Experiment 4.3

Student Name: Muskan Pandey **UID: 22BCS12593**
Branch: CSE **Section/Group: 22BCS_IOT-618/B**
Semester: 6th **Date: 21-02-25**
Subject: Java **Subject Code: 22CSH-359**

1.Aim: Develop a ticket booking system with synchronized threads to ensure no double booking of seats. Use thread priorities to simulate VIP bookings being processed first.

2. Objective: The objective of this program is to develop a ticket booking system using synchronized threads to ensure no double booking of seats. Additionally, thread priorities are used to prioritize VIP bookingsAdd cards to the collection.

Search for all cards of a given symbol (e.g., Hearts, Diamonds).

Display all stored cards.

3. Code:

```
import java.util.*;

class TicketBookingSystem {
   private static final int TOTAL_SEATS = 5; // Set total available seats
   private final boolean[] seats = new boolean[TOTAL_SEATS]; // false means
available, true means booked

   // Synchronized method to ensure thread safety
   public synchronized boolean bookSeat(int seatNumber, String userType) {
      if (seatNumber < 0 || seatNumber >= TOTAL_SEATS) {
         System.out.println(userType + " tried to book an invalid seat: " + seatNumber);
         return false;
      }

      if (!seats[seatNumber]) { // If the seat is available, book it
         seats[seatNumber] = true;
         System.out.println(userType + " successfully booked Seat " + seatNumber);
         return true;
      } else {
         System.out.println(userType + " failed to book Seat " + seatNumber + " (Already
booked)");
         return false;
      }
   }
}

// Thread class for booking tickets
class UserThread extends Thread {
   private final TicketBookingSystem system;
```

```
    private final String userType;
    private final int seatNumber;

    public UserThread(TicketBookingSystem system, String userType, int seatNumber,
int priority) {
        this.system = system;
        this.userType = userType;
        this.seatNumber = seatNumber;
        this.setPriority(priority); // Set thread priority
    }

    @Override
    public void run() {
        system.bookSeat(seatNumber, userType);
    }
}

public class TicketBookingApp {
    public static void main(String[] args) {
        TicketBookingSystem system = new TicketBookingSystem();

        // Creating multiple threads (VIP users get higher priority)

        Thread vip1 = new UserThread(system,"VIP User 1", 2,
        Thread.MAX_PRIORITY);

        Thread vip2 = new UserThread(system, "VIP User 2", 3,
        Thread.MAX_PRIORITY);

        Thread normal1 = new UserThread(system, "Normal User 1", 2,
        Thread.MIN_PRIORITY);

        Thread normal2 = new UserThread(system, "Normal User 2", 4,
        Thread.NORM_PRIORITY);

        Thread normal3 = new UserThread(system, "Normal User 3", 1,
        Thread.NORM_PRIORITY);

        // Start threads
        vip1.start();
        vip2.start();
        normal1.start();
        normal2.start();
        normal3.start();
    }
```

```
}
```

4. Output:

```
VIP User 1 successfully booked Seat 2
VIP User 2 successfully booked Seat 3
Normal User 1 failed to book Seat 2 (Already booked)
Normal User 3 successfully booked Seat 1
Normal User 2 successfully booked Seat 4
```