Crowd Crunchers

October 11, 2020

AI for Crowd Control in GTBank Branches

Crowd Crunchers

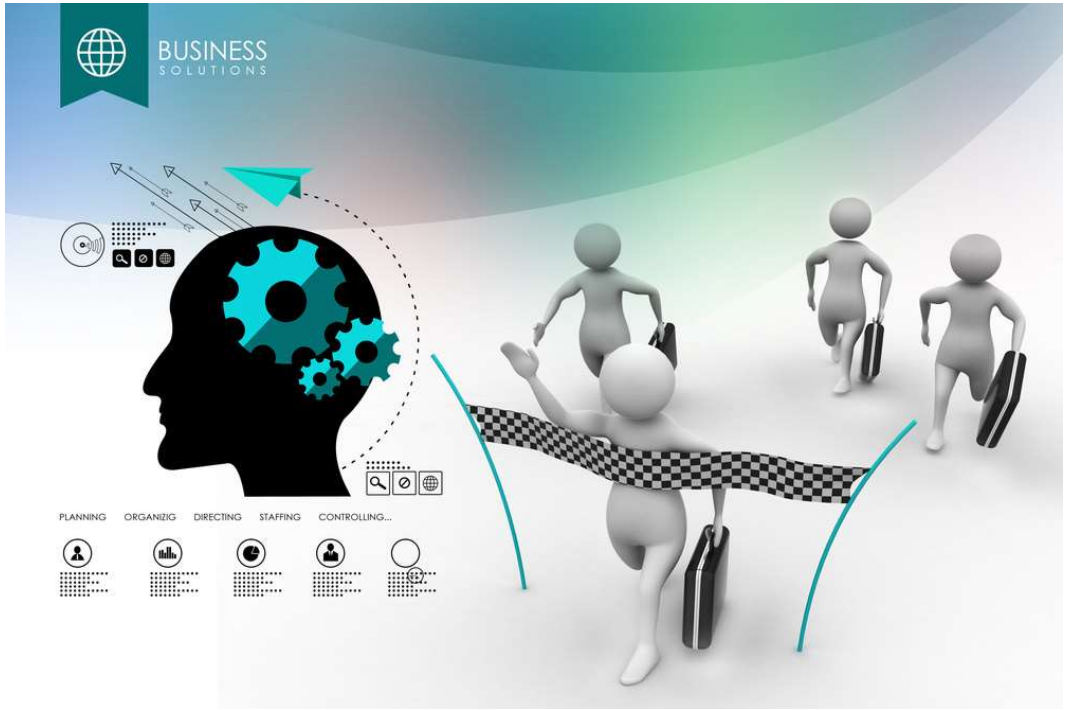
GTBank Tech Academy IV: Hackathon Challenge

AI for Crowd Control in GTBank Branches

Crowd Crunchers

Introduction

The goal of this team is to provide an efficient and effective means of crowd control with the help of artificial intelligence.



