**Дніпропетровський ліцей інформаційних технологій**

**при Дніпропетровському національному університеті**

**імені Олеся Гончара**

Випускна робота

На тему:

«The Helper For The US Student»

**Виконавець:**

**ліцеїстка 11-В класу**

**Диренкова Емілія**

**Наукові керівники:**

**Зуєва М.Г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Бондік І.Г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Дніпро

2017

Зміст

[Вступ 3](#_Toc471943723)

[Основна частина 5](#_Toc471943724)

[Теоретична частина з предмету 5](#_Toc471943725)

[The U.S. Educational System 5](#_Toc471943726)

[Living On Campus in The United States 6](#_Toc471943727)

[Теоретична частина з інформатики 8](#_Toc471943728)

[Підключення бази даних 8](#_Toc471943729)

[Анімація в Blend for Visual Studio 2015 8](#_Toc471943730)

[Динамічний список кнопок на формі «MY. Useful Info» 10](#_Toc471943731)

[Створення розкладу 12](#_Toc471943732)

[Опис роботи 16](#_Toc471943733)

[Структура програми 21](#_Toc471943734)

[Напрямки, де можна використати роботу 21](#_Toc471943735)

[Програмно-апаратні вимоги 21](#_Toc471943736)

[Комплектація програми 21](#_Toc471943737)

[Використані програмні засоби 22](#_Toc471943738)

[Висновки 23](#_Toc471943739)

[Список використаних джерел 24](#_Toc471943740)

Вступ

Навчаючись в 11 класі, учні починають серйозно замислюватися над своїм майбутнім: ким вони хочуть бути, де хочуть жити і, звичайно, до якого вищого навчального закладу вступати. Багато з учнів 11 класів мріють або планують вчитися в одному з університетів Сполучених Штатів Америки. Це не дивує, адже США – чи не найвпливовіша країна у світі, одна з найрозвиненіших у сфері інформаційних технологій. У більшості рейтингів вищих навчальних закладів світу університети США займають майже всю першу десятку. Сполучені Штати Америки – країна вільних людей, великих можливостей. Найбільші й найвідоміші IT-компанії виникли саме в США: Microsoft Corporation, Google, Facebook.

Однак, переїжджаючи до іншої країни, ми зустрічаємо набагато більше проблем, аніж просто недостатнє знання мови. Різні дрібниці, що окремо можуть здатися зовсім не важливими, у великій кількості можуть просто збити з ніг. Особливо при плануванні студентського, самостійного життя за кордоном виникає багато питань, відповіді на які неможливо знайти на офіційному сайті університету, адже для людей, що виросли там, це здається настільки очевидним, що не потребує роз’яснення. Наприклад, величезна відмінність наших систем освіти – складання розкладу. Багато підлітків в Україні, на мою думку, мають проблеми з організацією власного часу, адже, на відміну від американських учнів, які зі старшої школи складають свій розклад самостійно, наші учні і навіть студенти мають фіксований розклад і групи, що з одного боку полегшує життя, з іншого боку, деколи ускладнює. Особисто мені, американська система освіти імпонує, тому що я люблю та вмію планувати свій час. Та ця відмінність життєвого стилю може ускладнити інтеграцію студентів з України в університетах США.

На основі цих роздумів у мене виникла ідея створити програму, що допомагає учням вищих навчальних закладів США складати свій власний розклад. Також, ця програма повинна мати багато необхідної інформації для швидкого доступу, наприклад адреси магазинів, бібліотек, їдалень на території кампусу тощо. Користь програми полягає в тому, що вона узагальнює життєво необхідні дані в одному місці.

Тож, **метою** моєї випускної роботи стало: створити програмний продукт мовою C# для студентів американських вищих навчальних закладів. Виходячи з цього, були сформульовані **задачі**, що потребують вирішення для виконання мети:

* Дослідити систему навчання вищих навчальних закладів США та зібрати необхідні дані.
* Структурувати матеріал.
* Розробити схему програми.
* Розробити необхідну базу даних Access і заповнити її знайденими даними.
* Створити WPF застосунок мовою С# та деякі елементи інтерфейсу в середовищі Expression Blend.
* Провести тестування програми для виявлення багів та проблем, виправити їх.

**Розділи програми** включають в себе:

* MY. Useful Info: Цікава та корисна інформація про сам університет та загалом про систему освіти США, що буде містити частину інформації, зібраної мною під час написання курсової роботи.
* Famous Alumni: галерея з портретами відомих випускників ВНЗ та інформацією про них.
* MY. Campus: інформація про бібліотеки, магазини, їдальні і кафе, лікарні, що знаходяться на території кампусу: адреси та номери телефону. А також факультети та коледжі, з яких складається університет.
* MY. Schedule: створення власного розкладу.
* MY. Settings: налаштування, що передбачають можливість вибору бази даних, що використовується, та директорій із фотографіями.

Основна частина

Теоретична частина з предмету

The U.S. Educational System

Undergraduate degrees at US universities typically take four years to complete. The first part of the degree is usually spent studying a wide range of subjects – some required and some electives – after which students choose the subject or subjects on which they want to focus – known as “major”. You may also earn a “minor” qualification alongside your major by completing classes in an additional field. This is often used to supplement the major degree. For example, business majors often minor in economics, while liberal arts majors may minor in philosophy. Some students also choose minors simply to explore a field they are interested in, without having to commit to a full study of the subject, and regardless of whether it pertains to the field in which they seek employment.

One of the most attractive features of the U.S. higher education system is the flexibility it provides through the number and diversity of institution types it encompasses.

This diversity offers students options to specialize in a variety of academic disciplines and even gain employment training.

More than 4,500 accredited institutions make up U.S. higher education in the United States. Unlike many countries, U.S. higher education institutions are not centrally organized or managed, but are accredited on a national or regional level by independent accrediting bodies.

A variety of institution types offer higher-education degrees. Liberal arts institutions, for example, offer courses in the arts, humanities, languages, and social and physical sciences. The majority of liberal arts institutions are private. Private colleges and universities are funded by a combination of endowments, gifts from alumni, research grants, and tuition fees. Private colleges and universities are usually smaller than public institutions and can have a religious affiliation or be single-sex schools.

Community colleges are another option and provide two-year associate degree programs to prepare students to continue studies for an undergraduate degree or help them gain occupational skills for immediate employment. State colleges and universities, also called "public universities", were founded and subsidized by U.S. state governments to provide a low-cost education to residents of that state. Public universities generally offer access to research opportunities and classes in a wide variety of fields of study. These universities tend to be very large and generally admit a wider range of students than private universities. Each student's interests will guide his/her choice among the many possibilities.

Regardless of the institution type, in the United States, students typically earn credits for courses they take and these credits count towards the completion of a program. Courses are often divided into "core" subject areas to provide the foundation of the degree program and "major" courses to provide specialization in a subject area. Students can also take "elective" courses to explore other topics of interest for a well-rounded educational experience.

The U.S. academic calendar typically runs from September to May and can be divided into two academic terms of 16-18 weeks known as semesters. Alternatively, some schools may operate on a quarter or trimester system of multiple terms of 10-12 weeks.

With the variety of available U.S. higher education options, students are sure to find the right fit for their academic, financial, and personal needs.

Many U.S. institutions offer a variety of student housing options and services on the college campus.

Living On Campus in The United States

Many U.S. institutions offer a variety of student housing options and services on the college campus.

On campus, students usually live in residence halls or dormitories, which are student apartment buildings that are often called "dorms." Among the options are dorms for men- or women-only, freshman-only, and academic-, language-, or theme-focused housing to support diverse student interests. Students usually share rooms, and the environment can be a great place to meet new friends quickly. Campus housing has on-site staff that is usually upperclassmen that know the campus well and serve as an information resource for all dorm residents.

Students dine in cafeterias located on campus where they can find a number of food options to meet cultural and dietary needs. Student health centers provider basic medical services to the student population.

Opportunities for recreation and club involvement help enrich the campus life experience. Academic, cultural, athletic, and special interest clubs meet to socialize and explore common interests. The housing, support services, and extracurricular activities on U.S. campuses help students form a community within their campus.

Теоретична частина з інформатики

Розробка роботи проводилось у середовищі Microsoft Visual Studio 2015 мовою C#, а база даних розроблена у програмі Access 2013. Для покращення інтерфейсу, деякі елементи були проектовані в середовищі Blend for Visual Studio 2015.

Підключення бази даних

Найважливішою частиною програми є підключення бази даних Access та обробка її даних програмними засобами. Для роботи з базою даних мені необхідно було ознайомитися з мовою SQL. Основні дії з базою даних виконуються за допомогою запитів SQL. Також у цій частині програми використовуються технології ADO.NET (ActiveX Data Objects .NET) — це набір бібліотек, що поставляється з Microsoft .NET Framework і призначений для взаємодії з різними сховищами даних з .NET-застосунків. Бібліотеки ADO.NET включають класи для приєднання до джерела даних, виконання запитів і обробки їхніх результатів.

Для роботи з базою даних необхідно підключити простір імен *System.Data.OleDb;* та *System.Data;.*

Подивимося на приклад заповнення DataGrid в залежності від вибору в ComboBox cmb\_major.

static string CommandText = "SELECT specialties.Code, directions.name\_d, credits.credits FROM specialties INNER JOIN (directions INNER JOIN credits ON directions.code\_d = credits.direction) ON specialties.Code = credits.specialty WHERE (((specialties.Code) = '" + cmb\_major.Text + "')) ";

static string ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=\"" + Environment.CurrentDirectory.ToString() + "\\db\_univ.mdb\"";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(CommandText, ConnectionString);

DataSet ds = new DataSet();

dataAdapter.Fill(ds, "specialties");

dataGrid.ItemsSource = ds.Tables["specialties"].DefaultView;

Анімація в Blend for Visual Studio 2015

При розробці анімації на формі Main Window, форма була спроектована за допомогою середовища Blend for Visual Studio 2015. У середовищі Microsoft Visual Studio 2015 при розробці WPF Application є можливість напряму перейти до проектування окремої форми у Blend. Ця функція дає змогу уникнути помилок, що виникають при підключенні форм стороннього проекту.

