Design Document

Authors: Richard Dyer

Zisis Damianidis

Tsanko Hadzhiev

Created: 02/12/2016

Last Revision: 16/12/2016

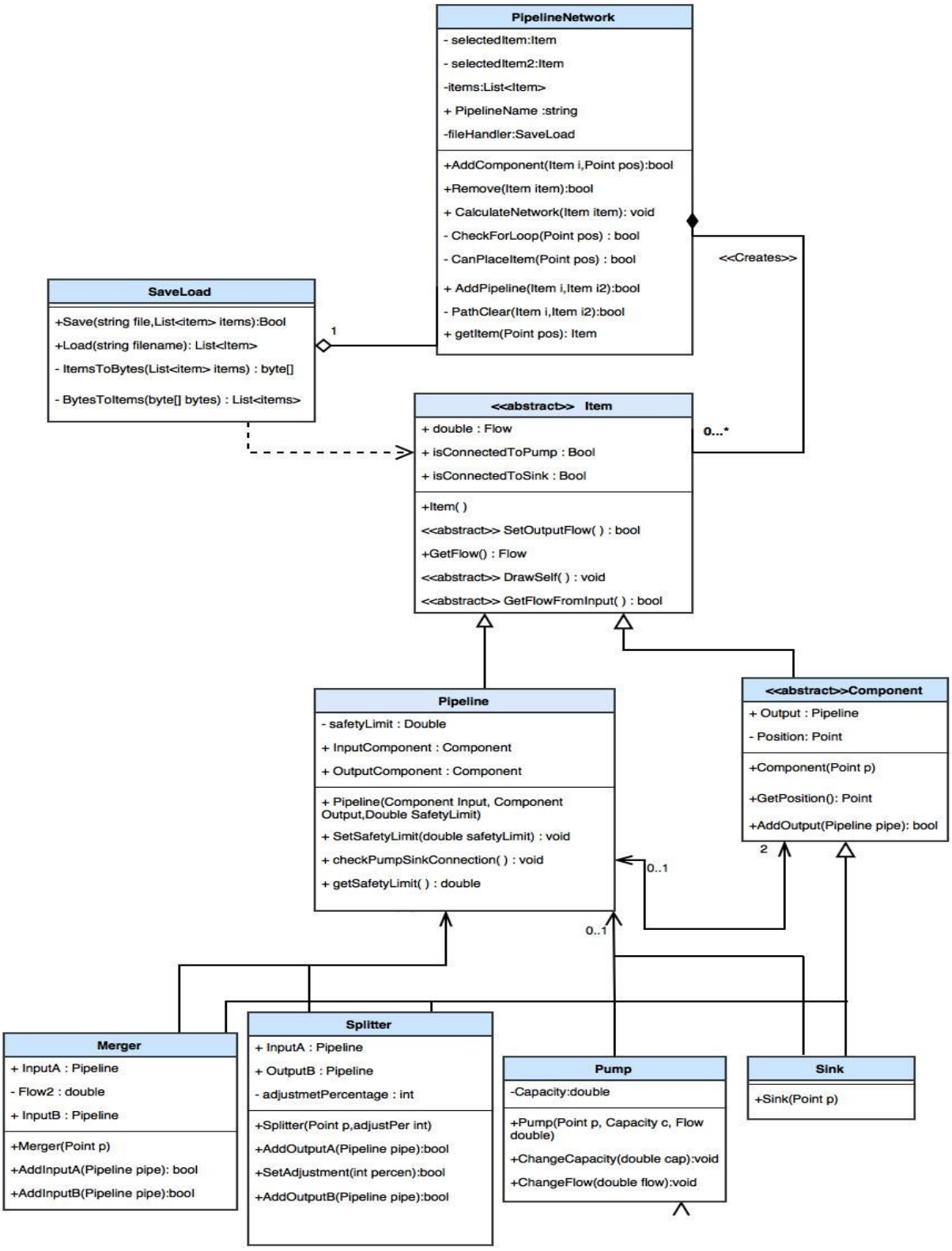
INTRODUCTION **2**
Purpose of Design Document 2
Class Diagram **3**
Class Description **4**
PipelineNetwork 4
SaveLoad 5
Item (Abstract) 6
Pipeline 7
Component (Abstract) 8
Pump 8
Merger 9
Splitter 9
Sink 10
Sequence diagrams **11**
Save As 11
Add Component 12
Change Flow 13

