DOCUMENTATION

ASSIGNMENT 2

STUDENT NAME: COMAN ALECSIA-COSTINA

GROUP: 30423

# CONTENTS

[1. Assignment Objective 3](#_Toc161693441)

[2. Problem Analysis, Modeling, Scenarios, Use Cases 3](#_Toc161693442)

[3. Design 6](#_Toc161693443)

[4. Implementation 9](#_Toc161693444)

[5. Results 13](#_Toc161693445)

[6. Conclusions 14](#_Toc161693446)

[7. Bibliography 14](#_Toc161693447)

# Assignment Objective

Create and develop a simulation tool that evaluates queuing systems to identify and reduce customer wait times. Queues are a familiar sight in both actual scenarios and theoretical models. The primary goal of a queue is to serve as a holding area for customers awaiting service. Queue system managers aim to decrease the duration customers spend in line.

Introducing additional service points, or queues, each with its own server, is one strategy to cut down on wait times. However, this can lead to higher operational costs. When a new service point is added, the waiting customers are redistributed evenly across all available queues.

The software will mimic a sequence of customers arriving for services, joining queues, waiting, receiving services, and then exiting the queue. It will monitor and report the average time customers wait in line.

To determine wait times, it’s essential to know the arrival, service completion, and service duration times. These times vary based on individual customer arrival and their specific service requirements. The completion time is influenced by the queue count, the number of customers already in line, and their service demands.

*Input parameters* include:

* The duration of the simulation;
* The number of queues;
* The range for customer arrivals;
* The range for service duration;
* The strategy policy: based on time or queue;

*The output* should include:

* The average waiting time, service duration and peak hour;
* A log detailing how tasks are processed inside the serves over time;
* An illustration of queue dynamics;

# Problem Analysis, Modeling, Scenarios, Use Cases

* PROBLEM ANALYSYS

The application is designed to simulate the experience of customers queuing for services, such as those found in supermarkets or banks. In these real-world settings, customers must wait their turn in lines, with each queue handling multiple clients at the same time. The core concept of the application is to assess the service capacity within a given timeframe. This is achieved by inputting various parameters into a straightforward and accessible graphical user interface, which allows for easy interaction and manipulation of the simulation variables. The goal is to provide insights into the operational efficiency and to explore different scenarios that could enhance the customer service process.

* MODELLING

In this simulation, customers are created using a random generator, with each assigned a unique service duration and time of arrival, based on predefined input ranges. The simulation allows the user to configure:

* The **maximum number of queues** that can be active for customer processing.
* The **arrival interval range**, specifying the minimum and maximum time between the arrival of customers seeking services, measured in seconds.
* The **service time range**, indicating the minimum and maximum duration, in seconds, that a customer requires to be serviced, selected at random.
* The **simulation interval**, marking the start and end times of the simulation period.

The user can access the following data:

* The **average wait time** for customers in each queue, also in seconds.
* The **busiest period**, denoted as the ‘peak hour’, during which the highest number of customers are served.
* The **aggregate number of customers** who have been serviced throughout the simulation.
* The **mean duration of service** provided, in seconds.
* The **count of customers** and the **mean wait time** during a user-specified interval.
* SCENARIOS

Use Case: simulation setup

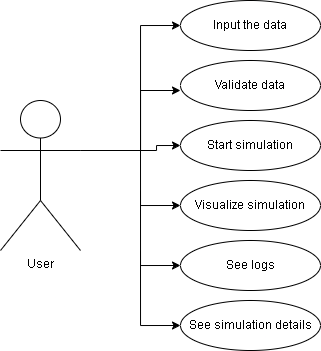
Primary Actor: user

Primary Success Path:

1. The user inputs the required figures for the client count, queue count, simulation duration, and the range for both arrival and service times.
2. The user selects the ‘validate input data’ option.
3. The system checks the data and, if correct, prompts the user to commence the simulation.