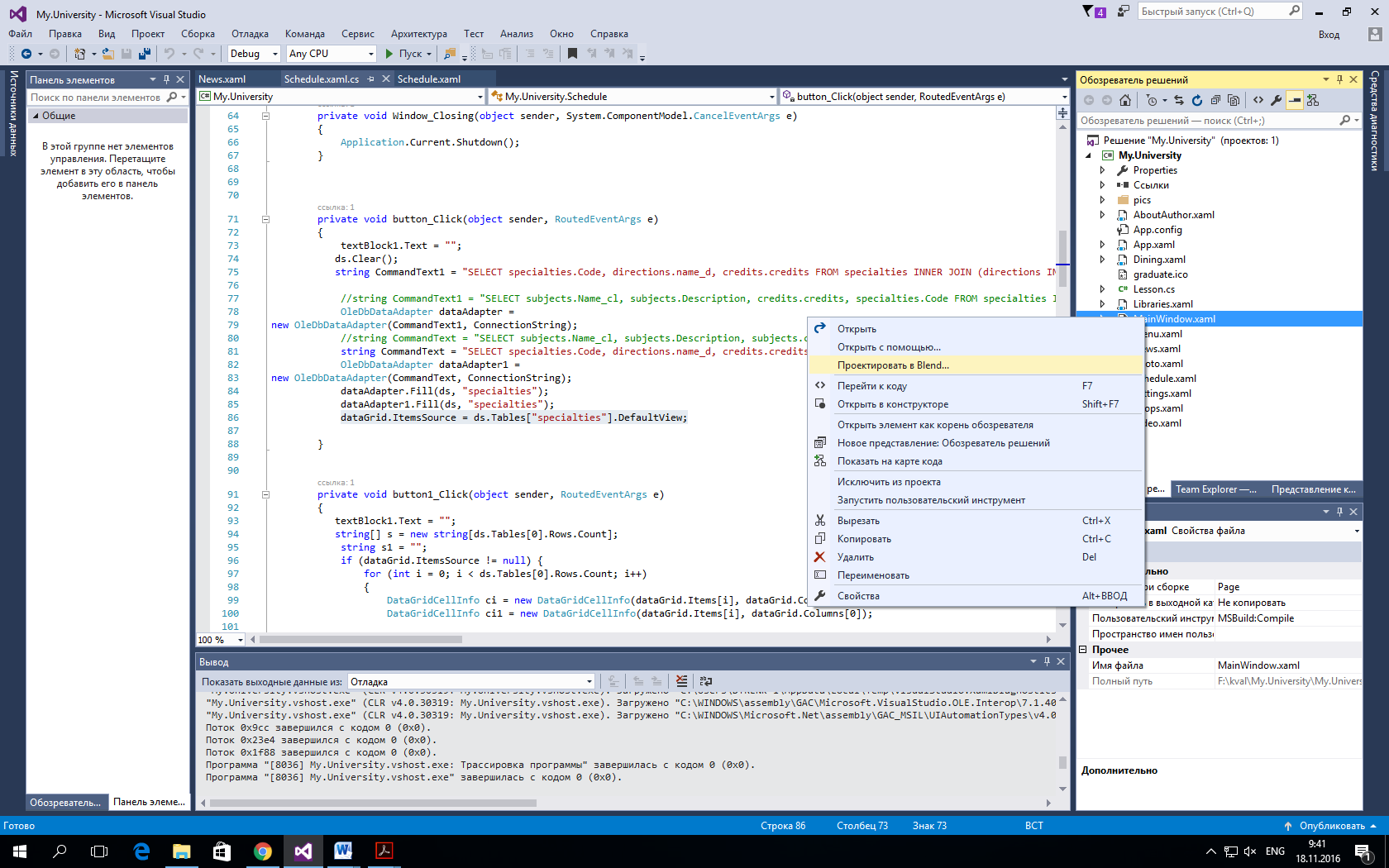


Рисунок 1

У додатку Blend for Visual Studio 2015 можливо створення анімації без кодування, на відміну від застосунку WPF Application у середовищі Microsoft Visual Studio 2015.

Для створення анімації, що відбувається при натисканні на об’єкт, необхідно створити тригер, який після виклику події [об’єкт].MouseLeftButtonDown відтворює Storyboard (OnMouseLeftButtonDown1). Її можна створити у вікні «Объекты и временная шкала» по ключовим кадрам. На рисунках ви бачите положення об’єктів для 0-ї секунди (рис. 2) та 1-ї секунди (рис. 3).

