**Title: ATM Management System**

**Team member details:**

PES2UG21EC037 – DEVESH B

PES2UG21CS250 – L SAI TEJAS

PES2UG21CS406 – RANJIVE R

**ABSTRACT**

The project aims at simulating the working of an ATM(automatic teller machine). The entire code of the project is typed in python. Modules like OpenCV , Tkinter , random and few user-defined modules were used in this program. This provides options to use either credit or debit card to either transfer or deposit money into the accounts. This program also uses facial recognition as a passcode for a user to access their accounts. Face recognition is achieved by using NumPy and PIL libraries from python. For the database, we use MySQL and MySQL. Connector to store the data of the users and modify them according to the requirements.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| SL.NO. | CONTENT | PAGE NUMBER |
| 1 | INTRODUCTION | 4 |
| 2 | DESIGN/PHOTOS | 5-15 |
| 3 | TESTING | 16 |
| 4 | RESULT/ANALYSIS | 17 |
| 5 | CONCLUSION | 18 |
| 6 | FUTURE ENHANCEMENT | 18 |

**INTRODUCTION :**

The program simulates the working of an ATM . Initially the user has to make a facial ID which is done by capturing 310 images of the user through the webcam/laptop camera and the images are trained under the name entered by the user .When the user tries to access the ATM through face ID a random account is created with random amount of money in savings and checking accounts. Now the user has option to either withdraw or deposit money from either of the accounts.

In the withdrawal option, the user can enter the amount of money to be withdrawn which should be in multiples of 10 and the maximum limit is Rs.10000. There is a time gap of 6 seconds in withdrawing money from the ATM and sending it at the slot from where the user can collect the money . In the deposit option, the user can enter the amount of money to be deposited which again has a limit up to Rs.10000 .The user can deposit the money at the cash deposit slot and there is a time gap of for the money to be validated and it is successfully deposited. if the user no longer wants to continue the procedure ,there is an option to exit the program.

**DESIGN :**

**gui.py :**

import tkinter as tk

from tkinter import font as tkfont

from tkinter import messagebox,PhotoImage

from PIL import Image, ImageTk

import mysql.connector

checkings\_amt=0.0

savings\_amt=0.0

userid=0

name=""

num=0

check=""

start={

"choice":"enter your choice:",

"not accepted":"\nchoice not accepted. please try again.",

"card type":" Are you using a credit or debit card?",

"print type":"Click the button below to proceed:",

"last":"have a nice day!"}

Ist={

"id":"Customer ID:",

"action":"Would you like to make a withdrawal or a deposit?",

"letter wd":"Type the letter\n'W' for a withdrawal\n'D' for a deposit\n'E' to exit.",

"c or s":"Would you like to withdraw the cash from your Checkings Account or your Savings Account?",

"letter cs":"Type the letter \n'C' for Checkings \n'S' for Savings.",

"amountw":"Enter the amount you would like to withdraw.\n\nATM only accepts multiples to 10 up to Rs.10000:",

"val not":"Value not accepted. Please enter value again.",

"collect":"Please collect the cash from the slot.\n\nThank you and have a nice day!",

"deposit":"Would you like to deposit the cash into your Checkings Account or your Savings Account?",

"amountd":"Enter the amount you would like to deposit.\nATM only accepts up to 200 bills upto Rs. 50000:",

"insert":"Please insert your cash into the slot.",

"wait":"Please wait while the ATM validates your cash.",

"last":"Have a nice day!",

"all":"Amount Present in All Accounts:",

"excess":"Withdrawal Amount Exceeds Amount in Account"

}

def sub\_app(usrname):

global name,start,Ist,savings\_amt,checkings\_amt,userid

name=usrname

def wde():

global name,start,Ist

global savings\_amt,checkings\_amt,userid,num

mydb = mysql.connector.connect(host="localhost",

user="root",

password="sai!@#2345",

database="ATM"

