**МІНІСТЕРТСВО ОСВІТИ І НАУКИ УКРАЇНИ**

Національний університет водного господарства та природокористування

Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки

Кафедра прикладної математики

**Курсова робота**

на тему:

«ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ «SMS ОПОВІЩЕННЯ УСПІШНОСТІ СТУДЕНТІВ»

Виконав: студент групи

ПМ-21

Золотарьов Валентин

Перевірив: ст. викладач

Жуковський В. В.

Рівне 2017

ЗМІСТ

[ВСТУП 2](#_Toc483926308)

[МЕТА 3](#_Toc483926309)

[ЗАВДАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ 4](#_Toc483926310)

[АКТУАЛЬНІСТЬ РОБОТИ 5](#_Toc483926311)

[ПЕРЕВАГИ ДОДАТКУ 6](#_Toc483926312)

[ПЕРЕВАГИ СМС ОПОВІЩЕНЬ 7](#_Toc483926313)

[ДІАГРАМА КЛАСІВ 8](#_Toc483926314)

[ДІАГРАМА ВИКОРИСТАНЯ 9](#_Toc483926315)

[ДІАГРАМА БАЗИ ДАНИХ 10](#_Toc483926316)

[ОПИС ПРОЕКТУ 11](#_Toc483926317)

[ОПИС ПРОГРАМИ 11](#_Toc483926318)

[ОПИС ІНТЕРФЕЙСУ 12](#_Toc483926319)

[КОД ПРОГРАМИ 17](#_Toc483926320)

[ВИСНОВКИ 22](#_Toc483926321)

[ЛІТЕРАТУРА 23](#_Toc483926322)

## ВСТУП

Нaявність мобільного телефону на сьогоднішній день є нормою для кожної цивілізованої людини. Зaвдяки широким функціональним можливостям даний пристрій вже давно витіснило стаціонарні телефони і стало незамінним помічником в повсякденному житті кожного.

Мобільні телефони змінили світ – немає потреби аргументувати очевидність цього факту. Змінився спосіб спілкування, та й саме поняття «бути на зв’язку» зазнало суттєвої зміни: більшість людей залишаються на зв’язку 24 години на добу – завдяки маленьким пристроям. Тому саме смс оповіщення є найбільш ефективним варіантом для оповіщення студентів.

Серед складових навчального процесу в університеті, які суттєво впливають на підвищення його ефективності, важливе місце займає контроль і оцінка знань студентів, як засіб педагогічного керування їх пізнавальною діяльністю.

## МЕТА

Метою такого проекту роботи є обґрунтування та розробка програмного забезпечення процесу автоматизованого оповіщення про рівень успішності студентів вузів.

Зміст процесу полягає в здійсненні регулярної поетапної оцінки та корегування підготовки фахівців з метою удосконалення організації навчального процесу в цілому.

Контроль здійснюється в різних формах для того, щоб встановити:

* Рівень якості знань, отриманих студентами
* Здібності та навички отримані під час навчання
* Рівень підготовки студентів до сприйняття нового матеріалу
* Напрямки стимулювання навчально-пізнавальної діяльності
* Рівень готовності випускників вузів до професійної діяльності

В умовах Болонського процесу актуальність розробки ефективних методів управління навчальним процесу вузу ще більше зростає.

Дослідженням і створенням автоматизованих систем навчання і контролю знань у вузах займалися і займаються багато вчених, але традиційно до основних функцій системи контролю фахівці відносять : навчальну, контролюючу, розвиваючу і виховну функції.

## ЗАВДАННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ

Ціллю виконання курсової роботи є програмна реалізація на об’єктно-орієнтованій мові програмування з безпечною системою типізації платформи ASP.NET - C#, SMS оповіщення успішності студентів».

Основним завданням є створення можливостей для підвищення рівня результативності виховної роботи викладача зі студентом, має відбутися за рахунок оперативності оповіщення осіб, які мають безпосередній вплив на нього(батьків, спонсорів, майбутніх роботодавців), про успішність і відвідуваність занять. Переваги даної системи полягають в її точності, невблаганності і неупередженості.

## АКТУАЛЬНІСТЬ РОБОТИ

Актуальними методами розсилки інформації про успішність слід вважати ті з них, які відповідають основним вимогам швидкості, надійності, низької собівартості та доступності.

Проаналізувавши всі перераховані вище вимоги до методів оповіщення, виділимо наступні: e-mail, блоги і форуми, розсилка SMS-повідомлень. При цьому останній метод найбільш підходить за всіма критеріями оцінки, адже мобільний телефон сьогодні є найпоширенішим і найдоступнішим засобом передачі інформації.

