# Disk scheduling

## Introduction

For week 1 assignment we have created an application which simulates different algorithms for processor scheduling. The application allows users to simulate 4 different algorithms (FCFS, SJB, Preemptive and Non-preemptive)

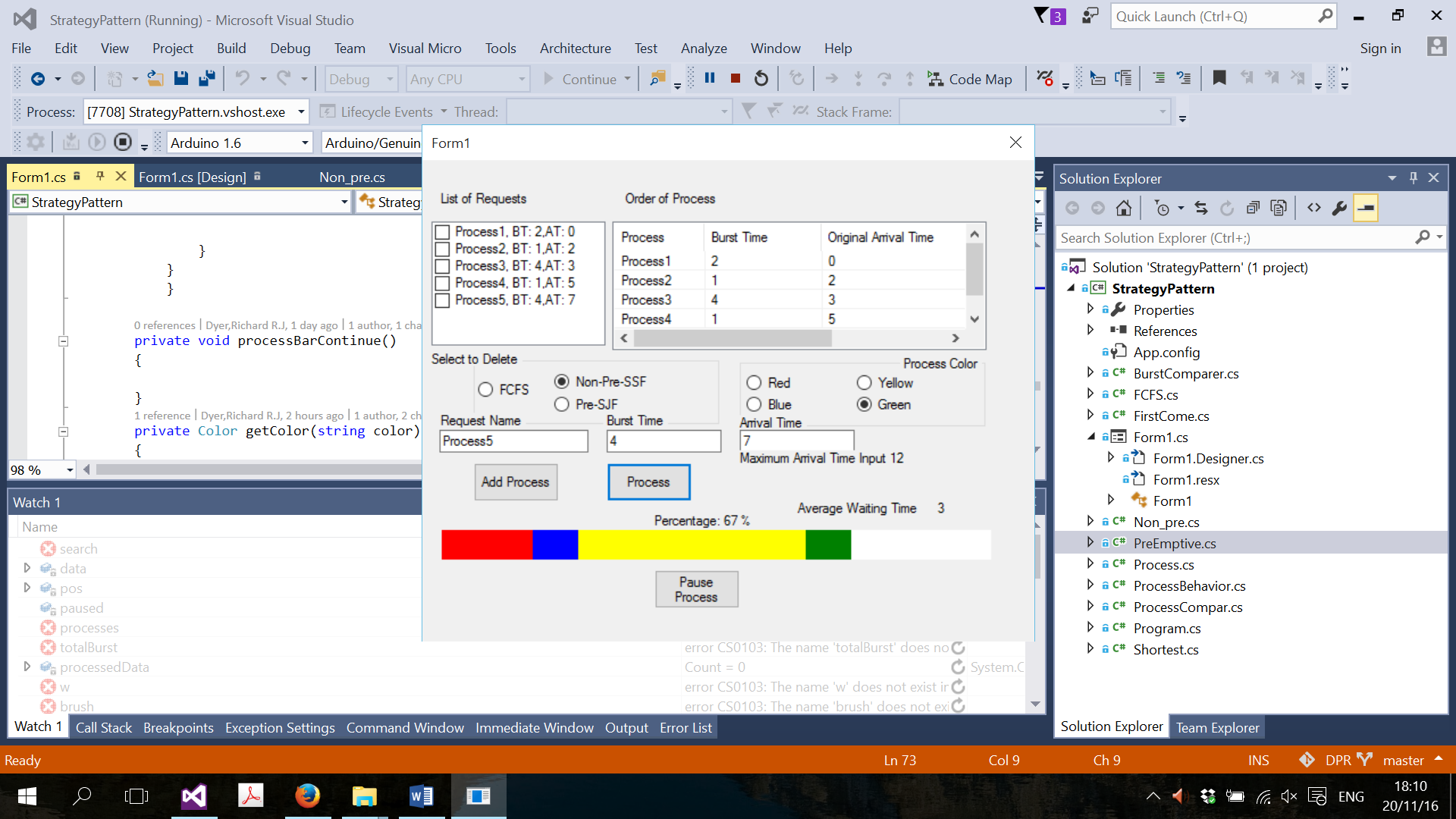
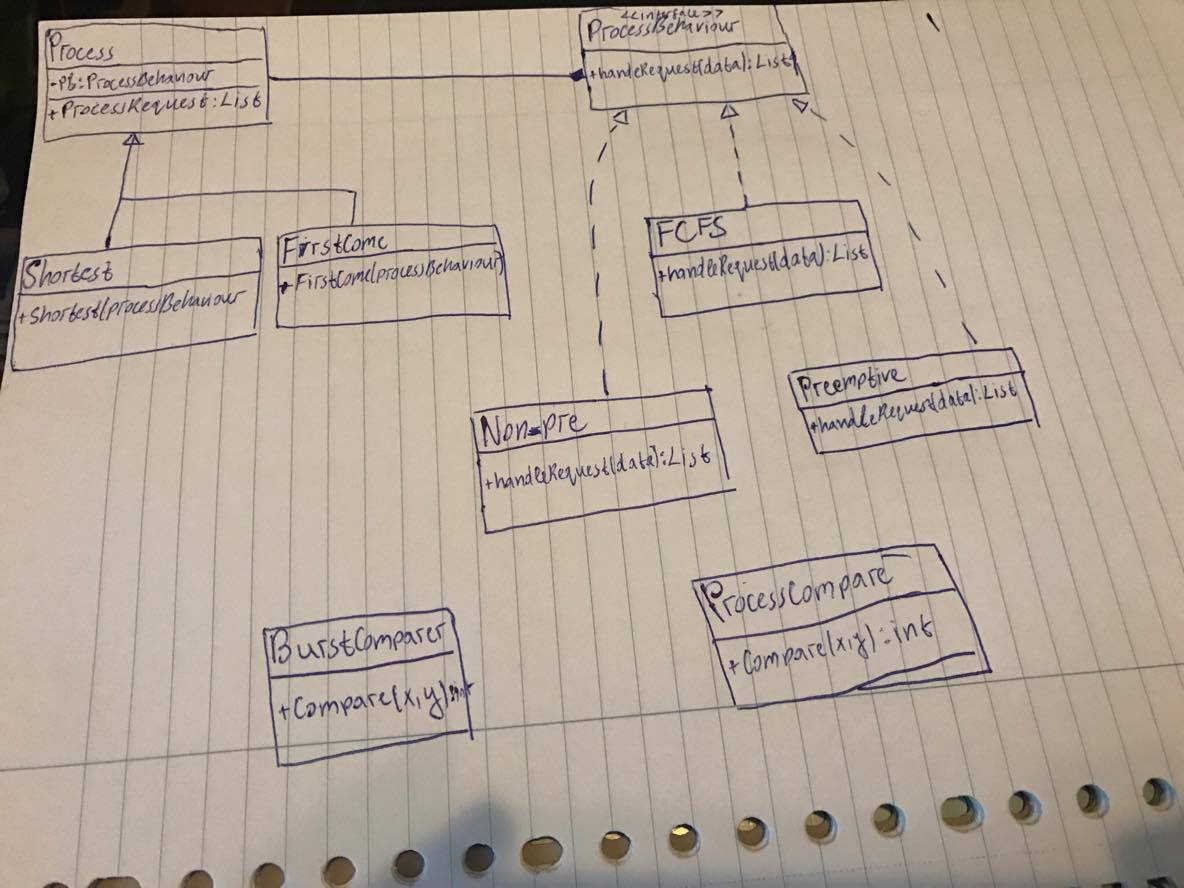


