UNIVERSITY OF KARACHI

BSCS-507 OPERATIONAL RESEARCH

BSCS SECTION B (EVENING)

GROUP MEMBERS:

ARHAM SHARIF EB21102022

HAMZA ALAM HASHMI EB21102031

MUHAMMAD RIAZ AKRAM EB21102077

SUBMITTED TO:

MISS. SHAISTA RAEES

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF KARACHI

Code Documentation

1. To run this project you need to have XCode installed on your macos (not for windows)
2. You can access the files in the project in windows to understand the source code
3. The Main Logic Files in this Project are under the following folders
    a. OperationalResearchProject/ViewModel
        i. QueuingViewModel.swift
        ii. RandomViewModel.swift
    b. OperationalResearchProject/Model/
        i. FitTest.swift
        ii. Result.swift
4. Below is the documentation of each file

QueuingViewModel.swift

Overview

The `QueuingViewModel.swift` file is part of the Operational Research project. This Swift class, `QueuingViewModel`, is responsible for handling various queuing models, conducting calculations, and performing goodness-of-fit tests. The class captures user inputs, calculates queuing metrics, and validates inputs to ensure accurate results.

Class Structure

The `QueuingViewModel` class includes:

1. **Published Properties**:

These properties store user inputs, calculation results, and other data used for displaying information in the UI.

2. **Calculation Methods**:

Private methods that perform calculations for different queuing models, including M/M/C, M/M/G, and G/G/C models. These methods compute utilization, average number of customers, average time metrics, and more.

3. **Validation Method**:

Ensures that user inputs are valid before performing calculations.

4. **Factorial Calculation Method**:

Computes the factorial of an integer using recursive logic.

Key Responsibilities

1. **Input Handling**:

The class captures user inputs for server type, number of servers, arrival rates, and service rates.

2. **Calculations**:

Methods calculate queuing metrics for different models, including utilization, average customers, and average times.

3. **Goodness-of-Fit Test**:

The `calculateGoodFitTest` method performs a goodness-of-fit test using observed and expected frequencies.

4. **Validation**:

The `isValidate` method checks if required input fields are filled before calculations.

Usage

1. Users provide input values based on the selected queuing model and its parameters.
2. The `calculateResults` method is invoked to initiate calculations.
3. Calculation methods appropriate to the selected server type are executed.
4. Results or error messages are presented to the user.

SimulationViewModel.swift

Overview

The `SimulationViewModel.swift` file is part of the Operational Research project. This Swift class, `RandomViewModel`, handles simulation-related calculations and provides methods to generate random values for simulation purposes.

Class Structure

The `RandomViewModel` class consists of:

1. **Published Properties**:

These properties store user input values, cumulative lists, arrival and service times, and other simulation data.

2. **Calculation and Simulation Methods**:

Methods that perform calculations and generate simulation data, including cumulative probabilities, inter-arrival times, arrival times, service times, and more.

3. **Helper Functions**:

Private functions for finding inter-arrival indices, calculating cumulative frequencies, and calculating factorials.

Key Responsibilities

1. **Input Handling**:

User inputs for lambda (λ) and meo (μ) are captured for simulation.

2. **Calculation and Simulation**:

The class calculates cumulative probabilities, inter-arrival times, arrival times, service times, and other simulation-related data.

Usage

1. Users provide lambda (λ) and meo (μ) values for simulation.
2. The `calculateValues` method initiates calculations and simulation.
3. Simulation data is computed and stored in various arrays.

Result.swift (QueuingResults Struct Documentation)

Overview:

The `QueuingResults` structure holds key metrics from queuing model calculations, including server utilization, average customers, and time metrics.

Properties:

- `utilization` (Double): Server utilization (P).
- `avgNumCustomersSystem` (Double): Average customers in the system (L).
- `avgNumCustomersQueue` (Double): Average customers in the queue (Lq).
- `avgTimeInSystem` (Double): Average time in the system (W).
- `avgTimeInQueue` (Double): Average time in the queue (Wq).

Method:

- `toText() -> String`: Converts metrics to formatted text.

FitTest.swift (FitTest Struct Documentation)

Overview:

The `FitTest` structure represents goodness-of-fit test results, including chi-square value, significance level, critical value, and hypothesis assessment.

