: int, in index2: int, out ds: Histogram): Histogram
+ find (in name: String, out dc: DataColumn): DataColumn
+ find (in index: int, out dc: DataColumn): DataColumn
DataColumn<T>
- name: final String
- value_type: String
- values: ArrayList<T>
+ DataColumn<T> (in n: String, in type: String)
+ getName (out n: String): const String
+ getType (out t: String): const String
+ add (in datum: T, out b: boolean): boolean
+ remove (in index: int, out b: boolean): boolean
+ remove (in datum: T, out b: boolean): boolean
= cleanData (out b: boolean): boolean
- toPieDataset (out ds: PieDataset): PieDataset



Template

- chart_type: String
- chart_title: String
- x_axis_label: String
- y_axis_label: String
- orientation: PlotOrientation
- using_legend: boolean
- using_tooltips: boolean
- generate_urls: boolean
- interpolation_type: String
- outlier_response_type: String

+ Template (): Template
+ Template (in name: String, in type: String, in x_label: String, in y_label: String, in legend: boolean, in tooltips: boolean, in urls: boolean, in o: PlotOrientation): Template
+ Template (in f: File): Template
+ saveTemplate (in f: File): void
+ getChartType (out type: String): String
+ setChartType (in type: String): void
+ getChartTitle (out title: String): String
+ setChartTitle (in title: String): void
+ getXAxisLabel (out x: String): String
+ setXAxisLabel (in x: String): void
+ getYAxisLabel (out y: String): String
+ setYAxisLabel (in y: String): void
+ getOrientation (out orientation: PlotOrientation): PlotOrientation
+ setOrientation (in orientation: PlotOrientation): void
+ getLegend (out legend: boolean): boolean
+ setLegend (in legend: boolean): void
+ getTooltips (out tooltips: boolean): boolean
+ setTooltips (in tooltips: boolean): void
+ getURLs (out urls: boolean): boolean
+ setURLs (in urls: boolean): void
+ getInterpolationType (out type: String): String
+ setInterpolationType (in type: String): void
+ getOutlierResponseType (out type: String): String
+ setOutlierResponseType (in type: String): void

Interpolator

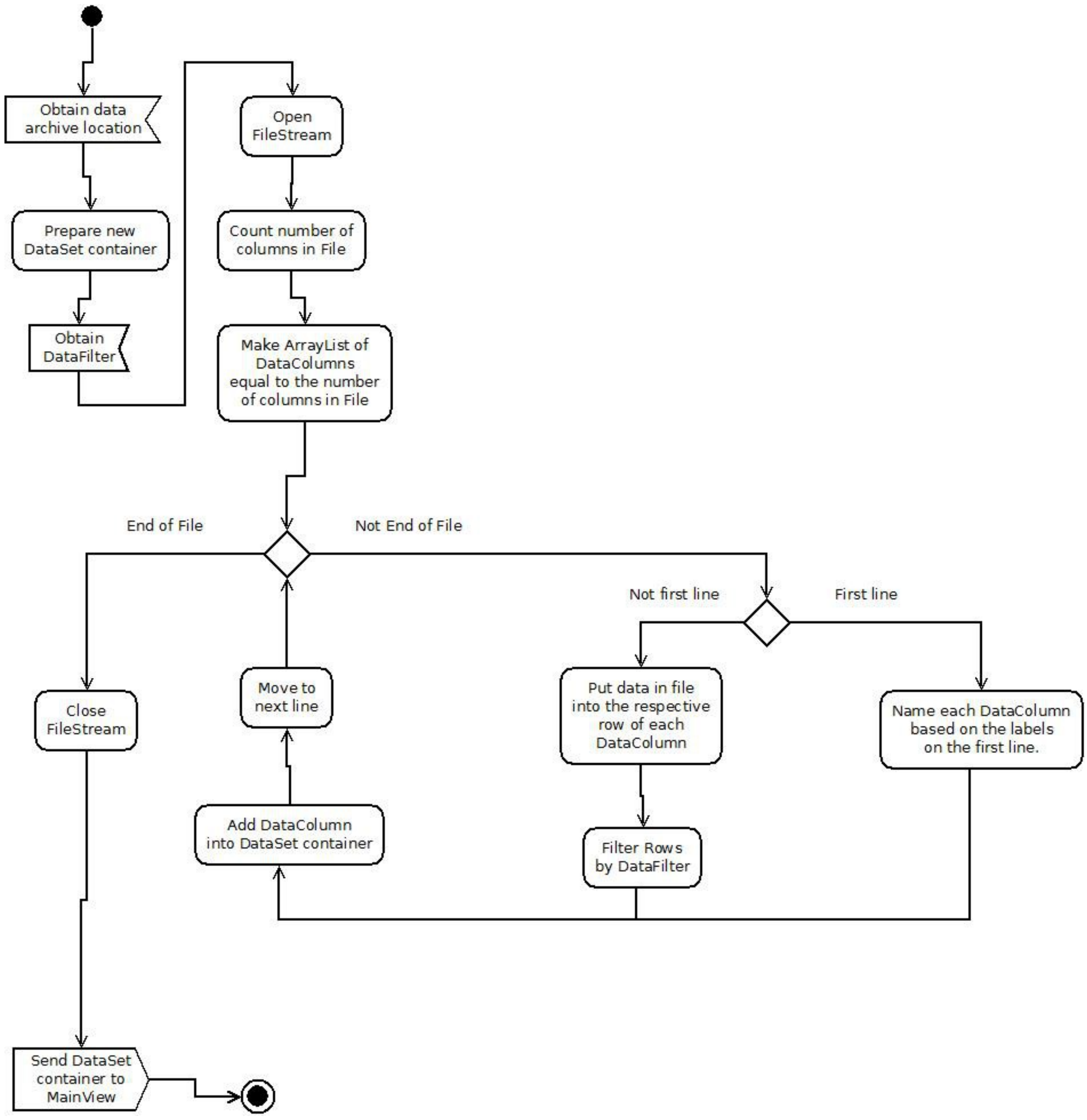
+ interpolate (inOut ds: DataSet, in regression_type: String): DataSet

