Computer Science Project

Sportify

2014-15

Saurabh Chheda

Hiranmaya Gundu

Sri Kumaran Children’s Home CBSE

Acknowledgements

Firstly, we would like to thank CBSE for giving us the opportunity to make a computer project as part of the CBSE 12th Std Portion.

We would also like to thank our school, Sri Kumaran Children’s Home for providing us with the faculty to make this project. We would also like to thank the school for providing us with the necessary guidance and education required to make this project.

We would also like to thank our Computer Science teachers, Kavitha Ma’am and Smitha Ma’am in their continued support, guidance and invaluable advice without which our project would not have been possible to complete.

Last but not the least, we would like to thank our friends and parents who gave their support, moral guidance and help.

Thank You.

Index

### Introduction

### Class Diagram

### Source Code

### Screen Shots

### Scope for Improvement

### Bibliography

### Webliography

Introduction

# Synopsis

Our program is an application that access the internet to use an API to download an XML file containing data and display the data to the user. In this case, our project downloads the schedule of the NBA League, and the EU Football leagues and displays it to the user.

# Reason for Project

The reasons for our project are threefold. Firstly, it provides us with a resource to access the schedule of upcoming EU Football matches, and NBA matches in one location. Secondly, on a more personal note, it opens up the world of real world application software development to us, and thirdly, it opens up new concepts inn coding, such as the REST SDK, which was used in accessing the internet and an API, and also parsing of XML files, using a library called TinyXml 2.

