Dynamic and Informed Branch Simulator

David D. Daffon* Jei Aldrindael B. Gamboa Mitch R. Kong and Paul Kenneth H. Remo

National Institute of Physics, UP Diliman
*Corresponding author: daviddaffon921@gmail.com

Abstract

On-site bank transactions are frequently perceived as inefficient by customers and burdensome for tellers during peak hours. To address this, we introduce the Dynamic and Informed Branch Simulator (DIBS) model which simulates customer arrival patterns and transaction queues to estimate customer wait times. Transactions are classified into two, long and short, with a designated teller for each type of transaction. The model predicts queue congestion during peak hours when simulating an entire workday, while month-long simulations highlight surges in client traffic on the 15th and 30th, due to scheduled salary releases. Using the information provided by the model, DIBS generates actionable recommendations for branch resource reallocation, aimed to reduce congestion and improve overall customer experience. The model is easily customizable and can be refined better inputs such as more detailed client databases and branch data, making it more accurate in simulating and providing insights for further optimization.

Keywords: agentic AI, simulation, hackathon

1 Introduction

Long queues and poor layouts waste customer time and inhibit proper bank resource allocation. In Metro Manila, the average in-branch wait time for banks is 32.9 minutes, which is a friction point for time-conscious clients [1]. 73% of in-branch visitors delay or abandon transactions due to crowding and inefficiency [2]. In addition, banks lack real-time visibility into customer flow and have difficulty simulating layout or staffing scenarios dynamically. This leads to underutilized assets, budget overruns, and frustrated clients especially for transactions that must still occur in-branch, such as loan applications and financial consultations. Simulating bank processes can allow banks to make informed decisions and reallocate resources to improve waiting times for customers.

A typical transaction in a bank follows the steps described below.

- Customers arrive at some frequency, which is determined by the inter-arrival time (IAT) $\Delta\tau$.
- Customers queue and wait in line to complete a specific transaction type.
- A teller trained to do a specific transaction completes the transaction with the customer for some transaction time Δt_{type}.
- Customer leaves the branch upon completion of the transaction.

Since bank processes involve the movement of a single customer and the dynamics that arise from having multiple independent customers, discrete-event simulation (DES) is an appropriate method [3] [4]. Combined with queueing theory, DES can accurately recreate bank processes.

The data gathered from the modeling of bank processes show all inadequacies, underutilized assets, inefficiencies, and possible improvements for every specific branch. This will be further expounded by the introduction of an Agentic AI which uses the model data and provides real-time feedback aimed to optimize not just the speed and efficiency of bank transactions, but consequently improving the average banking experience of customers.

The objectives of the Dynamic and Informed Branch Simulator (DIBS) project are as follows:

- Simulate typical daily and monthly processes in a bank branch,
- Create an interactive interface for simulation of such processes, and
- Implement an agentic AI chatbot to assist in decision-making.

2 Methodology

2.1 Model for Branch Processes

Our model makes some key assumptions. We assume that there are two transaction types by length: short and long. Each customer with queueing number i enters the branch at time τ_i to perform one transaction of a specific type and is categorized as a short transaction customer or a long transaction customer. The time between the successive arrival of two customers of a specific type is defined as $\Delta\tau_{type} = \tau_{type,i} - \tau_{type,i-1}$. Thus, $\Delta\tau_{short}$ is the inter-arrival time (IAT) of short transaction customers and $\Delta\tau_{long}$ is the IAT of long transaction customers. We assume that in most cases, short transaction customers arrive more frequently than long transaction customers, so $\Delta\tau_{short} < \Delta\tau_{long}$. Fig. 1 summarizes the bank branch processes that are simulated in DIBS.

