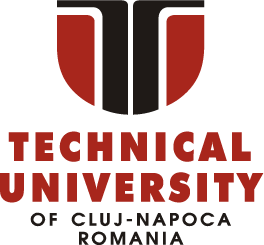
Technical University of Cluj-Napoca

Programming Techniques

Laboratory – Assignment 2

Queues Simulator

**

Teacher: prof. Ioan Salomie

Teacher Assistant: Viorica Rozina Chifu

Student: Andreica Ioan-Paul

Group: 30421

Contents

[1. Assignment Objective 3](#_Toc69776173)

[2. Problem analysis, modelling, scenarios, use cases 3](#_Toc69776174)

[3. Design 4](#_Toc69776175)

[4. Implementation 7](#_Toc69776176)

[5. Results 9](#_Toc69776177)

[6. Conclusions 12](#_Toc69776178)

[7. Bibliography 12](#_Toc69776179)

# Assignment Objective

Main objective:

Design and implement a simulation application aiming to analyse queueing-based systems for determining and minimizing clients’ waiting time.

Queues are commonly seen both in real world and in the models. The main objective of a queue is to provide a place for a "client" to wait before receiving a "service". The management of queue-based systems is interested in minimizing the time amount its "clients" are waiting in queues.

One way to minimize the waiting time is to add more servers i.e., more queues in the system (each queue is considered as having an associated processor) but this approach increases the costs of the supplier. When a new server is added the waiting clients will be evenly distributed to all current available queues.

The application should simulate a series of clients arriving for service, entering queues, waiting, being served and finally leaving the queue.

Secondary objectives:

1. Analyze the problem and identify requirements:
   1. Understand the required architectural pattern.
   2. Identify required classes.
   3. Understand use cases.
2. Design the queue simulator:
   1. Consider the best user interface.
   2. Understand the data flow from user to queue visualization and back.
   3. Identify the best strategy for queue implementation.
   4. Consider input validation method and how it affects data flow.
   5. Consider showing statistics.
3. Implement the queue simulator:
   1. Create sub-packages according to the architectural pattern.
   2. Start with the main classes (i.e., GUI, Controller, Client, Queue)
   3. Test input and output data manipulation
4. Test the queue simulator:
   1. Go through all possible scenarios from use cases.
   2. Use small and large data inputs to check for correctness.

# Problem analysis, modelling, scenarios, use cases

1. Analysing the problem:

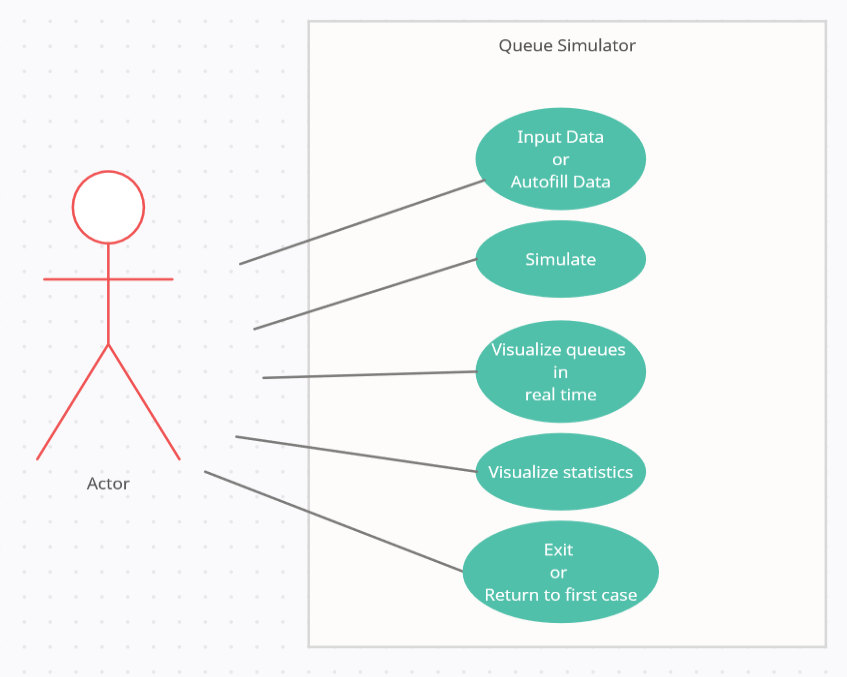
This application should simulate customers waiting to receive a service (e.g. supermarket, bank, etc.) just like in the real world, they have to wait in queues, each queue processing clients simultaneously. The idea is to analyse how many clients can be served in a certain simulation interval, by entering parameters in a graphical interface.

