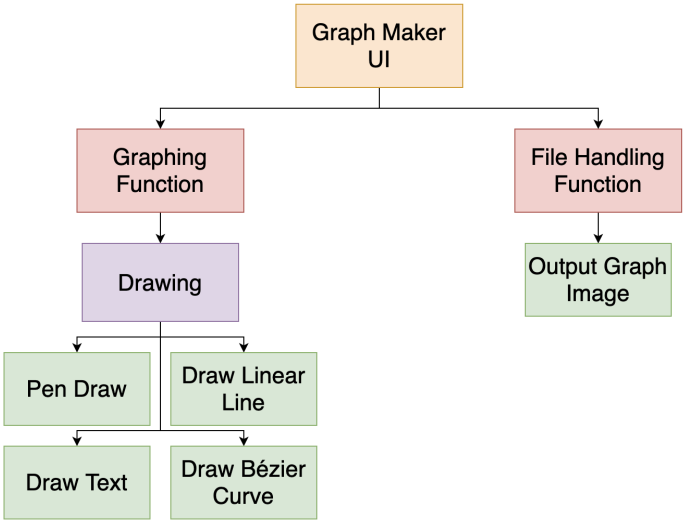
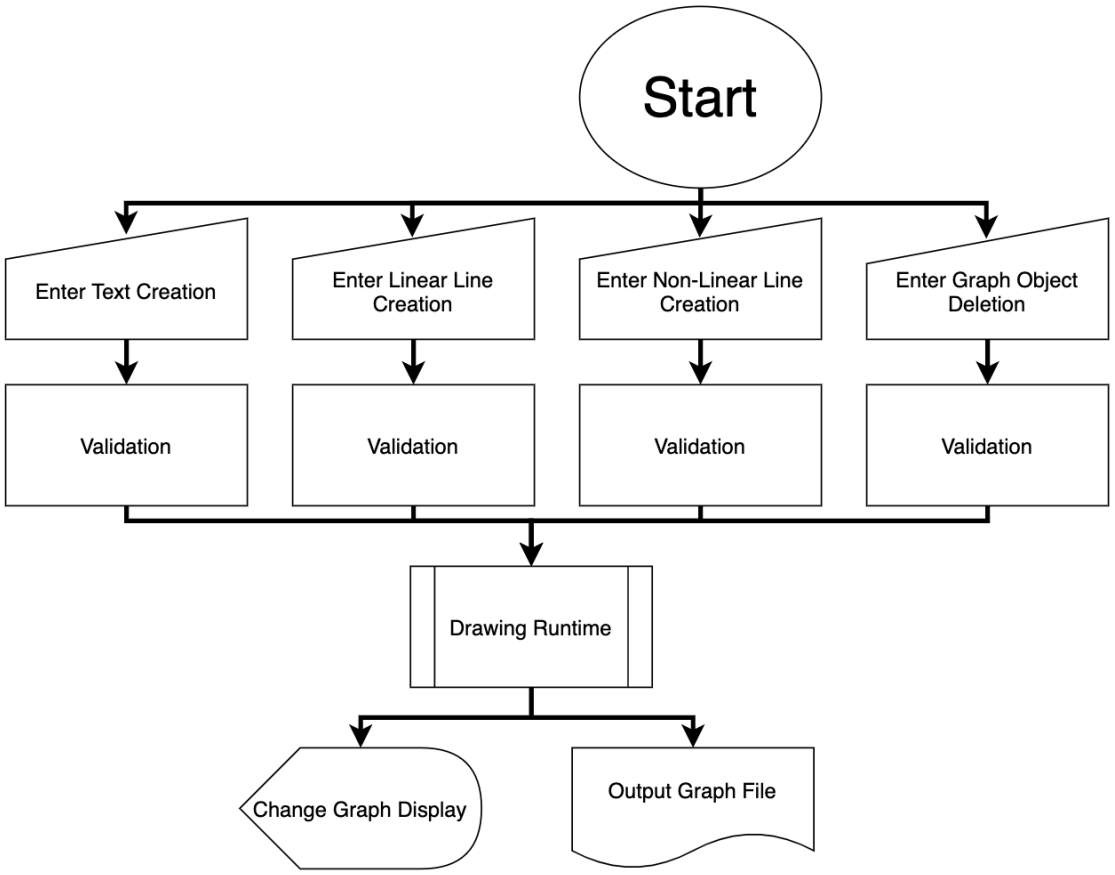
Criterion B Design

