**INDEX**

**Table of contents Page No.**

1. Preface 6
2. Need for developing management systems 7
3. System analysis 8
4. Core objective 8
5. Existing system and its disadvantages 9
6. Proposed system 11
7. Entity relation diagrams 12
8. Code 18
9. Outputs 44
10. Hardware and software requirements 48
11. System testing 49
12. Unit testing 49
13. Feasibility study- 50
14. Technical feasibility
15. Operational feasibility
16. Economic feasibility
17. Program maintenance 52
18. Corrective Maintenance
19. Adaptive Maintenance
20. Preventive Maintenance
21. Perfective Maintenance
22. Stylistic guidelines that we followed while building this program. 54
23. Characteristics of a good program 56
24. Conclusion and further scope 57

**PREFACE**

Project is a special significance for an IT student; it has a special role too. He learns to tackle different problems that strike while developing any computer application. He gets an opportunity to explore and use various functions available and some more really powerful features of the language. Personally it has been a highly rewarding and enriching experience for our team as it has given us insights into the immense plethora of opportunities and prospects that this field provides us and has also given us an idea of the expectations and requirements that the society has from IT professionals. This project has also kindled in us the growing fire in us to take up this field as a profession and contribute in its development.

**NEED FOR DEVELOPING**

**MANAGEMENT SYSTEMS**

Digitization has become the burning issue in our country. India is growingly becoming moving towards digitization and many organizations have been successful in this endeavour.

But while most leading firms stride forward in this area, many small scale businesses have lagged behind and it is our duty, as both responsible citizens as well as Computer Science students, to take everyone with us on the path of development.

**SYSTEM ANALYSIS**

This software will be a replacement of existing manual system. The incentive for replacement is that the former record keeping system was manual and more prone towards error, keeping in view that the user of the system will be an ordinary worker with little knowledge of computer so this software will be user friendly, efficient and easy to use.

**CORE OBJECTIVES**

* Minimize Wastage of Time.
* Minimize the use of Stationary
* Librarian with a little training on application can handle the system easily.
* Efficient data access

**existing SYSTEM**

Manual library management system:

A manual library management system includes data scrambled randomly. It includes tedious and cumbersome records of thousands of books and their details, which makes it very difficult to search for a particular book, author or genre. It requires meticulous updating and systematic arrangement of assets at all times.

All the members of the library were given individual book issuing cards and maintaining and safeguarding those documents was cumbersome. Those records were easily prone to manipulation and could be tampered with.

**DISADVANTAGES**

* Student accounts on registers (wastage of time in creating new account and making entries on behalf of a members were quite a lengthy process)
* Keeping track of the book records was very difficult.
* Purchase of books from supplier and keeping records on paper
* Book recommendations were also not considered properly.
* Redundancy and fake issue cards was difficult to keep track of.

**PROPOSED SYSTEM**

Services of New System:

* Hassle free record keeping
* No need for bulky files and registers.
* Saves storage and utility space
* No need for tedious maintenance of individual library cards
* Computerized-record keeping for books. (more easy and efficient)
* Creation, deletion and update of user account is possible in few minutes.
* Searching for a specific book on search -engine without any wastage of time.

**Entity Relation diagrams**



**CODE**

//PROGRAM FOR LIBRARY MANAGEMENT SYSTEM

//TEAM NAME:- IDA

/\*DISCLAMIER

You should have a lot of MEMORY to run this PROGRAM.

Otherwise PROGRAM would TERMINATE itself.

