BANKING ASSISTANCE SYSTEM

SUMMER INTERNSHIP PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF COMPUTER APPLICATION

By

MADHAV SHARMA

ENROLLMENT NO.: 04990202020

Guided by

Mrs. Jaspreet Kaur

Asst. Professor

Sri Guru Tegh Bahadur Institute of Management and Information Technology, DELHI – 110033

(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)

September,2022

DECLARATION

I hereby declare that the work, which is being presented in this project entitled "BANKING ASSISTANCE SYSTEM", is an authentic record of my own work carried out during a period from JUNE 2022 to AUGUST 2022 under the supervision and guidance of MR.SANJAY UPADHYAY project guide, TECH ACCESS LEARNING PVT LTD. This project was undertaken as a part of the summer internship project report as per the curriculum of GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI for the partial fulfillment of BCA from Sri Guru Tegh Bahadur Institute of Management and Information Technology.

The matter embodied here in this project has not been submitted by me for the award of any other Degree/Diploma.

MADHAV SHARMA

04990202020

CERTIFICATE

TECH ACCESS

TECH ACCESS LEARNING PVT LTD
Address: A-12, FIEE, Okhla Phase-II, New Delhi-20
Mobile: 9643320067/85/89
Email: info@techaccess.co.in
Website: www.techaccess.co.in

Dated: 25 August, 2022

This is to certify to Madhav Sharma S/O Mr. Ramesh Kumar Sri Guru Tegh Bahadur Institute of Management and Information Technology Roll No.4990202020 has done project training from 17 June,22 to 12 August,22 on the technology "Python Machine Learning" and make project on "Banking Assistance System" under the guidance of Mr. Sanjay Upadhyay.

During his internship, we found Madhav Sharma Honest, Hardworking and Analytical thinker.

We wish his all the best for his future endeavors.

For Tech Access Learning Pvt Ltd.

For Tech Access Learning Pvt. Ltd.

Authorised Signatory

ACKNOWLEDGEMENT

Firstly, I would like to thank MRS. ALIYA, HR, Head, of TECH ACCESS LEARNING PVT LTD. for giving me the opportunity to do an internship within the organization. I also would like all the people that worked along with me at TECH ACCESS LEARNING PVT LTD with their patience and openness they created an enjoyable working environment. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals. I am highly indebted to my project guide MR.SANJAY UPADHYAY, TECH ACCESS LEARNING PVT LTD for his supervision and guidance and for his constructive criticism throughout my internship. I would like to thank MRS. SWATI (TPO) for her support and advices to get and complete internship in above said organization. I am extremely grateful to my department staff members and friends who helped me in successful completion of this internship.

MADHAV SHARMA

04990202020

ABSTRACT

We built our Banking Assistance System with the motive of providing a solution for easy, convenient, secure and user-friendly banking solution to the customers. Our unique viewpoint while working on this project was to provide a banking solution to the customer that they find familiar with. In this project we wanted to create an application that could help the customer experience familiarity with both the atm and net banking system and make it easier for the user to understand the features of both and feel comfortable using both atm and net banking systems. Our objective was to create one single software solution working as both an atm and net banking system. Our software has 9 major modules Login, Signup, Create Account, Withdraw, Deposit, Fast Cash, Mini Statement, Transfer Funds and Balance Enquiry. Each module has specific roles and is properly functioning.

We demonstrate the efficiency and effectiveness of our approach through an experimental evaluation using our implemented prototype.

Organization Information: Tech Access is a professionally managed company with years of industry experience in developing and delivering Enterprise specific Software and Web development solutions using latest technologies. Quality is the buzz word in today's world without which no organization can survive. Along with quality we at Tech Access "Think Beyond" to take one step ahead and focus on Delivery of the solutions. We design processes that focus not just only on quality but also on delivery which increases the value to our clients. Apart from training our employees on latest technologies, we also empower them to deliver exciting solutions to our clients.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Banks are changing the ways of their operational activities by adopting new technologies. Internet banking eases and accelerates banking & financial undertakings. It offers various services and products. Internet banking helps bank customers to do their banking and financial activity flexibly as per their need and ease.

Still, we see line at banks, banks flooded with people for all the small task that can be easily done from their home or ATMs. But still people travel to far away situate bank branches.

This is because most of them do not understand he net banking or are afraid to use the ATM system, many people do not like to visit ATMs because it takes them time to understand and process the transaction, they find it much easier to make a detour to the banking branch and process the transaction.

To ease this problem, we made a one stop solution for people a same interface from which they can use full fledge banking facilities. ATM to all net banking features. Learn it once and use it.

1.1. OBJECTIVE OF PROJECT

Our objective was to create one single software solution working as both an ATM and net banking system. We built our Banking Assistance System with the motive of providing a solution for easy, convenient, secure and user-friendly banking solution to the customers.
While working on this project our objective was to provide a banking solution to the customer that they find familiar with. In this project we wanted to create an application that could help the customer experience familiarity with both the ATM and net banking system and make it easier for the user to understand the features of both and feel comfortable using both ATM and Net banking systems.
We made a one stop solution for people a same interface from which they can use full fledge banking facilities. ATM to all net banking features. Learn it once and use it.

1.2. SCOPE OF PROJECT

In first iteration we launched basic ATM features like deposit withdraw and fast cash , in the next version or we can say second iteration we launched a few net banking features like transfer funds, password resetting(forget password) and in the upcoming release we are planning to release personalized banking features in which the banking assistance system helps the user by providing personalized investment and saving plans, the system uses machine learning algorithms and analysis the user's financial habits and provide a personalized assistance with fund management.

1.3. FEATURES OF PROJECT

Our software has major 9 modules. Each module has specific roles and is properly functioning.

1. Login Module
    a. Forget Password Module
    b. Sign-up Module
2. Create Account Module
3. Main Page Module
    a. Update Contact Module
4. Withdraw Module
5. Deposit Module
6. Fast Cash Module
7. Transfer Funds Module
    a. HDFC Module
    b. PNB Module
    c. Other Banks Module
8. Balance Enquiry Module
9. Mini Statement Module

CHAPTER-2

REQUIREMENT ANALYSIS

2.1. SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

We incorporated the Iterative Model, in which we can initialize the development process with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.

The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.

- The various phases of Iterative model are as follows:
  1. Requirement gathering & analysis: In this phase, requirements are gathered from customers and check by an analyst whether requirements will fulfil or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.

  2. Design: In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.

  3. Implementation: In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.

  4. Testing: After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

  5. Deployment: After completing all the phases, software is deployed to its work environment.

  6. Review: In this phase, after the product deployment, review phase is performed to check the behavior and validity of the developed product. And if there are any error found then the process starts again from the requirement gathering.

  7. Maintenance: In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

- Advantage (Pros) of Iterative Model:
  - Testing and debugging during smaller iteration are easy.
  - A Parallel development can plan.
  - It is easily acceptable to ever-changing needs of the project.
  - Risks are identified and resolved during iteration.
  - Limited time spent on documentation and extra time on designing.

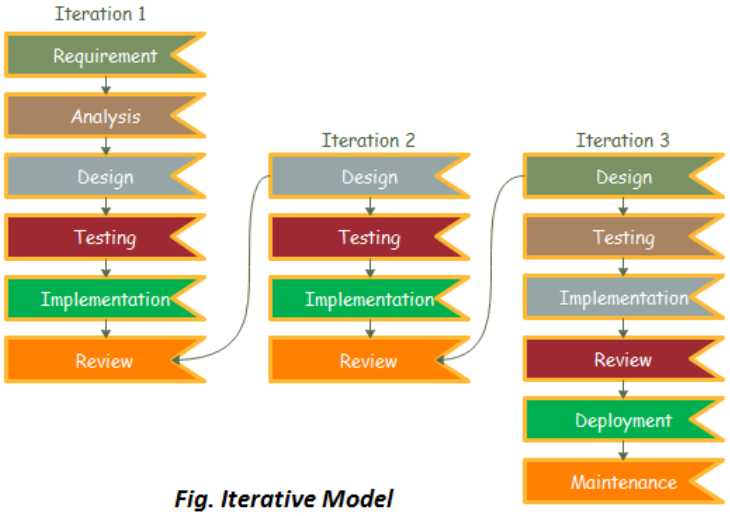


Fig. Iterative Model

Fig-1

I. In first iteration (version) we launched basic ATM features like deposit withdraw and fast cash.

II. In the next version or we can say second iteration we launched a few net banking features like transfer funds, password resetting (forget password).

III. And in the upcoming release we are planning to release personalized banking features in which the banking assistance system helps the user by providing personalized investment and saving plans, the system uses machine learning algorithms and analysis the user's financial habits and provide a personalized assistance with fund management.

2.2. SOFTWARE REQUIREMENT SPECIFICATION

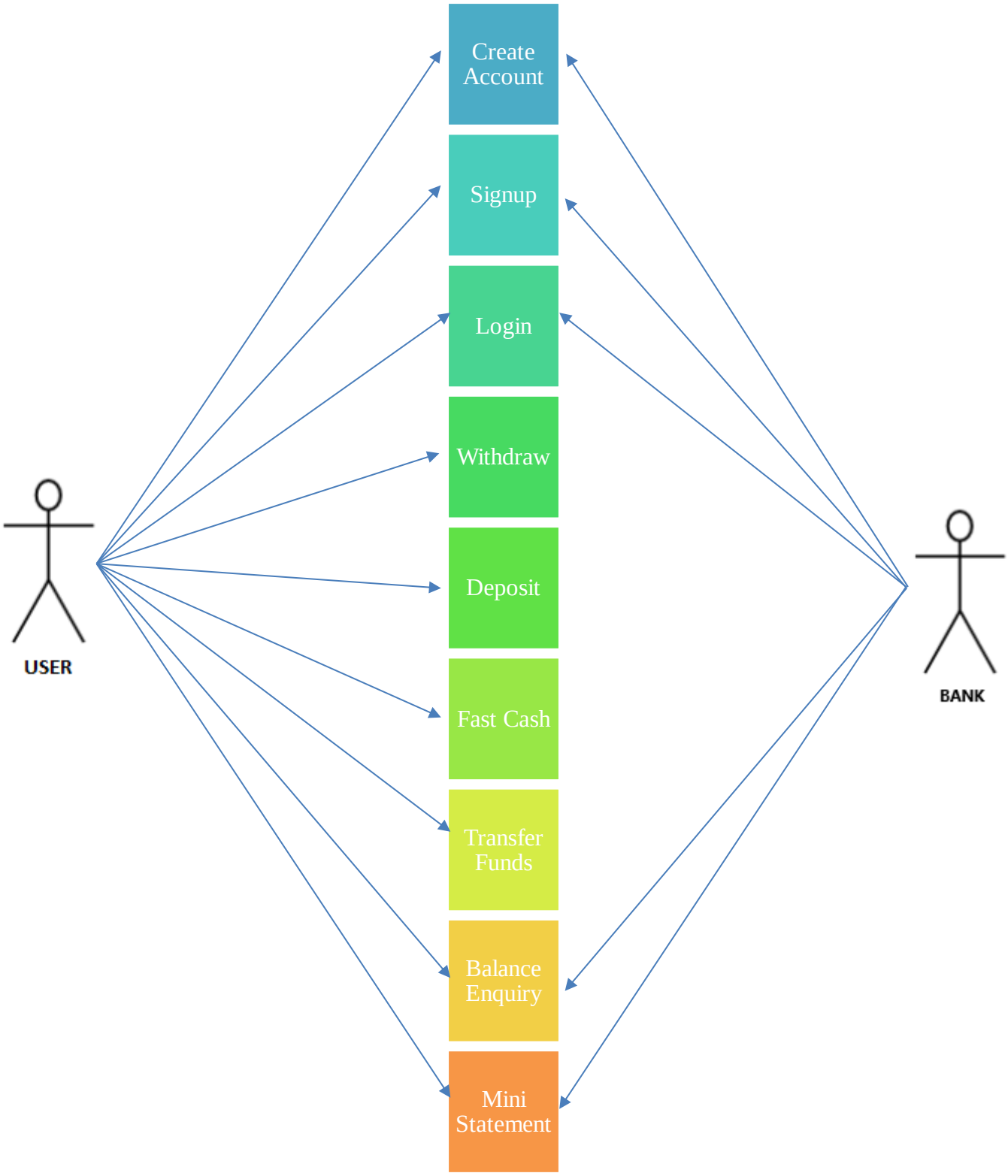
SOFTWARE REQUIREMENTS

- Operating System: Windows 8/10/11, linux , macOS
- Programming Language: Python 3.9.10
- Database: MySQL

