**BANKING SYSTEM**

**IN**

**PYTHON**

****

**Submitted by:**

**ANIMESH 18BT010407**

**HAPPY RANA 18BT010423**

**SHAGUN PANDIT 18BT010445**

**Under the Supervision of:**

**Anchal Singh Manhas**

**JAWAHARLAL NEHRU** **GOVT. ENGG. COLLEGE**

**MANDI(H.P.)**

**10 JULY | 2021**

**DECLARATION**

I , Shagun pandit , student of ‘Bachelor of Technology, session: 2021-2021, Jawaharlal nehru Govt. Engg. College Mandi (HP), hereby declare that the work presented in this Project Work entitled **Banking System** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Shagun pandit

Roll no. 18BT010445

**Date:**10thJuly, 2021

**Place:** Solitaire Infosys

**CERTIFICATE**

This is to certify that the work embodies in this dissertation entitled **Banking System** being submitted by **Shagun** Roll No. - **18BT010445** for partial fulfillment of the requirement for the award of Bachelor of Engineering Jawaharlal nehru Govt. Engg. College ,Mandi (HP) during the academic year 2021-2021 is a record of bonafide piece of work, undertaken by him/her the supervision of the undersigned.

**Anchal Singh Manhas**

**(Python &DSA Trainer)**

**INTRODUCTION:**

A bank is a [financial institution](https://en.wikipedia.org/wiki/Financial_institution) that accepts [deposits](https://en.wikipedia.org/wiki/Deposit_account) from the public and creates [credit](https://en.wikipedia.org/wiki/Demand_deposit). Lending activities can be performed either directly or indirectly through [capital markets](https://en.wikipedia.org/wiki/Capital_market). Due to their importance in the financial stability of a country, banks are [highly regulated](https://en.wikipedia.org/wiki/Banking_regulation) in most countries. Most nations have institutionalized a system known as [fractional reserve banking](https://en.wikipedia.org/wiki/Fractional_reserve_banking) under which banks hold liquid assets equal to only a portion of their current liabilities. In addition to other regulations intended to ensure liquidity, banks are generally subject to [minimum capital requirements](https://en.wikipedia.org/wiki/Minimum_capital_requirement) based on an international set of capital standards, known as the [Basel Accords](https://en.wikipedia.org/wiki/Basel_Accords).

Banking in its modern sense evolved in the 14th century in the prosperous cities of [Renaissance Italy](https://en.wikipedia.org/wiki/Renaissance_Italy) but in many ways was a continuation of ideas and concepts of [credit](https://en.wikipedia.org/wiki/Credit_(finance)) and [lending](https://en.wikipedia.org/wiki/Lending) that had their roots in the [ancient world](https://en.wikipedia.org/wiki/Ancient_world). In the [history of banking](https://en.wikipedia.org/wiki/History_of_banking), a number of [banking dynasties](https://en.wikipedia.org/wiki/List_of_banking_families) – notably, the [Medicis](https://en.wikipedia.org/wiki/House_of_Medici), the [Fuggers](https://en.wikipedia.org/wiki/Fugger), the [Welsers](https://en.wikipedia.org/wiki/Welser), the [Berenbergs](https://en.wikipedia.org/wiki/Berenberg_family), and the [Rothschilds](https://en.wikipedia.org/wiki/Rothschild_family) – have played a central role over many centuries. The [oldest existing](https://en.wikipedia.org/wiki/List_of_oldest_banks_in_continuous_operation) [retail bank](https://en.wikipedia.org/wiki/Retail_bank) is [Banca Monte dei Paschi di Siena](https://en.wikipedia.org/wiki/Banca_Monte_dei_Paschi_di_Siena), while the oldest existing [merchant bank](https://en.wikipedia.org/wiki/Merchant_bank) is [Berenberg Bank](https://en.wikipedia.org/wiki/Berenberg_Bank).