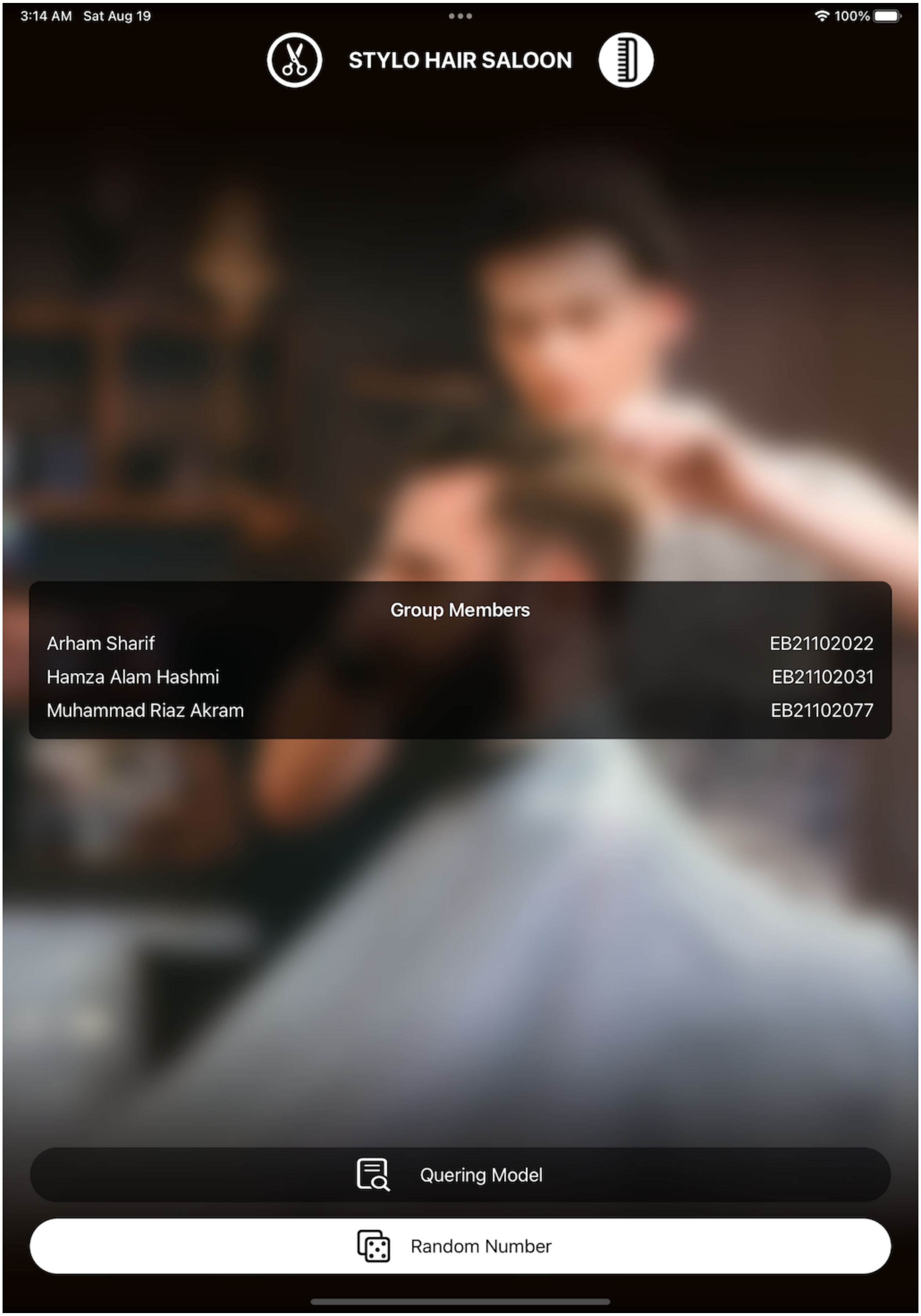
Properties:

- `chiSquare` (Double): Chi-square value from the test.
- `significanceLevel` (Double): Chosen significance level.
- `criticalValue` (Double): Critical value for significance level.

Method:

- `toResult() -> String`: Converts test results to formatted text.