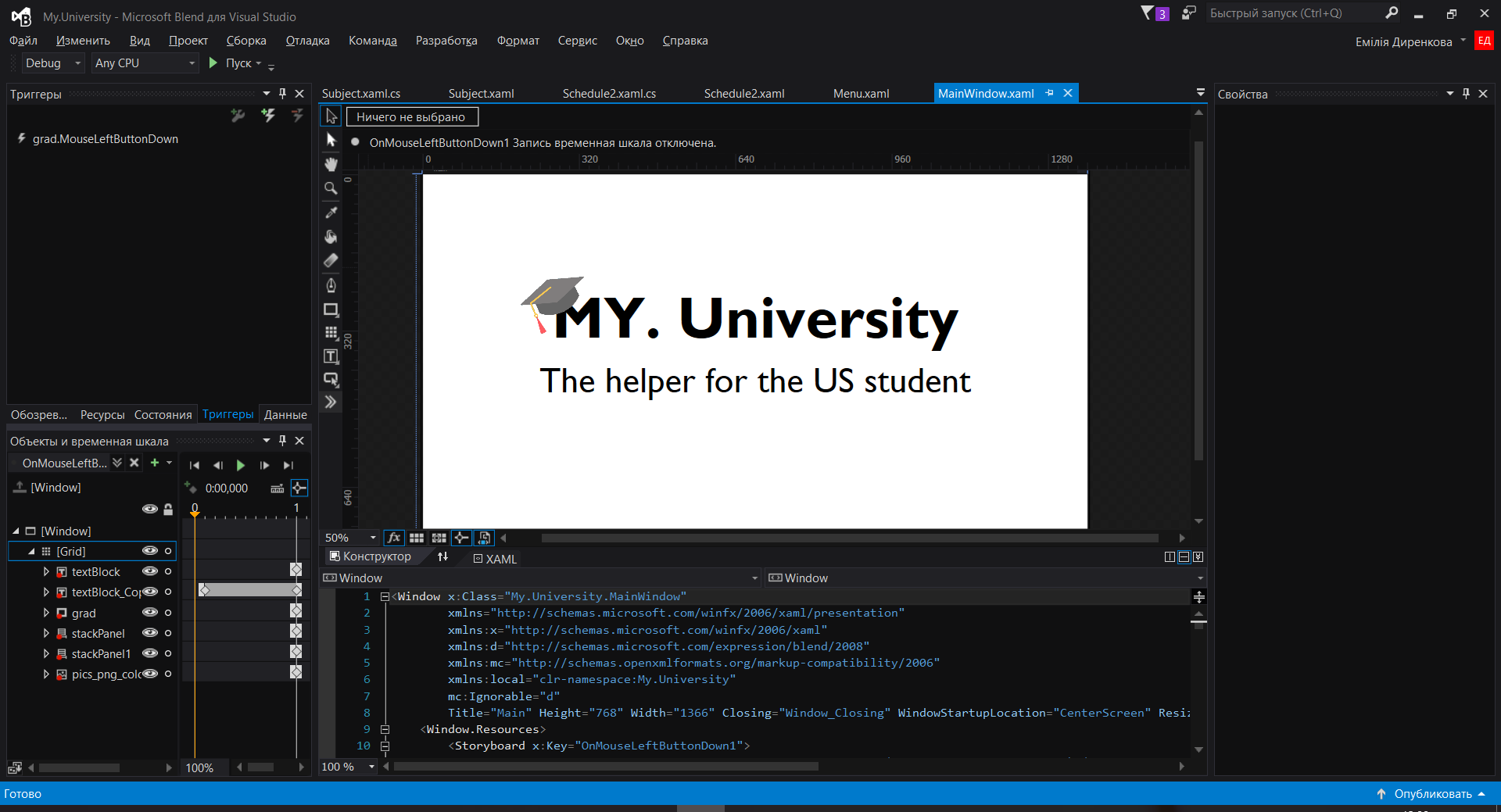


Рисунок 2. Blend for Visual Studio 2015

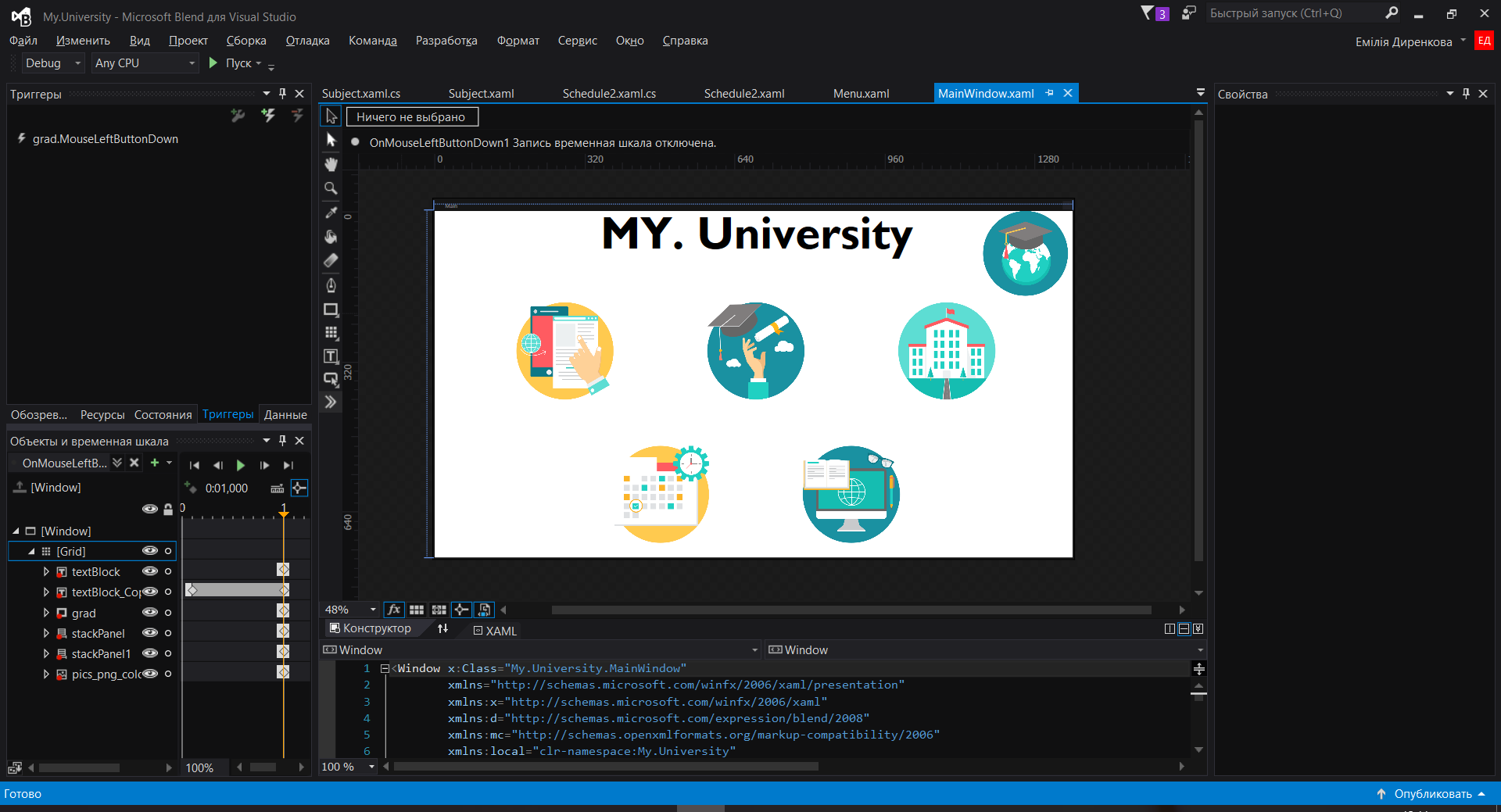


Рисунок 3. Blend for Visual Studio 2015

Динамічний список кнопок на формі «MY. Useful Info»

На рис. 4 можна побачити чотири елементи Image з логотипами соцмереж. Їх кількість та вид не є сталими, а залежать від даних у основній базі даних та вказаній теці.

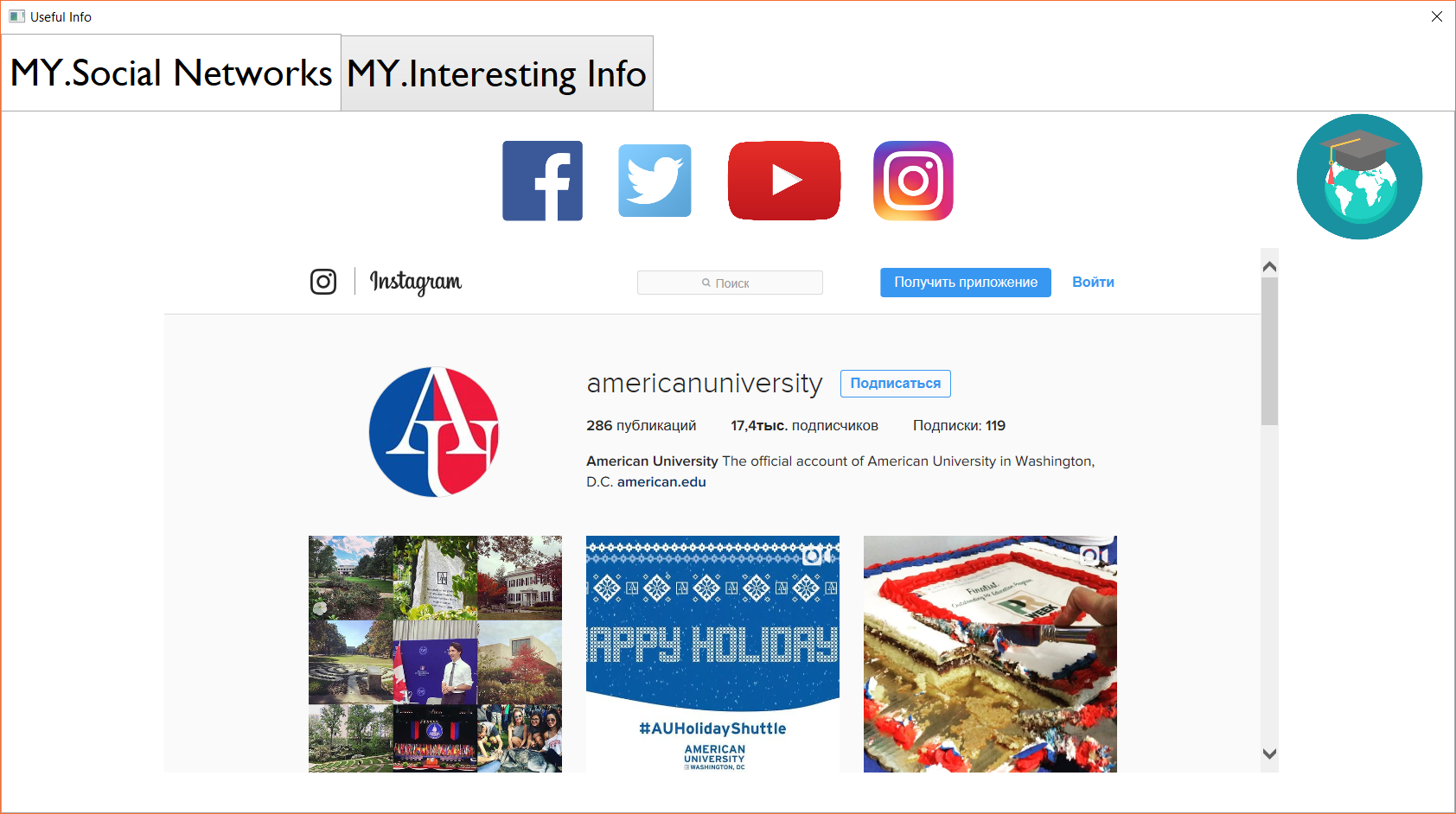


Рисунок 4. Форма "MY. Useful Info", розділ “MY. Social Networks”

На рис. 5 зображено зв'язок таблиць у базі даних Access. У таблиці «Social networks» містяться назви всіх можливих соціальних мереж і їх логотипи. У таблиці «University in social networks» вказано адреси сторінок університету в деяких соцмережах.

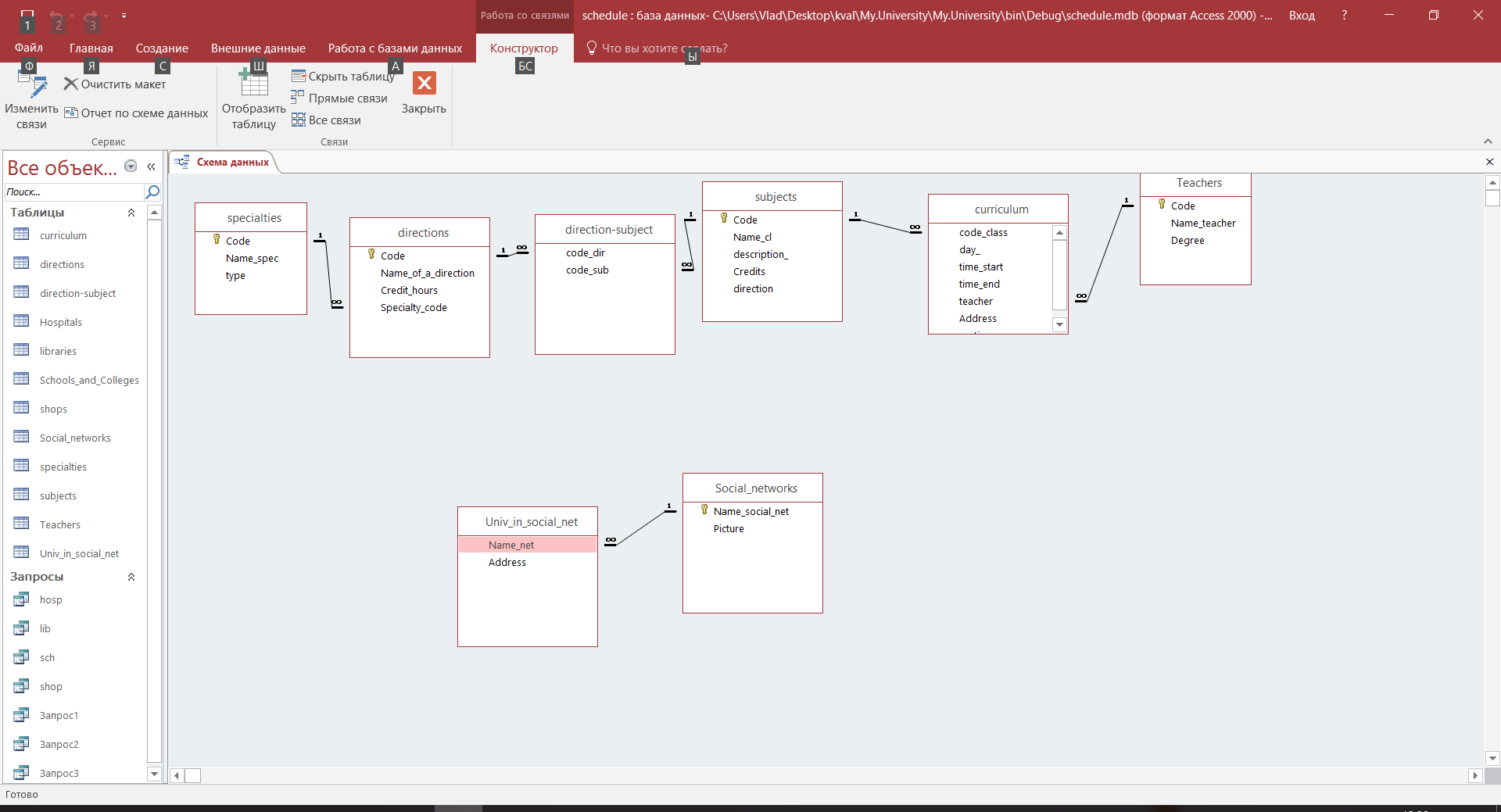


Рисунок 5. Зв'язок таблиць соціальних мереж

Програма підключається до бази даних за допомогою класу OleDbDataAdapter як було показано вище та «поєднує» дві таблиці за допомогою запиту: *"SELECT Univ\_in\_social\_net.Name\_net, Univ\_in\_social\_net.Address, Social\_networks.Picture FROM Social\_networks INNER JOIN Univ\_in\_social\_net ON Social\_networks.Name\_social\_net = Univ\_in\_social\_net.Name\_net".*

