****

**Engineering Exploration Project Report On**

**ATM MACHINE**

**G VIJAYA VENKATA SAI 160120733179**

**Under the mentorship of J Shiva Sai Asst. Prof.**

**Core Branch: CSE-3**

**Submitted to**

**Mechanical Engineering Department**

**Chaitanya Bharathi Institute of Technology (A)**

**July, 2021**

**LIST OF CONTENTS**

**1.**Abstract

**2.**Introduction

**3.**Requirement specifications

**4.**Design

**5.**Implementation

**6.**Results

**7.**Conclusion and Future work

References

**ABSTRACT**

The ATM System is the project which is used to access their bank accounts in order to make cash withdrawals. Whenever the user need to make cash withdraws, they can enter their PIN number (personal identification number). Once their withdrawn was successful, the amount will be debited in their account.

The ATM will service one customer at a time. A customer will be required to enter ATM Card number, personal identification number (PIN) – both of which will be sent to the database for validation as part of each transaction. The customer will then be able to perform one or more transactions. Also customer must be able to make a balance inquiry of any account linked to the card.

If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction.

System requirements : C compiler.

Benefits of ATM machine:

• Efficient

• Time saving

• Easy to follow

**1.** **INTRODUCTION**

Present system of banking needs more of both men and time so to make the bank transactions faster and efficient a system like ATM machine is required.

**Present system**

In the manual system ,firstly the bank manager and the staff have to manage information regarding the accounts and transaction of all the customers manually.doing this manual transaction was a really tedious job.secondly information regarding accounts and transactions of customers were to be maintained .This process is time consuming and requires a great manual effort .

Disadvantages of existing system:

* More time consuming process
* More hard work to maintain all the records
* Bulk of papers are to be searched for a single search

**Proposed system**   
This system provides paperless maintenance. Initially a cashier or an clerk can be appointed to do all the transaction and update and maintain records. In the new system the customer himself can do all the transactions and the computerized system automatically updates and maintains the records.

Advantages :

* Less effort to complete transaction.
* Less time required.
* No need to maintain the bulk of papers.
* Makes it easy to search

**Feasibility of the proposed system**

* Technical feasibility

The system provides comprehensive function to make it user friendly.the data entry and report generation is also made easy . It also provides easy retrieval of data . Any machine configuration supports this software.

* Economic feasibility

The cost of converting from manual system to new system is not probably more . We can use the existing hardware resources available and only we are going to use the new software. So it is economically feasible system.

* Operational feasibility  
  Since the system is being user friendly way,the new customers can understand it easily and learn to operate it in less time.

**2. REQUIREMENT SPECIFICATIONS**

**Hardware**

**RAM**

As a developer, you’re going to want to increase every bit of memory that you can. Demanding projects can take up a lot of time and hard drive space, so you need as much RAM as you can get.The best specs for RAM are 16GB. Ideally, it would be best if you shot for something like 32GB. At a minimum, 8GB might suffice.

**Storage**

Arguably the most significant upgrade to your processor is your SSD, or solid-state drive, for optimal speed. The last thing you need while programming is for your system to move at a snail’s pace using a traditional hard drive. That’s why having a SSD drive is arguably more critical than other minimum requirements.

SSD, is necessary for smooth performance and optimal internal storage. You want a ton of data storage for all of your projects, especially if programming turns out to be more than just a hobby. Ideally, you’re going to want something like a 256GB SSD for optimal performance and storage space.

**CPU**

Having the best CPU for programming allows you to code with speed and precision.Your CPU must be up to date. Central processing units are considerably more challenging to replace than graphics cards and SSD drives, especially in laptops . If your laptop CPU is getting old, you may need to buy a new laptop outright. I might suggest to use Intel core i5 or i7 processor .  It lasts a long time and is a powerhouse CPU.