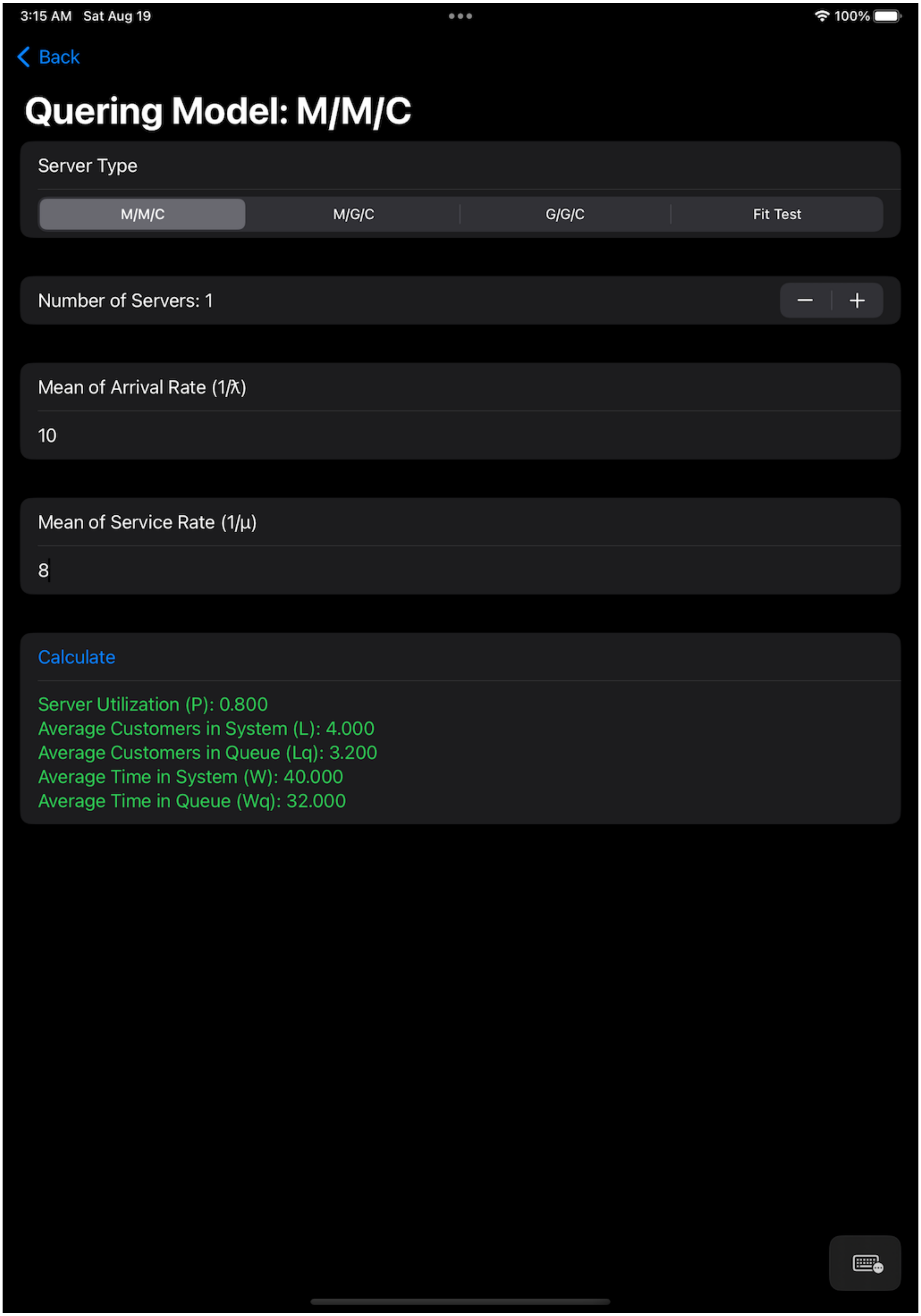
Outputs ScreenShots



STYLO HAIR SALOON

Group Members

Arham Sharif	EB21102022
Hamza Alam Hashmi	EB21102031
Muhammad Riaz Akram	EB21102077



Quering Model: M/M/C

Server Type: M/M/C

Number of Servers: 1

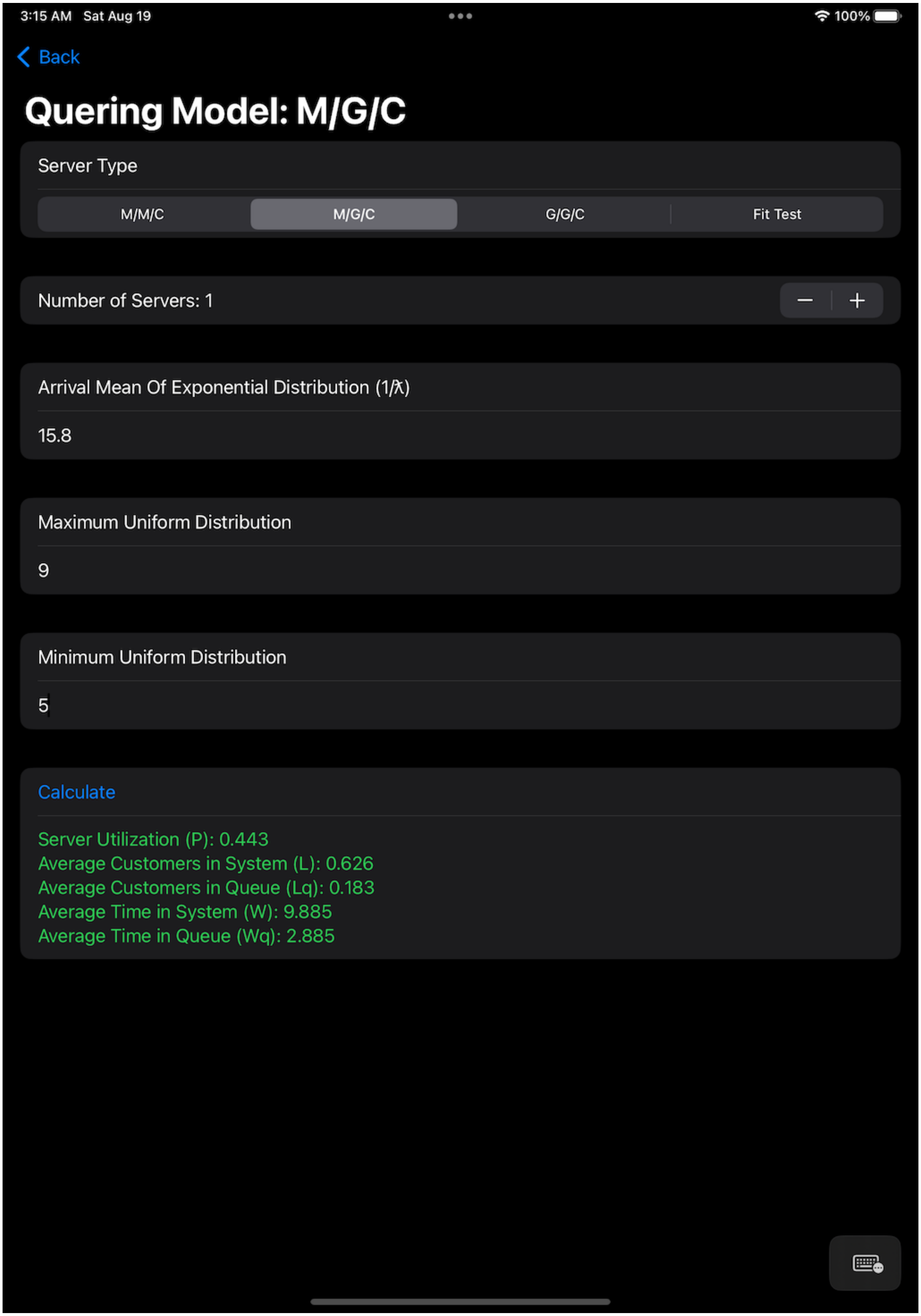
Mean of Arrival Rate (1/ƛ)

10

Mean of Service Rate (1/μ)

8

Server Utilization (P): 0.800
Average Customers in System (L): 4.000
Average Customers in Queue (Lq): 3.200
Average Time in System (W): 40.000
Average Time in Queue (Wq): 32.000



Quering Model: M/G/C

Server Type

Number of Servers: 1

Arrival Mean Of Exponential Distribution (1/λ)