# System Requirements

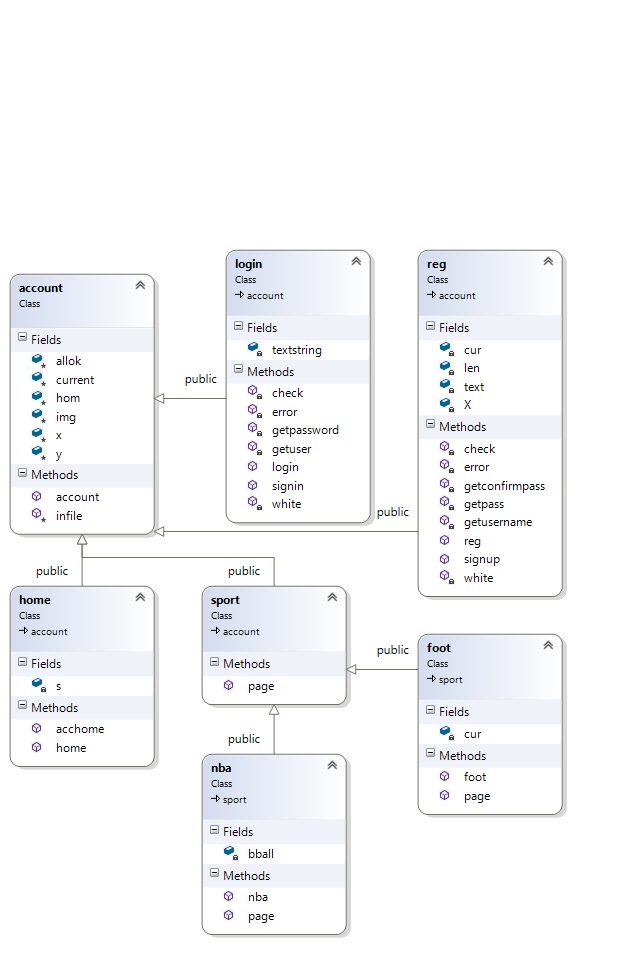
## Hardware

* Minimum 70 MB RAM
* Intel Pentium 2 equivalent or higher
* Internet Connection

## Software

* Windows OS 95 or higher
* C++ Rest SDK Casablanca

Class Diagram



# Function Description

## Class Account

### Infile

Writes data into users.dat

## Class Login

### Signin

Displays the background and handles clicks

### Getuser

Inputs username

### Getpassword

Inputs password

### Check

Authenticates Login

### Error

Checks for login errors

### White

Checks for white spaces

## Class Reg

### Signup

Displays the background and handles clicks

### Getusername

Inputs username

### Getpass

Inputs password

### Getconfirmpass

Confirms password

### Check

Authenticates Login

### Error

Checks for login errors

### White

Checks for white spaces

## Class Home

### Acchome

Displays the background and handles clicks

## Class Sport

### Page

Virtual function

## Class Foot

### Page

Displays the background and handles clicks

### Dlsoc

Downloads result.xml

### Xmlsoc

Reads and displays result.xml

### Sdatetime

Gets current time

## Class nba

### Page

Displays the background and handles clicks

### Dlnba

Downloads resultNBA.xml

### Xmlnba

Reads and displays resultNBA.xml

### Ndatetime

Gets current time

Source Code

# Source.cpp

#include "login.h"

#include "register.h"

int account::hom = -6;

int main()

{

int x, y;

bool e = 0; //Exit

login l;

reg r;

initwindow(913, 682, "Welcome");

readimagefile("Images\\Pages\\Account\\splash.jpg", 0, 0, getmaxx(), getmaxy()); //Background

//clicks

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 388 && y >= 348 && x <= 545 && y <= 408)

r.signup(); //register page

else if (x >= 390 && y >= 251 && x <= 534 && y <= 314)

l.signin(); //login page

else if (x >= 440 && y >= 440 && x <= 490 && y <= 470)

e = 1;

if (!e)

main();

else

closegraph();

}

# Account\_det.h

#include<fstream>

#include<string.h>

#include"graphics.h"

#ifndef ACCOUNT\_DET\_H

#define ACCOUNT\_DET\_H

struct detail //Account Details

{

char username[20], password[20];

};

class account

{

protected:

int x, y;

static int hom;

char img[40]; //Background Image

detail current; //current account

int allok; //login successful/unsuccessful

/\*----------------------------

allok references

+2 - password wrong

+1 - username does not exist

0 - everything ok

-1 - username taken

-2 - passwords do not match

-------------------------------\*/

void infile() //to input into file

{

std::ofstream f("Data//users.dat", std::ios::binary | std::ios::app);

//f.seekg(0, std::ios\_base::end);

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

f.close();

}

public:

account()

{

x = y = -1;

img[0] = current.username[0] = current.password[0] = '\0';

allok = 3;

}

};

#endif

# Register.h

#include"account\_det.h" //base class

#ifndef REGISTER\_H

#define REGISTER\_H

class reg : public account

{

int len, cur; //string length, current window

char X[20], text[40]; //confirm pass

void getusername(); //username input

void getpass(); //password input

void getconfirmpass(); //confirm password

void check(); //check for password match

bool white(char); //to check for white spaces

void error(); //to check which error

public:

void signup(); //to signup

reg()

{

len = 0;

cur = -6;

text[0] = X[0] = '\0';

strcpy\_s(img, "Images\\Pages\\Account\\register.jpg"); //background

allok = 1;

}

};

void reg::signup()

{

cur = initwindow(912, 684, "Register"); //new window opens

readimagefile(img, 0, 0, getmaxx(), getmaxy());

while (allok)

{

//clicks

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 334 && y >= 291 && x <= 628 && y <= 331)

getusername(); //input username

else if (x >= 334 && y >= 373 && x <= 628 && y <= 414)

getpass(); //input password

else if (x >= 334 && y >= 457 && x <= 628 && y <= 499)

getconfirmpass(); //Confirmation of Password

else if (x >= 410 && y >= 535 && x <= 523 && y <= 568)

check(); //check for availiblity

else if (x >= 435 && y >= 595 && x <= 493 && y <= 670)

{

closegraph(); //control returned to main function

allok = 0;

}

}

}

void reg::getusername()

{

int i = 0, ch = 0;

//Text styling

setusercharsize(1, 2, 1, 2);

settextstyle(4, HORIZ\_DIR, 0);

//Text onto graphic window

while (1)

{

ch = getch();

if (ch == 8) //Backspace

{

i--;

text[i] = '\0';

}

else if (white(ch))

break;

else //Accept Text

{

text[i] = ch;

++i;

text[i] = '\0';

}

cleardevice();

readimagefile(img, 0, 0, getmaxx(), getmaxy());

outtextxy(341, 298, text);