\*/

#include<graphics.h>

#include<process.h>

#include<iomanip.h>

#include<fstream.h>

#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<string.h>

int TOTAL\_BOOKS=0;

void CREATE\_CIRCLE(int ,char\*, char\* , char\*, char\*);

void LOAD();

void Layout();

void Welcome\_Screen();

//CLASS LIBRARY~TO RECORD DETAILS OF BOOKS

class LIBRARY{

char \*Name\_of\_Book;

char \*Name\_of\_Author;

int Book\_No;

char rent;

public : LIBRARY()

{ strcpy(Name\_of\_Book,"10 Ways to kill you ");

strcpy(Name\_of\_Author,"Yourself ");

Book\_No=69;

rent='y';

}

void ADD();

void DELETE();

void DISPLAY();

void RENT();

void RETURN();

int getBook\_No()

{ return Book\_No; }

}L;

void LIBRARY::ADD()

{

int l=getmaxx()/2,m=getmaxy()/2;

fstream f("LIBRARY.txt",ios::out||ios::in||ios::app);

f.seekg(0);

char ch='y';

while(ch=='y'||ch=='Y')

{

clearviewport();

rectangle(l-200,m-50,l+200,m+75);

rectangle(l-195,m-45,l+195,m+70);

gotoxy(23,14);

cout<<"Enter Book No : ";

cin>>Book\_No;

gotoxy(23,16);

cout<<"Enter Name of Book : ";

gets(Name\_of\_Book);

gotoxy(23,18);

cout<<"Enter Name of Author : ";

gets(Name\_of\_Author);

rent='y';

f.write((char\*)&this,sizeof(this));

gotoxy(18,25);

cout<<"Do you want to Enter More Books (y/n): ";

cin>>ch;

TOTAL\_BOOKS++;

clearviewport();

}

f.close();

}

void LIBRARY::DELETE()

{

fstream f("LIBRARY.txt",ios::in);

fstream f1("temp.txt",ios::out);

int l=getmaxx()/2,m=getmaxy()/2;

int u;

clearviewport();

gotoxy(28,15);

rectangle(l-200,m-50,l+200,m+75);

rectangle(l-195,m-45,l+195,m+70);

cout<<"Enter Book No : ";

cin>>u;

int z=0;

while(f.read((char\*)&this,sizeof(this)))

{

if(!(Book\_No==u))

{

TOTAL\_BOOKS--;

f1.write((char\*)&this,sizeof(this));

}

else

z=1;

}

rectangle(l-200,m-50,l+200,m+75);

rectangle(l-195,m-45,l+195,m+70);

gotoxy(28,18);

cout<<"Status : ";

if(z==0)

{

gotoxy(37,18);

cout<<"Book Doesn't Exist ";

}

else

{

gotoxy(37,18);

cout<<"Book Successfully Deleted !!";

}

remove("LIBRARY.txt");

rename("temp.txt","LIBRARY.txt");

f.close();

f1.close();

getch();

clearviewport();

}

void LIBRARY::DISPLAY()

{

int l=getmaxx()/2,m=getmaxy()/2;

int k;

rectangle(l-200,m-50,l+200,m+75);

rectangle(l-195,m-45,l+195,m+70);

gotoxy(25,16);

cout<<"Enter Book No : ";

cin>>k;

fstream f("LIBRARY.txt",ios::in);

int u=0;

clearviewport();

rectangle(l-200,m-50,l+200,m+75);

rectangle(l-195,m-45,l+195,m+70);

while(f.read((char\*)&L,sizeof(L)))

{

if(Book\_No==k)

{

u=1;

cout<<"Book : "<<Name\_of\_Book

<<"Author : "<<Name\_of\_Author

<<"Book No : "<<Book\_No

<<"Available : "<<rent;

}

}

if(u==0)

{

gotoxy(28,18);

cout<<"Book Doesn't Exist!!";

}

f.close();

getch();

clearviewport();

}

void LIBRARY::RENT()

{

clearviewport();

int l=getmaxx()/2,m=getmaxy()/2;

rectangle(l-200,m-50,l+200,m+75);

rectangle(l-195,m-45,l+195,m+70);

fstream f("LIBRARY.txt",ios::in||ios::out);

int y;

gotoxy(25,16);

cout<<"Enter Book no : ";

cin>>y;

int u=1;

while(f.read((char\*)&this,sizeof(this)))

{

if(Book\_No==y)

{

rent='n';

f.seekp(-1\*sizeof(this));

f.write((char\*)&this,sizeof(this));

u=0;

}

}

f.close();

gotoxy(24,18);

cout<<"Status : ";

if(u==1)

