University of Ottawa

Faculty of Engineering

SYS5110: Foundation of Modelling and Simulation

Project Report

Title: Optimizing Queuing Model and Simulation for Administration Department of a University using QuickPass System

Submitted to:

Dr. Md Mahmud Hasan

Submitted by:
SYS-Python Group 3

Vikram Khanzode

Radman Hosseinzadehsemnani

Zafar Mohammed

Kailash Suresh

Date: 12th December 2021

Optimizing Queueing Model and Simulation for Administration Department of a University using QuickPass System

Vikram Khanzode, Radman Hosseinzadehsemnani, Zafar Mohammed, Kailash Suresh

Faculty of Engineering

University of Ottawa

Abstract: Due to the COVID-19 restrictions, there have been many difficulties in organizing and processing the systems efficiently. In this project, we are focusing on the queuing model of the system giving importance to the precautions and considering them as the basic conditions for designing our queuing model for the administration department of the university, after referring to different research papers based on queuing models and going through various methodologies, we found QuickPass method as the most suitable one for this application. It has not only the flexibility of satisfying all the conditions set for the model but also has efficiently optimized them as required. We have used the M/M/1 queuing model in this approach which has the worldview of activity scanning as well as event scheduling.

Keywords: QuickPass System, M/M/1 queueing model, event scheduling.

1) Introduction:

The Traditional Implementation of the Queues uses the FIFO (First-In-First-Out) principle where all the items (or) people are kept according to the time they enter the queue and are processed (or) served one after the other in order. The administration department of the universities follow this principle for serving everyone.

The number of people to be served is inconsistent and difficult to predict. There have always been real-time issues with handling random and inconsistent customers in any system to be served. There are many theories, methodologies, and solutions proposed on handling such systems efficiently which helps to run the system smoother, like taking the statistical analysis of the previous simulations and utilizing this for dynamic system analysis (or) can be called as predictive analysis, the problem with such systems is that it is difficult to deal with state and time both in a single regression and other paper suggested to take the servers with different working efficiencies and allocate the queues accordingly, where they need previous data for predicting the customer arrival and allocating the number of service windows to be open for different scenarios.

Additionally, due to the ongoing COVID-19 pandemic, there is a severe need for maintaining social distancing. Hence, a QuickPass System is proposed to limit the number of customers that stay in the system at a particular time.

This study is organized in the following sections: Related Work section presents a discussion on some of the previous research work done on the QuickPass queueing systems. In the Initial System section, the simulation for M/M/1 queueing system by which the university administration department traditionally operates. In the QuickPass System section, the proposed QuickPass system is discussed and analyzed. Finally, in the Results sections, both the traditional and QuickPass systems were compared based on certain output metrics.

2) Related Work:

Several studies have been done on the related topic. In a 2010 study by Wang et al., QuickPass system was used to optimize the existing M/M/c queueing system which decreased the waiting times of customers in a bank [1]. Similarly, in a 2006 study done by Guo et al., a Multi-Objective QuickPass queueing model was used to optimize the existing system for two objective functions of minimizing average waiting times of all the customers and minimizing the average free time of the customers entering the QuickPass queue [2]. Finally, in the paper by Li and Tao (2014), the authors have constructed QuickPass system and designed the optimal operation strategy to minimize the time cost of amusement park tourists [3].

3) Program Description:

The simulation of the traditional and proposed QuickPass systems was done in Python programming language using data analysis packages like NumPy, Pandas and Matplotlib. The simulation was done using a class named Simulation which has methods like advance_time(), handle_arrival_event(), handle_depart_event() and generate_return(). The interarrival times and service times for customers were generated using generators generate_interarrival() and generate_service(), respectively.

```
# system parameters
mean_arrival_num = 1.5   # no. of arrivals per minute
mean_service_num = 1  # no. of services completed per minute
threshold_control = 10  # max customers in queue while QuickPass is in action
quick_switch = 1   # 0 for regular queue | 1 for QuickPass in action
seed = 34  # random seed
np.random.seed(seed)
```