15.8

Maximum Uniform Distribution

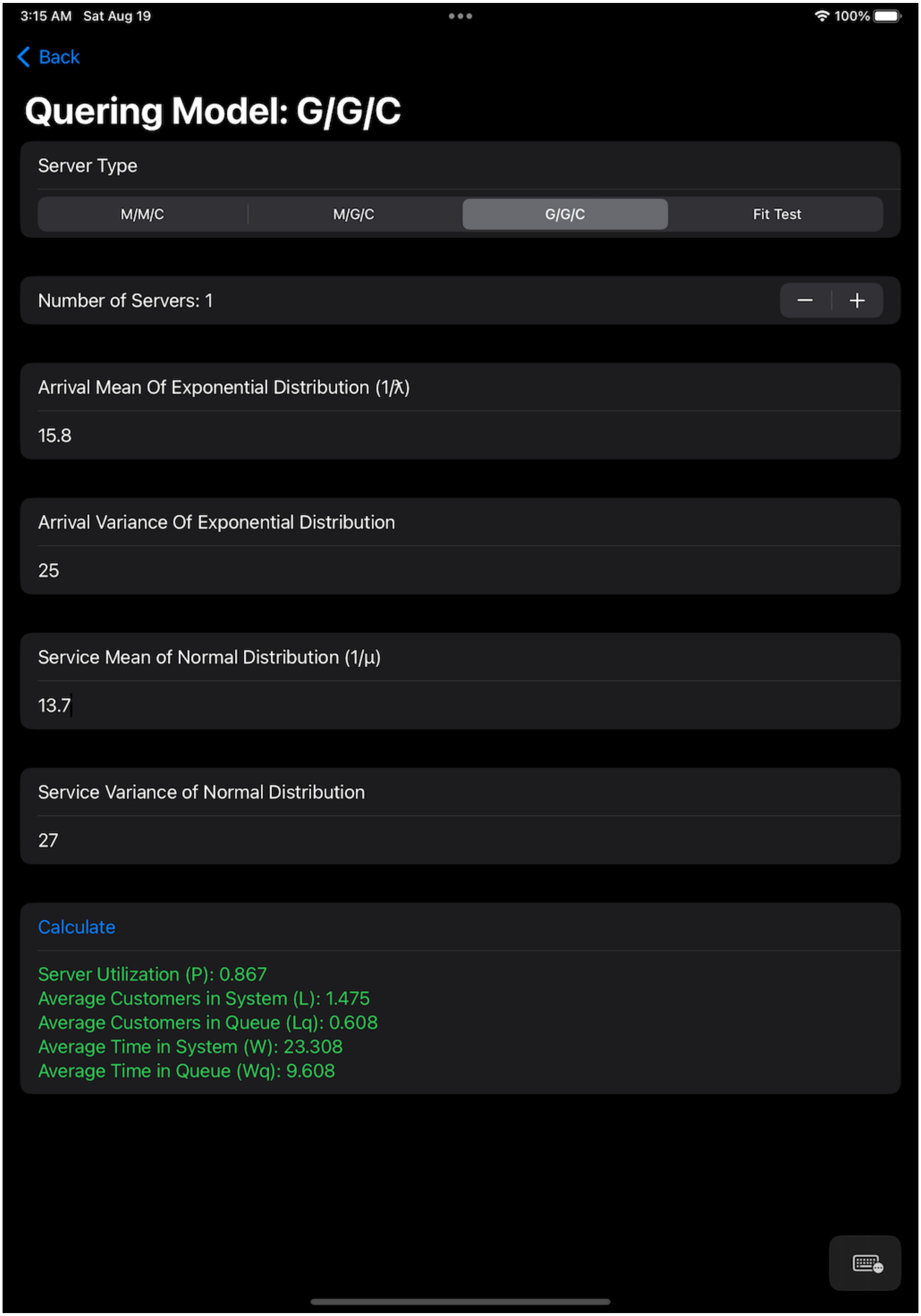
9

Minimum Uniform Distribution

5

Calculate

Server Utilization (P): 0.443
Average Customers in System (L): 0.626
Average Customers in Queue (Lq): 0.183
Average Time in System (W): 9.885
Average Time in Queue (Wq): 2.885



3:15 AM Sat Aug 19

100%

Back

Quering Model: G/G/C

Server Type

M/M/C | M/G/C | G/G/C | Fit Test

Number of Servers: 1

− +

Arrival Mean Of Exponential Distribution (1/ƛ)

15.8

Arrival Variance Of Exponential Distribution

25

Service Mean of Normal Distribution (1/µ)

