MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Lab Report

Lab Report No: 03

Lab Report on: Multithreading-based project: Car Parking Management System

Course Title: Software Engineering and Project Management Lab

Course Code: ICT-3108

Submitted By	Submitted To
Name: Farhad Ali ID: IT-23035 3rd Year, 1st Semester Session: 2022-2023 Dept. of ICT, MBSTU	Dr. Ziaur Rahman Associate Professor Dept of ICT, MBSTU

Date of Performance: Date of Submission:

Lab Report 3:

In a real world parking system, multiple cars may arrive at the same time requesting parking, while multiple parking agents handle these requests concurrently. This scenario is a classic example of producer consumer problem in multithreading.

(1) Cars act as producers
(2) Parking Agents act as consumers
(3) A shared parking pool holds pending parking requests

Code:

RegistrationParking.java:

```
class RegistrarParking {
    Private String carNumber;
    public RegistrarParking(String
carNumber) { this.carNumber = carNumber
}
```

```
Public String getCarNumber(){
        return carNumber;
}
```

ParkingPool.java:

```
import java.util.LinkedList;
import java.util.Queue;
class ParkingPool {
  private Queue<RegistrarParking> queue =
                new LinkedList<>();

Public synchronized void addCar(RegistrarParking car)

{ queue.add(car);
  System.out.println("Car" + car.getCarNumber
  () + " requested parking.")
  notify();

}
```

ParkingAgent.java:

```
class ParkingAgent extends Thread {
    private ParkingPool pool;
    private int agentId;
    public ParkingAgent(ParkingPool pool, int
                        agentId) {
        this.pool = pool;
        this.agentId = agentId;
    }
```

MainClass.java:

```
public class MainClass {
    public static void main(String[] args) {
        ParkingPool pool = new ParkingPool();
        ParkingAgent agent1 = new ParkingAgent(pool, 1);
        ParkingAgent agent2 = new ParkingAgent
                              (pool, 2);
        agent1.start();
        agent2.start();
    }
}
```