{

gotoxy(32,18);

cout<<"Book Doesn't Exist";

}

else

{

gotoxy(32,18);

cout<<"Book has been Issued";

}

getch();

clearviewport();

}

void LIBRARY::RETURN()

{

clearviewport();

int l=getmaxx()/2,m=getmaxy()/2;

rectangle(l-200,m-50,l+200,m+75);

rectangle(l-195,m-45,l+195,m+70);

gotoxy(25,16);

fstream f("LIBRARY.txt",ios::in||ios::out||ios::app);

int y;

int u=1;

cout<<"Enter Book no : ";

cin>>y;

while(f.read((char\*)&this,sizeof(this)))

{

if(Book\_No==y)

{

rent='y';

f.seekp(-1\*sizeof(this));

f.write((char\*)&this,sizeof(this));

u=0;

}

}

f.close();

gotoxy(24,18);

cout<<"Status : ";

if(u==1)

{

gotoxy(32,18);

cout<<"Book Doesn't Exist";

}

else

{

gotoxy(32,18);

cout<<"Book has been Returned";

}

getch();

clearviewport();

}

//MEMBERSHIP~FOR LOGIN PURPOSE

struct MEMBERSHIP{

char \*Username;

char \*Password;

}M;

void REGISTER()

{

clearviewport();

gotoxy(25,14);

cout<<"Enter Your Username : ";

cin>>M.Username;

gotoxy(25,16);

cout<<"Enter Password : ";

cin>>M.Password;

fstream f("MEMBERSHIP.txt", ios::in);

f.write((char\*)&M,sizeof(M));

f.close();

LOGIN();

}

void LOGIN()

{

int l=getmaxx()/2,m=getmaxy()/2;

char \*ch,\*ch1;

clearviewport();

rectangle(l-200,m-15,l+200,m+75);

rectangle(l-205,m-20,l+205,m+80);

outtextxy(l-175,m+5,"ENTER MEMBERSHIP ID : ");

gotoxy(42,16);

cin>>ch;

outtextxy(l-175,m+40,"ENTER PASSWORD : ");

gotoxy(38,18);

cin>>ch1;

fstream f("MEMBERSHIP.txt",ios::in||ios::out);

int p=0;

while(!f.eof())

{

f.read((char\*)&M,sizeof(M));

if(!(strcmpi(ch,"SONAM\_GUPTA"))&&!(strcmpi(ch1,"BEWAFA"))||(!strcmp(M.Username,ch)&&strcmp(M.Password,ch1)))

p=1;

}

f.close();

if(p==1)

{

LOAD();

}

else

{

clearviewport();

outtextxy(l-100,m,"STOP TRYING");

delay(2000);

exit(0);

}

}

char \*p="Library Management System";

int n;

//LOADING SCREEN

void LOAD()

{

int l=getmaxx()/2,m=getmaxy()/2;

clearviewport();

setfillstyle(5,getmaxcolor()-5);

fillellipse(l,m,l/2+10,l/2+10);

setfillstyle(5,0);

fillellipse(l,m,l/2,l/2);

settextstyle(6,0,4);

outtextxy(l-60,m-20,"LOADING!!");

outtextxy(l-58,m-20,"LOADING!!");

delay(3000);

}

//MENU\_DRIVEN FUNCTIONS

void ABOUT()

{

clearviewport();

gotoxy(0,18);

cout<<"ABOUT\n\nDHANUSH : TEAM CAPTAIN\nISHITHA : PROJECT MANAGER\nAARYA : PROGRAMMER ";

getch();

clearviewport();

}

void MODIFY()

{

int v;

CREATE\_CIRCLE(3,"1.ADD","2.DELETE","3.DISPLAY","MODIFY");

cout<<"Enter Choice : ";

cin>>v;

clearviewport();

switch(v)

{

case 1: clearviewport();

L.ADD();

break;

case 2: clearviewport();

L.DELETE();

break;

case 3: clearviewport();

L.DISPLAY();

break;

default: ;

}

}

void RENT\_RETURN()

{

int v;

