ACKNOWLEGDEMENT

Developing Soccer Manager has been a very informative experience for us, and we would like to thank each and every person associated with the completion of the Software.

First of all, we would like to thank our Computer Science teacher, Mrs. Kavitha, for guiding us throughout the development of the project. Without her, the project would have never taken the shape that it has taken today.

We thank our school for providing us the facilities to build this project. Without the support provided by the school in terms of infrastructure, our project would have never reached completion.

Last but not the least, we would like to thank our parents for providing us facilities at home to take the project to greater heights, and for tolerating our constant usage of computers.

*Nagabharan N*

*and Arvind P*

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **NO.** | **CHAPTER** | **PAGE NO.** |
| **1** | **SYSTEM REQUIRMENTS** | **3** |
| **2** | **NEED FOR THE PROJECT** | **4** |
| **3** | **ABOUT THE PROJECT** | **5** |
| **4** | **FLOW CHART** | **7** |
| **5** | **ALGORITHM** | **9** |
| **6** | **SOURCE CODE** | **10** |
| **7** | **OUTPUT SCREENSHOTS** | **56** |
| **8** | **SCOPE FOR IMPROVEMENT** | **59** |
| **9** | **BIBLIOGRAPHY** | **60** |

SYSTEM REQUIREMENTS

Soccer Manager is designed to run successfully on any computer manufactured within the last five years. However, due to the high precision requirement of Soccer Manager, some processes may take considerable time.

Soccer Manager will run with appreciable speed at high resolution on any computer that meets the following specifications:

1. Intel® 1.5 GHz (Intel® Pentium® 4 or better) or AMD® (AMD® Athlon® XP or better) equivalent.
2. 1 MB of free hard disk space.
3. 128 MB or more of non-shared System Memory (256 MB advised on Windows XP).
4. Floppy drive / CD Drive / USB Port for installation source.
5. Microsoft® Windows® 95/98/2000/ME/XP Operating System.
6. Colour Monitor capable of 16-bit color and minimum 640x480 Resolution.
7. PS/2 or USB Keyboard for navigation.

In addition to the above, the Borland® Turbo C++ IDE (Version 3.0 or later) maybe required to view and edit the Soccer Manager Source Code.

Soccer Manager will not execute on Windows® Vista due to full-screen limitations of Windows® Vista.

NEED FOR THE PROJECT

As we started doing our project, we realized that Database Management is simple to understand and it a lot easier to maintain data in an organized fashion with the usage of computer technology. Also the concepts covered in this project – classes and file handling were topics that a lot of students studying in 12th grade had trouble understanding.

Understanding these concepts is no doubt very difficult for a person who has been introduced to it for the first time. It requires a lot of visualization and perception on the part of the student and when different parameters are introduced it becomes all the more difficult. Hence, this project provides a very comprehensive view of the topic and makes learning a lot easier and better.

This project also provided us with hands-on experience in usage of a lot of concepts of C++ and improved our understanding skills. There is also no readily available software that combines these concepts and presents them in such an easy manner.

ABOUT THE PROJECT

*C++ in its vast treasure of programming features offers many avenues where even a seemingly inimitable real life feature can be successfully stimulated to form a virtual world of sorts.*

*In such a background it is our effort to design a project which aims to simplify and give a user-friendly application “SOCCER MANAGER”. It is an application that can be used by football managers to maintain a record on their team and players.*

*The basic features of the afore mentioned project will be as follows:*

* *Use of various C++ standard library functions including those of graphics.h*
* *Considerable use of graphic aids and supporting text wherever required*
* *User friendly approach- Use of Menus: help, credits etc as also filtered outputs depending on user’s choice*
* *Importing pictures using the concept of file pointers*
* *Extensive use of file handling concepts*
* *Security using the password check*

j0292020

This project imports different files for execution during

runtime.

This project, named as SOCCER MANAGER since it is a one stop destination for everything related to managing a soccer team and player efficiently. It accomplishes this with the use of many pictures and user friendly menus.

