School of Engineering
TEMASEK POLYTECHNIC

Diploma in Computer Engineering

Computer Architecture & Operating Systems Project

Declaration of Originality

By submitting this work, I / we declare that

- I am / we are the originator(s) of this work.
- I / we have appropriately acknowledged all other original sources used in this work.
- I / We understand that Plagiarism is the act of taking and using the whole or any part of another person's work and presenting it as my/ our own without proper acknowledgement.
- I / We understand that Plagiarism is an academic offence and if I am/we are found to have committed or abetted the offence of plagiarism in relation to this submitted work, disciplinary action will be enforced.

Student Name:	Admin No:	Student Signature	Date
Teoh Hong Wei	1501429C		14/2/18
Daniel Ng	1505178B		14/2/18
Amirul Haqim	1503214G		14/2/18
Timothy James Conde	1500358E		14/2/18

I. Introduction

a. Project Description

The purpose of this project is to apply CPU Scheduling algorithms using Non-Preemptive Priority and First-Come, First-Serve (FCFS) Scheduling, as well as contiguous memory allocation using Best Fit, Worst Fit, and First Fit.

The program was designed and developed using C# to develop the following functions:

1. Display a homepage for users to access various methods
2. Store and display user input values and calculate the turnaround time and burst time according to the CPU Scheduling algorithm selected.
3. Store and display user input values and allocate the process size to the appropriate block size according the main memory management allocation selected.

b. Task Allocation

Tasks allocated to individual members:

Author	Program Segment
Teoh Hong Wei	FCFS GUI Design Integration
Daniel Ng	FCFS Non-preemptive Priority
Amirul Haqim	Best Fit First Fit
James	Best Fit Worst Fit

II. Theoretical Background

a. CPU Scheduling

i. First-Come, First-Serve Algorithm

This algorithm firstly sorts the processes by arrival time,

Afterwards it checks the first process if there is any idle time, moving the time over to the first process arrival time and adds to the idle counter if there are any lapses in between them.

It will then calculate the completion time by adding the burst time,
calculate turnaround time (completion time - arrival time)
and calculate waiting time (turnaround time - burst time)

Repeat the checks and calculation for each and every process till completion.

ii. Non-Preemptive Priority

This algorithm firstly sorts the processes by arrival time,

Afterwards it checks the first process if there is any idle time, moving the time over to the first process arrival time and adds to the idle counter if there are any lapses in between them.

It will then check if the processes have arrived (arrival time < completion time)

If processes have arrived check their priority, if the process below have a higher priority, both of them are swapped and a bool check is triggered to repeat the priority check after the loop is broken by a process that have not arrived, this will be done until the priority checking is done with no swaps.

It will then calculate the completion time by adding the burst time,
calculate turnaround time (completion time - arrival time)
and calculate waiting time (turnaround time - burst time)

Repeat the checks and calculation for each and every process till completion.

b. Main Memory Management

i. Best-Fit Algorithm

Best-Fit allocates the process block size into the memory block size (partitions) which is the smallest among all available partitions. The advantage is that the memory utilization is much better than first fit because it searches for the smallest free partition first available.

ii. Worst-Fit Algorithm

Worst-Fit allocates the process block size into the largest memory block (partitions) among all available partitions. The advantage of using worst-fit is that it reduces the rate of production of small gaps.

iii. First-Fit Algorithm

First-Fit allocates the process block size which is the first sufficient memory block size (partitions) from the top of it. The advantage of it is that it i the fastest search as it searches only the first block according to sequence.

Internal Fragmentation

It arises when the allocations made are only in multiples of a subunit.

External Fragmentation

External fragmentation is the wasted space after allocation to the memory block size(partitions).