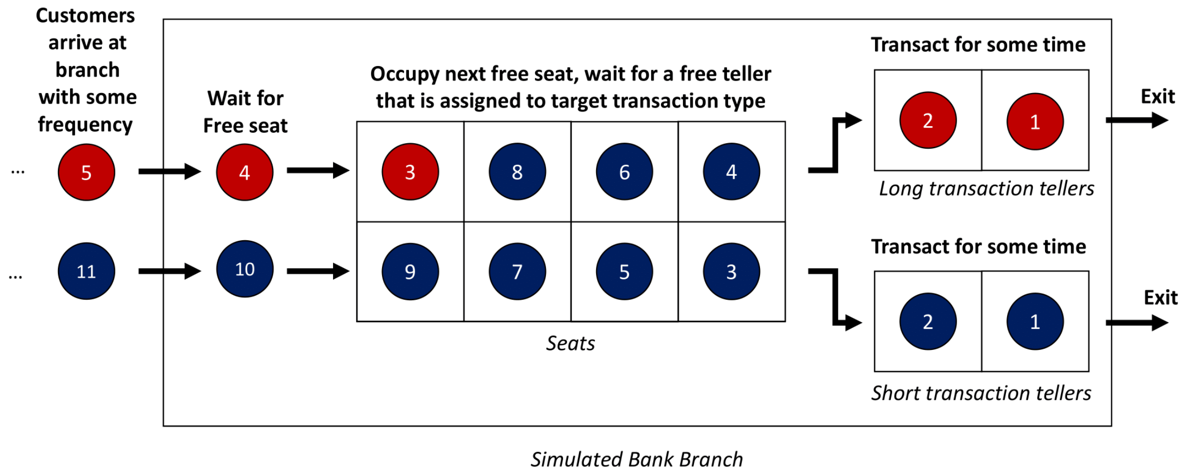


Figure 1: Illustration of the simulated processes in the bank. Circles correspond to customers and they are labeled with a queueing number i. Red circles correspond to long transaction customers and blue circles correspond to short transaction customers.

According to queueing theory, the Poisson distribution is useful since it returns the probability that a number of arrivals n can occur in a given time interval Δt. It also states that when a number of arrivals in a given time interval follows a Poisson distribution, IATs follow an exponential distribution [5] given by Eq. 1. In this equation, μ is the mean inter-arrival time. We assume that the inter-arrival time for each customer type follows the exponential distribution.

$$P(\Delta\tau) = \mu e^{-\mu\Delta\tau} \tag{1}$$

Furthermore, we assume that the mean IAT μ varies depending on the time of day t – that is, $\mu = \mu(t)$. We impose that there is a maximum mean IAT μ_{max} and a minimum mean IAT $\mu(t_{min}) = \mu_{min}$ that occurs on the peak hour t_{min} on a given day. This replicates the rush hour phenomenon in the simulation. We suppose some base mean IAT μ_0, some peak hour t_{min}, and standard deviation of the mean IAT σ_0. Then Eq. 2 returns the dynamic mean IAT to be used in the simulation of a single day of operations. Fig. 2a illustrates the variation of $\mu(t)$ depending on the time of day.

$$\mu(t) = \mu_0 + (\mu_{max} - \mu_{min})(1 - e^{-(t-t_{min})^2/2\sigma_0^2}) \tag{2}$$

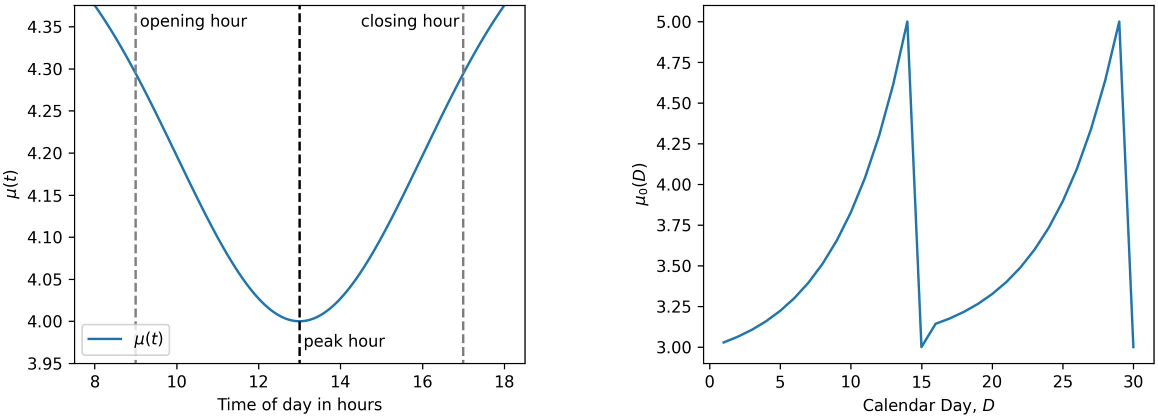
Hence, the probability distribution of the IAT of the short-transaction customers follows Eq. A, which is obtained by plugging Eq. 2 into Eq. 1.