Structure Diagram

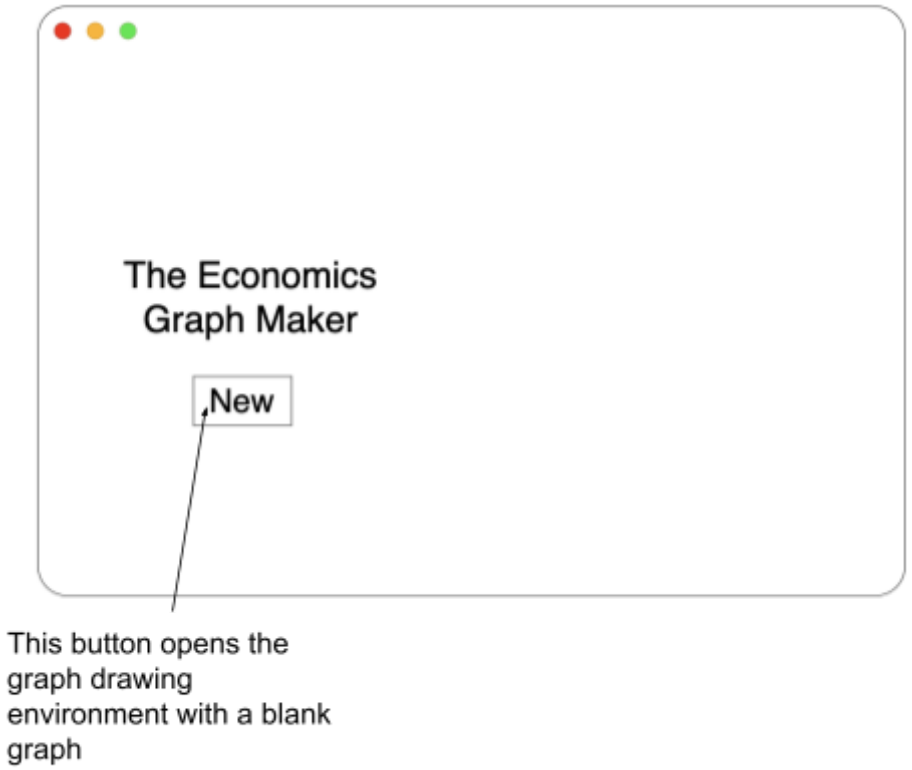


System Flowchart



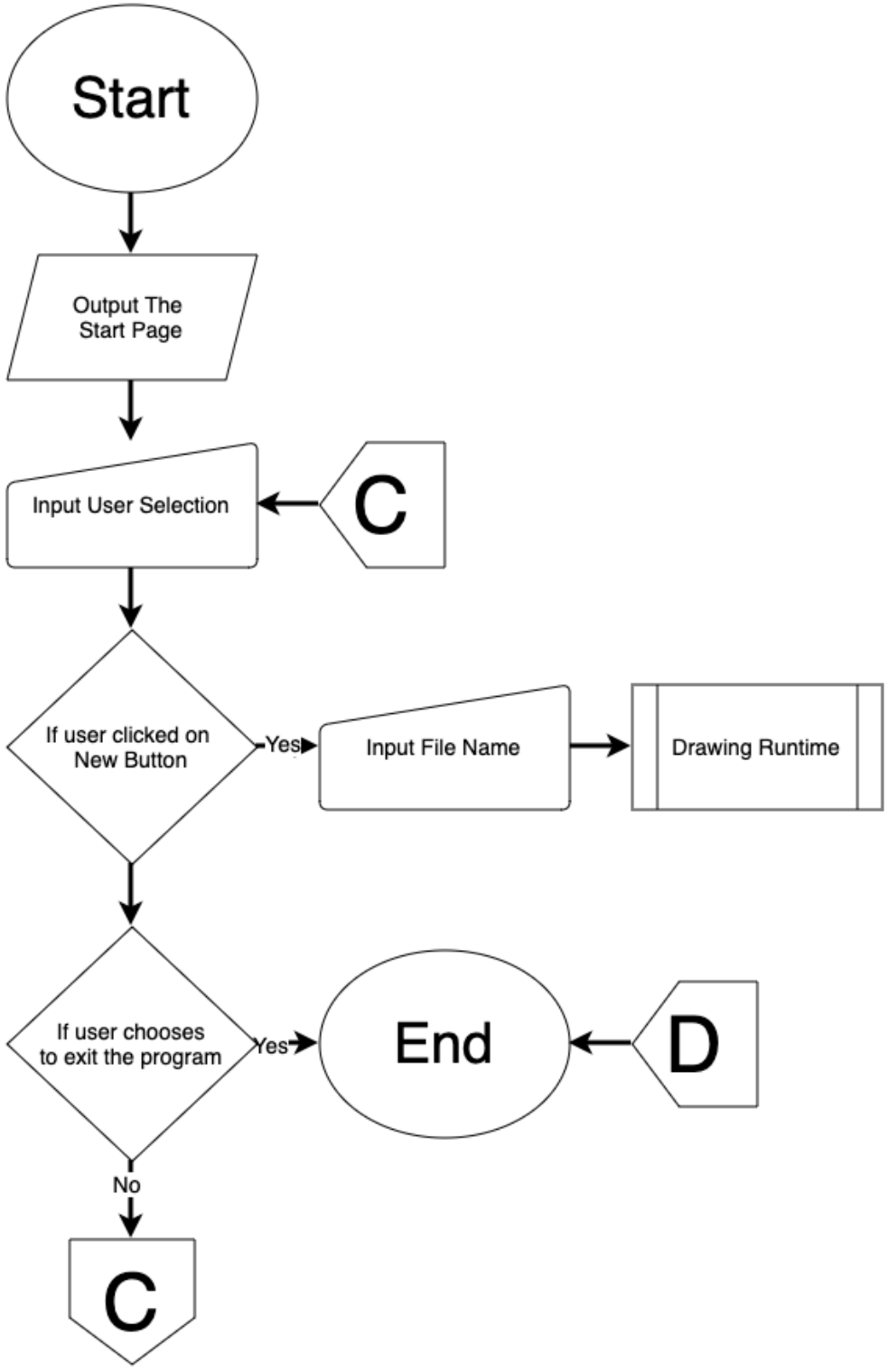
Data Flow Diagram

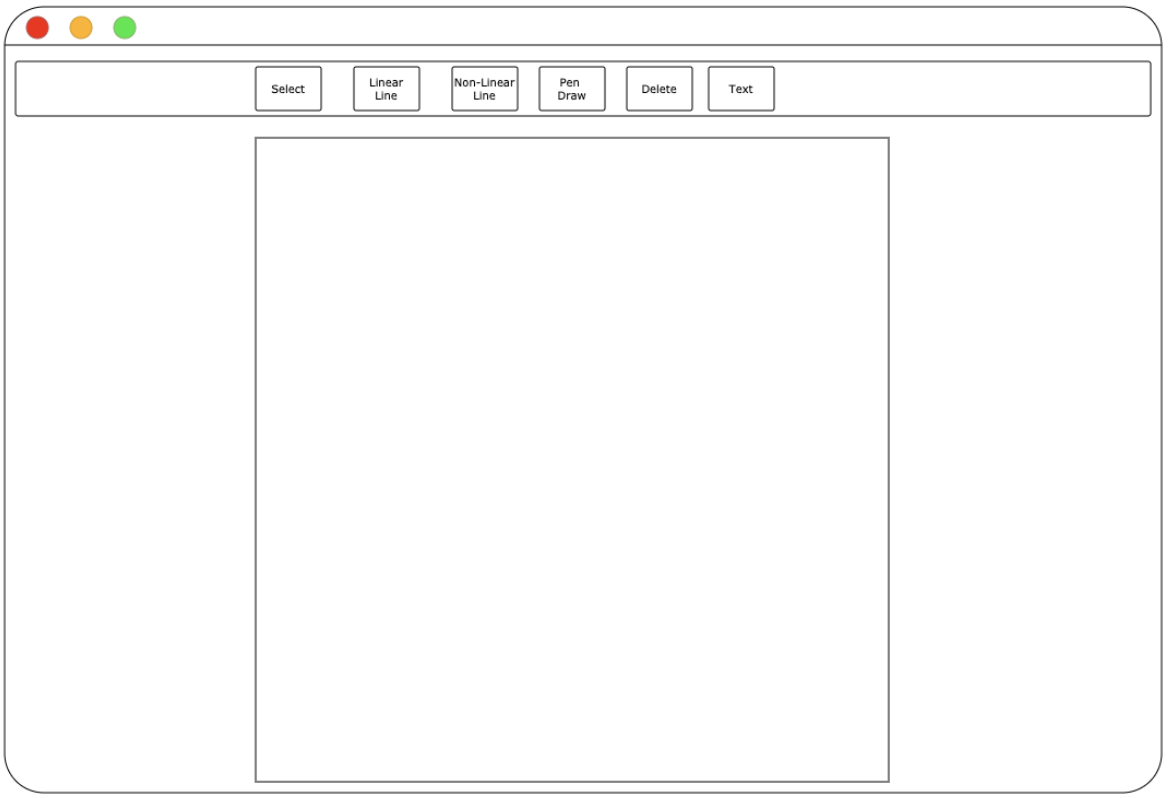
Prototype of the Application



Starting Page

The starting page will be where the user begins at the start of the program. It will serve as the foundation for all further activity.