III. Software Design

a. GUI Designs

This application is contains three main parts: Homepage, CPU Scheduling and Memory Management.

i. Homepage Design

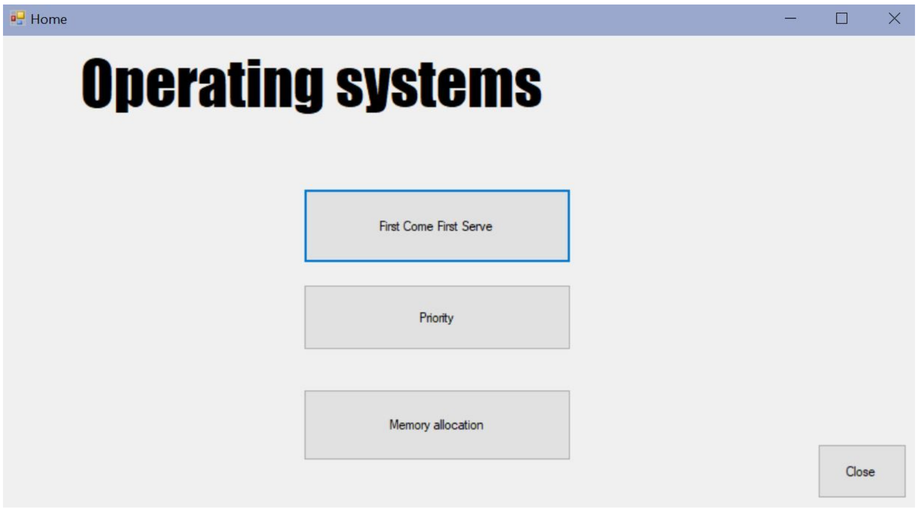
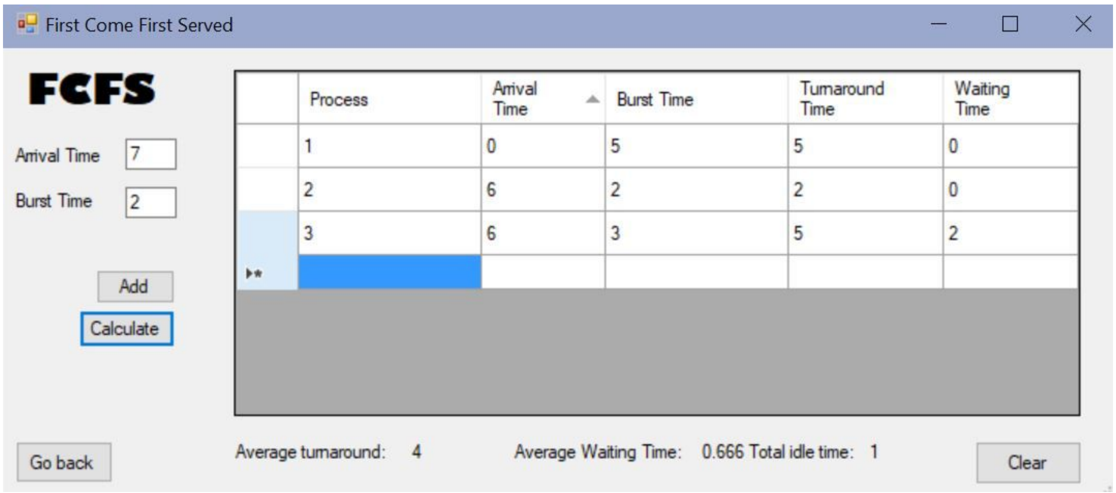


