***Sleeping Barber Shop Problem***

Starting with the problem definition:

The problem is based on a hypothetical barbershop with a certain number of barbers. When there are no customers, the barber sleeps on his chair. If any customer enters, he should wake up the barber to get his hair cut. If there are no chairs empty for another customers, they have to wait in the waiting room/chairs.

The solution pseudocode:

* If there is no customer to be served, the barber goes to sleep
* The barber waits until a customer enters the shop
* If the customer enters the barbershop, and the barber is sleeping, the customer wakes up the barber
* The barber receives a signal to start
* The customer is being served and getting his hair cut
* If the customer enters the barbershop, and the barber is busy, the customer had to wait in the waiting room till a chair is available then he sits and being served
* Accept the customer when the limit is reached
* If a customer enters the shop and there are no available chairs, the customer leaves (Additional case for illustration the problem in the real world)

**\*Examples on deadlock in this problem:**

**Deadlock definition:**

Deadlock is a situation when two threads are waiting for each other and the waiting never ends. Here both threads can’t complete their tasks.

***In this problem*, the deadlock occurs if the customer ends up waiting for the barber and the barber ends up waiting for the customer to arrive.**

**How to solve the deadlock here?**

To handle this problem code should use reentrant locks and after a thread acquires a lock it sleeps for few milliseconds and then release the lock.

A screenshot of a computer

Description automatically generated with medium confidence

\*Examples on starvation in this problem:

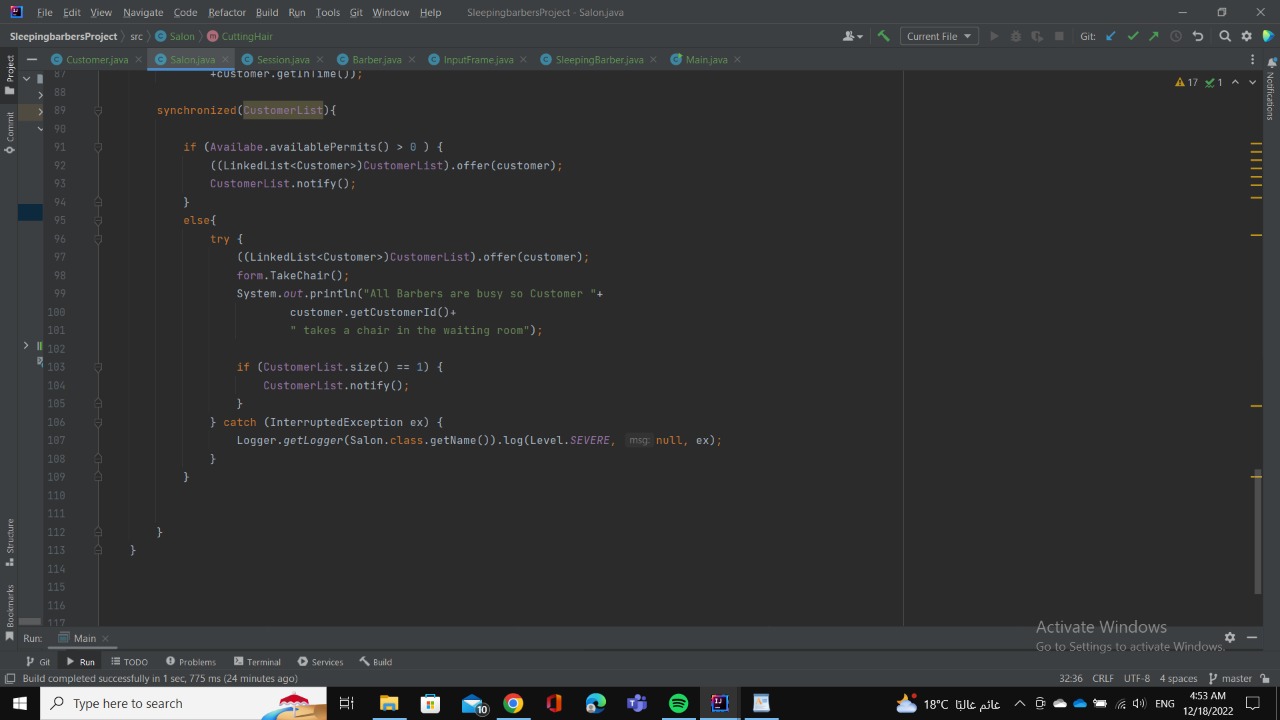
**Definition of starvation:**

Starvation is a situation when a thread is in waiting state from long period because it not getting access of shared resources or because higher priority threads are coming.

***In this problem*, starvation will occur if the customers don’t follow any order for getting a haircut, as some won’t get a haircut even though even after waiting for a long time.**

**How to handle the starvation in this problem?**

To handle this problem in the code, insert the customers in a linked list which follows the first in first out property. So, every time a customer sits in a waiting room, they will be selected by the barber in first come first serve basis



**Explanation for real world application and how did apply the problem:**

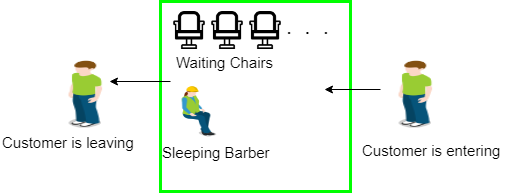
This design is the best analogy for a customer care call center. Initially when there is no customer on-call all call-executives just relax and wait for the call. The moment the first customer dials the number he/she is connected to any call-executive and in a scenario when all call-executives are busy the customer will have to wait in a queue till they are assigned to a call-executive. If all executives are busy and the waiting line is full, the customers are disconnected with a message that executives are busy and customers will be contacted later by the company. This best relates to this design as the customers are picked from the queue in a first come first serve basis and call-executives are utilized in such a way that everyone executive gets at least one call.

***In this scenario we can have the following design similarities:***

1. The critical section will be the call between executive and customer

2. The waiting room will be the waiting queue over a call, where customers will be held in a FIFO manner.

3. Locks can be acquired on the waiting queue so that no two executives pick the same customer.



Diagram

Description automatically generated

Shape

Description automatically generated with medium confidence

***Team members***

1. **Salah El-deen Kamel Mahmoud**
2. **Diaa Hany Ahmed Abdulmuniem**
3. **Abdelrahman Mohamed Youssef**
4. **Fatma Saed Foad Mohamed**
5. **Mariam Amr Fawzy Mohamed**
6. **Mariam Mohamed Moawad**
7. **Hisham Mohamed Ahmed**

***Thank you***