HARDWARE REQUIREMENTS

- Device: PC/Laptop 32/64 Bit System
- Storage Size: 128 GB
- Ram: 4GB
- Space On Disc: 2GB
- Processor: Intel core 3/4/5,RYZEN-3

2.3. USE CASE DIAGRAM



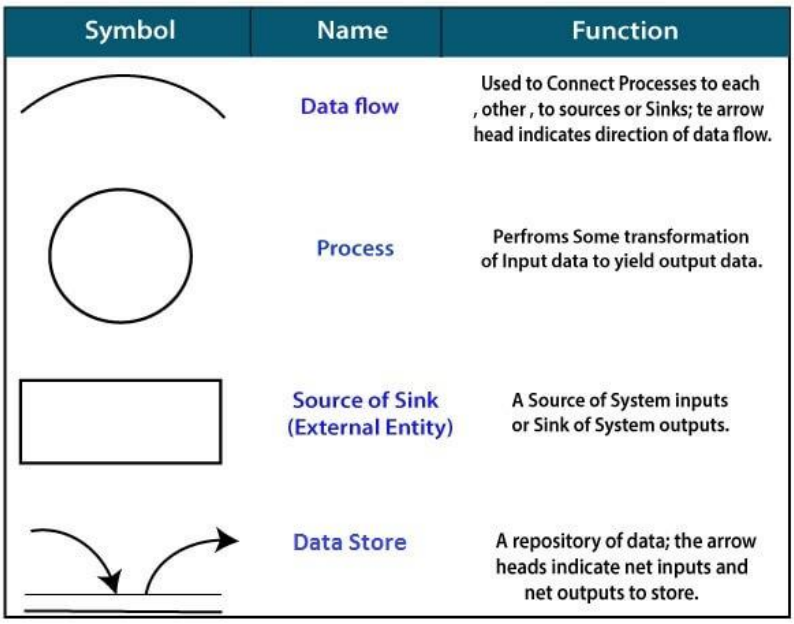
CHAPTER-3

SOFTWARE DESIGN

Software design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the client's requirement, as described in SRS (Software Requirement Specification) document, into a form, i.e., easily implementable using programming language.

Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.
It shows how data enters and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.



Symbols for Data Flow Diagrams

The Data flow diagram can be explained as the separate levels indicating the individuals complexity in the each level of the system and gives a detailed explanation in the further levels that are following them.

LEVEL 0

It is also known as a context diagram .It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

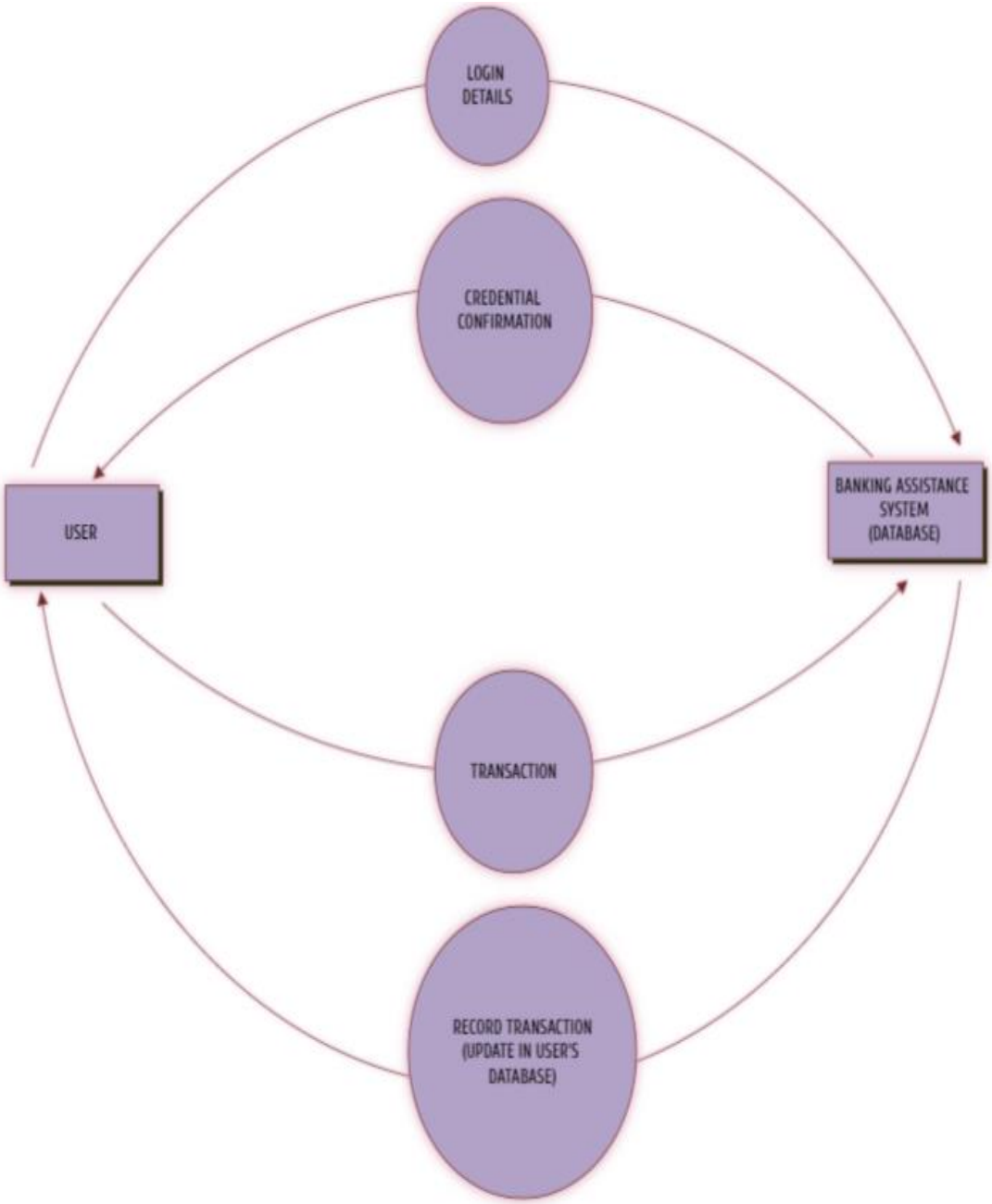
LEVEL 1

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.The level 1 of the Data Flow Diagram gives explain in detail about packet watching system which was marked under as 0 in the previous level.

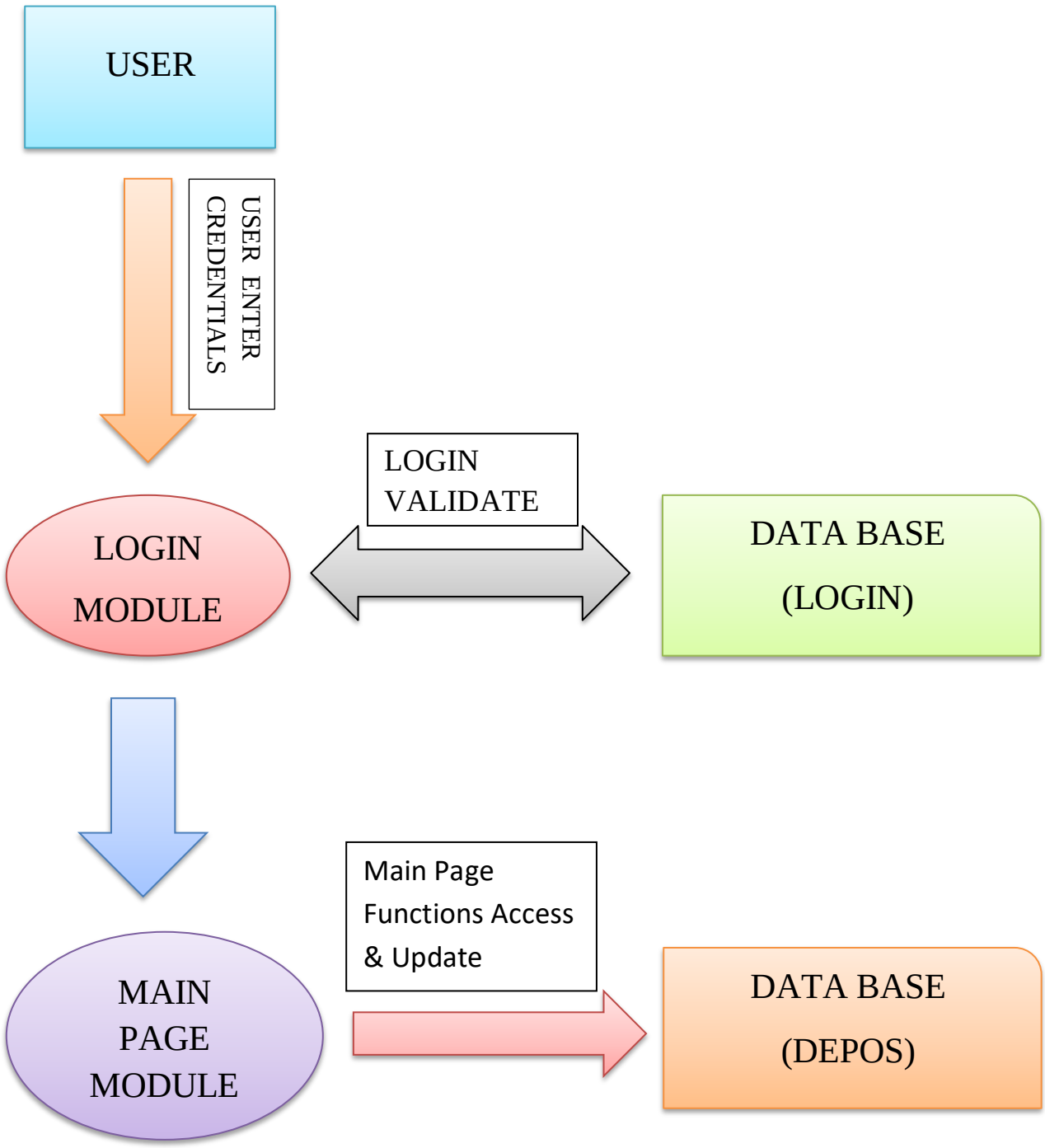
LEVEL 2

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning. This level two data flow diagram (DFD) template can map out information flow, visualize an entire system.