Figure 1: Home Page Design

ii. CPU Scheduling: First-Come, First-Serve Algorithm and Priority Algorithm



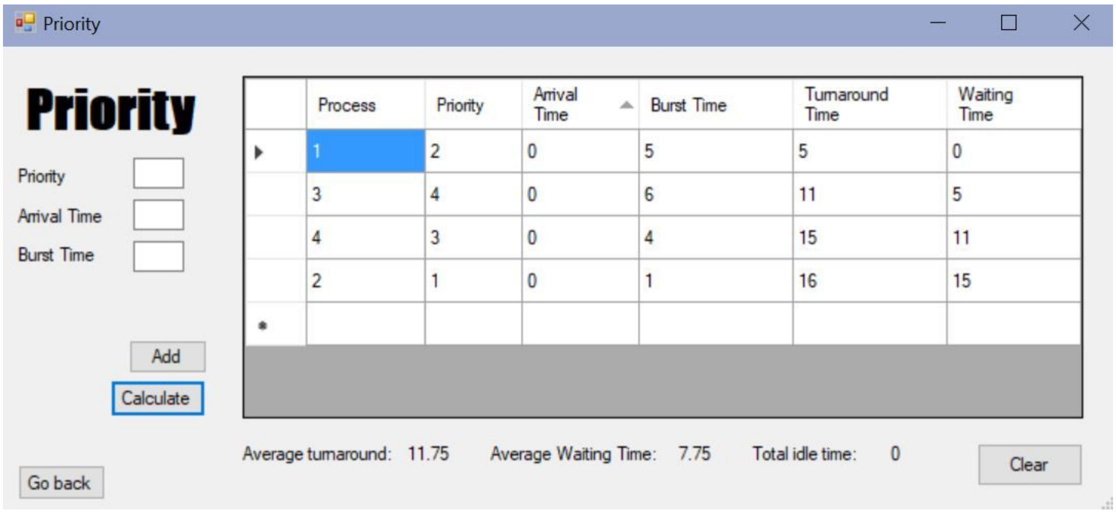
Arrival Time 7

Burst Time 2

Process	Arrival Time	Burst Time	Turnaround Time	Waiting Time
1	0	5	5	0
2	6	2	2	0
3	6	3	5	2

Average turnaround: 4 Average Waiting Time: 0.666 Total idle time: 1

Figure 2: FCFS Algorithm



Priority

Arrival Time

Burst Time

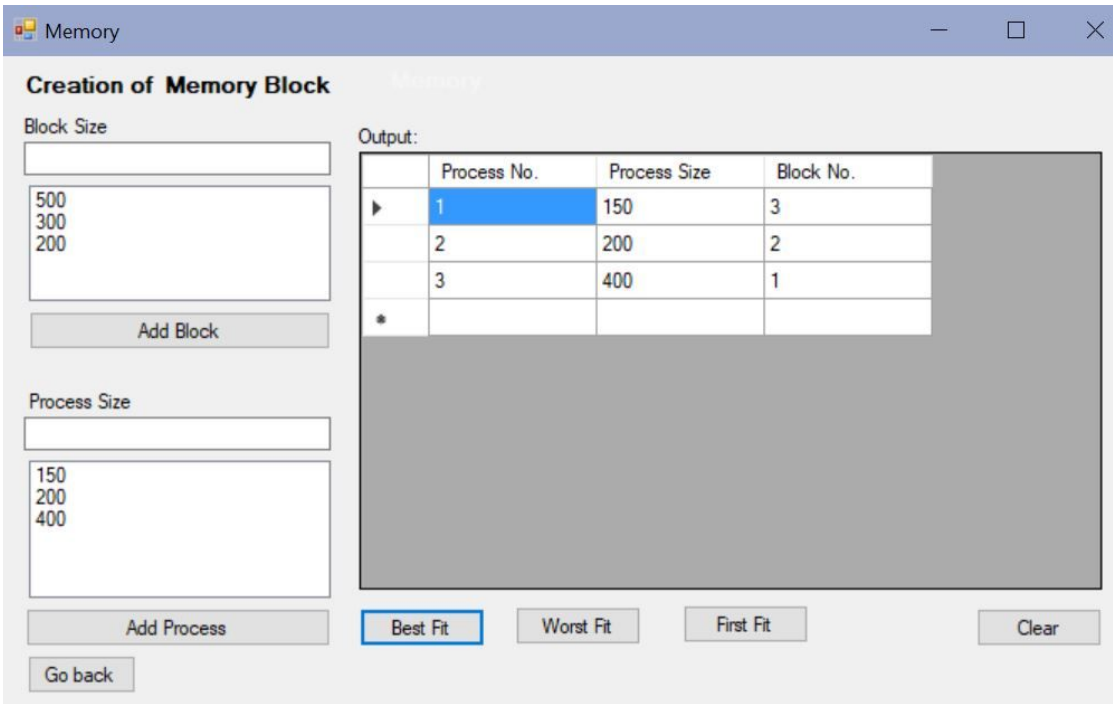
Process	Priority	Arrival Time	Burst Time	Turnaround Time	Waiting Time
1	2	0	5	5	0
3	4	0	6	11	5
4	3	0	4	15	11
2	1	0	1	16	15

Average turnaround: 11.75 Average Waiting Time: 7.75 Total idle time: 0

Figure 3: Priority Algorithm

This application requires user inputs, which the application will make use of user declared variables to calculate turnaround/waiting time, average turnaround/waiting time and idle time. The user is also given the option to clear all processes.