Потім створюється список елементів Image та заповнюється згідно назв картинок у базі даних. Також до кожного елемента додається подія, яка дозволяє при натисканні на елемент завантажити відповідну сторінку у WebBrowser.

List<Image> icons = new List<Image>();

for (int i = 0; i < ds.Tables["social\_networks"].Rows.Count; i++)

{

Image img = new Image();

ImageSourceConverter imgsc = new ImageSourceConverter();

img.Source = (ImageSource)imgsc.ConvertFromString(Settings.public\_folder\_name + "\\" + ds.Tables["social\_networks"].Rows[i].ItemArray[2].ToString());

img.Stretch = Stretch.Uniform;

img.Cursor = Cursors.Hand;

img.Margin = new Thickness(5, 5, 5, 5);

img.ToolTip = ds.Tables["social\_networks"].Rows[i].ItemArray[1].ToString();

img.MouseDown += new MouseButtonEventHandler(img\_MouseDown);

icons.Add(img);

stp\_icons.Children.Add(icons[i]);

}

private void img\_MouseDown(object sender, RoutedEventArgs e)

{

Image img = e.Source as Image;

wb\_socnet.Source = new Uri(img.ToolTip.ToString());

}

Створення розкладу

Найцікавішим розділом програми є “MY. Schedule”. Вище вже було показано як підключається база даних.

Після вибору major та minor при натисканні на кнопку “Show requirements” елемент TreeView заповнюється списком курсів, необхідних для отримання диплому.

private void Show\_button\_Click(object sender, RoutedEventArgs e)

{

trv\_req.Items.Clear();

ds.Tables["directions"].Clear();

string ComTxtMajor = "SELECT specialties.Code, directions.Name\_of\_a\_direction, directions.Credit\_hours FROM subjects INNER JOIN (specialties INNER JOIN(directions INNER JOIN[direction-subject] ON directions.Code = [direction-subject].code\_dir) ON specialties.Code = directions.Specialty\_code) ON subjects.Code = [direction-subject].code\_sub GROUP BY specialties.Code, directions.Name\_of\_a\_direction, directions.Credit\_hours HAVING(((specialties.Code) = '" + cmb\_major.Text + "'))";

string ComTxtMinor = "SELECT specialties.Code, directions.Name\_of\_a\_direction, directions.Credit\_hours FROM subjects INNER JOIN (specialties INNER JOIN(directions INNER JOIN[direction-subject] ON directions.Code = [direction-subject].code\_dir) ON specialties.Code = directions.Specialty\_code) ON subjects.Code = [direction-subject].code\_sub GROUP BY specialties.Code, directions.Name\_of\_a\_direction, directions.Credit\_hours HAVING(((specialties.Code) = '" + cmb\_minor.Text + "'))";

OleDbDataAdapter DaMajor = new OleDbDataAdapter(ComTxtMajor, ConnectionString);

OleDbDataAdapter DaMinor = new OleDbDataAdapter(ComTxtMinor, ConnectionString);

DaMajor.Fill(ds, "directions");

DaMinor.Fill(ds, "directions");

for (int i = 0; i < ds.Tables["directions"].Rows.Count; i++)

{

TreeViewItem trvi1 = new TreeViewItem();

string a = ds.Tables["directions"].Rows[i].ItemArray[1].ToString();

trvi1.Header = ds.Tables["directions"].Rows[i].ItemArray[1] + " (" + ds.Tables["directions"].Rows[i].ItemArray[2] + ")";

string ComTxtSub = "SELECT directions.Name\_of\_a\_direction, subjects.Name\_cl, subjects.Credits FROM subjects INNER JOIN (specialties INNER JOIN(directions INNER JOIN[direction-subject] ON directions.Code = [direction-subject].code\_dir) ON specialties.Code = directions.Specialty\_code) ON subjects.Code = [direction-subject].code\_sub GROUP BY directions.Name\_of\_a\_direction, subjects.Name\_cl, subjects.Credits HAVING(((directions.Name\_of\_a\_direction) = '" + a + "'))";

OleDbDataAdapter DaTrV = new OleDbDataAdapter(ComTxtSub, ConnectionString);

DaTrV.Fill(ds, "children-courses");

for (int j = 0; j < ds.Tables["children-courses"].Rows.Count; j++)

{

trvi1.Items.Add(new TreeViewItem() { Header = ds.Tables["children-courses"].Rows[j].ItemArray[1].ToString() + " (" + ds.Tables["children-courses"].Rows[j].ItemArray[2].ToString() + ")" });

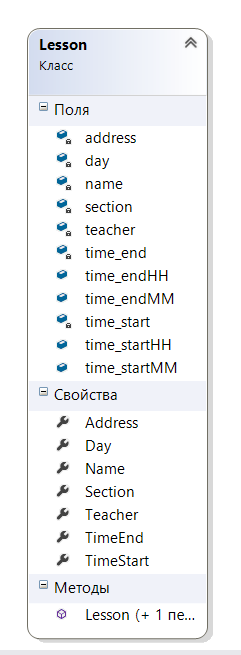
}

trv\_req.Items.Add(trvi1);

ds.Tables["children-courses"].Clear();

}

}

Для того, щоб переносити курси до розкладу було створено клас Lesson(рис. 6):

public string TimeEnd

{

get { return time\_end; }

set

{

time\_end = value;

time\_endHH = Convert.ToInt32(time\_end.Remove(2));

time\_endMM = Convert.ToInt32(time\_end.Substring(3, 2));

}

}

public string TimeStart

{

get { return time\_start; }

set

{

time\_start = value;

time\_startHH = Convert.ToInt32(time\_start.Remove(2));

time\_startMM = Convert.ToInt32(time\_start.Substring(3, 2));

}

}

Приклад отримання даних з DataGrid:

public static List<Lesson> result = new List<Lesson>();

DataGridCellInfo iname1 = new DataGridCellInfo(dataGrid\_cur.Items[i], dataGrid\_cur.Columns[0]);

var name1 = iname1.Column.GetCellContent(iname1.Item) as TextBlock;

result.Add(new Lesson(namee1.Text, day1.Text, time\_start1.Text, time\_end1.Text, address1.Text, teacher1.Text, section1.Text));

Рисунок 6. Склад класу Lesson

При натисканні на кнопку “Check” здійснюється перевірка розкладу на наявність однакових курсів та занять, що відбуваються за взаємовиключним розкладом:

private void CheckButton\_Click(object sender, RoutedEventArgs e)

{

if (dataGrid.Items.Count > 1)

{

for (int i = 0; i < Subject.result.Count; i++)

{

for (int j = 0; j < Subject.result.Count; j++)

{

if (Subject.result[i].Name == Subject.result[j].Name && Subject.result[i].Section != Subject.result[j].Section && i != j)

{var dgri = dataGrid.ItemContainerGenerator.ContainerFromIndex(i) as DataGridRow;

var dgrj = dataGrid.ItemContainerGenerator.ContainerFromIndex(j) as DataGridRow;

dgri.Background = Brushes.Red;

dgrj.Background = Brushes.Red;}

else {

if (Subject.result[i].Day == Subject.result[j].Day && i != j)

{

DateTime dtstarti = new DateTime(2000, 10, 1, Subject.result[i].time\_startHH, Subject.result[i].time\_startMM, 0);

DateTime dtendi = new DateTime(2000, 10, 1, Subject.result[i].time\_endHH, Subject.result[i].time\_endMM, 0);

DateTime dtstartj = new DateTime(2000, 10, 1, Subject.result[j].time\_startHH, Subject.result[j].time\_startMM, 0);

DateTime dtendj = new DateTime(2000, 10, 1, Subject.result[j].time\_endHH, Subject.result[j].time\_endMM, 0);

if (dtstarti == dtstartj || (dtstarti > dtstartj && dtstarti < dtendj) || (dtstartj > dtstarti && dtstartj < dtendi))

{

var dgri = dataGrid.ItemContainerGenerator.ContainerFromIndex(i) as DataGridRow;

var dgrj = dataGrid.ItemContainerGenerator.ContainerFromIndex(j) as DataGridRow;

dgri.Background = Brushes.Red;

dgrj.Background = Brushes.Red;

}}}}}}}

Коли розклад перевірено, можна створити документ та роздрукувати його. Для цього були підключені бібліотеки System.Windows.Xps.Packaging та Microsoft.Office.Interop.Word. Документ створюється таким чином:

private void DocButton\_Click(object sender, RoutedEventArgs e)

{

n++;

Microsoft.Office.Interop.Word.Application schedule = new Microsoft.Office.Interop.Word.Application();

schedule.Visible = false;

object missing = System.Reflection.Missing.Value;