INTRODUCTION

Purpose of Design Document

The Design Documents describes and shows the details of the classes, methods and variables involved in building the application. The Class Description provides a description for all the classes. The Method Description clarifies some of the methods. In the interaction section some of the more abstract methods are shown how they will interact with each other.

Class Diagram



Class Description

Method/Variable	Name	Description	Returns/Type
	PipelineNetwork		
	PipelineNetwork is the main class, manager class which is in charge of Adding/Removing components,checking for placement issues, and telling the components how and where to draw themselves.		
Variable	- SelectedItem	Represents the Item the user has selected in the Pipeline Network layout, or in the Component selection panel.	Object(Item)
Variable	- SelectedItem2	Represents the second selected Item, used for creating a pipeline.	Object(Item)
Variable	- Items	A collection of all the items placed in the Pipeline Network Layout.	List<Object(Item)>
Variable	+ PipelineName	The name of the Network. Used for saving and loading.	string
Variable	- fileHandler	Object used for handling saving and loading.	Object SaveLoad
Method	+AddComponent(Item i,Point pos):bool	This methods adds a new component(item) to the list of “items”. And will make the item draw itself. Will call “canPlaceItem” before executing.	True if component is added successfully or false if failed
Method	- CheckForLoop(Point pos):bool	Checks if the network would loop if pipeline was placed	Returns true if network would loop.

Method	Remove(Item i): bool	Goes through list of items and removes the selected item..	Returns true or false, based on the result. If deleting was successful returns true else return false
Method	- calculateNetwork()	Calculates and orders items to either draw themselves a specific color, or set values based on connections.	
Method	- canPlaceItem(Point pos):bool	Calculates if a component can be placed in the area.	Returns true if component can be placed, returns false if area is already occupied.
Method	+AddPipeline(Item i, Item i2):bool	Checks if both items can have pipelines and then uses “PathClear” and “checkForLoop” then creates a pipeline connected to the two items and tells the pipeline to draw itself.	Returns true if pipeline was added. Returns false if items already has pipelines or if a loop would have occurred or if path was not clear.
Method	- PathClear(Item i, Item i2):bool	Check if a pipeline were drawn in between the two items if it intersects with a component.	Returns true if no intersection would occur.
Method	+ getItem(Point pos)	Gets the item with the sought after position.	Returns the item with the position, returns null if no item was found.
	SaveLoad		
	The purpose of this class is to Save/Load a pipeline network to/from a binary file.		

Method	+Save(string file,List<item> items):Bool	Uses “ItemsToBytes” then saves the Items to a file.	bool
Method	+Load(string filename): List<Item>	Uses “BytesToItems” then returns a List of items.	Returns a list of litems.
Method	- ItemsToBytes(List<item> items) : byte[]	Turns the list of items into bytes.	Returns null if unsuccessful.Otherwise returns a byte array.
Method	- BytesToItems(byte[] bytes) : List<items>	Turns bytes into a list of items	Return null if unsuccessful. Otherwise returns a list of items.
	Item (Abstract)		
	Item is the base class for all components and pipelines.		
Variable	Flow	Each Item has a flow	int
Variable	IsConnectedToPump	If this item or one of its connecting items are connected to a pump	bool
Variable	IsConnectedToSink	If this item or one of its connected items are connected to sink.	bool
Method	<<abstract>> SetOutputFlow( ) : bool	Each item needs to be able to set the flow of the connected component.	Returns false if no output item or if output item is a sink.
Method	<<abstract>> DrawSelf( ) : void	Each type item needs to have a draw itself method.	Void
Method	<<abstract>> GetFlowFromInput( ) : bool	Each item type needs to be able to get its flow from the input component.	True if successful
Method	+ GetFlow() :double	Gets the items current	Returns the flow