Fig 1.1

The app provides wide variety of customizations to users:

* User can add manually add Burst time and Arrival time of the processes, the application checks that the user has entered the correct arrival times.
* For better visualization user, can chose between different colors, this way each process execution can be shown in different color.
* User can delete Processes
* User can see each process waiting time and more details
* User can pause the process and then continue the process
* Program calculates waiting time
* Program shows the percentage complete

## UML



## Problems

* Existentiality

Extensibility is how the application can be developed/extended easily.

The reason we had problem with existentiality is because the application supports many algorithms, and we had to come up with solution which combines and separates them. This way program doesn’t get too big and can be easily extended.

* Reusability

Reusability is how the code/class that already created can be used again.

We had the same problem with making the program code reusable, but we have decided to use interface, therefore each component is similar but has different functionality and can be reused.

## Improvements

The program could compare exactly what was better instead of user having to compare himself.

And program could have a timer for each process and make some sort of log.

Program could have better enacted reusable code. As many parts in the Pre-emptive algorithm and non-preemptive algorithm were the same.

## Testing

Because processes are added manually application testing can be done very easily.

We tested our app by adding one of each process to the list and executing it. The application passed the test successfully.

We included a test project to test the algorithms, after doing the test, all algorithms passed the test.

## Reflection

We had some background in Design patterns from System Development 3, but everything was only theoretical. After doing this assignment, we learnt how useful this approach could be. For example we basically had two sorts of algorithms first come or shortest first. We could apply which either one we wanted to act on but then we were also able to include specific behaviors for the shortest first algorithms (pre-emptive or non-preemptive). Uses this design not only made the code easier for someone else to follow but it made it easier to code and it was more organized and practical.