$$P(\Delta\tau, t) = \mu(t)e^{-\mu(t)\Delta\tau} \tag{3}$$

To replicate rush days in the simulation of a month of operations, μ_0 varies depending on the calendar day D and is lowest on peak days $D = D_0$, where $\mu_0(D_0) = \mu_{0,min}$. The slowest days are right before the peak days where $\mu_0(D_0 - 1) = \mu_{0,max}$.Typically, D_0 is a day where salaries are disbursed, so $D_0 = 15, 30$. Suppose that after a peak day and before the next peak day, μ_0 increases at a constant rate $1/\kappa$ daily. Eq. 4 satisfies these conditions with k given by Eq. 5. Fig. 2b shows the variation of μ_0 depending on the calendar day.

$$\mu_0(D) = \begin{cases} \mu_{0,min} & \text{when } D = 15, 30 \\ (\mu_{0,max} - k)e^{(D-D_0)/\kappa} + k & \text{when } D < \text{succeeding } D_0 \end{cases} \tag{4}$$

$$k = \frac{(\mu_{0,min} - \mu_{0,max}e^{-D_0/\kappa})}{1 - e^{-D_0/\kappa}} \tag{5}$$



(a) $\mu(t)$ over the course of a day.

(b) $\mu_0(D)$ over the course of a month.

Figure 2: Variation of the mean IAT over two different time scales.

We also assume that the time consumed performing a specific transaction type Δt_{type} follows a triangular probability distribution given by Eq. 6. This is because transactions have some minimum consumed time a, some maximum consumed time b, and a mode of time consumed c.

$$P(\Delta t_{type}) = \begin{cases} 0, & \Delta t_{type} < a, \Delta t_{type} > b \\ \frac{2(\Delta t_{type} - a)}{(b-a)(c-a)}, & a \leq \Delta t_{type} \leq c \\ \frac{2(b - \Delta t_{type})}{(b-a)(b-c)}, & a \leq \Delta t_{type} \leq c \end{cases} \tag{6}$$

The number of short transaction tellers and long transaction tellers in the bank branch are operational variables that can be controlled in the simulation. These control the efficiency and speed of the branch in handling transactions of a specific type.

Additionally, we assume that each bank branch has a total floor area A and a waiting area W. The ratio of the waiting area to the total floor area is defined as $r = W/A$ and we impose $W < A$. To calculate the customer capacity C of a branch, we assume that when the branch is at maximum capacity, each customer occupies some surface area S in the waiting area. C is then given by Eq. 7.

$$C = \lceil (Ar/S) \rceil \tag{7}$$

2.2 Simulation

The simulation was done using the Python programming language. For random processes such as customer arrivals and the transaction time, random sampling was done. IATs were generated by random sampling from an exponential distribution (Eq. 1) while transaction times were generated by random sampling from a triangular distribution (Eq. 6).

2.3 Datasets utilized

DIBS leverages publicly available data to power its simulations and projections. Branch information, including name, address, coordinates, and estimated floor area, was sourced from OpenStreetMap [6]. Population data from the Philippine Statistics Authority's 2020 census was also incorporated at the barangay level within Quezon City [7]. By linking population figures to nearby branches, we approximate the customer base each branch serves.

