# WORKING OF ATM MACHINE

Team Members:

1. Rujuta Shah (121044)
2. Shreya Gokani (121053)

Problem Statement:

Write a program on working of ATM machine using linked list which saves the data (like user’s name, Account number, PIN etc) of the user in a file from the bank perspective and from the user’s perspective show the working of ATM machine including these four points:

1. Cash Withdrawal:

User can withdraw the money using this option

1. Balance Inquiry:

User can check the balance.

1. Fast Cash:

In fast cash, user can withdraw the money instantly by clicking the options of different amount.

1. Change PIN:

User can change PIN number.

5. display the edited information

Brief description:

We always have one question in our mind that how this machines are working? What type of languages and machinery are there in it? Here we have taken working of an ATM machine as a part of our project. Here we have taken two banks named HDFC Bank and Kotak Mahindra Bank. And we have five users in each of the file (as an example, we can take as many as we want)We have made two files and stored the information of the user using singly linked list of these two banks and in these two files. This information is added according to user’s bank in which he/she is carrying account. Basically it will work as a data base for our program. This all work is done from the bank’s perspective. Now from the user’s perspective, we have asked user to enter his/her account number then we will compare user’s account number from the file of a particular bank’s all the account numbers.

If the account number is not correct then it will close the display. And if the user have forgotten the account number or had mistake in writing the account number then it will ask for a pin number and the account will be accessed by him/her if and only if both the account number and pin number is correct. Here for comparison we have loaded all the data of the file in the dummy linked list and then it will traverse in the list. If the account number and pin number gets matched then the user will get these four options:

1. Cash withdrawal
2. Balance inquiry
3. Fast cash
4. Change PIN number

5. Display the edited information

When user will click on the cash withdrawal there will be an option of withdrawing the cash ,if user want to withdraw money which is greater than 20000 or which is less than 0 or which is greater than the current account then this program will show the error. User can with draw the money as much as he/she wants and as many times he/she wants.

When user will click on Balance inquiry then user can check balance in his/her account. This information will be of edited account. If user had made any changes in his/her account then the current edited information will be showed.

When the user is in hurry and user wants to withdraw money then user will get so many options of particular amount so that his/her time can be saved.

When the user wants to change the PIN number then user has to select this option. First of all it will ask his/her old pin number, after verifying it only user can have the chance of changing the pin number and can have his/her new pin number.

After any of the above four actions, we have to change the some of the information in the file or we have to display it from the file to the output terminal. For that we have used ‘iostream’. We have stored all the information from file to the linked list again. And we have done the traversal in the linked list to reach to the user who wants to get the information about his/her account or change some things or withdraw the money

List of data structures used in the project with logical design:

1. Singly Linked list

There are four linked lists in our program:

* For Kotak Mahindra bank’s users’ information is stored in Kotak Mahindra’s file through it.
* For HDFC bank’s users’ information is stored in HDFC’s file through it.
* We have loaded the data stored in Kotak Mahindra file in the linked list.
* We have loaded the data stored in HDFC file in the linked list.

1. Array:

We have made the character array of user name, account number and pin number.

Operations to be performed on each data structure:

1. Insertion :

We have inserted the information like account number, PIN number, user name and bank balance of the users’ of both of the bank in their bank’s file using singly linked list.

1. Updation :

After user withdraw the money or change the PIN number then we are updating that user’s detail in his/her bank’s file.

1. List or print all the values:

List of functions:

1. void createFile\_kotak () :

In this function we have stored Kotak Mahindra bank’s users’ account number, PIN number, user name and account balance using singly linked list and array in the Kotak Mahindra bank’s file.

1. void display\_kotak () :

Here we have displayed all the information of all the users stored in the Kotak Mahindra’s file.

1. void createFile\_HDFC () :

In this function we have stored HDFC bank’s users’ account number, PIN number, user name and account balance using singly linked list and array in the HDFC bank’s file.

1. void display\_HDFC () :

Here we have displayed all the information of all the users stored in the HDFC’s file.

1. void create\_database () :

This function is from bank’s perspective. In this we have made menu (switch-case) to select the bank in which the bank will add their accounts and will display it. So we have called four functions given above in it and these all functions are from the bank’s perspective.