)

sql\_select\_Query = "SELECT \* FROM cust\_details"

cursor = mydb.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

for row in records:

if row[1]==name:

userid=row[0]

savings\_amt=row[2]

checkings\_amt=row[3]

cursor.close()

mydb.close()

c=tk.Tk()

fonts = ('Helvetica',14)

c.title("WELCOME TO YOUR ACCOUNT")

c.geometry("400x250")

def exit():

if messagebox.askokcancel("Quit", "Are you sure?"):

c.destroy()

c.protocol("WM\_DELETE\_WINDOW", exit)

t1="{} {}".format(Ist["id"],int(userid))

ids=tk.Label(c,font=fonts,text=t1)

t2="{} {}".format(Ist["all"],(savings\_amt+checkings\_amt))

total=tk.Label(c,font=fonts,text=t2)

action=tk.Label(c,font=fonts,text=Ist["action"])

def withdraw():

global name,start,Ist

global amnt,userid,num

c.destroy()

wid=tk.Tk()

wid.title("WITHRAW")

fonts = ('Helvetica',14)

wid.geometry("400x350")

def exit():

if messagebox.askokcancel("Quit", "Are you sure?"):

wid.destroy()

wid.protocol("WM\_DELETE\_WINDOW", exit)

acnt=""

def back():

wid.destroy()

starting()

wid.protocol("WM\_DELETE\_WINDOW", back)

w=tk.Label(wid,font=fonts,text=Ist["amountw"])

w.place(x=0,y=0)

wdrw=15

v=tk.Label(wid,font=fonts,text=Ist["val not"])

global savings\_amt,checkings\_amt,num

if(wdrw%10!=0 or wdrw==0 or wdrw>10000 or (amnt[num]-wdrw)<0):

a=tk.Entry(wid)

a.place(x=70,y=70)

def submit():

if messagebox.askyesno("Proceed to withdraw", "Are you sure"):

global savings\_amt,checkings\_amt,num,check

wdrw=float(a.get())

mydb = mysql.connector.connect(host="localhost",

user="root",

password="sai!@#2345",

database="ATM"

)

if check=="s":

if(wdrw%10!=0 or wdrw==0 or wdrw>10000 or (savings\_amt-wdrw)<0): #validate input

v.place(x=20,y=110)

if (wdrw%10==0 and wdrw!=0 and wdrw<10000 and (savings\_amt-wdrw)>0):

v.destroy()

txt1="Amount of Rs.{} has been credited.".format(wdrw)

wait=tk.Label(wid,font=fonts,text=txt1)

for i in range(30000000):i+=1 #time delay

coll=tk.Label(wid,font=fonts,text=Ist["collect"])

wait.place(x=30,y=150)

coll.place(x=50,y=185)

sub.place(x=150,y=260)

s=float(savings\_amt)

cursor = mydb.cursor()

s-=wdrw

sql = "UPDATE cust\_details SET amt\_checkings = %s WHERE amt\_checkings = %s "

val = (s, savings\_amt)

cursor.execute(sql,val)

mydb.commit()

if check=="c":

if(wdrw%10!=0 or wdrw==0 or wdrw>10000 or (checkings\_amt-wdrw)<0): #validate input

v.place(x=20,y=110)

if (wdrw%10==0 and wdrw!=0 and wdrw<10000 and (checkings\_amt-wdrw)>0):

v.destroy()

txt1="Amount of Rs.{} has been credited.".format(wdrw)

wait=tk.Label(wid,font=fonts,text=txt1)

for i in range(30000000):i+=1 #time delay

coll=tk.Label(wid,font=fonts,text=Ist["collect"])

wait.place(x=30,y=150)

coll.place(x=50,y=185)

sub.place(x=150,y=260)

s=float(checkings\_amt)

s-=wdrw

cursor = mydb.cursor()

sql = "UPDATE cust\_details SET amt\_checkings = %s WHERE amt\_checkings = %s "

val = (s, checkings\_amt)

cursor.execute(sql,val)

mydb.commit()

cursor.close()

sub\_btn=tk.Button(wid,font=fonts,text = 'Submit', command = submit)

sub\_btn.place(x=240,y=70)

sub=tk.Button(wid,font=fonts,text = 'Go back', command = back)

wid.mainloop()

def deposit():

global name,start,Ist

global savings\_amt,checkings\_amt,userid,num

n=0

c.destroy()

dep=tk.Tk()

dep.title("DEPOSIT")

dep.geometry("450x350")

def exit():

if messagebox.askokcancel("Quit", "Are you sure?"):

dep.destroy()

dpst=0

def back():

dep.destroy()

starting()

dep.protocol("WM\_DELETE\_WINDOW", back)

v=tk.Label(dep,font=fonts,text=Ist["val not"])

if(dpst>50000 or dpst==0): #validate input

d=tk.Label(dep,font=fonts,text=Ist["amountd"])

d.place(x=0,y=0)

a=tk.Entry(dep)

a.place(x=70,y=70)

def submit1():

if messagebox.askyesno("Proceed to deposit", "Are you sure"):

global name,start,Ist

global savings\_amt,checkings\_amt,num,check

dpst=int(a.get())

