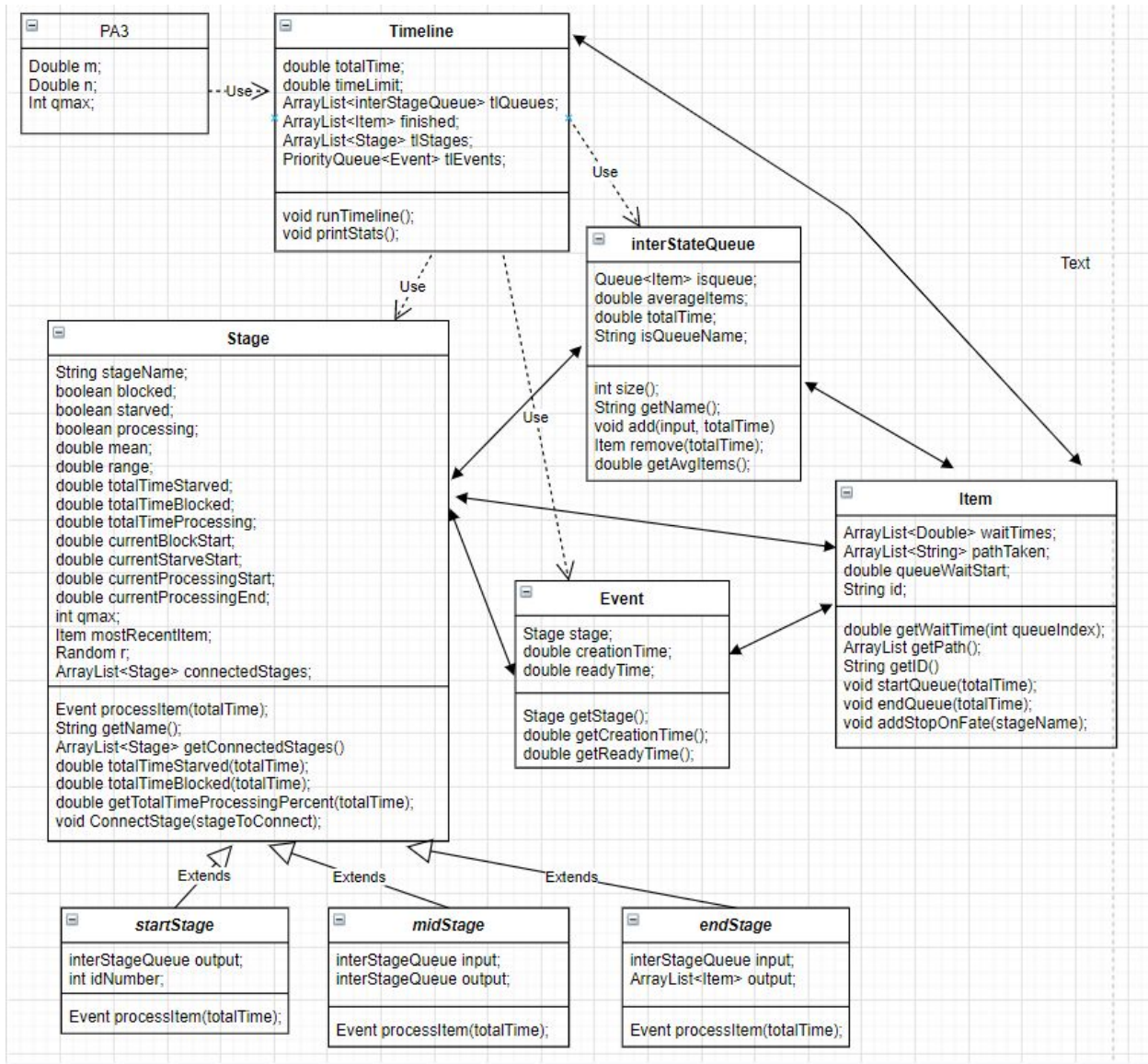


Assignment 3 Report

Scott Lonsdale (C3303788)

UML Diagram



Comment on your use of Inheritance and Polymorphism and how you arrived at the particular Inheritance/Polymorphic relationships you used in your program.

When I was designing the initial class structure I originally had only one Stage class for all of the case, but for simplicity and incorporation of inheritance I split them up into begin, mid and end states. I used a similar polymorphic implementation of Comparable from Assignment 2 for the Event class to overwrite the compareTo() function to compare events according to their finish times.

How easy will it be to alter your program to cater for a production line with a different topology – e.g. one with 4 stations or 10 stations, or one that has stations S3a/b/c rather than just S3a/b?

Adapting my program to these cases would be relatively easy for most cases, just by adapting the Timeline() class to initiate the correct queues, stages and stage connections necessary. The output function for certain paths might need to be radically changed depending on the needed output for the new system as well.

How easy will it be to alter your program to cater for a production line that is more complicated than the “straight line” item processing that your program does – e.g. one that involves taking two different types of items and assembling them to make a new type of item? Would you design your program differently if you had known that this might be a possibility?

Depending on the proposed changes I think that my design could handle most derivation of the production line with a few tweaks. Using multiple triggers inside the check for starvation for each input queue going into the state could solve the given complication for example. Allowing for multiple end states would also be relatively easy, as all you would have to do is use the same output array for each of the end states. Loops should already work within the system although the print functions for paths would need to change. Leading into that, I think the main change I would have is implement a path output that could be initialised easier as the current one is messy and needs editing for most changes