Crowd Crunchers

Team Members

1. Anthony Ubani
2. Fikemi Femi-Fred
3. Joel Anyam
4. Oluwajuwon Ademuyiwa

Crowd Crunchers

Outline

Crowd Crunchers

GTBank Stats

Est. **8 million** Customers

Relbanks.com 2018

231
Branches

1165
ATM Machines

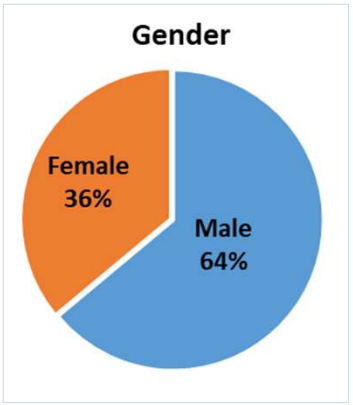
17
Cash Centres

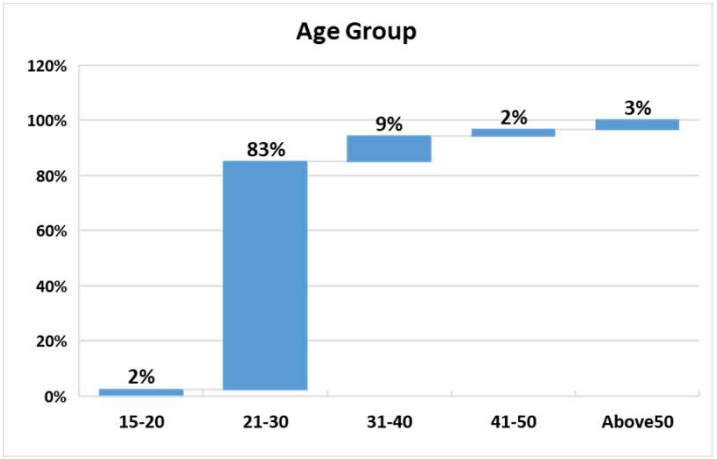
18
E-Branches

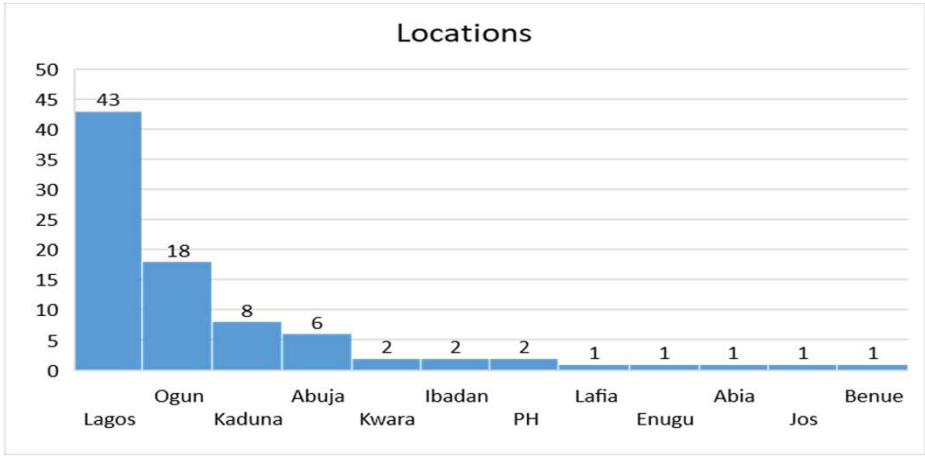
41
GTExpress Locations

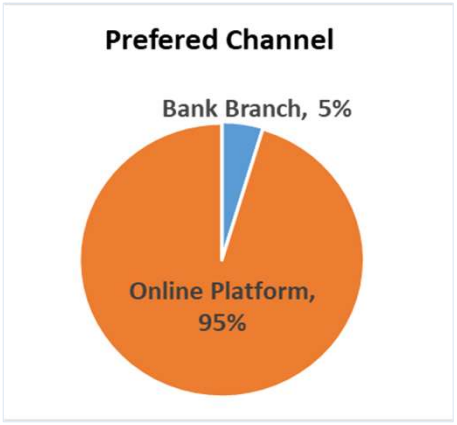
Wikipedia

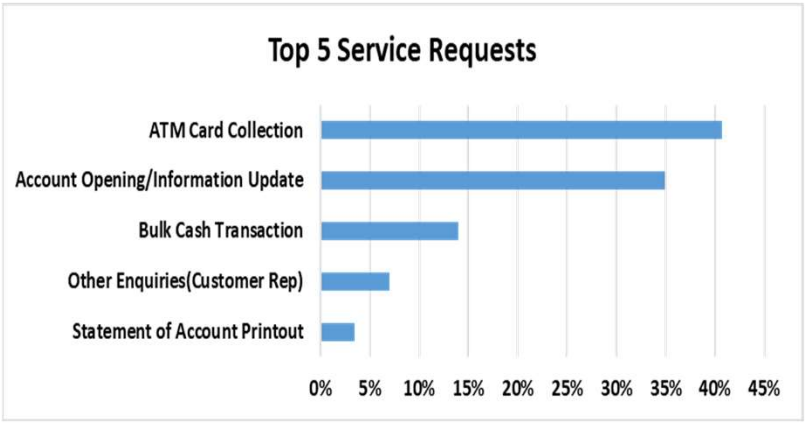
Customer Service Improvement Survey

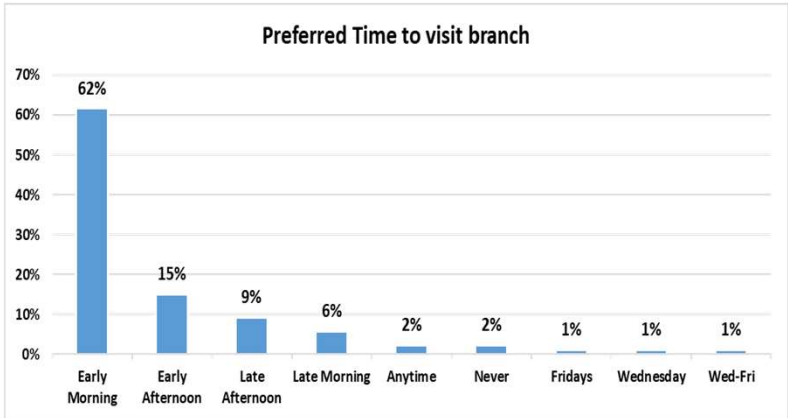












Our Respondents!

Customer Service Improvement Survey

Customer Complaints

ATM Collection

I had been debited for an ATM card and asked to go to the bank where I opened it to get the card. All the way in Jalingo as the bank did not send ATM cards across state lines anymore. Needless to say, after more than a week no one could resolve this issue and I was asked to pay for again for the same card to be issued to me in Lagos. So I was charged twice for the same service and despite my complaints, the initial automated charge from Jalingo was not reversed. I asked for a mail to write to and complain and a customer service agent gave me an invalid email address so my email never went through.

ATM Dispense Error

As stated above, issues of dispense error (debited but cash not dispensed by the machine) almost forfeited the cash due to elongated time for rectification and refund.

Existing Account Error

I went to GTBank to open an account. It took a long time before they realized I had a previous account from when I was younger that was proving difficult to address

Account Unlock

I went to a bank for them to unblock my account as I had requested them to block it via a phone call some days back. I got there and filled the necessary forms and all and was told the issue will be resolved within 24 hours but it actually took them four days to resolve the issue as I kept on going to the bank afterwards before it was eventually resolved after the fourth day..

Account Upgrade

The last time I went to upgrade my account. I practically spent the whole day at the bank. It was really a horrible experience, not just for me but for most of my colleagues that went too, really banks can be frustrating these days.

Poor Customer Relationship

Went to open account and the que was much and slow the person attending to us also did not have customer relationship there I spent a good hours of my day.

Failed Bank-App Transaction

I made a transaction via my bank app and was debited without any value for the transaction. I thought it was a hitch free process but was frustrated afterwards because it wasn't resolved as planned. I had to be following up with phone calls despite filling out a complaint form

Error in BVN Details

The name on my BVN and on my account differs which is as a result of the mistake done by the bank they made me visit the bank for weeks asking me to bring court affidavit an ID card and so on. It was really frustrating though

Delayed Account Document

A letter of account confirmation. It took me about a week of back and forth. This experience is especially painful because I needed the document urgently

Can you share briefly an experience of when you went to the bank to get something done and it wasn't resolved on time?

Crowd Crunchers

Are you ready?

