Criterion B: Design

The Program will rely heavily on object oriented program to allow me to store custom data types as well as create panels that can be easily manipulated to be shown visible or invisible. This will allow the flexibility required to manipulate two different graphs as well as make it easier to find bugs in lines of code pertaining to either graph, then their display, the menus or etc.

Class Method Table:

Data: *Stores data and holds methods to retrieve data*

Name of Method	*Return type*	*Task*	*Description*
Data()	Null	Records values and sets up data	Constructor that grabs values and then stores them in the object
getCPU()	Double	A call method	Returns a double value indicating percent of CPU used at the time of recording
getRAM()	Double	A call method	Returns a double value of percent of RAM used
getHDD()	Double	A call method	Returns percent of HDD used
getTimeStamp()	Long	Returns	Returns the time of the recording

Realtime: *User interface for realtime tab, stores linkedlist and draws line*

Name of Method	*Return type*	*Task*	*Description*
Form(DataManagement)	Null	Sets up and initializes form	Creates the display that the user interacts with
export()	null	Handles export of data to a text document	Exports data
applyChanges()	void	Changes the restrictions of the drawing and redraws the graph	Grabs information from the display and then change restrictions
Draw()	void	Draws a line	Grabs data from the linkedlist and sends it to the data processor, then draws a line
Add(Data)	void	Adds a data point to the LinkedList and	Is fed with data that is added and drawn in

		draws it	
getQueuedBoolean()	void	Internally checks for changes and applies	Checks that Realtime has had a change and redraws graph

DataManagement: *Converts data into points by using pixel by pixel commands*

Name of Method	*Return type*	*Task*	*Description*
DataManagement(Int,int)	void	constructor	Sets up the class for use by setting the graph panel size and using defaults for domain and range
getPoints(double,long)	Int[]	Calculates the points on the graph to be plotted	Uses the size of the display panel and figures the ratio per percent and then calculates a value
setDomain(int,int)	void	Sets the domain values	Changes the values of domain value when calculating points
setRange(int,int)	void	Sets the range values	Changes the values of the Range value when calculating points
setSize(int,int)	Void	Sets the Panel size parameter	Changes Pixel size when calculating pixel/size ratio

LoadGraph: *allows you to import graph data and manages the display of that data*

Method Name	*Return type*	*Task*	*Description*
LoadGraph(DataManagement)	void	Creates and sets up Panel	Creates options that are not part of the inner Panel such as the combobox selector and the load graph button and texbox
Import()	void	Creates a new GraphPanel and draws the line	Deals with the import of data and creation of a new combobox item and drawing graph
applyChanges()	Void	Goes through items and applies changes	Iterates through list and checks Booleans then redraws

GraphPanel: *Stores location data of a graph and options pertaining to that graph*

Method Name	*Return type*	*Task*	*Description*
GraphPanel(LinkedList)	void	Creates graph panel and stores the list	Initializes all panel elements and keeps the data
getColorCPU()	Int[][]	Returns values for the color of the CPU line for this graph	Returns a double array with a size of 3 and 3 respectively for rgb and 0-255 color values
getColorRAM()	Int[][]	Same as above but for RAM	Same as above but for RAM
getColorHDD()	Int[][]	Same as above but for HDD	Same as above but for HDD
getQueuedBoolean()	Boolean	Returns a value if any Boolean are true	A way to speed up the changes to graphs
getVisibilityBoolean()	Boolean	Returns if the line needs to be visible	Returns a Boolean that indicates the visibility of the line
getRemoveBoolean()	Boolean	Returns if the Panel and graph need to be removed	Tells the LoadGraph Panel if it should get rid of the Graph

LoadGraphSettings: *a Panel that houses Domain and Range values as well as view CPU,RAM or HDD checkboxes*

Method Name	*Return type*	*Task*	*Description*
LoadGraphSettings()	void	Constructor that sets up Panel and Panel Dimensions	Sets up the checkboxes and other graphical interfaces
getDomain()	Int[]	Returns Domain values from textboxes	Parses int values from textboxes and sends them
getRange()	Int[]	Same as above but for Range Values	Same as above but for Range Values
getCPULineBoolean()	Boolean	Returns whether Cpu Line(s) should be shown	A Boolean value to indicate the visibility of a CPU line
getRAMLineBoolean()	Boolean	Same as above but for RAM	Same as above but for RAM
getHDDLineBoolean()	Boolean	Same as above but for HDD	Same as above but for HDD
getQueuedBoolean()	Boolean	Returns a Boolean indicating changes were made	A boolean value indicating changes were made in this Panel