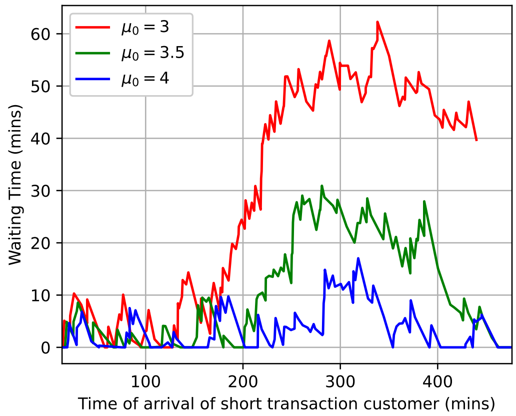
These datasets are then refined and integrated into the model described in 2.1, enabling simulations of average bank traffic both on a daily basis and across a monthly cycle.

2.4 Agentic AI

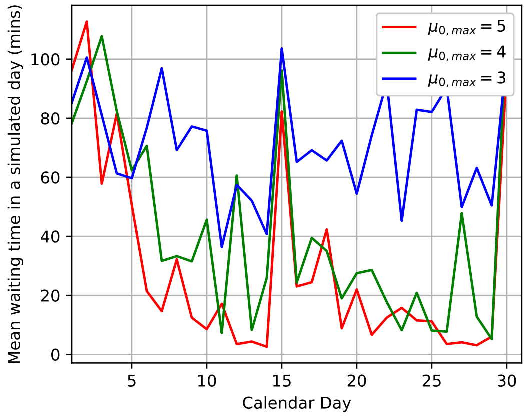
DIBS uses agentic retrieval-augmented AI to assist its user in the interpretation of the simulation results. Langchain, OpenAI's API, and Supabase were used for its implementation. An ingesting script feeds various publicly available BPI reports, descriptions, and the raw data for training into the LLM (GPT-4o-mini) allowing tailored assistance to relevant user prompts. The data is stored in a vectorized database in Supabase and is queried by a Langchain-wrapped tool. Interaction with the agentic RAG AI is enabled by streamlit.

3 Results and Discussion

Congestion happens when the frequency of customer arrival in a time interval is greater than the number of processed transactions in the same time interval.



(a) Waiting times for each customer in a simulated day with scenarios $\mu_0 = 3, 3.5, 4$ mins. For all scenarios, $\mu_{max} = \mu_0 + 0.5, \mu_{min} = \mu_0 - 0.5, t_{min} = 240$, and $\sigma_0 = 180$.

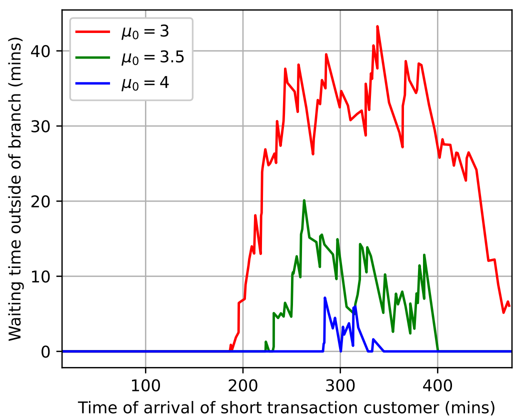


(b) Mean waiting times in a branch for a simulated month with scenarios $\mu_{0,max} = 5, 4, 3$ mins. For all scenarios, $\mu_{0,min} = 2$ mins and $\kappa = 20$.

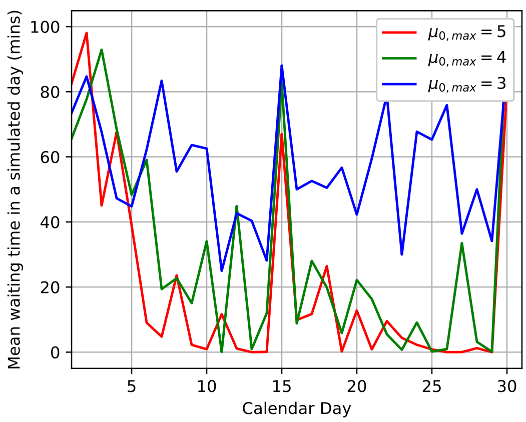
Figure 3: Waiting times for a (a) simulated day and a (b) simulated month.

Figure 3a illustrates the relationship between customer arrival rates for short transactions and the corresponding total waiting time. The results highlight that during peak hours, as more customers arrive, total waiting times sharply rise since the number of active bank tellers cannot fully keep pace with the growing queue.