if(dpst>50000 or dpst==0):

v.place(x=20,y=120)

if(dpst<50000 and dpst!=0):

v.destroy()

ins=tk.Label(dep,font=fonts,text=Ist["insert"])

wait=tk.Label(dep,font=fonts,text=Ist["wait"])

ins.place(x=50,y=150)

wait.place(x=10,y=180)

for i in range(30000000):i+=1 #time delay

txt="Rs{}. has been deposited in your Account.".format(dpst)

final=tk.Label(dep,font=fonts,text=txt)

final.place(x=15,y=210)

sub.place(x=150,y=260)

mydb = mysql.connector.connect(

host="localhost",

user="root",

password="sai!@#2345",

database="ATM"

)

mycursor = mydb.cursor()

if check=="c":

s=float(checkings\_amt)

s+=dpst

sql = "UPDATE cust\_details SET amt\_checkings = %s WHERE amt\_checkings = %s "

val = (s, checkings\_amt)

mycursor.execute(sql,val)

mydb.commit()

if check=="s":

s=float(savings\_amt)

s+=dpst

sql = "UPDATE cust\_details SET amt\_checkings = %s WHERE amt\_checkings = %s "

val = (s, savings\_amt)

mycursor.execute(sql,val)

mydb.commit()

cursor.close()

sub\_btn=tk.Button(dep,font=fonts,text = 'Submit', command = submit1)

sub\_btn.place(x=240,y=70)

sub=tk.Button(dep,font=fonts,text = 'Go back', command = back)

dep.mainloop()

def exit1():

if messagebox.askokcancel("Quit", "Are you sure?"):

c.destroy()

fonts = ('Helvetica',14)

w = tk.Button(c,font=fonts,text = 'Withdraw',command=withdraw)

d = tk.Button(c,font=fonts,text ='Deposit',command=deposit)

e = tk.Button(c,font=fonts,text ='Exit',command=exit1)

ids.place(x=100,y=10)

total.place(x=30,y=40)

w.place(x=135,y=80)

d.place(x=145,y=130)

e.place(x=160,y=180)

c.mainloop()

def starting():

s=tk.Tk()

fonts = ('Helvetica',16)

s.title("PESU BANK ATM SYSTEM")

s.geometry("400x250")

def exit():

if messagebox.askokcancel("Quit", "Are you sure?"):

s.destroy()

s.protocol("WM\_DELETE\_WINDOW", exit)

def join1():

global check

check="c"

s.destroy()

wde()

def join2():

global check

check="s"

s.destroy()

wde()

hello=tk.Label(s,text=("Hello "+name),font=fonts)

print\_type=tk.Label(s,font=fonts,text=start["print type"])

fonts = ('Helvetica',16)

c = tk.Button(s,font=fonts,text = 'Checkings Acc',command=join1)

d = tk.Button(s,font=fonts,text ='Savings Acc',command=join2)

e = tk.Button(s,font=fonts,text ='Exit',command=exit)

hello.place(x=135,y=10)

print\_type.place(x=30,y=50)

c.place(x=120,y=90)

d.place(x=125,y=140)

e.place(x=155,y=190)

s.mainloop()

starting()

The link to the rest of the modules and main module( to run the python program we need to start from app\_gui.py) is given below:

<https://pastebin.com/xm1v8xAd>

**TESTING :**

We assume the account details of a person for test cases and then predict the expected output by taking values from the user (in tkinter) .Here we first get the user’s name from the facial recognition page and then get their account number and details for further modifications.

**TEST CASES :**

1. **INPUT** : Checkings Account (Button pressed), Withdraw (Button pressed), val=10000

**USERNAME** : Tej (After logging in using facial recognition)

**ACC DETAILS** : 202123 , 10000 (savings account), 40000 (checkings account) (From sql database)

**New value=30000 (expected) , Got 30000**

**NEW ACC DETAILS** : 202123 , 10000 (savings account), 30000 (checkings account) , From sql database

1. **INPUT** : Savings Account (Button pressed), Deposit (Button pressed), val=6000

**USERNAME** : Tejas (After logging in using facial recognition)

**ACC DETAILS** : 202104 , 20000 (savings account), 30000 (checkings account) (From sql database)

**New value=26000 (expected) , Got 26000**

**NEW ACC DETAILS** : 202123 , 26000 (savings account), 30000 (checkings account) , From sql database

1. **INPUT** : Savings Account (Button pressed), Deposit (Button pressed), val=60000