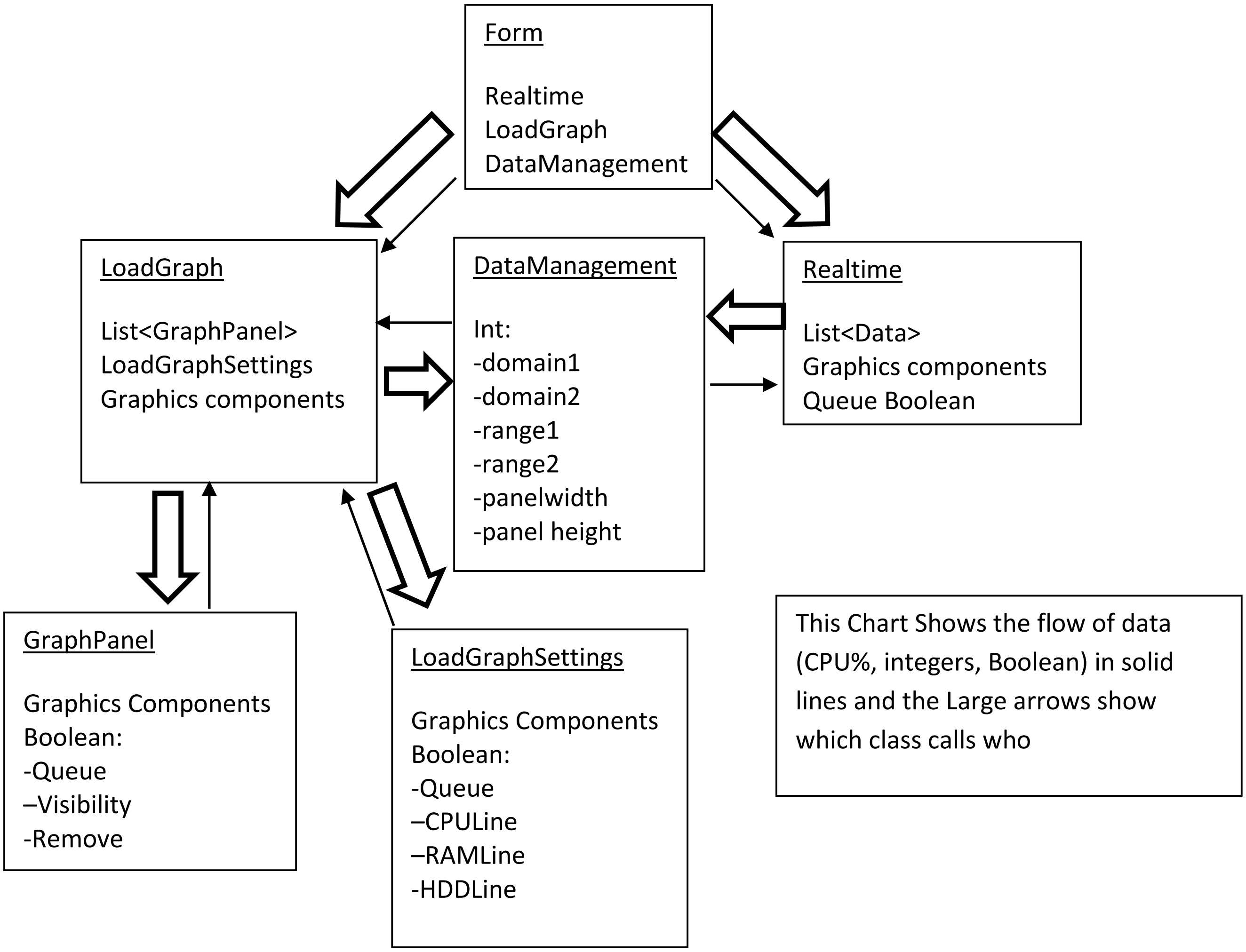
Form: *The Display, changes whether the menu shows Realtime or LoadGraph Panels*