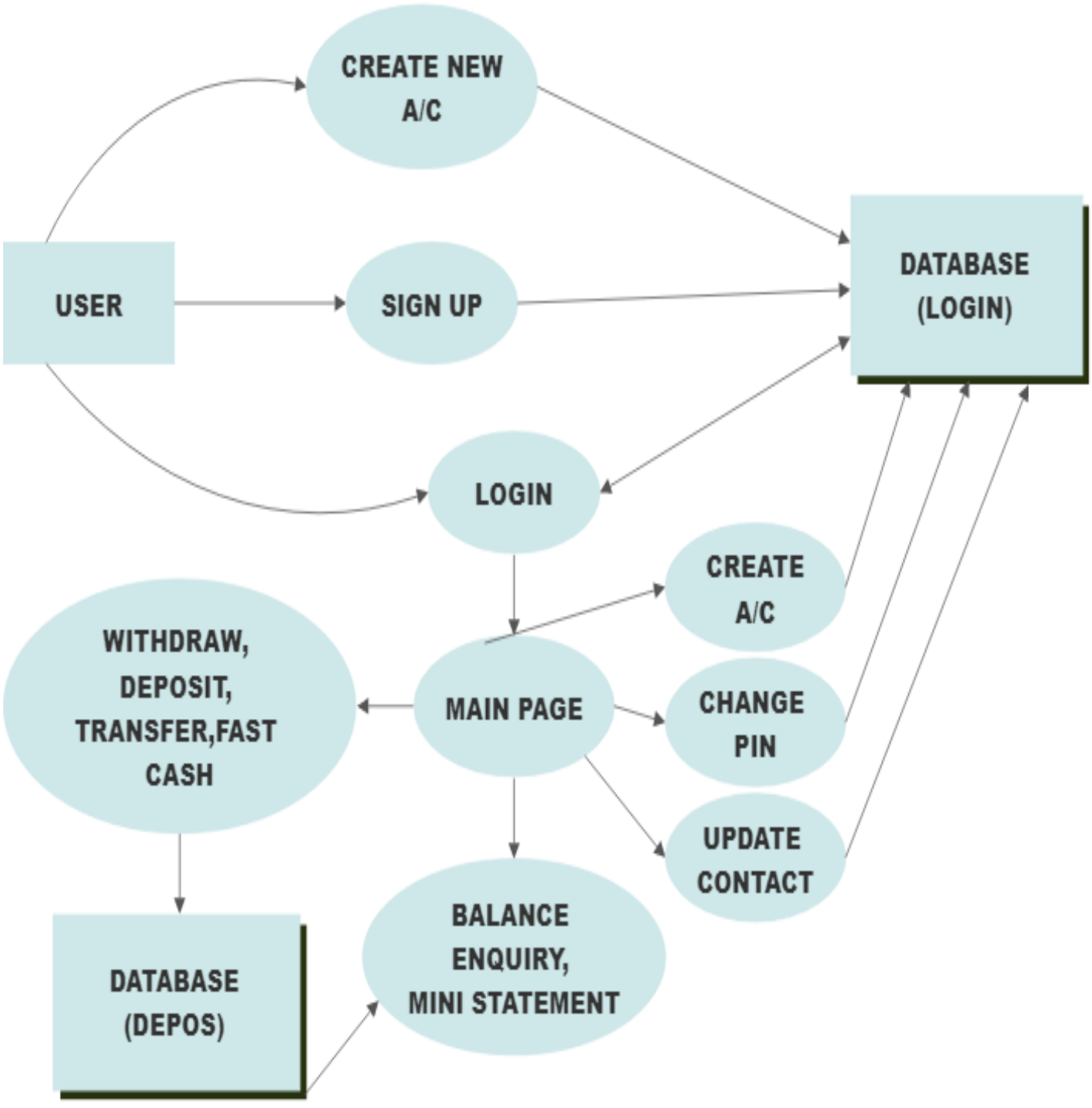
3.1. CONTEXT DIAGRAM (0-LEVEL DFD)



3.2. LEVEL - 1 DFD



3.3. LEVEL-2 DFD



CHAPTER-4

DATABASE DESIGN

4.1. ER DIAGRAM

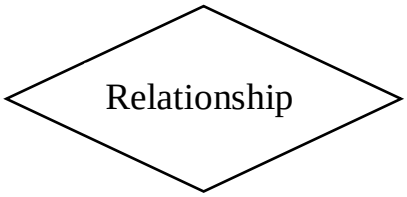
ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

Entities

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

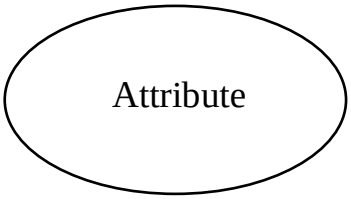
Relationship

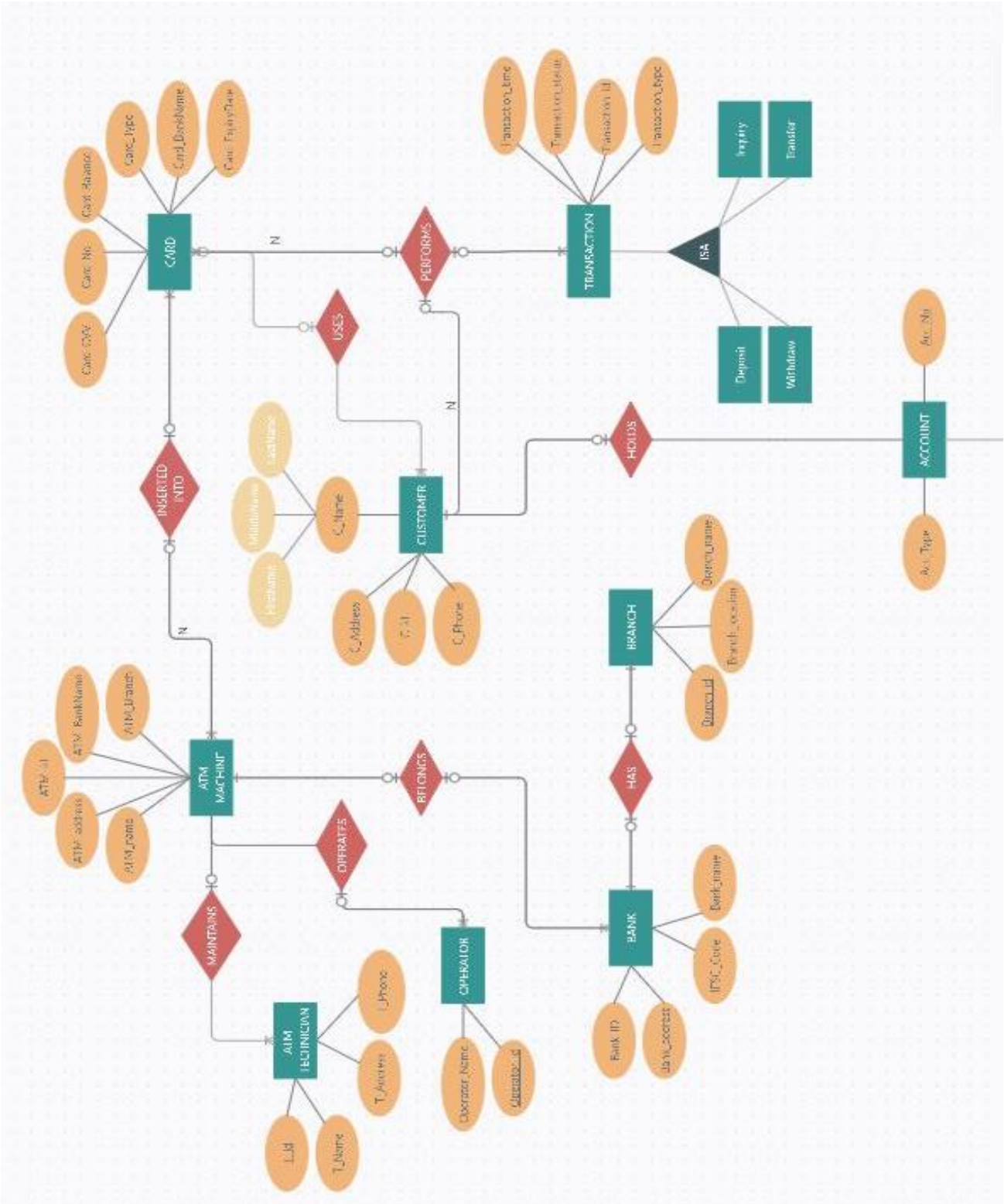
Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box



Attribute

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).




CARD
Card_Balance
Card_Type
Card_BankName
Card_ExpiryDate
Card_No
Card_CVV
PERFORMS
TRANSACTION
Transaction_Time
Transaction_Status
Transaction_ID
Transaction_Type
ISA
Inquiry
Transfer
Deposit
Withdraw
USES
INSERTED INTO
HOLDS
ACCOUNT
Acc_No
Acc_Type
C_Name
C_Address
C_Phone
C_ID
CUSTOMER
ATM MACHINE
ATM_BankName
ATM_Branch
ATM_ID
ATM_address
ATM_name
BRANCH
Branch_name
Branch_location
BranchId
HAS
BANK
Bank_name
IFSC_Code
Bank_ID
Bank_address
MAINTAINS
OPERATES
OPERATOR
Operator_Name
OperatorId
ATM TECHNICIAN
T_Phone
T_Address
T_ID
T_Name

4.2. TABLES DESIGN

```
mysql> select * from login;
+------------+-----------+------------------+-------------+
| first_name | Last_name | username         | password    |
+------------+-----------+------------------+-------------+
| nancy      | kapoor    | nancy@gmail.com  | nancykapoor |
| Rajeev     | Kumar     | rajeev@gmail.com | 123456789   |
+------------+-----------+------------------+-------------+
2 rows in set (0.00 sec)
```

```
mysql> select * from depos;
+-------------------+-----------+--------------+------------+
| Enter_Card_Number | Enter_Pin | Enter_Amount | contact    |
+-------------------+-----------+--------------+------------+
|        1234567890 |      1244 |       800000 | 9810688575 |
|        1122334455 |      2255 |        22900 |   54525455 |
+-------------------+-----------+--------------+------------+
2 rows in set (0.00 sec)
```

```
mysql> select * from bank;
+------------------+------------+--------------+
| Card_Number_To | Enter_Pin | Enter_Amount |
+------------------+------------+--------------+
|      1234567890 |       2255 |          600 |
|      1234567890 |       2255 |          400 |
+------------------+------------+--------------+
2 rows in set (0.00 sec)
```

```
mysql> select * from type;
+-------------------+--------------+-----------------+
| Enter_Card_Number | Enter_Amount | type            |
+-------------------+--------------+-----------------+
|          45454545 |          600 | Withdraw        |
|          45454545 |          600 | Transfer        |
|          45454545 |         2000 | Fast Cash       |
|          45454545 |          200 | Transfer        |
|         780051513 |          200 | Withdraw        |
|         780051513 |          200 | Transfer        |
|         780051513 |         2000 | Fast Cash       |
|          92119211 |          500 | Withdraw        |
|          92119211 |         2000 | Transfer        |
|          92119211 |         2000 | Fast Cash       |
|         329871924 |        50000 | Withdraw        |
|              2255 |          100 | Transfer        |
|        1122334455 |        25000 | Account Created |
|        1122334455 |          200 | Deposit         |
|        1122334455 |          100 | Withdraw        |
|        1122334455 |          200 | Transfer        |
|        1122334455 |         2000 | Fast Cash       |
+-------------------+--------------+-----------------+
17 rows in set (0.00 sec)
```

CHAPTER-5

Testing

5.1. What is Software Testing ?

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.