OutlierSearch

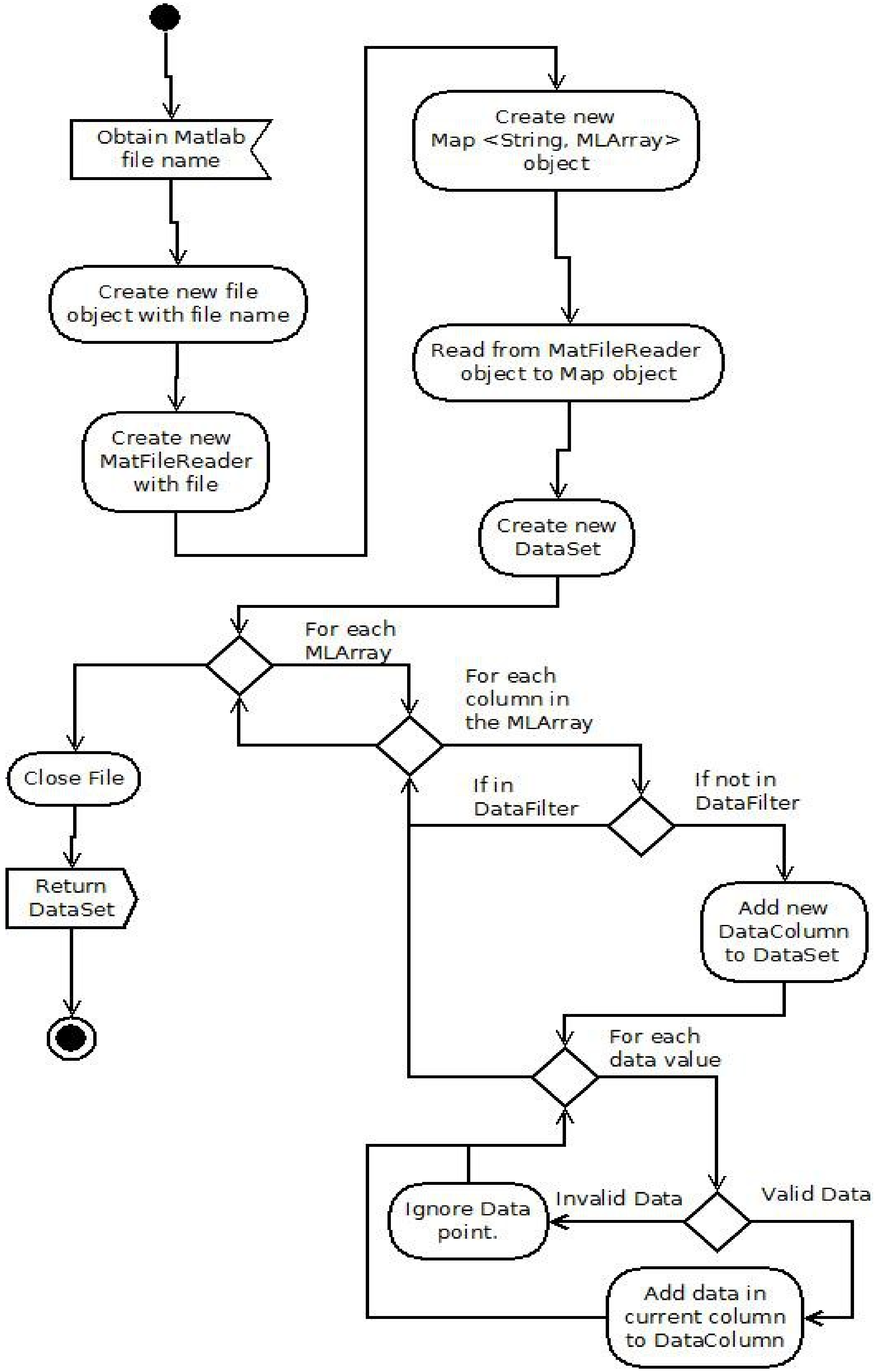
+ search (inOut ds: DataSet): DataSet

III. Data Importing and Processing

1. CSV File Importing


Obtain data archive location
Prepare new DataSet container
Obtain DataFilter
Open FileStream
Count number of columns in File
Make ArrayList of DataColumns equal to the number of columns in File
End of File
Not End of File
Not first line
First line
Close FileStream
Move to next line
Put data in file into the respective row of each DataColumn
Name each DataColumn based on the labels on the first line.
Add DataColumn into DataSet container
Filter Rows by DataFilter
Send DataSet container to MainView

