**QUEUES SIMULATOR**

**Neag Dragoș-Ion**

1. **Assignment objective**

**Main objective**

The main objective for this assignment was implementing a queue simulator that aims to analyze queuing-based systems to determine and minimize clients’ waiting time.

**Sub-objectives**

* Analyze the problem and identify requirements.
* Design the queues and choose an appropriate representation of the clients.
* Implement the queue simulator.
* Test the queue simulator.

1. **Problem analysis**
2. Modeling

The purpose of the problem is creating a realistic queue simulator, able to distribute clients efficiently to the queues.

To achieve the purpose, a series of requirements had to be met:

* Functional requirements
* The simulator should allow users to insert the number of clients
* The simulator should allow users to insert the number of queues
* The simulator should allow users to insert the simulation time
* The simulator should allow users to insert the minimum and maximum service time
* The simulator should allow users to insert the minimum and maximum arrival time
* The simulator should display the real-time evolution of the queues
* The simulator should stop the simulation when the simulation time was reached or when there are no clients waiting and in the queues
* Non-Functional requirements
* The simulator should be intuitive and easy to use by the user
* The simulator should indicate the user when the format he typed in is wrong

1. Scenarios

A picture containing graphical user interface

Description automatically generated

**Use Case:** start simulation

**Primary Actor:** user

**Main Success Scenario:**

Step1. The user inserts the number of clients, the number of queues, the simulation time, the minimum and maximum service time and the minimum and maximum arrival time in the graphical user interface.

Step2. The user presses the “start” button

Step3. The simulator begins the simulation and displays the evolution of the queues.

**Alternative Sequence:** Incorrect input values

* The user inserts incorrect input values (e.g. inserts not only digits, some fields are empty, the maximum service time is smaller than the minimum service time and the same for arrival time)
* The scenario returns to step 1

1. **Design**

Diagram

Description automatically generatedLevel 1: Overall system design

Level 2: Division into packages

In order to create a good work environment, I decided to make use of the architectural patterns, specifically the Model View Controller. The architectural patterns define structures for software systems in terms of predefined subsystems and their responsibility.

* The model is responsible with encapsulating the application state and responding to state queries.
* The controller is the actuator, that defines the actual behaviour of the calculator and is in charge with handling the response.
* The view is basically the graphical part of the application, as it renders the models and requests updates from models.

Chart, diagram

Description automatically generated

Level 3: Division into classes

Chart, diagram, box and whisker chart

Description automatically generated

* View package

Inside this package, we can find the class Dashboard, the user interface class, which holds the pieces of code related to the graphical interface. Besides it I’ve also created an abstract class named AppFrame, which inherits the JFrame class, and an exception that gets thrown when the polynomials typed in by the user do not respect the required format.

* Model package

In this package I’ve stored two classes, which are: Client, Server. These classes encapsulate the queues and the clients that would take part in the simulation.

* Controller package

Lastly, in the controller package there are six classes: Scheduler, ConcreteStrategyTime, ConcreteStrategyQueue, SimulationManager, SelectionPolicy and Strategy. The scheduler class is responsible with the distribution of each client to one queue, and also with removing a client from a queue. The strategy, along with the concrete strategies have the role of distributingthe clients into a queue based on the waiting time or on the number of clients in the queues. The simulation manager initiates the simulation by generating random clients and measuring the time at each step.

1. **Implementation**

* Main

The Main class is only responsible for initializing the dashboard (the GUI).

* Dashboard

The class Dashboard is the class that implements the graphical user interface using Java Swing.

The design I went for is presented in the picture below:

The attributes for the Dashboard class consists of seven JTextFields: clientsField, queuesField, timeField, arrivalMinField, arrivalMaxField, serviceMinField and serviceMaxField, one JTextArea: updateTextArea, one JButton: startButton, a path that specifies the location where the log file will be created, a simulationManager that creates a bond between the GUI and the simulation manager, used to update the results, and a fileWriter, which is used in order to write in the log file.

* initialize – this method creates the JFrame and sets its size and initializes the panel
* initializeSimulator – the method is used to add the necessary text fields in the panel, as well as the button. The interface contains only one button as it is unnecessary to include any others since the purpose of the button is to start the simulation.
* initializeSimulatorListeners – this method is responsible for the actions that occur when the button is pressed. First, a format check will be performed on all of the fields, and if the tests are passed the main thread of the simulation manager will be started.
* updateResults – the method that is used by the simulation manager in order to update the interface at each step so the user can have a real-time evolution of the queues. It overwrites the frame with a new panel that contains the latest status of the queues.
* checkFormat – this class is used for checking the format of each user input and it uses general expressions to check if the fields are valid. The main condition for a field to be valid is to only contain digits.