FLOW CHARTS

Start

Menu

Accept key. If yes down else main

1

Exit

Team Details

Player Details

3

2

2

3

1

Exit from program

Print credits and other information

Print all details about operation

Database

Help

Credits

Exit

2

Accept key. If yes down else main

Add details of team or player accordingly

Modify details of team or player accordingly using the gicen code

Display details of team or player accordingly using the code

Search details of team or player accordingly using the code

Delete details of team or player accordingly using the given code

Add

Modify

Delete

Search

Display

Exit

ALGORITHMS

3

Add details of team or player accordingly

Modify details of team or player accordingly using the given code

Delete details of team or player accordingly using the given code

Search details of team or player accordingly using the code

Display details of team or player accordingly using the code

Accept key. If yes down else main

* passw.cpp : A cpp file which includes the inbuilt fuctions and defines the various functions that are used in the project
* void clear(): Clears the screen and reinitializes the graphic drivers
* void img(): Prints the image on the screen using graphics
* void bodr(): Prints the border in the welcome screen
* void welcome(): Shows the welcome screen with the border embedded at the corners
* void options1(): Shows the main menu which links to other submenus and defines keyboard keys for various options
* void options2(): Shows file handling menu to perform different operations
* void player(): Imports an image and performs all functions related to maintaining a player database
* void tmdet(): Imports an image and performs all functions related to maintaining a team database
* void options3(): Shows a submenu which links to the respective database using various keyboard keys
* void database(): Links to the various databases
* void help(): Shows the help screen and calls back calling function on execution
* void credits(): Shows the credits and exits the program
* void main(): Initializes graphics and calls various functions to run the program effectively
* int ranc(): Generates a random number and returns it to the calling function
* void end(): Prints a thank you with different colors( by incrementing the color code)

SOURCE CODE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

passw.cpp

For use in our project

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//HEADER FILES

#include<iostream.h>

#include<conio.h>

#include<graphics.h>

#include<stdlib.h>

#include<string.h>

#include<stdio.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int pass()

{

int gdriver = DETECT, gmode;

initgraph(&gdriver,&gmode,"C:\\tc\\bgi");

int i=0,left,top,right,bottom;

setcolor(RED); //background

setfillstyle(1,RED);

rectangle(0,0,getmaxx(),getmaxy());

floodfill(1,1,RED);

setcolor(15); //password box[outer]

rectangle(170,150,470,300);

setfillstyle(9,BLUE);

floodfill(175,180,WHITE); //password box[inner]

rectangle(180,230,460,270);

setfillstyle(1,WHITE);

floodfill(185,235,WHITE);

char pass[20],pass1[20];

int ch;

strcpy(pass,"hell");

settextstyle(4,0,4);

outtextxy(215, 180, "Enter password");

do

{

ch=getch();

if(ch==13)

{

pass1[i]=NULL;

break;

}

if(ch==8)

{

if(i==0)

i++;

setcolor(15);

i--;

outtextxy(215+(15\*(i)),220,"\*");

pass1[i]=NULL;

}

else

{

if(i>12)

{

outtextxy(15,20,"password exceeding word limit");

outtextxy(15,60,"type from beginning");

getch();

setcolor(4);

outtextxy(15,20,"password exceeding word limit");

outtextxy(15,60,"type from beginnin");

while(i>=0)

{

setcolor(15);

outtextxy(215+(15\*(i)),220,"\*");

pass1[i]='\0';

i--;

}

i=0;

}

pass1[i]=ch;

setcolor(1);

outtextxy(215+(15\*i),220,"\*");

i++;

}

}while(ch);

int j=strcmp(pass1,pass);

return j;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Player Management Database System

For use in the project

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//HEADER FILES

#include<fstream.h>

#include<dos.h>

#include<conio.h>

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include<iomanip.h>

#include<graphics.h>

#include<iostream.h>

#include<time.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int gdriver = DETECT, gmode;

int ranc(void) //generating random numbers

{

int i;

time\_t t;

srand((unsigned) time(&t));

i=(rand()%1000);

return i;

}

class team\_player //player class

{

private:

struct person

{

char name[25],nation[25],reput[25],pob[25],lang[100],descp[100],foot[25],flag;

int age,pcode;

}p;

fstream file;

public:

team\_player();

void addrec();

void listrec();

void modirec();

void delrec();

void recallrec();

void end();

void getd();

void pdisp();

};

void team\_player::pdisp() //displaying data

{

closegraph();

clrscr();

initgraph(&gdriver,&gmode,"c:\\tc\\bgi");

cout<<"Player Information";

cout<<"\n\n\n\tCode: "<<p.pcode;

cout<<"\n\n\tName: ";

puts(p.name);

cout<<"\n\tAge: "<<p.age;

cout<<"\n\n\tPlace of birth: ";

puts(p.pob);

cout<<"\n\n\tNation: ";

puts(p.nation);

cout<<"\n\tReputation: ";

puts(p.reput);

cout<<"\n\tDescription: ";

puts(p.descp);

cout<<"\n\tLanguages known: ";

puts(p.lang);

cout<<"\n\tFoot: "<<p.foot;}

void team\_player::getd() //getting data

{

cout<<"PLAYER DETAILS";

cout<<"\n\n\n\n\tEnter Player Name: ";

gets(p.name);

cout<<"\n\tEnter Nation: ";

gets(p.nation);

cout<<"\n\tEnter Reputation: ";

gets(p.reput);

cout<<"\n\tEnter Place of birth: ";

gets(p.pob);

cout<<"\n\tEnter Languages known: ";

gets(p.lang);

cout<<"\n\tEnter Position: ";

gets(p.descp);

cout<<"\n\tEnter Foot(L/R): ";

gets(p.foot);

cout<<"\n\tEnter Age(More than 14): ";

cin>>p.age;

while(p.age<14)

{

cout<<"\n\tEnter Age: ";

cin>>p.age;