**SYSTEM SPECIFICATION**

**Hardware Requirements:**

1. Processor – core i7

2. RAM – 8 GB

3. Hard Disk – 40GB

4. Mouse – Standard Mouse

5. Keyboard – Logitech Keyboard

6. Processor Speed – 2.4GHZ

**Software Requirements:**

1. Operating System – Microsoft Windows 10

2. Front-End – Microsoft Visual Studio 2013

3. python

4. pythonqt and sql lite tool

**PROJECT DESCRIPTION**

**Need For The Software:**

Now a days every one very busy in their work. So they feel that the job must be easier so the system is used to reduce their work which is done in the ATM system. Instead of keeping lots of paper into a record or file and it may be missed somewhere so, this system help to keep the record of the customer it also keeps the details of he customer. It is also easy to access.

**Problem description :**

The system mainly used by the bank clients. It reduces the time consumption and lot of paperwork. For any single operation it involves numerous references and updating also takes subsequent changes in other places.

**SYSTEM STUDY AND ANALYSIS**:

**Existing System:**

* The existing system is manual system.
* The manual system is prone to error.
* This system involves a lot of manual entries with the application to perform a desired task.
* Usage of papers and records in the process leads to less efficiently less productivity.
* Increase lots of mistakes while writing in paper.
* Time delay between the user and customer is reduced.
* For this reason the new system in invented.

**Proposed System:**

The system customer transactions, satisfies the requirements of the existing system in full-fledged manner. Through this system, customer can make fast transactions and view the last transactions easily.

**Feasibility Study:**

**Technology**: This system is technically feasible, because the system activated by computers and recent technology. We use client / server technology which is powerful and very user friendly.

**Finance**: It is financially feasible. There is no need of spending over money. Mainly this system constructed by existing devices only. Since we use visual studio dot net as a front-end it was most power-full, small and portable across platforms and operating systems both at the source and at the binary level. This project reduces the number of workers wage also.

**Time**: This system really time-to-market beat the competition. Because the system developed with in a time span and worked based on time event. The time taken to access the account is very less and avoids unnecessary waiting that was in the traditional system. Although it uses less time but its performance is very well.

