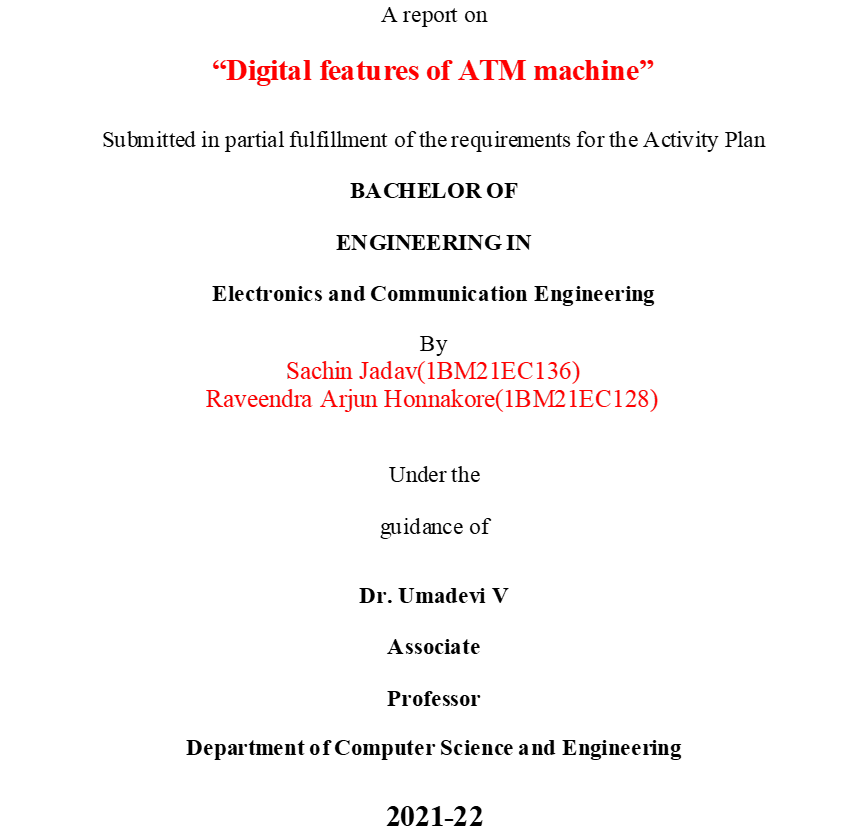
BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)Bull Temple Road, Basavanagudi, Bangalore – 560019





**C E R T I F I C A T E**

This is to certify that the report on**“Digital features of ATM machine”**  is a bona-fide work carried out by Sachin Jadav(1BM21EC136), ,Raveendra Arjun Honnakore(1BM21EC128) as a part of An Activity Plan for the Course **PROBLEM-SOLVING THROUGH PROGRAMMING with course code** **211CC1ESPSP/ 211CC2ESPSP** **Electronics and Communication Engineering** from **Visvesvaraya Technological University, Belgaum** during the year **2021-22**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library.

|  |
| --- |
| **Dr. Umadevi V** |
| **Associate Professor**  **Department of CSE**  **B. M. S. College of Engineering** |

**Acknowledgements**

We would like to thank our guide, Dr. Umadevi V for her patience, first and foremost, and guidance throughout this project. With her support and help, this project rose from the ground up.

Lastly, we would like to thank our college, B. M. S. College of Engineering, for providing us with opportunities to develop new and creative project.

**TABLE OF CONTENTS**

**SL. NO.**

**TITLE**

**PAGE NO.**

1

Introduction

1.1

1.2

1.3

2

Problem Definition

Scope

Abstract

Project Flow

3

Implementation

Experimental Analysis and Results

Conclusion

4

5

6

GitHub link

**1. Introduction**

**1.1 Problem Definition**

Problem related to core domain- One Paragraph

**1.2 Scope**

Background information and where exactly this solution is applicable one to two pages

**1.3 Abstract**

Overview of the exact solution provided, 1 to 2 paragraphs

**2. Project Flow**

Flowchart

**3. Implementation**

Implementation details**.**

**4. Experimental Analysis and Results**

Screenshot of the results and analysis.