}

p.pcode=ranc();

}

team\_player::team\_player() //default constructor

{

}

void team\_player::addrec() //adding records

{

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

char ch;

file.seekp(0L,ios::end);

do

{

team\_player::getd();

file.write((char\*)&p,sizeof(p));

cout<<"Add another record ?(y/n) :";

cin>>ch;

clrscr();

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

file.close();

}

void team\_player::listrec() //displaying records

{

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

int j=0,a;

file.seekg(0L,ios::beg);

cout<<"List of records present are as under>>>"<<endl<<endl;

cout<<" "<<"|CODE|"<<" "<<"|NAME|"<<" "<<"|AGE|"<<" "<<"|POSITION|"<<" "<<"|RATING|"<<endl;

while(file.read((char\*)&p,sizeof(p)))

{

cout<<endl<<"Record#"<<"\t"<<++j<<setw(10)<<p.pcode<<setw(16)<<p.name<<setw(5)<<p.age<<setw(9)<<p.descp<<setw(10)<<p.reput<<endl;

}

file.clear();

if(j==0)

{

gotoxy(10,10);

cout<<"No record exit";

gotoxy(10,13);

cout<<"Press any key...";

getch();

}

else

{

cout<<endl<<"Press any key...";

getch();

}

file.close();

}

void team\_player::modirec() //modifying records

{

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end(); }

int code;

int count=0;

long int pos;

int che=0;

char ch;

fstream f1;

f1.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!f1)

{

cout<<endl<<"Unable to open file";

end();

}

cout<<"Do you want to know the team code for any team?";

cin>>ch;

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

{

file.close();

team\_player::listrec();

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

}

cout<<"\n\nEnter player code whose record is to be modified :";

cin>>code;

file.seekg(0L,ios::beg);

while(file.read((char\*)&p,sizeof(p)))

{

if(p.pcode==code)

{

cout<<endl<<"\n\nEnter new record "<<endl;

team\_player::pdisp();

char name[25],nation[25],reput[25],pob[25],lang[100],descp[100],foot[25];

cout<<"\n\n\tNew name:(Enter '.'to retain old one) ";

gets(name);

cout<<"\n\tNew nation:(Enter '.'to retain old one) ";

gets(nation);

cout<<"\n\tNew reputation:(Enter '.'to retain old one) ";

gets(reput);

cout<<"\n\tNew place of birth:(Enter '.'to retain old one) ";

gets(pob);

cout<<"\n\tNew language:(Enter '.'to retain old one) ";

gets(lang);

cout<<"\n\tNew foot:(Enter '.'to retain old one) ";

gets(foot);

cout<<"\n\tNew position:(Enter '.'to retain old one) ";

gets(descp);

if(strcmp(name,".")!=0)

strcpy(p.name,name);

if(strcmp(nation,".")!=0)

strcpy(p.nation,nation);

if(strcmp(reput,".")!=0)

strcpy(p.reput,reput);

if(strcmp(pob,".")!=0)

strcpy(p.pob,pob);

if(strcmp(lang,".")!=0)

strcpy(p.lang,lang);

if(strcmp(foot,".")!=0)

strcpy(p.foot,foot);

if(strcmp(descp,".")!=0)

strcpy(p.descp,descp);

pos=count\*sizeof(p);

file.seekp(pos,ios::beg);

file.write((char\*)&p,sizeof(p));

break;

} count++;}

if(count==0)

{

cout<<endl<<"\n\nNo player in file with code= "<<code;

getch();

}

file.close();

f1.close();

}

void team\_player::delrec() //deleting records

{

char ch;

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

int code;

long int pos;

char conf,fnd='f';

int che=0;

fstream f1;

f1.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!f1)

{

cout<<endl<<"Unable to open file";

end();

}

cout<<"Do you want to know the team code for any team?";

cin>>ch;

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

{

file.close();

team\_player::listrec();

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

}

fstream file2("temp.dat",ios::binary|ios::in|ios::out);

cout<<"\n\nEnter player's code to be deleted :";

cin>>code;

file.seekg(0L,ios::beg);

file.read((char \*)&p,sizeof(p));

while(!file.eof())

{

if(p.pcode==code)

{

fnd='t';

cout<<"\n\n\tAre you sure you want to delete this record?(y/n): "; cin>>conf;

if(conf=='n')

file2.write((char \*)&p,sizeof(p));

}

else

file2.write((char \*)&p,sizeof(p));

file.read((char \*)&p,sizeof(p));

}

if(fnd=='f')

cout<<endl<<"\n\nNo player in file with code="<<code;

file.close();

file2.close();

remove("team\_player.dat");

rename("temp.dat","team\_player.dat");

cout<<endl<<"\n\nPress any key...";

f1.close();

getch();

}

void team\_player::recallrec() //searching records

{

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

char ch;

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

int code,count=0;

long int pos;

char fnd='f';

int che=0;

fstream f1;

f1.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!f1)

