# Bank Teller Simulation Report

## I. Motivation

People often wait a long time in line at the bank. This can make people uncomfortable. Banks also want to work efficiently. We want to find ways to make service faster and better. Testing this in real life is hard, so we use simulation to test different setups. This helps us understand what works best without real-world risks.

## II. Problem Statement

We simulate a bank with 160 customers arriving during the working hour of the bank (8 hours). Each customer needs help from a teller. The amount of work needed is between 5 and 15 units. We call this "work units". Tellers process 10 work units in one hour. If all tellers are busy, customers wait in line. We want to see how different setups affect waiting time and how many people are served.

## III. Related Work and Background Material

This project uses ideas from discrete event simulation (DES). DES is used in many areas like factories, hospitals, and computers. It helps model systems where events happen at different times. Similar problems are studied in queueing theory and CPU scheduling. We also look at ideas from customer service models, like express lanes in supermarkets.

## IV. Approach

We use Python to build a simulation. Customers arrive at random times following a uniform distribution. They are each given a work-unit value. Tellers serve them based on who is available the soonest. If no teller is free, the customer waits. We run different setups: with fewer or more tellers, with priority queues, or with faster tellers.

The main parts of our simulation:

- **Customer Arrival**: Random times across the day
- **Teller Service**: Fixed or variable speed
- **Event Handling**: Each action happens at a certain time
- **Queues**: Regular and priority-based

## V. Experiment Set-up

### Metrics

- Average wait time (in minutes)
- Number of people served
- Number of people not served

### Benchmarks

1. 9 tellers at 10 work units/hour

2. 10 tellers at 10 work units/hour
3. 11 tellers at 10 work units/hour
4. Priority queue for light work
5. All tellers working slightly faster (11 work units/hour)

**Data Collection**

- 160 customers per run
- Arrivals Arrivals follow a uniform distribution during the 8-hour day
- Work units follow a truncated normal distribution centered around 5

## VI. Results and Discussion

The setup with 10 tellers gave good results. Adding one more teller helped reduce wait time and served more people. Removing one teller made things worse. When using a priority queue, results were better only if the threshold was chosen well. In our case, using a threshold of 5.4 caused too many people to go to the light queue. This made it slower and less efficient. We also tested having all tellers work faster (11 work units/hour instead of 10). This setup gave the best results, with the lowest wait time and the highest number of people served.

**Sample Results:**

| Setup | Avg Wait (min) | Not Served |
|---|---|---|
| 9 tellers (10 WU/hr) | 48.36 | 34 |
| 10 tellers (10 WU/hr) | 27.64 | 20 |
| 11 tellers (10 WU/hr) | 9.31 | 12 |
| Priority Queue (5.4 WU, 1 light teller) | 32.92 | 75 |
| All tellers at 11 WU/hr | 9.49 | 10 |

## VII. Contributions and Conclusions

We showed how to use simulation to test different bank setups. We found that:

- Priority queues help if tuned correctly, but can be worse if the light queue is overloaded
- Balanced speed across all tellers is better than making just one teller faster
- More tellers reduce wait time but may cost more
- Faster teller also help the efficiency of serving customers

This method can be used in other places like hospitals or help desks. Future work could test customer behavior, like leaving if the line is too long, or compare cost vs. benefit of adding tellers.