The Graph Drawing Runtime

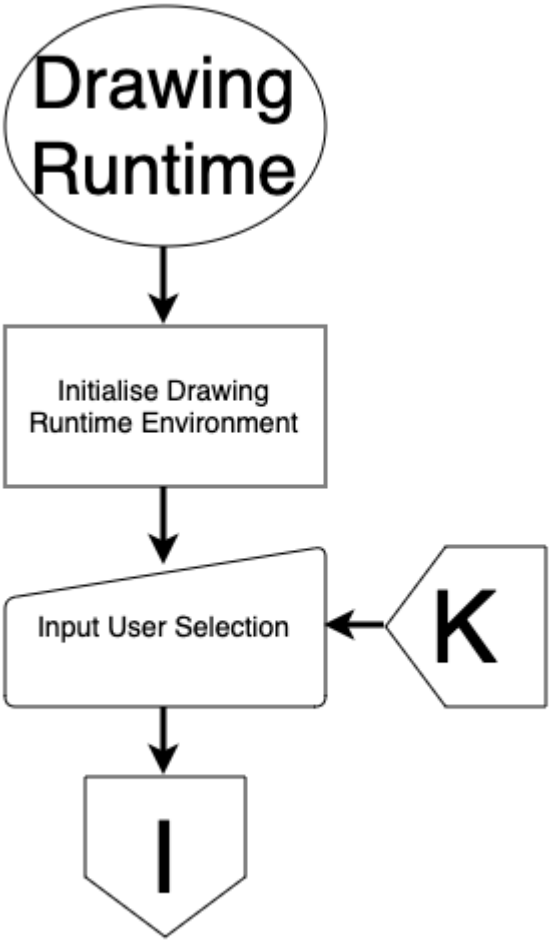
The graph drawing runtime is where all graph drawing and editing takes place in order to achieve the success criterias two and three (see Crit_A_Planning). When the user creates a new graph, the program first initialises the drawing runtime environment, such as the toolbar or the closures that contain the functions that will be used.

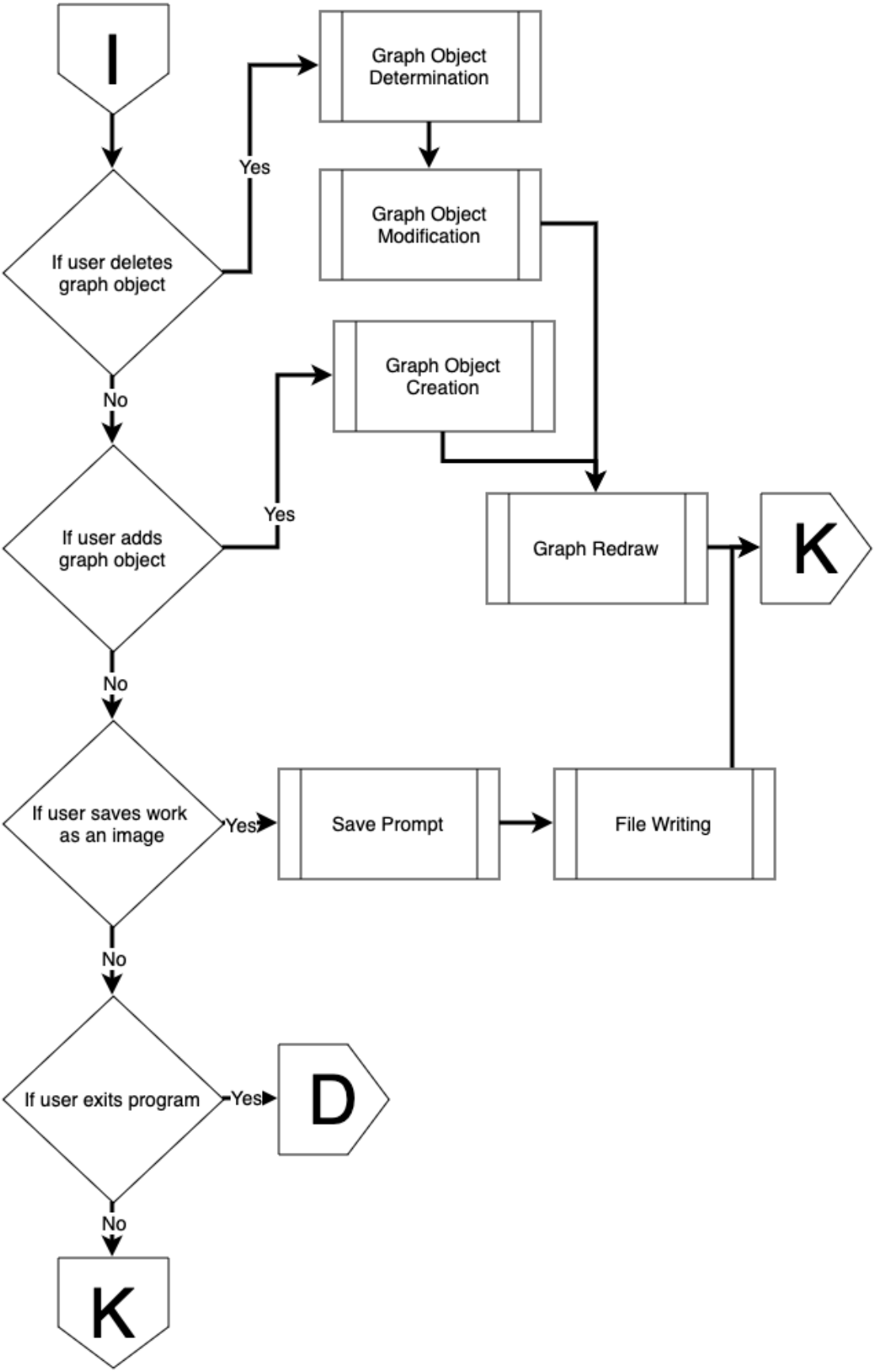
Once initialisation is completed, the program is ready for the next action that the user wishes to take.