		flow.	or -1 if no it has no flow.
	Pipeline		
	Pipeline inherits from Item class. Pipeline has extra parameters such as an Input and OutPut Component.		
Variable	SafetyLimit	The safety limit of the pipeline if the pipeline flow is over the safety limit the pipeline will know to draw itself red.	double
Variable	InputComponent	The component connected to the pipeline's input.	Object Component
Variable	OutputComponent	The component connected to the pipeline's output.	Object Component
Constructor	+Pipeline(Component comp, Component comp2, SafetyLimit Double)	This is the constructor for Pipeline class , it needs to know which two components it will be placed. And the Safety limit to be able to create a Pipeline object.	
Method	+SetSafetyLimit(double safetyLimit):void	Assigns the given value to the safety limit.	void
Method	+ CheckPumpSinkConnection( ):void	Checks if the pipelines input or output is a sink or pump if yes, then it will change all connected item's "IsConnectedToPump"/ "IsConnectToSink" variables to true.	void

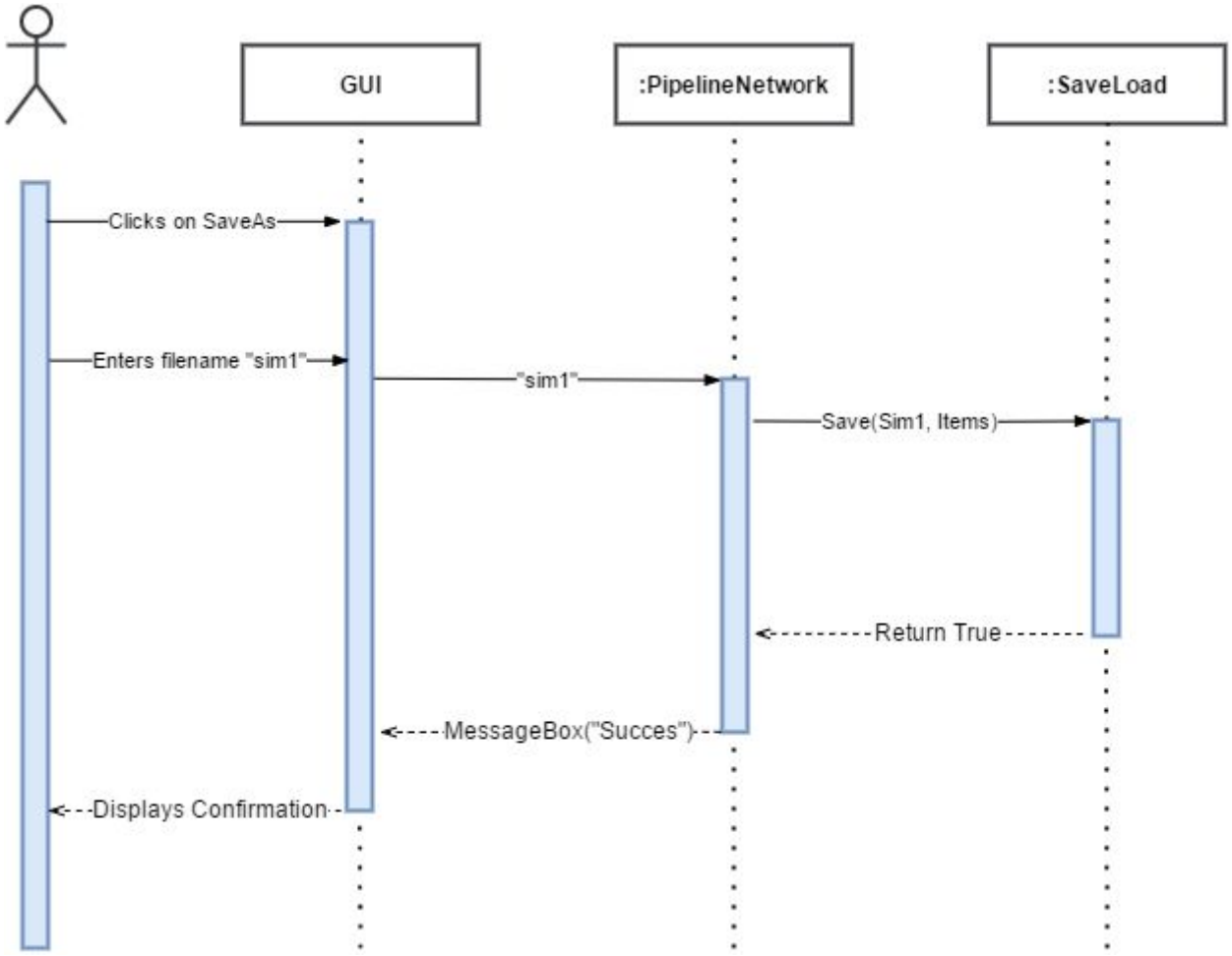
Method	+ getSafetyLimit( ) : double	Returns the current safety limit of the pipeline.	double
	Component (Abstract)		
	A component is the base class for all components. It also inherits from Item but has additional attributes such as a point.		
Variable	+ Output	The Pipeline in the components output.	Object(Pipeline)
Variable	- Position	The location of the component.	Point
Constructor	+Component(Point p)	The location of the component is needed to set a component	void
Method	+GetPosition():Point	Returns the location of the selected component	Point
Method	+AddOutput(Pipeline pipe): bool	Set the Component's output pipeline.	True if successful, false if it failed or component already has a pipeline attached.
	Pump		
	Stores information for pump components. Pump has capacity as extra parameter. Pump is the only component where a flow can be directly set. And is the origin of a networks flow.		
Variable	- Capacity	The maximum flow the pump can output.	
Constructor	+Pump(Point p, Capacity c, Flow double)	A pump needs a point, capacity, and flow to be created.	void