Fig. 1 System Parameters set in the program

Figure 1 shows the system parameters set in the program. When the “quick_switch” parameter is set as 0, the program will run the simulation for the traditional M/M/1 queuing system and write its event table data into ‘event_table_Regular.csv’ file. Similarly, when the “quick_switch” parameter is set as 1, the program will run the simulation for the proposed QuickPass system and write its event table data into ‘event_table_QuickPass.csv’. Figure 2 shows the execution for the same.

```
if quick_switch == 1:
    et.to_csv('event_table_QuickPass.csv')
else:
    et.to_csv('event_table_Regular.csv')
```

Fig. 2 Generation of simulation Event Tables

Figure 3 describes the time advancing algorithm for updating the simulation clock, customer arrivals, customer departures and waiting times of the customers.

```
# running the simulation for the defined period
while s.clock < 240.0:
    previous_depart = s.t_depart

    s.advance_time()

    queue_wait.append(max(0, previous_depart - s.t_arrival))
    quick_wait_data.append(s.quick_wait)
    regular_wait_data.append(s.regular_wait)
    clock_data.append(float(s.clock))
    num_in_sys_data.append(float(s.num_in_system))
    arrival_time_data.append(float(s.t_arrival))
    departure_time_data.append(float(s.t_depart))
    arrival_no_data.append(float(s.num_arrivals))
    departure_no_data.append(float(s.num_departs))
```

Fig 3. Time advancing for 4 hours (240 minutes).

In Figure 3, the wait in queue parameter has been derived through the following equation:

$$w_j = \mathrm{Max}(0, \Delta_j)$$

$$\Delta_j = d_{j-1} - a_j$$

Where,

$j = 1,2,3,\ldots =$ customer number

$d =$ departure time

$a =$ arrival time

System time for both the QuickPass and regular customers is calculated as follows:

$$t_j = d_j - a_j$$

4) Initial System:

For Simulating the Initial System, we have used the event scheduling worldview which is also known as time-advancing worldview. The initial system uses M/M/1 queuing model with the interarrival times and service times exponentially distributed. There is a single service station. The arrival rate for the customers is assumed to be 2.5 customers per minute whereas the service rate is assumed to be 2 customers per minute. This queuing system is simulated over 240 minutes (4hours). Figure 4 shows the algorithm for simulating M/M/1 queue by time advancing. The simulation is done using Python.

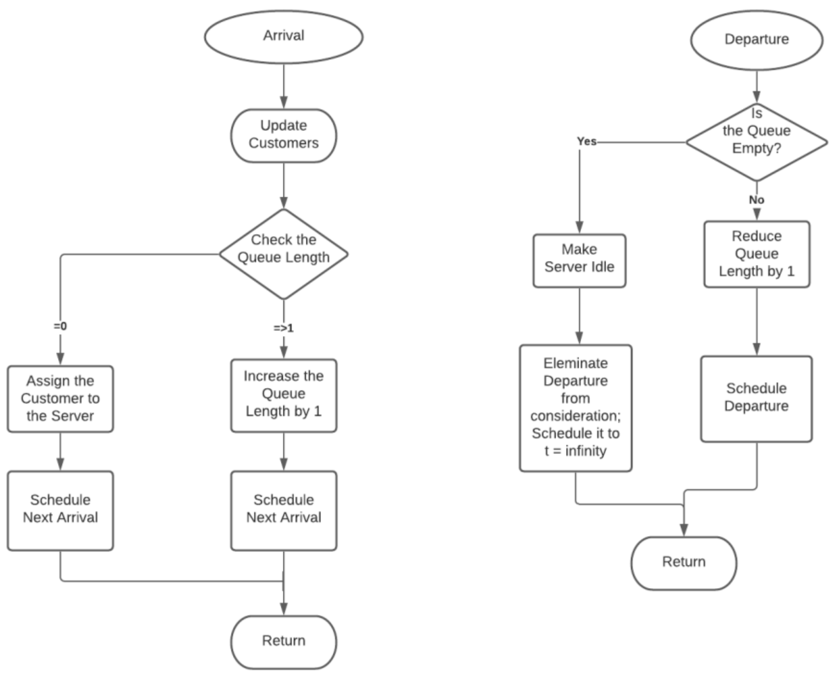

Fig. 4 The Flowcharts for simulating Arrival and Departure events in the Traditional M/M/1 queueing system using Event Scheduling worldview.