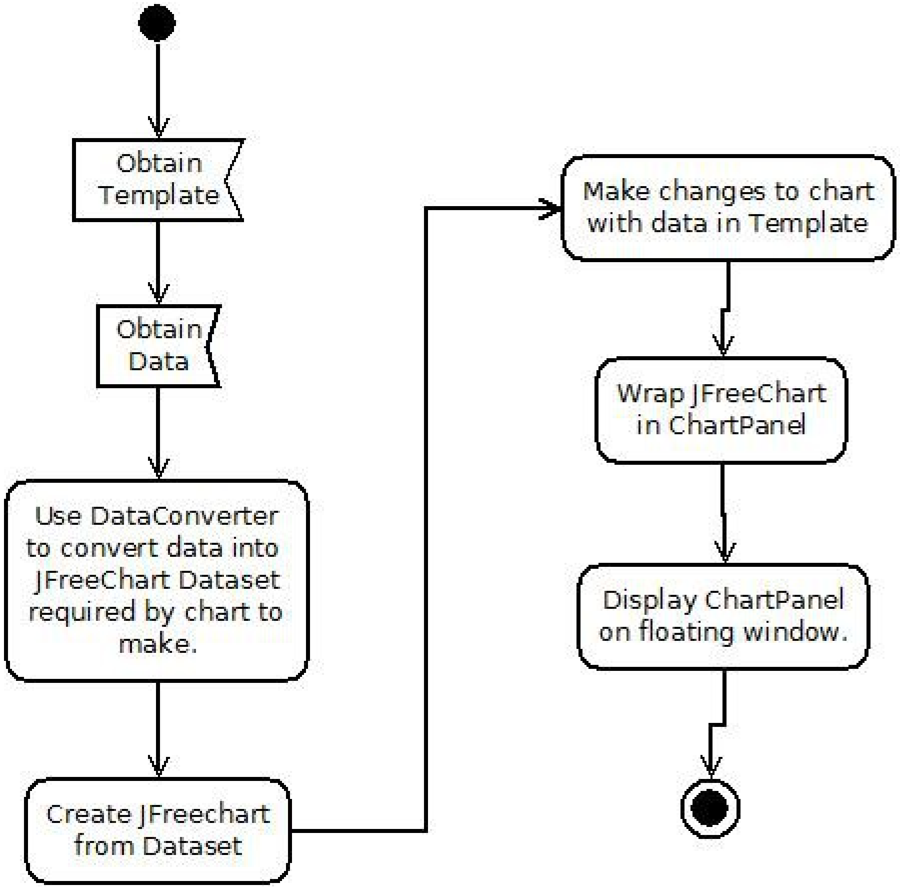
2. Matlab File Importing



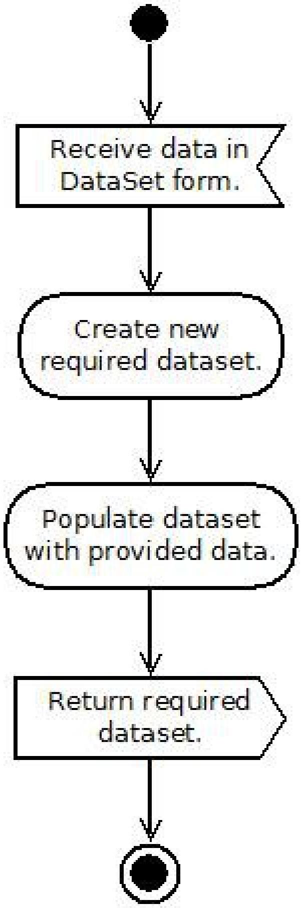
Obtain Matlab file name
Create new file object with file name
Create new MatFileReader with file
Create new Map <String, MLArray> object
Read from MatFileReader object to Map object
Create new DataSet
For each MLArray
For each column in the MLArray
If in DataFilter
If not in DataFilter
Add new DataColumn to DataSet
For each data value
Invalid Data
Valid Data
Ignore Data point.
Add data in current column to DataColumn
Close File
Return DataSet