CREATE\_CIRCLE(2,"1.ISSUE ","2.RETURN","asdas","ISSUE/RETURN");

cout<<"Enter Choice : ";

cin>>v;

switch(v)

{

case 1: clearviewport();

L.RENT();

break;

case 2: clearviewport();

L.RETURN();

break;

default: ;

}

}

//FUNCTION FOR PROCESSING FROM THE INPUT

void SWITCH()

{

int l=getmaxx()/2,m=getmaxy()/2;

switch(n)

{

case 1: clearviewport();

ABOUT();

break;

case 2: clearviewport();

MODIFY();

break;

case 3: clearviewport();

RENT\_RETURN();

break;

case 4: clearviewport();

outtextxy(l-30,m,"Thanks for using!!");

delay(2000);

exit(0);

default : ;

}

}

//GRAPHICAL USER INTERFERANCE IS DONE BY USING THIS FUNCTION

void Layout()

{

int l=getmaxx()/2,m=getmaxy()/2;

circle(l,m,100); //Primary Circle

circle(l,m,95);

settextstyle(1,0,3);

outtextxy(l-40,m-15,"Library");

circle(l-250,m-50,50); //Seconday Circle-1

circle(l-250,m-50,45);

line(l-100,m,l-200,m-50);

settextstyle(1,0,1);

outtextxy(l-290,m-60,"1.ABOUT");

circle(l+250,m-50,50); //Seconday Circle-2

circle(l+250,m-50,45);

line(l+100,m,l+200,m-50);

settextstyle(1,0,1);

outtextxy(l+210,m-60,"2.MODIFY");

circle(l-200,m+150,50); //Seconday Circle-3

circle(l-200,m+150,55);

line(l-65,m+75,l-150,m+150);

settextstyle(1,0,1);

outtextxy(l-235,m+132,"3.ISSUE/");

outtextxy(l-235,m+150,"RETURN");

circle(l+200,m+150,50); //Seconday Circle-4

circle(l+200,m+150,45);

line(l+65,m+75,l+150,m+150);

settextstyle(1,0,1);

outtextxy(l+175,m+140,"4.EXIT");

int t=150/8;

rectangle(l-155,m-t-175-5,l+155,m+t-175+5);

rectangle(l-150,m-t-175,l+150,m+t-175);

settextstyle(5,0,2);

outtextxy(l-5\*strlen(p),m-strlen(p)/4-175-10,p);

gotoxy(30,24);

cout<<"Enter Choice: ";

cin>>n;

SWITCH();

Layout(); //USER CAN ONLY QUIT THE PROGRAM MANUALLY

}

//IN ORDER TO CREATE CIRCLE AT SPECIFIED POSITION

void CREATE\_CIRCLE(int q,char \*x, char \*y, char \*z,char \*p)

{

int l=getmaxx()/2,m=getmaxy()/2;

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

{

clearviewport();

circle(l,m-i,l/2);

circle(l,m-i,l/2+5);

delay(1);

}

int h=60;

if(q==3)

h=40;

settextstyle(3,0,1);

outtextxy(l-h,40,p);

int f=3\*l/4;

circle(l-f,m,50);

circle(l-f,m,55);

line(l-f,m-50,l/2+45 ,m/2-8);

outtextxy(l-f-30,m-15,x);

circle(l+f,m,50);

circle(l+f,m,55);

line(l+f,m-50,3\*l/2-45,m/2-8);

outtextxy(l+f-40,m-15,y);

if(q==3)

{

circle(l,m+m/3,50);

circle(l,m+m/3,55);

line(l,m-80,l,m+m/3-50);

outtextxy(l-40,m+m/3-15,z);

}

}

//THIS FUNCTION IS USED FOR LOGGING IN

void LOGON()

{

int l=getmaxx()/2,m=getmaxy()/2;

char \*ch,\*ch1;

rectangle(l-100,m-15,l+100,m+85);

rectangle(l-105,m-20,l+105,m+90);

gotoxy(35,16);

cout<<"1.LOGIN ";

gotoxy(35,18);

cout<<"2.REGISTER ";

gotoxy(30,20);

cout<<"Enter Choice : ";

int n;

cin>>n;

switch(n)