Method	+ChangeCapacity(double cap):void	Change the capacity of the pump.	void
Method	+ChangeFlow(double flow):void	Change the flow produced by the pump and change the flow of all connected items.	
	Merger		
	Component merging two pipelines into one.		
Variable	+ InputA	The first input of the merger.	Object (Pipeline)
Variable	+ InputB	The second input of the merger	Object(Pipeline)
Variable	- flow2	If the first flow is filled when the second input changes this component's flow it will be set instead of the first.	double
Constructor	+Merger(Point p)	Pump constructor, similar to the base class.	
Method	+AddIntput(Pipeline pipe):void	This method assigns input to the selected component.	void
Method	+AddIntputB(Pipeline pipe):void	Assigns inputB to this component	void
	Splitter		
	Splitter splits one pipeline into two (can also adjust the ratio of the split)		
Variable	-InputA	The input of the splitter	Object (Pipeline)
Variable	-OutputA	This is the first output of the splitter.	Object (Pipeline)

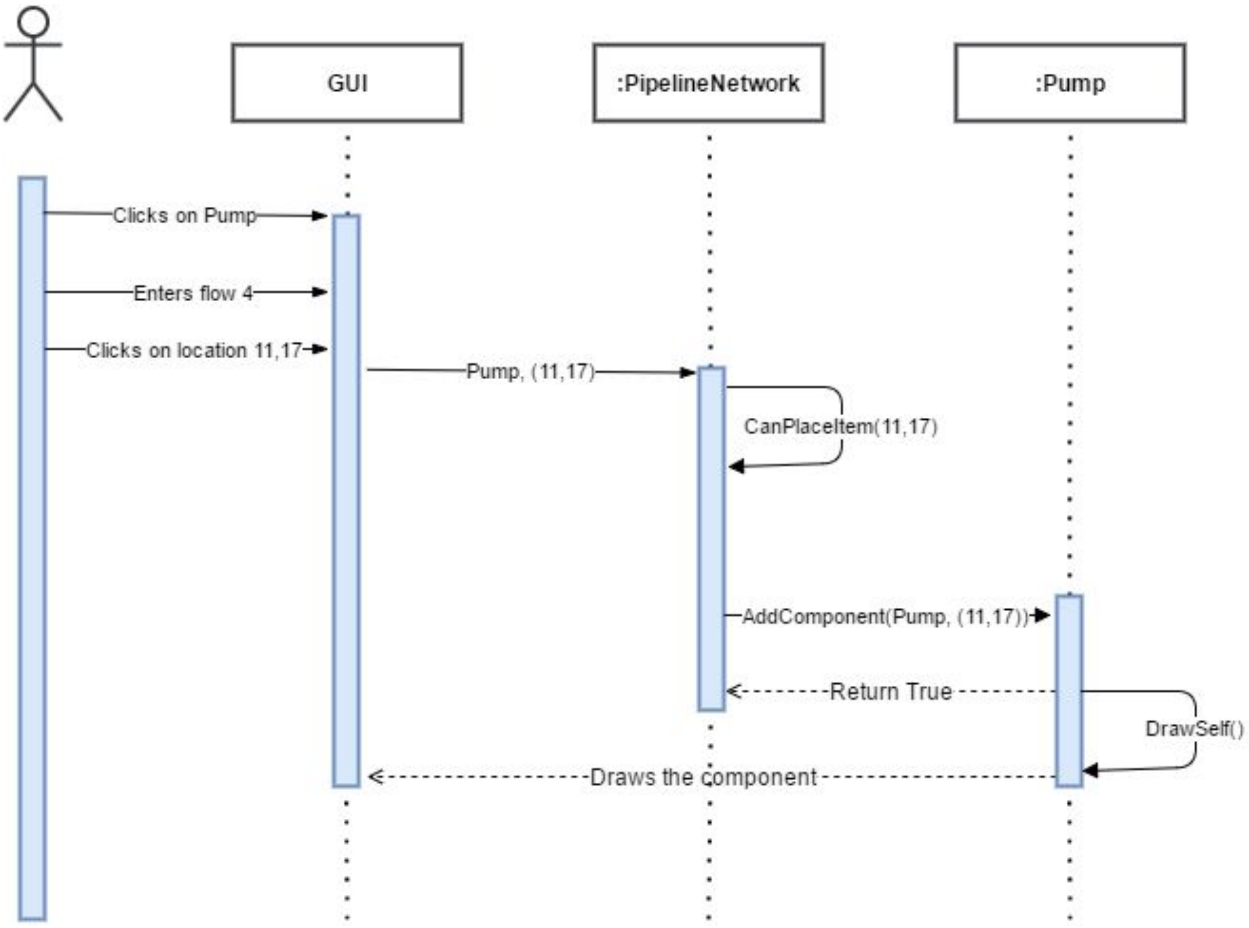
Variable	-OutputB	This is the second output of the splitter	Object (Pipeline)
Variable	-adjustPercenatage	This variable adjust the flow of output pipelines.	int
Constructor	+Splitter(Point p, adjustPer int)	Splitter constructor is similar to the base constructor , the only difference is that Splitter has adjustPercentage variable instead of flow.	
Method	+AddOutputA(Pipeline pipe)	Sets the first Splitter's output pipeline.	bool
Method	+SetAdjustment(int percen)	Sets the adjustmentPercentage value to the given value	void
Method	+AddOutputB(Pipeline pipe)	Set the second Splitter's output pipeline.	bool
	Sink		
	Sink is the last component of a pipeline network.		
Variable	-Input	The input of the splitter	Object (Pipeline)
Constructor	+Sink(Point p)	Constructor only has location , no flow.	
Method	+AddInPut(Pipeline pipe)	This method assigns input to the selected sink.	bool