Application should take as inputs the maximum number of queues available, the number of incoming clients and their limitations (i.e., minimum/maximum time of arrival, min/max time of service), the simulation duration.

The user can visualize the changes in the queues and waiting list at every unit of time (second). After the simulation has ended the user is provided with some statistics like average waiting time, average service time and peak hour.

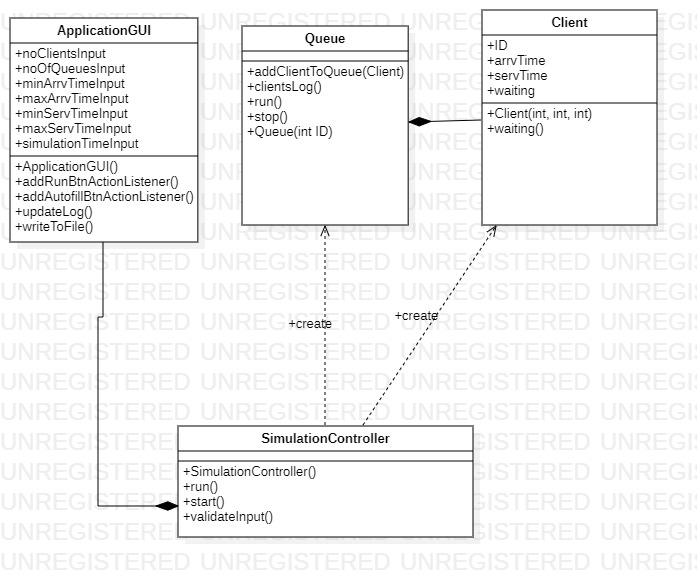
After the simulation has ended the user can rerun it with different or same input data.

1. Scenarios and Use Cases:



# Design

Class design:



**The packaging is done using MVC architecture:**

Java packages help in organizing multiple modules and group together related classes and interfaces.

In object-oriented programming development, model-view-controller (MVC) is the name of a methodology or design pattern for successfully and efficiently relating the user interface to underlying data models. The MVC pattern is widely used in program development with programming languages such as Java, Smalltalk, C, and C++.

The MVC pattern has been heralded by many developers as a useful pattern for the reuse of object code and a pattern that allows them to significantly reduce the time it takes to develop applications with user interfaces.

The model-view-controller pattern proposes three main components or objects to be used in software development:

- Model, which represents the underlying, logical structure of data in a software application and the high-level class associated with it. This object model does not contain any information about the user interface.

- View, which is a collection of classes representing the elements in the user interface (all of the things the user can see and respond to on the screen, such as buttons, display boxes, and so forth)

- Controller*,* which represents the classes connecting the model and the view, and is used to communicate between classes in the model and view.

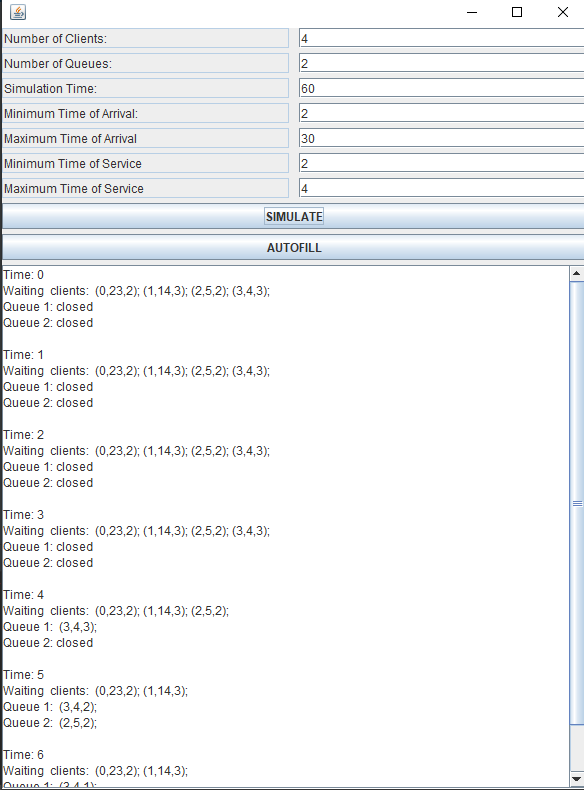
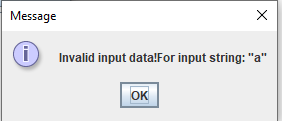
**Data Structures:**