{

case 1: LOGIN();

break;

case 2: REGISTER();

break;

default : ;

}

Welcome\_Screen();

}

//THIS FUNCTION USED AS ANIMATION FOR WELCOMING AFTER LOOGING INTO YOUR ACCOUNT

void Welcome\_Screen()

{

int l=getmaxx()/2,m=getmaxy()/2;

int t;

for(int i=100;i<=150;i++)

{

t=i/8;

clearviewport();

rectangle(l-i,m-t,l+i,m+t);

delay(50);

}

for(i=0;i<175;i++)

{

clearviewport();

rectangle(l-155,m-t-i-5,l+155,m+t-i+5);

rectangle(l-150,m-t-i,l+150,m+t-i);

settextstyle(5,0,2);

outtextxy(l-5\*strlen(p),m-strlen(p)/4-i-10,p);

delay(15);

}

Layout();

}

//I KNOW THIS IS JUST ONE LINE MAIN FUNCTION

void main()

{

int x=DETECT,y;

initgraph(&x,&y,"C:\\TURBOC3\\BGI");

LOGON();

}

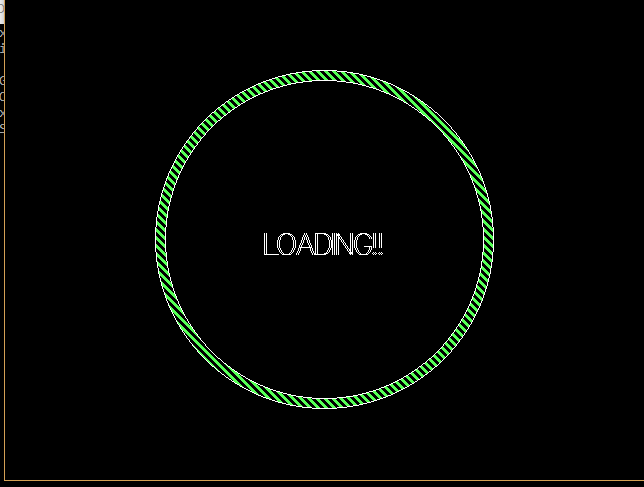
/\* PROGRAM BY:-DHANUSH

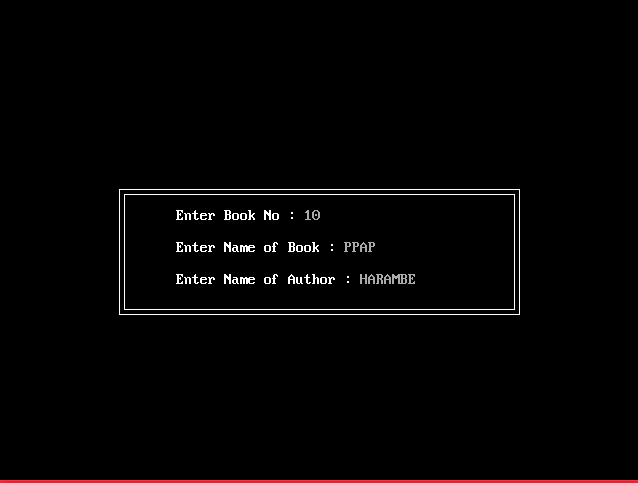
ISHITHA

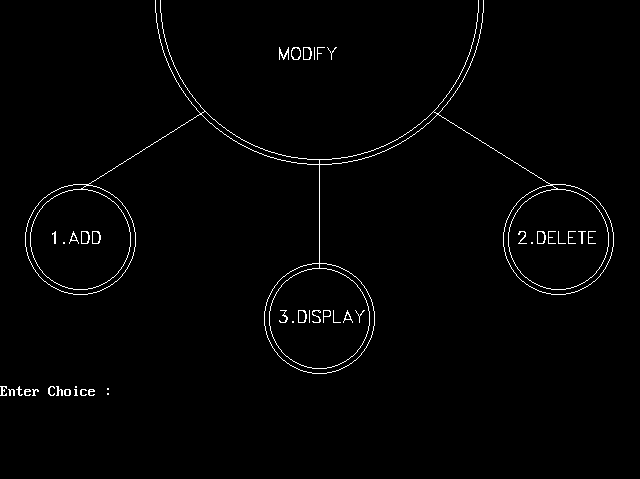