IV. Graph Creation (WIP; requires more diagrams.)

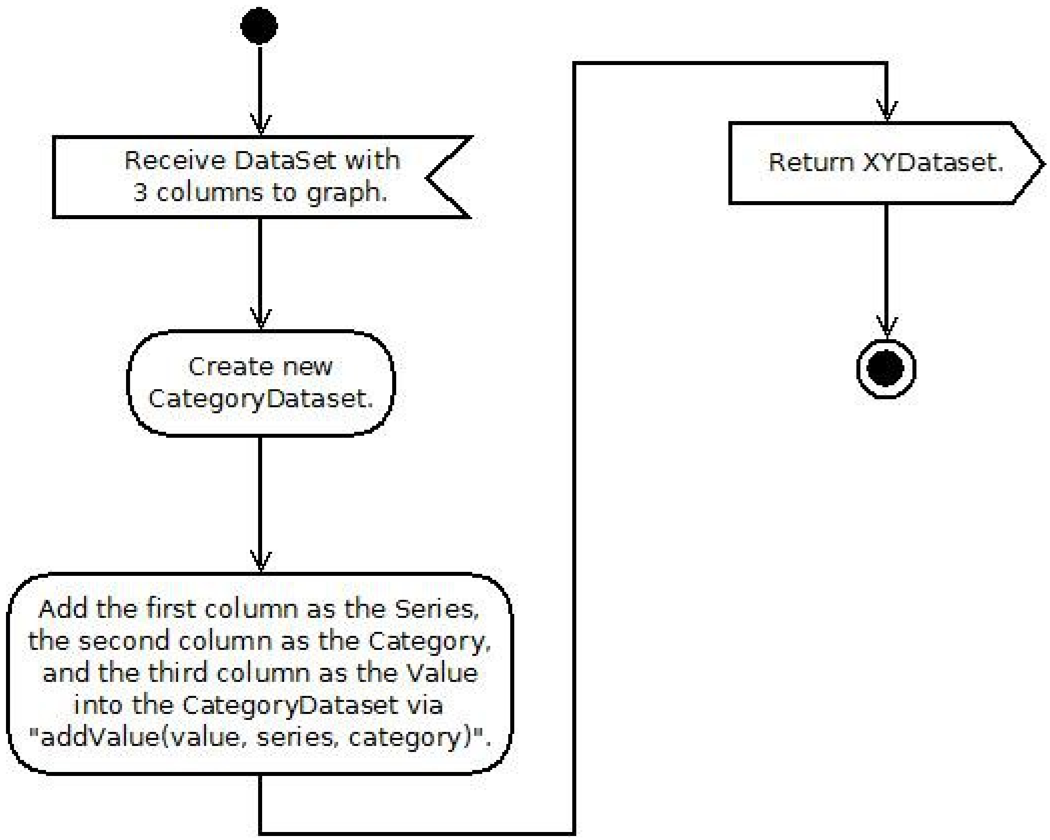
1. Create Graph


Obtain Template
Obtain Data
Use DataConverter to convert data into JFreeChart Dataset required by chart to make.
Create JFreechart from Dataset
Make changes to chart with data in Template
Wrap JFreeChart in ChartPanel
Display ChartPanel on floating window.

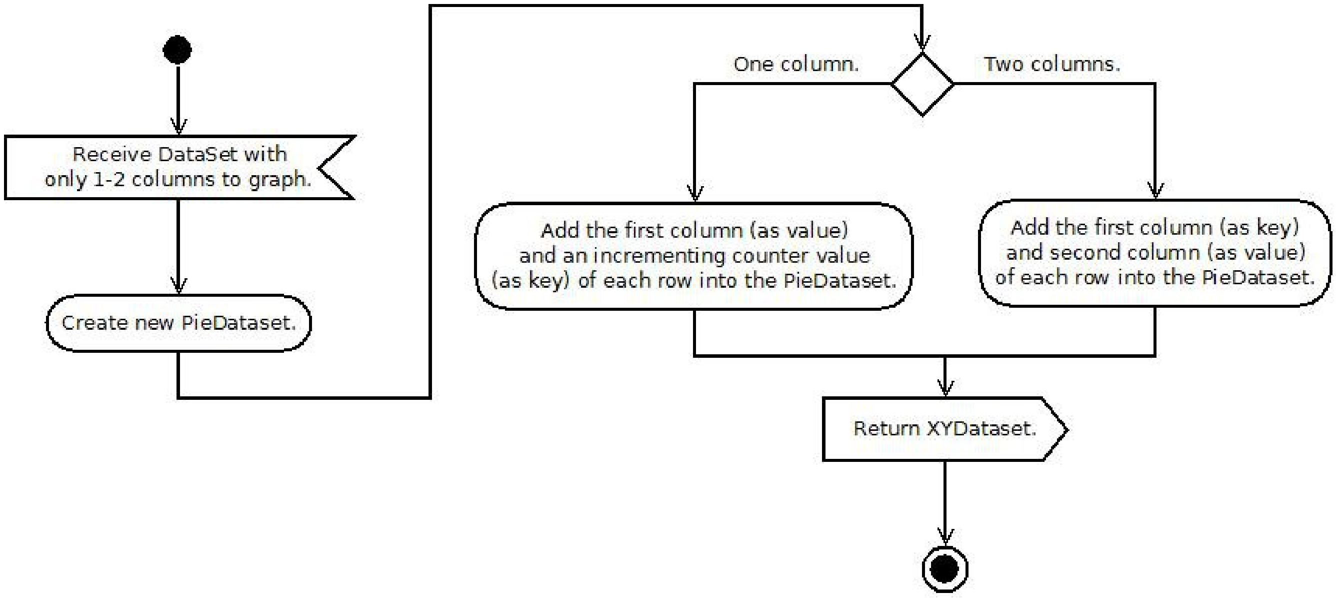
2. Create “Dataset”s (JfreeChart data containers) from “DataSet”s (PlasmaGraph interim data containers)


Receive data in DataSet form.
Create new required dataset.
Populate dataset with provided data.
Return required dataset.

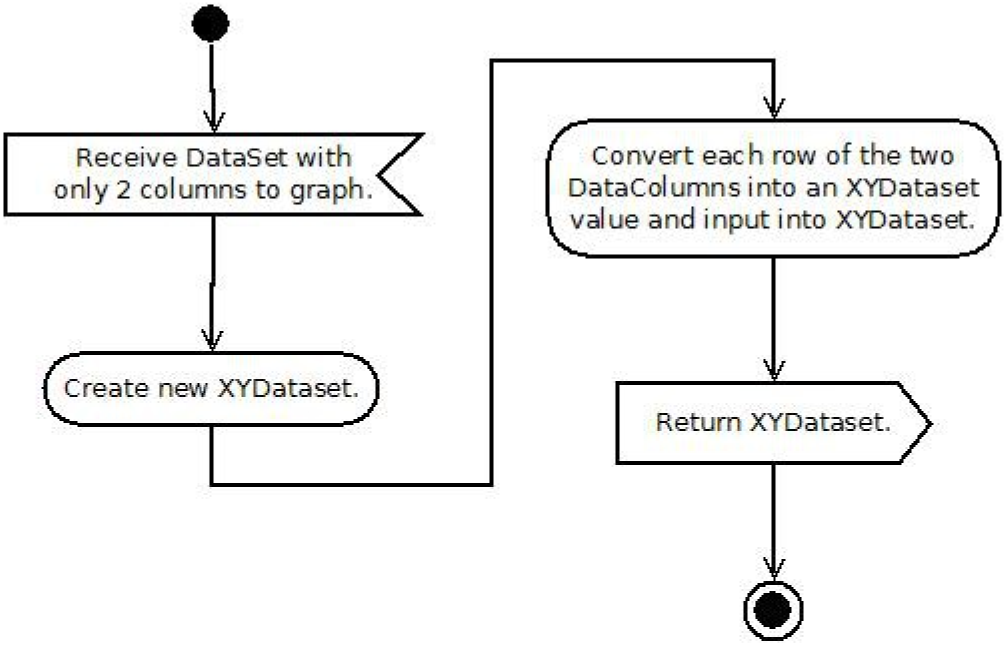
3. DataSet to CategoryDataset


Receive DataSet with 3 columns to graph.
Create new CategoryDataset.
Add the first column as the Series, the second column as the Category, and the third column as the Value into the CategoryDataset via "addValue(value, series, category)".
Return XYDataset.