Add: User is required to fill in the blanks before adding the
process to the DataGridView
Calculate: Calculate the times based on user inputs and selected algorithm
Clear: Clears all processes.
Go back: Returns the user to the home page.



Block Size

500
300
200

Process Size

150
200
400

Process No.	Process Size	Block No.
1	150	3
2	200	2
3	400	1

Figure 4: Memory Allocation Algorithm

This application allows users to allocate the process size into appropriate block size according to the algorithm selected.

The user inserts the value in the textbox provided. Clicking “Add Block” would insert the value into the listbox and stores the value into an array. The “Block Size” textbox is also cleared to allow the user to insert more values. A similar response is expected when clicking the button “Add Process”, storing the value in the process listbox and in a different array.

Clicking the “Best Fit” button would sort the process in the smallest sufficient partition among the available partitions. The results would then be displayed in a dataGridview.

Clicking the “Worst Fit” button would allocate the process in the largest sufficient among all the available partitions. The results would then be displayed in a dataGridview.

Clicking the “First Fit” button would allocate the process which is first sufficient from the main memory. The results would then be displayed in a listbox.

Clicking the “Clear” button would clear the arrays, lists, and the data displayed in the listbox and dataGridview.

Lastly, clicking the “Go Back” button would bring the user back to the homepage to access the other algorithms available.