Method Name	*Return type*	*Task*	*Description*
Form()	void	Constructor	Creates Application and infinitely loops after a time period for any changes to the Panels

Test Plan: *Actions and desired results*

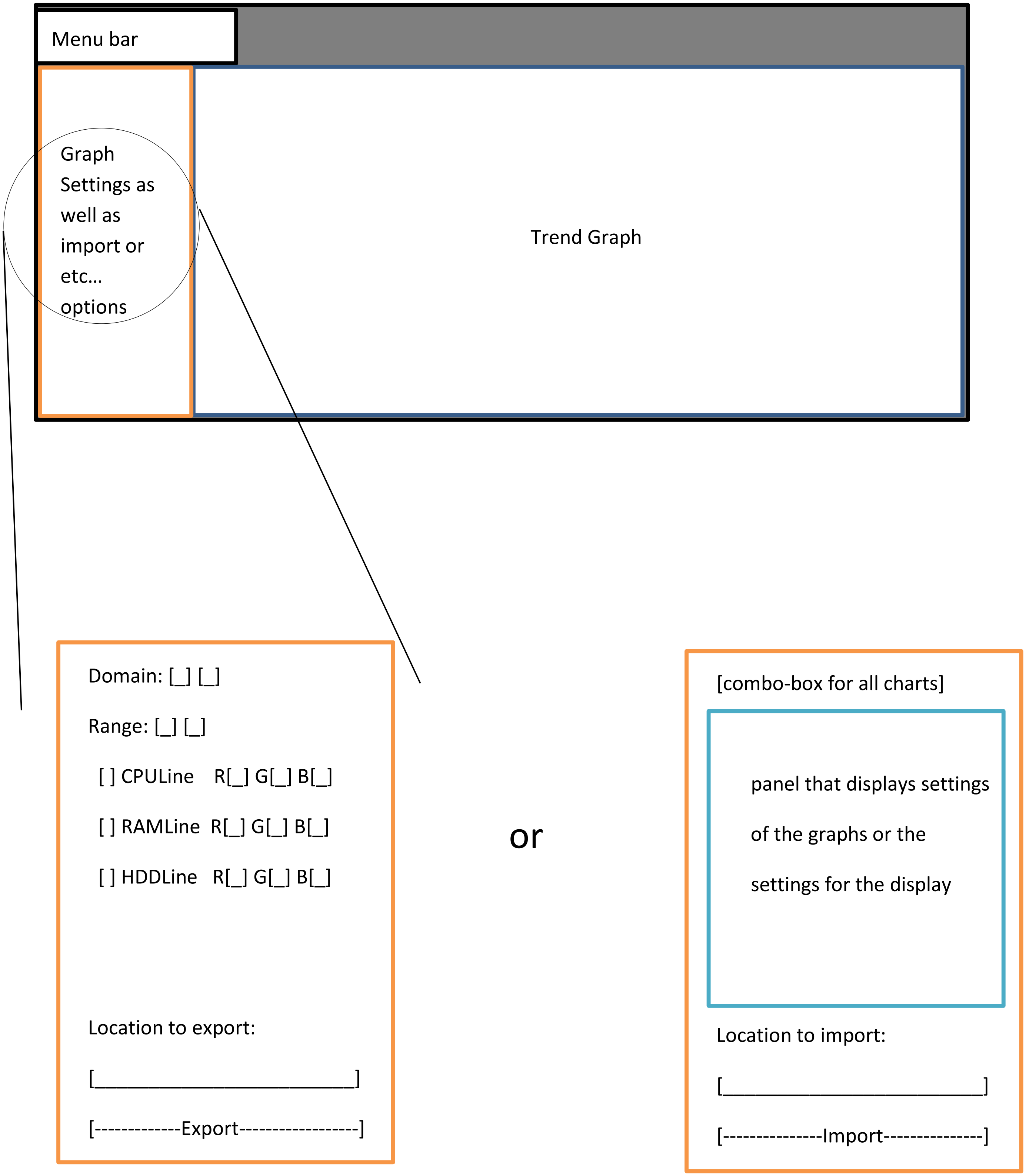
Action	*Method of testing - Result*
Start-up program	Graph begins to load and data is displayed
Domain or range changed	Change settings multiple times, graph should adjust accordingly
Colour value changed	Repeat multiple times, should accept only 0-255
Show Lines	Repeatedly deactivate and activate with different combinations, should disable, enable lines
Export Button	Export multiple times and check that data has been exported
Click load graph tab	Switch back and forth between tabs, multiple times making sure that nothing changes
Use combo box to change panels	Make sure that the user is able to easily and accurately choose the graph to modify
Import Button	Make sure the program can add and handle multiple graphs

