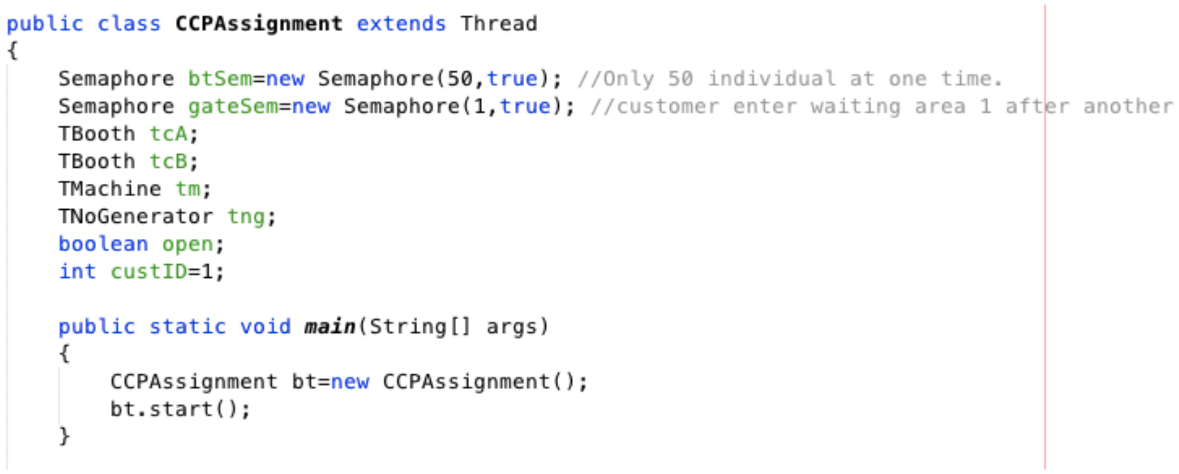
Bus Ticketing Service System
(Concurrent Programming)

Developed a concurrent Java application to model bus terminal operations, ensuring efficient ticketing and boarding processes. Implemented multithreading and synchronization techniques to enhance performance and user experience

TABLE OF CONTENTS

1. Java concurrent concepts

1.1. Thread



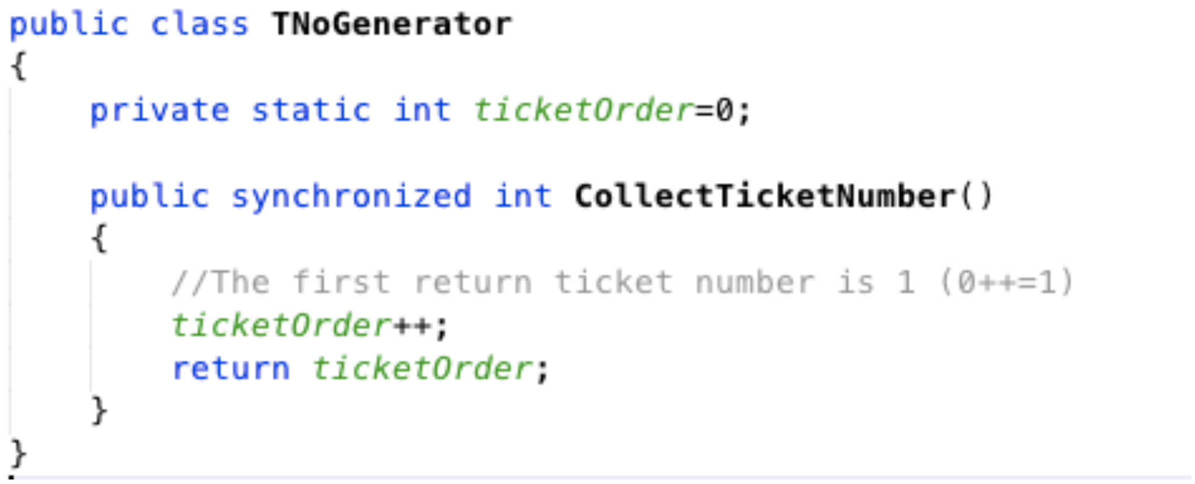
```
public class CCPAssignment extends Thread
{
    Semaphore btSem=new Semaphore(50,true); //Only 50 individual at one time.
    Semaphore gateSem=new Semaphore(1,true); //customer enter waiting area 1 after another
    TBooth tcA;
    TBooth tcB;
    TMachine tm;
    TNoGenerator tng;
    boolean open;
    int custID=1;

    public static void main(String[] args)
    {
        CCPAssignment bt=new CCPAssignment();
        bt.start();
    }
```

Figure 1. Code Screenshot for Thread.