Our Solutions

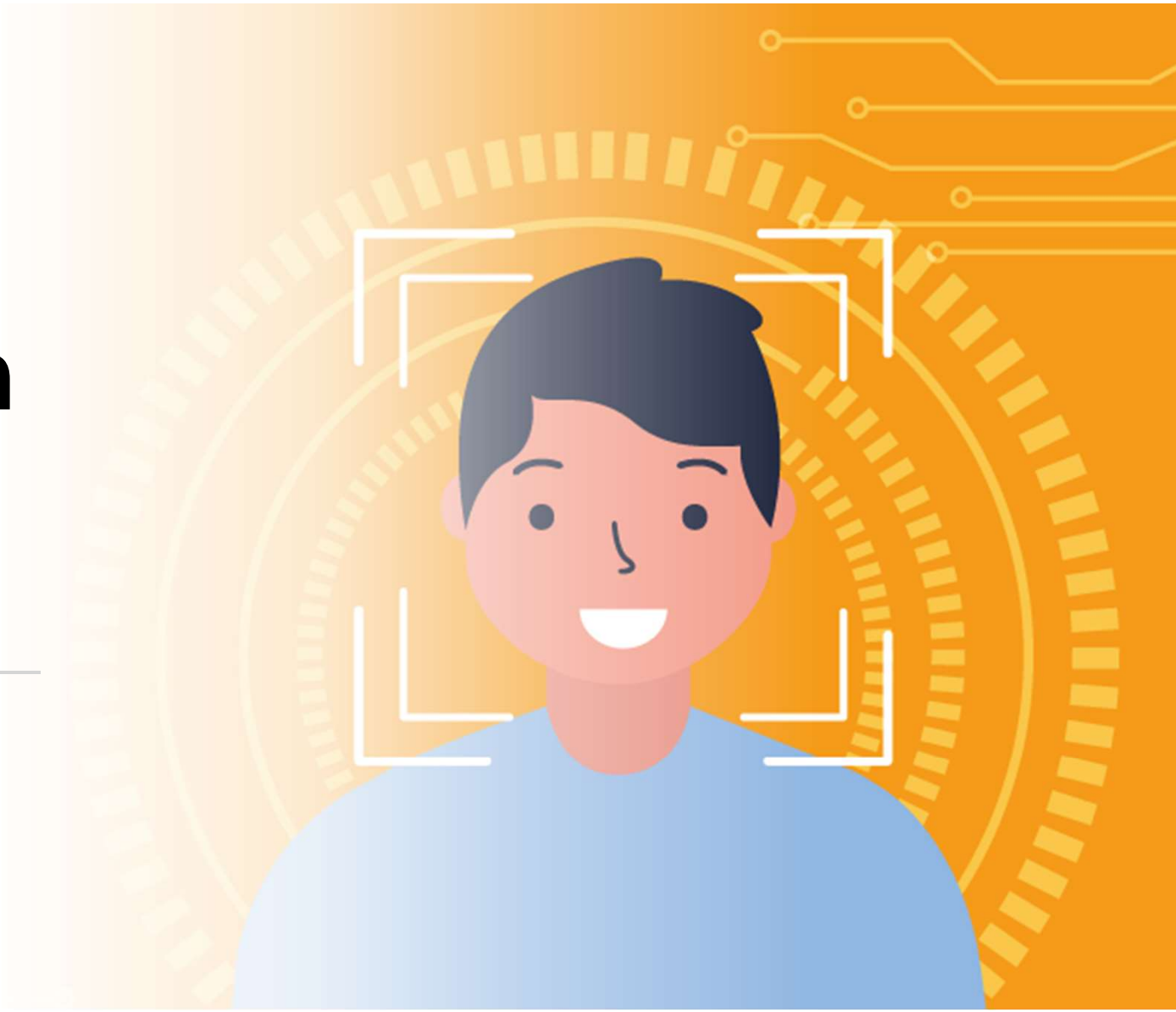
Our Solution

Phase 1
Crowd Estimation

Phase 2
Crowd Control

Phase 3
Crowd Elimination

Crowd Estimation by Face Detection



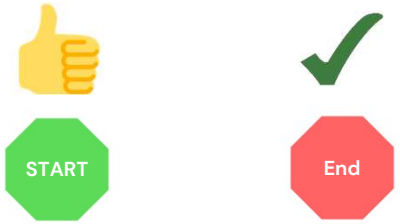
Face Detection

1. Real-time info on crowd density
2. Track crowd movement
3. Identify busy periods & branches

Face Detection

1. Write code
2. Load images

3. Detect faces



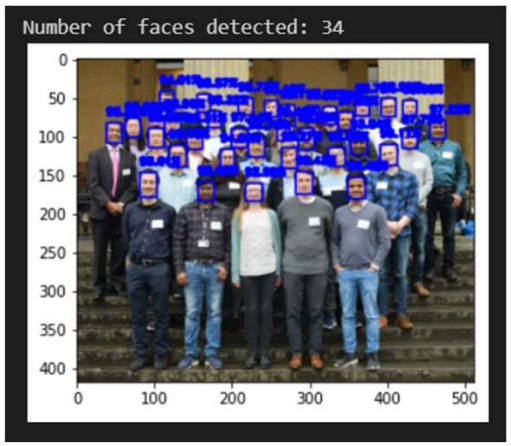
Our Code

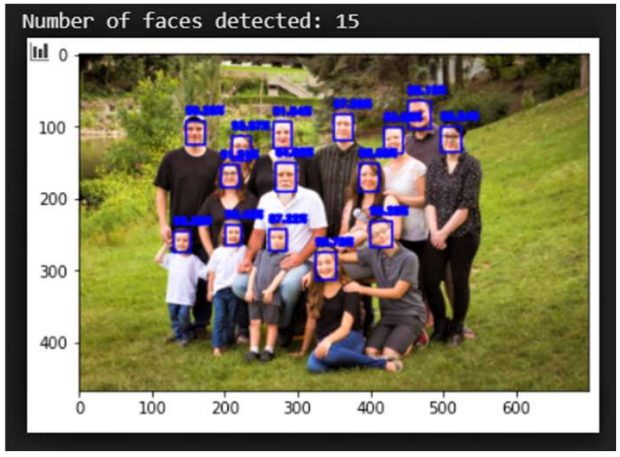
Deep learning based face detector in OpenCV

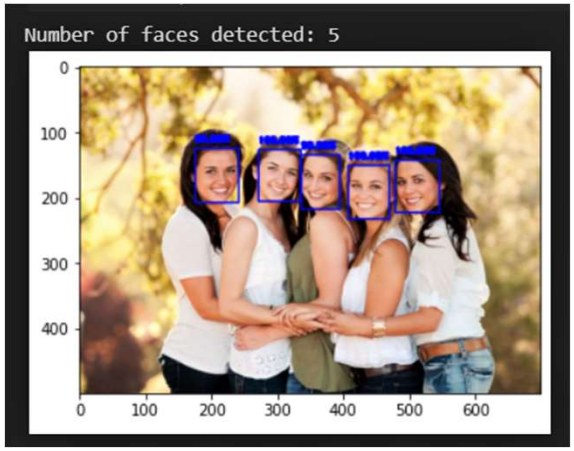
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def detectDNN(imgpath):
    modelFile = r'C:\\Users\User\Documents\Life\GTB\Tech Academy\Hackathon\
    Face Detection\Code\DNN Method\Models\res10_300x300_ssd_iter_140000.caffemodel'
    configFile = r"C:\\Users\User\Documents\Life\GTB\Tech Academy\Hackathon\
    Face Detection\Code\DNN Method\Models\deploy.prototxt.txt"
    net = cv2.dnn.readNetFromCaffe(configFile, modelFile)

    frame = cv2.imread(imgpath)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (800, 600)), 1.0,
        (800, 600), (104.0, 117.0, 123.0))

    net.setInput(blob)
    detections = net.forward()

    #[,frame,number of detection,[classid,class score,conf,x,y,h,w]]
    # loop over the detections
    # initialize a count for the number of faces
    count = 0
```

```
for i in range(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with the
    # prediction

    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the `confidence` is
    # greater than the minimum confidence
    # count = 0
    if confidence > 0.5:
        count+=1
        faces = count
        # compute the (x, y)-coordinates of the bounding box for the
        # object
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # draw the bounding box of the face along with the associated
        # probability
        text = "{:.2f}%".format(confidence * 100)
        y = startY - 10 if startY - 10 > 10 else startY + 10
        cv2.rectangle(frame, (startX, startY), (endX, endY),
            (0, 0, 255), 2)
        cv2.putText(frame, text, (startX, y),
            cv2.FONT_HERSHEY_DUPLEX, 0.45, (0, 0, 255), 3)