Class Relationship Diagram:



This Chart Shows the flow of data (CPU%, integers, Boolean) in solid lines and the Large arrows show which class calls who

Diagram of basic Guidance Components



Menu bar

Graph Settings as well as import or etc… options

Trend Graph

Domain: [_] [_]

Range: [_] [_]

[ ] CPULine R[_] G[_] B[_]

[ ] RAMLine R[_] G[_] B[_]

[ ] HDDLine R[_] G[_] B[_]

Location to export:

[___________________________]

[-------------Export------------------]

or

[combo-box for all charts]

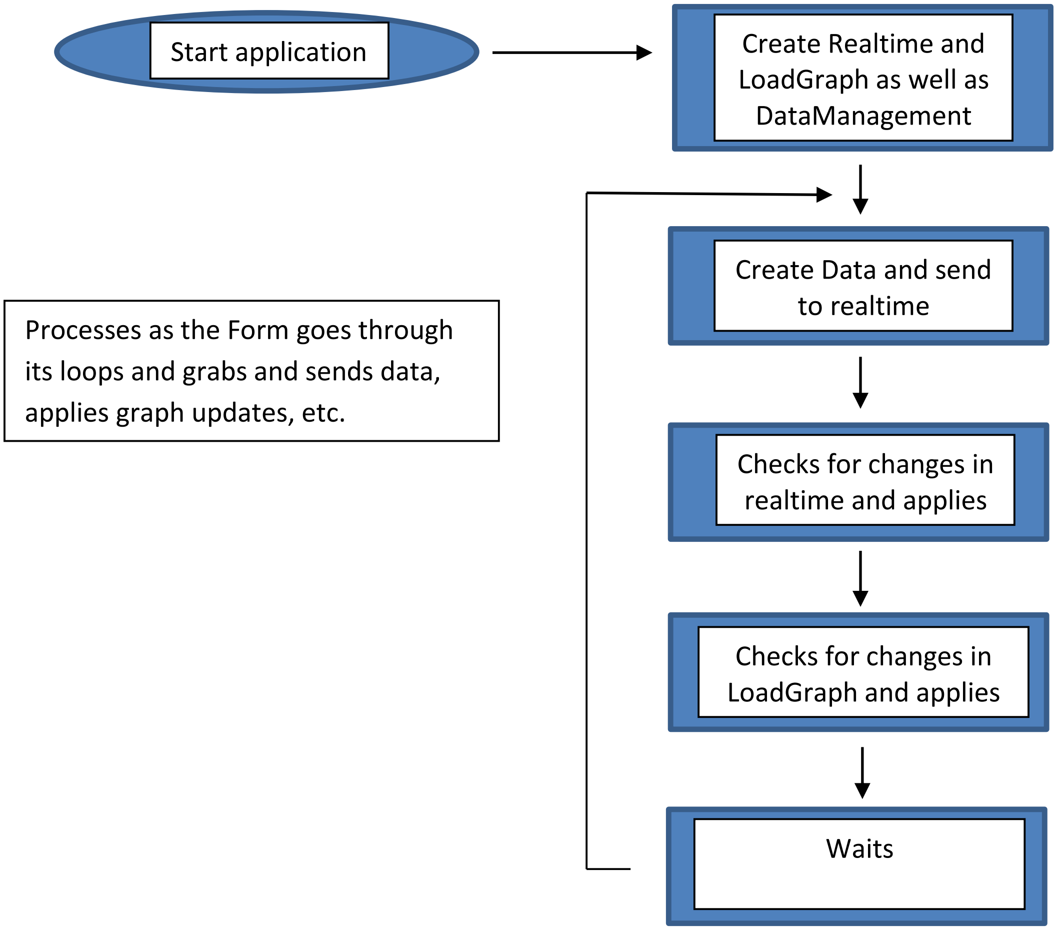
panel that displays settings of the graphs or the settings for the display

Location to import:

[___________________________]

[---------------Import---------------]

Program Flow Diagrams


Start application
Create Realtime and LoadGraph as well as DataManagement
Create Data and send to realtime
Checks for changes in realtime and applies
Checks for changes in LoadGraph and applies
Waits

Processes as the Form goes through its loops and grabs and sends data, applies graph updates, etc.