Thread allows a programme to work efficiently by permitting it to do multiple things at once. Threads can be used to carry out complicated tasks behind the scenes without interfering with the programme. Each resource/entity in this code, for example, is a thread.

In this code, Each customer as well as each service in the terminal is a thread. They execute the code simultaneously and independently. They even share resources like scanner, etc.

1.2. Synchronization



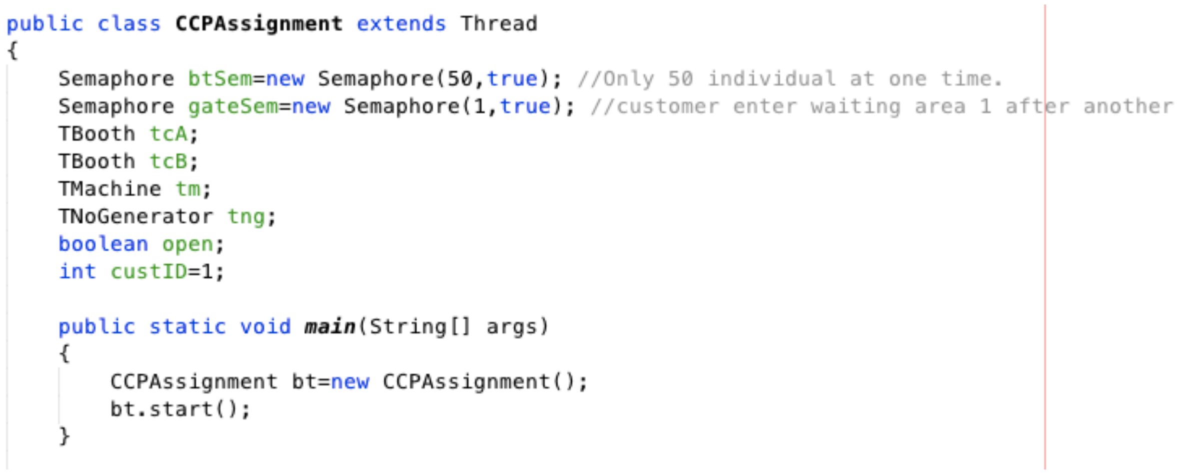
```
public class TNoGenerator
{
    private static int ticketOrder=0;

    public synchronized int CollectTicketNumber()
    {
        //The first return ticket number is 1 (0++=1)
        ticketOrder++;
        return ticketOrder;
    }
}
```

Figure 2. Code Screenshot for Synchronization.

The ability to regulate the entry of multiple strings to any common asset is known as synchronisation in the concurrency notion in Java. When only one thread needs to access a shared resource, Java Synchronization is a better option.

In this code, Synchronization is used as all customers entering the APBT bus terminal will require a ticket, we are currently prepared to only allow one conjuring of the CollectTicketNumber() strategy. However, string obstruction or dispute may occur, causing the strings to run slower as they cling to one another, potentially resulting in suspension.

1.3. Semaphores



```
public class CCPAssignment extends Thread
{
    Semaphore btSem=new Semaphore(50,true); //Only 50 individual at one time.
    Semaphore gateSem=new Semaphore(1,true); //customer enter waiting area 1 after another
    TBooth tcA;
    TBooth tcB;
    TMachine tm;
    TNoGenerator tng;
    boolean open;
    int custID=1;

    public static void main(String[] args)
    {
        CCPAssignment bt=new CCPAssignment();
        bt.start();
    }
```

Figure 3. Code Screenshot for Semaphores.