Передавати інформацію з сервера через повідомлення можна кількома способами, відрізняються вони лише наявністю GSM модуля. Перший спосіб – існує можливість підключити GSM модуль ( мобільний телефон, модем і т.д.) безпосередньо до персонального комп’ютера, другий – існує можливість відправляти повідомлення через шлюз. Тобто, за допомогою однієї з багатьох фірм – посередників(починаючи від самих операторів стільникового зв’язку, і закінчуючи компаніями, які спеціалізуються на розсилці) виходить порт, на який надсилаються повідомлення, потім приходять до адресата.

При цьому SMS-повідомлення перетворюються в повідомлення електронної пошти або HTTP – запити і навпаки. Повідомлення, надходить через подібний шлюз, може бути безкоштовним, проте при цьому накладається багато технічних обмежень, таких як обмеження на кількість відправлених повідомлень. Тому в контексті переглянутої таким чином є актуальним тільки в разі придбання можливості розсилки попереднього повідомлення.

## ПЕРЕВАГИ ДОДАТКУ

ДЛЯ ВНЗ:

* Дозволяє підвищити успішність і відвідуваність учнів
* Покращує взаємодію з батьками і зі студентами
* Швидко інформує батьків і студентів про успішність студента
* Оперативний обмін інформацією між ВУЗом та батьками

ДЛЯ БАТЬКІВ:

* Контроль за оцінками своєї дитини, можливість прийняття своєчасних заходів для забезпечення його успішності
* Брати активну участь в освіті своїх дітей
* Підвищення успішності учнів
* Постійно перебувати в контакті з ВУЗом, економлячи при цьому свій час
* Можливість оперативно отримувати актуальну інформацію про навчальний процес

ДЛЯ СТУДЕНТІВ:

* Бути поінформованим про результати успішності та постійно володіти актуальною інформацією про навчальний процес

## ПЕРЕВАГИ СМС ОПОВІЩЕНЬ

Перевагами SMS оповіщень над будь-якими іншим способами є:

* **Ефективність** - SMS-розсилка швидше, зручніше і ефективніше, ніж будь-який інший спосіб інформування Клієнтів. SMS-повідомлення у Клієнта завжди під рукою і може бути прочитано кілька разів.
* **Низька вартість контакту** - SMS-розсилка дешевше, ніж розсилка звичайних листів.
* **Пізнаваність** - можливість вказати як номер відправника будь-яку літерну назву (до 11 символів), що збігається з назвою ВНЗ.
* **Персоніфікація** - звернення до кожного Клієнта по імені і автоматична підстановка інших даних в текст SMS-повідомлення. Навіть відправляючи розсилку на кілька тисяч Клієнтів, Ви звернетеся до кожного Клієнта по імені.
* **Прозорість** - ніяких прихованих платежів немає, Ви платите тільки за SMS-повідомлення. За результатами кожної SMS-розсилки надається деталізований звіт.
* **Своєчасність** - висока швидкість відправки. Всі Ваші пропозиції будуть доставлені Клієнтам в лічені хвилини незалежно від обсягу розсилки. При цьому можлива розсилка і по частинах, розтягнута на декілька годин або днів, щоб Ваші менеджери змогли якісно обробити всі запити.

## ДІАГРАМА КЛАСІВ

(CLASS DIAGRAM (UML))