{

cout<<endl<<"Unable to open file";

end();

}

cout<<"Do you want to know the team code for any team?";

cin>>ch;

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

{

file.close();

team\_player::listrec();

file.open("team\_player.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

}

cout<<"\n\nEnter player's code to be searched:";

cin>>code;

file.seekg(0L,ios::beg);

file.read((char \*)&p,sizeof(p));

while(!file.eof())

{

if(p.pcode==code)

{

fnd='t';

pos=count\*sizeof(p);

file.seekp(pos,ios::beg);

team\_player::pdisp();

file.write((char\*)&p,sizeof(p));

}

count++;

file.read((char \*)&p,sizeof(p));

}

if(fnd=='f')

cout<<endl<<"\n\nNo player in the file with code="<<code;

cout<<endl<<"\n\nPress any key...";

getch();

file.close();

f1.close();

}

void team\_player::end() //exit function

{

for(int i=0;i<16;i++)

{

setcolor(i+5);

settextstyle(1, HORIZ\_DIR, 9);

outtextxy(60,120,"THANK YOU");

delay(100);

}

exit(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Team Management Database System

For use in the project

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//HEADER FILES

#include<fstream.h>

#include<dos.h>

#include<conio.h>

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include<iomanip.h>

#include<graphics.h>

#include<iostream.h>

#include<math.h>

#include<time.h>

#include<ctype.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int gd=DETECT,gm;

class soccer //statistics class

{

public:

void setinfo();

void display();

private:

int played;

int won;

int draw;

int lost;

int scored;

int conceded;

int points;

int gd;

};

class team //team class

{

private:

struct teamp

{

char name[25],nation[25],flag,riv[100];

int tcode,rat,year;

soccer s;

}t;

fstream file;

public:

team();

void addrec();

void listrec();

void modirec();

void delrec();

void recallrec();

void end();

void getd();

void tdisp();

};

void soccer::setinfo() //getting stats

{

again:

won=draw=lost=scored=conceded=points=gd=0;

cout<<"\n\n\tEnter the number of matches played: ";

cin>>played;

if(played!=0)

{

cout<<"\n\n\tEnter the number of matches won: ";

cin>>won;

if(won!=played)

{

cout<<"\n\n\tEnter the number of matches drawn: ";

cin>>draw;

cout<<"\n\n\tEnter the number of matches lost: ";

cin>>lost;

if(played>(won+draw+lost)||played<(won+draw+lost))

{

cout<<"\n\n\tNo. of matches played not equal";

goto again;

}

}

cout<<"\n\n\tEnter the number of goals scored: ";

cin>>scored;

cout<<"\n\n\tEnter the number of goals conceded : ";

cin>>conceded;

points=won\*3+draw;

gd=scored-conceded;

}

}

void soccer::display() //stats display

{

cout<<"\n\t\tTeam statistics\n";

cout << "\n\n\tNumber of games played: " << played << endl;

cout << "\n\n\tNumber of games won: " << won << endl;

cout << "\n\n\tNumber of games drawn: " << draw << endl;

cout << "\n\n\tNumber of games lost: " << lost << endl;

cout << "\n\n\tNumber of goals scored: " << scored << endl;

cout << "\n\n\tNumber of goals conceded: " << conceded << endl;

cout << "\n\n\tNumber of points collected: " << points << endl;

cout << "\n\n\tGoal difference: " << gd << endl;

}

void team::tdisp() //displaying data

{

closegraph();

clrscr();

initgraph(&gd,&gm,"c:\\tc\\bgi");

cout<<"\n\t\*\*\*Team Information\*\*\*";

cout<<"\n\n\tName: ";

puts(t.name);

cout<<"\n\n\tNation: ";

puts(t.nation);

cout<<"\n\n\tReputation: "<<t.rat;

cout<<"\n\n\tYear founded: "<<t.year;

cout<<"\n\n\tFierce Rivals: ";

puts(t.riv);

cout<<"\n\nDo you want to see team stats for this season??(y/n)";

char c;

cin>>c;

if(c=='y')

t.s.display();

else

getch();

}

void team::getd() //getting data

{

cout<<"\n\n\n\n\tEnter Team Name: ";

gets(t.name);

t.tcode=ranc();

cout<<"\n\tEnter Nation: ";

gets(t.nation);

agn:

cout<<"\n\tEnter Reputation(0-10): ";

cin>>t.rat;

while(t.rat>10||t.rat<0)

{

cout<<"\n\n\tInvalid entry!! Please enter again";

cout<<"\n\n\tEnter Reputation: ";

cin>>t.rat;

}

cout<<"\n\tEnter Year founded(More than 1800): ";

cin>>t.year;

while(t.year<1800)

{

cout<<"\n\n\tInvalid entry!! Please enter again";

cout<<"\n\n\tEnter Year founded: ";

cin>>t.year;

}

cout<<"\n\tEnter fierce rivals: ";

gets(t.riv);

t.s.soccer::setinfo();