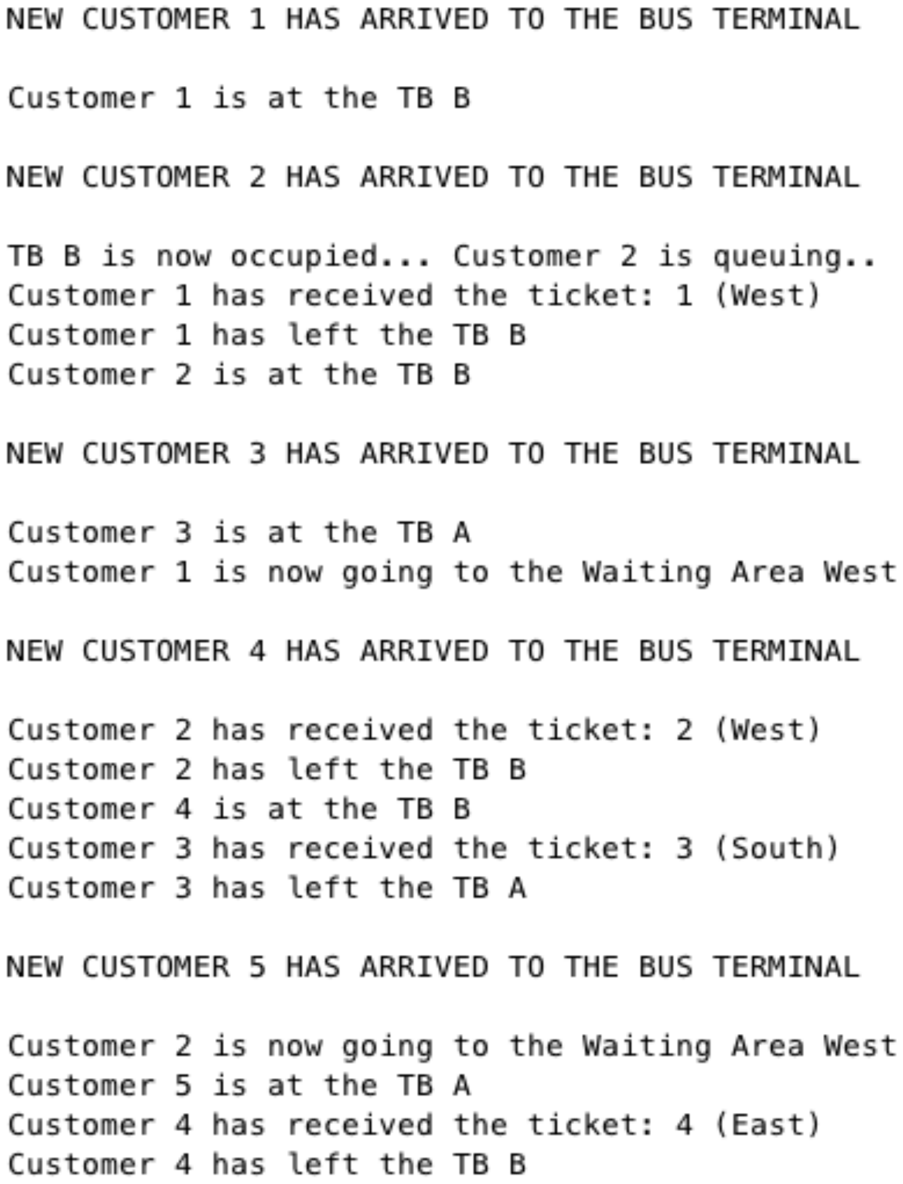
A Semaphore is a thread synchronisation mechanism that allows you to send signals between threads to avoid missing signals or to protect a critical part.

Through a counter, semaphore is used to manage access to a shared resource.

In this code, Semaphore is used to so that only limited amount of people can enter the building, in the code 50 individual are allowed in the building at one time.