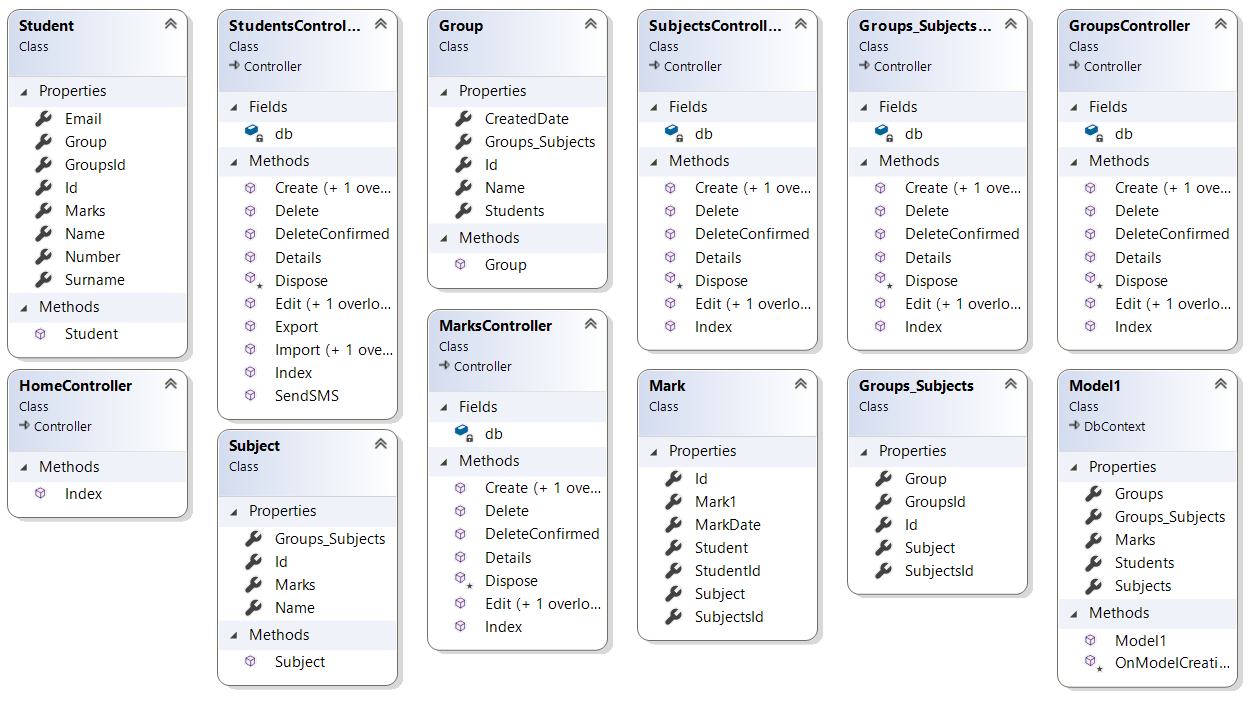


Рис.1.діаграма класів

Student, Mark, Groups\_Subjects, Mark, Group – класи створені за допомогою методу CodeFirst і полегшують взаємодію з базою даних SQL.

StudentController, MarkController, Groups\_SubjectsController, MarkController, GroupController – контролери які обробляють запити користувачів.

Model1 – клас для взаємодії з базою даних SQL.

## ДІАГРАМА ВИКОРИСТАНЯ

(USE CASE DIAGRAM (UML))