}

team::team() //default constructor

{

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

file.close();

}

void team::addrec() //adding records

{

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

char ch;

file.seekp(0L,ios::end);

cout<<"\n\nTEAM DETAILS"<<endl;

do

{

team::getd();

file.write((char\*)&t,sizeof(t));

cout<<"\n\nAdd another record ?(y/n) :";

cin>>ch;

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

file.close();

}

void team::listrec() //displaying records

{

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

int j=0,a;

file.seekg(0L,ios::beg);

cout<<"\n\nNames of teams present in the database:\n\n"<<endl<<endl;

cout<<" "<<"CODE"<<" "<<"NAME"<<" "<<"COUNTRY"<<"\t"<<"RATING"<<endl;

while(file.read((char\*)&t,sizeof(t)))

{

cout<<endl<<"Record#"<<"\t"<<++j<<setw(7)<<t.tcode<<setw(16)<<t.name<<setw(10)<<t.nation<<setw(107)<<setw(10)<<t.rat<<endl;

}

if(j==0)

{

gotoxy(12,10);

cout<<"\n\nNo record exit";

gotoxy(12,13);

cout<<"\n\nPress any key...";

getch();

}

else

{

cout<<endl<<"Press any key...";

getch();

}

file.close();

}

void team::modirec() //modifying records

{

int che=0;

fstream f1;

f1.open("team1.dat",ios::binary|ios::in|ios::out);

if(!f1)

{

cout<<endl<<"Unable to open file";

end();

}

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

int code;

int count=0;

long int pos;

char ch;

cout<<"Do you want to know the team code for any team?";

cin>>ch;

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

{

file.close();

team::listrec();

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

}

cout<<"\n\nEnter team code whose record is to be modified :";

cin>>code;

file.read((char\*)&t,sizeof(t));

while(!file.eof())

{

if(t.tcode!=code)

{

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

}

else

{

che=1;

cout<<"\n";

team::tdisp();

cout<<"\nEnter the new team details ";

team::getd();

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

file.read((char\*)&t,sizeof(t)); }

if(che==0)

cout<<"\n\nCode number does not exist\n";

remove("team.dat");

rename("team1.dat","team.dat");

cout<<endl<<"Press any key...";

getch();

f1.close();

file.close();

}

void team::delrec() //deleting records

{

int che=0;

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

int code,count=0;

long int pos;

char conf,fnd='f';

char ch;

cout<<"Do you want to know the team code for any team?";

cin>>ch;

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

{

file.close();

team::listrec();

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

}

cout<<"\n\nEnter team code to be deleted :";

cin>>code;

fstream f1;

f1.open("team1.dat",ios::binary|ios::in|ios::out);

if(!f1)

{

cout<<endl<<"Unable to open file";

end();

}

file.read((char\*)&t,sizeof(t));

while(!file.eof())

{

if(t.tcode!=code)

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

else

che=1;

file.read((char\*)&t,sizeof(t));

}

if(che==0)

cout<<"\n\nCode number does not exist\n";

remove("team.dat");

rename("team1.dat","team.dat");

cout<<endl<<"Press any key...";

getch();

f1.close();

file.close();

}

void team::recallrec() //searching records

{

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

int code,count=0;

long int pos;

char fnd='f';

char ch;

cout<<"Do you want to know the team code for any team?";

cin>>ch;

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

{

file.close();

team::listrec();

file.open("team.dat",ios::binary|ios::in|ios::out);

if(!file)

{

cout<<endl<<"Unable to open file";

end();

}

}

cout<<"\n\n\tEnter team code to be recalled :";

cin>>code;

file.seekg(0L,ios::beg);

while(!file.eof())

{

file.read((char \*)&t,sizeof(t));

if(t.tcode==code)

{

fnd='t';

pos=count\*sizeof(t);

file.seekp(pos,ios::beg);

team::tdisp();

file.write((char\*)&t,sizeof(t));

}

count++;

}

if(fnd=='f')

cout<<endl<<"No such team in the file with code="<<code;

cout<<endl<<"Press any key...";

getch();

file.close();

}

void team::end() //exit function

{

for(int i=0;i<16;i++)

{

outtextxy(60,120,"THANK YOU");

delay(100);

}

exit(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SOCCER MANAGER

VERSION 1.0

BY Arvind and Nagabharan

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//HEADER FILES

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <fstream.h>

#include <iostream.h>

#include <bios.h>

#include <stdlib.h>

#include <string.h>

#include <dos.h>

#include "passw.cpp"

#include "plp.cpp"

#include "tmp.cpp"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//GLOBAL DECLARATIONS

FILE \*f;

char str[50];

int ch;

team\_player tmp;

team t;

col[3]={15,4,2};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void clear()

{

closegraph();

clrscr();

initgraph(&gdriver,&gmode,"c:\\tc\\bgi");

}

void end()

{

clear();

for(int i=0;i<16;i++)