Testing is mandatory because it will be a dangerous situation if the software fails any of time due to lack of testing. So, without testing software cannot be deployed to the end user.

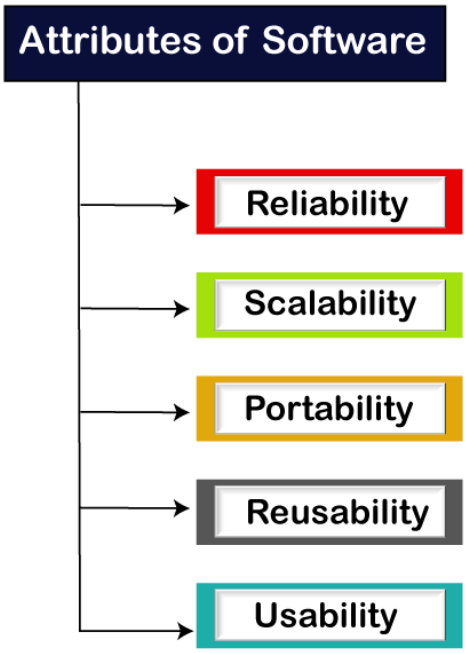


Fig-2

LEVELS OF TESTING

- Unit testing
- Component integration testing
- System testing
- Acceptance testing

Fig-3

5.1.1 Unit testing

A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

5.1.2 Component Integration Testing

A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

5.1.3 System Testing

A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

5.1.4 Acceptance Testing

A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

5.2. Types of Software Testing

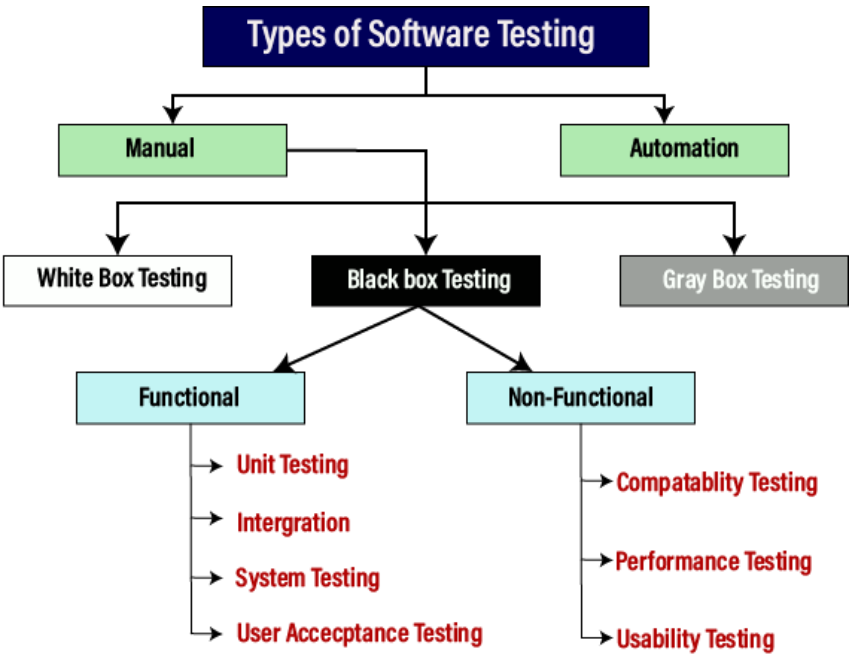


Fig-4

We adopted **BLACK BOX TESTING**, Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

- Types of Black Box Testing

- **Functional Testing** – This black box testing type is related to the functional requirements of a system; it is done by software testers.
- **Non-Functional Testing** – This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.
- **Regression Testing** – Regression Testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

5.3. Test Cases

SNO.	TEST CASE	INPUT DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
LOGIN MODULE					
1.	User Name Text Field	Username Not Matching From Database	Error Message : Please Enter The Correct Username And Password.	Error Message : Please Enter The Correct Username And Password.	Success
		Username Matching From Database	Redirects To Main Operation Page.	Redirected To Main Operation Page.	Success
		Username Matching From Database But Case Not Matching	Error Message : Please Enter The Correct Username And Password.	Error Message : Please Enter The Correct Username And Password.	Success
		User Name Field Blank	Error Message : Please Enter The Correct Username And Password.	Error Message : Please Enter The Correct Username And Password.	Success
2.	Password Text Field	Password Not Matching From Database	Error Message : Please Enter The Correct Username And Password.	Error Message : Please Enter The Correct Username And Password.	Success
		Password Matching From Database	Redirects To Main Operation Page.	Redirected To Main Operation Page.	Success
		Password Matching From Database But Case Not Matching	Error Message : Please Enter The Correct Username And Password.	Error Message : Please Enter The Correct Username And Password.	Success
		Password Field Blank	Error Message : Please Enter The Correct Username And Password.	Error Message : Please Enter The Correct Username And Password.	Success
3.	Forget Password Button	On Click	Redirects To Forget Password Page.	Redirected To Forget Password Page.	Success
4.	Do Not Have An Admin Account Button	On Click	Redirects To Create New Bank Account Page.	Redirected To Create New Bank Account Page.	Success
5.	Sign Up	On Click	Redirects To Create Sign Up Page.	Redirected To Sign Up Page.	Success
MAIN PAGE MODULE					
1.	Deposit Cash Button	On Click	Redirects To Deposit Cash Page.	Redirected To Deposit Cash Page.	Success
2.	Withdraw Cash Button	On Click	Redirects To Withdraw Cash Page.	Redirected To Withdraw Cash Page.	Success
3.	Balance Enquiry Button	On Click	Redirects To Balance Enquiry Page.	Redirected To Deposit Cash Page.	Success
4.	Deposit Cash Button	On Click	Redirects To Deposit Cash Page.	Redirected To Deposit Cash Page.	Success
5.	Transfer Fund Button	On Click	Redirects To Transfer Fund Page.	Redirected To Transfer Fund Page.	Success
6.	Mini Statement Button	On Click	Combs Through Database And Generate A Page of Latest Transactions.	Combed Through Database And Generated A Page of Latest Transactions.	Success

7.	Fast Cash Button	On Click	Redirects To Fast Cash Page.	Redirected To Fast Cash Page.	Success
8.	Change Pin Button	On Click	Redirects To Change Pin Page.	Redirected To Change Pin Page.	Success
9.	Update Contact Button	On Click	Redirects To Update Contact Page.	Redirected To Update Contact Page.	Success
10.	Do Not Have An Account Button	On Click	Redirects To Create New Bank Account Page.	Redirected To Create New Bank Account Page.	Success
11.	Exit Button	On Click	Closes Current (Main page) and Redirects To Login Page.	Closed Current (Main page) and Redirected To Login Page.	Success
WITHDRAW MODULE					
1.	Card Number Text Field	Card No. Not Matching From Database	Error Message : Please Enter The Correct Card No. And Pin.	Error Message : Please Enter The Correct Card No. And Pin.	Success
		Card No. Matching From Database	Redirects To Main Operation Page.	Redirected To Main Operation Page.	Success
		Card No. Matching From Database But Case Not Matching	Error Message : Please Enter The Correct Card No. And Pin.	Error Message : Please Enter The Correct Card No. And Pin.	Success
		User Name Field Blank	Error Message : Please Enter The Correct Card No. And Pin.	Error Message : Please Enter The Correct Card No. And Pin.	Success
2.	Card Pin Text Field	Pin Not Matching From Database	Error Message : Please Enter The Correct Card No. And Pin.	Error Message : Please Enter The Correct Card No. And Pin.	Success
		Pin Matching From Database	Redirects To Main Operation Page.	Redirected To Main Operation Page.	Success
		Pin Matching From Database But Case Not Matching	Error Message : Please Enter The Correct Card No. And Pin.	Error Message : Please Enter The Correct Card No. And Pin.	Success
		Pin Field Blank	Error Message : Please Enter The Correct Card No. And Pin.	Error Message : Please Enter The Correct Card No. And Pin.	Success
3.	Amount Text Field	Int or float datatype entered	No Output	No Output	Success
		Character datatype entered	Blank Text Field Error Message: Enter correct Value.	Blank Text Field Error Message: Enter correct Value.	Success
		Negative or meager value (less than 1 entered)	Error Message: Enter correct Value.	Error Message: Enter correct Value.	Success

CHAPTER-6

ROLES AND RESPONSIBILITY

ROLE

Work as a developer, designer, tester of the application.

RESPONSIBILITIES

- Work on definition of development requirements and priorities.
- Data migration.
- Interfaces with other systems.
- Reporting configuration and deployment • Set up and maintence of security rights and access permission.
- Contributing to technical strategy.policy and procedure.
- Development and operation of technical testing programmes.
- Production of technical documentation to agreed quality standards.
- Reporting on progress/issues to management and users.

CHAPTER-7

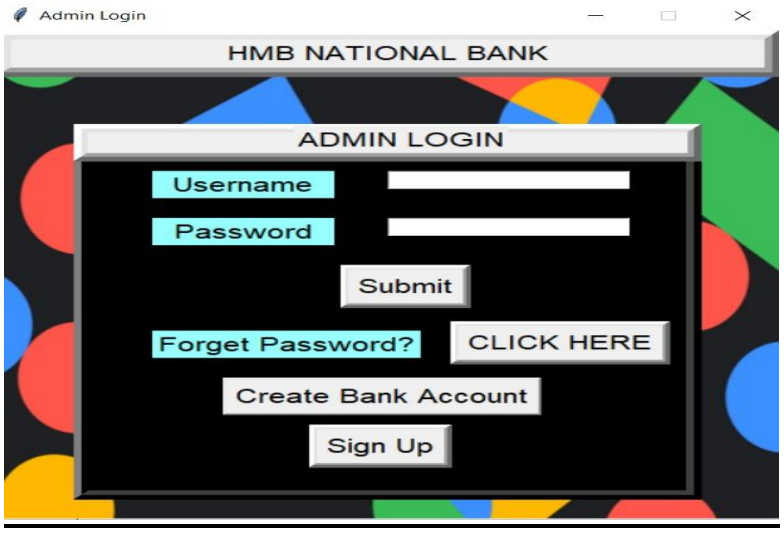
CONCLUSION AND FUTURE ENHANCEMENTS

In Conclusion, I would say the project so far is a success, all the features of the project are working and are in respect to the industrial standards. In first iteration we launched basic ATM features like deposit withdraw and fast cast , in the next version or we can say second iteration we launched a few net banking features like transfer funds, password resetting(forget password) and in the upcoming release we are planning to release personalized banking features in which the banking assistance system helps the user by providing personalized investment and saving plans, the system uses machine learning algorithms and analysis the user's financial habits and provide a personalized assistance with fund management.