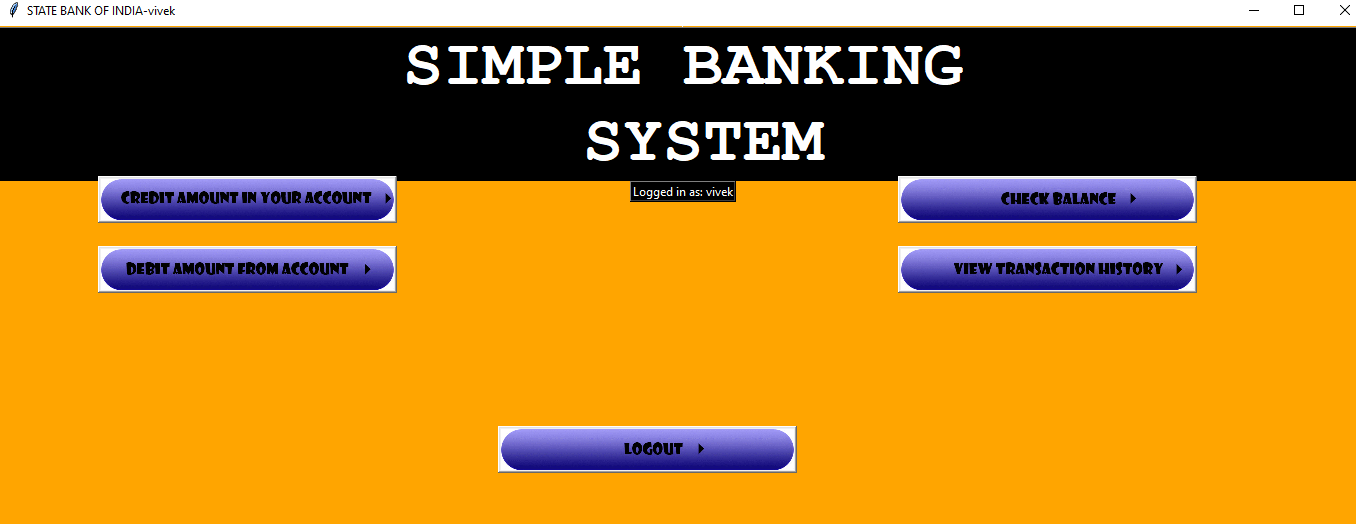
**Code:**

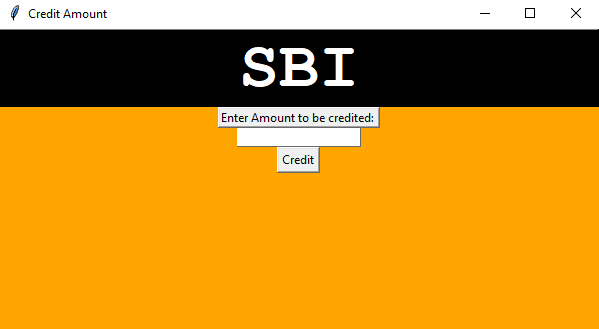
**import** tkinter **as** tk  
**from** tkinter **import** messagebox  
**from** time **import** gmtime, strftime  
  
  
**def** is\_number(s):  
 **try**:  
 float(s)  
 **return** 1  
 **except** ValueError:  
 **return** 0  
  
**def** check\_acc\_nmb(num):  
 **try**:  
 fpin=open(num+**".txt"**,**'r'**)  
 **except** FileNotFoundError:  
 messagebox.showinfo(**"Error"**,**"Invalid Credentials!\nTry Again!"**)  
 **return** 0  
 fpin.close()  
 **return   
  
def** home\_return(master):  
 master.destroy()  
 Main\_Menu()  
  
**def** write(master,name,oc,pin):  
   
 **if**( (is\_number(name)) **or** (is\_number(oc)==0) **or** (is\_number(pin)==0)**or** name==**""**):  
 messagebox.showinfo(**"Error"**,**"Invalid Credentials\nPlease try again."**)  
 master.destroy()  
 **return** f1=open(**"Accnt\_Record.txt"**,**'r'**)  
 accnt\_no=int(f1.readline())  
 accnt\_no+=1  
 f1.close()  
  
 f1=open(**"Accnt\_Record.txt"**,**'w'**)  
 f1.write(str(accnt\_no))  
 f1.close()  
  
 fdet=open(str(accnt\_no)+**".txt"**,**"w"**)  
 fdet.write(pin+**"\n"**)  
 fdet.write(oc+**"\n"**)  
 fdet.write(str(accnt\_no)+**"\n"**)  
 fdet.write(name+**"\n"**)  
 fdet.close()  
  
 frec=open(str(accnt\_no)+**"-rec.txt"**,**'w'**)  
 frec.write(**"Date Credit Debit Balance\n"**)  
 frec.write(str(strftime(**"[%Y-%m-%d] [%H:%M:%S] "**,gmtime()))+**" "**+oc+**" "**+oc+**"\n"**)  
 frec.close()  
   
 messagebox.showinfo(**"Details"**,**"Your Account Number is:"**+str(accnt\_no))  
 master.destroy()  
 **return  
  
def** crdt\_write(master,amt,accnt,name):  
  
 **if**(is\_number(amt)==0):  
 messagebox.showinfo(**"Error"**,**"Invalid Credentials\nPlease try again."**)  
 master.destroy()  
 **return** fdet=open(accnt+**".txt"**,**'r'**)  
 pin=fdet.readline()  
 camt=int(fdet.readline())  
 fdet.close()  
 amti=int(amt)  
 cb=amti+camt  
 fdet=open(accnt+**".txt"**,**'w'**)  
 fdet.write(pin)  
 fdet.write(str(cb)+**"\n"**)  
 fdet.write(accnt+**"\n"**)  
 fdet.write(name+**"\n"**)  
 fdet.close()  
 frec=open(str(accnt)+**"-rec.txt"**,**'a+'**)  
 frec.write(str(strftime(**"[%Y-%m-%d] [%H:%M:%S] "**,gmtime()))+**" "**+str(amti)+**" "**+str(cb)+**"\n"**)  
 frec.close()  
 messagebox.showinfo(**"Operation Successfull!!"**,**"Amount Credited Successfully!!"**)  
 master.destroy()  
 **return  
  
def** debit\_write(master,amt,accnt,name):  
  