2. Code Requirements

2.1. Customers entering the Bus Terminal

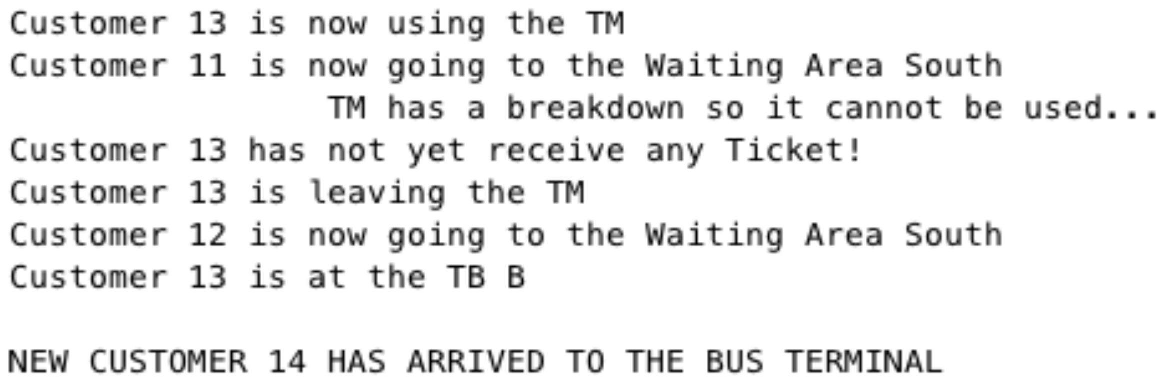


```
NEW CUSTOMER 1 HAS ARRIVED TO THE BUS TERMINAL

Customer 1 is at the TB B

NEW CUSTOMER 2 HAS ARRIVED TO THE BUS TERMINAL

TB B is now occupied... Customer 2 is queuing..
Customer 1 has received the ticket: 1 (West)
Customer 1 has left the TB B
Customer 2 is at the TB B

NEW CUSTOMER 3 HAS ARRIVED TO THE BUS TERMINAL

Customer 3 is at the TB A
Customer 1 is now going to the Waiting Area West

NEW CUSTOMER 4 HAS ARRIVED TO THE BUS TERMINAL

Customer 2 has received the ticket: 2 (West)
Customer 2 has left the TB B
Customer 4 is at the TB B
Customer 3 has received the ticket: 3 (South)
Customer 3 has left the TB A

NEW CUSTOMER 5 HAS ARRIVED TO THE BUS TERMINAL

Customer 2 is now going to the Waiting Area West
Customer 5 is at the TB A
Customer 4 has received the ticket: 4 (East)
Customer 4 has left the TB B
```

Figure 4. Output screenshot of Customer entering the bus terminal.

In the above screenshot, we can see that the Customer 1 arrives at the bus terminal, then the customer 1 goes to ticket booth B. Then the customer 1 receives a ticket with ticket number and waiting area assigned on it and then the customer waits at the assigned waiting area (West) till the bus arrives and other procedure are in process.

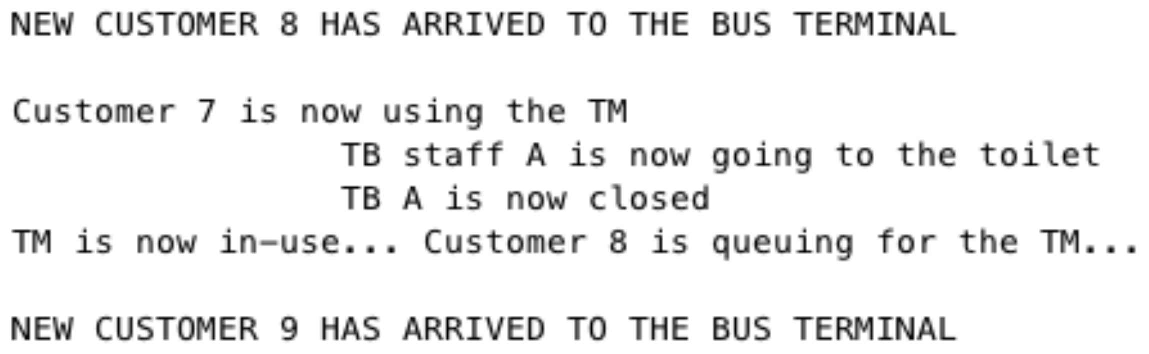
2.2. Breakdown of Ticket Machine



```
Customer 13 is now using the TM
Customer 11 is now going to the Waiting Area South
                TM has a breakdown so it cannot be used...
Customer 13 has not yet receive any Ticket!
Customer 13 is leaving the TM
Customer 12 is now going to the Waiting Area South
Customer 13 is at the TB B

NEW CUSTOMER 14 HAS ARRIVED TO THE BUS TERMINAL
```