b. Flowchart Diagrams

i. CPU Scheduling

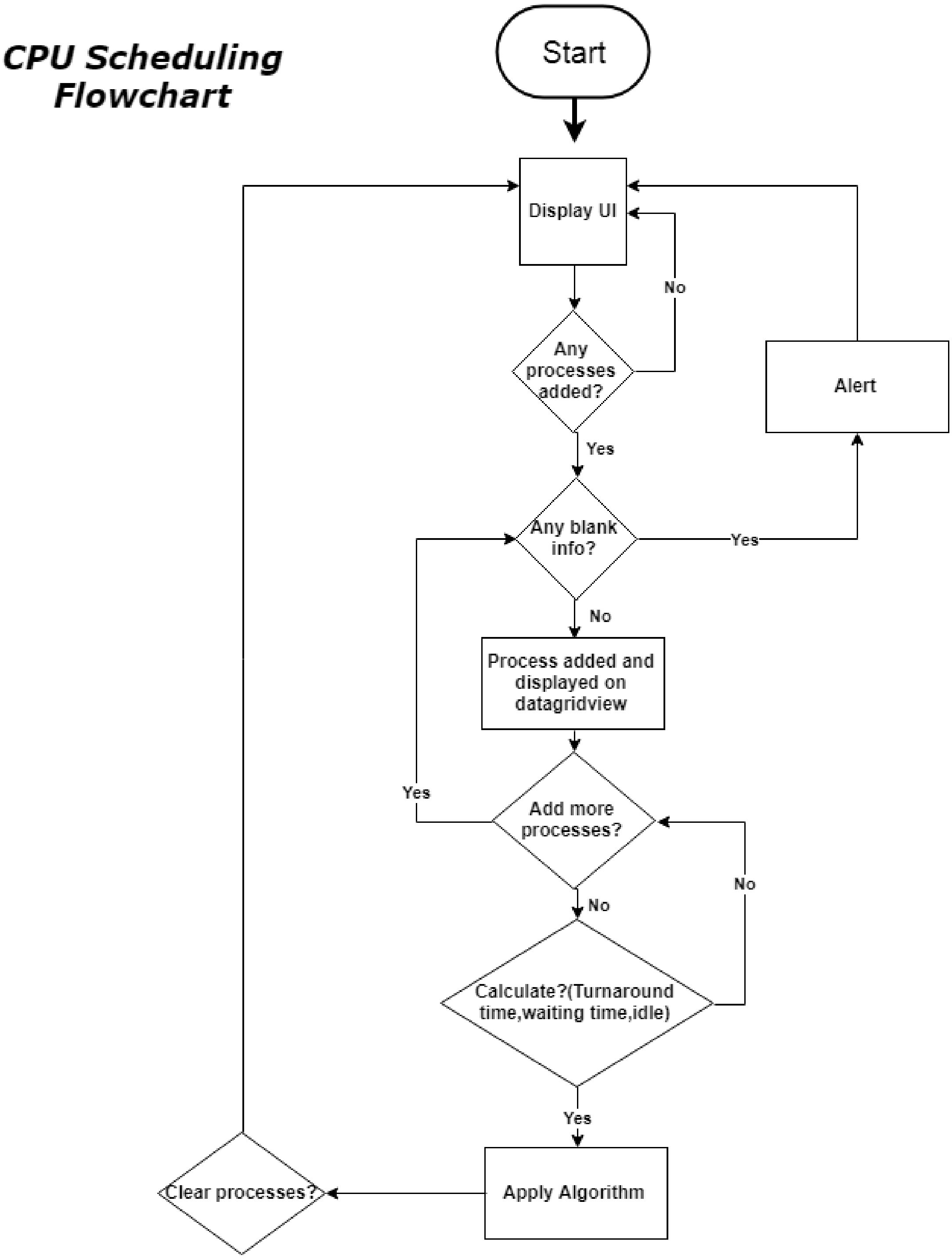
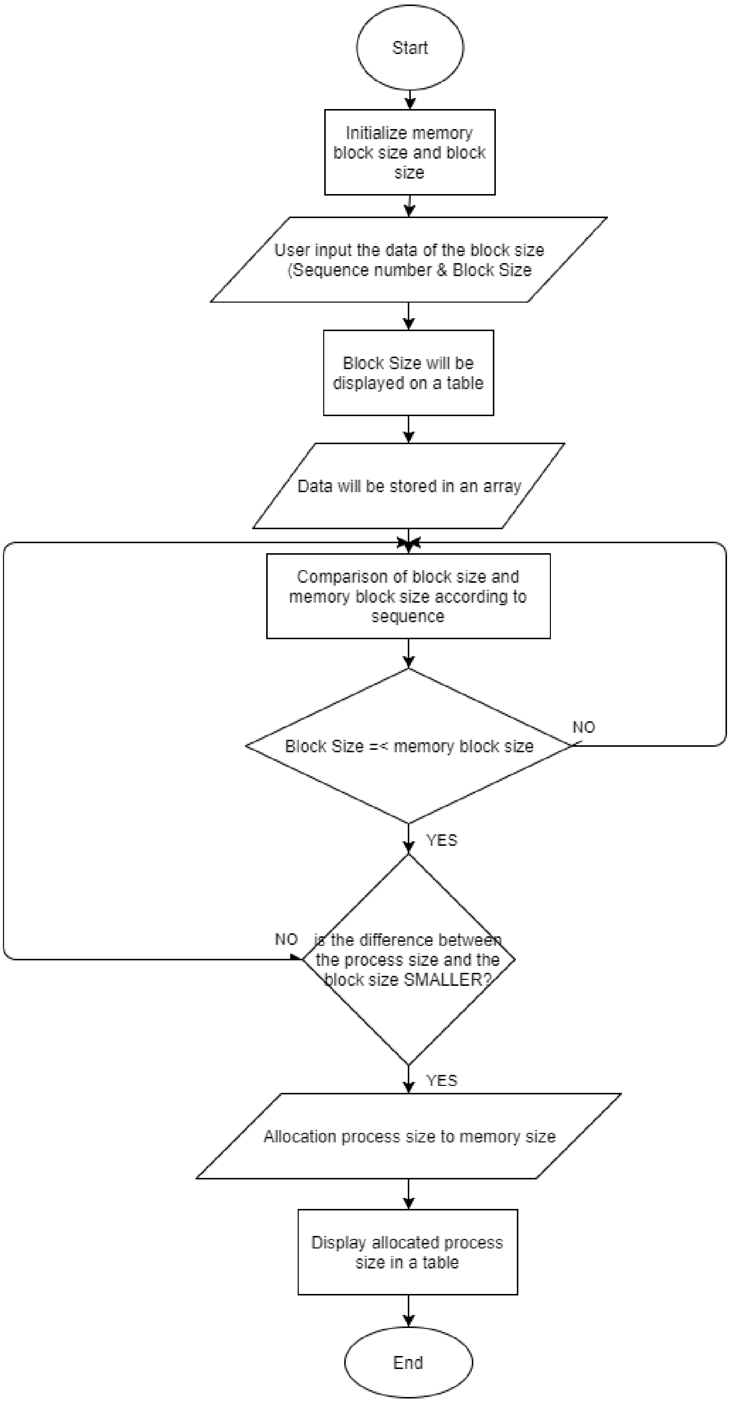


