King Saud University

College of Computer and Information Sciences

Computer Science Department

CSC227: Operating Systems

Course Project-S2-1445

Group Members

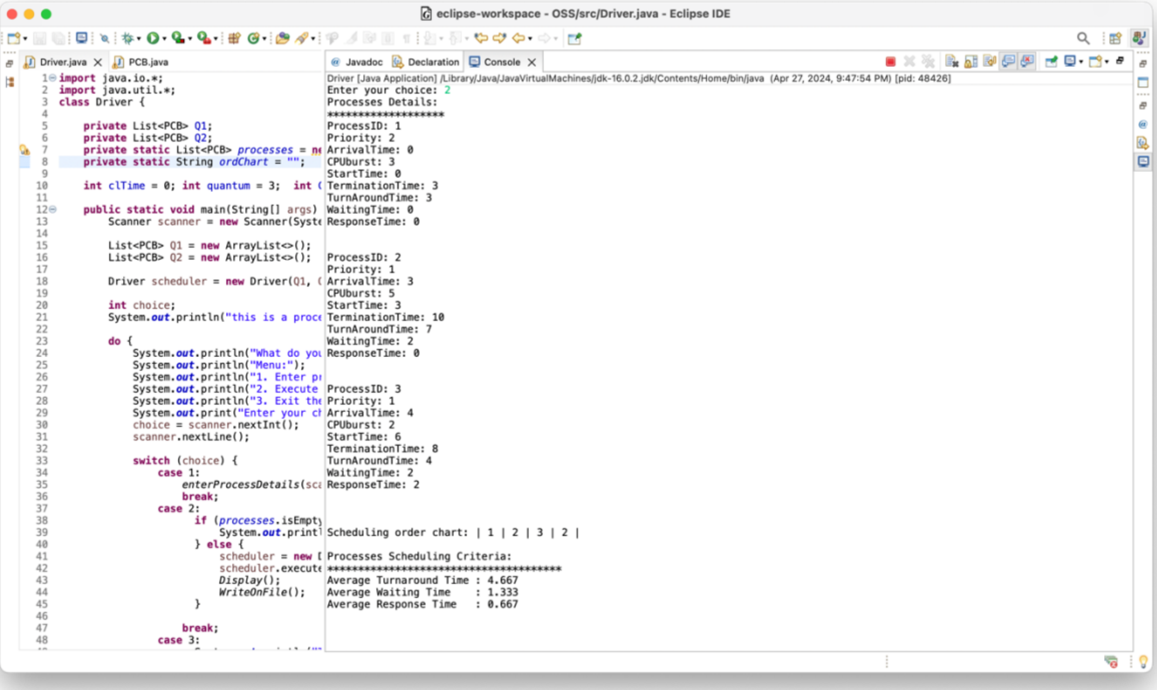
Name	ID
Riham Alangari	443200879
Nouv Al-Qahtani “Leader”	442201905
AlJouri AlSarami	442202991

Instructor: Abeer Al-Shaya

Section: 69232

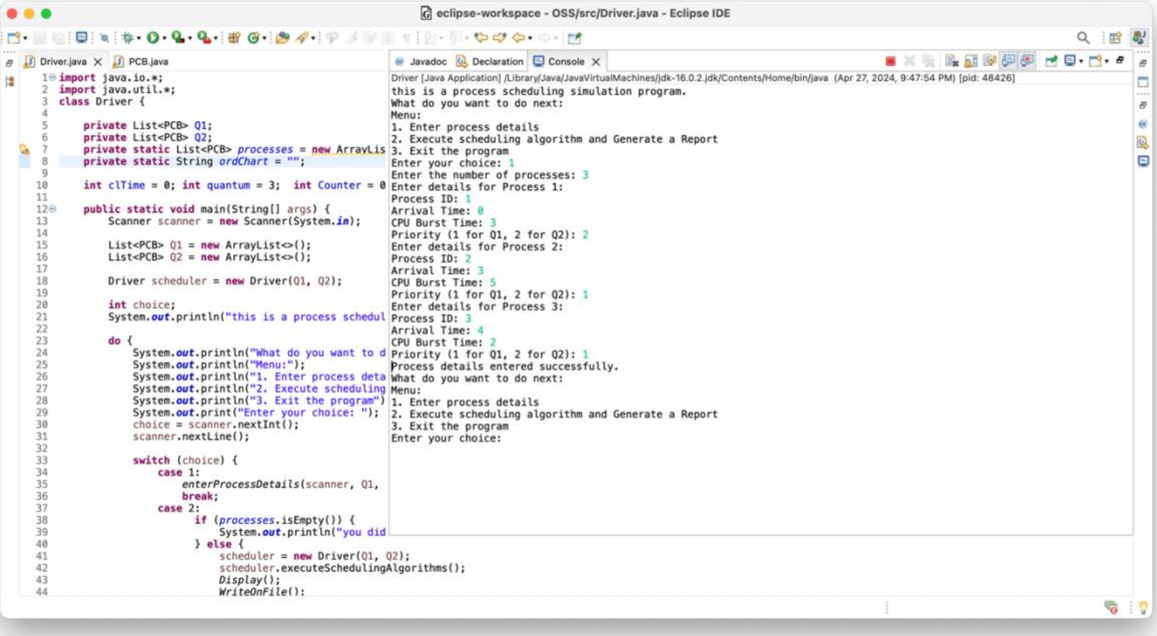
NAME	WORK DISTRIBUTION
Riham Alangari	Worked on: -PCB.Java -README File
Nouv Al-Qahtani	Worked on: -The methods related to Scheduling algorithms -Display Method & Writing on the File
AlJouri AlSarami	Worked on: -Shortest Job First method -Menu creation -Test cases