Graphical user interface, application, Word

Description automatically generated

* Client

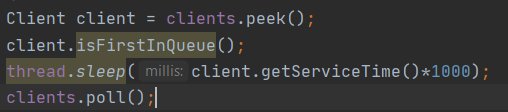
The public class Client stores four attributes: id, arrivalTime, serviceTime, isFirst. When it comes to methods, besides the getters and setters, there also is a method isFirstInQueue, which instantiates the isFirst attribute with 1 when the client becomes the head of a queue.

* Server

The Server class also stores four attributes: id, clients, waitingPeriod and thread. This class is the representation of a server that manages a queue.

The methods of the class are:

* Getters and setters
* startThread – methods that creates a new thread and starts it
* addClient – method that adds a given client to the queue ‘clients’ and increases the waitingPeriod by the service time of the client
* run – this method executes itself since the thread is started and while the queue is not empty, it takes the client at the head of the queue and puts the thread on sleep for the amount of seconds that client has as service time, only to delete that client afterwards.



* Scheduler

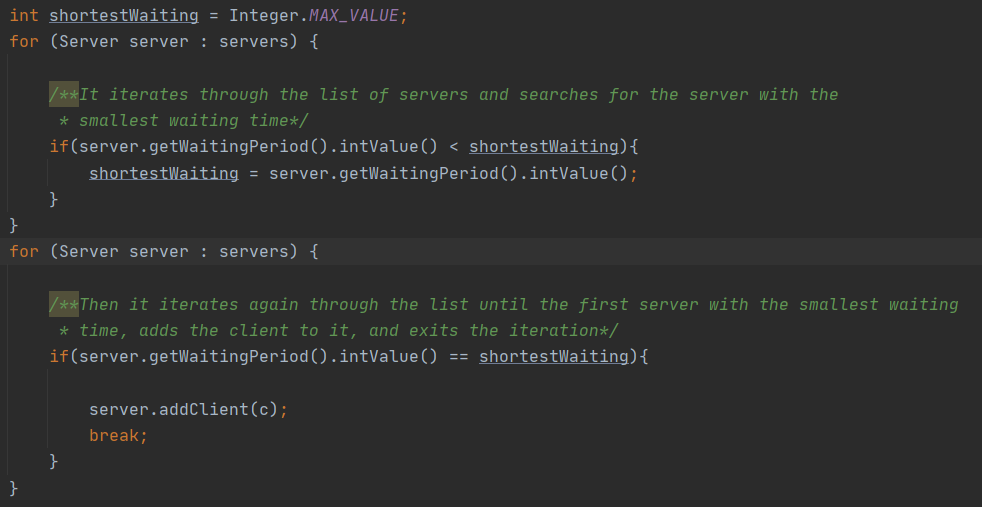
The Scheduler class has the role of distributing each client to a specific queue, based on a criteria the developer chooses: based on time or on persons in front of the client.

The class uses three attributes: servers, maxNoServers, strategy.

* Scheduler - the constructor initializes the attributes and also creates maxNoServers servers and starts the thread of each server
* changeStrategy – method used to change the client distribution strategy based on a given policy, that can select either the shortest queue or the shortest time
* dispatchClient – based on the selected strategy, the method adds the client into a queue
* ConcreteStrategyTime

The ConcreteStrategyTime class implements the Strategy interface and so it only has one method:

* addClient – the method iterates through all the queues to find the queue that has the shortest waiting time. After that, it begins to iterate once again through all the queues until it finds the first queue that has the minimum waiting time. When that queue is found, the client is added to the queue and the iteration is stopped.