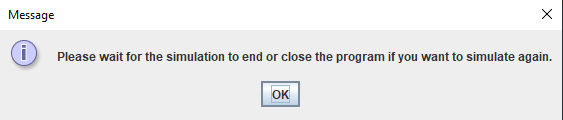
For this implementation I used only primitive data types to ensure the thread safety.

**Algorithms:**

Strategy for selecting the best queue was done by selecting the queue with smallest waiting time.

**User Interface:**



The user interface is simplistic, it assists the user by offering an autofill option with the default options, it provides auto scroll for the log window and shows an error dialog pointing him to the wrong input or if he attempts to restart a running simulation.

# Implementation

1. **The model** (containing the logic of the program):

The model is composed of the Client and Queue class both being created and dependent on the controller class.

Client:

* Contains client fields

Queue:

* Represents a thread
* Is composed of multiple clients waiting in queue
* Processes clients
* Remembers all the time client spent in queue and other statistical features

1. **The Controller**:

The controller class is built extremely straight-forward it has only one method that returns ActionListeners for the buttons and in which it checks if the input is valid or not.

Controller also provides the default data for auto fill option and shows alert dialogs.

At the end of a simulation the controller closes all the queues(threads) and allows the user to rerun the simulation

It connects the view and the model and handles errors by choosing the appropriate response.

Its constructor calls a method for data validation and initializes the empty queues and client wait list if everything is ok.

1. **The GUI**:

Its purpose is to display data to the user and feed input to the controller through its text fields and buttons.

It also has a function used to write to the log and text files at the end of each simulation.

Important methods:

private Queue selectBestQueue(Queue[] queues){  
 int bestQueueID = 0;  
 int smallestWaitingTime = Integer.*MAX\_VALUE*;  
 for (Queue q : queues) {  
 if (q.getWaitingTime() < smallestWaitingTime) {  
 bestQueueID = q.getID();  
 smallestWaitingTime = q.getWaitingTime();  
 }  
 }  
 return queues[bestQueueID];  
}

Best queue selection.

public void addClientToQueue(Client c){  
 Client client = new Client(c.getArrvTime(),c.getServTime(),c.getID());  
 clients[numberOfClients] = client;  
 numberOfClients = numberOfClients + 1;  
 if(numberOfClients == 1) clients[0].setServTime(client.getServTime()+1);  
}

Add client(task) to queue.

public String clientsLog(){  
 StringBuilder message = new StringBuilder();  
 for (int i = 0; i < numberOfClients; i++) {  
 message.append(" (").append(clients[i].getID()).append(",").append(clients[i].getArrvTime()).append(",").append(clients[i].getServTime()).append(")").append(";");  
 }  
 return message.toString();  
}

Log clients for queue.

private void startReviveThread(){  
 if(!this.localThread.isAlive()) {  
 this.localThread = new Thread(this);  
 }  
 simulationTime = 0;  
 localThread.start();  
}

Restart simulation controller thread.