Sample input/output:



```java
import java.io.*;
import java.util.*;
class Driver {

    private List<PCB> Q1;
    private List<PCB> Q2;
    private static List<PCB> processes = ne
    private static String ordChart = "";

    int clTime = 0; int quantum = 3;  int (

    public static void main(String[] args)
        Scanner scanner = new Scanner(Syste

        List<PCB> Q1 = new ArrayList<>();
        List<PCB> Q2 = new ArrayList<>();

        Driver scheduler = new Driver(Q1, (

        int choice;
        System.out.println("this is a proce

        do {
            System.out.println("What do you
            System.out.println("Menu:");
            System.out.println("1. Enter pr
            System.out.println("2. Execute 
            System.out.println("3. Exit the
            System.out.print("Enter your ch
            choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    enterProcessDetails(sca
                    break;
                case 2:
                    if (processes.isEmpty
                        System.out.printl
                    } else {
                        scheduler = new [
                        scheduler.execute
                        Display();
                        WriteOnFile();
                    }

                    break;
                case 3:
```

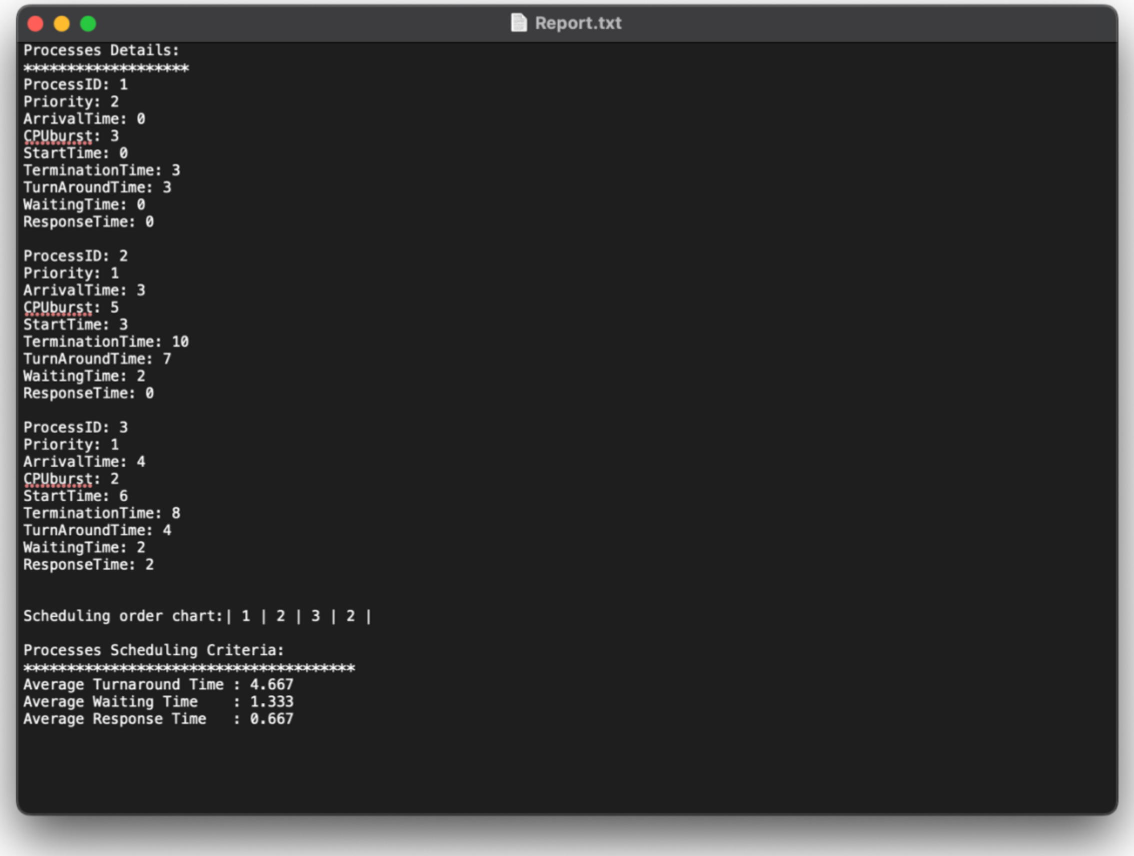
```
Driver [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java  (Apr 27, 2024, 9:47:54 PM) [pid: 48426]
Enter your choice: 2
Processes Details:
*******************
ProcessID: 1
Priority: 2
ArrivalTime: 0
CPUburst: 3
StartTime: 0
TerminationTime: 3
TurnAroundTime: 3
WaitingTime: 0
ResponseTime: 0


ProcessID: 2
Priority: 1
ArrivalTime: 3
CPUburst: 5
StartTime: 3
TerminationTime: 10
TurnAroundTime: 7
WaitingTime: 2
ResponseTime: 0


ProcessID: 3
Priority: 1
ArrivalTime: 4
CPUburst: 2
StartTime: 6
TerminationTime: 8
TurnAroundTime: 4
WaitingTime: 2
ResponseTime: 2



Scheduling order chart: | 1 | 2 | 3 | 2 |

Processes Scheduling Criteria:
***************************************
Average Turnaround Time : 4.667
Average Waiting Time    : 1.333
Average Response Time   : 0.667
```



```java
import java.io.*;
import java.util.*;
class Driver {

    private List<PCB> Q1;
    private List<PCB> Q2;
    private static List<PCB> processes = new ArrayLis
    private static String ordChart = "";

    int clTime = 0; int quantum = 3;  int Counter = 0

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        List<PCB> Q1 = new ArrayList<>();
        List<PCB> Q2 = new ArrayList<>();

        Driver scheduler = new Driver(Q1, Q2);

        int choice;
        System.out.println("this is a process schedul

        do {