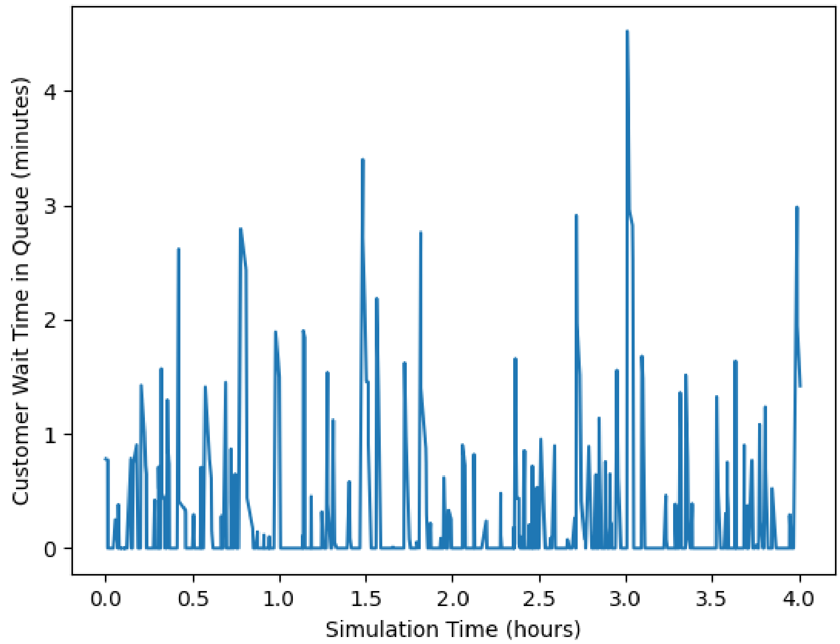

Fig. 5 Customer waiting time in the initial system measured over 4 hours.

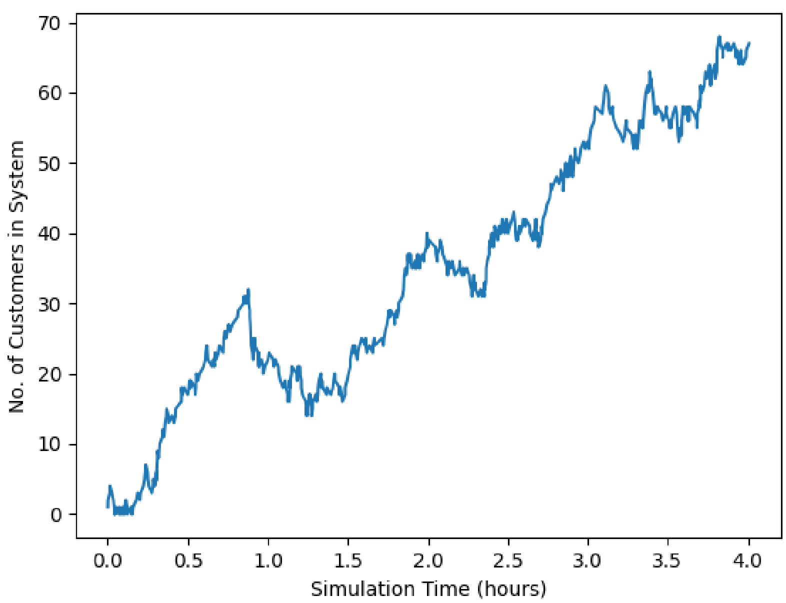

Fig. 6 Total number of customers in the system in the initial system measured over 4 hours.

After simulating the initial system, we find that the total number of customers served over the period of 4 hours is 258 with an average waiting time of 1.749501 minutes per customer.