Alternate Flow: Incorrect Configuration Data

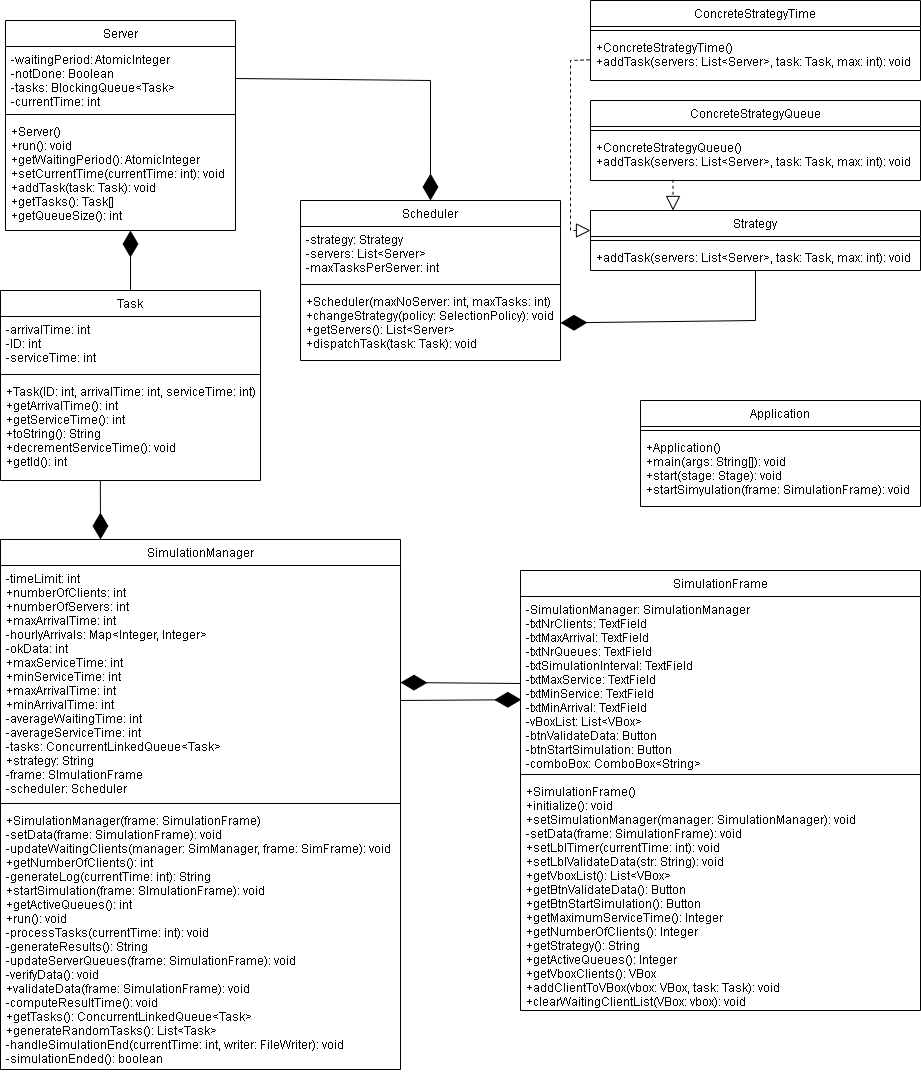
* The user enters incorrect figures for the setup parameters of the application.
* The system flags an error and instructs the user to provide correct values.
* The process reverts to the initial step.
* USE CASE DIAGRAM



The stickman depicted in the diagram symbolizes the actor, which is the user interacting with the application. Connected to the stickman are six ovals representing its potential actions: inputting the data for the simulation, validate it, start the simulation if introduced data is correct, visualize how the queues work, see logs and simulation details.

# Design

* CLASS DIAGRAM



* OOP DESIGN
* DATA STRUCTURES
* PAKAGES

# Implementation

* POLYNOMIAL CLASS
* OPERATIONS CLASS
* CONTROLLER CLASS
* APP CLASS
* CALCULATOR CLASS (GUI)

# Results

# Conclusions

# Bibliography