}

strcpy\_s(current.username, text);

}

void reg::getpass()

{

int i = 0, ch = 0;

//text styling

setusercharsize(1, 2, 1, 2);

settextstyle(4, HORIZ\_DIR, 0);

//Graphic response

while (1)

{

ch = getch();

if (ch == 8) //Backspace

{

i--;

text[i] = '\0';

}

else if (white(ch))

break;

else //Accept Text

{

text[i] = ch;

++i;

text[i] = '\0';

}

cleardevice();

readimagefile(img, 0, 0, getmaxx(), getmaxy());

outtextxy(341, 298, current.username);

setfillstyle(SOLID\_FILL, WHITE);

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

{

circle(350 + (20 \* j), 390, 8);

floodfill(350 + (20 \* j), 390, WHITE);

}

}

len = i;

strcpy\_s(current.password, text);

}

void reg::getconfirmpass()

{

int i = 0, ch = 0;

//Text Styling

setusercharsize(1, 2, 1, 2);

settextstyle(4, HORIZ\_DIR, 0);

//Graphic Response

while (1)

{

ch = getch();

if (ch == 8) //Backspace

{

i--;

text[i] = '\0';

}

else if (white(ch))

break;

else //Accept Text

{

text[i] = ch;

++i;

text[i] = '\0';

}

cleardevice();

readimagefile(img, 0, 0, getmaxx(), getmaxy());

outtextxy(341, 298, current.username);

setfillstyle(SOLID\_FILL, WHITE);

for (int j = 0; j < len; ++j)

{

circle(350 + (20 \* j), 390, 8);

floodfill(350 + (20 \* j), 390, WHITE);

}

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

{

circle(350 + (20 \* j), 480, 8);

floodfill(350 + (20 \* j), 480, WHITE);

}

}

strcpy\_s(X, text);

}

void reg::check()

{

allok = 0; //Assuming everything is alright

std::fstream f("users.dat", std::ios::in | std::ios::binary);

detail acc;

while (f)

{

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

if (!strcmp(current.username, acc.username)) //username taken

allok = -1;

else if (strcmp(current.password, X)) //passwords don't match

allok = -2;

}

f.close();

error();

allok = 0; //to break out of while loop

}

bool reg::white(char x)

{

if (x == 0 || x == 7 || x == 8 || x == 9 || x == 10 || x == 13 || x == 32 || x == 255)

return 1;

else

return 0;

}

void reg::error()

{

switch (allok)

{

case -1://username taken

initwindow(911, 683, "Error!");

strcpy\_s(img, "Images\\Pages\\Account\\usertaken.jpg");

readimagefile(img, 0, 0, getmaxx(), getmaxy());

break;

case -2://passwords dont match

initwindow(913, 678, "Error!");

strcpy\_s(img, "Images\\Pages\\Account\\passnomatch.jpg");

readimagefile(img, 0, 0, getmaxx(), getmaxy());

break;

case 0://Registered!

infile();

initwindow(906, 682, "Registered!");

strcpy\_s(img, "Images\\Pages\\Account\\registerd.jpg");

readimagefile(img, 0, 0, getmaxx(), getmaxy());

}

//clicks

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 435 && y >= 595 && x <= 493 && y <= 670) //back

closegraph();

}

#endif

# Login.h

#include"home.h"

#include"register.h"

#ifndef LOGIN\_H

#define LOGIN\_H

class login : public account

{

char textstring[20];

void check(); //to check for login

void getuser(); //inpput username

void getpassword(); //input password

bool white(char); //to check for white spaces

void error(); //to check which error

public:

void signin(); //graphic window

login()

{

textstring[0] = '\0';

strcpy\_s(img, "Images\\Pages\\Account\\login.jpg"); //background

allok = 1;

}

};

void login::signin()

{

initwindow(914, 683, "Signin"); //new window opens

readimagefile(img, 0, 0, getmaxx(), getmaxy());

while (allok) //to check for signin

{

//clicks

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 345 && y >= 360 && x <= 640 && y <= 400)

getuser(); //username

else if (x >= 345 && y >= 460 && x <= 640 && y <= 500)

getpassword(); //password

else if (x >= 420 && y >= 520 && x <= 510 && y <= 550)

check(); //sign in

else if (x >= 430 && y >= 600 && x <= 490 && y <= 670)