**USERNAME** : Tejas (After logging in using facial recognition)

**ACC DETAILS** : 202104 , 26000 (savings account), 30000 (checkings account) (From sql database)

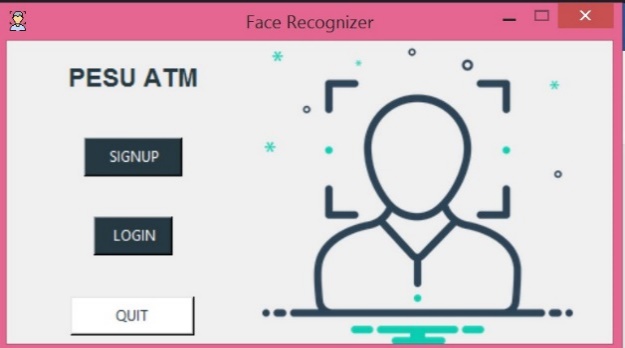
**New value=86000 (expected),Got “ Value not accepted. Please enter value again.”**

**New val=34000**

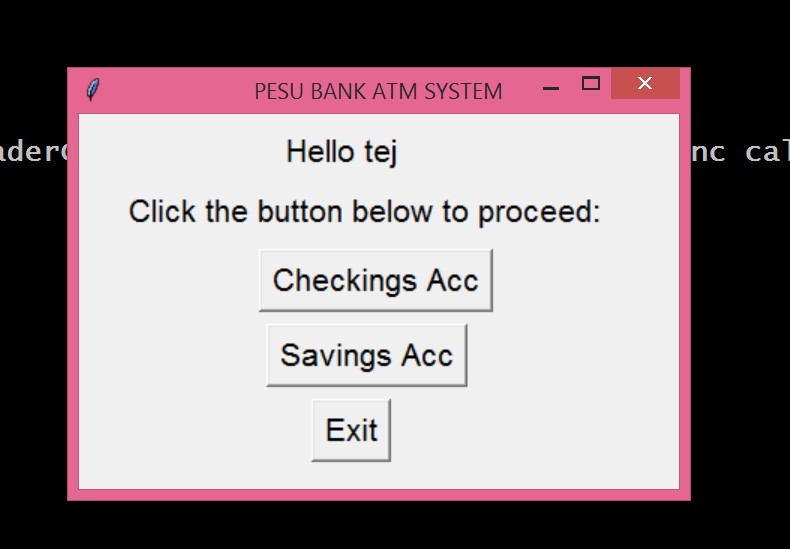
**New value=60000 (expected) , Got 60000**

**NEW ACC DETAILS** : 202123 , 60000 (savings account), 30000 (checkings account) , From sql database

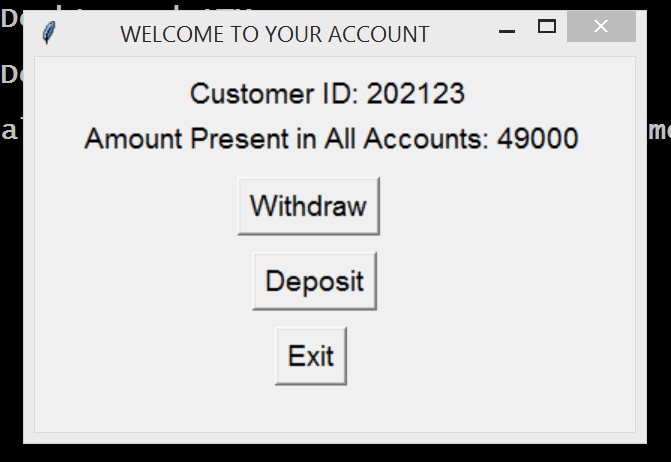
**RESULT/ANAYLSIS :**



**Front page**



**Introductory menu**



**Account details and ATM execution**



**Withdraw menu**

**CONCLUSIONS :**

Taking the idea of security and user-system interaction into account our program directly tackles such a problem by providing facial recognition as a security measure. Our program also solves another problem of banking apps which is their messy and clunky design by providing a simple and efficient interface. Our program has been carefully thought out for a period of 3 months and its usage involves many complex modules like Tkinter , random , OpenCV etc . By creating such a program our knowledge in python has expanded by a lot.

This whole project was possible due to my faculty Prof. Navya ma’am who guided us throughout this journey of exploring and learning new concepts.

**FUTURE ENHANCEMENTS :**

We are planning to continue to work on this project in the future by adding feature such

data encryption, database biometric scan , security pin code etc. This will help the security and user-system interaction more secure and make this model an advanced digitalized system for future possibilities.