Figure 5. Output screenshot of Ticket Machine Breakdown.

In the above screenshot, we can see that when the customer goes to use the ticket machine the Ticket Machine has a breakdown and it can not be used. So the customer doesn't receive any ticket and leaves from the Ticket Machine area and goes to the Ticket Booth to collect the ticket.

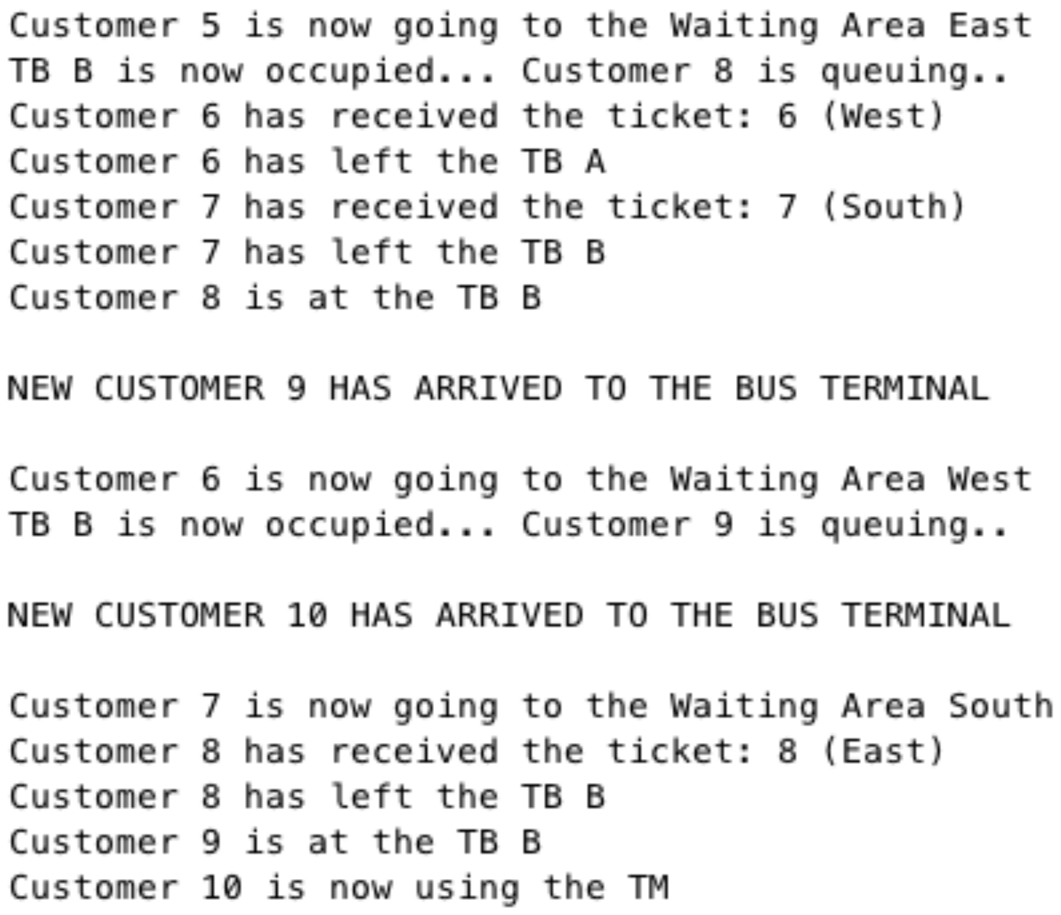


```
NEW CUSTOMER 8 HAS ARRIVED TO THE BUS TERMINAL

Customer 7 is now using the TM
                TB staff A is now going to the toilet
                TB A is now closed
TM is now in-use... Customer 8 is queuing for the TM...

NEW CUSTOMER 9 HAS ARRIVED TO THE BUS TERMINAL
```

Figure 6. Output screenshot of Ticket Machine Repaired.