 **if**(is\_number(amt)==0):  
 messagebox.showinfo(**"Error"**,**"Invalid Credentials\nPlease try again."**)  
 master.destroy()  
 **return** fdet=open(accnt+**".txt"**,**'r'**)  
 pin=fdet.readline()  
 camt=int(fdet.readline())  
 fdet.close()  
 **if**(int(amt)>camt):  
 messagebox.showinfo(**"Error!!"**,**"You dont have that amount left in your account\nPlease try again."**)  
 **else**:  
 amti=int(amt)  
 cb=camt-amti  
 fdet=open(accnt+**".txt"**,**'w'**)  
 fdet.write(pin)  
 fdet.write(str(cb)+**"\n"**)  
 fdet.write(accnt+**"\n"**)  
 fdet.write(name+**"\n"**)  
 fdet.close()  
 frec=open(str(accnt)+**"-rec.txt"**,**'a+'**)  
 frec.write(str(strftime(**"[%Y-%m-%d] [%H:%M:%S] "**,gmtime()))+**" "**+**" "**+str(amti)+**" "**+str(cb)+**"\n"**)  
 frec.close()  
 messagebox.showinfo(**"Operation Successfull!!"**,**"Amount Debited Successfully!!"**)  
 master.destroy()  
 **return  
  
def** Cr\_Amt(accnt,name):  
 creditwn=tk.Tk()  
 creditwn.geometry(**"600x300"**)  
 creditwn.title(**"Credit Amount"**)  
 creditwn.configure(bg=**"orange"**)  
 fr1=tk.Frame(creditwn,bg=**"blue"**)  
 l\_title=tk.Message(creditwn,text=**"SBI"**,relief=**"raised"**,width=1000,padx=600,pady=0,fg=**"white"**,bg=**"black"**,justify=**"center"**,anchor=**"center"**)  
 l\_title.config(font=(**"Courier"**,**"50"**,**"bold"**))  
 l\_title.pack(side=**"top"**)  
 l1=tk.Label(creditwn,relief=**"raised"**,text=**"Enter Amount to be credited: "**)  
 e1=tk.Entry(creditwn,relief=**"raised"**)  
 l1.pack(side=**"top"**)  
 e1.pack(side=**"top"**)  
 b=tk.Button(creditwn,text=**"Credit"**,relief=**"raised"**,command=**lambda**:crdt\_write(creditwn,e1.get(),accnt,name))  
 b.pack(side=**"top"**)  
 creditwn.bind(**"<Return>"**,**lambda** x:crdt\_write(creditwn,e1.get(),accnt,name))  
  
  
**def** De\_Amt(accnt,name):  
 debitwn=tk.Tk()  
 debitwn.geometry(**"600x300"**)  
 debitwn.title(**"Debit Amount"**)   
 debitwn.configure(bg=**"orange"**)  
 fr1=tk.Frame(debitwn,bg=**"blue"**)  
 l\_title=tk.Message(debitwn,text=**"SBI"**,relief=**"raised"**,width=2000,padx=600,pady=0,fg=**"white"**,bg=**"black"**,justify=**"center"**,anchor=**"center"**)  
 l\_title.config(font=(**"Courier"**,**"50"**,**"bold"**))  
 l\_title.pack(side=**"top"**)  
 l1=tk.Label(debitwn,relief=**"raised"**,text=**"Enter Amount to be debited: "**)  
 e1=tk.Entry(debitwn,relief=**"raised"**)  
 l1.pack(side=**"top"**)  
 e1.pack(side=**"top"**)  
 b=tk.Button(debitwn,text=**"Debit"**,relief=**"raised"**,command=**lambda**:debit\_write(debitwn,e1.get(),accnt,name))  
 b.pack(side=**"top"**)  
 debitwn.bind(**"<Return>"**,**lambda** x:debit\_write(debitwn,e1.get(),accnt,name))  
  