public boolean validateInput(){  
 int tempMaxArrvTime, tempMinArrvTime, tempNumberOfExpectedClients, tempNumbOfQueues, tempMinServTime , tempMaxServTime , tempSimulationTime;  
 try {  
 tempMaxArrvTime = Integer.*parseInt*(this.applicationGUI.getMaxArrvTimeInput());  
 tempMinArrvTime = Integer.*parseInt*(this.applicationGUI.getMinArrvTimeInput());  
 tempNumberOfExpectedClients = Integer.*parseInt*(this.applicationGUI.getNoClientsInput());  
 tempNumbOfQueues = Integer.*parseInt*(this.applicationGUI.getNoOfQueuesInput());  
 tempMinServTime = Integer.*parseInt*(this.applicationGUI.getMinServTimeInput());  
 tempMaxServTime = Integer.*parseInt*(this.applicationGUI.getMaxServTimeInput());  
 tempSimulationTime = Integer.*parseInt*(this.applicationGUI.getSimulationTimeInput());  
 }catch (Exception e){  
 JOptionPane.*showMessageDialog*(null, "Invalid input data!" + e.getMessage());  
 return false;  
 }  
 if(tempMaxServTime > tempMinServTime && tempMaxArrvTime > tempMinArrvTime  
 && tempNumbOfQueues > 0 && tempSimulationTime > 0 && tempNumberOfExpectedClients > 0 && tempMinArrvTime > 0 && tempMinServTime > 0) {  
 this.maxArrvTime = tempMaxArrvTime;  
 this.minArrvTime = tempMinArrvTime;  
 this.numberOfExpectedClients = tempNumberOfExpectedClients;  
 this.totalClients = numberOfExpectedClients;  
 this.numberOfQueues = tempNumbOfQueues;  
 this.totalSimulationTime = tempSimulationTime;  
 this.minServTime = tempMinServTime;  
 this.maxServTime = tempMaxServTime;  
 return true;  
 }  
 JOptionPane.*showMessageDialog*(null, "Invalid input data!");  
 return false;  
}

Input validation method.

private void sendTaskToQueue(Queue queue, Client client){  
 queue.addClientToQueue(client);  
}

Send client to queue.

# Results

Test 1 results:

Time: 0  
Waiting clients: (0,2,2); (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 1  
Waiting clients: (0,2,2); (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 2  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: (0,2,2);  
Queue 2: closed  
  
Time: 3  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: (0,2,1);  
Queue 2: closed  
  
Time: 4  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 5  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 6  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 7  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 8  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 9  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 10  
Waiting clients: (1,21,3); (2,18,2); (3,11,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 11  
Waiting clients: (1,21,3); (2,18,2);  
Queue 1: (3,11,2);  
Queue 2: closed  
  
Time: 12  
Waiting clients: (1,21,3); (2,18,2);  
Queue 1: (3,11,1);  
Queue 2: closed  
  
Time: 13  
Waiting clients: (1,21,3); (2,18,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 14  
Waiting clients: (1,21,3); (2,18,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 15  
Waiting clients: (1,21,3); (2,18,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 16  
Waiting clients: (1,21,3); (2,18,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 17  
Waiting clients: (1,21,3); (2,18,2);  
Queue 1: closed  
Queue 2: closed  
  
Time: 18  
Waiting clients: (1,21,3);  
Queue 1: (2,18,2);  
Queue 2: closed  
  
Time: 19  
Waiting clients: (1,21,3);  
Queue 1: (2,18,1);  
Queue 2: closed  
  
Time: 20  
Waiting clients: (1,21,3);  
Queue 1: closed  
Queue 2: closed  
  
Time: 21  
Waiting clients:   
Queue 1: (1,21,3);  
Queue 2: closed  
  
Time: 22  
Waiting clients:   
Queue 1: (1,21,2);  
Queue 2: closed  
  
Time: 23  
Waiting clients:   
Queue 1: (1,21,1);  
Queue 2: closed  
  
Time: 24  
Waiting clients:   
Queue 1: closed  
Queue 2: closed  
  
Average Waiting Time: 0.0  
Average Service Time: 2.25  
Peak Time: 2 with 1 clients  
------------------SIMULATION ENDED.------------------

Test 2:

Time: 0  
Waiting clients: (0,2,4); (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (26,2,2); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (34,5,5); (35,7,5); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (47,4,5); (48,24,3); (49,21,3);  
Queue 1: closed  
Queue 2: closed  
Queue 3: closed  
Queue 4: closed  
Queue 5: closed  
  
Time: 1  
Waiting clients: (0,2,4); (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (26,2,2); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (34,5,5); (35,7,5); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (47,4,5); (48,24,3); (49,21,3);  
Queue 1: closed  
Queue 2: closed  
Queue 3: closed  
Queue 4: closed  
Queue 5: closed  
  
Time: 2  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (34,5,5); (35,7,5); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (47,4,5); (48,24,3); (49,21,3);  
Queue 1: (0,2,4);  
Queue 2: (26,2,2);  
Queue 3: closed  
Queue 4: closed  
Queue 5: closed  
  