In the above screenshot, we can see that the Ticket Machine is now repaired and is ready for customers to use it again.

2.3. Waiting Area

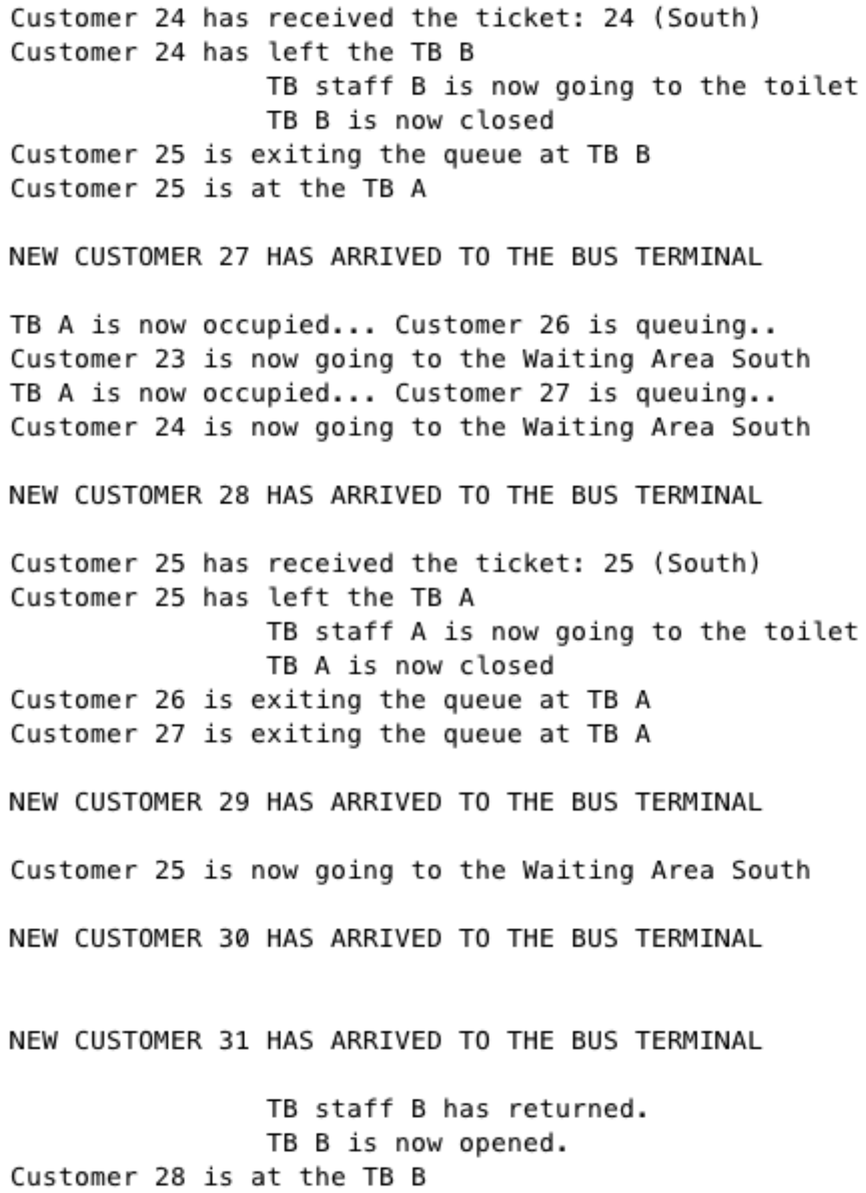


```
Customer 5 is now going to the Waiting Area East
TB B is now occupied... Customer 8 is queuing..
Customer 6 has received the ticket: 6 (West)
Customer 6 has left the TB A
Customer 7 has received the ticket: 7 (South)
Customer 7 has left the TB B
Customer 8 is at the TB B

NEW CUSTOMER 9 HAS ARRIVED TO THE BUS TERMINAL

Customer 6 is now going to the Waiting Area West
TB B is now occupied... Customer 9 is queuing..

NEW CUSTOMER 10 HAS ARRIVED TO THE BUS TERMINAL

Customer 7 is now going to the Waiting Area South
Customer 8 has received the ticket: 8 (East)
Customer 8 has left the TB B
Customer 9 is at the TB B
Customer 10 is now using the TM
```

Figure 7. Output screenshot of Waiting Area.