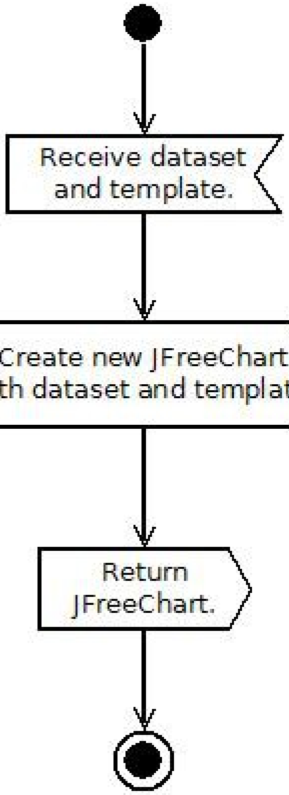
4. DataSet to PieDataset


Receive DataSet with only 1-2 columns to graph.
Create new PieDataset.
One column.
Two columns.
Add the first column (as value) and an incrementing counter value (as key) of each row into the PieDataset.
Add the first column (as key) and second column (as value) of each row into the PieDataset.
Return XYDataset.

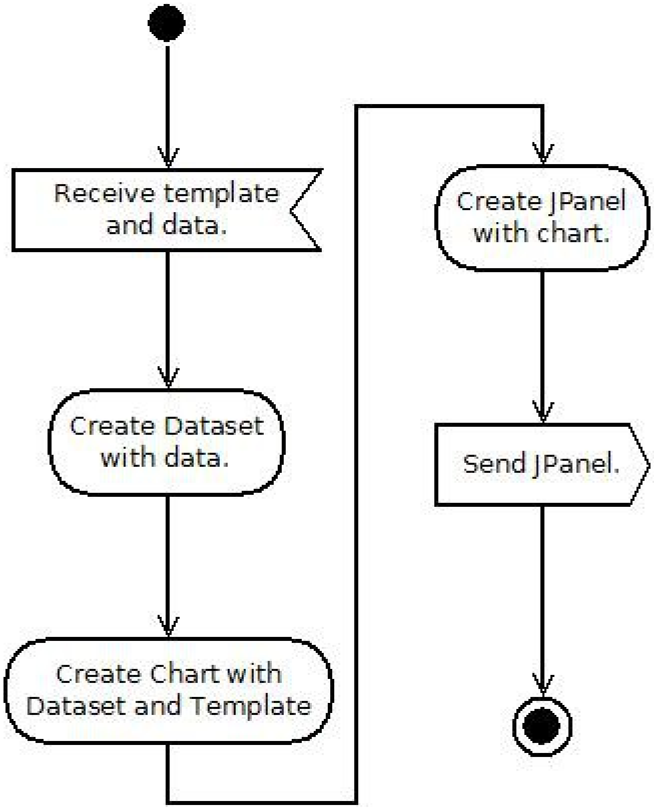
5. DataSet to XYDataset


Receive DataSet with only 2 columns to graph.
Create new XYDataset.
Convert each row of the two DataColumns into an XYDataset value and input into XYDataset.
Return XYDataset.

6. Create Chart

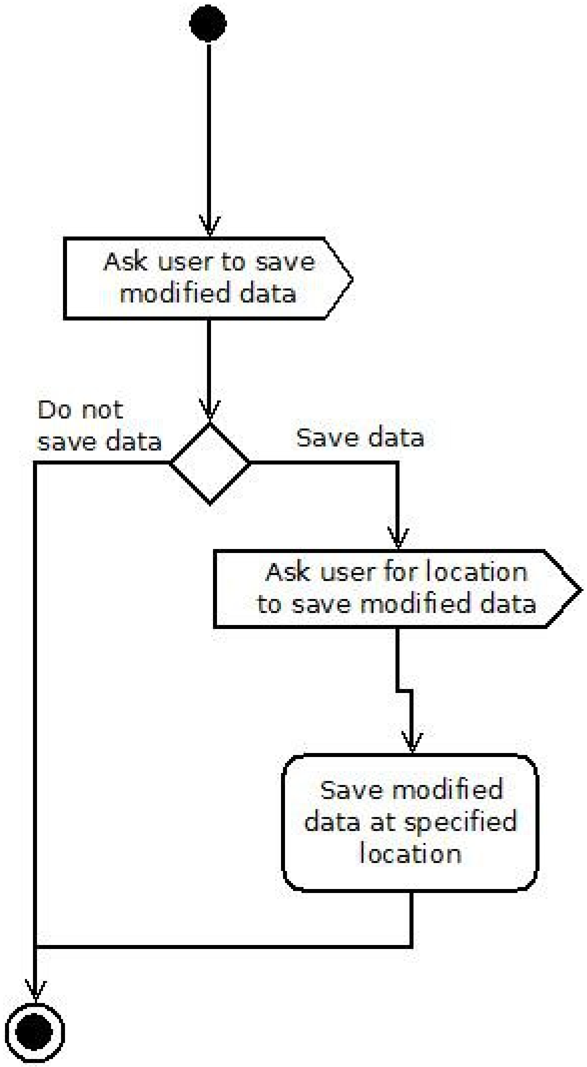

Receive dataset and template.
Create new JFreeChart with dataset and template.
Return JFreeChart.