1. void hdfc\_bank() :

This function is for the users. Here we have loaded the file into the linked list. Then we are asking the user to enter the account number. If the account number is correct then user will be asked to enter the PIN of that account number. If the PIN is correct then user can withdraw the money till 20000 Rs, balance inquiry, change PIN number, fast cash as well as get all the information of his/her account.

1. Void kotak\_bank() :

This function carries same things as ‘hdfc\_bank()’ function.

1. void display\_ATM () :

In this function we are asking user whether je/she is carrying account in HDFC bank or Kotak Mahindra bank. And from that we have made menu and we have called ‘void hdfc\_bank ()’ and ‘Void kotak\_bank ()’.

List of input data:

1. Account number

2. User name

3. Pin number

4. Current balance

Search for a specific elements for given input values

From the input values and according to the information we can search and edit it from traversing it to the file.

Pseudo code:

Algorithm header:

ATM\_Machine\_working

Pre condition:

Pre

The data should be entered to create linked list and should be saved in a file.

Post condition:

Post

Print all the values from the node of the linked list. Change it if needed.

Variables:

struct kotakMahindra \*next, \*head = NULL, \*temp;

struct HDFC \*next1;

struct HDFC\*temp4=NULL;

struct HDFC\*next3, \*head1 = NULL, \*temp1, \*head2, \*temp2;

struct HDFC\*head3=NULL;

struct HDFC\*curr3=NULL;

1. Create function “createFile\_kotak ()”
2. Make head to be NULL.
3. if head is NULL then go to step 4 else go to step 12.
4. Make a pointer temp <- head which points to all the data of this Linked list.
5. Make a node temp giving a size of a structure using malloc.
6. Print “enter the account number of Kotak Mahindra” and scan the number in variable temp->AccountNo.
7. Print “enter the user name of Kotak Mahindra bank” and scan the name in variable temp->userName.
8. Print “enter the PIN number of Kotak Mahindra” and scan the PIN number in variable temp->PIN.
9. Print “enter the current balance of Kotak Mahindra” and scan the balance in variable temp->currBal.
10. Make next node of the temp NULL. That is temp->next=NULL.
11. Assign values of temp to head.
12. Make a node temp->next giving a size of a structure using malloc.
13. Make temp to be equal to temp->next.
14. Print “enter the account number of Kotak Mahindra” and scan the number in variable temp->AccountNo.
15. Print “enter the user name of Kotak Mahindra bank” and scan the name in variable temp->userName.
16. Print “enter the PIN number of Kotak Mahindra” and scan the PIN number in variable temp->PIN.
17. Print “enter the current balance of Kotak Mahindra” and scan the balance in variable temp->currBal.
18. Make next node of the temp NULL. That is temp->next=NULL.
19. Make a file having a pointer fp that is “FILE \*fp.
20. Give the name of the file as KOTAK.dat and open it in an append mode.
21. If file exists then go to step 22.
22. Print account number, user name, PIN number and current balance in the file.
23. Close the file.
24. Print if user wants to continue and scan yes or no that is ‘y’ or ‘n’ in the variable ch.
25. If ch is ‘y’ or ‘Y’ then go to step 3 else go to step 26.
26. Make a function display\_kotak ()
27. Open the file KOTAK.dat in read mode.
28. If file doesn’t exist then go to step 29 else go to step 30
29. Print: "Error while opening the file." And go to step
30. Take a variable ch to get the character from the file using getc.
31. If ch is not reached to the end of the file then go to step 32 else go to step 34.
32. Display ch.
33. Ch is at next character now go to step 30.
34. Close the file.
35. Create a function “createFile\_HDFC ()”
36. Make head1 to be NULL.
37. if head1 is NULL then go to step 38 else go to step 12.
38. Make a pointer temp1 <- head1 which points to all the data of this Linked list.
39. Make a node temp1 giving a size of a structure using malloc.
40. Print “enter the account number of HDFC bank” and scan the number in variable temp1->AccountNo1.
41. Print “enter the user name of HDFC bank” and scan the name in variable temp1->userName1.
42. Print “enter the PIN number of HDFC bank” and scan the PIN number in variable temp1->PIN1.
43. Print “enter the current balance of HDFC bank” and scan the balance in variable temp1->currBal1.
44. Make next node of the temp1 NULL. That is temp1->next1=NULL.
45. Assign values of temp1 to head1.
46. Make a node temp1->next1 giving a size of a structure using malloc.
47. Make temp1 to be equal to temp1->next1.
48. Print “enter the account number of HDFC bank” and scan the number in variable temp1->AccountNo1.
49. Print “enter the user name of HDFC bank” and scan the name in variable temp1->userName1.
50. Print “enter the PIN number of HDFC bank” and scan the PIN number in variable temp1->PIN1.
51. Print “enter the current balance of HDFC bank” and scan the balance in variable temp1->currBal1.
52. Make next node of the temp1 NULL. That is temp1->next1=NULL.
53. Make a file having a pointer fp1 that is “FILE \*fp1.
54. Give the name of the file as HDFC.dat and open it in an append mode.
55. If file exists then go to step 56.
56. Print account number, user name, PIN number and current balance in the file.
57. Close the file.
58. Print if user wants to continue and scan yes or no that is ‘y’ or ‘n’ in the variable ch1.
59. If ch1 is ‘y’ or ‘Y’ then go to step 37 else go to step 60.
60. Make a function display\_HDFC ()
61. Open the file HDFC.dat in read mode.
62. If file doesn’t exist then go to step 63 else go to step 64.
63. Print: "Error while opening the file." And go to step
64. Take a variable ch to get the character from the file using getc.
65. If ch is not reached to the end of the file then go to step 66 else go to step 68.
66. Display ch.
67. Ch is at next character now go to step 64.
68. Close the file.
69. Make a function “create\_database ().
70. Print "Select your bank: write 123 for HDFC bank and 456 for Kotak Mahindra.
71. Scan it from the variable a.
72. If a=123 then go to step 73 else go to step 83.
73. Select your choice from below 1, 2 & 3.