  
  
  
**def** disp\_bal(accnt):  
 fdet=open(accnt+**".txt"**,**'r'**)  
 fdet.readline()  
 bal=fdet.readline()  
 fdet.close()  
 messagebox.showinfo(**"Balance"**,bal)  
  
  
  
  
**def** disp\_tr\_hist(accnt):  
 disp\_wn=tk.Tk()  
 disp\_wn.geometry(**"900x600"**)  
 disp\_wn.title(**"Transaction History"**)  
 disp\_wn.configure(bg=**"orange"**)  
 fr1=tk.Frame(disp\_wn,bg=**"blue"**)  
 l\_title=tk.Message(disp\_wn,text=**"SBI"**,relief=**"raised"**,width=2000,padx=600,pady=0,fg=**"white"**,bg=**"black"**,justify=**"center"**,anchor=**"center"**)  
 l\_title.config(font=(**"Courier"**,**"50"**,**"bold"**))  
 l\_title.pack(side=**"top"**)  
 fr1=tk.Frame(disp\_wn)  
 fr1.pack(side=**"top"**)  
 l1=tk.Message(disp\_wn,text=**"Your Transaction History:"**,padx=100,pady=20,width=1000,bg=**"blue"**,fg=**"orange"**,relief=**"raised"**)  
 l1.pack(side=**"top"**)  
 fr2=tk.Frame(disp\_wn)  
 fr2.pack(side=**"top"**)  
 frec=open(accnt+**"-rec.txt"**,**'r'**)  
 **for** line **in** frec:  
 l=tk.Message(disp\_wn,anchor=**"w"**,text=line,relief=**"raised"**,width=2000)  
 l.pack(side=**"top"**)  
 b=tk.Button(disp\_wn,text=**"Quit"**,relief=**"raised"**,command=disp\_wn.destroy)  
 b.pack(side=**"top"**)  
 frec.close()  
  
**def** logged\_in\_menu(accnt,name):  
 rootwn=tk.Tk()  
 rootwn.geometry(**"1600x500"**)  
 rootwn.title(**"STATE BANK OF INDIA-"**+name)  
 rootwn.configure(background=**'orange'**)  
 fr1=tk.Frame(rootwn)  
 fr1.pack(side=**"top"**)  
 l\_title=tk.Message(rootwn,text=**"SIMPLE BANKING\n SYSTEM"**,relief=**"raised"**,width=2000,padx=600,pady=0,fg=**"white"**,bg=**"black"**,justify=**"center"**,anchor=**"center"**)  
 l\_title.config(font=(**"Courier"**,**"50"**,**"bold"**))  
 l\_title.pack(side=**"top"**)  
 label=tk.Label(text=**"Logged in as: "**+name,relief=**"raised"**,bg=**"black"**,fg=**"white"**,anchor=**"center"**,justify=**"center"**)  
 label.pack(side=**"top"**)  
 img2=tk.PhotoImage(file=**"credit.gif"**)  
 myimg2=img2.subsample(2,2)  
 img3=tk.PhotoImage(file=**"debit.gif"**)  
 myimg3=img3.subsample(2,2)  
 img4=tk.PhotoImage(file=**"balance1.gif"**)  
 myimg4=img4.subsample(2,2)  
 img5=tk.PhotoImage(file=**"transaction.gif"**)  
 myimg5=img5.subsample(2,2)  
 b2=tk.Button(image=myimg2,command=**lambda**: Cr\_Amt(accnt,name))  
 b2.image=myimg2  
 b3=tk.Button(image=myimg3,command=**lambda**: De\_Amt(accnt,name))  
 b3.image=myimg3  
 b4=tk.Button(image=myimg4,command=**lambda**: disp\_bal(accnt))  
 b4.image=myimg4  
 b5=tk.Button(image=myimg5,command=**lambda**: disp\_tr\_hist(accnt))  
 b5.image=myimg5  
   