Document document = schedule.Documents.Add(ref missing, ref missing, ref missing, ref missing);

List<string> days\_of\_week = new List<string>() { "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun" };

foreach (Microsoft.Office.Interop.Word.Section section in document.Sections)

{

Range headerRange = section.Headers[WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;

headerRange.Fields.Add(headerRange, WdFieldType.wdFieldPage);

headerRange.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphCenter;

headerRange.Font.Name = "Gill Sans MT";

headerRange.Font.Size = 20;

headerRange.Text = "My Schedule";

}

for (int i = 0; i < days\_of\_week.Count; i++)

{

Microsoft.Office.Interop.Word.Paragraph para1 = document.Content.Paragraphs.Add(ref missing);

para1.Range.Text += days\_of\_week[i] + Environment.NewLine;

List<Lesson> day = new List<Lesson>();

for (int j = 0; j < Subject.result.Count; j++)

{

if (Subject.result[j].Day.Contains(days\_of\_week[i]) == true)

{

day.Add(Subject.result[j]);

}

}

Microsoft.Office.Interop.Word.Table schedule\_table = document.Tables.Add(para1.Range, day.Count + 1, 4, ref missing, ref missing);

schedule\_table.Borders.Enable = 1;

schedule\_table.Rows[1].Cells[1].Range.Text = "Name of the class";

schedule\_table.Rows[1].Cells[2].Range.Text = "Time";

schedule\_table.Rows[1].Cells[3].Range.Text = "Teacher";

schedule\_table.Rows[1].Cells[4].Range.Text = "Address";

for (int a = 0; a < day.Count; a++)

{

for (int j = 0; j < day.Count - a - 1; j++)

{

DateTime dtstartj1 = new DateTime(2000, 10, 1, day[j].time\_startHH, day[j].time\_startMM, 0);

DateTime dtstartj2 = new DateTime(2000, 10, 1, day[j + 1].time\_startHH, day[j + 1].time\_startMM, 0);

if (dtstartj1 > dtstartj2)

{

Lesson temp = day[j];

day[j] = day[j + 1];

day[j + 1] = temp;

}

}

}

for (int j = 0; j < day.Count; j++)

{

schedule\_table.Rows[j + 2].Cells[1].Range.Text = day[j].Name;

schedule\_table.Rows[j + 2].Cells[2].Range.Text = day[j].TimeStart + " - " + day[j].TimeEnd;

schedule\_table.Rows[j + 2].Cells[3].Range.Text = day[j].Teacher;

schedule\_table.Rows[j + 2].Cells[4].Range.Text = day[j].Address;

}

}

object filename = Settings.schedule\_save\_folder\_name + "\\schedule" + n + ".docx";

object filename1 = Settings.schedule\_save\_folder\_name + "\\schedule" + n + ".xps";

document.SaveAs2(ref filename);

document.SaveAs2(ref filename1, WdSaveFormat.wdFormatXPS);

document.Close(ref missing, ref missing, ref missing);

document = null;

schedule.Quit(ref missing, ref missing, ref missing);

schedule = null;

XpsDocument doc = new XpsDocument(Settings.schedule\_save\_folder\_name + "\\schedule" + n + ".xps", FileAccess.Read);

documentViewer.Document = doc.GetFixedDocumentSequence();

}

Опис роботи

Застосунок не є орієнтованим на якийсь один університет, він повністю залежить від підключеної бази даних та змісту обраних тек.

Після запуску програми користувач бачить форму *About the Author (рис.7).*

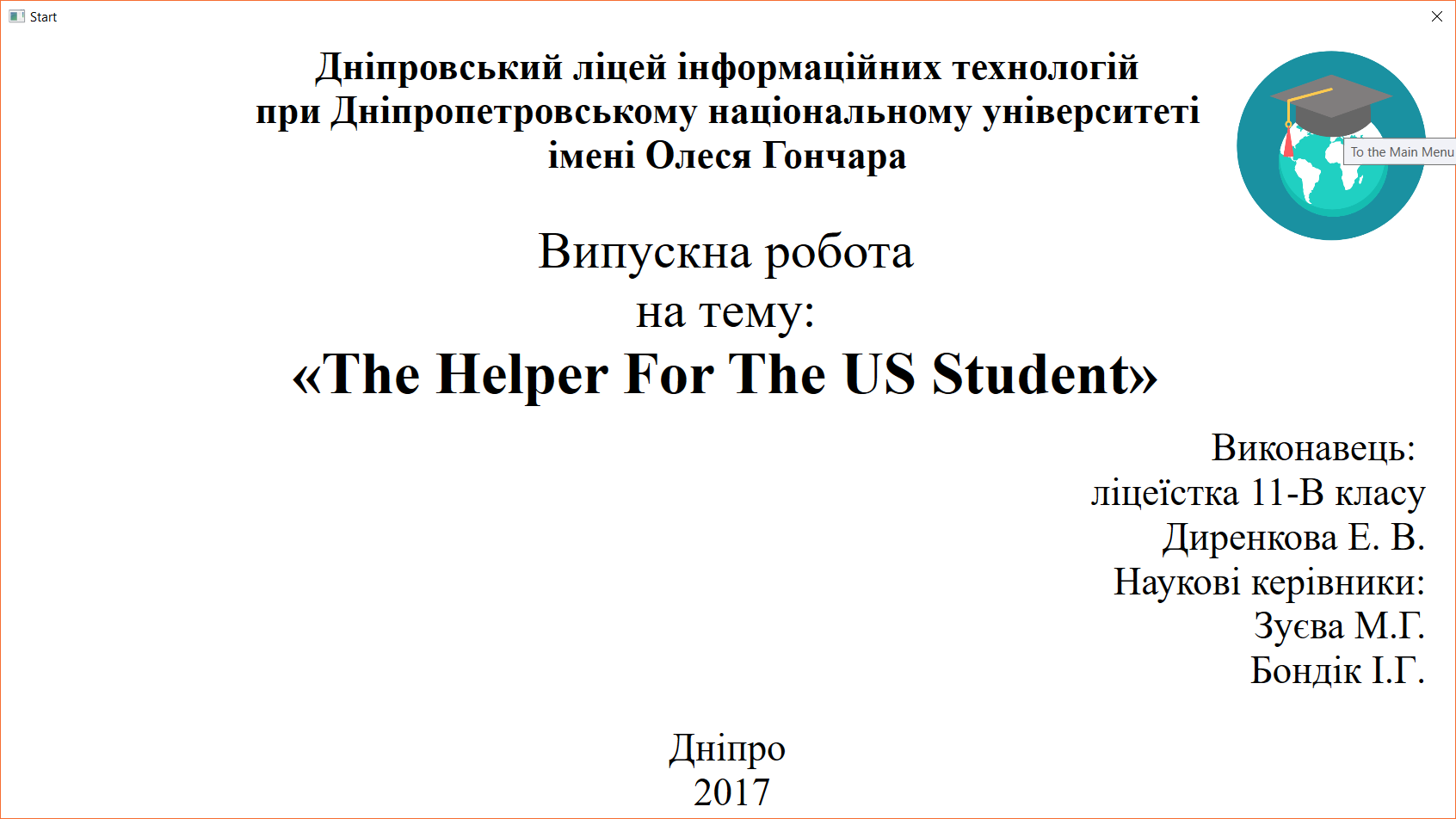


Рисунок 7. About the Author

При натисканні на кнопку відбувається перехід до головного меню *(рис.8).* На інших формах ця кнопка означає перехід до меню.





MY. Settings

MY. Schedule

MY. Campus

Famous Alumni

MY. Useful Info

Start

Рисунок 8. Main Menu

При виборі розділу *Useful Info* ми переходимо до відповідної форми *(рис. 9,10).* Її мета – ознайомлення з цікавою та корисною інформацією для студента: посилання сторінок соціальних мереж університету та інформація про десятку кращих університетів США.