{

closegraph(); //back

allok = 0;

}

}

}

void login::getuser()

{

int i = 0, ch = 0;

//Text Styling

setusercharsize(1, 2, 1, 2);

settextstyle(4, HORIZ\_DIR, 0);

//text onto graphic window

while (1)

{

ch = getch();

if (ch == 8) //Backspace

{

i--;

textstring[i] = '\0';

}

else if (white(ch))

break;

else //Accept Text

{

textstring[i] = ch;

++i;

textstring[i] = '\0';

}

cleardevice();

readimagefile(img, 0, 0, getmaxx(), getmaxy());

outtextxy(355, 370, textstring);

}

strcpy\_s(current.username, textstring);

}

void login::getpassword()

{

int i = 0, ch = 0;

//text styling

setusercharsize(1, 2, 1, 2);

settextstyle(4, HORIZ\_DIR, 0);

//Graphic response

while (1)

{

ch = getch();

if (ch == 8) //Backspace

{

i--;

textstring[i] = '\0';

}

else if (white(ch))

break;

else //Accept Text

{

textstring[i] = ch;

++i;

textstring[i] = '\0';

}

cleardevice();

readimagefile(img, 0, 0, getmaxx(), getmaxy());

outtextxy(355, 370, current.username);

setfillstyle(SOLID\_FILL, WHITE);

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

{

circle(365 + (20 \* j), 485, 8);

floodfill(360 + (20 \* j), 485, WHITE);

}

}

strcpy\_s(current.password, textstring);

}

void login::check()

{

bool chk = 0; //1 if login success

detail temp;

std::fstream f("users.dat", std::ios::in | std::ios::binary);

while (f.read((char\*)&temp, sizeof(detail)) && !chk)

{

if (!strcmp(temp.username, current.username)) //if username exists

{

if (!strcmp(temp.password, current.password)) //passwords match

{

allok = 0;

chk = 1;

}

else

allok = 2;

}

else

allok = 1;

}

f.close();

error();

allok = 0; //to break while loop

}

bool login::white(char x)

{

if (x == 0 || x == 7 || x == 8 || x == 9 || x == 10 || x == 13 || x == 32 || x == 255)

return 1;

else

return 0;

}

void login::error()

{

home h;

reg r1;

switch (allok)

{

case 0://login successful

closegraph(CURRENT\_WINDOW);

h.acchome();

break;

case 1://username does not exist

initwindow(912, 682, "Error!");

strcpy\_s(img, "Images\\Pages\\Account\\usernoexist.jpg");

readimagefile(img, 0, 0, getmaxx(), getmaxy());

//clicks

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 435 && y >= 595 && x <= 493 && y <= 670) //back

closegraph();

else if (x >= 400 && y >= 460 && x <= 525 && y <= 505) //signup

r1.signup();

break;

case 2://password is wrong

initwindow(913, 678, "Error!");

strcpy\_s(img, "Images\\Pages\\Account\\passnomatch.jpg");

readimagefile(img, 0, 0, getmaxx(), getmaxy());

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 435 && y >= 595 && x <= 493 && y <= 670) //back

closegraph();

break;

}

}

#endif

# Sport.h

#include"account\_det.h"

#ifndef SPORT\_H

#define SPORT\_H

class sport : public account

{

public:

virtual void page(){}

};

#endif

# Home.h

#include"football.h"

#include"nba.h"

#ifndef HOME\_H

#define HOME\_H

class home : public account

{

sport \*s;

public:

home()

{

s = NULL;

strcpy\_s(img, "Images\\Pages\\Account\\Home.jpg");

}

void acchome();

};

void home::acchome()

{

hom = initwindow(910, 683, "Home");

readimagefile(img, 0, 0, getmaxx(), getmaxy());

while (1)

{

s = NULL;