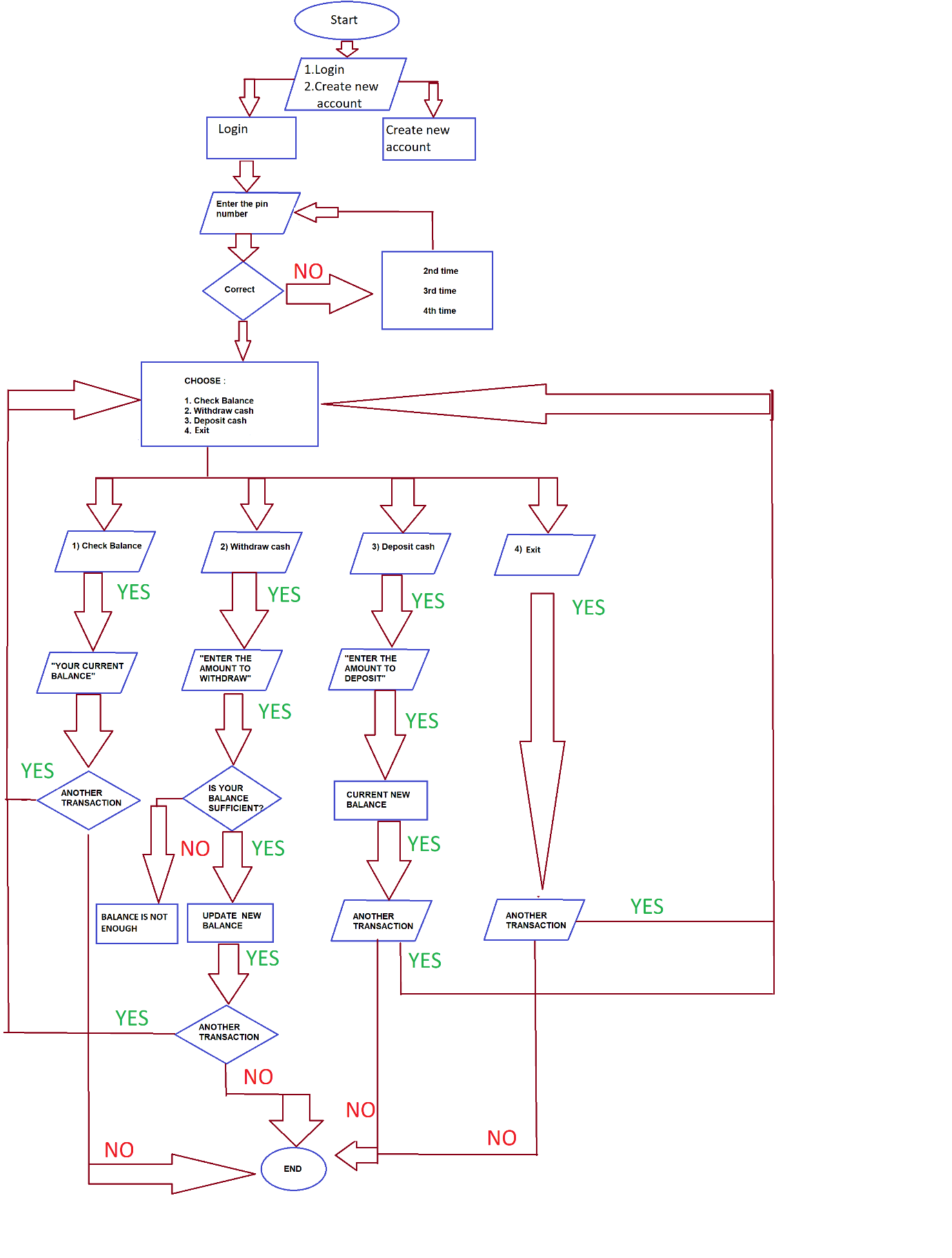
**Software**

**Operating system**

Operating system can be of any basic Windows , Linux, MAC OS .

**Compiler and Environment**

This project uses the GNU C Compiler (GCC) in it so I recommend installing this compiler for use in this project. In this project we are using code blocks environment ,you can use any environment as of your comfort.

**3. DESIGN** 

**Fig 1:** Design flow of atm machine

1. When we first run this code we would get three options 1. Login 2.Create an account 3. Exit .
2. If you are an already existing user then enter 1 to login into your account
3. If your are not existing user then enter 2 and fill the asked details
4. Once account is created you can login into your account.
5. After logining into your account it shows four options 1. Check balance 2. Withdraw cash 3. Deposit 4. Exit
6. Enter the corresponding number as per your choice
7. If you choose 1(Check balance) then you get a message showing your account balance.
8. If you choose 2(Withdraw) then the computer would ask how much amount you want to withdraw . If you enter an amount which exceeds your current balance then it will show a message that you are not having enough balance to continue the transaction else it deducts the amount entered from the current balance and gives the amount.
9. If you choose 3 (Deposit) then the computer would ask how much amount you want to deposit . Enter the amount you want to deposit.
10. At the end of each transaction it would ask whether you want to do another transaction or not .if yes then you would be directed to main menu again if no then the program would be terminated.