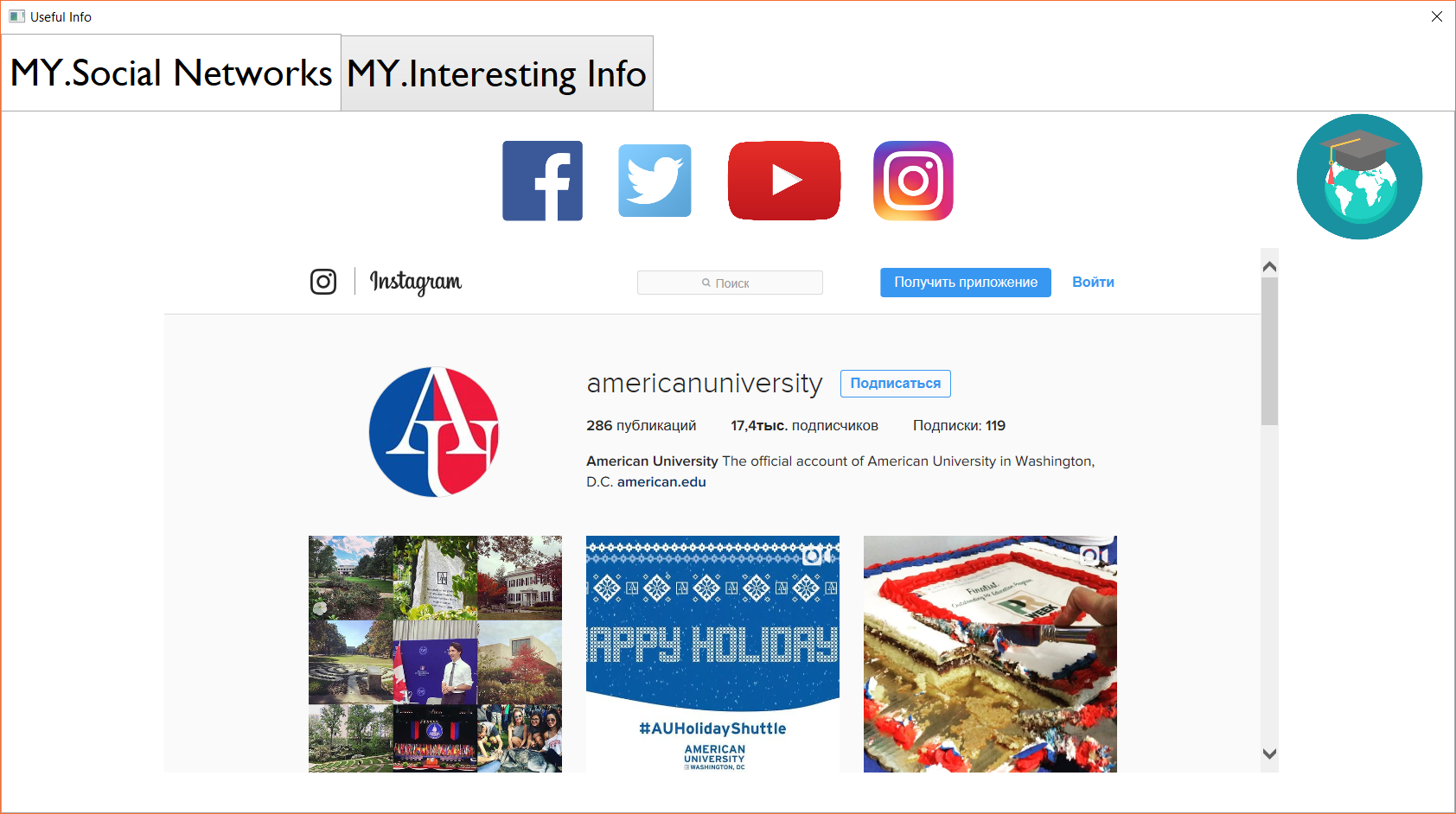


Рисунок 9. Social Networks



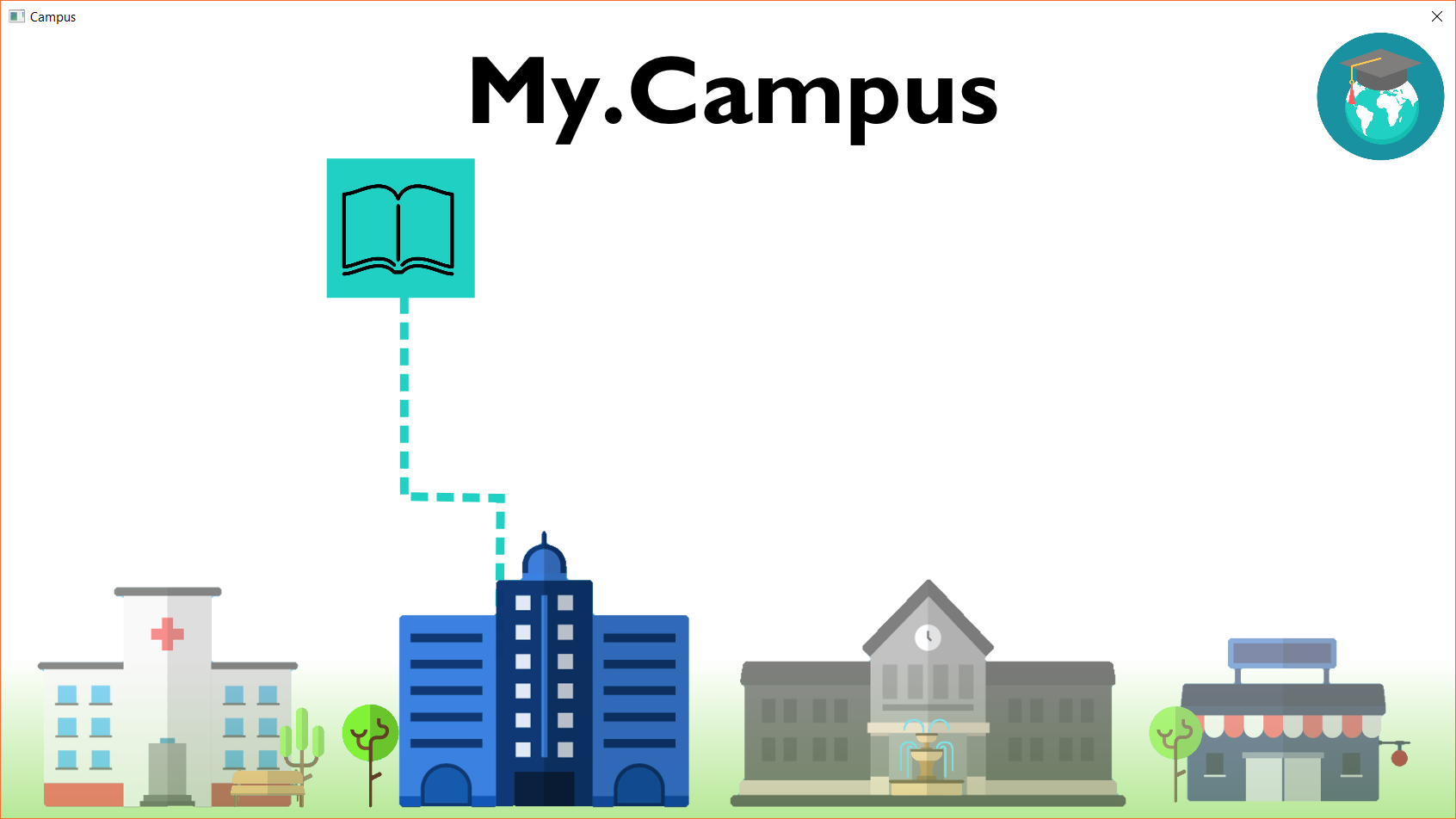
Рисунок 10. Interesting Info

При виборі розділу *Famous Alumni (рис. 11)* ми переходимо до форми із фотографіями відомих випускників, що завантажуються з папки обраної у формі *Settings*. Також доступна інформація про цих людей.



Рисунок 11. Famous Alumni

Переходячи до розділу *Campus (рис.12, 13)*, отримуємо доступ до форми із інформацією про бібліотеки, магазини, їдальні і кафе, що знаходяться на території кампусу: адреси та номери телефону.