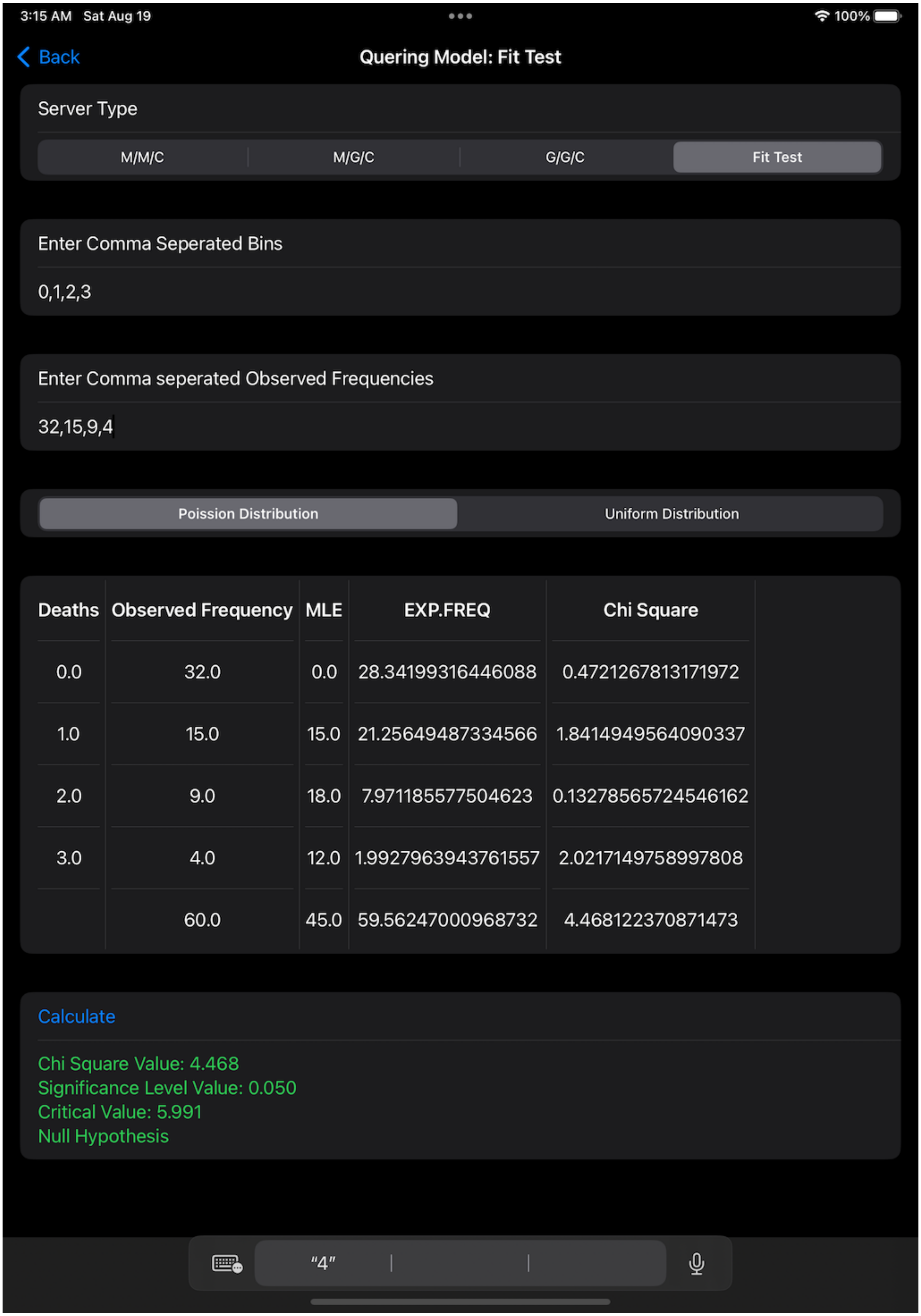
13.7

Service Variance of Normal Distribution

27

Calculate

Server Utilization (P): 0.867
Average Customers in System (L): 1.475
Average Customers in Queue (Lq): 0.608
Average Time in System (W): 23.308
Average Time in Queue (Wq): 9.608



Quering Model: Fit Test

Server Type: Fit Test

Enter Comma Seperated Bins

0,1,2,3

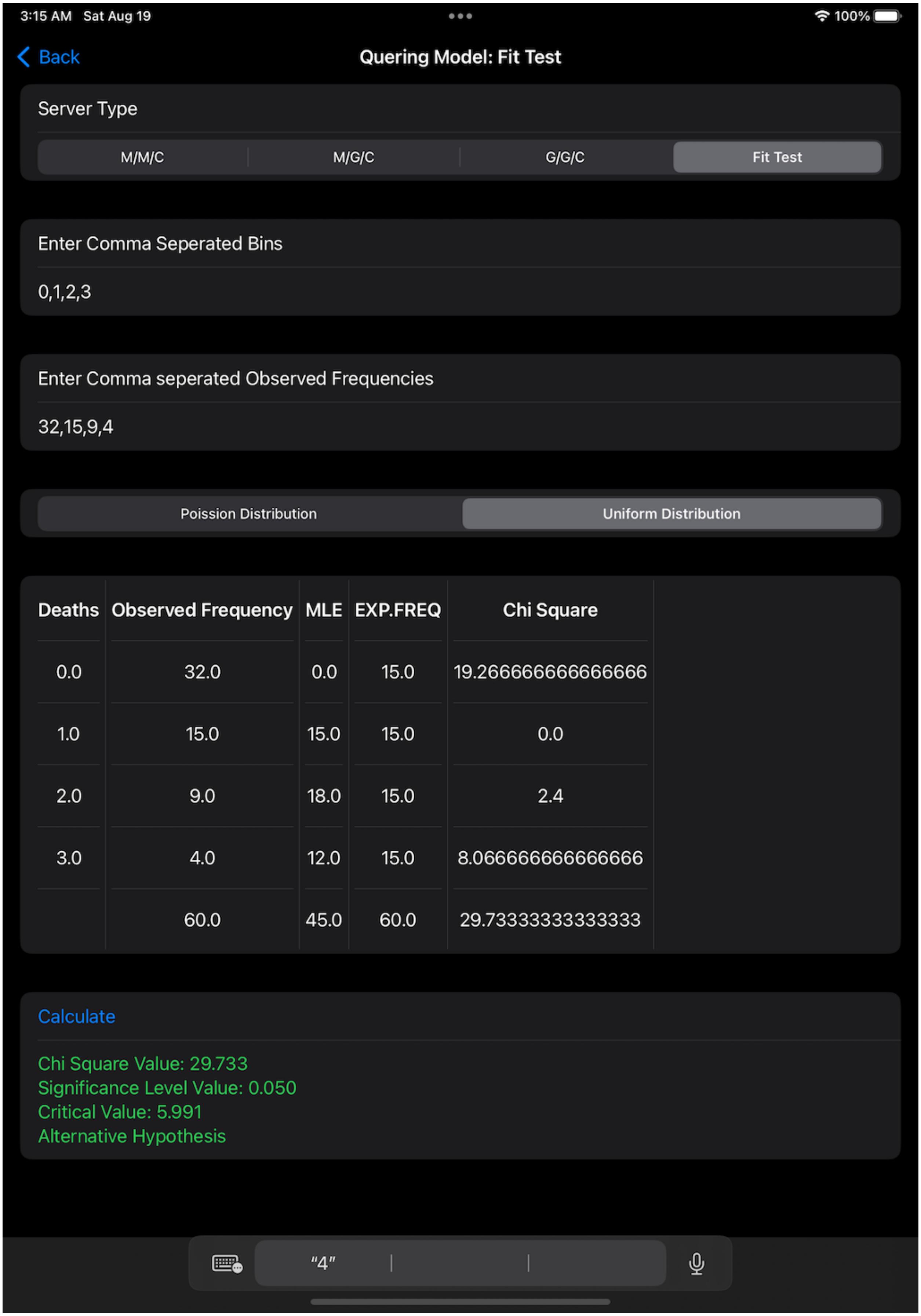
Enter Comma seperated Observed Frequencies