**5. Conclusion**

One Paragraph

**6. GitHub link**

GitHub Link were your code has been uploaded

**Introduction**

* The ATM Program in C is written in C programming language which provides an ease to read and comprehend the instructions used. This program for using ATM machine is built on the concept of handling an account individually.
* It can be defined as actually simple code structure of ATM transaction process to be understood by a user. For implementing this project, we may have to use function but in the meantime for easy coding, we may have to switch cause statement.
* From this ATM program in C, we can even use the mini-program for checking the total balance, depositing the amount, and withdrawing the amount from the account definitely since it is not time overwhelming.

**1.1 Problem Definition**

* It can be defined as actually simple code structure of ATM transaction process to be understood by a user. For implementing this project, we may have to use function but in the meantime for easy coding, we may have to switch cause statement.

This C Program performs ATM transaction. The types of ATM transaction are

1) Balance checking

2) Cash withdrawal

3) Cash deposition.

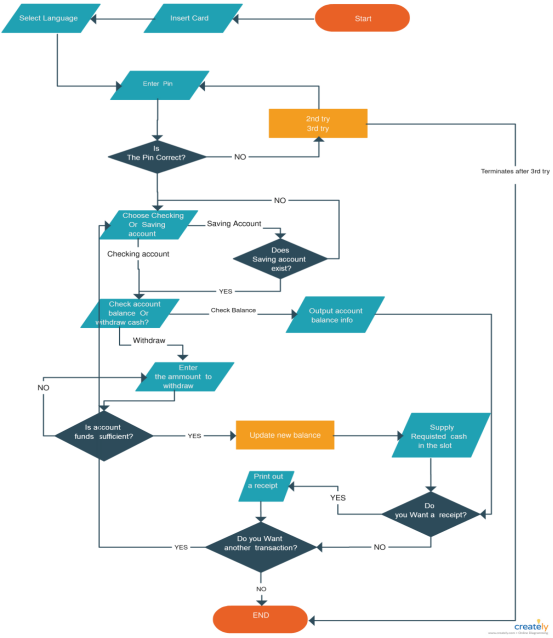
**SCOPE**

* ATM will be a popular "Public Technology". Original equipment manufacturers and vendors will get ample scope for handling ATM machines. Modern ATMs are now capable of personalized branding, CRM applications, integrated fraud alert, customer notifications, and flexible services. The ATM technology has developed to such an extent that some ATMs can memorize consumer preferences as per their past transactions, behavior, and tailor services accordingly. In many cases, ATMs have internet scope which facilitates two-way communications with live agents, provide biometric options, and have the ability to demonstrate personalized advertisements. Maintenance of web enabled ATMs are easy. These ATMs can be quickly connected to central monitoring system of vendors.

**Abstract**

* The ATM System is the project which is used to access their bank accounts in order to make cash withdrawals.
* Whenever user want to make withdrawals, they can enter their ATM card and verified PIN, then user select the withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM system.
* This system will provide for secure connections between users and the bank servers. The whole process will be automated right from PIN validation to transaction completion.
* The card details and PIN database will be a secure module that will not be open to routine maintenance, the only possibility of access to this database will be through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System will enable two important features of a ATM, reduction of human error in the banking system and the possibility of 24 hour personal banking

**PROJECT FLOW**



**An ATM Implementation**

* Asynchronous Transfer Mode (ATM) consists of a specification for packet format and network switching. Each ATM packet, or cell, is fifty-six bytes, forty of which are available for data.
* ATM networks have been praised for their promise to deliver high network bandwidth upon demand. It should be noted that these networks do not provide any form of reliability or flow control.
* Active messaging has been implemented on an ATM network by using the Fore Systems SPARCstation interface and 140 Mb/sec TAXI fiber. The work was done at Cornell University primarily by Thorsten von Eicken, one of the pioneers of active messaging. Von Eicken and his colleagues achieved a maximum bandwidth of 5.5 MB/sec using the remote memory access primitives outlined. It should be noted, however, that this implementation was highly optimized. Through cooperation with Sun and Fore, the Cornell group patched SunOS and installed a device driver that allowed direct access to the network device's FIFOs. Transmitting a message consisted of formatting a cell, writing it to the card's FIFO, and trapping to the kernel. For comparison with less optimized strategies, the group wrote a traditional device driver and achieved 4.5 MB/sec. This result agreed quite well with Fore Systems' own API, which averaged around 4 MB/sec. It was found through exhaustive profiling that throughput was limited almost entirely by the trap to the kernel. This result is not surprising, given the small payload of ATM packets. To achieve anything close to maximum throughput would require a DMA card with an onboard processor that automated cell formatting and transmission. Also, software would be required that would allow the user to initiate these events without suffering the overhead of a trap to the kernel. Despite these difficulties, the performance achieved by the Cornell group is quite impressive, especially when compared with traditional twisted pair networks