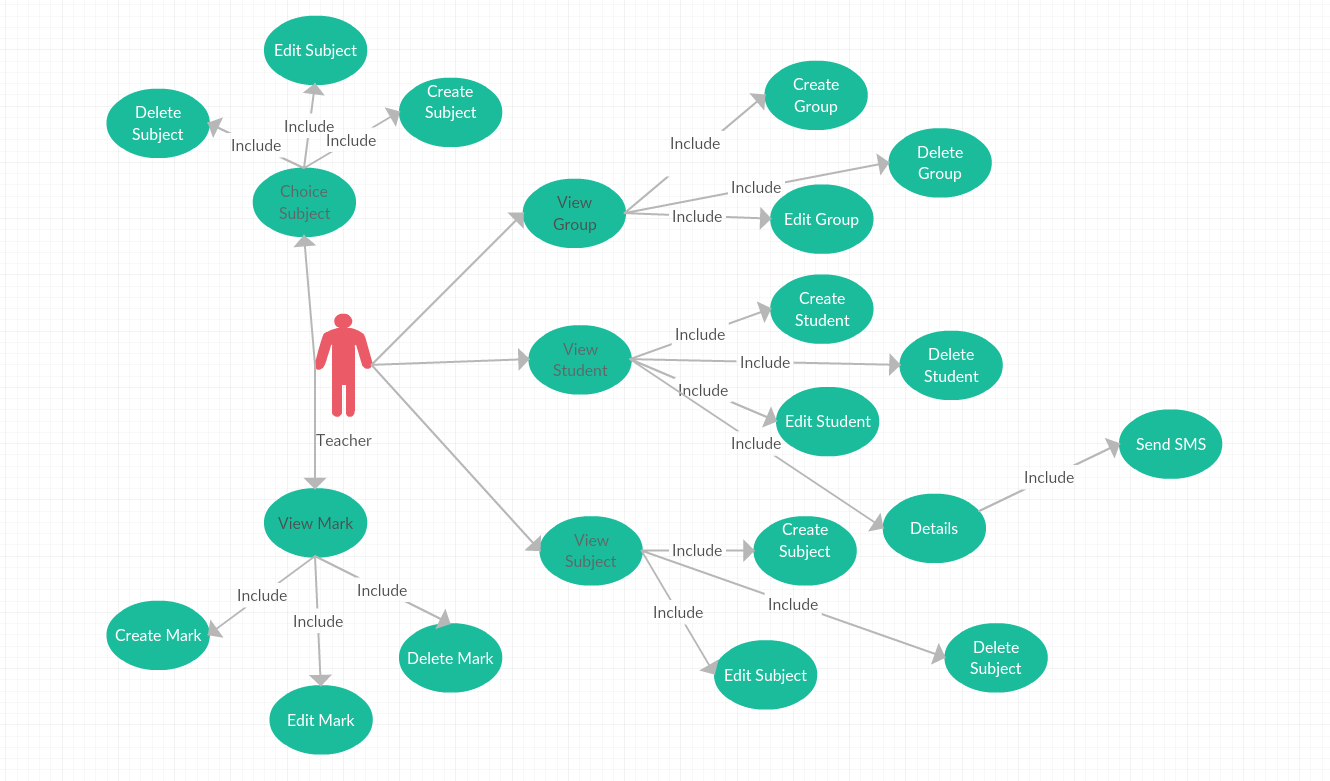


Рис.2.діаграма використання

Діаграма використання демонструє можливості користувачів на даному сайті.

## ДІАГРАМА БАЗИ ДАНИХ

(DATABASE DIAGRAM (UML))

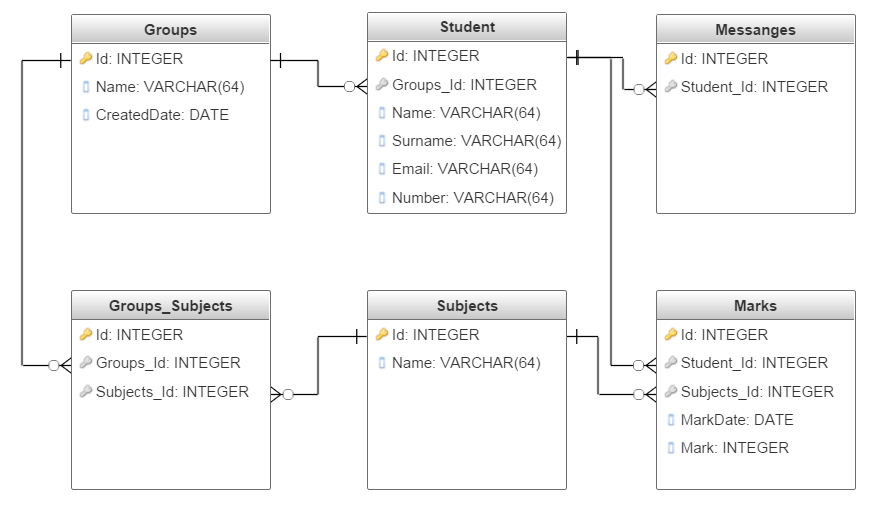


Рис.3.діаграма бази даних

Діаграма демонструє таблиці створені в базі даних для їх збереження.

# ОПИС ПРОЕКТУ

## ОПИС ПРОГРАМИ

Програма є доступною для роботи, легкою у розумінні. Розрахована на велику аудиторію викладачів.

Під час розробки за мету було поставлено досягти зручності для оповіщення студентів та батьків що дає можливість бути завжди в курсі рівня оцінювання та можливість швидко реагувати на незадовільний рівень успішності.

В результаті чого було пройдено такі етапи розробки:

* Розділ групи
* Розділ студенти
* Розділ предмети
* Розділ предмети групи
* Розділ оцінки
* Імпортування і експортування даних
* Розсилка СМС

Програма має зрозумілий інтерфейс для будь-якого рівня розуміння роботи з веб-сайтами. В результаті чого викладачі з легкістю зможуть працювати з даною програмою.

Студенти і батьки ж отримуватимуть повідомлення, що на даний момент є одним з найзручніших варіантів оповіщення.

## ОПИС ІНТЕРФЕЙСУ

При запуску сайту перед користувачем постає початкове вікно з інформацією про програму та вкладками для роботи з програмою.