Sequence diagrams

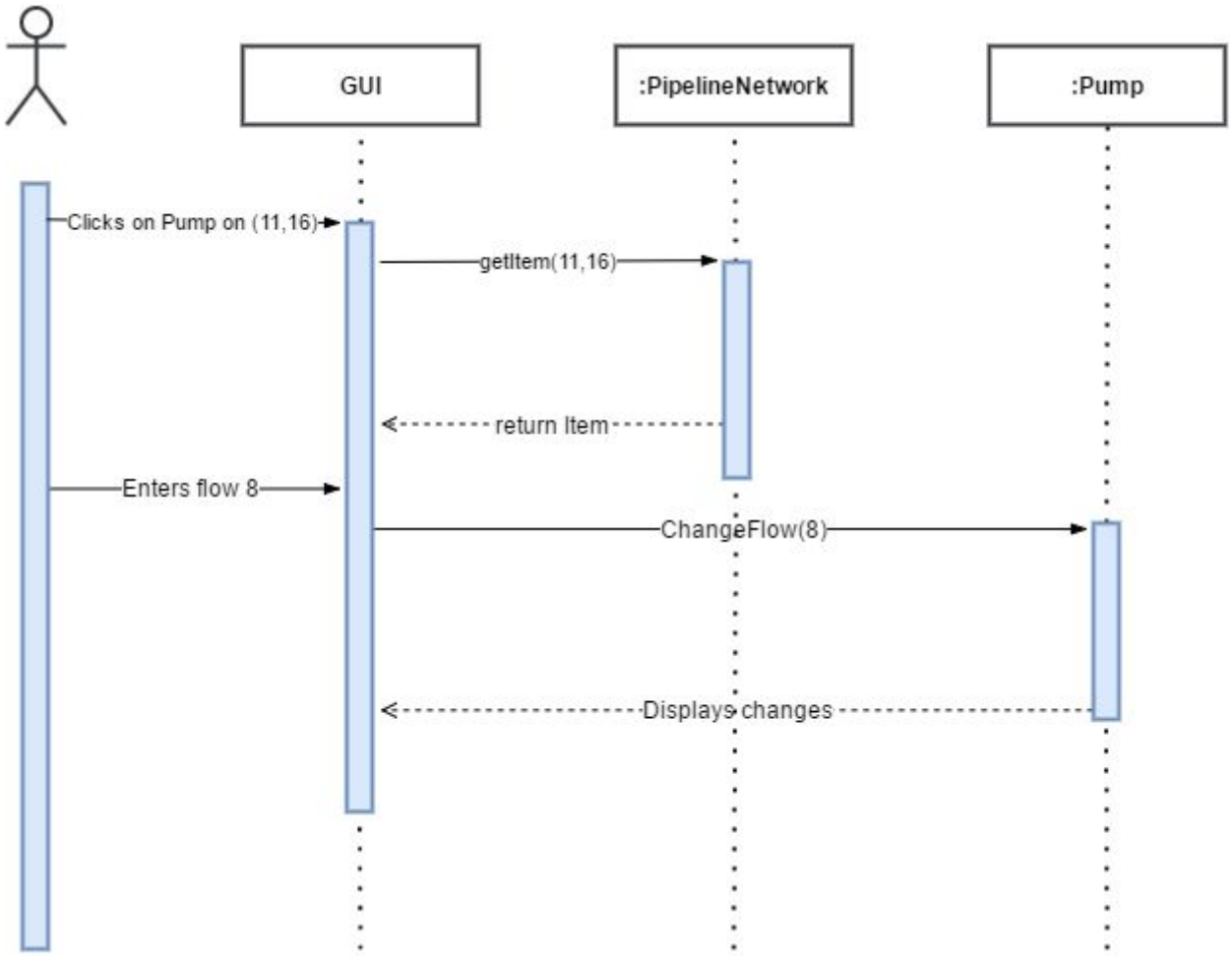
Save As



Add Component



Change Flow


GUI
:PipelineNetwork
:Pump
Clicks on Pump on (11,16)
getItem(11,16)
return Item
Enters flow 8
ChangeFlow(8)
Displays changes