            System.out.println("What do you want to d
            System.out.println("Menu:");
            System.out.println("1. Enter process deta
            System.out.println("2. Execute scheduling
            System.out.println("3. Exit the program")
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    enterProcessDetails(scanner, Q1,
                    break;
                case 2:
                    if (processes.isEmpty()) {
                        System.out.println("you did
                    } else {
                        scheduler = new Driver(Q1, Q2);
                        scheduler.executeSchedulingAlgorithms();
                        Display();
                        WriteOnFile();
```

```
Driver [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java  (Apr 27, 2024, 9:47:54 PM) [pid: 48426]
this is a process scheduling simulation program.
What do you want to do next:
Menu:
1. Enter process details
2. Execute scheduling algorithm and Generate a Report
3. Exit the program
Enter your choice: 1
Enter the number of processes: 3
Enter details for Process 1:
Process ID: 1
Arrival Time: 0
CPU Burst Time: 3
Priority (1 for Q1, 2 for Q2): 2
Enter details for Process 2:
Process ID: 2
Arrival Time: 3
CPU Burst Time: 5
Priority (1 for Q1, 2 for Q2): 1
Enter details for Process 3:
Process ID: 3
Arrival Time: 4
CPU Burst Time: 2
Priority (1 for Q1, 2 for Q2): 1
Process details entered successfully.
What do you want to do next:
Menu:
1. Enter process details
2. Execute scheduling algorithm and Generate a Report
3. Exit the program
Enter your choice:
```

Report:



```
Processes Details:
*******************
ProcessID: 1
Priority: 2
ArrivalTime: 0
CPUburst: 3
StartTime: 0
TerminationTime: 3
TurnAroundTime: 3
WaitingTime: 0
ResponseTime: 0

ProcessID: 2
Priority: 1
ArrivalTime: 3
CPUburst: 5
StartTime: 3
TerminationTime: 10
TurnAroundTime: 7
WaitingTime: 2
ResponseTime: 0

ProcessID: 3
Priority: 1
ArrivalTime: 4
CPUburst: 2
StartTime: 6
TerminationTime: 8
TurnAroundTime: 4
WaitingTime: 2
ResponseTime: 2


Scheduling order chart:| 1 | 2 | 3 | 2 |

Processes Scheduling Criteria:
**************************************
Average Turnaround Time : 4.667
Average Waiting Time    : 1.333
Average Response Time   : 0.667
```

Criteria	Riham	Jouri	Nouv
Work division: Contributed equally to the work	1	1	1
Peer evaluation: Level of commitments (Interactivity with other team members), and professional behavior towards team & TA	1	1	1
Project Discussion: Accurate answers, understanding of the presented work, good listeners to questions			
Time management: Attending on time, being ready to start the demo, good time management in discussion and demo.			
Total			