Time: 3  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (34,5,5); (35,7,5); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (47,4,5); (48,24,3); (49,21,3);  
Queue 1: (0,2,3);  
Queue 2: (26,2,1);  
Queue 3: closed  
Queue 4: closed  
Queue 5: closed  
  
Time: 4  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (34,5,5); (35,7,5); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (0,2,2);  
Queue 2: closed  
Queue 3: (47,4,5);  
Queue 4: closed  
Queue 5: closed  
  
Time: 5  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (35,7,5); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (0,2,1);  
Queue 2: (34,5,5);  
Queue 3: (47,4,4);  
Queue 4: closed  
Queue 5: closed  
  
Time: 6  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (35,7,5); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: closed  
Queue 2: (34,5,4);  
Queue 3: (47,4,3);  
Queue 4: closed  
Queue 5: closed  
  
Time: 7  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (35,7,5);  
Queue 2: (34,5,3);  
Queue 3: (47,4,2);  
Queue 4: closed  
Queue 5: closed  
  
Time: 8  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (7,9,2); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (14,9,3); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (24,9,5); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (35,7,4);  
Queue 2: (34,5,2);  
Queue 3: (47,4,1);  
Queue 4: closed  
Queue 5: closed  
  
Time: 9  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (15,13,2); (16,10,5); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (35,7,3);  
Queue 2: (34,5,1);  
Queue 3: (24,9,5);  
Queue 4: (7,9,2);  
Queue 5: (14,9,3);  
  
Time: 10  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (8,23,5); (9,11,3); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (15,13,2); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (35,7,2);  
Queue 2: (16,10,5);  
Queue 3: (24,9,4);  
Queue 4: (7,9,1);  
Queue 5: (14,9,2);  
  
Time: 11  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (13,12,5); (15,13,2); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (25,12,5); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (39,12,1); (40,34,5); (41,12,4); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (35,7,1);  
Queue 2: (16,10,4);  
Queue 3: (24,9,3);  
Queue 4: (9,11,3);  
Queue 5: (14,9,1);  
  
Time: 12  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (15,13,2); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (13,12,5);  
Queue 2: (16,10,3);  
Queue 3: (24,9,2); (39,12,1);  
Queue 4: (9,11,2); (41,12,4);  
Queue 5: (25,12,5);  
  
Time: 13  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (13,12,4);  
Queue 2: (16,10,2); (15,13,2);  
Queue 3: (24,9,1); (39,12,1);  
Queue 4: (9,11,1); (41,12,4);  
Queue 5: (25,12,4);  
  
Time: 14  
Waiting clients: (1,17,2); (2,28,5); (3,15,3); (4,16,2); (5,18,2); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (21,15,5); (22,38,6); (23,32,1); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (43,15,3); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (13,12,3);  
Queue 2: (16,10,1); (15,13,2);  
Queue 3: (39,12,1);  
Queue 4: (41,12,4);  
Queue 5: (25,12,3);  
  
Time: 15  
Waiting clients: (1,17,2); (2,28,5); (4,16,2); (5,18,2); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (22,38,6); (23,32,1); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (13,12,2); (21,15,5);  
Queue 2: (15,13,2); (43,15,3);  
Queue 3: (3,15,3);  
Queue 4: (41,12,3);  
Queue 5: (25,12,2);  
  
Time: 16  
Waiting clients: (1,17,2); (2,28,5); (5,18,2); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (22,38,6); (23,32,1); (27,38,5); (28,17,1); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (13,12,1); (21,15,5);  
Queue 2: (15,13,1); (43,15,3);  
Queue 3: (3,15,2);  
Queue 4: (41,12,2);  
Queue 5: (25,12,1); (4,16,2);  
  
Time: 17  
Waiting clients: (2,28,5); (5,18,2); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (20,18,5); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (30,21,4); (31,25,4); (32,18,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (21,15,5);  
Queue 2: (43,15,3);  
Queue 3: (3,15,1); (1,17,2);  
Queue 4: (41,12,1); (28,17,1);  
Queue 5: (4,16,2);  
  