I would like to say, that though the project itself is satisfactory but there is still a scope for future enhancements like chat bot service, automated payment services, reminder for payment services, etc.

Though these services have not been included in the project but these can be inculpated in the project at any stage and point of time. This makes our project, scalable which is a good feature.

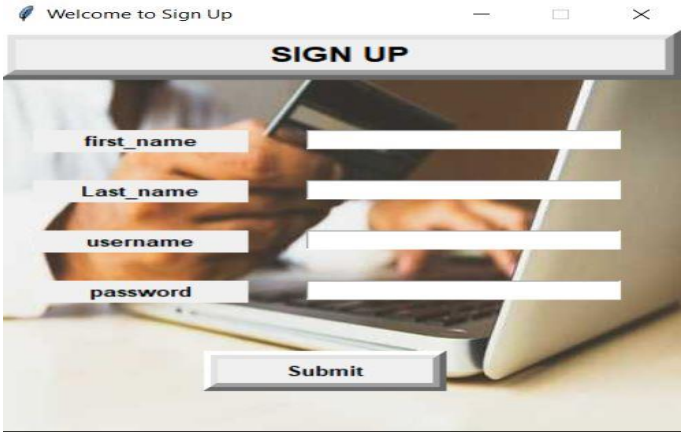
APPENDICES (CODING SNIPPETS)



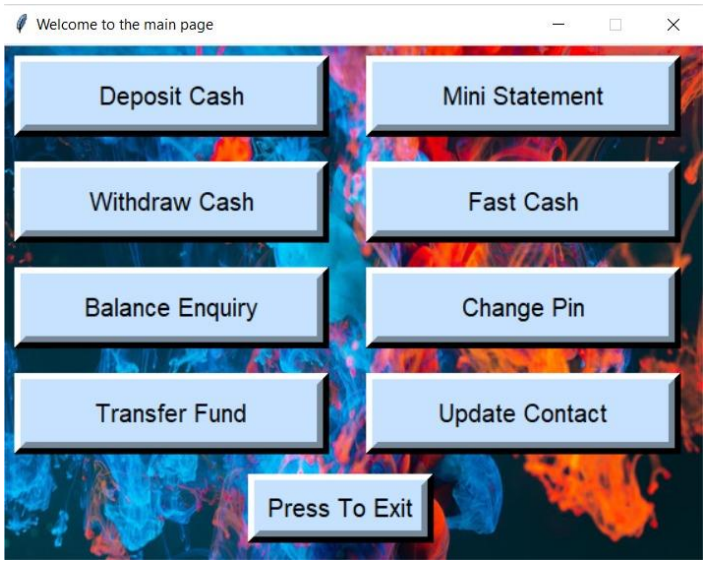

```
from tkinter import*
from tkinter import messagebox
import pymysql
import pymysql.cursors
import os
from PIL import ImageTk, Image
win=Tk()
win.geometry("500x520")
win.resizable(False,False)
win.title("Admin Login")
load=Image.open('Design.png')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)

def new():
   os.system("newACC.py")

def id():
    try:
       a=us.get()
       b=pas.get()
       conn =
pymysql.connect(host='localhost',user='root',pas
sword='123456',db='mad')
       mydb=conn.cursor()
       mydb.execute("select * from login where
username='"+a+"' and password = '"+b+"'")
       result=mydb.fetchall()
       count=mydb.rowcount
       if count>0:
          os.system('atmfirst.py')
       else:

messagebox.showerror("Message","Invalid
userid & password")
    except:

messagebox.showerror("Message","DATABAS
E NOT CONNECTED")

def fr():
    os.system("forgetpassword.py")

def nw():
    os.system("signup.py")

frame=Frame(win,bd=10,relief="raised",width=
500,height=50).grid(row=0)
lb=Label(frame,text="HMB NATIONAL
BANK",font=40).grid(row=0)
```

```
frame2=Frame(win,bd=10,relief="raised",width
=400,height=400,bg="black").place(x=50,y=10
0)

frame3=Frame(win,bd=9,relief="raised",width=
400,height=40).place(x=50,y=100)
lb2=Label(frame3,text="ADMIN
LOGIN",font=20).place(x=190,y=102)

lb=Label(frame2,text="Username",bg="#97FFF
F",font=20,width=10).place(x=100,y=150)
lb2=Label(frame2,text="Password",bg="#97FF
FF",font=20,width=10).place(x=100,y=200)

lb3=Label(frame2,text="Forget
Password?",bg="#97FFFF",font=20,width=15).
place(x=100,y=320)
btn2=Button(frame2,text="CLICK
HERE",command=fr,font=10,bd=5,relief="raise
d").place(x=290,y=310)

lb4=Button(frame2,text="Create Bank
Account",command=new,font=10).place(x=145,
y=370)

btn4=Button(frame2,text="Sign
Up",command=nw,font=10,bd=5,relief="raised"
).place(x=200,y=420)

us=StringVar()
tx=Entry(frame2,width=25,textvariable=us).plac
e(x=250,y=150)
pas=StringVar()
tx2=Entry(frame2,width=25,textvariable=pas).p
lace(x=250,y=200)

btn=Button(frame2,text="Submit",font=20,bd=5
,command=id,relief="raised").place(x=220,y=2
50)
```


```
from tkinter import*
import pymysql
import pymysql.cursors
import os
from tkinter import messagebox
from PIL import ImageTk, Image
win=Tk()
win.geometry("400x400")
win.title("Forget Password Page")
win.resizable(False,False)
load=Image.open('fo.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)

def insert():
    a=str(num1.get())
    b=str(num2.get())
    c=str(num3.get())
    conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
    mydb=conn.cursor()
    mydb.execute("select password from
login where username='"+c+"'")
    conn.commit()
    result=mydb.fetchall()
    count=mydb.rowcount
    print(result)
    print(count)
if count>0:
        for row in result:
                {

messagebox.showinfo("Message",row)
            }
    else:
        messagebox.showerror("Message","Fill
Details")

frame=Frame(win,bd=10,relief="raised",wid
th=400,height=50).grid(row=0)
lb=Label(frame,font=('arial',15,'bold'),text="
FORGET PASSWORD").grid(row=0)

lb=Label(win,font=('arial',10,'bold'),text="Fi
rst Name",width=16).place(x=20,y=100)
lb2=Label(win,font=('arial',10,'bold'),text="
Last Name",width=16).place(x=20,y=150)
lb3=Label(win,font=('arial',10,'bold'),text="
User Name",width=16).place(x=20,y=200)

num1=StringVar()
tx=Entry(win,width=30,textvariable=num1).
place(x=180,y=100)
num2=StringVar()
tx2=Entry(win,width=30,textvariable=num2
).place(x=180,y=150)
num3=StringVar()
tx3=Entry(win,width=30,textvariable=num3
).place(x=180,y=200)
btn=Button(win,text="Submit",font=('arial',
10,'bold'),command=insert,width=15,bd=8,r
elief="raised").place(x=130,y=280)
win.mainloop()
```

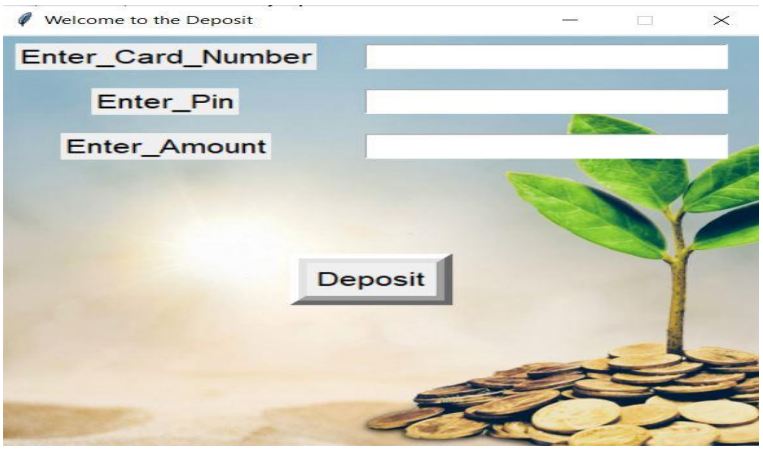



```
from tkinter import*
import pymysql
import pymysql.cursors
import os
from tkinter import messagebox
from PIL import ImageTk, Image
win=Tk()
win.geometry("400x400")
win.resizable(False,False)
win.title("Welcome to Sign Up")
load=Image.open('up.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
def insert():
    a=str(num1.get())
    b=str(num2.get())
    c=str(num3.get())
    d=str(num4.get())
    try:
        conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("insert into
login(first_name,Last_name,username,pass
word)
values('"+a+"','"+b+"','"+c+"','"+d+"')")
        conn.commit()

messagebox.showinfo("Message","Account
Created!!")
    except:
        conn.rollback()
    messagebox.showerror("Messge","Check
Details")
    conn.close()
frame=Frame(win,bd=10,relief="raised",wid
th=400,height=50).grid(row=0)
lb=Label(frame,font=('arial',15,'bold'),text="
SIGN UP").grid(row=0)
lb=Label(win,font=('arial',10,'bold'),text="Fi
rst_name",width=15).place(x=20,y=100)
lb2=Label(win,font=('arial',10,'bold'),text="
Last_name",width=15).place(x=20,y=150)
lb3=Label(win,font=('arial',10,'bold'),text="
username",width=15).place(x=20,y=200)
lb4=Label(win,font=('arial',10,'bold'),text="
password",width=15).place(x=20,y=250)
num1=StringVar()
tx=Entry(win,width=30,textvariable=num1).
place(x=180,y=100)
num2=StringVar()
tx2=Entry(win,width=30,textvariable=num2
).place(x=180,y=150)
num3=StringVar()
tx3=Entry(win,width=30,textvariable=num3
).place(x=180,y=200)
num4=StringVar()
tx4=Entry(win,width=30,textvariable=num4
).place(x=180,y=250)
btn=Button(win,text="Submit",font=('arial',
10,'bold'),command=insert,width=15,bd=8,r
elief="raised").place(x=120,y=320)
win.mainloop()
```