{

setcolor(i+5);

settextstyle(1, HORIZ\_DIR, 9);

outtextxy(60,120,"THANK YOU");

delay(100);

}

exit(0);

}

void menu1(int x,int n,int y,int z=200)

{

settextstyle(1,0,4);

char str[50];

if(n==4)

strcpy(str,"MAIN MENU");

else

strcpy(str,"MENU");

outtextxy(x+10,50,str);

settextstyle(1,0,4);

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

{

rectangle(x,y,x+z,y+40);

y=y+40;

}

}

void img(char str[],int x,int y,int z)

{

int p=x;

unsigned char bit;

f=fopen(str,"r");

fseek(f,0x76,1);

while(!feof(f))

{

bit=fgetc(f);

putpixel(x++,y,bit);

putpixel(x++,y,bit);

if (x>z)

{

x=p;

y--;

} }}

void bodr(int p,int q)

{

int g,h,f=0;

for(g=315;g>=5;g-=10)

{

++f;

if(f%2==0)

setfillstyle(SOLID\_FILL,LIGHTGRAY);

else if(f%3==0)

setfillstyle(SOLID\_FILL,BLUE);

else

setfillstyle(SOLID\_FILL,RED);

setcolor(p);

fillellipse(g,5,10,10);

setcolor(q);

fillellipse(640-g,5,10,10);

delay(50);

p=p+q;

q=p-q;

p=p-q;

}

for(h=5;h<=475;h+=10)

{

++f;

if(f%2==0)

setfillstyle(SOLID\_FILL,LIGHTGRAY);

else if(f%3==0)

setfillstyle(SOLID\_FILL,BLUE);

else

setfillstyle(SOLID\_FILL,RED);

setcolor(q);

fillellipse(5,h,10,10);

setcolor(p);

fillellipse(635,h,10,10);

delay(50);

p=p+q;

q=p-q;

p=p-q;

}

for(g=5,h=645;g<=305,h>=325;g+=10,h-=10)

{

++f;

if(f%2==0)

setfillstyle(SOLID\_FILL,LIGHTGRAY);

else if(f%3==0)

setfillstyle(SOLID\_FILL,BLUE);

else

setfillstyle(SOLID\_FILL,RED);

setcolor(q);

fillellipse(g,475,10,10);

setcolor(p);

fillellipse(h,475,10,10);

delay(50);

p=p+q;

q=p-q;

p=p-q;

}

}

void welcome()

{

bodr(7,4);

for(int i=0;!kbhit();i++)

{

setcolor(col[i]);

settextstyle(TRIPLEX\_FONT, HORIZ\_DIR, 9);

outtextxy(100,160,"WELCOME");

setcolor(col[i+1]);

settextstyle(TRIPLEX\_FONT, HORIZ\_DIR, 9);

outtextxy(103,160,"WELCOME");

setcolor(col[i+2]);

settextstyle(TRIPLEX\_FONT, HORIZ\_DIR, 9);

outtextxy(106,160,"WELCOME");

delay(300);

if(i>3)i=0;

}

getch();

}

void options1()

{

outtextxy(360,110,"DATABASE");

outtextxy(360,150,"HELP");

outtextxy(360,190,"CREDITS");

outtextxy(360,230,"EXIT");

outtextxy(60,400,"PRESS [Esc] TO GO EXIT");

}

void options2()

{

outtextxy(60,110,"ADD");

outtextxy(60,150,"MODIFY");

outtextxy(60,190,"DELETE");

outtextxy(60,230,"SEARCH");

outtextxy(60,270,"DISPLAY");

outtextxy(60,310,"EXIT");

outtextxy(60,400,"PRESS [<-] TO GO BACK");

}

void player()

{

int key,x=50,y=80;

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

options2();

while((key=bioskey(0))!=3592)

{

setcolor(0);

options2();

setfillstyle(1,0);

floodfill(x+50,y,WHITE);

setcolor(15);

menu1(50,6,110);

switch(key)

{

case 20480:if(y>=320) y=80;

y+=40;

break;

case 18432:if(y<=120) y=360;

y-=40;

break;

case 7181:switch(y)

{

case 120:cleardevice();

tmp.addrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 160:cleardevice();

tmp.modirec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 200:cleardevice();

tmp.delrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 240:cleardevice();

tmp.recallrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 280:cleardevice();

tmp.listrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 320:end();

break;

}

break;

}

setfillstyle(1,1);

floodfill(x+50,y,WHITE);

options2();

}

clear();

strcpy(str,"field.bmp");

img(str,200,450,511);

menu1(50,3,110,300);

}

void tmdet()

{

int key,x=50,y=80;

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

options2();

while((key=bioskey(0))!=3592)

{

setcolor(0);

options2();

setfillstyle(1,0);

floodfill(x+50,y,WHITE);

setcolor(15);

menu1(50,6,110);

switch(key)