AARYA

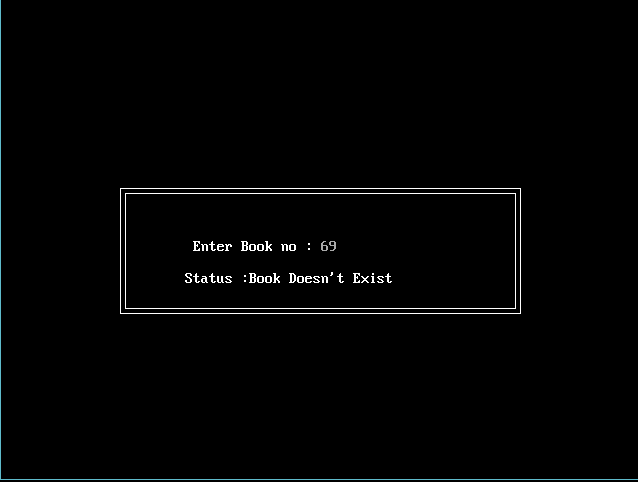
COPYRIGHTS TO THE TEAM IDA \*/

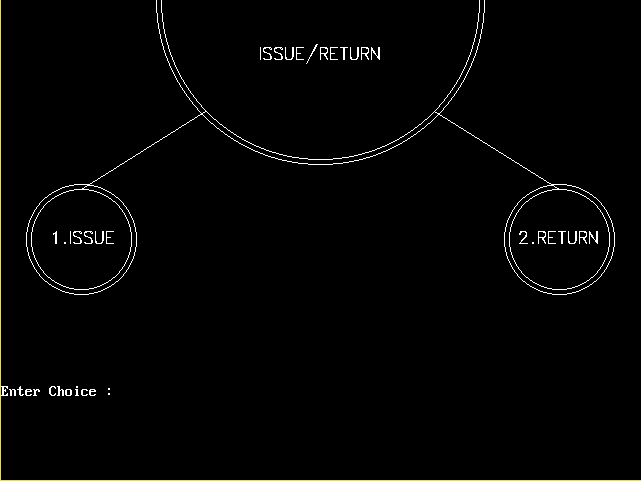
**OUTPUT**

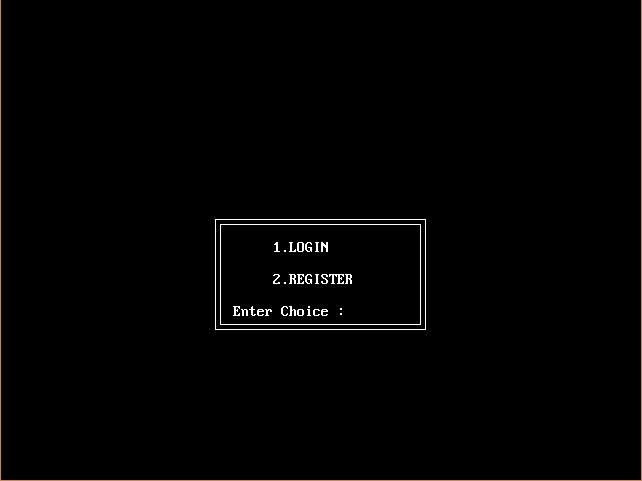




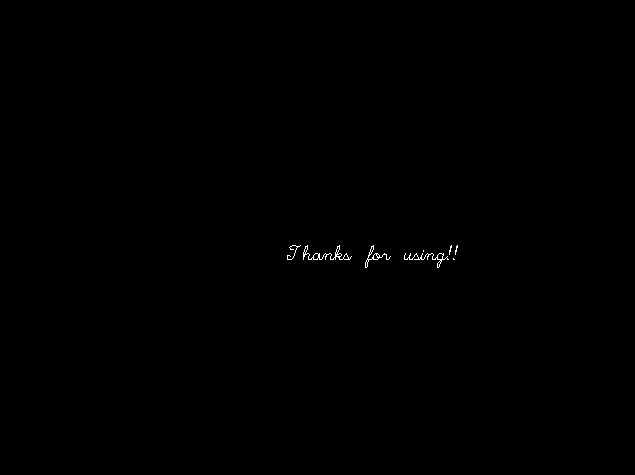












**HARDWARE AND SOFTWARE REQUIREMENTS**

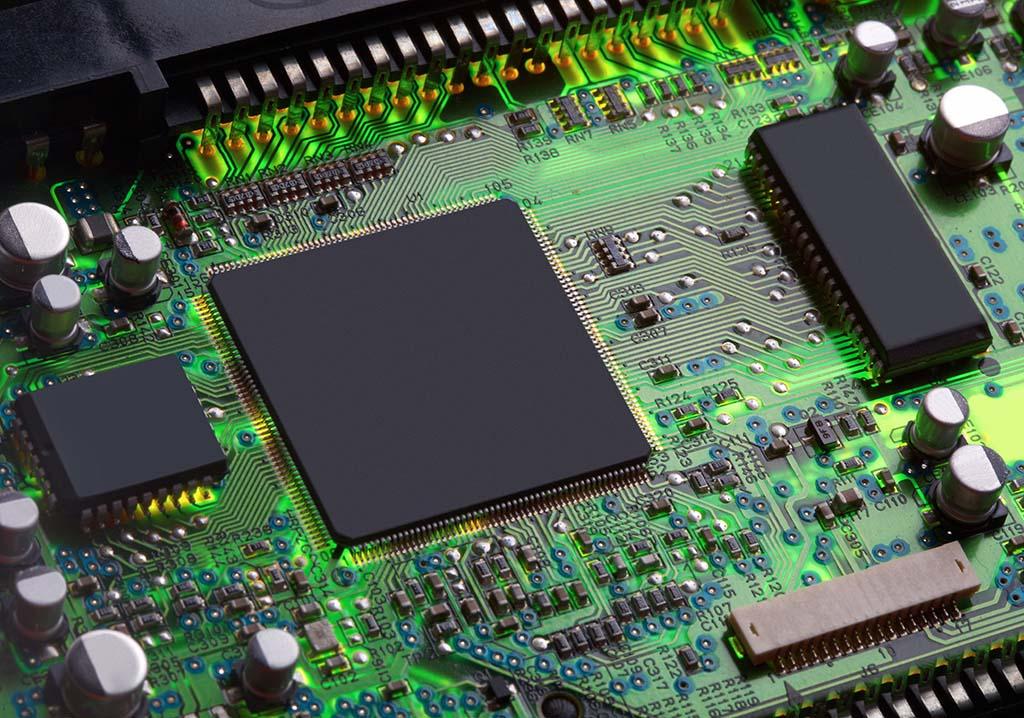
**Hardware specifications:**

Processor : INTEL CORE i3

RAM : 2GB or above

Hard disk capacity : More than 80 MB

Display Type : Standard VGA/SVGA



**Software specification:**

Operating system : Windows 7 or above

Backbend : ORACLE 8.0

Frontbend : Visual Basic 6.0

**SYSTEM TESTING**

The aim of the system testing process was to determine all defects in our project .The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing 1.Unit testing 2.Integration testing

**UNIT TESTING**

Unit testing is undertaken when a module has been created and successfully reviewed .In order to test a single module we need to provide a complete environment ie besides the module we would require ∙ The procedures belonging to other modules that the module under test calls ∙ Non local data structures that module accesses ∙ A procedure to call the functions of the module under test with appropriate parameters Unit testing was done on

**Feasibility Study**

Feasibility is a determination of whether or not a project is worth doing. Feasibility Study is performed for determining the feasibility of a project. The content and recommendations of such a study will be used as a sound basis for deciding to proceed, postpone, or cancel the project. In the conduct of feasibility study, we will usually consider following inter-related type of feasibility.