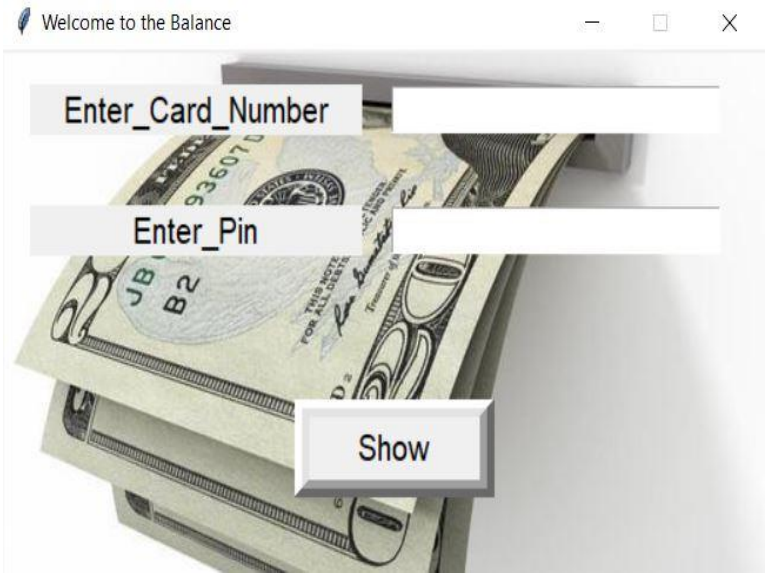

```
from tkinter import*
import os
from PIL import ImageTk, Image

def ab():
    os.system("deposit.py")
def wd():
    os.system("Withdraw.py")
def bl():
    os.system("Balance.py")
def tf():
    os.system("transfer2.py")
def ch():
    os.system("change.py")
def fc():
    os.system("fast.py")
def up():
    os.system("update.py")
def new():
    os.system("newACC.py")
def st():
    os.system('ministatement1.py')

lin=Tk()
lin.geometry("570x420")
lin.resizable(False,False)
lin.title("Welcome to the main page")
load=Image.open('E:\Summer Internship\ATM projects\photo.jpg')
render=ImageTk.PhotoImage(load)
img=Label(lin,image=render)
img.place(x=0,y=0)

tb1=Button(lin,text="Deposit Cash",bg="slategray1",command=ab,width=20,bd=10,relief="raised",font=20).grid(row=0,column=0,padx=10,pady=10,ipadx=5,ipady=5)
tb2=Button(lin,text="Withdraw Cash",bg="slategray1",command=wd,width=20,bd=10,relief="raised",font=20).grid(row=1,column=0,padx=10,pady=10,ipadx=5,ipady=5)
tb3=Button(lin,text="Balance Enquiry",bg="slategray1",command=bl,width=20,bd=10,relief="raised",font=20).grid(row=2,column=0,padx=10,pady=10,ipadx=5,ipady=5)
tb4=Button(lin,text="Transfer Fund",bg="slategray1",command=tf,width=20,bd=10,relief="raised",font=20).grid(row=3,column=0,padx=10,pady=10,ipadx=5,ipady=5)
tb5=Button(lin,text="Mini Statement",bg="slategray1",command=st,width=20,bd=10,relief="raised",font=20).grid(row=0,column=2,padx=20,pady=10,ipadx=5,ipady=5)
```

```
tb6=Button(lin,text="Fast
Cash",bg="slategray1",command=fc,bd=10,
width=20,relief="raised",font=20).grid(row
=1,column=2,padx=20,pady=10,ipadx=5,ipa
dy=5)
tb7=Button(lin,text="Change
Pin",bg="slategray1",command=ch,bd=10,w
idth=20,relief="raised",font=20).grid(row=2
,column=2,padx=15,pady=10,ipadx=5,ipady
=5)
tb8=Button(lin,text="Update
Contact",bg="slategray1",command=up,wid
th=20,bd=10,relief="raised",font=20).grid(r
ow=3,column=2,padx=20,pady=10,ipadx=5,
ipady=5)
tb10=Button(lin,text="Press To
Exit",bg="slategray1",bd=10,relief="raised"
,font=20,command=lin.destroy).place(x=20
0,y=350)
lin.mainloop()
```

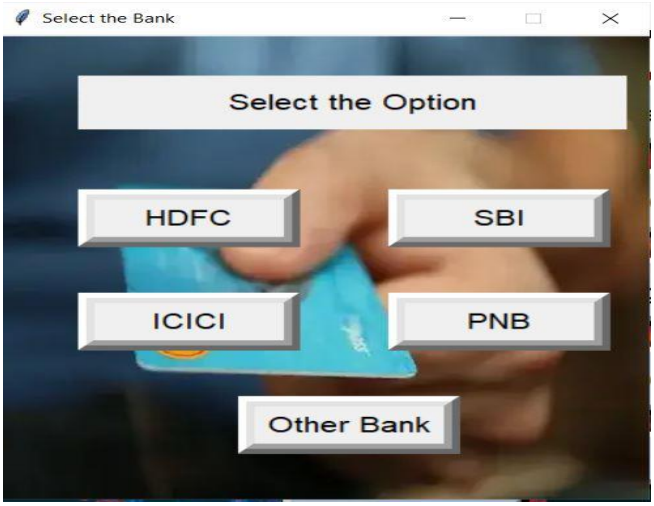



```
from tkinter import*
from tkinter import messagebox
import pymysql
import pymysql.cursors
from PIL import ImageTk, Image
win=Tk()
win.geometry("470x450")
win.title("Welcome to the Deposit")
win.resizable(False,False)
load=Image.open('E:\Summer Internship\ATM projects\de.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
def depo():
    a=str(num1.get())
    b=str(num2.get())
    c=str(num3.get())
    atmtype="Deposit"
    try:
        conn = pymysql.connect(host='localhost',user='root',password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("select * from depos where Enter_Card_Number="'+a+'"")
 mydb.execute("insert into type(Enter_Card_Number,Enter_Amount,type) values ("'+a+'","'+c+'","'+atmtype+'")")
        mydb.execute("update depos set Enter_Amount = Enter_Amount + "'+c+'" where Enter_Pin = "'+b+'"")
        conn.commit()
        result=mydb.fetchall()
        count=mydb.rowcount
        print(result)
        print(count)
        if count>0:
messagebox.showinfo("Message","Deposit")
        else:
messagebox.showerror("Message","Failed")
    except:
        conn.rollback()
        messagebox.showerror("Message","Not deposited")
    conn.close()
lb=Label(win,text="Enter_Card_Number",font=20).grid(row=0,column=0,padx=10,pady=10)
lb2=Label(win,text="Enter_Pin",font=20).grid(row=1,column=0,padx=10,pady=10)
lb3=Label(win,text="Enter_Amount",font=20).grid(row=3,column=0,padx=10,pady=10)
num1=StringVar()
tx=Entry(win,font=20,textvariable=num1).grid(row=0,column=2,padx=20,pady=10)
num2=StringVar()
tx2=Entry(win,font=20,textvariable=num2).grid(row=1,column=2,padx=20,pady=10)
num3=StringVar()
tx3=Entry(win,font=20,textvariable=num3).grid(row=3,column=2,padx=20,pady=10)
btn=Button(win,text="Deposit",command=depo,font=20,bd=10,relief="raised").place(x=180,y=240)
win.mainloop()
```

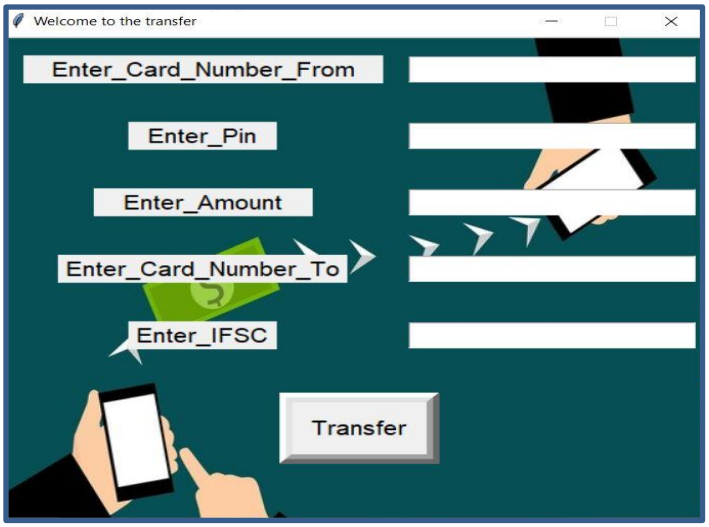

```
from tkinter import*
from tkinter import messagebox
import pymysql
import pymysql.cursors
from PIL import ImageTk, Image
win=Tk()
win.geometry("470x420")
win.title("Welcome to the Withdraw")
win.resizable(False,False)
load=Image.open('wi.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
def wit():
    a=num1.get()
    b=num2.get()
    c=num3.get()
    atmtype="Withdraw"
    try:
        conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("select * from depos
where Enter_Card_Number="'+a+'"")
        mydb.execute("insert into
type(Enter_Card_Number,Enter_Amount,ty
pe) values ("'+a+'","'+c+'","'+atmtype+'")")
 mydb.execute("update depos set
Enter_Amount = Enter_Amount - "'+c+'"
where Enter_Card_Number="'+a+'"")
        conn.commit()
        result=mydb.fetchall()
        count=mydb.rowcount
        print(result)
        print(count)
        if count>0:
messagebox.showinfo("Message","Withdra
w")
        else:
  messagebox.showerror("Message","Check
Card Number")
    except:
        conn.rollback()
        messagebox.showerror("Message","Not
deposited")
    conn.close()
lb=Label(win,text="Enter_Card_Number",w
idth=17,font=20).grid(row=0,column=0,pad
x=10,pady=10)
lb2=Label(win,text="Enter
Pin",width=15,font=20).grid(row=1,column
=0,padx=10,pady=10)
lb3=Label(win,text="Enter
Amount",width=15,font=20).grid(row=3,col
umn=0,padx=10,pady=10)
num1=StringVar()
tx=Entry(win,font=20,textvariable=num1).g
rid(row=0,column=2,padx=20,pady=10)
num2=StringVar()
tx2=Entry(win,font=20,textvariable=num2).
grid(row=1,column=2,padx=20,pady=10)
num3=StringVar()
tx3=Entry(win,font=20,textvariable=num3).
grid(row=3,column=2,padx=20,pady=10)
btn=Button(win,text="Withdraw",command
=wit,font=20,bd=10,relief="raised").place(x
=180,y=280)
win.mainloop()
```




```
from tkinter import*
from tkinter import messagebox
import pymysql
import pymysql.cursors
import os
from PIL import ImageTk, Image
win=Tk()
win.geometry("520x300")
win.title("Welcome to the Balance")
win.resizable(False,False)
load=Image.open('cash.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
def depo():
    a=num1.get()
    b=num2.get()
    conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
    mydb=conn.cursor()
    mydb.execute("select Enter_Amount from
depos where Enter_Card_Number='"+a+"'")
    result=mydb.fetchall()
    count=mydb.rowcount
 print(result)
    print(count)
    if count>0:

messagebox.showinfo("Balance",result)
    else:

messagebox.showerror("Balance","INVALI
D Card Number and PIN")
lb=Label(win,text="Enter_Card_Number",w
idth=20,font=10).grid(row=0,column=0,pad
x=20,pady=20)
lb2=Label(win,text="Enter_Pin",
width=20,font=10).grid(row=3,column=0,pa
dx=20,pady=20)
num1=StringVar()
tx=Entry(win,font=10,textvariable=num1).g
rid(row=0,column=1)
num2=StringVar()
tx2=Entry(win,font=10,textvariable=num2).
grid(row=3,column=1)
btn=Button(win,text="Show",command=de
po,font=10,width=10,bd=10,relief="raised")
.place(x=200,y=200)
win.mainloop()
```