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 404 && y >= 237 && x <= 527 && y <= 272) //football

s = new foot;

else if (x >= 391 && y >= 323 && x <= 539 && y <= 356) //baseball

s = new nba;

if (s)

s->page();

else if (x >= 396 && y >= 407 && x <= 536 && y <= 446) //sign out

break;

}

closegraph(ALL\_WINDOWS);

}

#endif

# Football.h

#include"sport.h"

#include"dlsoccer.h"

#include"readsoccer.h"

#ifndef FOOTBALL\_H

#define FOOTBALL\_H

class foot : public sport

{

int cur;

public:

void page();

foot()

{

strcpy\_s(img, "Images//Pages//Football//Home.jpg");

cur = initwindow(911, 684, "Football");

readimagefile(img, 0, 0, getmaxx(), getmaxy());

x = y = -1;

}

};

void foot::page()

{

dlsoc();

setcurrentwindow(cur);

xmlsoc();

while (!ismouseclick(WM\_LBUTTONUP));

getmouseclick(WM\_LBUTTONUP, x, y);

clearmouseclick(WM\_LBUTTONUP);

if (x >= 436 && y >= 596 && x <= 491 && y <= 668) //back

{

closegraph(CURRENT\_WINDOW);

setcurrentwindow(hom);

}

}

# Dlsoccer.h

#include <cpprest/http\_client.h>

#include <cpprest/filestream.h>

#include<iostream>

#include<string>

#include"graphics.h"

//using namespace std;

using namespace utility; // Common utilities like string conversions

using namespace web; // Common features like URIs.

using namespace web::http; // Common HTTP functionality

using namespace web::http::client; // HTTP client features

using namespace concurrency::streams; // Asynchronous streams

#ifndef DLSOCCER\_H

#define DLSOCCER\_H

int dlsoc()

{

initwindow(912, 681, "Football");

readimagefile("Images//Pages//Football//dlfoot.jpg", 0, 0, getmaxx(), getmaxy());

char \* url = "http://api.sportsdatallc.org/soccer-t2/eu/matches/2014/03/01/summary.xml?api\_key=m665dmrqynqn4t9r4vdxp2c7";

std::string s = "http://api.sportsdatallc.org/soccer-t2/eu/matches/2014/03/01/summary.xml?api\_key=m665dmrqynqn4t9r4vdxp2c7";

//auto fileStream = std::make\_shared<ostream>();

// Open a stream to the file to write the HTTP response body into.

auto fileBuffer = std::make\_shared<concurrency::streams::streambuf<uint8\_t>>();

return file\_buffer<uint8\_t>::open(L"result.xml", std::ios::out).then([=](concurrency::streams::streambuf<uint8\_t> outFile) -> pplx::task<http\_response>

{

\*fileBuffer = outFile;

// Create an HTTP request.

http\_client client(U("http://api.sportsdatallc.org/soccer-t2/eu/matches/schedule.xml?api\_key=m665dmrqynqn4t9r4vdxp2c7") );

return client.request(methods::GET);

})

// Write the response body into the file buffer.

.then([=](http\_response response) -> pplx::task<size\_t>

{

std::cout<<"Response status code %u returned.\n"<< response.status\_code();

return response.body().read\_to\_end(\*fileBuffer);

})

// Close the file buffer.

.then([=](size\_t datasize)

{

std::cout << datasize;

closegraph(CURRENT\_WINDOW);

return fileBuffer->close();

}).wait();

return 0;

}

#endif

# Readsoccer.h

#if defined( \_MSC\_VER )

#if !defined( \_CRT\_SECURE\_NO\_WARNINGS )

#define \_CRT\_SECURE\_NO\_WARNINGS

#endif

#endif

#include "graphics.h"

#include "tinyxml2.h"

#include <cstdlib>

#include <cstring>

#include <ctime>

using namespace tinyxml2;

#include<iostream>

using namespace std;

#ifndef READSOCCER\_H

#define READSOCCER\_H

void sdatetime(char buffer[25])

{

time\_t rawtime;

struct tm \* timeinfo;

time(&rawtime);

timeinfo = localtime(&rawtime);

strftime(buffer, 80, "%Y-%m-%dT%I:%M:%SZ", timeinfo);