 img6=tk.PhotoImage(file=**"logout.gif"**)  
 myimg6=img6.subsample(2,2)  
 b6=tk.Button(image=myimg6,relief=**"raised"**,command=**lambda**: logout(rootwn))  
 b6.image=myimg6  
  
   
 b2.place(x=100,y=150)  
 b3.place(x=100,y=220)  
 b4.place(x=900,y=150)  
 b5.place(x=900,y=220)  
 b6.place(x=500,y=400)  
  
   
**def** logout(master):  
   
 messagebox.showinfo(**"Logged Out"**,**"You Have Been Successfully Logged Out!!"**)  
 master.destroy()  
 Main\_Menu()  
  
**def** check\_log\_in(master,name,acc\_num,pin):  
 **if**(check\_acc\_nmb(acc\_num)==0):  
 master.destroy()  
 Main\_Menu()  
 **return  
  
 if**( (is\_number(name)) **or** (is\_number(pin)==0) ):  
 messagebox.showinfo(**"Error"**,**"Invalid Credentials\nPlease try again."**)  
 master.destroy()  
 Main\_Menu()  
 **else**:  
 master.destroy()  
 logged\_in\_menu(acc\_num,name)  
  
  
**def** log\_in(master):  
 master.destroy()  
 loginwn=tk.Tk()  
 loginwn.geometry(**"600x300"**)  
 loginwn.title(**"Log in"**)  
 loginwn.configure(bg=**"orange"**)  
 fr1=tk.Frame(loginwn,bg=**"blue"**)  
 l\_title=tk.Message(loginwn,text=**"SBI"**,relief=**"raised"**,width=2000,padx=600,pady=0,fg=**"white"**,bg=**"black"**,justify=**"center"**,anchor=**"center"**)  
 l\_title.config(font=(**"Courier"**,**"50"**,**"bold"**))  
 l\_title.pack(side=**"top"**)  
 l1=tk.Label(loginwn,text=**"Enter Name:"**,relief=**"raised"**)  
 l1.pack(side=**"top"**)  
 e1=tk.Entry(loginwn)  
 e1.pack(side=**"top"**)  
 l2=tk.Label(loginwn,text=**"Enter account number:"**,relief=**"raised"**)  
 l2.pack(side=**"top"**)  
 e2=tk.Entry(loginwn)  
 e2.pack(side=**"top"**)  
 l3=tk.Label(loginwn,text=**"Enter your PIN:"**,relief=**"raised"**)  
 l3.pack(side=**"top"**)  
 e3=tk.Entry(loginwn,show=**"\*"**)  
 e3.pack(side=**"top"**)  
 b=tk.Button(loginwn,text=**"Submit"**,command=**lambda**: check\_log\_in(loginwn,e1.get().strip(),e2.get().strip(),e3.get().strip()))  
 b.pack(side=**"top"**)  
 b1=tk.Button(text=**"HOME"**,relief=**"raised"**,bg=**"black"**,fg=**"white"**,command=**lambda**: home\_return(loginwn))  
 b1.pack(side=**"top"**)  
 loginwn.bind(**"<Return>"**,**lambda** x:check\_log\_in(loginwn,e1.get().strip(),e2.get().strip(),e3.get().strip()))  
   