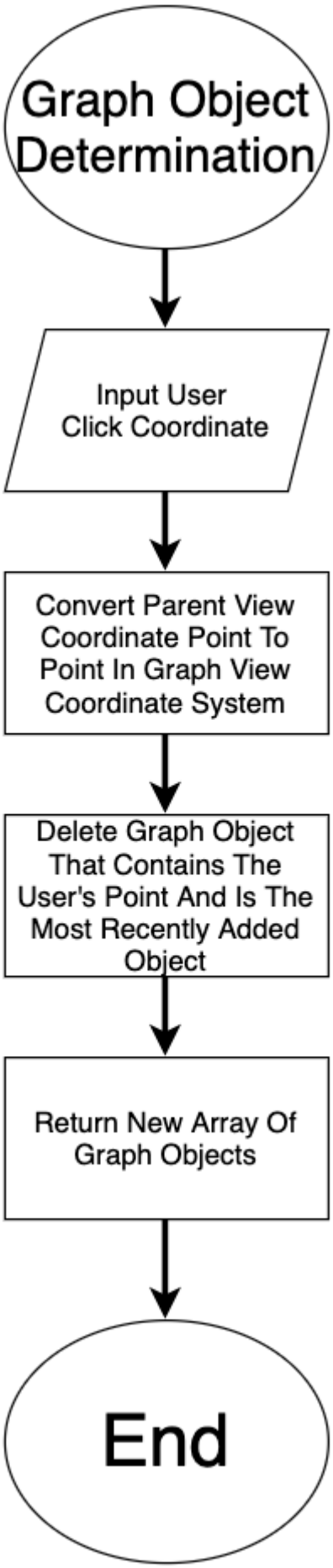
If the user chooses to add a graph object, then the program will create the new graph object, and then redraw the graph view.

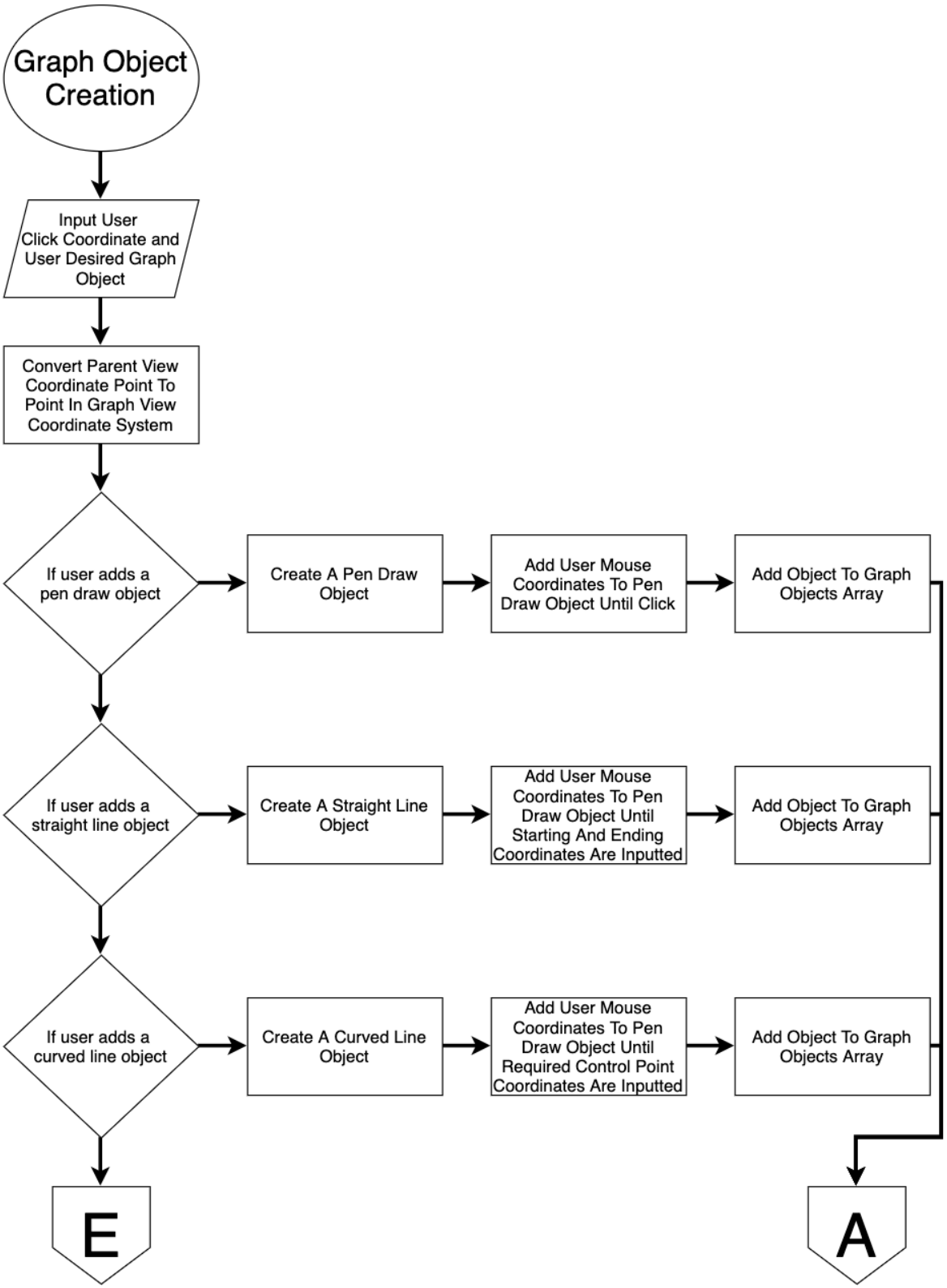
If the user saves their graph as an image file, then a popup will appear where the user can choose the location of where they would like to save their graph image. This will appear as a TIFF image file in order to achieve success criteria 4 (see Crit_A_Planning)