**4.IMPLEMENTATION**

**Source Code**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <time.h>

struct User

{

int pin;

float balance;

int accountNumber;

char name[40];

};

void loginMenu();

void signupMenu();

struct User getUserDetails(int);

void checkBalance(struct User);

void moneyWithdraw(struct User, float);

void moneyDeposit(struct User, float);

void updateUserDetails(struct User);

int noUser = 0;

int main()

{

int mainMenu;

int exit = 1;

int anotherTransaction = 0;

while (exit)

{

system("CLS");

printf("\nMain Menu\n1. Login to your account\n2. Create a new account\n3. Exit\n\n");

scanf("%d", &mainMenu);

switch (mainMenu)

{

case 1:

loginMenu();

break;

case 2:

signupMenu();

break;

case 3:

exit = 0;

break;

default:

break;

}

if (mainMenu != 3)

{

if (mainMenu == 2) {

anotherTransaction = 2;

} else {

if (noUser) {

anotherTransaction = 2;

} else {

printf("=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n");

printf("Exit?\n");

printf("< 1 > Yes\n");

printf("< 2 > No\n");

scanf("%d", &anotherTransaction);

}

}

if (anotherTransaction == 1)

{

exit = 0;

}

}

}

}

void loginMenu() {

int accountNumber, optionsChoice, exitLoop=1, pin, actualPin, nextTransaction, depositAmount, withdrawAmount;

system("CLS");

printf("Welcome back! Enter your account number:\n");

scanf("%d", &accountNumber);

struct User userStructure;

int proceed;

userStructure = getUserDetails(accountNumber);

if (userStructure.accountNumber == 0 && userStructure.pin == 0) {

printf("\nUser does not exist :( Create a new user, please enter any number to proceed to main menu\n");

scanf("%d", &proceed);

noUser = 1;

return;

}

actualPin = userStructure.pin;

printf("\nAccount Number: %d,Name:%s, Balance: %.2f\n",userStructure.accountNumber,userStructure.name, userStructure.balance);

printf("Enter your PIN:\n");

scanf("%d", &pin);

if (pin != actualPin) {

int tries = 4;

printf("Invalid PIN entered, ");

while(pin != actualPin && tries > 0) {

printf("You have %d tries left, try again\n", tries);

scanf("%d", &pin);

tries--;

}

system("CLS");

if (tries == 0) {

printf("\n Maximum try limit reached, sending you back to the entry screen\n\n");

return;

}

}

while(exitLoop) {

/\*

1. Check Balance 2. Deposit Money 3. Withdraw Money 4. Exit

\*/

userStructure = getUserDetails(accountNumber);

system("CLS");

printf("\nChoose what you want to do:\n1. Check Balance\n2. Deposit Money\n3. Withdraw Money\n4. Exit\n\n");

scanf("%d", &optionsChoice);

switch (optionsChoice) {

case 1:

system("CLS");

checkBalance(userStructure);

break;

case 2:

system("CLS");

printf("How much do you want to deposit?\n");

scanf("%d", &depositAmount);

moneyDeposit(userStructure, depositAmount);

break;

case 3:

system("CLS");

printf("How much do you want to withdraw?\n");

scanf("%d", &withdrawAmount);

moneyWithdraw(userStructure, withdrawAmount);

break;

case 4:

system("CLS");

exitLoop = 0;

break;

default:

break;

}

if (optionsChoice != 4) {

system("CLS");

printf("=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n");

printf("Would you like to do another transaction?\n");

printf("< 1 > Yes\n");

printf("< 2 > No\n");

scanf("%d", &nextTransaction);

system("CLS");

} else {

nextTransaction = 2;

}

if (nextTransaction == 2) {

exitLoop = 0;

return;

}

}

}

struct User getUserDetails(int accountNumber) {

FILE \*createdFile;

struct User tempuser;

char tempStr[10];

itoa(accountNumber, tempStr, 10);

char \*fileName = strcat(tempStr, ".txt");

createdFile = fopen(fileName, "r");

if(createdFile == NULL) {

tempuser.accountNumber = 0;

tempuser.balance = 0.0;

tempuser.pin = 0;

return tempuser;

}

while(fread(&tempuser, sizeof(struct User),1 , createdFile))

fclose(createdFile);

return tempuser;

}

void createUserAccount(struct User user) {

FILE \*file;

char tempStr[10];

itoa(user.accountNumber, tempStr, 10);

char \*fileName = strcat(tempStr, ".txt");

file = fopen(fileName, "w");

struct User tempUser = user;

fwrite (&tempUser, sizeof(struct User), 1, file);

if (fwrite != 0) {

printf("User created succesfully! %d %.2f %d\n\n", tempUser.accountNumber, tempUser.balance,tempUser.pin);

printf("Enter any number to proceed to main menu\n");

int proceed;

scanf("%d", &proceed);