7. Create JPanel Around Chart


Receive template and data.
Create Dataset with data.
Create Chart with Dataset and Template
Create JPanel with chart.
Send JPanel.

V. Menu Functions

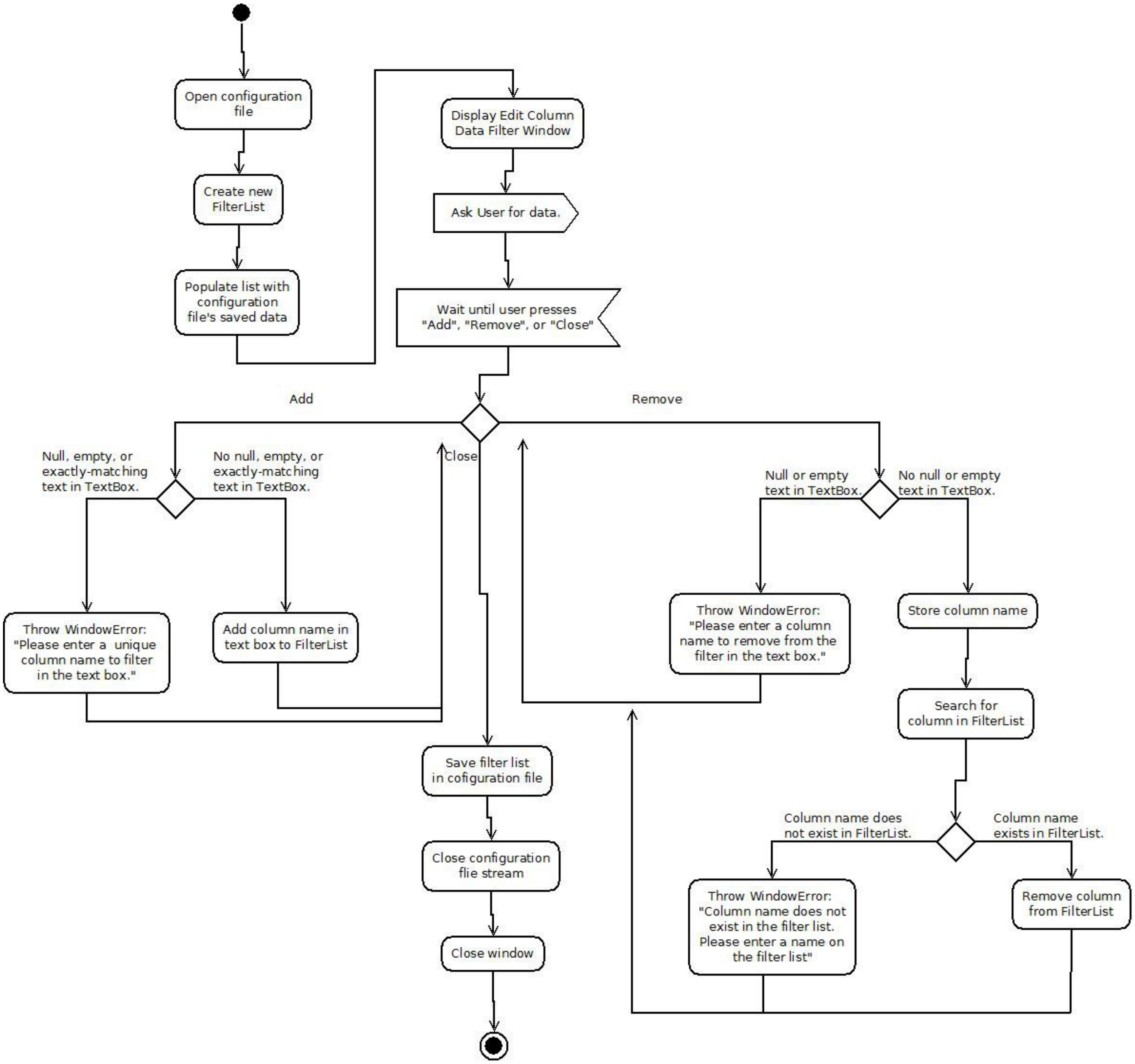
1. Saving Templates and DataFilters


Ask user to save modified data
Do not save data
Save data
Ask user for location to save modified data
Save modified data at specified location

2. Loading Data, Templates, and DataFilters

[N/A]

3. Edit DataFilter


Open configuration file
Create new FilterList
Populate list with configuration file's saved data
Display Edit Column Data Filter Window
Ask User for data.
Wait until user presses "Add", "Remove", or "Close"
Add
Remove
Close
Null, empty, or exactly-matching text in TextBox.
No null, empty, or exactly-matching text in TextBox.
Throw WindowError: "Please enter a unique column name to filter in the text box."
Add column name in text box to FilterList
Null or empty text in TextBox.
No null or empty text in TextBox.
Throw WindowError: "Please enter a column name to remove from the filter in the text box."
Store column name
Search for column in FilterList
Save filter list in cofiguration file
Close configuration flie stream
Close window
Column name does not exist in FilterList.
Column name exists in FilterList.
Throw WindowError: "Column name does not exist in the filter list. Please enter a name on the filter list"
Remove column from FilterList

VI. Support Functions

1. Interpolate Data

<Interpolation_Activity_Diagram.jpeg>

[Cut due to lack of space. Will be separated in two images.]