Case 1: create file HDFC

Case 2: Display the information of it.

Case 3: exit

1. Scan the choices in variable s.
2. If case 1 then go to step 75 else go to step 77
3. Call function “createFile\_HDFC ()”
4. If case 2 then go to step 78 else go to step 79
5. Call function “display\_HDFC ()
6. If case 3 then go to step 80.
7. Print “Do you want to enter more information of users into HDFC bank? Enter 1 to continue 0 to exit (1/0) “
8. Scan the value 1 or 0 by variable n.
9. If n<-1 then go to step 73 else go to step 83.
10. If a<-456 then go to step 84 else go to step
11. Select your choice from below 1, 2 & 3.

Case 1: create file Kotak Mahindra.

Case 2: Display the information of it.

Case 3: exit

1. Scan the choices in variable s1.
2. If case 1 then go to step 87 else go to step 88
3. Call function “createFile\_kotak()”
4. If case 2 then go to step 89 else go to step 90
5. Call function “display\_kotak ()
6. If case 3 then go to step 91.
7. Print “Do you want to enter more information of users into Kotak Mahindra bank? Enter 1 to continue 0 to exit (1/0) “
8. Scan the value 1 or 0 by variable n1.
9. If n1<-1 then go to step 84 else go to step 94.
10. Print “do you want to add more accounts?( y/n)”.
11. Scan the answer by variable b.
12. If b<-‘y’ or ‘Y’ then go to step 70 else go to step 97.
13. Make a function “hdfc\_bank().
14. Ask user for the account number.
15. Scan the account number in variable pq.
16. Open file HDFC.dat in read mode that is:

ifstream k;

k.open("HDFC.dat")

1. If file is open then go to step 102 else go to step 103.
2. Loop condition:

for (int i=0;i<=5;i++)

{

1. if pointer curr3 is NULL then go to step 2 else go to step 13
2. make node head3 and assign the size of a structure HDFC using malloc.
3. Make curr3<-head3 which points to the frst user’s data.
4. Read the first data of the file and make it d1.
5. Copy this data d1that is account number into curr3->fd1.
6. Read the second data that is user name of the file and make it d2.
7. Copy this data d2 into curr3->fd2.
8. Read the third data that is PIN number of the file and make it d3.
9. Copy this data d3 into curr3->fd3.
10. Read the fourth data that is current balance of the file and make it d4.
11. Copy this data d4 into curr3->fd4.
12. Make next of curr3 NULL that is curr3->next3=NULL
13. make node curr3->next3 and assign the size of a structure HDFC using malloc.
14. Make curr3=curr3->next3 which points to the frst user’s data.
15. Read the first data of the file and make it d1.
16. Copy this data d1that is account number into curr3->fd1.
17. Read the second data that is user name of the file and make it d2.
18. Copy this data d2 into curr3->fd2.
19. Read the third data that is PIN number of the file and make it d3.
20. Copy this data d3 into curr3->fd3.
21. Read the fourth data that is current balance of the file and make it d4.
22. Copy this data d4 into curr3->fd4.
23. Make next of curr3 NULL that is curr3->next3=NULL
24. Print “fail to open a file” and go to step last.
25. Loop condition:

/\*for (curr3=head3;curr3!=NULL;curr3=curr3->next3)

{

if (strcmp(curr3->fd1,pq)==1)

{

cout<<"Enter your PIN:\n";

cin>>z;

if (strcmp(curr3->fd4,z)==1)

{

cout<<"Correct PIN & Account no.\n";

do

{

cout<< " \n \n Select your action from below";

cout<< " \n 1. cash withdrwal ";

cout<<" \n 2. Fast cash";

cout<<" \n 3.Account inquiry";

cout<<" \n 4. Change PIN";

cout<<"\n 5. Display the informatin after editing";

cin>>abc;

switch(abc)

}

}

}

}

\*/

for (curr3=head3;curr3!=NULL;curr3=curr3->next3)

1. Display curr3->fd1 that is account number
2. Display curr3->fd2 that is user name
3. Display curr3->fd3 that is PIN number
4. Display curr3->fd4 that is current balance.

1. Loop condition

For (curr3=head3;curr3!=NULL;curr3=curr3->next3)

1. If curr3->next3->fd1 & pq both are equal then go to step 2 else
2. Display that you have successfully enterd your account number.
3. Display the name of the user saying hello.

TERMINAL:

1. Ask user to enter the PIN number and scan it in z.
2. if curr3->next3->fd3(a PIN number) is same as z then go to step 6 else go to 60

EDIT:

1. Display “successfully entered the PIN number"
2. Select your action from below:
3. cash withdrawal
4. Fast cash
5. Account inquiry
6. Change PIN
7. Display the information after editing
8. Scan these choices in abc.
9. If choice is 1 then go to step 11 else go to step 26.

RUJUTA:

1. Print “ Enter the amount to withdraw"
2. Scan the amount in lh.
3. If lh is less than or equal to Rs.20000 and lh is less than or equal to curr3->fd4 then go to step 15 else go to step 17.
4. New (curr3->fd4)=(curr3->fd4)-(lh)
5. Display the new balance curr3->fd4 on the terminal and go to step 22.
6. if lh is greater than 20000 and less than curr3->fd4 then go to step 18 else go to step 19.
7. Print “Amount should not be more than 20,000" and go to step 22
8. If lh is less than 0 then go to step 20 else go to step 21.
9. Print “Error : Negative amount will not be accepted "
10. Print “balance is insufficient” and go to step 22.
11. Print “press y to re-enter the amount"
12. Scan it in variable rt.
13. If rt<-‘y’ or ‘Y’ then go to step 11 else 25.
14. Go to TERMINAL2
15. If choice is 2 then go to step 27 else go to -30.

SLIDE:

1. Print “withdraw fast cash amount”.
2. Print “Select the amount from below"
3. 500
4. 1000
5. 2000
6. 5000
7. Scan the choice in variable hy.
8. If choice is 1 then go to step 7 else go to step 11.
9. if(500 <=curr3->fd4) then go to step 8 else go to step 10
10. new(curr3->fd4)=(curr3->fd4)-500
11. Display the new balance to the user for the successful withdraw and go to step 29.
12. Print insufficient balance and go to step 29.
13. If choice is 2 then go to step 12 else go to step 16.
14. if(1000 <=curr3->fd4) then go to step 13 else go to step 15
15. new(curr3->fd4)=(curr3->fd4)-1000
16. Display the new balance to the user for the successful withdraw and go to step 29.
17. Print insufficient balance and go to step 29.
18. If choice is 3 then go to step 17 else go to step 21
19. if(2000 <=curr3->fd4) then go to step 18 else go to step 20
20. new(curr3->fd4)=(curr3->fd4)-2000
21. Display the new balance to the user for the successful withdraw and go to step 29.
22. Print insufficient balance and go to step 29.
23. If choice is 4 then go to step 22 else go to step 26.
24. if(5000 <=curr3->fd4) then go to step 23 else go to step 25
25. new(curr3->fd4)=(curr3->fd4)-5000
26. Display the new balance to the user for the successful withdraw and go to step 29.
27. Print insufficient balance and go to step 29.
28. Print “insufficient choice” and go to step 27.
29. print “please re-enter a proper choice”
30. Go to SLIDE (step 27).
31. Print ‘Do you want to do fast cash again? (1/0)’ and scan it in variable fr.
32. If fr = 1 then go to SLIDE (step 27) else go to TERMINAL2.
33. If choice is 3 then go to step 31 else go to step
34. Print “ Account inquiry "
35. Print “Hello Mr/Mrs :"display the name by scanning curr3->fd2
36. Print “Your pin number is :” display the PIN by scanning curr3->fd3
37. Print “Current balance in your account is : "display the balance by scanning curr3->fd4
38. Print “Your account number is: "display the balance by scanning curr3->fd1.
39. If choice is 4 then go to step 37 else go to step 49.
40. Print “Are you sure you want to change your PIN number y/n?"
41. Scan the answer in variable kg.
42. If kg=’y’ or ‘Y’ then go to step 40 else go to step 48.

PIN:

1. Print “please re-enter enter your old pin number :"
2. Scan the old PIN in variable hr.
3. If hr and old PIN number that is curr3->fd3 are same then go to step 44 else go to step 47.
4. Print “please enter your new PIN number: "
5. Scan the result in variable fv.
6. Change curr3->fd3 by fv and display it and go to.
7. Print “incorrect PIN” and go to PIN.
8. Go to TERMINAL2.
9. If choice is 5 then go to step 49 else go to step 55.
10. Print “Your information after editing:”
11. Display account number that is curr3->fd1
12. Display user name that is curr3->fd2
13. Display PIN that is curr3->fd3
14. Display balance in account that is curr3->fd4.
15. Print “improper choice” go to step 56.

TERMINAL2:

1. Print “Do you want to edit again in your account y/n?"
2. Scan the answer in variable js.
3. If js=’y’ or ‘Y’ then go to EDIT else go to FILE.
4. Print “incorrect PIN” and go to TERMINAL.7
5. Close file.

FILE:

1. Write new information in file after opening it that is “ofstream fwrt;

fwrt.open("HDFC.dat");”

if (fwrt.is\_open())

1. Loop condition:

For (temp4=head3;temp4!=NULL;temp4=temp4>next3)

1. fwrt<< temp4->fd1<<"\n";
2. fwrt<< temp4->fd2<<"\n";
3. fwrt<< temp4->fd3<<"\n";
4. fwrt<< temp4->fd4<<"\n";
5. Close the file.
6. Make a function “kotak\_bank().
7. Ask user for the account number.
8. Scan the account number in variable pq1.
9. Open file KOTAK.dat in read mode that is:

ifstream k1;

k1.open("KOTAK.dat")

1. If file is open then go to step 111 else go to step 112.
2. Loop condition:

for (int i=0;i<=5;i++)

{

1. if pointer curr5 is NULL then go to step 2 else go to step 13
2. make node head5 and assign the size of a structure HDFC using malloc.
3. Make curr5<-head5 which points to the first user’s data.
4. Read the first data of the file and make it dd1.
5. Copy this data d1that is account number into curr5->fdd1.
6. Read the second data that is user name of the file and make it dd2.
7. Copy this data d2 into curr5->fdd2.
8. Read the third data that is PIN number of the file and make it dd3.
9. Copy this data dd3 into curr5->fdd3.
10. Read the fourth data that is current balance of the file and make it dd4.
11. Copy this data d4 into curr55->fdd4.
12. Make next of curr5 NULL that is curr5->next5=NULL
13. make node curr5->next5 and assign the size of a structure kotakMahindra using malloc.
14. Make curr5=curr5->next5 which points to the first user’s data.
15. Read the first data of the file and make it dd1.
16. Copy this data d1that is account number into curr5->fdd1.
17. Read the second data that is user name of the file and make it dd2.
18. Copy this data dd2 into curr5->fdd2.
19. Read the third data that is PIN number of the file and make it dd3.
20. Copy this data d3 into curr5->fdd3.
21. Read the fourth data that is current balance of the file and make it dd4.
22. Copy this data dd4 into curr5->fdd4.
23. Make next of curr5 NULL that is curr5->next5=NULL
24. Print “fail to open a file” and go to step last.
25. Loop condition:

/\*for (curr3=head3;curr3!=NULL;curr3=curr3->next3)

{

if (strcmp(curr3->fd1,pq)==1)

{

cout<<"Enter your PIN:\n";

cin>>z;

if (strcmp(curr3->fd4,z)==1)

{

cout<<"Correct PIN & Account no.\n";

do

{

cout<< " \n \n Select your action from below";

cout<< " \n 1. cash withdrwal ";

cout<<" \n 2. Fast cash";

cout<<" \n 3.Account inquiry";

cout<<" \n 4. Change PIN";

cout<<"\n 5. Display the informatin after editing";

cin>>abc;

switch(abc)

}

}

}

}

\*/

for (curr3=head3;curr3!=NULL;curr3=curr3->next3)

1. Display curr5->fdd1 that is account number
2. Display curr5->fdd2 that is user name
3. Display curr5->fdd3 that is PIN number
4. Display curr5->fdd4 that is current balance.

1. Loop condition

For (curr5=head5;curr5!=NULL;curr5=curr5->next5)

1. If curr5->next5->fdd1 & pq1 both are equal then go to step 2 else
2. Display that you have successfully entered your account number.
3. Display the name of the user saying hello.

TERMINAL1:

1. Ask user to enter the PIN number and scan it in z1.
2. if curr5->next5->fdd3(a PIN number) is same as z1 then go to step 6 else go to 60

EDIT1:

1. Display “successfully entered the PIN number"
2. Select your action from below:
3. cash withdrawal
4. Fast cash
5. Account inquiry
6. Change PIN
7. Display the information after editing
8. Scan these choices in abc1.
9. If choice is 1 then go to step 11 else go to step 26.

RUJUTA1:

1. Print “ Enter the amount to withdraw"
2. Scan the amount in lh1.
3. If lh1 is less than or equal to Rs.20000 and lh is less than or equal to curr5->fdd4 then go to step 15 else go to step 17.
4. New (curr5->fdd4)=(curr5->fdd4)-(lh1)
5. Display the new balance curr5->fdd4 on the terminal and go to step 22.
6. if lh1 is greater than 20000 and less than curr5->fdd4 then go to step 18 else go to step 19.
7. Print “Amount should not be more than 20,000" and go to step 22
8. If lh is less than 0 then go to step 20 else go to step 21.
9. Print “Error : Negative amount will not be accepted "
10. Print “balance is insufficient” and go to step 22.
11. Print “press y to re-enter the amount"
12. Scan it in variable rt1.
13. If rt<-‘y’ or ‘Y’ then go to step 11 else 25.
14. Go to TERMINAL3
15. If choice is 2 then go to step 27 else go to -30.

SLIDE1:

1. Print “withdraw fast cash amount”.
2. Print “Select the amount from below"
3. 500
4. 1000
5. 2000
6. 5000
7. Scan the choice in variable hy1.
8. If choice is 1 then go to step 7 else go to step 11.
9. if(500 <=curr5->fdd4) then go to step 8 else go to step 10
10. new(curr5->fdd4)=(curr5->fdd4)-500
11. Display the new balance to the user for the successful withdraw and go to step 29.
12. Print insufficient balance and go to step 29.
13. If choice is 2 then go to step 12 else go to step 16.
14. if(1000 <=curr5->fdd4) then go to step 13 else go to step 15
15. new(curr5->fdd4)=(curr5->fdd4)-1000
16. Display the new balance to the user for the successful withdraw and go to step 29.
17. Print insufficient balance and go to step 29.
18. If choice is 3 then go to step 17 else go to step 21
19. if(2000 <=curr5->fdd4) then go to step 18 else go to step 20
20. new(curr5->fdd4)=(curr5->fdd4)-2000
21. Display the new balance to the user for the successful withdraw and go to step 29.
22. Print insufficient balance and go to step 29.
23. If choice is 4 then go to step 22 else go to step 26.
24. if(5000 <=curr3->fd4) then go to step 23 else go to step 25
25. new(curr3->fd4)=(curr3->fd4)-5000
26. Display the new balance to the user for the successful withdraw and go to step 29.
27. Print insufficient balance and go to step 29.
28. Print “insufficient choice” and go to step 27.
29. print “please re-enter a proper choice”
30. Go to SLIDE1 (step 27).
31. Print ‘Do you want to do fast cash again? (1/0)’ and scan it in variable fr1.
32. If fr1 = 1 then go to SLIDE1 (step 27) else go to TERMINAL3.
33. If choice is 3 then go to step 31 else go to step
34. Print “ Account inquiry "
35. Print “Hello Mr/Mrs :"display the name by scanning curr5->fdd2
36. Print “Your pin number is :” display the PIN by scanning curr5->fdd3
37. Print “Current balance in your account is : "display the balance by scanning curr5->fdd4
38. Print “Your account number is: "display the balance by scanning curr5->fdd1.
39. If choice is 4 then go to step 37 else go to step 49.
40. Print “Are you sure you want to change your PIN number y/n?"
41. Scan the answer in variable kg1.
42. If kg=’y’ or ‘Y’ then go to step 40 else go to step 48.