5) QuickPass System:

For Simulating the proposed QuickPass System, we have used the combination of event scheduling and activity scanning worldview. Like the initial system, QuickPass system is simulated with the interarrival times and service times both exponentially distributed, single service station and arrival rate for the customers is assumed to be 2.5 customers per minute whereas the service rate is assumed to be 2 customers per minute. This queuing system is also simulated over 240 minutes (4hours). However, the capacity of the system is limited to 10 customers at a time. This means that whenever a customer arrives after the system has already filled to the capacity, they are given a QuickPass ticket and a returning time in the future when they are given priority over the other waiting customers. The returning time of a QuickPass ticket-holding customer is scheduled using the formula given below:

$$\text{Return time} = t + t_0 + [Lq(t)/\mu]$$

Where,

t = time when customer arrives

t_0 = time lag

$Lq(t)$ = length of queue at time t

μ = service rate

Figures 7 and 8 show the flowchart for scheduling arrival and departure events in the QuickPass system.

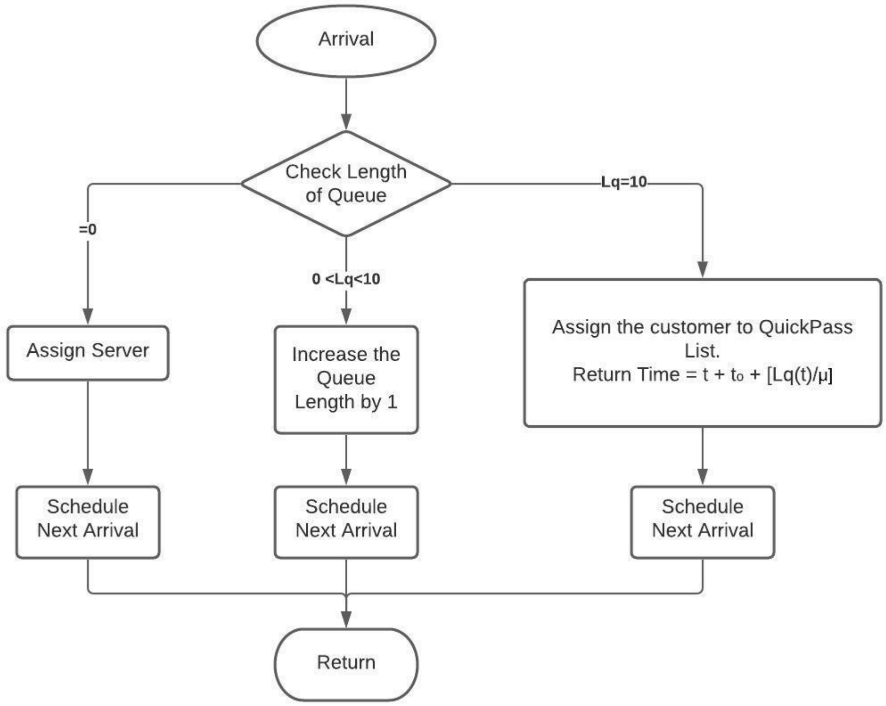

Fig. 7 The Flowchart for simulating Arrival event in a QuickPass queueing system using Event Scheduling worldview.