{

case 20480:if(y>=320) y=80;

y+=40;

break;

case 18432:if(y<=120) y=360;

y-=40;

break;

case 7181:switch(y)

{

case 120:cleardevice();

t.addrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 160:cleardevice();

t.modirec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 200:cleardevice();

t.delrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 240:cleardevice();

t.recallrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 280:cleardevice();

t.listrec();

clear();

strcpy(str,"imagef.bmp");

img(str,300,300,530);

menu1(50,6,110);

break;

case 320:end();

break;

}

break;

}

setfillstyle(1,1);

floodfill(x+50,y,WHITE);

options2();

}

clear();

strcpy(str,"field.bmp");

img(str,200,450,511);

menu1(50,3,110,300);

}

void options3()

{

outtextxy(60,110,"TEAM DETAILS");

outtextxy(60,150,"PLAYER DETAILS");

outtextxy(60,190,"EXIT");

outtextxy(60,10,"PRESS [<-] TO GO BACK");}

void database()

{

int key,x=50,y=80;

strcpy(str,"field.bmp");

img(str,200,450,511);

menu1(50,3,110,300);

options3();

while((key=bioskey(0))!=3592)

{

setcolor(0);

options3();

setfillstyle(1,0);

floodfill(x+50,y,WHITE);

setcolor(15);

menu1(50,3,110,300);

switch(key)

{

case 20480:if(y>=200) y=80;

y+=40;

break;

case 18432:if(y<=120) y=240;

y-=40;

break;

case 7181:switch(y)

{

case 120:cleardevice();

tmdet();

break;

case 160:cleardevice();

player();

break;

case 200:end();

break;

}

break;

}

setfillstyle(1,1);

floodfill(x+50,y,WHITE);

options3();

}

clear();

strcpy(str,"imag21.bmp");

img(str,20,300,323);

menu1(350,4,110);

}

void help()

{

int chh;

s:

clear();

setcolor(14);

settextstyle(5, HORIZ\_DIR,6);

outtextxy(175,40,"HELP");

cout<<"\n\n\n\n\n\n\n\n";

cout<<"\n Type the respective numbers to take you to the screen";

cout<<"\n of your choice.The screens have information and options based on";

cout<<"\n the selection made.";

cout<<"\n Your suggestions and ideas will be of great assistance";

cout<<"\n for us to do our project better in future.";

cout<<"\nIn the main menu ";

cout<<"\nPress 1st option to database";

cout<<"\nPress 2nd option to get help";

cout<<"\nPress 3rd option to view credits";

cout<<"\nPress 4th option to exit";

cout<<"\nIn database menu: ";

cout<<"\nPress 1st option to go to player database";

cout<<"\nPress 2nd option to go to team database";

cout<<"\nIn each database: ";

cout<<"\nPress 1st option to enter details";

cout<<"\nPress 2nd option to list the records";

cout<<"\nPress 3rd option to modify details";

cout<<"\nPress 4th option to delete records";

cout<<"\nPress 5th option to search and view details";

cout<<"\nPress 6th option to exit from the program";

cout<<"\nPress backspace key to navigate back to previous menu";

getch();

d:

cout<<"\n\n\n\n\n\n\n\n\n\n\n Do you want to do again??(0-exit/1-continue)";

chh=bioskey(0);

if(chh==20273||chh==561)

goto s;

else if(chh==21040||chh==2864)

{

clear();

strcpy(str,"imag21.bmp");

img(str,20,300,323);

menu1(350,4,110);

}

else

goto d;

}

void credits()

for(int z=480;!kbhit();z--)

{

if(z==-500)z=480;

cleardevice();

setcolor(4);

settextstyle(3,0,3);

outtextxy(100,z-10,"~~~~~~~~COMPUTER PROJECT~~~");

outtextxy(100,z+10,"~~~~~~~~BY~~~~~~~~");

outtextxy(200,z+30,"\*NAGABHARAN N");

outtextxy(200,z+60,"\*ARVIND P");

outtextxy(100,z+90,"~~~~~~~~DATABASE DEVELOPER~~");

outtextxy(200,z+120,"\*NAGABHARAN N");

outtextxy(100,z+150,"~~~~~~~~GRAPHICS DEVELOPER~~");

outtextxy(200,z+180,"\*ARVIND P");

outtextxy(100,z+210,"~~~~~~~~SUPPORT~~~~~~~~~~~");

outtextxy(200,z+240,"\*KAVITHA MA'AM");

outtextxy(200,z+270,"\*FRIENDS");

outtextxy(100,z+300,"~~~~~~~~THANK YOU!!!~~~~~~~~");

delay(10);

}

getch();

}

void main()

{

int key,x=350,y=80,j,p;

initgraph(&gdriver,&gmode,"c:\\tc\\bgi");

welcome();

a:

j=pass();

if(j==0)