} else {

printf("Error writing the file\n");

}

fclose(file);

}

void checkBalance(struct User user) {

time\_t timeNow;

time(&timeNow);

printf("Time: %s", ctime(&timeNow));

printf("\nYour balance is: %.2f\n", user.balance);

printf("Enter any number to proceed to the menu\n");

int proceed;

scanf("%d", &proceed);

}

void moneyDeposit(struct User user, float amount) {

user.balance += amount;

updateUserDetails(user);

time\_t timeNow;

time(&timeNow);

printf("Time: %s", ctime(&timeNow));

printf("Money deposited. Your current balance: %.2f\nEnter any number to proceed to the menu\n", user.balance);

int proceed;

scanf("%d", &proceed);

}

void moneyWithdraw(struct User user, float amount) {

if (user.balance > amount) {

user.balance -= amount;

time\_t timeNow;

time(&timeNow);

printf("Time: %s", ctime(&timeNow));

printf("Money deducted from your account. Available balance: %.2f\nEnter any number to proceed to the menu\n", user.balance);

int proceed;

scanf("%d", &proceed);

} else {

printf("Insufficient balance in your account :(\n");

int proceed;

scanf("%d", &proceed);

}

updateUserDetails(user);

}

void updateUserDetails(struct User user) {

FILE \*file;

struct User tempUser = user;

char tempStr[10];

itoa(user.accountNumber, tempStr, 10);

char \*fileName = strcat(tempStr, ".txt");

file = fopen(fileName, "w");

fwrite (&tempUser, sizeof(struct User), 1, file);

if (fwrite != 0) {

printf("User details updated succesfully!\n");

} else {

printf("Error writing the file");

}

fclose(file);

return;

}

void signupMenu() {

printf("\n=====Account Creation=====\n\n");

time\_t seconds;

seconds = time(NULL) % 100000;

struct User user;

int pin;

user.accountNumber = seconds;

user.balance = 0.0;

printf("Welcome! Your account number is %d\n", seconds);

printf("Please choose a 4-digit PIN: \n");

scanf("%d", &pin);

printf("Please enter your name:");

scanf("%s",&user.name);

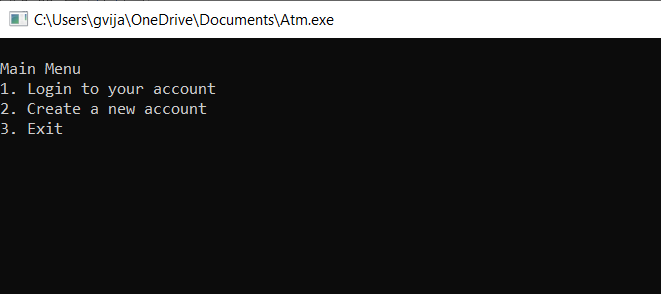
user.pin = pin;

createUserAccount(user);

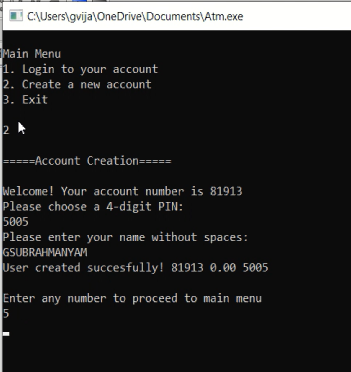
}

**5. RESULT ANALYSIS**

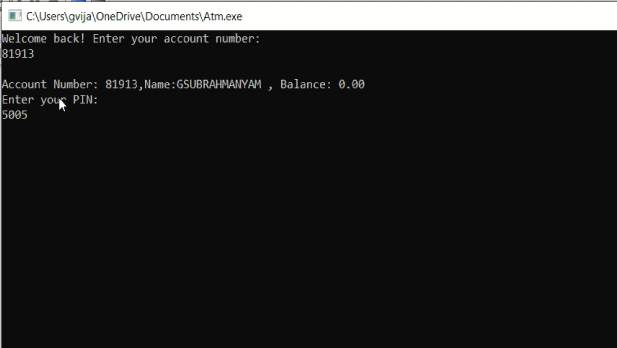
The following pictures show about the execution of the program