# print out the number of faces detected
print ("Number of faces detected: {}".format(faces))

# show the output image
plt.imshow( frame)
plt.show()
```

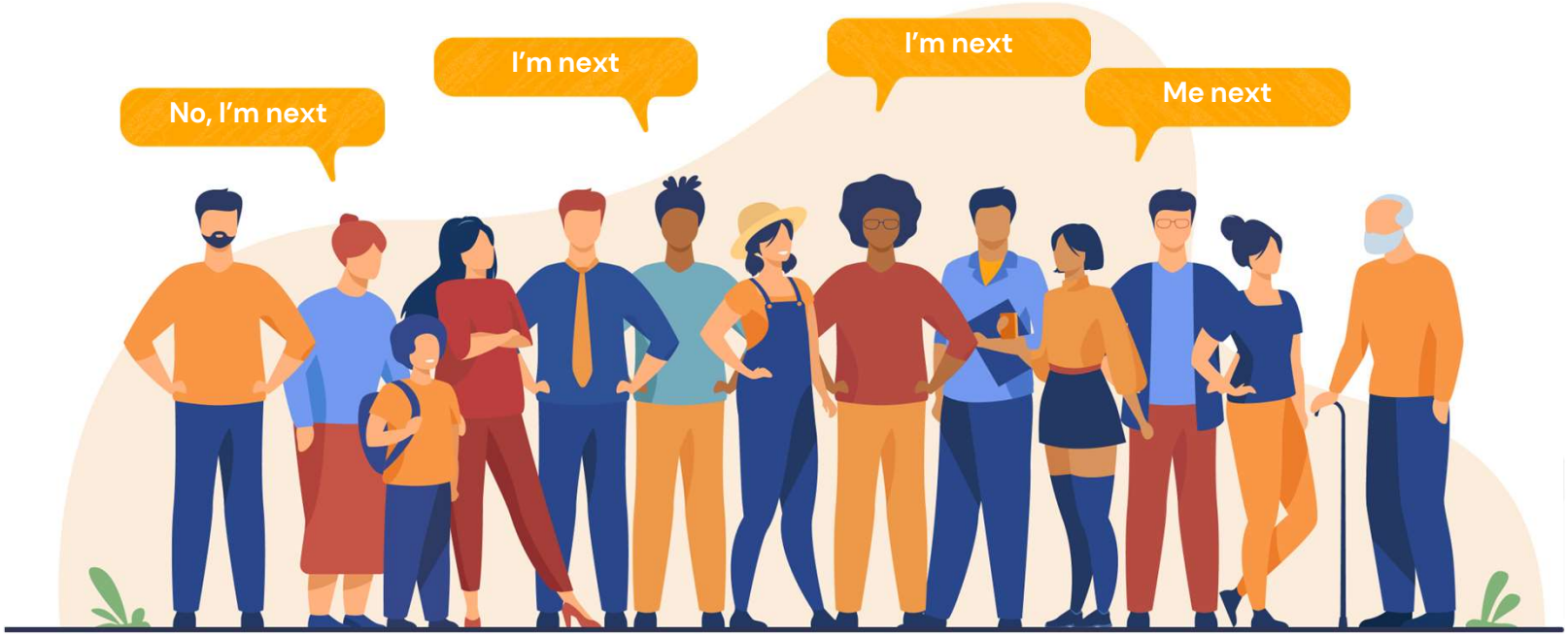
Our Results


Number of faces detected: 34
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

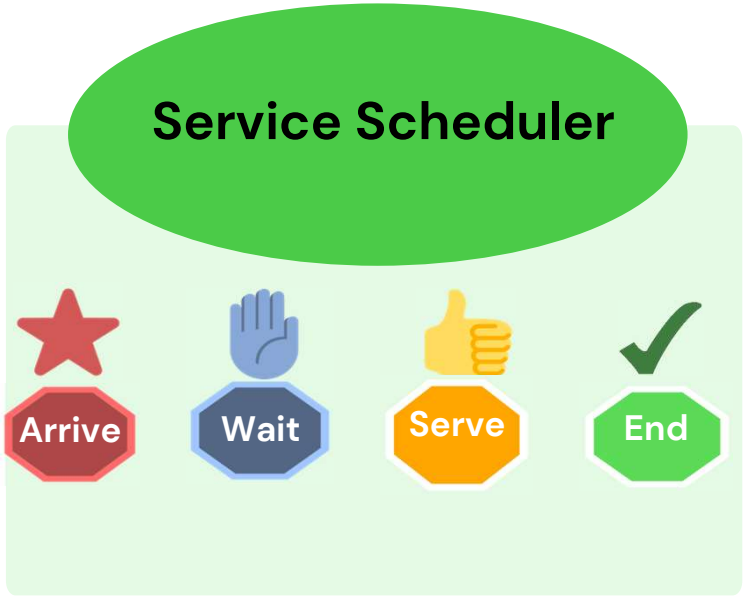

Number of faces detected: 15
0
100
200
300
400
0
100
200
300
400
500
600


Number of faces detected: 5
0
100
200
300
400
0
100
200
300
400
500
600

Crowd Crunchers



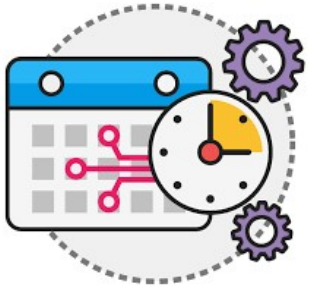
Who goes first?



Key Factors

1. Arrival Time/Order
2. Service Time
4. Service delivery
3. Waiting Time

Software Simulation



Action Items

1. Service Selection
2. Schedule Service
3. Calculate wait time
4. Token/Ticket

Shortest Service First!

Code Snippet