32,15,9,4

Poission Distribution

Deaths	Observed Frequency	MLE	EXP.FREQ	Chi Square
0.0	32.0	0.0	28.34199316446088	0.4721267813171972
1.0	15.0	15.0	21.25649487334566	1.8414949564090337
2.0	9.0	18.0	7.971185577504623	0.13278565724546162
3.0	4.0	12.0	1.9927963943761557	2.0217149758997808
	60.0	45.0	59.56247000968732	4.468122370871473

Chi Square Value: 4.468
Significance Level Value: 0.050
Critical Value: 5.991
Null Hypothesis



Quering Model: Fit Test

Server Type: Fit Test

Enter Comma Seperated Bins

0,1,2,3

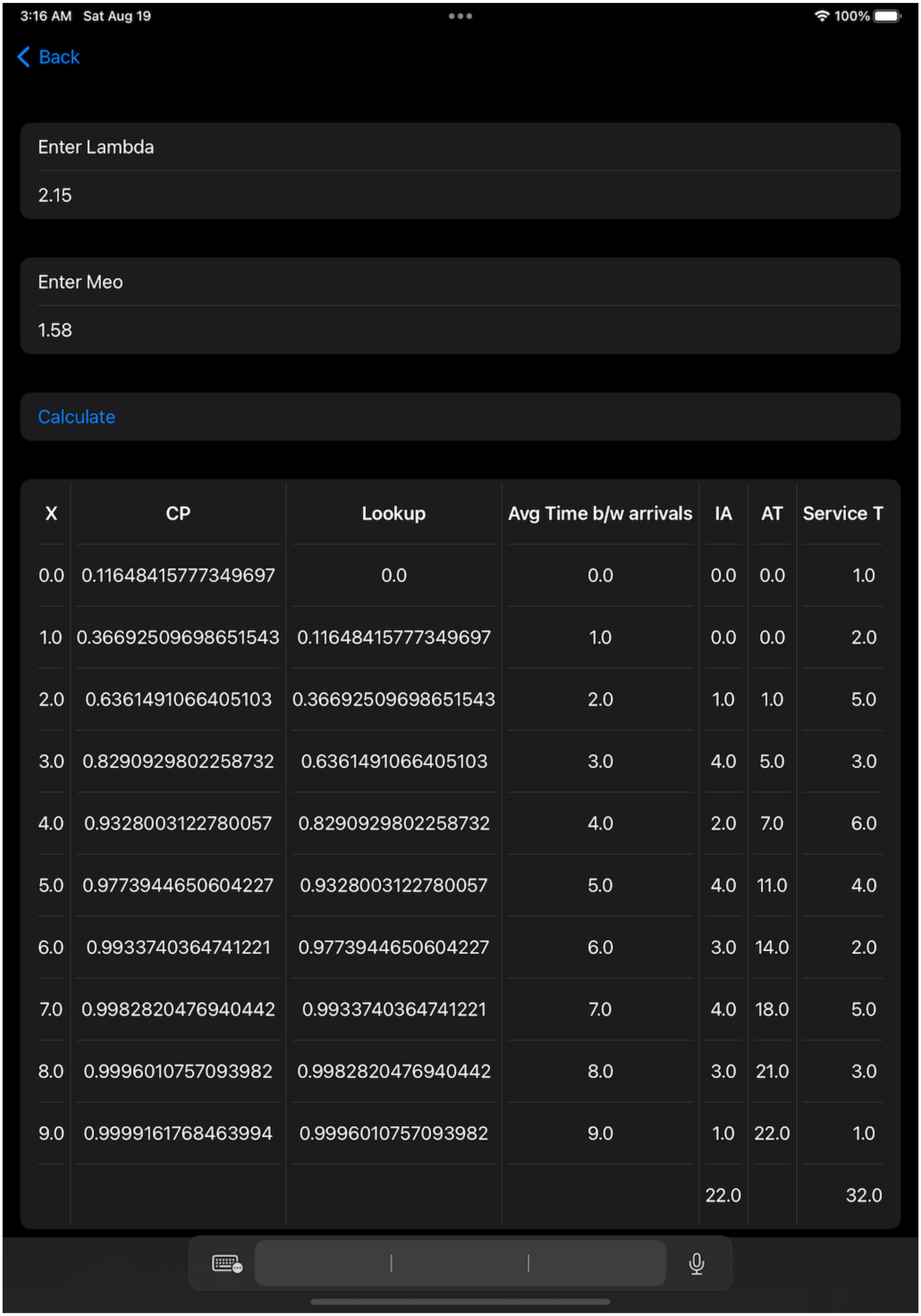
Enter Comma seperated Observed Frequencies

32,15,9,4

Uniform Distribution

Deaths	Observed Frequency	MLE	EXP.FREQ	Chi Square
0.0	32.0	0.0	15.0	19.266666666666666
1.0	15.0	15.0	15.0	0.0
2.0	9.0	18.0	15.0	2.4
3.0	4.0	12.0	15.0	8.066666666666666
	60.0	45.0	60.0	29.733333333333333