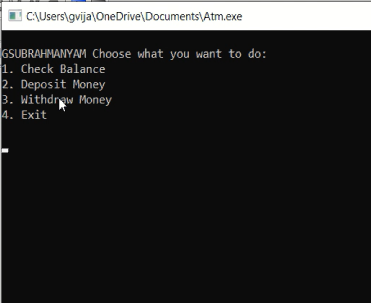
**Fig 2:**  Main menu of ATM



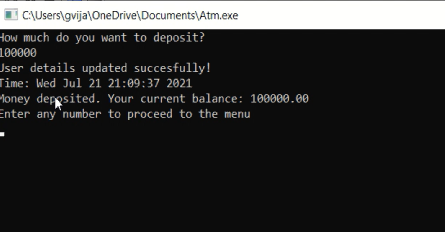
**Fig 3:** Creating a new account



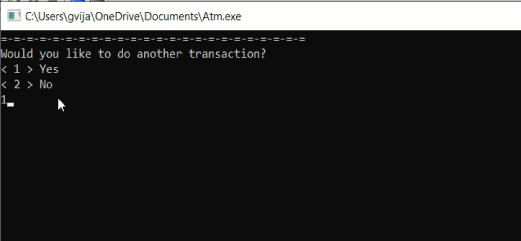
**Fig 4:** Login into the created account



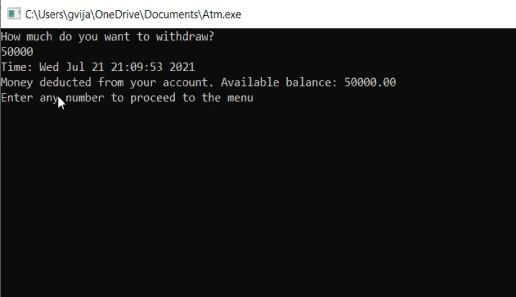
**Fig 5:** Menu to choose what transaction to be done



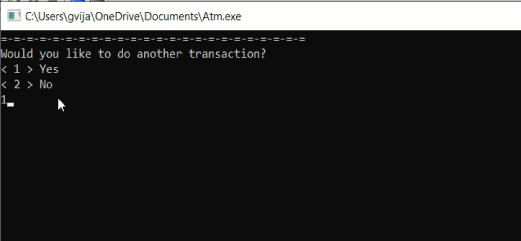
**Fig 6:**  Depositing some amount of money



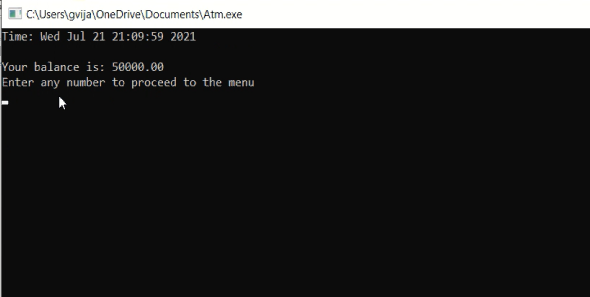
**Fig 7:** Asking for another transaction



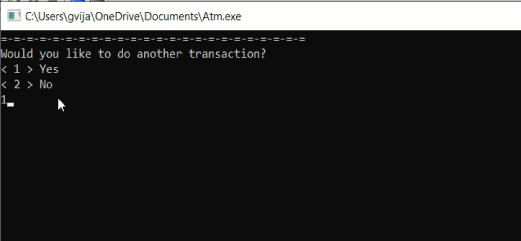
**Fig 8:** Withdrawing some amount of money



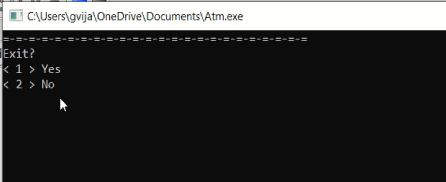
**Fig 8:** Asking for another transaction



**Fig 10:** Checking balance



**Fig 11:** Asking for another transaction



**Fig 11:** Exit option

**7. Conclusion**

ATM machine is an amazing machine helping people to save their time and effort .The project is done completely, if any improvements can be done we can update the code accordingly. If any bugs are found in the future then I will debug the accordingly. This project has helped me a lot in learning c language especially with the files management concept which has many applications.

**Future work**

I want to make changes to this program ,changes like if a person wants to make a transfer of money from his bank account to another bank account then there will be a choice included in the menu to make a money transfer to another account.Generation of OTP(One Time Password) to the person’s mobile number so as to make the transactions secure and many other user friendly options.

**References**

[1].Documentation of c : <https://devdocs.io/c/>

[2].Tutorial for basic atm machine code: <https://www.youtube.com/watch?v=wuAzSLIJnTw>

[3]For login page : [https://stackoverflow.com/questions/29732161/c-login-using-file-handling-and- linked-list](https://stackoverflow.com/questions/29732161/c-login-using-file-handling-and-%20%20linked-list)