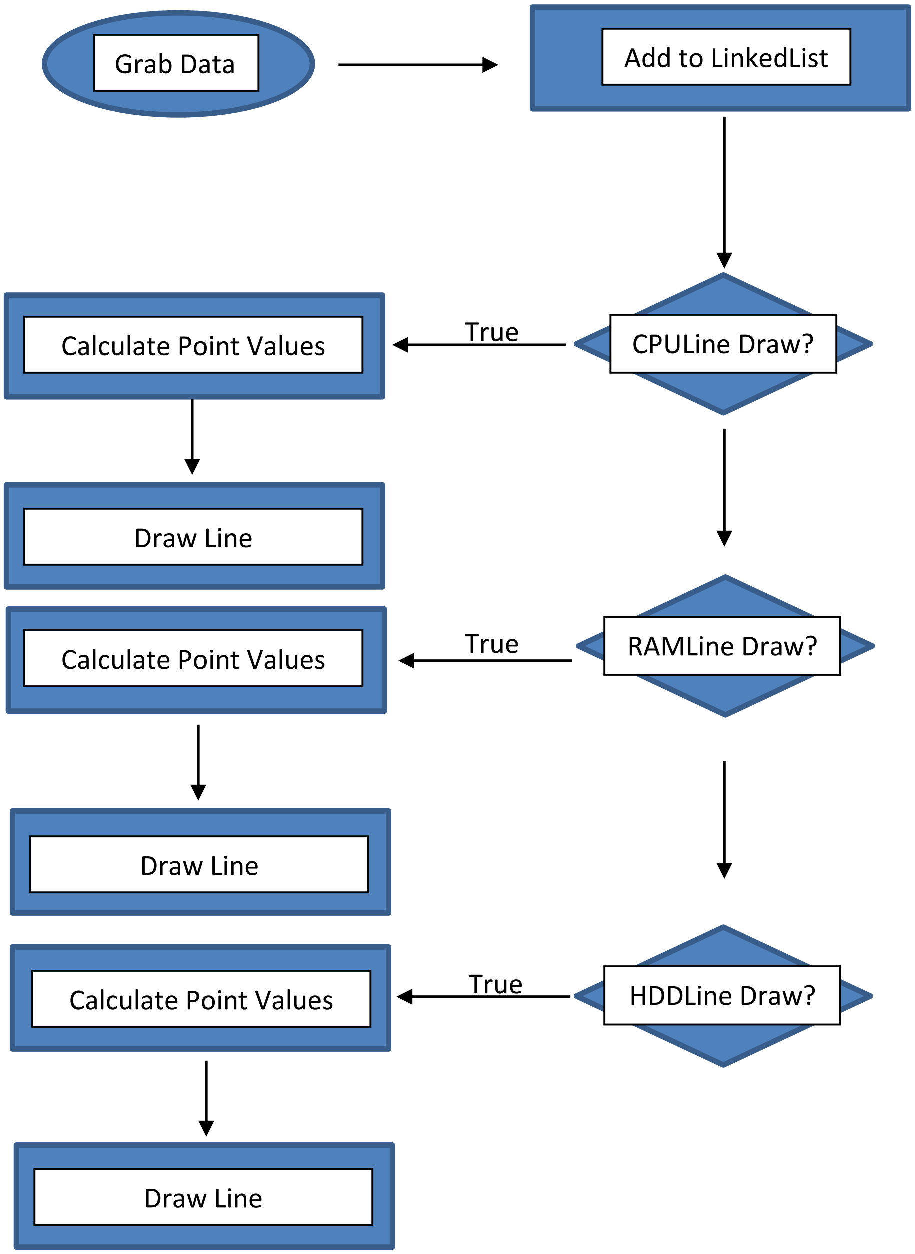


Diagram of Data draw process in which the program checks a Boolean value from the Graphics in order to