Chi Square Value: 29.733
Significance Level Value: 0.050
Critical Value: 5.991
Alternative Hypothesis



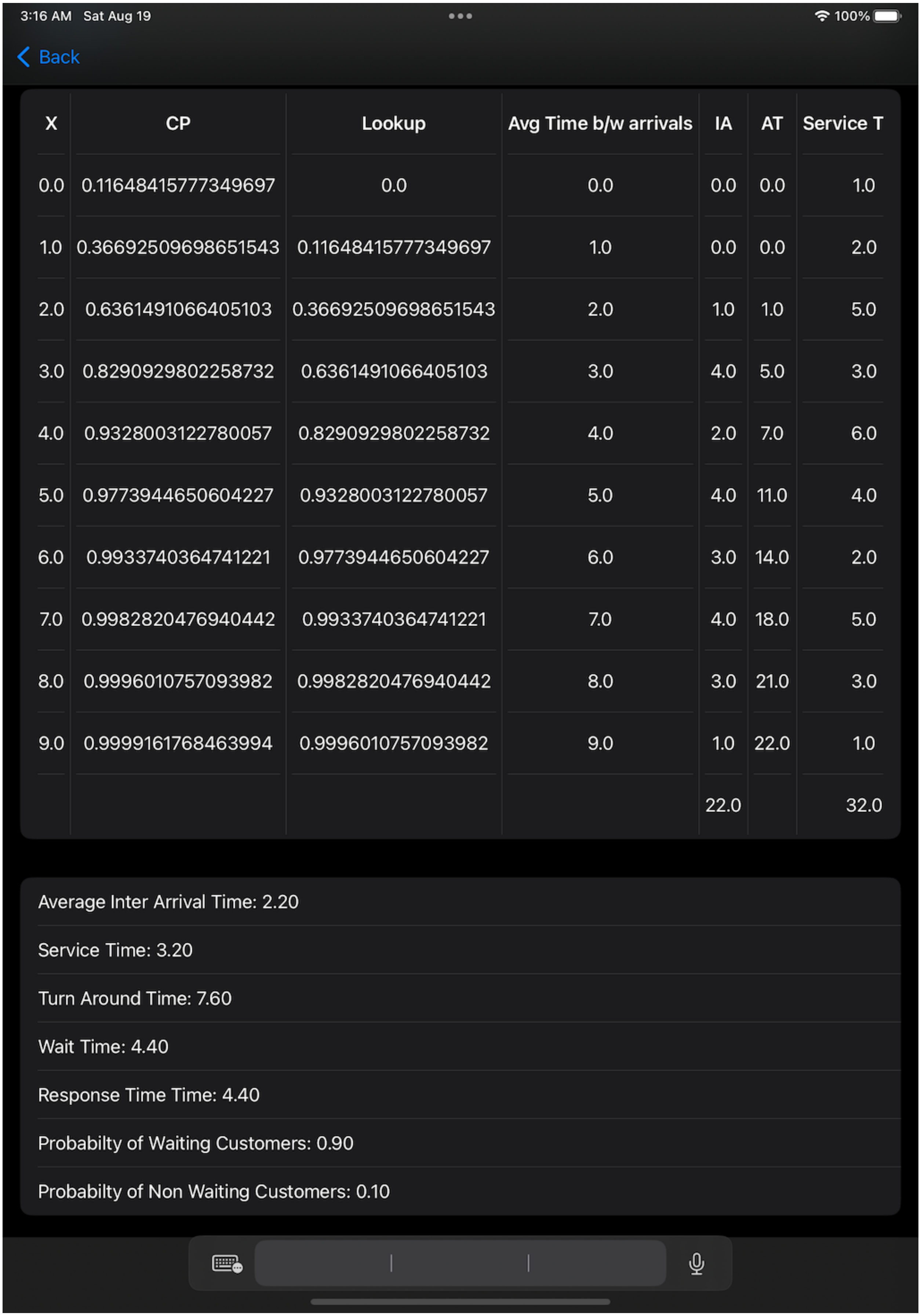
Enter Lambda

2.15

Enter Meo

1.58

X	CP	Lookup	Avg Time b/w arrivals	IA	AT	Service T
0.0	0.11648415777349697	0.0	0.0	0.0	0.0	1.0
1.0	0.36692509698651543	0.11648415777349697	1.0	0.0	0.0	2.0
2.0	0.6361491066405103	0.36692509698651543	2.0	1.0	1.0	5.0
3.0	0.8290929802258732	0.6361491066405103	3.0	4.0	5.0	3.0
4.0	0.9328003122780057	0.8290929802258732	4.0	2.0	7.0	6.0
5.0	0.9773944650604227	0.9328003122780057	5.0	4.0	11.0	4.0
6.0	0.9933740364741221	0.9773944650604227	6.0	3.0	14.0	2.0
7.0	0.9982820476940442	0.9933740364741221	7.0	4.0	18.0	5.0
8.0	0.9996010757093982	0.9982820476940442	8.0	3.0	21.0	3.0
9.0	0.9999161768463994	0.9996010757093982	9.0	1.0	22.0	1.0
				22.0		32.0



X	CP	Lookup	Avg Time b/w arrivals	IA	AT	Service T
0.0	0.11648415777349697	0.0	0.0	0.0	0.0	1.0
1.0	0.36692509698651543	0.11648415777349697	1.0	0.0	0.0	2.0
2.0	0.6361491066405103	0.36692509698651543	2.0	1.0	1.0	5.0
3.0	0.8290929802258732	0.6361491066405103	3.0	4.0	5.0	3.0
4.0	0.9328003122780057	0.8290929802258732	4.0	2.0	7.0	6.0
5.0	0.9773944650604227	0.9328003122780057	5.0	4.0	11.0	4.0
6.0	0.9933740364741221	0.9773944650604227	6.0	3.0	14.0	2.0
7.0	0.9982820476940442	0.9933740364741221	7.0	4.0	18.0	5.0
8.0	0.9996010757093982	0.9982820476940442	8.0	3.0	21.0	3.0
9.0	0.9999161768463994	0.9996010757093982	9.0	1.0	22.0	1.0
				22.0		32.0