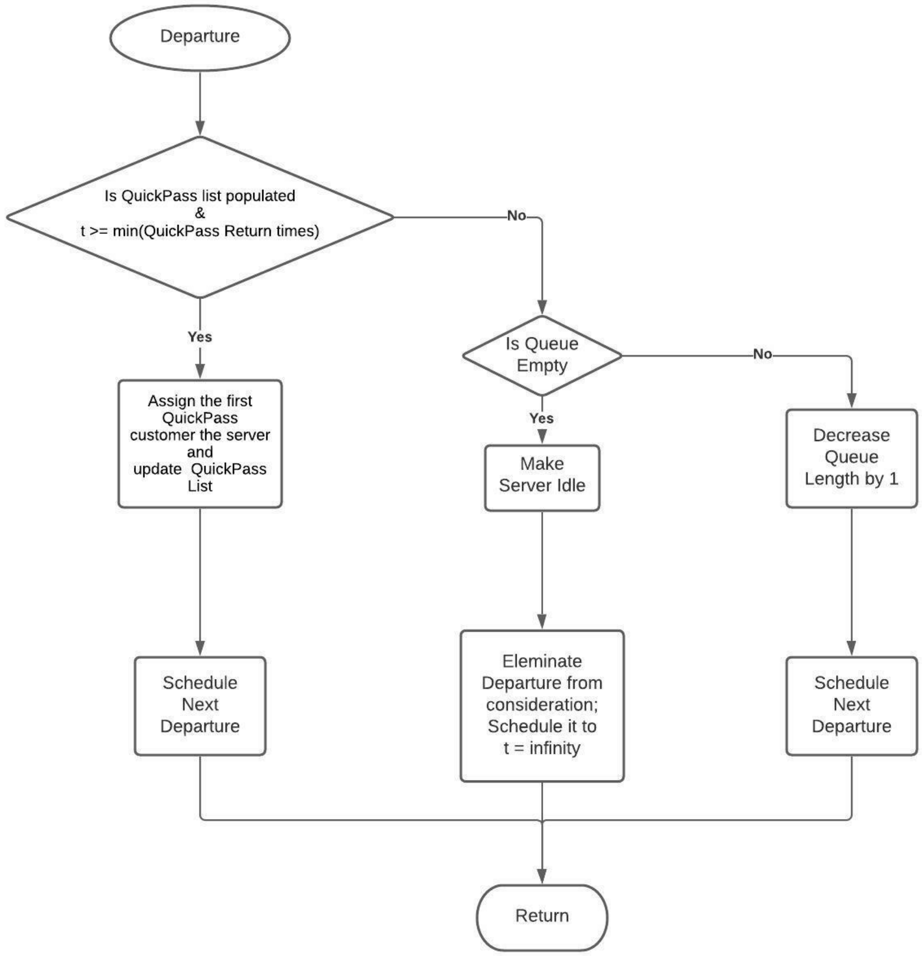

Fig. 8 The Flowchart for simulating Departure event in the QuickPass queueing system using Event Scheduling worldview.

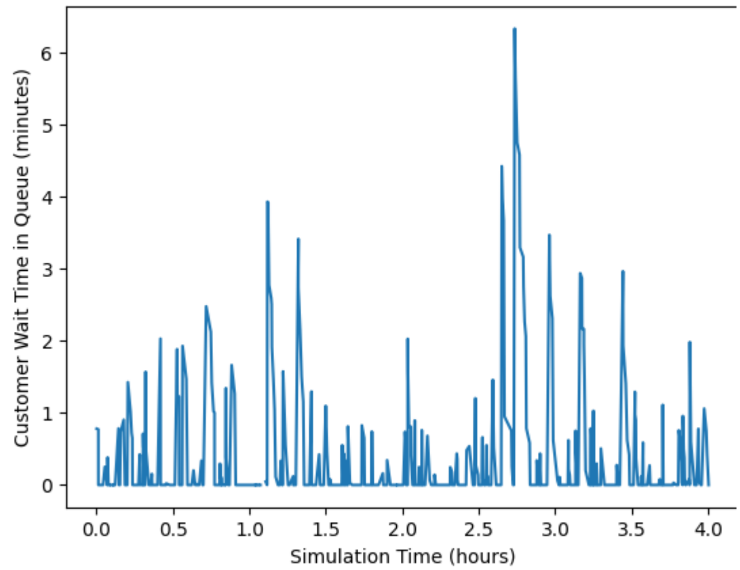

Fig. 9 Customer waiting time in the QuickPass system measured over 4 hours.