```
class SJF:

    def processData(self, no_of_customers):
        process_data = []
        for i in range(no_of_customers):
            temporary = []
            customer_id = str(input("Please enter Customer ID: "))

            arrival_order = i+1
            service_id = int(input(f"Dear {customer_id}, Please select the service you want:
```

Schedule Key

Service Code	Service Description	Estimated Duration (Minutes)
1	Account Opening/Information Update	30
2	ATM Collection	25
3	Bulk Cash Transaction	40
4	Cash Deposit (Under 50k)	10
5	Statement of Account Printout	20
6	Internet/Mobile Banking Setup	5
7	Bill Payments(FIRS, PHCN, School Fees, etc.)	20
8	Funds Transfer(Western Union, MoneyGram etc.)	35
9	Other Enquiries(Talk to a Rep)	5

```
def calculateWaitingTime(self, process_data):
    total_waiting_time = 0
    for i in range(len(process_data)):
        waiting_time = process_data[i][5] - process_data[i][2]
        '''
        waiting_time = turnaround_time - service_time
        '''
        total_waiting_time = total_waiting_time + waiting_time
        process_data[i].append(waiting_time)
    average_waiting_time = total_waiting_time / len(process_data)
    '''
    average_waiting_time = total_waiting_time / no_of_customers
    '''
    return total_waiting_time
```

BATCH SIZE: 10 Customers

Unscheduled Batch

S/N	Customer_ID	Service_Code	Service_Description	Service_Time	Completion_Time	Turnaround_Time	Waiting_Time
1	Customer1	1	Account Opening/Information Update	30	30	30	0
2	Customer2	5	Statement of Account Printout	20	50	50	30
3	Customer3	8	Funds Transfer(Western Union, MoneyGram e.t.c.)	35	85	85	50
4	Customer4	3	Bulk Cash Transaction	40	125	125	85
5	Customer5	4	Cash Deposit (Under 50k)	10	135	135	125
6	Customer6	2	ATM Collection	25	160	160	135
7	Customer7	2	ATM Collection	25	185	185	160
8	Customer8	4	Cash Deposit (Under 50k)	10	195	195	185
9	Customer9	6	Internet/Mobile Banking Setup	5	200	200	195
10	Customer10	2	ATM Collection	25	225	225	200

Average Turnaround Time: **139 minutes**

Total Waiting Time: **1165 minutes**

39% Reduction

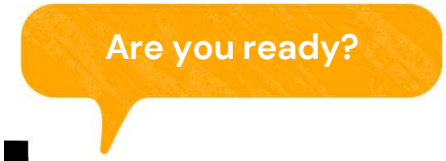
Scheduled Batch

S/N	Customer_ID	Service_Code	Service_Description	Service_Time	Completion_Time	Turnaround_Time	Waiting_Time
1	Customer 9	6	Internet/Mobile Banking Setup	5	5	5	0
2	Customer 5	4	Cash Deposit (Under 50k)	10	15	15	5
3	Customer 8	4	Cash Deposit (Under 50k)	10	25	25	15
4	Customer 2	5	Statement of Account Printout	20	45	45	25
5	Customer 6	2	ATM Collection	25	70	70	45
6	Customer 7	2	ATM Collection	25	95	95	70
7	Customer 10	2	ATM Collection	25	120	120	95
8	Customer 1	1	Account Opening/Information Update	30	150	150	120
9	Customer 3	8	Funds Transfer(Western Union, MoneyGram e.t.c.)	35	185	185	150
10	Customer 4	3	Bulk Cash Transaction	40	225	225	185

Average Turnaround Time: 93.5 minutes

Total Waiting Time: **710 minutes**

Crowd Crunchers



Crowd Elimination

Recommendation

In line with the report of our survey, the team strongly recommends that GTBank implements more online solutions that will render seamless financial services, provide optimal customer satisfaction and in turn reduce crowd in the bank.

Don’t follow the crowd, let the crowd follow you.”

— Margaret Thatcher

Crowd Crunchers

Thank you

Have a great day ahead.