Customer Service Simulator

Ahmed Magdy Ibrahim 3003
Basma Kadry Osman 3052

Introduction

This program is a simulator which calculates the average waiting time for a random number of customers who have arrived at random times. A technically detailed explanation of the code of this program is coming up in this report.

First of all, the program is divided into 4 classes.

1. Class Program

It is created by default where it has the main entry point for the application as it contains the Main function.

2. Class MzQueue

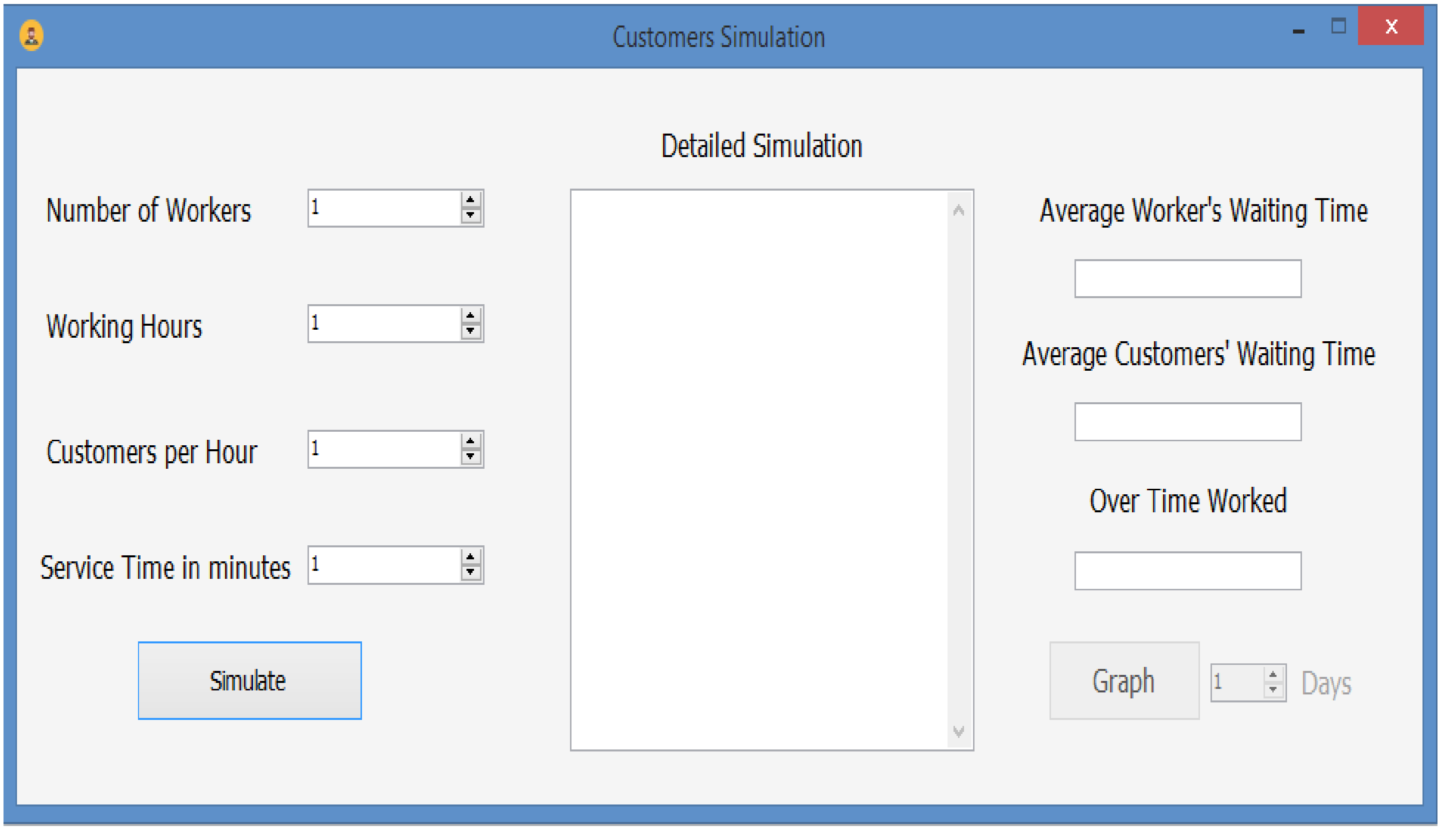
It consists of 7 functions. The MzQueue initializes the queue. The Enqueue, Dequeue, isFull, and isEmpty are the implementation of the queue. Qsort is a function that sorts the numbers in the queue. Swap function swaps 2 numbers.

3. Class Worker

It consists of 3 functions and the variable Finished is the time when the worker finishes the last customer. Worker function initializes the variable by zero. Work function calculates it. IsBusy function checks whether the worker has a customer at the arrival of another one or not

4. Class Form

This class is where all the work is done. We designed a simple form as shown in Figure.



Customers Simulation

Number of Workers 1

Working Hours 1

Customers per Hour 1

Service Time in minutes 1

Average Worker's Waiting Time

Average Customers' Waiting Time

Over Time Worked

1 Days

The class form consists of the function called when simulate button is pressed and other functions. When the button is pressed a random number is created whose range is from 0 to the max number of customers that can be received. This random number is the number of customers. If the number of customers is zero the outputs are zero and the program will end. If it's not zero another random variable will be created that stands for the arrival time of each customer and it's range is from 0 to the amount of working minutes. The function Qsort is then used to sort these random numbers. Then for each worker there are 3 cases, either the worker has no customers then the waiting time of the workers will increase, or the worker has just finished a customer at the arrival of another one, or the worker is busy with a customer at the arrival of another and in this case the waiting time of the customers will increase. The last value assigned to variable Finished will be the closing time. We can then calculate if there is extra time waited by the workers or if the workers have worked for overtime. Then if the user wants to graph the customer's waiting time he will click on Graph which will access the function Graph. This function does all the above process for the number of days the user has entered and graph the average customers' waiting time in these days.

This is a brief explanation of our code. The user guide explains how the program exactly works.