**EEX5362 -Performance Modeling**

**Deliverable 01**

Name: K.W.P.G.A.N. Samarathunga

Registration No: 621425733

S No: S92065733

Date: 31/10/2025

**High-Level Problem Description**

Managing a call center efficiently is essential to ensure customer happiness and efficient use of resources. The call center will be the most common interaction a customer has with an organization for support. During high call volume times, the call center can have more than enough queues, customers are stuck on hold for too long, which gets many long waits and service becomes poor. On the other hand, having too many agents working can often lead to agents sitting idle just wasting unnecessary labor costs.

The primary objective is to meet the optimal number of agents, which means customers are not waiting too long and agents are not overworked. By simulating the call center operations such as call durations, arrivals and queues we can measure wait times, agent workload, calls handled and queue lengths. This allows for finding the best level that ensures customer satisfaction, effective resource use, and call center continuity, finally allowing the call center to maintain efficient operations and allow for data driven decisions.

**System Overview**

The simulation models a customer service call center where customers call the center seeking assistance from available agents. Calls are answered in order of arrival (First-Come, First-Served) and if all agents are busy, new callers enter the queue and wait until they are served when an agent becomes available.

Main Components

* Customers: Arrive randomly (every 1–6 minutes) and wait if no agent is free
* Agents: Each handles one call at a time, with service times lasting 3–8 minutes.
* Queue: Stores waiting customers until an agent becomes available.
* Simulation Environment: Uses SimPy to control call arrivals, service times, and resource allocation.

Workflow

* A customer arrives at the call center.
* If an agent is available → the customer is served immediately.
* If all agents are busy → the customer joins the waiting queue.
* When an agent finishes a call → the next waiting customer is served.

Performance Metrics

* Average waiting time
* Agent utilization
* Number of calls handled per hour (throughput)
* Maximum queue length

**Simulation Input**

Simulation Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Value / Range** |
| Simulation Duration | Total operation time | 480 minutes (8 hours) |
| Number of Agents | Staff available | 2, 3, 5 (tested scenarios) |
| Inter-Arrival Time | Time between incoming calls | Uniform (1, 6) minutes |
| Service Time | Duration per call | Uniform (3, 8) minutes |
| Queue Discipline | Order of service | FIFO (First-Come, First-Served) |
| Replications | Repeated simulation runs | 30 |

Data Generation

All call arrivals and service times are randomly created using Python’s random.uniform() function to make the simulation more realistic.

Performance Data Collected

* Average waiting time
* Agent utilization
* Calls handled per hour
* Maximum queue length

Output Dataset

The simulation generates an output dataset containing detailed performance statistics. This output is automatically saved as a CSV file for later analysis and visualization.

**Performance Aspects**

The simulation measures four main factors of call center performance

* Average Waiting Time: How long customers wait
* Agent Utilization: How busy agents are.
* System Throughput: Number of calls handled in 8 hours.
* Queue Length: How many customers are waiting.

|  |  |  |
| --- | --- | --- |
| **Performance Aspect** | **Primary Focus** | **Objective** |
| Average Waiting Time | Customer satisfaction | Minimize |
| Agent Utilization | Resource efficiency | Optimize (70–85%) |
| System Throughput | Service capacity | Maximize |
| Queue Length | System stability | Minimize congestion |