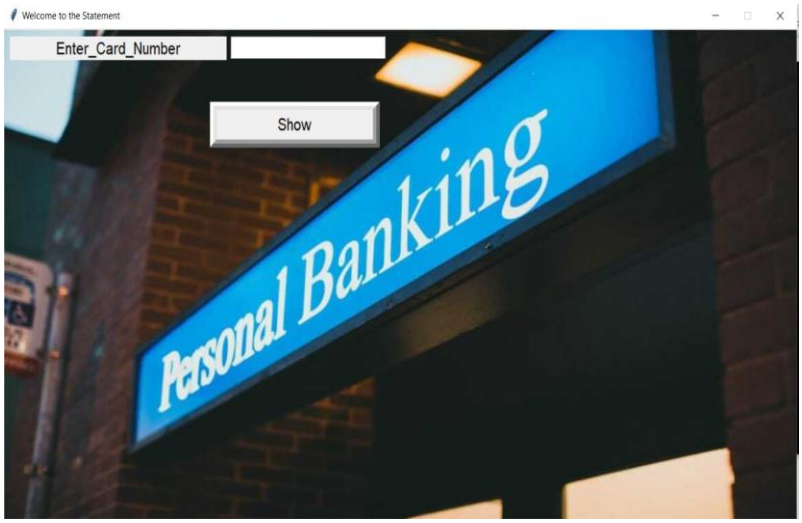



```
from tkinter import*
import os
from PIL import ImageTk, Image
def bank():
    os.system("transfer3.py")
win=Tk()
win.geometry("400x450")
win.title("Select the Bank")
win.resizable(False,False)
load=Image.open('tr.png')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
lb=Label(win,text="Select the
Option",width=30,height=2,font=7).place(x
=50,y=40)
bt1=Button(win,text="HDFC",command=ba
nk,font=20,width=10,bd=10,relief="raised")
.place(x=50,y=150)
bt2=Button(win,text="ICICI",command=ba
nk,font=20,width=10,bd=10,relief="raised")
.place(x=50,y=250)
bt3=Button(win,text="SBI",command=bank
,font=20,width=10,bd=10,relief="raised").pl
ace(x=240,y=150)
bt4=Button(win,text="PNB",command=ban
k,font=20,width=10,bd=10,relief="raised").
place(x=240,y=250)
bt4=Button(win,text="Other
Bank",command=bank,font=20,width=10,bd
=10,relief="raised").place(x=148,y=350)
win.mainloop()
```

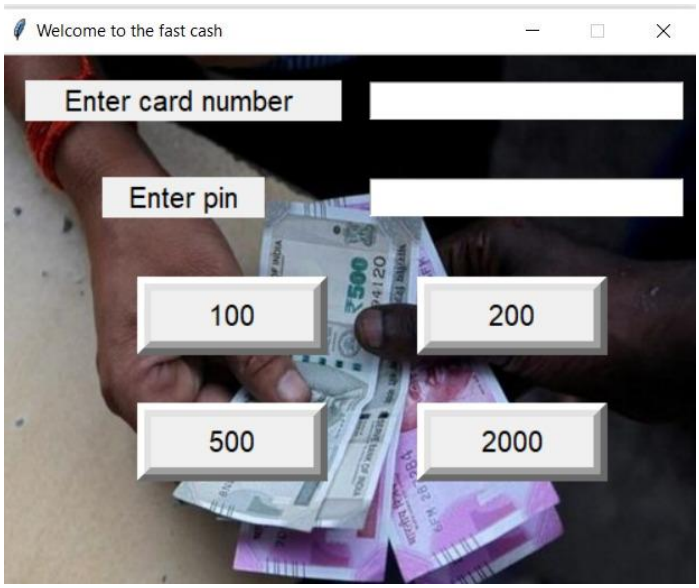



```
from tkinter import*
from tkinter import messagebox
import pymysql
import pymysql.cursors
from PIL import ImageTk, Image
win=Tk()
win.geometry("550x500")
win.resizable(False,False)
win.title("Welcome to the transfer")
load=Image.open('tra.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)

def transfer():
    a=str(num1.get())
    b=str(num2.get())
    c=str(num3.get())
    d=str(num4.get())
    e=str(num5.get())
    atmtype="Transfer"
    try:
        conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("insert into
type(Enter_Card_Number,Enter_Amount,ty
pe) values ('"+a+"','"+c+"','"+atmtype+"')")
        mydb.execute("update depos set
Enter_Amount = Enter_Amount - '"+c+"'
where Enter_Pin='"+b+"'")
        mydb.execute("update bank set
Enter_Amount = Enter_Amount + '"+c+"'
where Enter_Card_Number='"+d+"'")
        conn.commit()

    messagebox.showinfo("Message","Transfed
")
    except:
        conn.rollback()
        messagebox.showinfo("Message","Not
Transfered")
    conn.close()
lb=Label(win,text="Enter_Card_Number_Fr
om",font=20,width=25).grid(row=0,column
=0,padx=20,pady=20)
lb2=Label(win,text="Enter_Pin",font=20,wi
dth=10).grid(row=1,column=0,padx=20,pad
y=20)
lb3=Label(win,text="Enter_Amount",font=2
0,width=15).grid(row=2,column=0,padx=20
,pady=20)
lb4=Label(win,text="Enter_Card_Number_
To",font=20,width=20).grid(row=3,column=
0,padx=20,pady=20)
lb5=Label(win,text="Enter_IFSC",font=20,
width=10).grid(row=4,column=0,padx=20,p
ady=20)
```

```
num1=StringVar()
tx=Entry(win,font=10,width=20,textvariable
=num1).grid(row=0,column=1)
num2=StringVar()
tx2=Entry(win,font=10,width=20,textvariabl
e=num2).grid(row=1,column=1)
num3=StringVar()
tx3=Entry(win,font=10,width=20,textvariabl
e=num3).grid(row=2,column=1)
num4=StringVar()
tx4=Entry(win,font=10,width=20,textvariabl
e=num4).grid(row=3,column=1)
num5=StringVar()
tx5=Entry(win,font=10,width=20,textvariabl
e=num5).grid(row=4,column=1)
btn=Button(win,text="Transfer",command=t
ransfer,relief="raised",bd=10,font=20,highli
ghtbackground="blue",highlightthickness=1
0).place(x=220,y=370)
win.mainloop()
```




```
from tkinter import*
import os
import pymysql
import pymysql.cursors
from tkinter import messagebox
from PIL import ImageTk, Image
win=Tk()
win.geometry("1150x600")
win.resizable(False,False)
win.title("Welcome to the Statement")
load=Image.open('mini.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
def mini():
    a=st.get()
    win2=Frame(win).place(x=80,y=200)
    conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
    mydb=conn.cursor()
    mydb.execute("select * from type where
Enter_Card_Number='"+a+"'")
    result=mydb.fetchall()
 count=mydb.rowcount
    print(result)
    print(count)
    num=5
    tx=Text(win2,font="vendata
20",width=60,height=count+5)

tx.insert(END,"\n\tCard_Number\t\tAmount
\t\tType")
    tx.place(x=num,y=200)
    for i in result:

tx.insert(END,"\t\t\n\t{0}\t\t{1}\t\t{2}".form
at(i[0],i[1],i[2]))
        num+=1
lb=Label(win,text="Enter_Card_Number",f
ont=35,width=28).place(x=10,y=10)
st=StringVar()
tx=Entry(win,width=20,font=30,textvariable
=st).place(x=330,y=10)
btn=Button(win,command=mini,text="Show
",width=20,font=30,bd=10,relief="raised").p
lace(x=300,y=90)
win.mainloop()
```

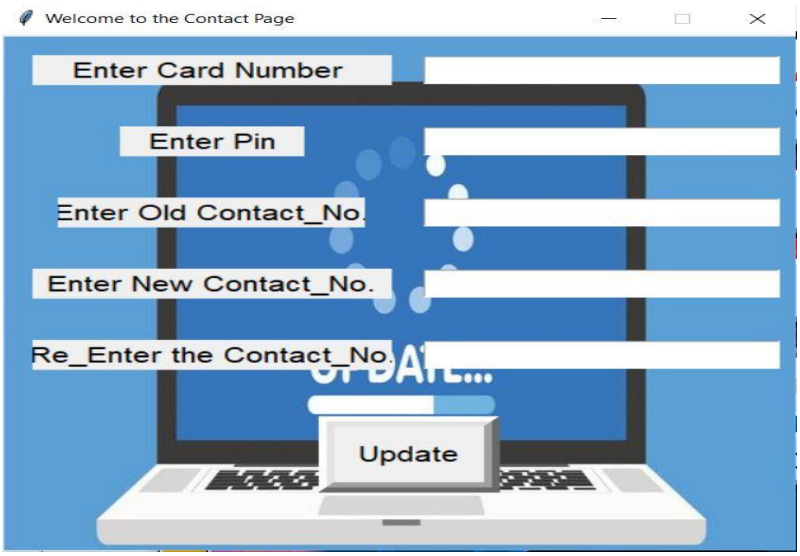



```
from tkinter import*
from tkinter import messagebox
import pymysql
import pymysql.cursors
import os
from PIL import ImageTk, Image

win=Tk()
win.geometry("500x380")
win.title("Welcome to the fast cash")
win.resizable(False, False)
load=Image.open('fa.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)



def cash1():
   a=num1.get()
   b=num2.get()
   amount='2000'
   atmtype="Fast Cash"
   try:
      conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
      mydb=conn.cursor()
      mydb.execute("select * from depos
where Enter_Card_Number="'+a+"'")
      mydb.execute("insert into
type(Enter_Card_Number,Enter_Amount,ty
pe) values
("'+a+"',"'+amount+"',"'+atmtype+"')")
      mydb.execute("update depos set
Enter_Amount=Enter_Amount - 100 where
Enter_Pin="'+b+"'")
      conn.commit()
      result=mydb.fetchall()
      count=mydb.rowcount
      print(result)
      print(count)
      if count>0:

messagebox.showinfo("Message","Rs 2000
Withdrawed")
      else:

messagebox.showerror("Message","Failed")
   except:
      conn.rollback()
      messagebox.showinfo("Message","Not
Withdrawed")
   conn.close()

def cash2():
   a=num1.get()
   b=num2.get()
   amount='5000'
   atmtype="Fast Cash"
```

```
    try:
        conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("select * from depos
where Enter_Card_Number="'+a+"'")
        mydb.execute("insert into
type(Enter_Card_Number,Enter_Amount,ty
pe) values
('"+a+"','"+amount+"','"+atmtype+"')")
        mydb.execute("update depos set
Enter_Amount = Enter_Amount - 200 where
Enter_Pin="'+b+"'")
        conn.commit()
        result=mydb.fetchall()
        count=mydb.rowcount
        print(result)
        print(count)
        if count>0:

messagebox.showinfo("Message","Rs 5000
Withdrawed")
        else:

messagebox.showerror("Message","Failed")
    except:
        conn.rollback()
        messagebox.showinfo("Message","Not
Withdrawed")
    conn.close()

def cash3():
    a=num1.get()
    b=num2.get()
    amount='7000'
    atmtype="Fast Cash"
    try:
        conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("select * from depos
where Enter_Card_Number="'+a+"'")
        mydb.execute("insert into
type(Enter_Card_Number,Enter_Amount,ty
pe) values
('"+a+"','"+amount+"','"+atmtype+"')")
        mydb.execute("update depos set
Enter_Amount = Enter_Amount - 500 where
Enter_Pin="'+b+"'")
        conn.commit()
        result=mydb.fetchall()
        count=mydb.rowcount
        print(result)
        print(count)
        if count>0:

messagebox.showinfo("Message","Rs 7000
Withdrawed")
        else:

messagebox.showerror("Message","Failed")
    except:
        conn.rollback()
        messagebox.showinfo("Message","Not
Withdrawed")
    conn.close()

def cash4():
    a=num1.get()
    b=num2.get()
    amount='10000'
    atmtype="Fast Cash"
    try:
        conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("select * from depos
where Enter_Card_Number="'+a+"'")
        mydb.execute("insert into
type(Enter_Card_Number,Enter_Amount,ty
pe) values
('"+a+"','"+amount+"','"+atmtype+"')")
        mydb.execute("update depos set
Enter_Amount = Enter_Amount - 2000
where Enter_Pin="'+b+"'")
        conn.commit()
        result=mydb.fetchall()
        count=mydb.rowcount
        print(result)
```