Time: 18  
Waiting clients: (2,28,5); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (30,21,4); (31,25,4); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (46,19,2); (48,24,3); (49,21,3);  
Queue 1: (21,15,4);  
Queue 2: (43,15,2); (32,18,2);  
Queue 3: (1,17,2);  
Queue 4: (28,17,1); (5,18,2);  
Queue 5: (4,16,1); (20,18,5);  
  
Time: 19  
Waiting clients: (2,28,5); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (30,21,4); (31,25,4); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (48,24,3); (49,21,3);  
Queue 1: (21,15,3);  
Queue 2: (43,15,1); (32,18,2);  
Queue 3: (1,17,1); (46,19,2);  
Queue 4: (5,18,2);  
Queue 5: (20,18,5);  
  
Time: 20  
Waiting clients: (2,28,5); (6,21,6); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (30,21,4); (31,25,4); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (48,24,3); (49,21,3);  
Queue 1: (21,15,2);  
Queue 2: (32,18,2);  
Queue 3: (46,19,2);  
Queue 4: (5,18,1);  
Queue 5: (20,18,4);  
  
Time: 21  
Waiting clients: (2,28,5); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (31,25,4); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (48,24,3);  
Queue 1: (21,15,1); (30,21,4);  
Queue 2: (32,18,1); (49,21,3);  
Queue 3: (46,19,1);  
Queue 4: (6,21,6);  
Queue 5: (20,18,3);  
  
Time: 22  
Waiting clients: (2,28,5); (8,23,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (31,25,4); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (44,23,5); (45,36,5); (48,24,3);  
Queue 1: (30,21,4);  
Queue 2: (49,21,3);  
Queue 3: closed  
Queue 4: (6,21,5);  
Queue 5: (20,18,2);  
  
Time: 23  
Waiting clients: (2,28,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (31,25,4); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5); (48,24,3);  
Queue 1: (30,21,3);  
Queue 2: (49,21,2);  
Queue 3: (8,23,5);  
Queue 4: (6,21,4);  
Queue 5: (20,18,1); (44,23,5);  
  
Time: 24  
Waiting clients: (2,28,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (31,25,4); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (30,21,2);  
Queue 2: (49,21,1); (48,24,3);  
Queue 3: (8,23,4);  
Queue 4: (6,21,3);  
Queue 5: (44,23,5);  
  
Time: 25  
Waiting clients: (2,28,5); (10,32,2); (11,37,3); (12,26,1); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (29,26,2); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (30,21,1); (31,25,4);  
Queue 2: (48,24,3);  
Queue 3: (8,23,3);  
Queue 4: (6,21,2);  
Queue 5: (44,23,4);  
  
Time: 26  
Waiting clients: (2,28,5); (10,32,2); (11,37,3); (17,35,5); (18,27,3); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (31,25,4);  
Queue 2: (48,24,2); (29,26,2);  
Queue 3: (8,23,2);  
Queue 4: (6,21,1); (12,26,1);  
Queue 5: (44,23,3);  
  
Time: 27  
Waiting clients: (2,28,5); (10,32,2); (11,37,3); (17,35,5); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (33,28,4); (36,28,2); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (31,25,3);  
Queue 2: (48,24,1); (29,26,2);  
Queue 3: (8,23,1); (18,27,3);  
Queue 4: (12,26,1);  
Queue 5: (44,23,2);  
  
Time: 28  
Waiting clients: (10,32,2); (11,37,3); (17,35,5); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (31,25,2); (36,28,2);  
Queue 2: (29,26,2);  
Queue 3: (18,27,3);  
Queue 4: (2,28,5);  
Queue 5: (44,23,1); (33,28,4);  
  
Time: 29  
Waiting clients: (10,32,2); (11,37,3); (17,35,5); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (31,25,1); (36,28,2);  
Queue 2: (29,26,1);  
Queue 3: (18,27,2);  
Queue 4: (2,28,4);  
Queue 5: (33,28,4);  
  
Time: 30  
Waiting clients: (10,32,2); (11,37,3); (17,35,5); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (36,28,2);  
Queue 2: closed  
Queue 3: (18,27,1);  
Queue 4: (2,28,3);  
Queue 5: (33,28,3);  
  
