CSE 228

INITIAL PROJECT REPORT

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE WITH MACHINE LEARNING)

TOPIC: Discrete Event Simulation

By

Pathakanuru Hari Hara Naidu

Registration No: 12210363

Supervisor

AMAN KUMAR

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

DECLARATION STATEMENT

I hereby declare that the work reported in the Assignment Project entitled “DISCRETE EVENT SIMULATION” in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering – Data Science with Machine Learning at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr. Aman Kumar. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Pathakanuru Hari Hara Naidu

Reg No: 12210363

TABLE OF CONTENTS

INTRODUCTION

Discrete Event Simulation in Java is a powerful technique for modeling and analyzing complex real-world systems using computer programs. It mimics the behavior of systems where events occur at distinct points in time. In this approach, we represent events as discrete points in time, such as customer arrivals in a queue, and use Java to simulate their occurrence and effects. This helps us study and optimize systems like traffic networks, manufacturing processes, or even healthcare operations.

Java, as a versatile programming language, provides a robust platform for implementing these simulations. By coding the logic of how events interact and progress through time, we gain insights into system performance and can test various scenarios. Discrete Event Simulation in Java is a valuable tool for decision-making and problem-solving in diverse fields.

OBJECTIVES AND SCOPE OF THE PROJECT

Objectives:

The primary objectives of a Discrete Event Simulation (DES) project in the context of data structures and Java are as follows:

Model Validation: Create an accurate representation of a real-world system using data structures in Java to ensure that the simulation model behaves in a manner consistent with the actual system

Performance Analysis: Analyze system performance by using DES to simulate various scenarios, helping identify bottlenecks, optimize processes, and make informed decisions for resource allocation.

Decision Support: Employ DES to assess different strategies

and policies, aiding decision-makers in choosing the most efficient and cost-effective options for system improvement.

Sensitivity Analysis: Conduct experiments and "what-if" scenarios to understand the impact of parameter variations on the system, helping in risk assessment and decision-making.

Real-time Adaptation: Develop the ability to adapt and modify the simulation model in real-time to address changing requirements and conditions, showcasing the flexibility of data structures in Java.

Reporting and Visualization: Provide clear and meaningful output through reports and graphical representations to communicate simulation results effectively to stakeholders.

Scope:

A Discrete Event Simulation project in Java involves modeling and analyzing dynamic systems with discrete, well-defined events.

- Use data structures like queues and arrays to manage event scheduling and state changes.
- Simulate real-world scenarios, such as traffic flow, manufacturing processes, or network congestion, to gain insights and optimize system performance.
- Create event-driven models that mimic the behavior of entities or objects in the system.
- Implement algorithms to advance the simulation clock and handle events in chronological order.
- Gather statistical data and perform experiments to evaluate system performance and make informed decisions.
- Ideal for testing and optimizing system designs without the need for costly real-world trials.

- Valuable for various domains, including logistics, healthcare, and finance.
- Enhance your programming and problem-solving skills while addressing real-world problems.
- Ensure efficient code design and data structure utilization to achieve accurate and fast simulations.

METHODOLOGY:

A Discrete Event Simulation (DES) project in Java, focused on data structures, follows a systematic methodology:

Problem Definition: Clearly define the problem to be simulated, its objectives, and the system's components and events.

Data Structures: Identify the data structures needed to represent the system state, events, and entities. Use collections like queues or lists to manage event scheduling and entity attributes.

Event Scheduling: Develop an event-driven model, where events are scheduled in a priority queue based on their timestamps.

Simulation Clock: Maintain a simulation clock to keep track of the current time and advance it as events are processed.

Event Handling: Implement event handlers to process events, update the system state, and generate new events.

Experimentation: Run the simulation multiple times with different inputs and scenarios, collecting relevant statistics and data.

Validation: Validate the simulation results against real-world data or analytical models.

Analysis: Analyze and interpret the simulation output to draw conclusions and make informed decisions.

Optimization: Fine-tune the simulation model and data structures for better performance and accuracy.

Documentation: Document the code, assumptions, and results for future reference. Ensure modularity and efficiency in your Java code, making use of appropriate data structures to handle events and system state efficiently.

FLOWCHART:

The flowchart of the project illustrates the step-by-step process from initializing the simulation to generate reports and visualization.

Algorithm Implementation (Pseudocode):

```
// Define the data structures for the simulation
class Event {
    // Properties: event type, timestamp, and any relevant
data
}

class EventQueue {
    // Implement a priority queue to manage events based on
their timestamps
    // Define methods for adding, removing, and checking
events
}

class Simulation {
    EventQueue eventQueue;
    // Other data structures to represent the state of the
simulation

    // Initialize the simulation
    InitializeSimulation() {
```

```
        // Set up initial state and populate the event queue
    }

    // Main simulation loop
    RunSimulation() {
        while (!eventQueue.isEmpty()) {
            Event currentEvent = eventQueue.pop(); // Get
the next event
            ProcessEvent(currentEvent); // Handle the event
        }
    }

    // Event handling logic
    ProcessEvent(Event event) {
        // Implement logic to update the simulation state
based on the event type
    }
}

// Main program
public class Main {
    public static void main(String[] args) {
        Simulation simulation = new Simulation();
        simulation.InitializeSimulation();
        simulation.RunSimulation();
    }
}
```

SUMMARY

In conclusion, the Discrete Event Simulation project in Java successfully showcased the power of efficient data structures in modeling complex systems. By leveraging structures like priority queues, lists, and arrays, we achieved precise event sequencing and system state management.

This project not only exemplified the importance of data structures in simulation but also highlighted their role in optimizing real-world processes. Through meticulous design and implementation, it demonstrated how well-chosen data

structures can enhance the accuracy and performance of simulations, making it an invaluable tool for decision-making in various domains.