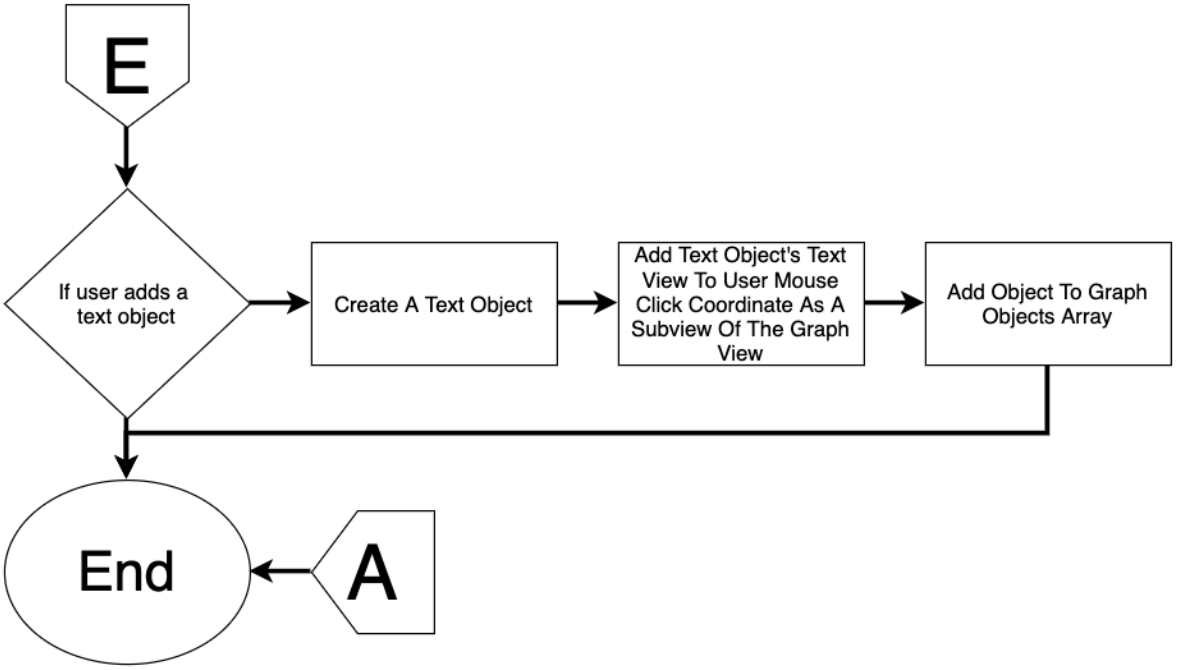
Finally, if the user modifies the graph view itself or the window of the drawing runtime, then the graph may need to be redrawn as well.