Hospitals

Colleges

Shops, cafes

Libraries

Рисунок 12. MY. Campus

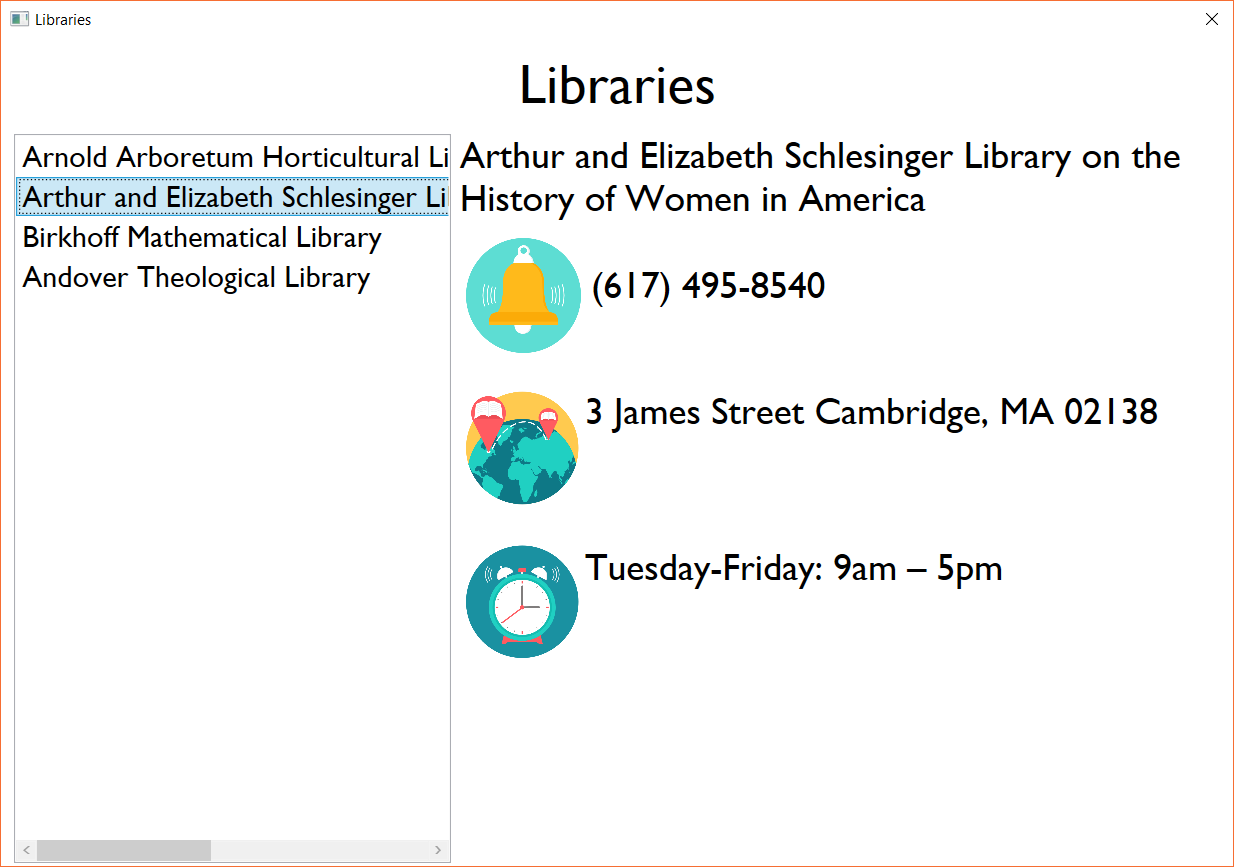
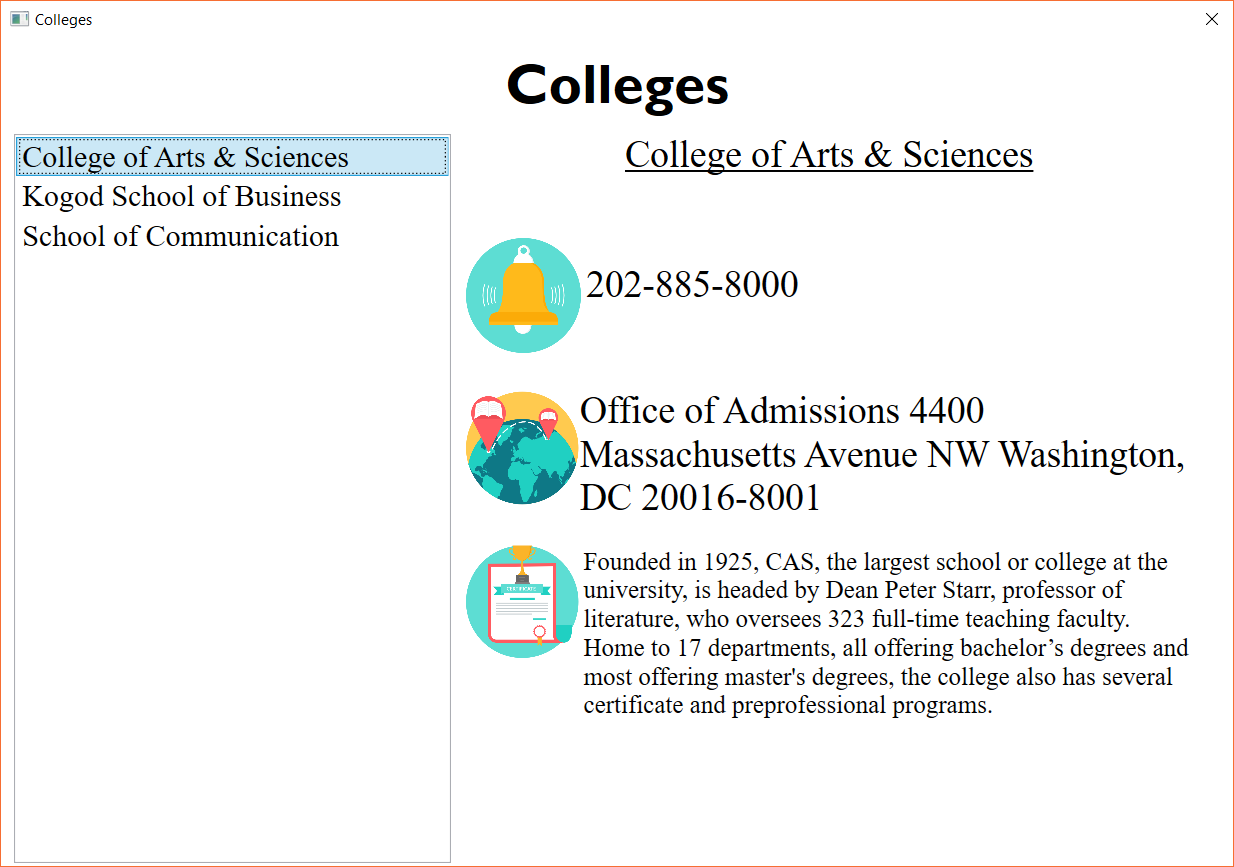
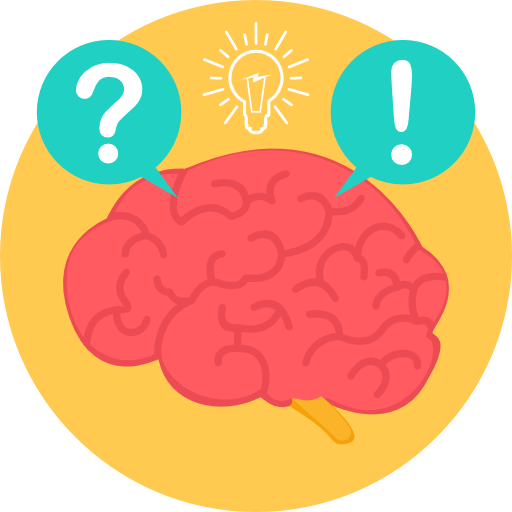


Рисунок 13. Детальна інформація у MY. Campus

Найцікавішим розділом програми є *MY. Schedule* *(рис.14, 15).* Тут можливе інтерактивне створення власного розкладу та його роздрукування. Користувачем обирається головна (major) та другорядна (minor) спеціальність (якщо вона є). Після цього у TreeView з’являється список напрямків курсів учня та кількість кредитів для цього напрямку, що треба пройти для успішного отримання диплому. При розкритті одного з напрямків, користувач бачить курси, що можна пройти для отримання кредитів з цього напрямку. Клацнувши два рази на курс, користувач отримує доступ до докладного опису курсу (рис. 15) та його розкладу занять. Обравши одне заняття із секції, потрібно натиснути на кнопку «Add to the schedule». До DataGrid на основній формі автоматично додаються всі заняття цієї секції. Користувачу слід якомога частіше натискати кнопку “Check”, щоб перевіряти на наявність занять, що проходять в один час або на наявність дублювання однакових курсів. Щоб видалити курс, потрібно натиснути на кнопку “Delete course”. Вона видаляє всі заняття одного курсу аналогічно як вони й додаються.

Після завершення вибору курсів, є можливість створити документ .docx та .xps. Для цього потрібно натиснути на кнопку “Make document”. Готовий документ буде відображатися в елементі DocumentViewer. Серед його функцій є роздрукування.

Якщо у користувача виникають труднощі можна скористатися допомогою, клацнувши на кнопку .



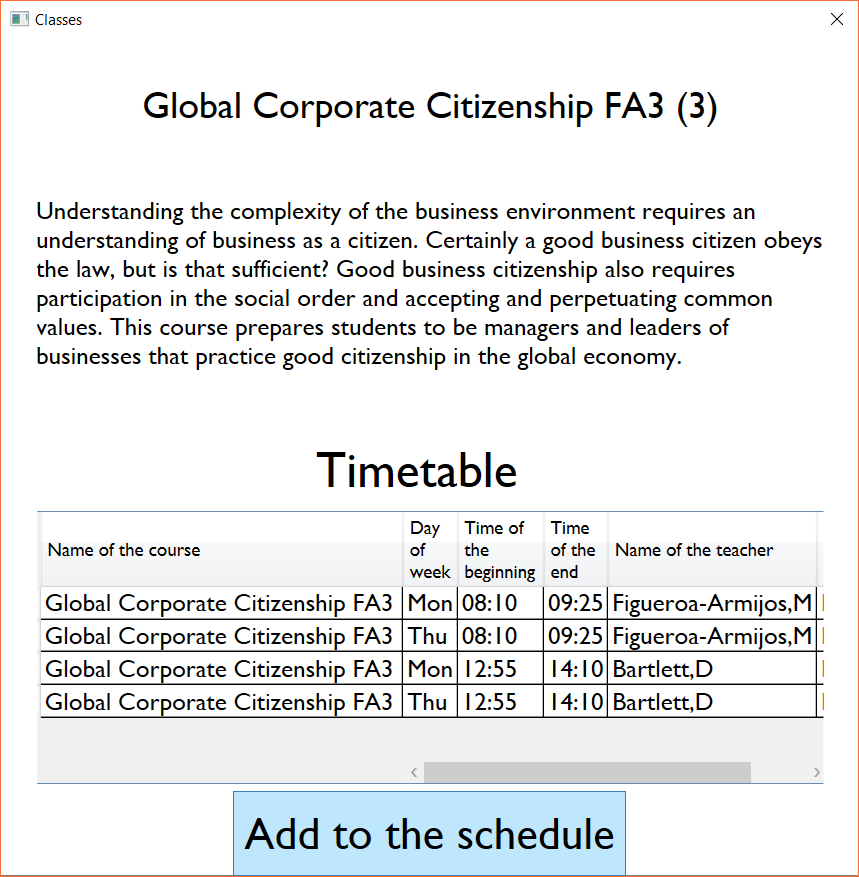
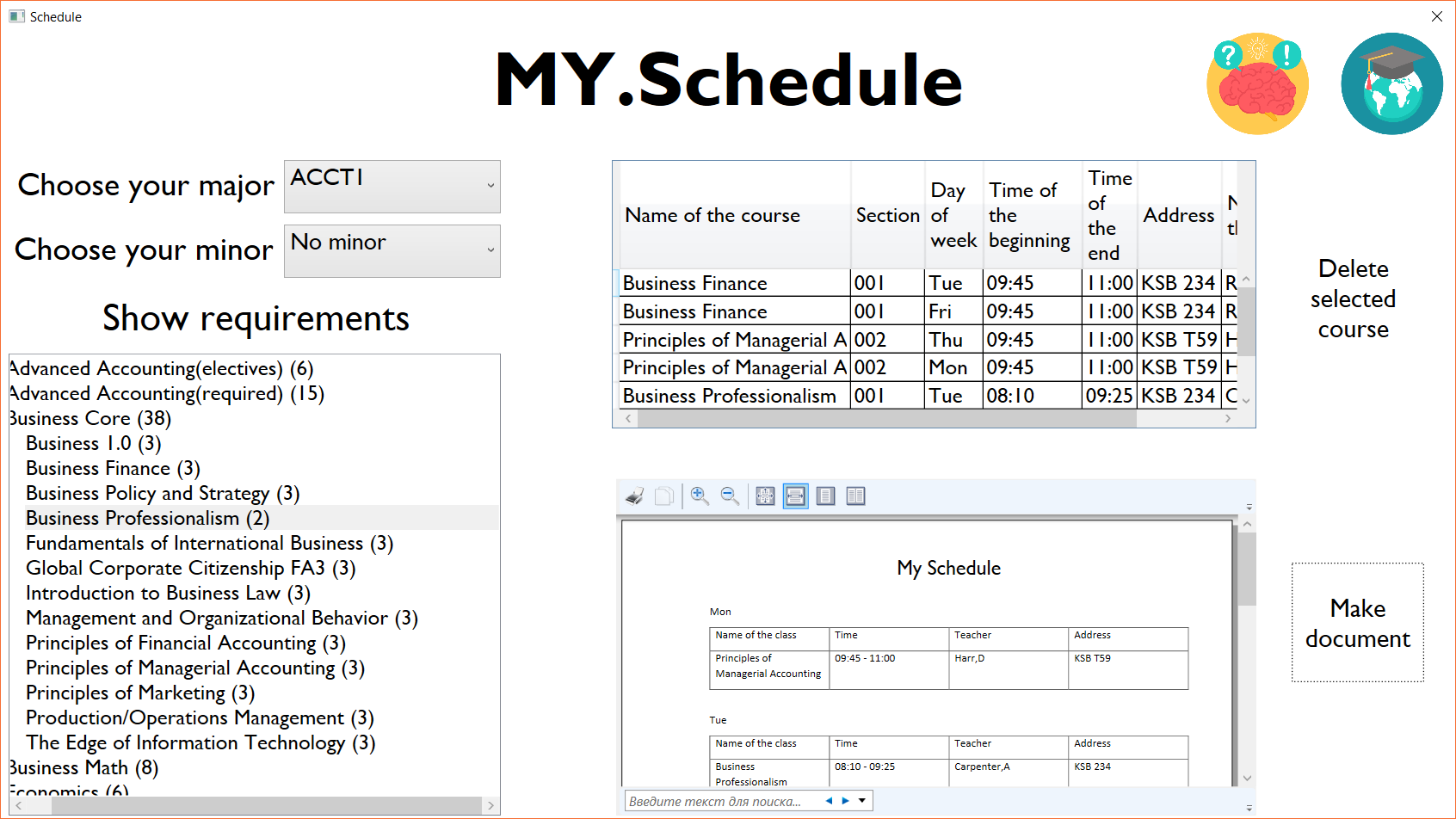