* ConcreteStrategyQueue

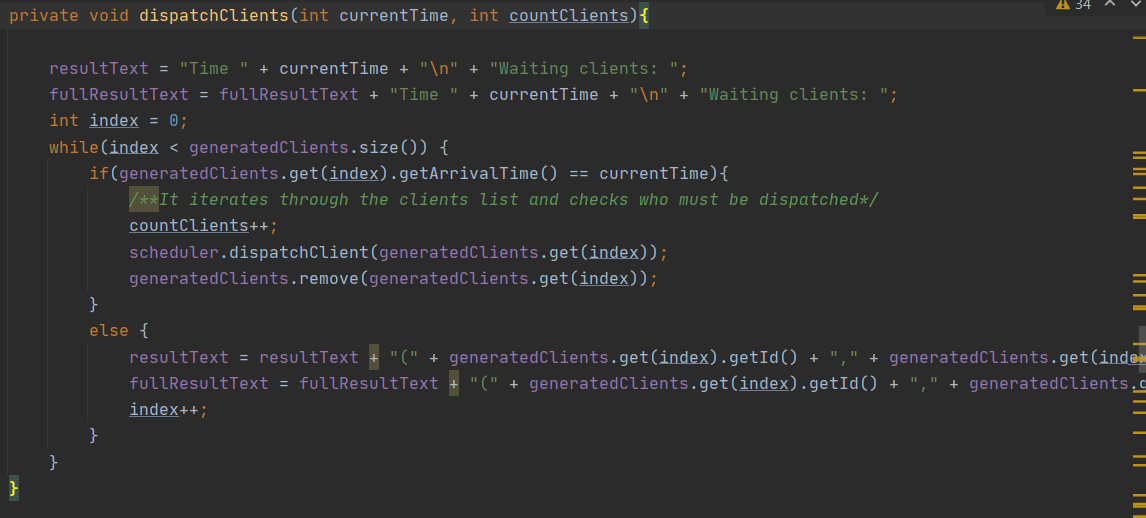
The ConcreteStrategyQueue class also implements the Strategy interface and so it only has one method:

* addClient – the method iterates through all the queues to find the queue that has the minimum number of clients in it at that time. After that, it begins to iterate once again through all the queues until it finds the first queue that has the minimum clients stored. When that queue is found, the client is added to the queue and the iteration is stopped.
* SimulationManager

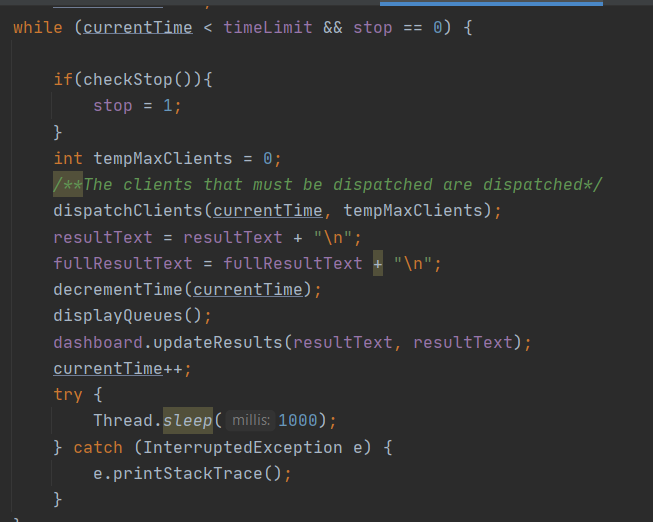
The SimulationManager is the most important class, as it holds the application together. It receives the inputs from the GUI, generates random clients and then proceeds with the proper simulation of clients entering and exiting a number of queues.

The attributes of the class are: timeLimit, maxProcessingTime, minProcessingTime, maxArrivalTime, minArrivalTime, numberOfServers, numberOfClients, selectionPolicy, which represent the inputs of the application, as well as: averageWaitingTime, averageServiceTime, peakHour, maximumClients, which are related to statistics calculus, and lastly: scheduler, generatedClients, resultText, fullResultText, dashboard, stop.

* Getters and setters
* generateNRandomClients – the method generates a given number of random clients using the random functions. Each client gets a random arrival and service time, both between some given input values. After all the clients were generated, the list is sorted depending on the clients’ arrival time.
* compare – this method is used by the previous one in order to sort out the clients list
* dispatchClients – this method is responsible with iterating through the list of clients and checks which clients have the arrival time equal to the current time. If any such clients are found, they are dispatched into a queue and deleted from the list.



* decrementTime – this method is used in order to decrement the waiting time of a server at each step time, as well as decrementing the service time of each client. The method iterates through the servers, and also through the clients of each server, and if a client just became the head of the queue, the service time doesn’t decrement until the next iteration, and that’s where the isFirst attribute of a client comes into play. If that attribute is 1, it will become 2, and if tha attribute is 2, the service time gets decremented.
* checkStop – the method is used to check if the simulation must be stopped, meaning if there are no more clients left in the queues and if there are still clients waiting to get into a queue.
* displayQueues – this method is used to print the content of the queues at each iteration
* run – the run method executes itself until the simulation has to stop. It dispatches the clients that have to get into the queues by calling the dispatchClients method and then it uses the dashboard to update the results in the GUI. At the end it will compute the statistics, which consist in average waiting time, average service time and peak hour.