Drawing Runtime
Initialise Drawing Runtime Environment
Input User Selection
K
I

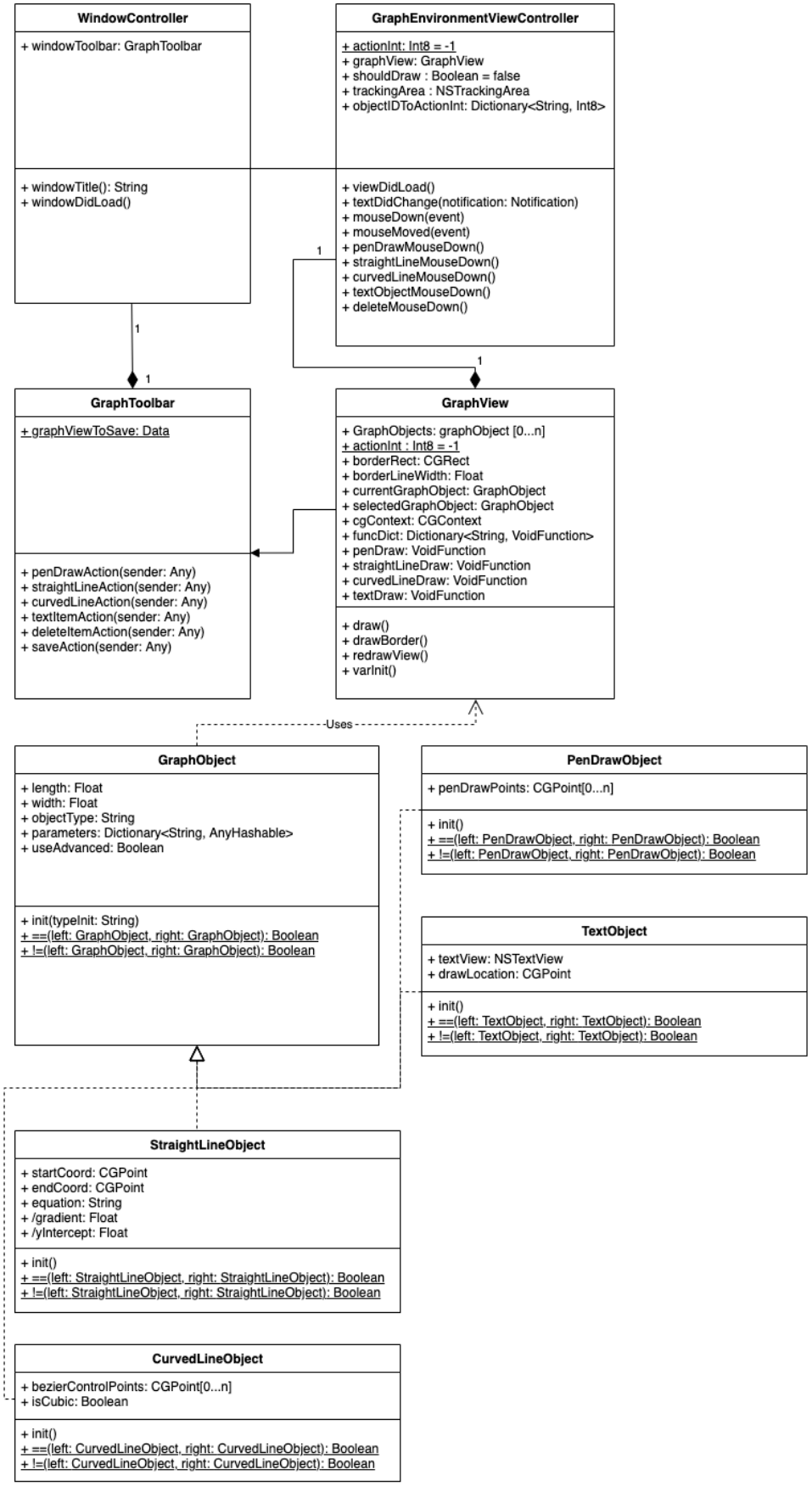

I
Graph Object Determination
Yes
Graph Object Modification
If user deletes graph object
No
Graph Object Creation
Yes
If user adds graph object
Graph Redraw
K
No
If user saves work as an image
Yes
Save Prompt
File Writing
No
If user exits program
Yes
D
No
K


Graph Object Determination
Input User Click Coordinate
Convert Parent View Coordinate Point To Point In Graph View Coordinate System
Delete Graph Object That Contains The User's Point And Is The Most Recently Added Object
Return New Array Of Graph Objects
End


Graph Object Creation
Input User Click Coordinate and User Desired Graph Object
Convert Parent View Coordinate Point To Point In Graph View Coordinate System
If user adds a pen draw object
Create A Pen Draw Object
Add User Mouse Coordinates To Pen Draw Object Until Click
Add Object To Graph Objects Array
If user adds a straight line object
Create A Straight Line Object
Add User Mouse Coordinates To Pen Draw Object Until Starting And Ending Coordinates Are Inputted
Add Object To Graph Objects Array
If user adds a curved line object
Create A Curved Line Object
Add User Mouse Coordinates To Pen Draw Object Until Required Control Point Coordinates Are Inputted
Add Object To Graph Objects Array
E
A