Average Inter Arrival Time: 2.20

Service Time: 3.20

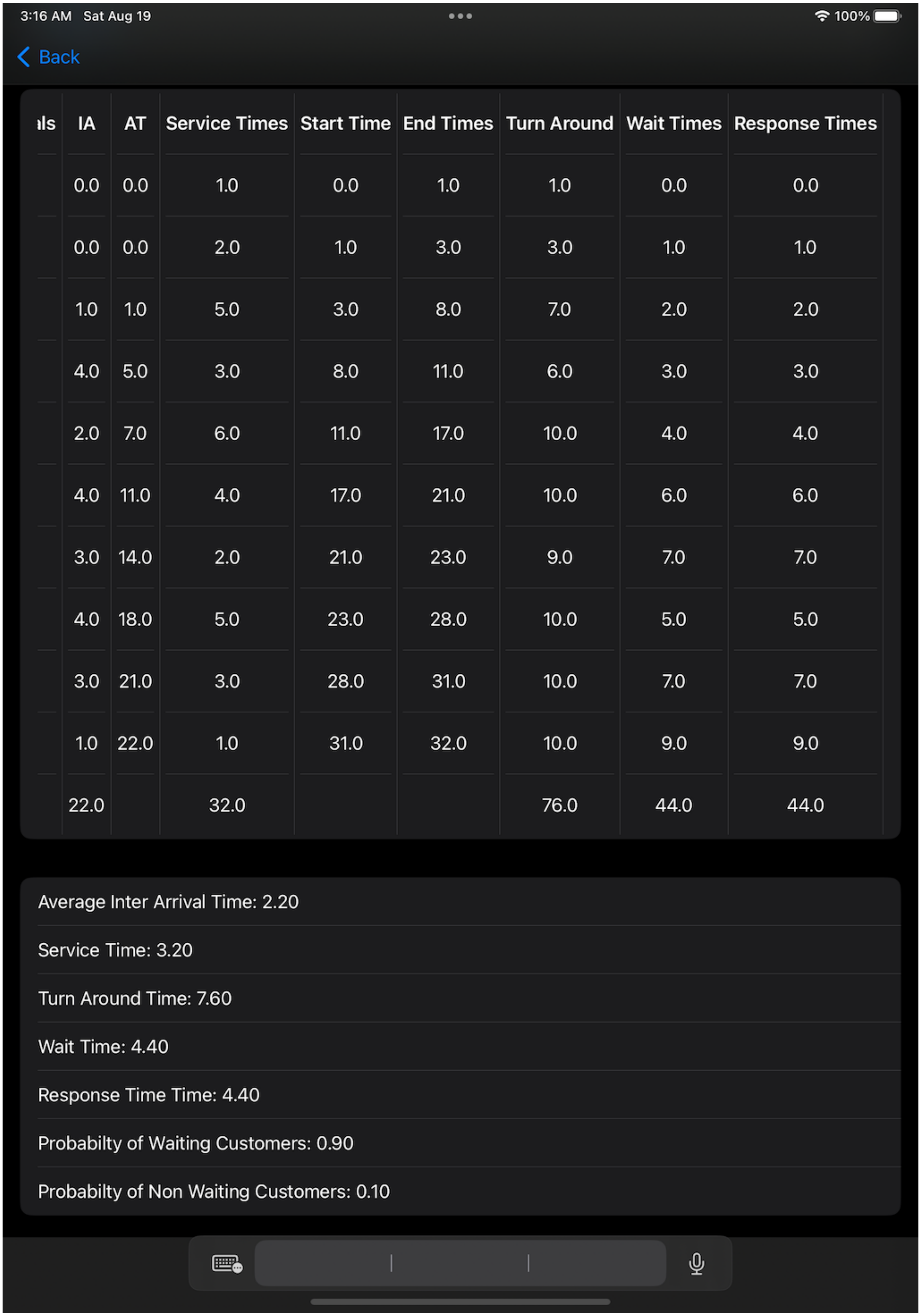
Turn Around Time: 7.60

Wait Time: 4.40

Response Time Time: 4.40

Probabilty of Waiting Customers: 0.90

Probabilty of Non Waiting Customers: 0.10



ıls	IA	AT	Service Times	Start Time	End Times	Turn Around	Wait Times	Response Times
	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
	0.0	0.0	2.0	1.0	3.0	3.0	1.0	1.0
	1.0	1.0	5.0	3.0	8.0	7.0	2.0	2.0
	4.0	5.0	3.0	8.0	11.0	6.0	3.0	3.0
	2.0	7.0	6.0	11.0	17.0	10.0	4.0	4.0
	4.0	11.0	4.0	17.0	21.0	10.0	6.0	6.0
	3.0	14.0	2.0	21.0	23.0	9.0	7.0	7.0
	4.0	18.0	5.0	23.0	28.0	10.0	5.0	5.0
	3.0	21.0	3.0	28.0	31.0	10.0	7.0	7.0
	1.0	22.0	1.0	31.0	32.0	10.0	9.0	9.0
	22.0		32.0			76.0	44.0	44.0

Average Inter Arrival Time: 2.20

Service Time: 3.20

Turn Around Time: 7.60

Wait Time: 4.40

Response Time Time: 4.40

Probabilty of Waiting Customers: 0.90

Probabilty of Non Waiting Customers: 0.10