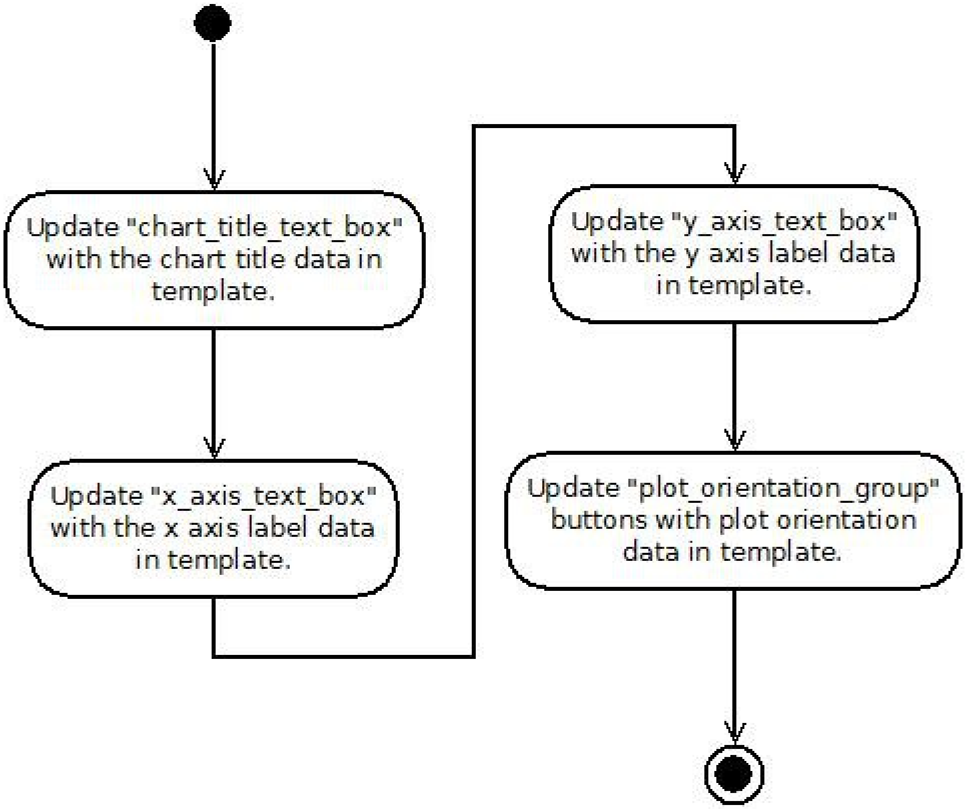
2. Outlier Rejection

<Weak_Outlier_Rejection_Activity_Diagram.jpeg>

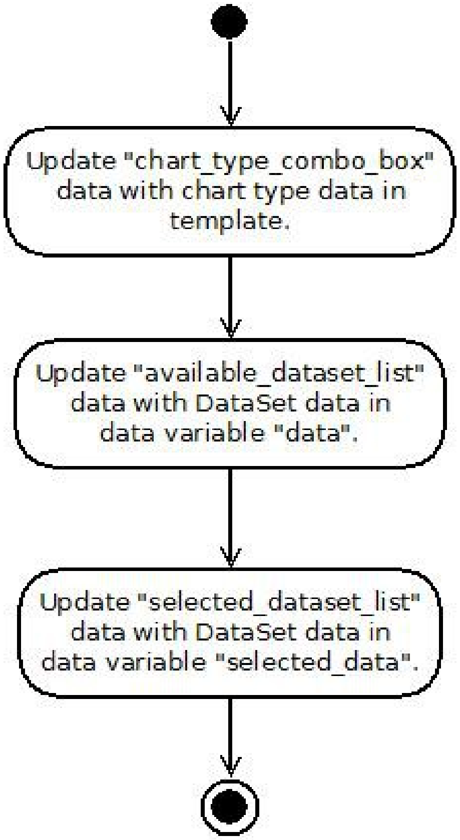
[Cut due to lack of space. Will be separated in two images.]

3. Update Views from Templates

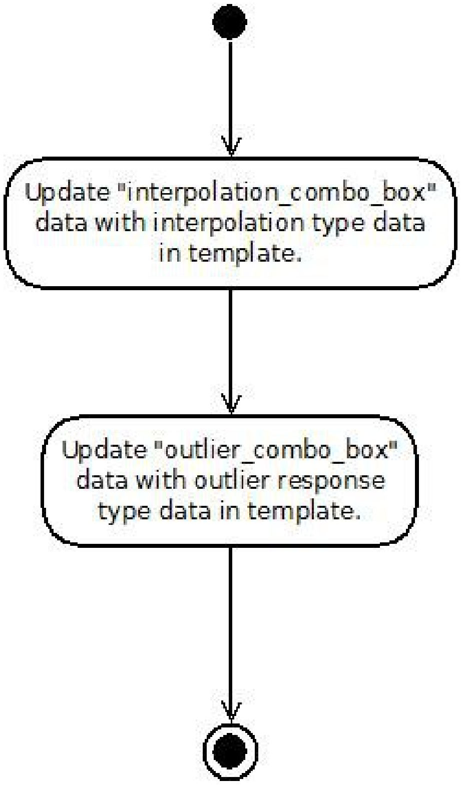
a. Update Aesthetic View from Template


Update "chart_title_text_box" with the chart title data in template.
Update "x_axis_text_box" with the x axis label data in template.
Update "y_axis_text_box" with the y axis label data in template.
Update "plot_orientation_group" buttons with plot orientation data in template.