E
If user adds a text object
Create A Text Object
Add Text Object's Text View To User Mouse Click Coordinate As A Subview Of The Graph View
Add Object To Graph Objects Array
End
A

UML Diagram


WindowController
+ windowToolbar: GraphToolbar
+ windowTitle(): String
+ windowDidLoad()
GraphEnvironmentViewController
+ actionInt: Int8 = -1
+ graphView: GraphView
+ shouldDraw : Boolean = false
+ trackingArea : NSTrackingArea
+ objectIDToActionInt: Dictionary<String, Int8>
+ viewDidLoad()
+ textDidChange(notification: Notification)
+ mouseDown(event)
+ mouseMoved(event)
+ penDrawMouseDown()
+ straightLineMouseDown()
+ curvedLineMouseDown()
+ textObjectMouseDown()
+ deleteMouseDown()
1
1
1
1
GraphToolbar
+ graphViewToSave: Data
+ penDrawAction(sender: Any)
+ straightLineAction(sender: Any)
+ curvedLineAction(sender: Any)
+ textItemAction(sender: Any)
+ deleteItemAction(sender: Any)
+ saveAction(sender: Any)
GraphView
+ GraphObjects: graphObject [0...n]
+ actionInt : Int8 = -1
+ borderRect: CGRect
+ borderLineWidth: Float
+ currentGraphObject: GraphObject
+ selectedGraphObject: GraphObject
+ cgContext: CGContext
+ funcDict: Dictionary<String, VoidFunction>
+ penDraw: VoidFunction
+ straightLineDraw: VoidFunction
+ curvedLineDraw: VoidFunction
+ textDraw: VoidFunction
+ draw()
+ drawBorder()
+ redrawView()
+ varInit()
Uses
GraphObject
+ length: Float
+ width: Float
+ objectType: String
+ parameters: Dictionary<String, AnyHashable>
+ useAdvanced: Boolean
+ init(typeInit: String)
+ ==(left: GraphObject, right: GraphObject): Boolean
+ !=(left: GraphObject, right: GraphObject): Boolean
PenDrawObject
+ penDrawPoints: CGPoint[0...n]
+ init()
+ ==(left: PenDrawObject, right: PenDrawObject): Boolean
+ !=(left: PenDrawObject, right: PenDrawObject): Boolean
TextObject
+ textView: NSTextView
+ drawLocation: CGPoint
+ init()
+ ==(left: TextObject, right: TextObject): Boolean
+ !=(left: TextObject, right: TextObject): Boolean
StraightLineObject
+ startCoord: CGPoint
+ endCoord: CGPoint
+ equation: String
+ /gradient: Float
+ /yIntercept: Float
+ init()
+ ==(left: StraightLineObject, right: StraightLineObject): Boolean
+ !=(left: StraightLineObject, right: StraightLineObject): Boolean
CurvedLineObject
+ bezierControlPoints: CGPoint[0...n]
+ isCubic: Boolean
+ init()
+ ==(left: CurvedLineObject, right: CurvedLineObject): Boolean
+ !=(left: CurvedLineObject, right: CurvedLineObject): Boolean

Test Plan

Test Type	Success Criteria Being Tested	Nature of Test	Example (include steps to create output) (abnormal data should be tested)
GUI Operation Check	1	All of the GUI's components must work.	Normal Data: All GUI components will be used with as many permutations of actions as possible. Abnormal Data: The GUI must not allow the user to do certain actions when not warranted.
Graph drawing tool check	3	All of the program's drawing tools must work.	Normal Data: The pen draw, straight line and curved line creation tools must work, and text creation must work without visual errors. Abnormal Data: The tools must not operate when attempted to be used outside of the graph view. For example, text editing must end when the user clicks out of the text box
Graph creation	1, 2, 3,	The program is able to create graphs. The program must require less mouse movements and clicks compared to previous solutions. Test users must also state that the GUI is user friendly.	Normal Data: The creation of multiple economic graphs selected at random from the Tragakes Supplementary Resource Booklet. A graph file will be created, and the graphs are to be recreated as faithfully as possible, then exported as a TIFF file. Abnormal Data: The attempt to draw graph objects outside of the area of graph creation area through the user clicking outside of the area of the graph creation area
Assisting students with creating economics graphs digitally, both for homework and their Economic IAs	3	The program must create economic graphs that cannot be deducted marks. It must also use templates that are not complete enough for plagiarism to take place. This will be at my client's	Normal Data: The program must allow for graphs to be created under users with the needed economics knowledge and must not lose any marks under the IBDP Economics Mark Schemes.

		discretion.	
Graphs must be exportable as an image file	4	The program must export the graph in whole without corruption, and without image artefacts.	Normal Data: Test graphs that use a wide range of graphing features must export as .tiff files without unexpected artefacts or corruption.
Utilising error handling to ensure that the program cannot break easily	5	The program must not crash under malicious activity	Maxing out the number of graph objects.

[Words not included in word count]