}

void xmlsoc()

{

int i = 0;

char buf[25];

sdatetime(buf);

tinyxml2::XMLDocument doc;

doc.LoadFile("result.xml");

int check;

check = doc.ErrorID();

if (check != 0)

std::cout << "Document load failed.";

XMLNode \* pRoot = doc.FirstChild();

if (pRoot == NULL)

cout << "ERROR";

XMLElement \* pMatchesElement = doc.FirstChildElement();

if (pMatchesElement == NULL)

cout << "ERROR2";

XMLElement \* pIDKElement = pMatchesElement->FirstChildElement();

if (pIDKElement == NULL)

cout << "ERROR3";

XMLElement \* pMatchElement = pIDKElement->FirstChildElement("match");

XMLElement \* pHomeElement = NULL;

XMLElement \* pAwayElement = NULL;

XMLElement \* pVenueElement = NULL;

XMLElement \* pCategoryElement = NULL;

while (pMatchElement != NULL && i < 10)

{

//const char\* status = pMatchElement->Attribute("status");

const char\* schedule = pMatchElement->Attribute("scheduled");

if (/\*!strcmp(status, "scheduled") && \*/strcmp(schedule, buf) == 1)

{

pHomeElement = pMatchElement->FirstChildElement("home");

pAwayElement = pMatchElement->FirstChildElement("away");

pVenueElement = pMatchElement->FirstChildElement("venue");

pCategoryElement = pMatchElement->FirstChildElement("category");

const char\* text = pHomeElement->Attribute("alias");

const char\* text1 = pAwayElement->Attribute("alias");

const char\* venue = pVenueElement->Attribute("city");

const char\* category = pCategoryElement->Attribute("name");

//Text Styling

setusercharsize(1, 2, 1, 2);

settextstyle(4, HORIZ\_DIR, 0);

setbkcolor(COLOR(0, 144, 0));

outtextxy(33, 233 + 37 \* i, (char\*)text);

outtextxy(125, 233 + 37 \* i, (char\*)text1);

outtextxy(266, 233 + 37 \* i, (char\*)venue);

outtextxy(469, 233 + 37 \* i, (char\*)schedule);

outtextxy(727, 233 + 37 \* i, (char\*)category);

i++;