b. Update Data Set View from Template

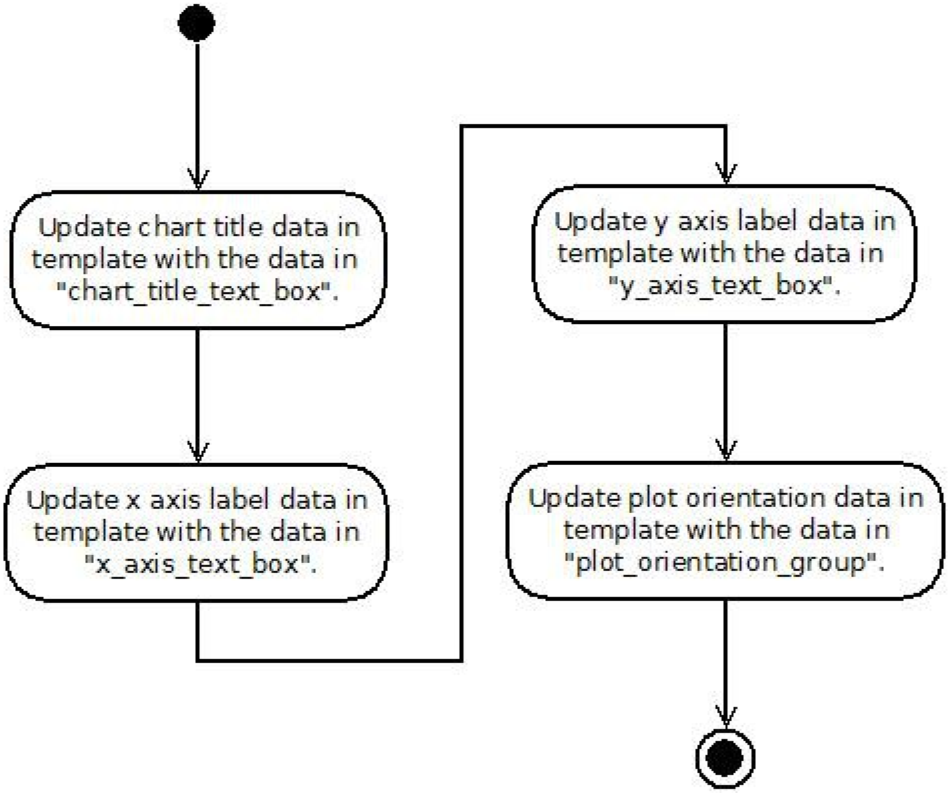

Update "chart_type_combo_box" data with chart type data in template.
Update "available_dataset_list" data with DataSet data in data variable "data".
Update "selected_dataset_list" data with DataSet data in data variable "selected_data".

c. Update Tools View from Template

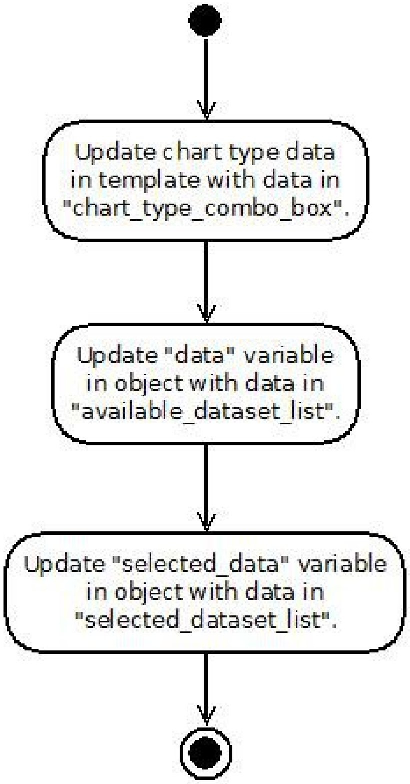

Update "interpolation_combo_box" data with interpolation type data in template.
Update "outlier_combo_box" data with outlier response type data in template.

4. Update Templates from View

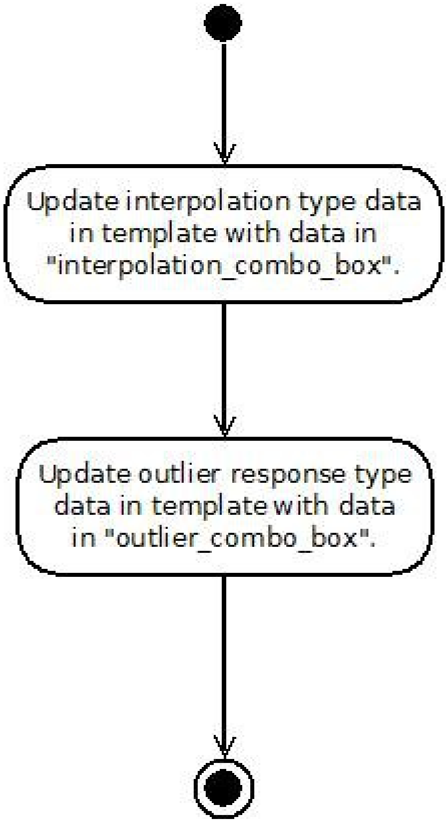
a. Update Template from Aesthetic View


Update chart title data in template with the data in "chart_title_text_box".
Update x axis label data in template with the data in "x_axis_text_box".
Update y axis label data in template with the data in "y_axis_text_box".
Update plot orientation data in template with the data in "plot_orientation_group".

b. Update Template from Data Set View


Update chart type data in template with data in "chart_type_combo_box".
Update "data" variable in object with data in "available_dataset_list".
Update "selected_data" variable in object with data in "selected_dataset_list".

c. Update Template from Tools View


Update interpolation type data in template with data in "interpolation_combo_box".
Update outlier response type data in template with data in "outlier_combo_box".