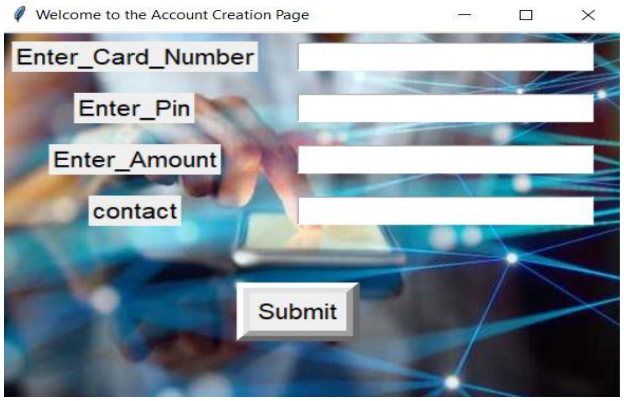
```
        print(count)
        if count>0:

messagebox.showinfo("Message","Rs 10000
Withdrawed")
        else:

messagebox.showerror("Message","Faileds"
)
    except:
        conn.rollback()
        messagebox.showinfo("Message","Not
Withdrawed")
    conn.close()


lb=Label(win,text="Enter card number
",font=20,width=20).grid(row=0,column=0,
padx=20,pady=20)
lb2=Label(win,text="Enter
pin",font=20,width=10).grid(row=1,column
=0,padx=20,pady=20)
num1=StringVar()
tx=Entry(win,font=10,width=20,textvariable
=num1).grid(row=0,column=1)
num2=StringVar()
tx2=Entry(win,font=10,width=20,textvariabl
e=num2).grid(row=1,column=1)

btn=Button(win,text="100",command=cash
1,relief="raised",width=10,bd=10,font=20).
place(x=100,y=160)
btn2=Button(win,text="200",command=cas
h2,relief="raised",width=10,bd=10,font=20)
.place(x=300,y=160)
btn3=Button(win,text="500",command=cas
h3,relief="raised",width=10,bd=10,font=20)
.place(x=100,y=250)
btn4=Button(win,text="2000",command=ca
sh4,relief="raised",width=10,bd=10,font=20
).place(x=300,y=250)

win.mainloop()
```


```
from tkinter import*
import pymysql
import pymysql.cursors
from tkinter import messagebox
from PIL import ImageTk, Image
win=Tk()
win.geometry("510x420")
win.title("Welcome to the Change pin")
win.resizable(False,False)
load=Image.open('E:\Summer
Internship\ATM projects\ch.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
def insert():
a=str(num.get())
b=str(num1.get())
c=str(num2.get())
d=str(num3.get())
try:
if(c==d):
conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
mydb=conn.cursor()
mydb.execute("update depos set
Enter_Pin="'+c+"' where
Enter_Pin="'+b+"'")
conn.commit()
messagebox.showinfo("Message","Pin
Updated")
else:
messagebox.showinfo("Message","Not
Match")
except:
conn.rollback()
print("Not Changed")
conn.close()
lb=Label(win,text="Enter Card Number
",font=20,width=20).grid(row=0,column=0,
padx=20,pady=20)
lb2=Label(win,text="Enter Old
Pin",font=20,width=12).grid(row=1,column
=0,padx=20,pady=20)
lb3=Label(win,text="Enter New
Pin",font=20,width=15).grid(row=2,column
=0,padx=20,pady=20)
lb4=Label(win,text="Re_enter New
Pin",font=20,width=20).grid(row=3,column
=0,padx=20,pady=20)
num=StringVar()
tx=Entry(win,font=10,width=20,textvariable
=num).grid(row=0,column=1)
num1=StringVar()
tx2=Entry(win,font=10,width=20,textvariabl
e=num1).grid(row=1,column=1)
num2=StringVar()
tx3=Entry(win,font=10,width=20,textvariabl
e=num2).grid(row=2,column=1)
num3=StringVar()
tx4=Entry(win,font=10,width=20,textvariabl
e=num3).grid(row=3,column=1)
btn=Button(win,text="Change",command=i
nsert,relief="raised",bd=10,font=20,highligh
tbackground="blue",highlightthickness=10).
place(x=200,y=310)
win.mainloop()
```




```
from tkinter import*
import pymysql
import pymysql.cursors
from tkinter import messagebox
from PIL import ImageTk, Image

win=Tk()
win.geometry("500x500")
win.resizable(False,False)
win.title("Welcome to the Contact Page")
load=Image.open('si.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)

def update():
    a=str(num.get())
    b=str(num1.get())
    c=str(num2.get())
    d=str(num3.get())
    e=str(num4.get())

    try:
        if(e==d):
            conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
            mydb=conn.cursor()
            mydb.execute("update depos set
contact='"+d+"' where contact='"+c+"'")
            conn.commit()

messagebox.showinfo("Message","Contact
Updated")
        else:

messagebox.showinfo("Message","Not
Match")
    except:
        conn.rollback()
        print("Not Changed")
    conn.close()

lb=Label(win,text="Enter Card Number
",font=20,width=20).grid(row=0,column=0,
padx=20,pady=20)
lb2=Label(win,text="Enter
Pin",font=20,width=10).grid(row=1,column
=0,padx=20,pady=20)
lb3=Label(win,text="Enter Old
Contact_No.",font=20,width=17).grid(row=
2,column=0,padx=20,pady=20)
lb4=Label(win,text="Enter New
Contact_No.",font=20,width=20).grid(row=
3,column=0,padx=20,pady=20)
lb5=Label(win,text="Re_Enter the
Contact_No.",font=20,width=20).grid(row=
4,column=0,padx=20,pady=20)
num=StringVar()
```

```
tx=Entry(win,font=10,width=20,textvariable
=num).grid(row=0,column=1)
num1=StringVar()
tx2=Entry(win,font=10,width=20,textvariabl
e=num1).grid(row=1,column=1)
num2=StringVar()
tx3=Entry(win,font=10,width=20,textvariabl
e=num2).grid(row=2,column=1)
num3=StringVar()
tx4=Entry(win,font=10,width=20,textvariabl
e=num3).grid(row=3,column=1)
num4=StringVar()
tx5=Entry(win,font=10,width=20,textvariabl
e=num4).grid(row=4,column=1)
btn=Button(win,command=update,text="Up
date",relief="raised",bd=10,font=20,highlig
htbackground="blue",highlightthickness=10
).place(x=200,y=370)
win.mainloop()
```




```
from tkinter import*
from tkinter import messagebox
import pymysql
import pymysql.cursors
from PIL import ImageTk, Image
win=Tk()
win.geometry("470x350")
win.title("Welcome to the Account Creation
Page")
load=Image.open('ne.jpg')
render=ImageTk.PhotoImage(load)
img=Label(win,image=render)
img.place(x=0,y=0)
def new():
    a=num1.get()
    b=num2.get()
    c=num3.get()
    d=num4.get()
    atmtype="Account Created"
    try:
        conn =
pymysql.connect(host='localhost',user='root'
,password='123456',db='mad')
        mydb=conn.cursor()
        mydb.execute("insert into
type(Enter_Card_Number,Enter_Amount,ty
pe) values ("'+a+"','"+c+"','"+atmtype+"')")
        mydb.execute("insert into
depos(Enter_Card_Number,Enter_Pin,Enter

_Amount,contact)
values("'+a+"','"+b+"','"+c+"','"+d+"')")
        conn.commit()
        messagebox.showinfo("Message","Account
Created!!")
    except:
        conn.rollback()
    messagebox.showinfo("Messge","Account
Not Created")
    conn.close()
lb=Label(win,text="Enter_Card_Number",f
ont=20).grid(row=0,column=0,padx=10,pad
y=10)
lb2=Label(win,text="Enter_Pin",font=20).gr
id(row=1,column=0,padx=10,pady=10)
lb3=Label(win,text="Enter_Amount",font=2
0).grid(row=3,column=0,padx=10,pady=10)
lb4=Label(win,text="contact",font=20).grid(
row=4,column=0,padx=10,pady=10)
num1=StringVar()
tx=Entry(win,font=20,textvariable=num1).g
rid(row=0,column=2,padx=20,pady=10)
num2=StringVar()
tx2=Entry(win,font=20,textvariable=num2).
grid(row=1,column=2,padx=20,pady=10)
num3=StringVar()
tx3=Entry(win,font=20,textvariable=num3).
grid(row=3,column=2,padx=20,pady=10)
num4=StringVar()
tx4=Entry(win,font=20,textvariable=num4).
grid(row=4,column=2,padx=20,pady=10)
btn=Button(win,text="Submit",command=n
ew,font=20,bd=10,relief="raised").place(x=
180,y=240)
win.mainloop()
```

REFERENCES

- https://www.javatpoint.com/software-engineering-iterative-model
- https://www.smartdraw.com/
- https://www.geeksforgeeks.org/software-engineering-software-design-process/
- https://en.wikipedia.org/wiki/Software_design
- https://www.worldwidejournals.com/paripex/recent_issues_pdf/2013/July/issues-and-challenges-faced-by-atm-customers-of-state-bank-of-india-in-south-tamilnadu_July_2013_1901600040_8405487.pdf