1. **Results**

In order to test the application, a series of 3 tests were applied, as suggested in the assignment presentation.

* This is the first test case:

N = 4

Q = 2

T simulation = 60 sec

T min/max arrival = [2,30]

T min/max service = [2,4]

* And the results are:

Time 0

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 1

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 2

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 3

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 4

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 5

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 6

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 7

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 8

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 9

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 10

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 11

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 12

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 13

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 14

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 15

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 16

Waiting clients: (1,17,2); (2,19,2); (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 17

Waiting clients: (2,19,2); (3,25,3); (4,29,2);

Queue 1: (1,17,2);

Queue 2:

Time 18

Waiting clients: (2,19,2); (3,25,3); (4,29,2);

Queue 1: (1,17,1);

Queue 2:

Time 19

Waiting clients: (3,25,3); (4,29,2);

Queue 1: (2,19,2);

Queue 2:

Time 20

Waiting clients: (3,25,3); (4,29,2);

Queue 1: (2,19,1);

Queue 2:

Time 21

Waiting clients: (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 22

Waiting clients: (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 23

Waiting clients: (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 24

Waiting clients: (3,25,3); (4,29,2);

Queue 1:

Queue 2:

Time 25

Waiting clients: (4,29,2);

Queue 1: (3,25,3);

Queue 2:

Time 26

Waiting clients: (4,29,2);

Queue 1: (3,25,2);

Queue 2:

Time 27

Waiting clients: (4,29,2);

Queue 1: (3,25,1);

Queue 2:

Time 28

Waiting clients: (4,29,2);

Queue 1:

Queue 2:

Time 29

Waiting clients:

Queue 1: (4,29,2);

Queue 2:

Time 30

Waiting clients:

Queue 1: (4,29,1);

Queue 2:

Time 31

Waiting clients:

Queue 1:

Queue 2:

Average waiting time: 0.041666668

Average service time: 2.5

Peak hour: 17

* The second test case is:

N = 50

Q = 5

T simulation = 60 sec

T min/max arrival = [2,40]

T min/max service = [1,7]

* The results are:

Time 0

Waiting clients: (1,2,6); (2,2,2); (3,2,5); (4,3,5); (5,3,6); (6,4,2); (7,4,3); (8,5,2); (9,5,1); (10,6,6); (11,7,5); (12,7,6); (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1:

Queue 2:

Queue 3:

Queue 4:

Queue 5:

Time 1

Waiting clients: (1,2,6); (2,2,2); (3,2,5); (4,3,5); (5,3,6); (6,4,2); (7,4,3); (8,5,2); (9,5,1); (10,6,6); (11,7,5); (12,7,6); (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1:

Queue 2:

Queue 3:

Queue 4:

Queue 5:

Time 2

Waiting clients: (4,3,5); (5,3,6); (6,4,2); (7,4,3); (8,5,2); (9,5,1); (10,6,6); (11,7,5); (12,7,6); (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (1,2,6);

Queue 2: (2,2,2);

Queue 3: (3,2,5);

Queue 4:

Queue 5:

Time 3

Waiting clients: (6,4,2); (7,4,3); (8,5,2); (9,5,1); (10,6,6); (11,7,5); (12,7,6); (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (1,2,5);

Queue 2: (2,2,1);

Queue 3: (3,2,4);

Queue 4: (4,3,5);

Queue 5: (5,3,6);

Time 4

Waiting clients: (8,5,2); (9,5,1); (10,6,6); (11,7,5); (12,7,6); (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (1,2,4);

Queue 2: (6,4,2); (7,4,3);

Queue 3: (3,2,3);

Queue 4: (4,3,4);

Queue 5: (5,3,5);

Time 5

Waiting clients: (10,6,6); (11,7,5); (12,7,6); (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (1,2,3); (9,5,1);

Queue 2: (6,4,1); (7,4,3);

Queue 3: (3,2,2); (8,5,2);

Queue 4: (4,3,3);

Queue 5: (5,3,4);

Time 6

Waiting clients: (11,7,5); (12,7,6); (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (1,2,2); (9,5,1);