PIN1:

1. Print “please re-enter enter your old pin number :"
2. Scan the old PIN in variable h1r.
3. If h1r and old PIN number that is curr3->fd3 are same then go to step 44 else go to step 47.
4. Print “please enter your new PIN number: "
5. Scan the result in variable fv1.
6. Change curr5->fdd3 by fv and display it and go to.
7. Print “incorrect PIN” and go to PIN1.
8. Go to TERMINAL3.
9. If choice is 5 then go to step 49 else go to step 55.
10. Print “Your information after editing:”
11. Display account number that is curr5->fdd1
12. Display user name that is curr5->fdd2
13. Display PIN that is curr5->fdd3
14. Display balance in account that is curr5->fdd4.
15. Print “improper choice” go to step 56.

TERMINAL3:

1. Print “Do you want to edit again in your account y/n?"
2. Scan the answer in variable js1.
3. If js1=’y’ or ‘Y’ then go to EDIT1 else go to FILE1.
4. Print “incorrect PIN” and go to TERMINAL1.
5. Close file.

FILE:

1. Write new information in file after opening it that is “ofstream fwrt1

fwrt.open("KOTAK.dat");”

if (fwrt1.is\_open())

1. Loop condition:

For (temp5=head5;temp5!=NULL;temp5=temp5>next5)

1. Fwrt1<< temp5->fdd1<<"\n";
2. Fwrt1<< temp5->fdd2<<"\n";
3. Fwrt1<< temp5->fdd3<<"\n";
4. Fwrt1<< temp5->fdd4<<"\n";
5. Close the file.
6. create function “display\_ATM ()”
7. print “welcome to the ATM”

DREAM:

1. Print “In which bank do you have your account?
2. KOTAK
3. HDFC
4. Scan the choice in variable kr.
5. If the choice is 1 then go to step 112 else go to step 114
6. Print “Welcome to Kotak bank”
7. Call function “kotak\_bank()”
8. If choice is 2 then go to step 115 else go to step 117
9. Print “welcome to HDFC bank”
10. Call function “hdfc\_bank()”
11. Print “Improper choice”
12. Print “please select again”
13. Go to DREAM.
14. Create “int main()”

BUG:

1. Print “Please enter your choice"
2. Print “Enter user's information for database”
3. Print “2. Working of ATM machine”
4. Scan the choice in variable l.
5. If choice is 1 then go to step 127 else go to step 129
6. Call function “create\_database ()”
7. If choice is 2 then go to step 129 else go to step 130
8. Call function “display\_ATM ()”
9. Print “improper selection”
10. Print “please select again:”
11. Go to BUG.
12. Print “press y for the main screen of the ATM(y/n)”
13. Scan the answer in ch.
14. If ch=y or Y then go to step 121 else stop.

List of program:

We have only one program of atm machine in which have two subs part rather two linked list of two banks. And five users in each bank.

Instruction to run the program:

We have only one program so for executing the sub parts we have given the options like to continue the program press 1 else 0, and also we have clearly mentioned the options of yes and no for execution.

File name:

We have one program and two files:

1. HDFC.dat

2. KOTAk.dat

Output generated:

We have generated an output of user’s information of a particular bank, with its current balance and edited information, we have changed it also in from the file for generating an output.

Screen shots:

