# Project Documentation: Queueing Calculator and Simulation Tool


## Developed By

- Ahmed Raza (B21110006009)
- Dinesh Kumar (B21110006026)
- Syed Muhammad Abbas Shah (B21110006131)
- Syed Zohaib Ahmed Qadri (B21110006142)

## Group No

- 12


## Course Code

- BSCS – 506 (Modelling and Simulation)

## Date of Development

- 26-12-2024

## Submitted To

- Dr. Shaista Raees


## UBIT, University Of Karachi

# Table of Contents

# 1. Introduction

The Queueing Calculator and Simulation Tool is designed to model and simulate various queueing systems, including MM1, MMC, MG1, MGC, GG1, and GGC. It combines analytical calculation with dynamic simulation to provide insights into the performance and characteristics of queueing systems.

# 2. Features

- **Analytical Calculations**: Computes queueing metrics for predefined models (MM1, MMC, MG1, MGC, GG1, GGC).
- **Simulation Capability**: Simulates user-defined queueing systems with customizable parameters.
- **Graphical User Interface (GUI)**: User-friendly interface for input, output, and visualization.
- **Visualization**: Charts for metrics such as waiting time, turnaround time, server utilization, and inter-arrival times.
- **Dynamic Gantt Charts**: Visualize server activity and process scheduling.

# 3. Project Structure

- File 1: **MMC.py**
    - Handles MMC queueing model with multiple servers where C represents number of servers and M represents **Exponential** distribution.
- File 2: **MGC.py**
    - Simulates the MG1 model with generalized arrival and service distributions where C represents number of servers and M represents **Normal, Uniform, Gamma** distribution.

- File 3: **GGC.py**
    - Simulates the GGC model with multiple servers and generalized distributions where C represents number of servers and M represents **Normal, Uniform, Gamma** distribution.
- File 4: **GGC.py**
    - It simulates the data on the basis of formulas.
    - It simulates below models:
        1. MG1
        2. MGC
        3. MM1
        4. MMC
        5. GG1
        6. GGC

    - It calculates:
        1. Utilization of system (ρ ).
        2. Length of queue (Lq)
        3. Average wait in queue (Wq)
        4. Average wait time in the system (W)
        5. Average number of customers in the system (L)

- File 5: **Simulator.py**
    - It is main file which displays 3 choices.
        1. Queuing Calculator.
        2. Simulator.
        3. Exit (Close Application).

# 4. Detailed Functionality

## 4.1 Queueing Models

1. **MM1**:
    a. Single-server queue with exponential inter-arrival and service times.
    b. Outputs metrics like average Start time, End time, waiting time, utilization, queue length, turnaround time, Response time.
2. **MMC**:
    a. Multi-server queue with exponential distributions.
    b. Outputs metrics like average Start time, End time, waiting time, utilization, queue length, turnaround time, Response time.
3. **MG1**:
    a. Single-server queue with general service times.
    b. Outputs metrics like average Start time, End time, waiting time, utilization, queue length, turnaround time, Response time.
4. **MGC**:
    a. Extends MG1 to multi-server queues.
    b. Outputs metrics like average Start time, End time, waiting time, utilization, queue length, turnaround time, Response time.
5. **GG1**:
    a. General inter-arrival and service time distributions.
    b. Outputs metrics like average Start time, End time, waiting time, utilization, queue length, turnaround time, Response time.
6. **GGC**:
    a. General inter-arrival and service time distributions with multiple servers.
    b. Outputs metrics like average Start time, End time, waiting time, utilization, queue length, turnaround time, Response time.

## 4.2 Simulation Parameters

- **Arrival Distribution**: Poisson, Normal, Uniform, or Gamma.
- **Service Distribution**: Exponential, Normal, Uniform, or Gamma.
- **Number of Servers**: Configurable for multi-server models.
- **Customer Count**: Determines the total number of customers (processes) in the simulation.

# 5. User Interface

The GUI provides the following:

- **Input Section**: Allows users to select distributions, set parameters, and define server counts.
- **Output Section**: Displays metrics in tabular form and provides visualization options.
- **Interactive Buttons**:
    - Show Gantt charts for individual servers.
    - View charts for Arrival, turnaround, waiting, response times, and server utilization.

# 6. Dependencies

- **Python 3.8+**
- **SimPy**: For event-based simulation.
- **Tkinter**: For the graphical user interface.
- **Matplotlib**: For data visualization.
- **Numpy**: For statistical computations.
- **PIL**: Python Image Library for image processing
- **Math**: For mathematic calculations.
- **Random**: For Random numbers.
- **Subprocess**: To run multiple files from same directory.

# 7. Usage Instructions

1. **Setup**:
    a. Ensure Python and required libraries are installed.
    b. Clone the project repository.
    c. Navigate to the project directory.
2. **Run the Application**:
    ❖ Simulator.py
3. **Input Parameters**:
    a. Select arrival and service distributions.
    b. Enter parameters for chosen distributions.
    c. Define the number of servers and customer count.

4. **Run Simulation**:
    a. Click the "Simulate" button.
    b. View results in the output table and generate charts.
5. **Visualization**:
    a. Click buttons to view Gantt charts and metric-specific graphs.

# 8. Challenges and Solutions

## Challenges

1. **Handling Negative Inter-Arrival and Service Times**:
    a. Resolved by clamping values to zero.
2. **Server Assignment in Simulations**:
    a. Implemented dynamic server selection based on next available time.
3. **Graphical Overlap in Charts**:
    a. Added text annotations and spacing for clarity.
4. **Making system Preemptive:** Difficulty is making system run on priority.

# 9. Future Enhancements

- **Performance Optimization**: Improve simulation speed for large-scale systems.
- **Enhanced Visualizations**: Introduce 3D visualizations and heatmaps.
- **Making system Preemptive:** To make system run of priorities.

# 10. Advantages of System

1. **Pre-Simulation Capability**: Virtually simulates the queueing system without the need for an actual setup, saving time and resources.
2. **Real-Time Analysis**: Provides immediate feedback on system performance, allowing for quick adjustments and optimization.
3. **Scenario Testing**: Enables testing of various configurations and parameters before implementation, reducing risks in real-world deployment.
4. **Cost-Effective**: Avoids the expense of building and maintaining physical systems for testing different queueing scenarios.

5. **Decision Support**: Assists in making data-driven decisions on system design, staffing, and resource allocation based on simulation results.
6. **Flexibility**: Offers the ability to adjust and experiment with different queueing models and parameters without actual hardware constraints.
7. **Visualization of Outcomes**: Provides visual representation of simulated system behavior, aiding in the understanding of complex processes and bottlenecks.
8. **Improved System Design**: Helps in designing more efficient and optimized systems by simulating various real-world conditions and identifying potential issues early.

# 11. Conclusion:

The Queueing Calculator and Simulation Tool combines analytical calculations and dynamic simulations for various queueing models, offering valuable insights into system performance. Its intuitive interface and detailed metrics make it a practical resource for both educational and professional use. Despite challenges, the tool provides robust functionality with accurate results. Future enhancements aim to further improve its performance and visualization features.