1. Technical feasibility

We concern here with specifying Equipment and software that will satisfy the user requirement. It will run on any platform (machine), since the C# is considered platform independent. It will run with minimum system requirements and with minimum system resources acquired during run. It will need a web server, to which it gets from the internet; at run time. Expandability will be maintained in the new system. New modules can be added later on the application, if required in the future.

1. Operational feasibility

The system will be easy to use as user interface is GUI based. The system is easy to use so no any special skills will be required to use the system. New user will find it easy to use. So the project will be operationally feasible.

1. Economic feasibility

The procedure is to determine the benefit and savings that are expected from the project and compare them with the cost. As internet is the cheapest way of communication, we can perform communication using web. The cost is just the cost of using the internet based on the channel allocation. So the project will be economically feasible.

**PROGRAM MAINTENANCE**

Programming maintenance refers to the modification of a program, after it has been completed, in order to meet changing requirements or to take care of errors that show up. There could be four types of maintenance:

1)Corrective Maintenance: After completion when it put to operations some errors might show up because of unexpected situations. Such errors are there, which is done to correct the errors is called corrective maintenance.

2)Adaptive Maintenance: change in the environment in which an information system operates may also lead to the system maintenance. To accommodate changing needs time to time maintenance is called Adaptive Maintenance.

3)Preventive Maintenance: If possible errors could be anticipated before they actually occur. This type of maintenance aims at preventing errors called Preventive Maintenance.

4)Perfective Maintenance: Every year new technologies come with new features and new facilities. If the existing system is maintained to keep attuned with new features, new facilities, new capabilities is said to be perfective maintenance.

**STYLISTIC GUIDELINES**

**THAT WE FOLLOWED WHILE**

**BUILDING THIS PROGRAM**

1. Meaningful Names for identifiers: A programmer to give the meaningful names to each section of the program so that it can help him to identify the variable used for specific purpose. This helps him to execute the right elements during the complex run of a program.

2. Ensure clarity of expression: Expression carry out the specified action. Thus they must be clearly understood by the users. There should not be any compromise with the clarity of expression.

3. Use of comments and indentations: Comments generally are used as the internal documentation of a program. If comments are used in the program they will guide the program while debugging and checking. While indentation is the proper way of writing to avoid the confusion regarding the flow of program, this highlights nesting of groups of control statements.

4. Insert blank lines and blank spaces: Blank lines should be used to separate long, logically related blocks of code. Specifically normally in programming the standard for the use of spaces is to follow normal English rules.

5. Statements: Each statement should appear on a separate line. The opening brace following a control statement such as if or while should appear on the line after the if or while, lined up with the left of the control statement, and the closing brace should appear on its own line, lined up with the left of the control statement.

**CHARACTERISTICS OF A GOOD PROGRAM**

Following are the characteristics of a good program:

1. Effective and efficient: The program produces correct results and is faster, taking into account the memory constraints.

2. User friendly: The program should be user friendly. The user should not be confused during the program execution. The user should get correct direction and alerts when he is going through the program.

3. Self-documenting code: A good program must have self-documenting code. This code will help the programmer to identify the part of the source code and clarify their meaning in the program.

**CONCLUSION & FUTURE SCOPE**  
This website provides a computerized version of library management system which will benefit the students as well as the staff of the library. It makes entire process online where student can search books, staff can generate reports and do book transactions. It also has a facility for student login where student can login and can see status of books issued as well request for book or give some suggestions. It has a facility of teacher’s login where teachers can add lectures notes and also give necessary suggestion to library and also add info about workshops or events happening in our college or nearby college in the online notice board. There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility , a feature Of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfills each users need in the best way possible.

**Bibliography**

* Help from our parents.
* Help from teachers.
* Websites that we reffered to –
* Codeblocks.com
* Treehouse.in
* Scribd.in
* Slideshare.in
* Books that we reffered to –
* Computer science with C++ by Sumita Arora