This analysis is further extended in Figure 3b, which models the average waiting times over the course of a month. The simulation accounts for the surge of customers on the 15th and 30th, coinciding with salary releases. The findings indicate significantly heavier congestion during these peak periods and the days shortly after.



(a) Waiting times outside of the branch for each customer in a simulated day with scenarios $\mu_0 = 3, 3.5, 4$ mins. For all scenarios, $\mu_{max} = \mu_0 + 0.5, \mu_{min} = \mu_0 - 0.5, t_{min} = 240$, and $\sigma_0 = 180$.



(b) Mean waiting times outside of the branch for a simulated month with scenarios $\mu_{0,max} = 5, 4, 3$ mins. For all scenarios, $\mu_{0,min} = 2$ mins and $\kappa = 20$.

Figure 4: Waiting times outside of the branch for a (a) simulated day and a (b) simulated month.

Fig. 4a shows the relationship between the arrival time of the customer and the time spent waiting outside of the branch. In the scenario where $\mu_0 = 3$, the branch exceeded its capacity and customers arriving after peak hours had to queue and wait outside of the branch. Meanwhile, in the scenario where $\mu_0 = 4$, the branch exceeded its capacity for a short amount of time but recovered quickly. Thus, DIBS can simulate scenarios where the branch is overwhelmed in a day of operations and yield insight on how quickly the branch is able to resolve temporary traffic surges.

Fig. 4b shows the daily mean waiting time outside of a branch for different scenarios. The scenario where $\mu_{0,max} = 3$ was regularly overwhelmed due to the nonzero mean waiting time outside. This indicates that there is a need to increase the tellers operating every day in the branch. The scenario where $\mu_{0,max} = 5$ had certain days where the mean waiting time outside of the branch was nonzero, indicating that the branch typically gets overwhelmed during and after salary days. In such cases, it may be appropriate to deploy more tellers during and shortly after salary days.

4 Conclusions

The DIBS model can simulate customer behavior at both daily and monthly scales. It highlights inefficiencies in current banking practices that create strain on branch resources and negatively impact customer experience. By pinpointing these areas, the model provides actionable insights that can be fed back into DIBS to optimize resource allocation and branch space utilization—ultimately enabling smoother transactions and an improved customer experience.

DIBS can be further enhanced with more precise data on the number of customers served by each branch. Information such as the total registered client base would establish stronger baselines for customer behavior and branch traffic. Since DIBS is highly customizeable, it can readily incorporate these inputs to generate outputs that more accurately reflect real-world banking transactions.

Acknowledgments

We thank BPI and Eskwelabs mentors for their advice and domain expertise. We would also like to thank Ms. Beth Raymundo and Ms. Kristine Gonzales for their valuable insights that helped us in replicating branch processes.

References

[1] BSP, National Baseline Survey on Financial Inclusion (2015), https://www.bsp.gov.ph/Inclusive%20Finance/Financial%20Inclusion%20Reports%20and%20Publications/2015/NBSFIFullReport.pdf.

[2] M. Rodriguez, Transforming customer experience in banking through queue management systems: Benchmark metrics and best practices, *Fabworld Today* (2024).

[3] T. Monks, A. Harper, A. Heather, A. Mayne, and N. Mustafee, in *Proceedings of the Operational Research Society Simulation Workshop 2025 (SW25)*, edited by M. Luis, A. Harper, T. Monks, and N. Mustafee (Operational Research Society, 2025), https://doi.org/10.36819/SW25.004.

[4] T. Monks, A. Harper, and A. Heather, An introduction to discrete-event simulation using free and open source software. (2024), https://doi.org/10.5281/zenodo.13971859.

[5] DAL, (Data Action Lab) Basics of Queueing Theory, https://www.data-action-lab.com/wp-content/uploads/2020/09/Slides-Module-3-Basics-of-Queueing-Theory.pdf.

[6] OpenStreetMap contributors, Planet dump retrieved from https://planet.osm.org , https://www.openstreetmap.org (2017).

[7] PSA, Quezon City Population - 2020 Census (2020), https://psa.gov.ph/classification/psgc/barangays/1381300000.