Queue 2: (7,4,3);

Queue 3: (3,2,1); (8,5,2);

Queue 4: (4,3,2); (10,6,6);

Queue 5: (5,3,3);

Time 7

Waiting clients: (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (1,2,1); (9,5,1); (11,7,5);

Queue 2: (7,4,2); (12,7,6);

Queue 3: (8,5,2);

Queue 4: (4,3,1); (10,6,6);

Queue 5: (5,3,2);

Time 8

Waiting clients: (13,9,1); (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (9,5,1); (11,7,5);

Queue 2: (7,4,1); (12,7,6);

Queue 3: (8,5,1);

Queue 4: (10,6,6);

Queue 5: (5,3,1);

Time 9

Waiting clients: (14,10,5); (15,10,3); (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (11,7,5);

Queue 2: (12,7,6);

Queue 3: (13,9,1);

Queue 4: (10,6,5);

Queue 5:

Time 10

Waiting clients: (16,11,6); (17,11,1); (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (11,7,4);

Queue 2: (12,7,5);

Queue 3: (14,10,5);

Queue 4: (10,6,4);

Queue 5: (15,10,3);

Time 11

Waiting clients: (18,12,6); (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (11,7,3); (17,11,1);

Queue 2: (12,7,4);

Queue 3: (14,10,4);

Queue 4: (10,6,3);

Queue 5: (15,10,2); (16,11,6);

Time 12

Waiting clients: (19,13,1); (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (11,7,2); (17,11,1);

Queue 2: (12,7,3);

Queue 3: (14,10,3);

Queue 4: (10,6,2); (18,12,6);

Queue 5: (15,10,1); (16,11,6);

Time 13

Waiting clients: (20,14,1); (21,14,3); (22,14,3); (23,14,2); (24,14,3); (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (11,7,1); (17,11,1); (19,13,1);

Queue 2: (12,7,2);

Queue 3: (14,10,2);

Queue 4: (10,6,1); (18,12,6);

Queue 5: (16,11,6);

Time 14

Waiting clients: (25,15,2); (26,15,3); (27,15,4); (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (17,11,1); (19,13,1); (22,14,3);

Queue 2: (12,7,1); (20,14,1); (23,14,2); (24,14,3);

Queue 3: (14,10,1); (21,14,3);

Queue 4: (18,12,6);

Queue 5: (16,11,5);

Time 15

Waiting clients: (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (19,13,1); (22,14,3); (26,15,3);

Queue 2: (20,14,1); (23,14,2); (24,14,3);

Queue 3: (21,14,3); (25,15,2);

Queue 4: (18,12,5);

Queue 5: (16,11,4); (27,15,4);

Time 16

Waiting clients: (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (22,14,3); (26,15,3);

Queue 2: (23,14,2); (24,14,3);

Queue 3: (21,14,2); (25,15,2);

Queue 4: (18,12,4);

Queue 5: (16,11,3); (27,15,4);

Time 17

Waiting clients: (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (22,14,2); (26,15,3);

Queue 2: (23,14,1); (24,14,3);

Queue 3: (21,14,1); (25,15,2);

Queue 4: (18,12,3);

Queue 5: (16,11,2); (27,15,4);

Time 18

Waiting clients: (28,19,3); (29,19,3); (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (22,14,1); (26,15,3);

Queue 2: (24,14,3);

Queue 3: (25,15,2);

Queue 4: (18,12,2);

Queue 5: (16,11,1); (27,15,4);

Time 19

Waiting clients: (30,20,6); (31,20,4); (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (26,15,3);

Queue 2: (24,14,2);

Queue 3: (25,15,1); (28,19,3);

Queue 4: (18,12,1); (29,19,3);

Queue 5: (27,15,4);

Time 20

Waiting clients: (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (26,15,2); (31,20,4);

Queue 2: (24,14,1); (30,20,6);

Queue 3: (28,19,3);

Queue 4: (29,19,3);

Queue 5: (27,15,3);

Time 21

Waiting clients: (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (26,15,1); (31,20,4);

Queue 2: (30,20,6);

Queue 3: (28,19,2);

Queue 4: (29,19,2);

Queue 5: (27,15,2);

Time 22

Waiting clients: (32,23,6); (33,23,1); (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (31,20,4);

Queue 2: (30,20,5);

Queue 3: (28,19,1);

Queue 4: (29,19,1);

Queue 5: (27,15,1);

Time 23

Waiting clients: (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (31,20,3);