  
**def** Create():  
   
 crwn=tk.Tk()  
 crwn.geometry(**"600x300"**)  
 crwn.title(**"Create Account"**)  
 crwn.configure(bg=**"orange"**)  
 fr1=tk.Frame(crwn,bg=**"blue"**)  
 l\_title=tk.Message(crwn,text=**"SBI"**,relief=**"raised"**,width=2000,padx=600,pady=0,fg=**"white"**,bg=**"black"**,justify=**"center"**,anchor=**"center"**)  
 l\_title.config(font=(**"Courier"**,**"50"**,**"bold"**))  
 l\_title.pack(side=**"top"**)  
 l1=tk.Label(crwn,text=**"Enter Name:"**,relief=**"raised"**)  
 l1.pack(side=**"top"**)  
 e1=tk.Entry(crwn)  
 e1.pack(side=**"top"**)  
 l2=tk.Label(crwn,text=**"Enter opening credit:"**,relief=**"raised"**)  
 l2.pack(side=**"top"**)  
 e2=tk.Entry(crwn)  
 e2.pack(side=**"top"**)  
 l3=tk.Label(crwn,text=**"Enter desired PIN:"**,relief=**"raised"**)  
 l3.pack(side=**"top"**)  
 e3=tk.Entry(crwn,show=**"\*"**)  
 e3.pack(side=**"top"**)  
 b=tk.Button(crwn,text=**"Submit"**,command=**lambda**: write(crwn,e1.get().strip(),e2.get().strip(),e3.get().strip()))  
 b.pack(side=**"top"**)  
 crwn.bind(**"<Return>"**,**lambda** x:write(crwn,e1.get().strip(),e2.get().strip(),e3.get().strip()))  
 **return  
  
  
def** Main\_Menu():  
   
  
 rootwn=tk.Tk()  
 rootwn.geometry(**"1600x500"**)  
 rootwn.title(**"SBI"**)  
 rootwn.configure(background=**'orange'**)  
 fr1=tk.Frame(rootwn)  
 fr1.pack(side=**"top"**)  
 bg\_image = tk.PhotoImage(file =**"pile1.gif"**)  
 x = tk.Label (image = bg\_image)  
 x.place(y=-400)  
 l\_title=tk.Message(text=**"SIMPLE BANKING\n SYSTEM"**,relief=**"raised"**,width=2000,padx=600,pady=0,fg=**"white"**,bg=**"black"**,justify=**"center"**,anchor=**"center"**)  
 l\_title.config(font=(**"Courier"**,**"50"**,**"bold"**))  
 l\_title.pack(side=**"top"**)  
 imgc1=tk.PhotoImage(file=**"new.gif"**)  
 imglo=tk.PhotoImage(file=**"login.gif"**)  
 imgc=imgc1.subsample(2,2)  
 imglog=imglo.subsample(2,2)  
  
 b1=tk.Button(image=imgc,command=Create)  
 b1.image=imgc  
 b2=tk.Button(image=imglog,command=**lambda**: log\_in(rootwn))  
 b2.image=imglog  
 img6=tk.PhotoImage(file=**"quit.gif"**)  
 myimg6=img6.subsample(2,2)  
  
 b6=tk.Button(image=myimg6,command=rootwn.destroy)  
 b6.image=myimg6  
 b1.place(x=800,y=300)  
 b2.place(x=800,y=200)   
 b6.place(x=920,y=400)  
  
 rootwn.mainloop()  
  
Main\_Menu()

**Output:**

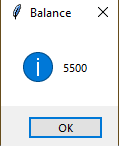
****

****

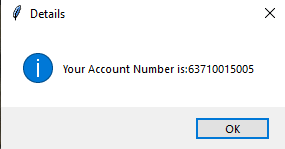
****

****

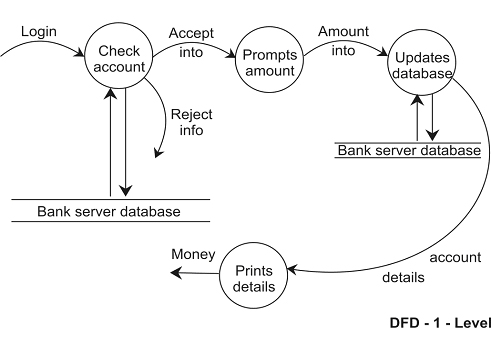
****

****

****

****

**Flow Chart:**

****

**References:**

* [www.google.com](http://www.google.com)
* internshala trainings
* youtube