Рисунок 14. Schedule

Останнім розділом програми є *Settings* – налаштування. що передбачають можливість вибору бази даних, що використовується та директорій із фотографіями та відео. Щоб відкрити OpenFileDialog треба натиснути на TextBox двічі.

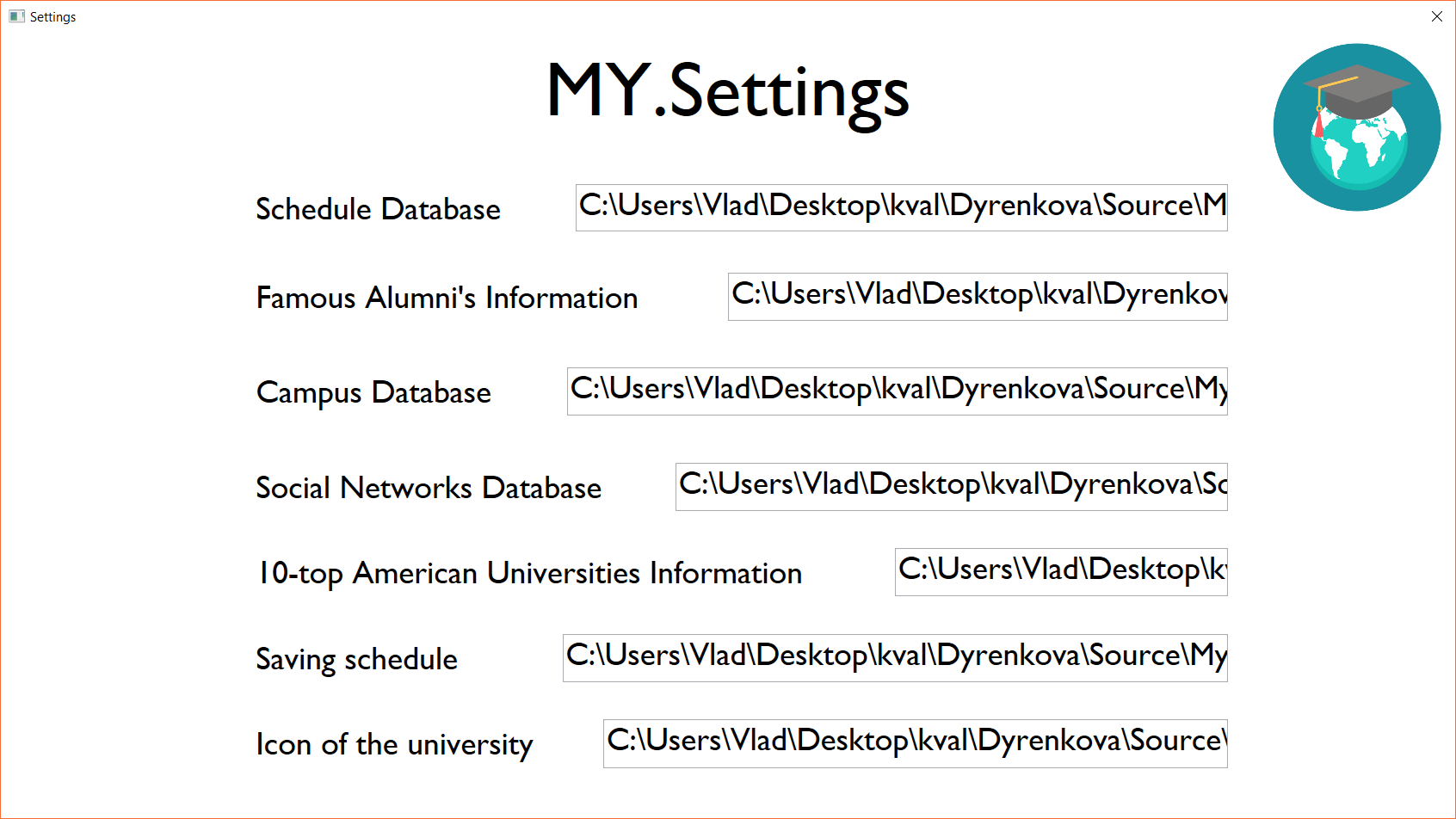


Рисунок 15. Settings

Структура програми

Рисунок 16. Блок-схема програми

Напрямки, де можна використати роботу

Застосунок призначений для студентів вищих навчальних закладів США, особливо для тих, хто походить з інших країн. Студент може використовувати її як посібник цікавої та необхідної інформації та як асистента у складанні розкладу. Також, студент сам може додавати необхідну для нього інформацію в базу даних або папки з документами чи фотографіями.

Можливе використання програми на заняттях англійської мови для ознайомлення з системою вищої освіти в США.

Програмно-апаратні вимоги

* Мінімальна роздільна здатність монітору: 1360х760;
* 20 MB вільного простору на диску;
* 350 МБ ОЗУ;
* ОС WindowsXP або вище з архітектурою x64.
* Доступ до інтернету (необов’зково)

Комплектація програми

Данна програма працює з такими файлами:

* MY.University.exe – файл запуску програми;
* Файли \*.jpg і \*.png – це картинки, що використовуються програмою;
* Файли \*.mdb – це файл бази даних;
* Файли \*.htm і \*.txt – це файли, що містять текстові дані для програми.

Використані програмні засоби

Для створення застосунку були використані:

* Microsoft Visual Studio 2015
* Blend for Visual Studio 2015
* Microsoft Access 2015
* Microsoft Word 2015
* Gimp 2

Висновки

Під час роботи над застосунком, я поглибила свої знання та навички, пов’язані з розробкою у середовищі Microsoft Visual Studio 2015 мовою С# та мовою розмітки XAML. Також для роботи з базою даних я оволоділа базовими знаннями мови SQL.

Для ознайомлення з системою вищої освіти США, я прочитала багато статей англійською мовою та спілкувалася зі студентами. Ця практика мови була дуже корисною та цікавою для мене.

Отже, результатом моєї праці є електронний допоміжник студента американських ВНЗ, написаний мовою С# у середовищі Microsoft Visual Studio 2015, що є надійним персональним асистентом для користувача.

Список використаних джерел

1. РАБОТА С БАЗАМИ ДАННЫХ НА ЯЗЫКЕ C#. ТЕХНОЛОГИЯ АDO .NET: учебное пособие / сост. О. Н. Евсеева, А. Б. Шамшев. – Ульяновск: УлГТУ, 2009.

Електронні ресурси:

1. Список відомих випускників: Famous American University Alumni [Електронний ресурс] – Режим доступу до ресурсу: <http://www.ranker.com/list/famous-american-university-alumni-and-students/reference>.
2. Система обучения в американском университете на специальность Computer Science [Електронний ресурс] – Режим доступу до ресурсу: <https://habrahabr.ru/post/193150/>.
3. Рейтинг 10 кращих університетів США взято: National Universities Rankings [Електронний ресурс] – Режим доступу до ресурсу: <http://colleges.usnews.rankingsandreviews.com/best-colleges/rankings/national-universities>.
4. Інформація про університети взята з офіційних сайтів:
   * Princeton University [Електронний ресурс] – Режим доступу до ресурсу: <http://www.princeton.edu/main/>.
   * Harvard University [Електронний ресурс] – Режим доступу до ресурсу: <http://www.harvard.edu/>.
   * Yale University [Електронний ресурс] – Режим доступу до ресурсу: <http://www.yale.edu/>.
   * Columbia University [Електронний ресурс] – Режим доступу до ресурсу: <http://www.columbia.edu/>.
   * Stanford University [Електронний ресурс] – Режим доступу до ресурсу: <http://www.stanford.edu/>.
   * University of Chicago [Електронний ресурс] – Режим доступу до ресурсу: <http://www.uchicago.edu/>.
   * Massachusetts Institute of Technology [Електронний ресурс] – Режим доступу до ресурсу: <http://www.mit.edu/>.
   * Duke University [Електронний ресурс] – Режим доступу до ресурсу: <https://duke.edu/>.
   * University of Pennsylvania [Електронний ресурс] – Режим доступу до ресурсу: <http://www.upenn.edu/>.
   * California Institute of Technology [Електронний ресурс] – Режим доступу до ресурсу: <http://www.caltech.edu/>.