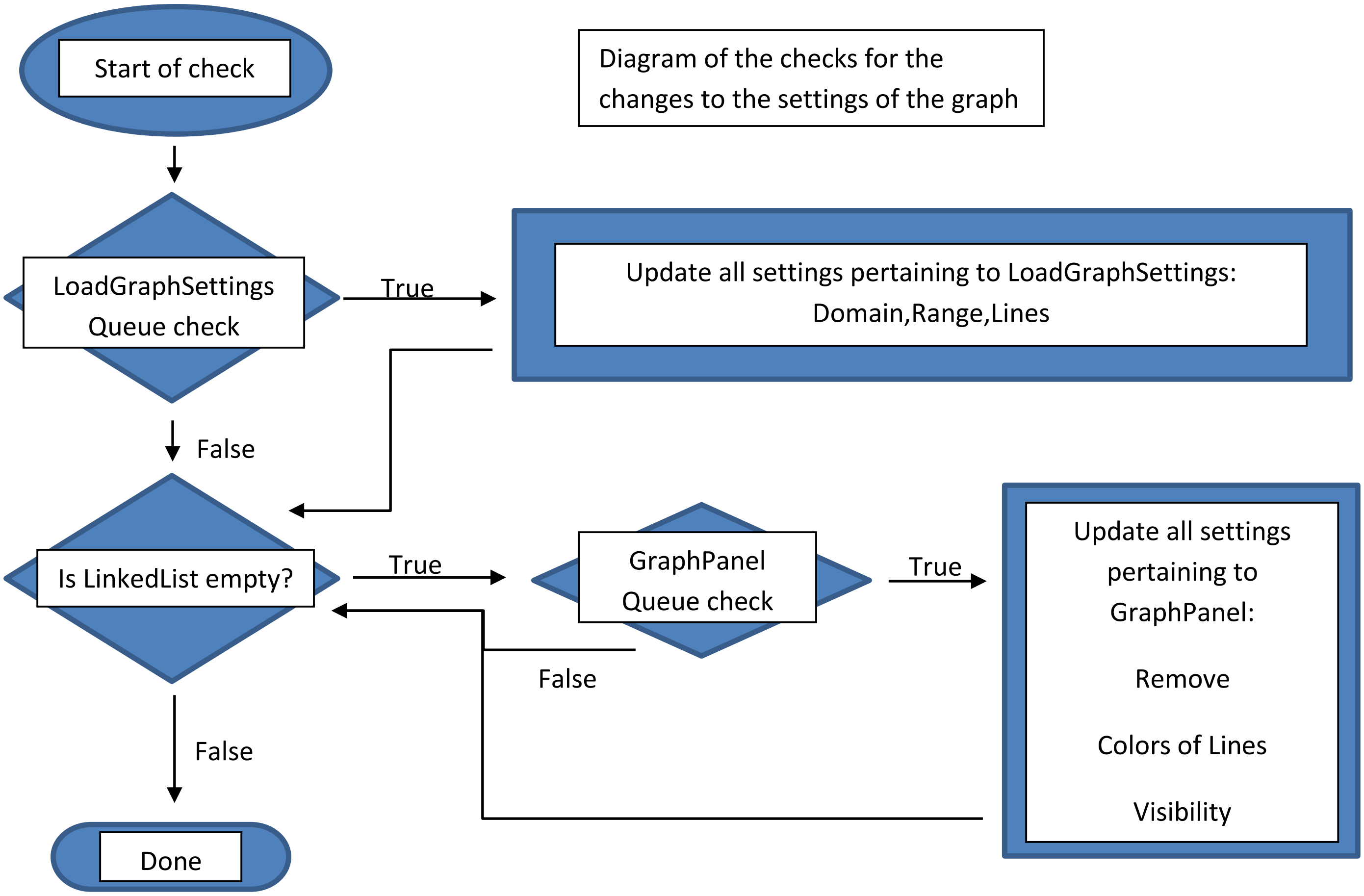


Diagram of the checks for the changes to the settings of the graph

Start of check
LoadGraphSettings Queue check
True
Update all settings pertaining to LoadGraphSettings: Domain,Range,Lines
False
Is LinkedList empty?
True
GraphPanel Queue check
True
Update all settings pertaining to GraphPanel:
Remove
Colors of Lines
Visibility
False
False
Done

Word Count: 240