In the above screenshot, we can see that there are three waiting areas. Once the customer receives the ticket from the Ticket Machine or Ticket Booth, the customer has a assigned waiting area on their ticket according to the destination they go and bus the they are going to board. There are three different waiting areas West, South and East.

2.4. Staff Break

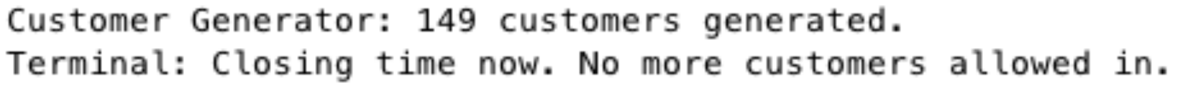


```
Customer 24 has received the ticket: 24 (South)
Customer 24 has left the TB B
                TB staff B is now going to the toilet
                TB B is now closed
Customer 25 is exiting the queue at TB B
Customer 25 is at the TB A

NEW CUSTOMER 27 HAS ARRIVED TO THE BUS TERMINAL

TB A is now occupied... Customer 26 is queuing..
Customer 23 is now going to the Waiting Area South
TB A is now occupied... Customer 27 is queuing..
Customer 24 is now going to the Waiting Area South

NEW CUSTOMER 28 HAS ARRIVED TO THE BUS TERMINAL

Customer 25 has received the ticket: 25 (South)
Customer 25 has left the TB A
                TB staff A is now going to the toilet
                TB A is now closed
Customer 26 is exiting the queue at TB A
Customer 27 is exiting the queue at TB A

NEW CUSTOMER 29 HAS ARRIVED TO THE BUS TERMINAL

Customer 25 is now going to the Waiting Area South

NEW CUSTOMER 30 HAS ARRIVED TO THE BUS TERMINAL


NEW CUSTOMER 31 HAS ARRIVED TO THE BUS TERMINAL

                TB staff B has returned.
                TB B is now opened.
Customer 28 is at the TB B
```

Figure 8. Output screenshot of Staff Break.

In the above screenshot, we can see that the staff from Ticket Booth B has gone for a toilet break so the Ticket Booth is closed and customers can either use Ticket Booth A or Ticket Machine to get the ticket. We can also see that the staff from Ticket Booth A has also gone for a toilet break and the staff from Ticket Booth B is still on break. At this time the customer can use the Ticket Machine to get the tickets. In the above screenshot, we can also see that the staff from Ticket Booth B has returned from the toilet break and Ticket Booth B is again open and customer can purchase tickets from there as well.

2.5. Terminal Capacity



```
Customer Generator: 149 customers generated.
Terminal: Closing time now. No more customers allowed in.
```

Figure 9. Output screenshot of Terminal Capacity.

In the above screenshot, we can see that the customer entry is blocked and no more customers are allowed inside as the terminal has reached max capacity.

3. References

eG Innovations. 2021. *Thread in Java - What is it and Why it is Used | eG Innovations*. [online] Available at: <https://www.eginnovations.com/blog/java-threads/> [Accessed 26 May 2021].

www.javatpoint.com. 2021. *Synchronization in Java - javatpoint*. [online] Available at: <https://www.javatpoint.com/synchronization-in-java> [Accessed 27 May 2021].

Sidana, U., 2021. *Semaphore in Java | How to use Semaphore with Example | Edureka*. [online] Edureka. Available at: <https://www.edureka.co/blog/semaphore-in-java/> [Accessed 28 May 2021].