{

setcolor(9);

rectangle(180,180,460,270);

setfillstyle(SOLID\_FILL,RED);

floodfill(185,235,9);

setcolor(15);

settextstyle(4,0,4);

outtextxy(227,200,"\naccess granted");

getch();

cleardevice();

strcpy(str,"imag21.bmp");

img(str,20,300,323);

menu1(350,4,110);

options1();

while((key=bioskey(0))!=0)

{

setcolor(0);

options1();

setfillstyle(1,0);

floodfill(x+50,y,WHITE);

setcolor(15);

menu1(350,4,110);

switch(key)

{

case 20480:if(y>=240) y=80;

y+=40;

break;

case 18432:if(y<=120) y=280;

y-=40;

break;

case 7181:switch(y)

{

case 120:clearviewport();

database();

break;

case 160:cleardevice();

help();

break;

case 200:cleardevice();

credits() ;

clear();

strcpy(str,"imag21.bmp");

img(str,20,300,323);

menu1(350,4,110);

break;

case 240:end();

break;

}

break;

case 283:end();

}

setfillstyle(1,1);

floodfill(x+50,y,WHITE);

options1();

} }

else

{

setcolor(9);

rectangle(180,180,460,270);

setfillstyle(SOLID\_FILL,RED);

floodfill(185,235,9);

setcolor(15);

settextstyle(4,0,4);

outtextxy(215,200,"\ninvalid password");

getch();

e:

cleardevice();

setcolor(9);

rectangle(100,180,550,270);

setfillstyle(SOLID\_FILL,RED);

floodfill(185,235,9);

setcolor(15);

settextstyle(4,0,4);

outtextxy(120,200,"\npress 1 to re-try,0 to exit");

p=bioskey(0);

if(p==20273||p==561) goto a;

else if(p==21040||p==2864) end();

else goto e;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

OUTPUT SCREENSHOTS



Welcome screen



Password check



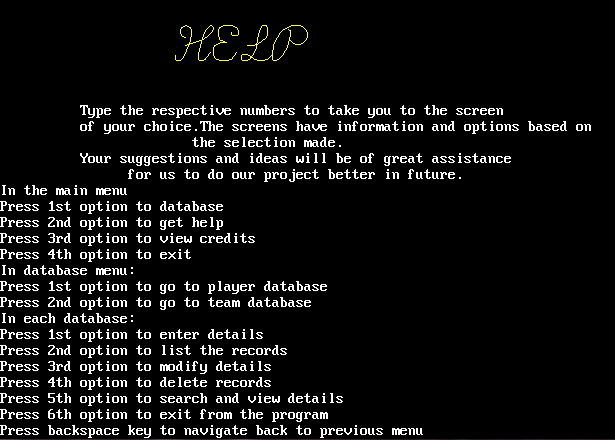
Main menu



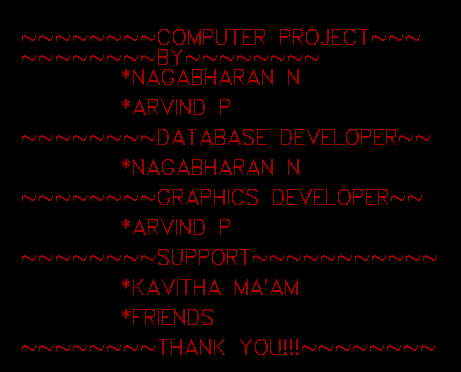
Sub-menu for databases



Database management menu



Help screen



Credits screen



Thank you screen

SCOPE FOR IMPROVEMENT

We, the developers, tried hard to integrate whatever we wished into Soccer Manager to make it as interactive, user-friendly and useful Software rather than just a project. However, we encountered a few setbacks during the development stages of Soccer Manager which propelled us into modifying or altogether removing some of the features.

The next feature we had to remove was a processor speed evaluation mechanism because of difference in CPU time allocated to Turbo C++ and standalone MS-DOS applications. At any point of time, Turbo C++ is allowed 100% CPU, so the processor evaluates to much higher speeds of the programs is accessed from within the compiler. Due to this discrepancy, we removed the speed test function which would have helped us to make the program much faster and efficient.

We wish to add more features to this project by interlinking the databases to reduce data redundancy, inconsistency and to facilitate sharing. There are a few standards introduced to maintain integrity of data. We can improve this project by introducing a mini-game, which utilizes all the data present in database and makes the project more user-friendly. We can also introduce various charts for compare and contrast of different players and teams.

BIBLIOGRAPHY

We have obtained material information and a lot of knowledge from these sources:

Computer Science: A Textbook for Class XII (C++) by Sumita Arora

*For helping us grasp the basics of C++.*

The Google® Search Engine (www.google.com)

*For bits of critical information at times when we needed it the most.*

Source Codes World (www.sourcecodesworld.com)

*For providing us with examples of source code to help us understand C++ better.*

C++ Resources Network (www.cplusplus.com)

*For C++ Documentation which we could not have found elsewhere.*

C++: The Complete Reference (4th Edition) by Herbert Scheldt

*For helping us understand the toughest concepts of C++.*