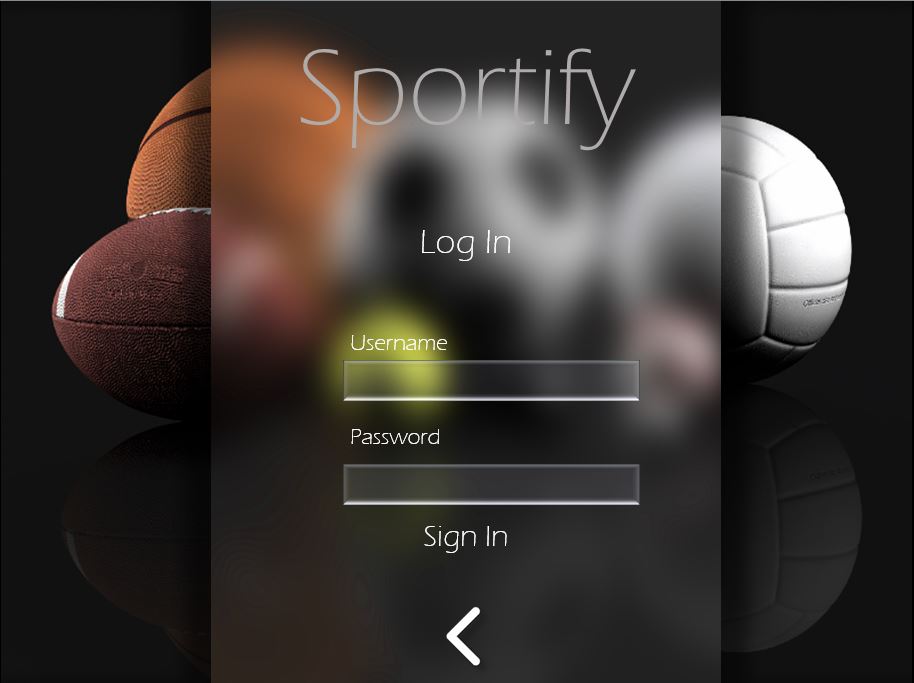
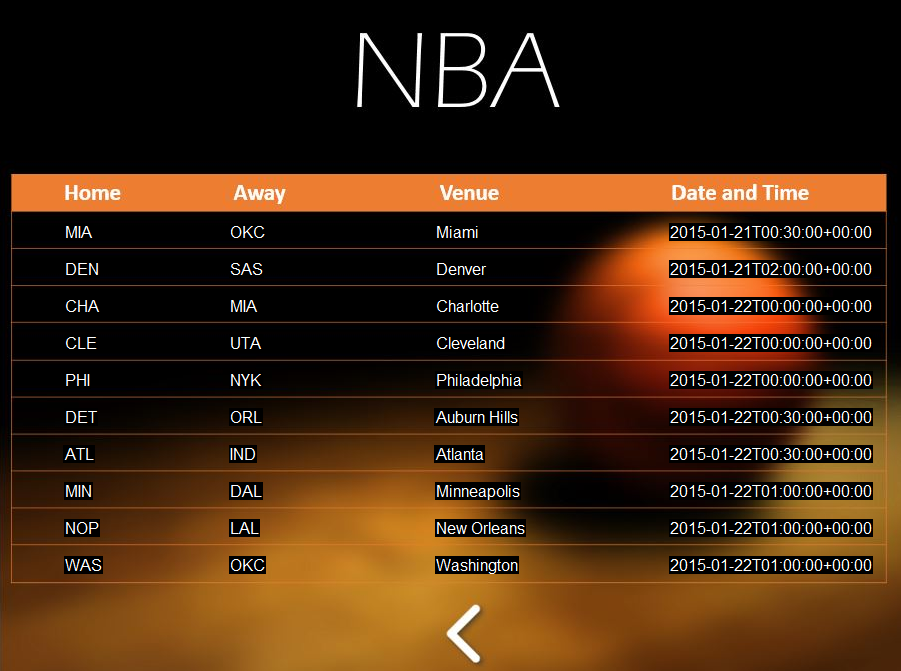
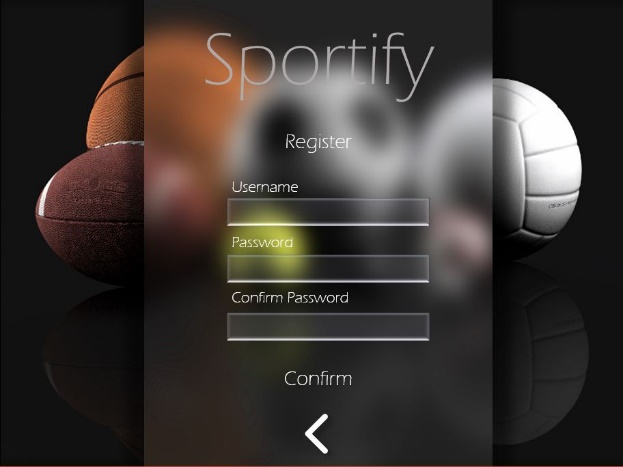
}

pMatchElement = pMatchElement->NextSiblingElement("match");

}

}

#endif

Screenshots

Scope for Improvement

In any Computer Software, there is always a scope for improvement. Our application has the possibilities of doing much more. For example, we can expand our program to many more sports, and not only that we can add the Standings of the team in table.

Personalization is another concept. A user could pick their favorite team when they first make an account. From then on, the schedule and standings of that team could be shown first, and then options to view the schedule of all the games as a whole. Much more can be done on these lines.

Bibliography

Computer Science textbooks by Sumita Arora

Webliography

TinyXml 2

<https://shilohjames.wordpress.com/2014/04/27/tinyxml2-tutorial/>

<http://www.grinninglizard.com/tinyxml2docs/>

Rest SDK

   -<https://casablanca.codeplex.com/>

   - <https://casablanca.codeplex.com/documentation/>

graphics.h library

<http://www.cs.colorado.edu/~main/bgi/doc/>