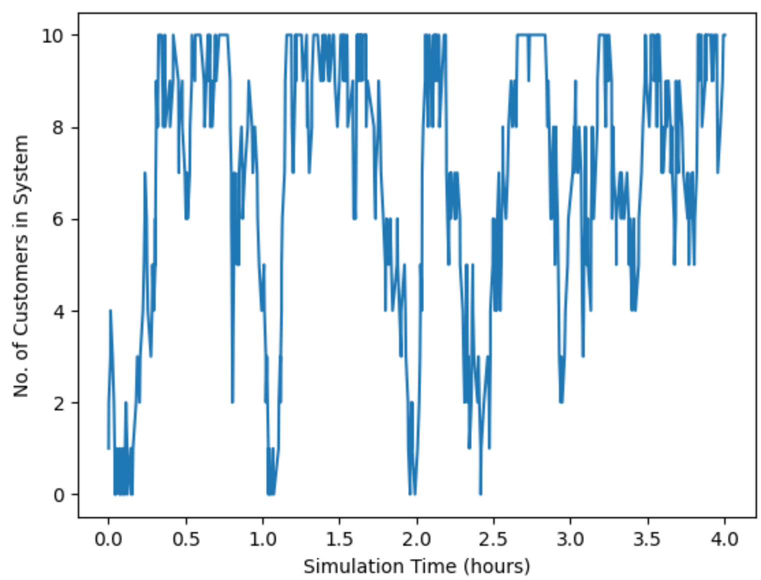

Fig. 10 Total number of customers in the system in the QuickPass system measured over 4 hours.

After simulating the QuickPass system, we find that the total number of customers served over the period of 4 hours is 254 with an average waiting time of 1.611896 minutes per customer. It was found that 73 customers were issued the QuickPass ticket of which 61 customers were served.

6) Results:

In this study, we have simulated and compared traditional M/M/1 queuing system and a QuickPass system using Python. For both the systems, the input parameters were assumed to 2.5 customers per minute for arrival rate and 2 customers per minute for service rate. Figure 11 depicts the comparison between the number of customers in the system for both systems measured over 4 hours' time.

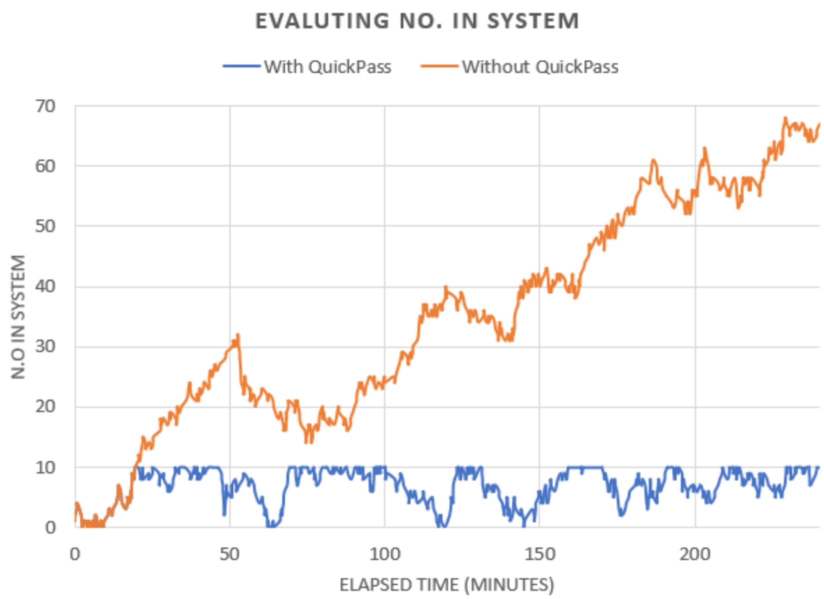

Fig. 11 Comparison between the number of customers in the system in both systems measured over 4 hours.

Figures 12 & 13 show the python console outputs for the traditional and QuickPass system simulations, respectively.

```
Number of customers departed: 258
Average time in system: 1.749501
```

Fig. 12 Console output for the Traditional M/M/1 queuing system simulation in Python.

```
Number of customers departed: 254
Average time in system: 1.611896

- QuickPass info -
Average wait time in QuickPass: 8.103054
Average overall wait time: 3.183973
No. sent to QuickPass queue: 73
No. served from QuickPass queue: 61
```

Fig. 13 Console output for the proposed QuickPass queuing system simulation in Python.