Queue 2: (30,20,4);

Queue 3: (32,23,6);

Queue 4: (33,23,1);

Queue 5:

Time 24

Waiting clients: (34,25,5); (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (31,20,2);

Queue 2: (30,20,3);

Queue 3: (32,23,5);

Queue 4:

Queue 5:

Time 25

Waiting clients: (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (31,20,1);

Queue 2: (30,20,2);

Queue 3: (32,23,4);

Queue 4: (34,25,5);

Queue 5:

Time 26

Waiting clients: (35,27,5); (36,27,5); (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1:

Queue 2: (30,20,1);

Queue 3: (32,23,3);

Queue 4: (34,25,4);

Queue 5:

Time 27

Waiting clients: (37,28,5); (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (35,27,5);

Queue 2: (36,27,5);

Queue 3: (32,23,2);

Queue 4: (34,25,3);

Queue 5:

Time 28

Waiting clients: (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (35,27,4);

Queue 2: (36,27,4);

Queue 3: (32,23,1);

Queue 4: (34,25,2);

Queue 5: (37,28,5);

Time 29

Waiting clients: (38,30,6); (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (35,27,3);

Queue 2: (36,27,3);

Queue 3:

Queue 4: (34,25,1);

Queue 5: (37,28,4);

Time 30

Waiting clients: (39,31,2); (40,31,5); (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (35,27,2);

Queue 2: (36,27,2);

Queue 3: (38,30,6);

Queue 4:

Queue 5: (37,28,3);

Time 31

Waiting clients: (41,32,1); (42,32,6); (43,32,4); (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (35,27,1); (40,31,5);

Queue 2: (36,27,1);

Queue 3: (38,30,5);

Queue 4: (39,31,2);

Queue 5: (37,28,2);

Time 32

Waiting clients: (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (40,31,5);

Queue 2: (41,32,1); (42,32,6);

Queue 3: (38,30,4);

Queue 4: (39,31,1); (43,32,4);

Queue 5: (37,28,1);

Time 33

Waiting clients: (44,34,3); (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (40,31,4);

Queue 2: (42,32,6);

Queue 3: (38,30,3);

Queue 4: (43,32,4);

Queue 5:

Time 34

Waiting clients: (45,35,1); (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (40,31,3);

Queue 2: (42,32,5);

Queue 3: (38,30,2);

Queue 4: (43,32,3);

Queue 5: (44,34,3);

Time 35

Waiting clients: (46,36,3); (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (40,31,2);

Queue 2: (42,32,4);

Queue 3: (38,30,1); (45,35,1);

Queue 4: (43,32,2);

Queue 5: (44,34,2);

Time 36

Waiting clients: (47,37,1); (48,38,3); (49,39,4); (50,39,3);

Queue 1: (40,31,1); (46,36,3);

Queue 2: (42,32,3);

Queue 3: (45,35,1);

Queue 4: (43,32,1);

Queue 5: (44,34,1);

Time 37

Waiting clients: (48,38,3); (49,39,4); (50,39,3);

Queue 1: (46,36,3);

Queue 2: (42,32,2);

Queue 3: (47,37,1);

Queue 4:

Queue 5:

Time 38

Waiting clients: (49,39,4); (50,39,3);

Queue 1: (46,36,2);

Queue 2: (42,32,1);

Queue 3: (48,38,3);

Queue 4:

Queue 5:

Time 39

Waiting clients:

Queue 1: (46,36,1);

Queue 2: (49,39,4);

Queue 3: (48,38,2);

Queue 4: (50,39,3);

Queue 5:

Time 40

Waiting clients:

Queue 1:

Queue 2: (49,39,3);

Queue 3: (48,38,1);

Queue 4: (50,39,2);

Queue 5:

Time 41

Waiting clients:

Queue 1:

Queue 2: (49,39,2);

Queue 3:

Queue 4: (50,39,1);

Queue 5:

Time 42

Waiting clients:

Queue 1:

Queue 2: (49,39,1);

Queue 3:

Queue 4:

Queue 5:

Time 43

Waiting clients:

Queue 1:

Queue 2:

Queue 3:

Queue 4:

Queue 5:

Average waiting time: 0.059666667

Average service time: 3.58

Peak hour: 14

1. **Conclusions**

Working with threads can be challenging, especially when there are multiple you have to deal with, as they require a lot of attention.

This project helped me learn something new and made me understand how to build a simulation on my own. I will definitely use these concepts in the future.