Time: 31  
Waiting clients: (10,32,2); (11,37,3); (17,35,5); (19,35,1); (22,38,6); (23,32,1); (27,38,5); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (36,28,1);  
Queue 2: closed  
Queue 3: closed  
Queue 4: (2,28,2);  
Queue 5: (33,28,2);  
  
Time: 32  
Waiting clients: (11,37,3); (17,35,5); (19,35,1); (22,38,6); (27,38,5); (37,33,2); (38,33,3); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: closed  
Queue 2: (10,32,2);  
Queue 3: (23,32,1);  
Queue 4: (2,28,1);  
Queue 5: (33,28,1);  
  
Time: 33  
Waiting clients: (11,37,3); (17,35,5); (19,35,1); (22,38,6); (27,38,5); (40,34,5); (42,35,6); (45,36,5);  
Queue 1: (37,33,2);  
Queue 2: (10,32,1);  
Queue 3: (38,33,3);  
Queue 4: closed  
Queue 5: closed  
  
Time: 34  
Waiting clients: (11,37,3); (17,35,5); (19,35,1); (22,38,6); (27,38,5); (42,35,6); (45,36,5);  
Queue 1: (37,33,1);  
Queue 2: closed  
Queue 3: (38,33,2);  
Queue 4: (40,34,5);  
Queue 5: closed  
  
Time: 35  
Waiting clients: (11,37,3); (22,38,6); (27,38,5); (45,36,5);  
Queue 1: (42,35,6);  
Queue 2: (17,35,5);  
Queue 3: (38,33,1);  
Queue 4: (40,34,4);  
Queue 5: (19,35,1);  
  
Time: 36  
Waiting clients: (11,37,3); (22,38,6); (27,38,5);  
Queue 1: (42,35,5);  
Queue 2: (17,35,4);  
Queue 3: (45,36,5);  
Queue 4: (40,34,3);  
Queue 5: closed  
  
Time: 37  
Waiting clients: (22,38,6); (27,38,5);  
Queue 1: (42,35,4);  
Queue 2: (17,35,3);  
Queue 3: (45,36,4);  
Queue 4: (40,34,2);  
Queue 5: (11,37,3);  
  
Time: 38  
Waiting clients:   
Queue 1: (42,35,3);  
Queue 2: (17,35,2); (27,38,5);  
Queue 3: (45,36,3);  
Queue 4: (40,34,1); (22,38,6);  
Queue 5: (11,37,2);  
  
Time: 39  
Waiting clients:   
Queue 1: (42,35,2);  
Queue 2: (17,35,1); (27,38,5);  
Queue 3: (45,36,2);  
Queue 4: (22,38,6);  
Queue 5: (11,37,1);  
  
Time: 40  
Waiting clients:   
Queue 1: (42,35,1);  
Queue 2: (27,38,5);  
Queue 3: (45,36,1);  
Queue 4: (22,38,5);  
Queue 5: closed  
  
Time: 41  
Waiting clients:   
Queue 1: closed  
Queue 2: (27,38,4);  
Queue 3: closed  
Queue 4: (22,38,4);  
Queue 5: closed  
  
Time: 42  
Waiting clients:   
Queue 1: closed  
Queue 2: (27,38,3);  
Queue 3: closed  
Queue 4: (22,38,3);  
Queue 5: closed  
  
Time: 43  
Waiting clients:   
Queue 1: closed  
Queue 2: (27,38,2);  
Queue 3: closed  
Queue 4: (22,38,2);  
Queue 5: closed  
  
Time: 44  
Waiting clients:   
Queue 1: closed  
Queue 2: (27,38,1);  
Queue 3: closed  
Queue 4: (22,38,1);  
Queue 5: closed  
  
Time: 45  
Waiting clients:   
Queue 1: closed  
Queue 2: closed  
Queue 3: closed  
Queue 4: closed  
Queue 5: closed  
  
Average Waiting Time: 1.375  
Average Service Time: 3.48  
Peak Time: 13 with 8 clients  
------------------SIMULATION ENDED.------------------

Test 3 is too big and is included only in the repository.

# Conclusions

This app was a great exercise in thread operations, syncronization and some more needed UI practice.

# Bibliography

* Programming Techniques – Lectures of prof. Ioan SALOMIE
* [www.stackoverflow.com](http://www.stackoverflow.com)
* www.wikipedia.org