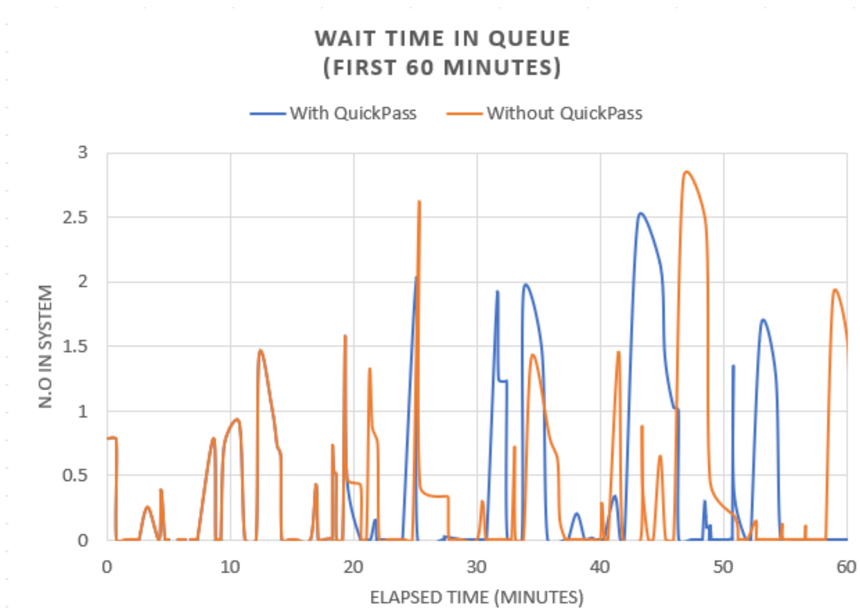

Fig. 14 Comparison between the customer waiting time in queue for both the models measured in the first hour.

Table 1 shows a comparison of waiting times of the customer under the traditional and proposed model:

Model	Average Time in System (minutes)	Average Time in Queue (minutes)	Average time in system for QuickPass (minutes)
Regular	1.74950115816686	0.27117477396	N/A
QuickPass	1.61189601829077	0.35763236081	8.10305414804

7) Conclusion:

The project concentrates on designing a stochastic model for the queuing system of the university administration department. There are different papers and methodologies proposed for the queuing systems, but there has not been any queuing model proposed specifically keeping the COVID-19 restrictions as the conditions for designing these models. There are different measures of COVID-19 like having a limited number of people in the queue, serving the maximum number of customers possible and having less waiting time for each customer. The model we have executed was potentially suitable, as it satisfied the conditions and was efficient enough for working with different capacities of the other applications if executed. This model has boosted the performance in a very economical way by having the limit of customers yet giving every customer equal priority of service and serving a maximum number of customers, it has held up different conditions and proved to be one of the best executable models for the university administration.

References:

1. Y. Wang, C. Qian, and J. Cao, "Optimized M/M/c model and simulation for bank queuing system," 2010 IEEE International Conference on Software Engineering and Service Sciences, 2010, pp. 474- 477, doi: 10.1109/ICSESS.2010.5552328.
2. Qiang Guo, Jianguo Liu, Xingwen Chen. “Optimization Model and Simulation of the Queuing System with QuickPass,” in Proc. the 6th World Congress on Intelligent Control and Automation, Dalian, China, Jun. 2006, pp. 1401-1403
3. M. Li, L. Tao, “Modelling and Simulation for Queuing System using QuickPass”, Advanced Materials Research Vols. 926-930 (2014) pp 3088-3091, © (2014) Trans Tech Publications, Switzerland doi: 10.4028/www.scientific.net/AMR.926-930.3088
4. J. Banks et al (2010) Discrete-Event System Simulation, Fifth Edition, Pearson Prentice Hall, Upper Saddle River, NJ 07458, USA.
5. “Queuing System Discrete Event Simulation in Python (Event-scheduling)”. YouTube, uploaded by Paul Grogan, 29 October 2016, https://www.youtube.com/watch?v=oJyf8Q0KLRY