Figure 5: CPU Scheduling Flowchart

ii. Main Memory Management



Start
Initialize memory block size and block size
User input the data of the block size (Sequence number & Block Size
Block Size will be displayed on a table
Data will be stored in an array
Comparison of block size and memory block size according to sequence
Block Size =< memory block size
NO
YES
NO
is the difference between the process size and the block size SMALLER?
YES
Allocation process size to memory size
Display allocated process size in a table
End

Figure 6: Best-Fit Algorithm Flowchart

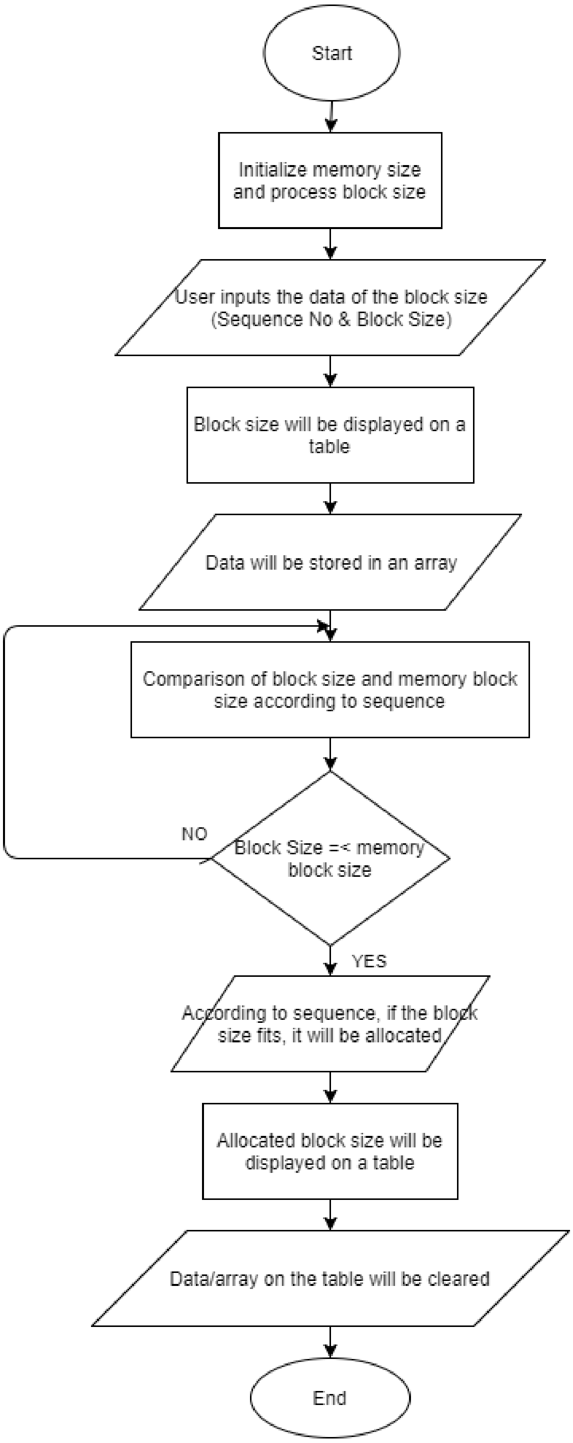

Start
Initialize memory size and process block size
User inputs the data of the block size (Sequence No & Block Size)
Block size will be displayed on a table
Data will be stored in an array
Comparison of block size and memory block size according to sequence
NO
Block Size =< memory block size
YES
According to sequence, if the block size fits, it will be allocated
Allocated block size will be displayed on a table
Data/array on the table will be cleared
End