**EXPERIMENTAL ANALYSIS**

#include <stdio.h>

unsigned long amount=1000, deposit, withdraw;

int choice, pin, k;

char transaction ='y';

void main()

{

while (pin != 1520)

{

printf("ENTER YOUR SECRET PIN NUMBER:");

scanf("%d", &pin);

if (pin != 1520)

printf("PLEASE ENTER VALID PASSWORD\n");

}

do

{

printf("\*\*\*\*\*\*\*\*Welcome to ATM Service\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("1. Check Balance\n");

printf("2. Withdraw Cash\n");

printf("3. Deposit Cash\n");

printf("4. Quit\n")

rintf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?\*\n\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("\n YOUR BALANCE IN Rs : %lu ", amount);

break;

case 2:

printf("\n ENTER THE AMOUNT TO WITHDRAW: ");

scanf("%lu", &withdraw);

if (withdraw % 100 != 0)

{

printf("\n PLEASE ENTER THE AMOUNT IN MULTIPLES OF 100");

}

else if (withdraw >(amount - 500))

{

printf("\n INSUFFICENT BALANCE");

}

else

{

amount = amount - withdraw;

printf("\n\n PLEASE COLLECT CASH");

printf("\n YOUR CURRENT BALANCE IS%lu", amount);

}

break;

case 3:

printf("\n ENTER THE AMOUNT TO DEPOSIT");

scanf("%lu", &deposit);

amount = amount + deposit;

printf("YOUR BALANCE IS %lu", amount);

break;

case 4:

printf("\n THANK U USING ATM");

break;

default:

printf("\n INVALID CHOICE");

}

printf("\n\n\n DO U WISH TO HAVE ANOTHER TRANSCATION?(y/n): \n");

fflush(stdin);

scanf("%c", &transaction);

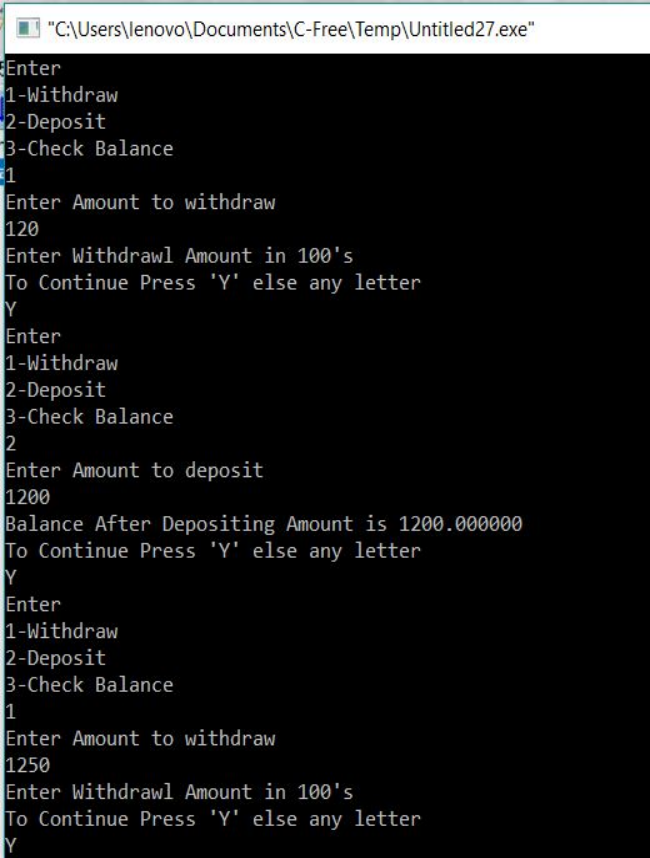
if (transaction == 'n'|| transaction == 'N')

k = 1;

} while (!k);

printf("\n\n THANKS FOR USING OUT ATM SERVICE");

}



**Conclusion**

* Programming in C is perfect for beginners to code and therefore before proceeding it is essential to gain a few basic codes of C programming.
* For creating an ATM machine program using C, we need to implement the four fundamental concepts of each ATM system that exists, it includes cash withdraw, cash deposit, account balance checks, and functionality for another transaction or termination.

**GITHUB LINK**

<https://github.com/1234raveendra/digital-atm-machine.git>