Figure 7: First-Fit Algorithm

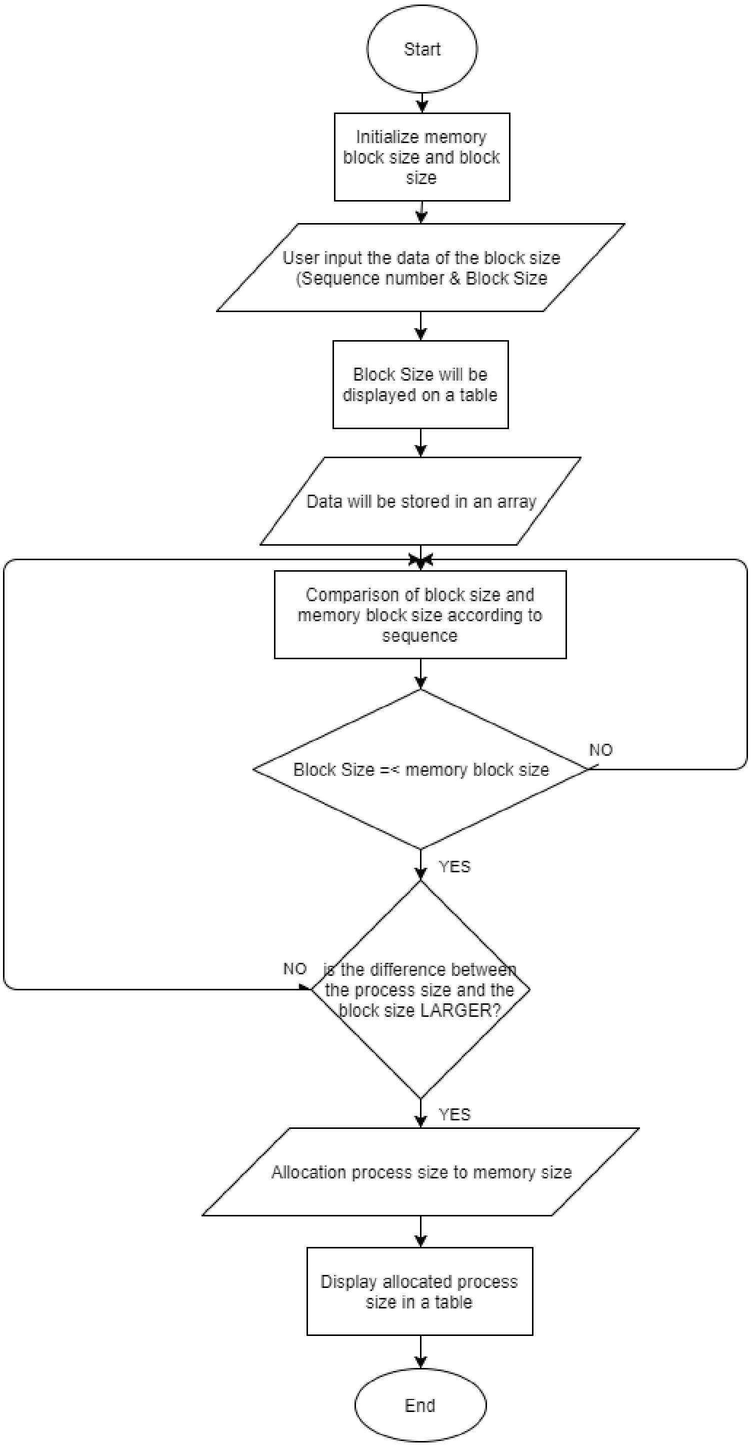

Start
Initialize memory block size and block size
User input the data of the block size (Sequence number & Block Size
Block Size will be displayed on a table
Data will be stored in an array
Comparison of block size and memory block size according to sequence
Block Size =< memory block size
NO
YES
is the difference between the process size and the block size LARGER?
NO
YES
Allocation process size to memory size
Display allocated process size in a table
End

Figure 8: Worst-Fit Algorithm

IV. Conclusion

In conclusion, our group has chosen to use Windows Form Application, and C# to create this application.

We have learnt about the usage of DataGridView and how Listbox can be used to display simulated operating systems algorithms through user inputs.

One of the limitations faced by our group was trying to implement a gantt chart for our CPU Scheduling, as we weren’t aware of how charts can be used to display values until the application was completed.

Overall, our group has learnt much while creating simulated operating systems algorithms.