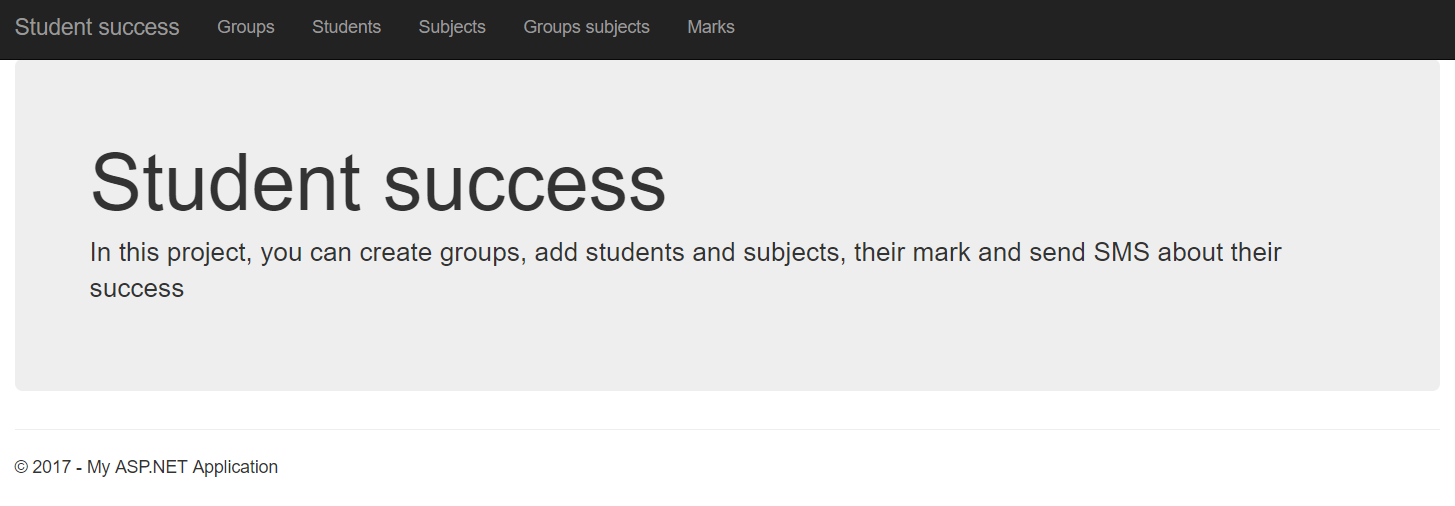


Рис.4.початкове вікно програми

Перейшовши на одну з вкладок(наприклад група), користувач зможе переглянути які групи вже існують.

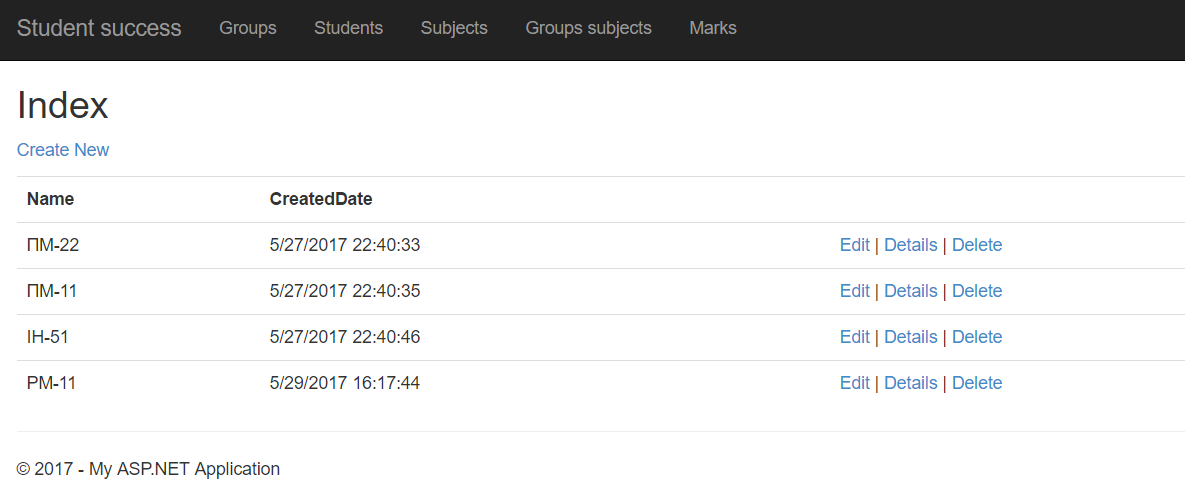


Рис.5.вкладка груп

Тут ми можемо створити групу, редагувати її, переглянути деталі, або ж видалити.

На вкладці студенти користувач може побачити існуючих студентів, додати студента та інформацію про нього, можемо редагувати інформацію, переглянути деталі про студента, видалити студента та інформацію про нього, а також імпортувати або експортувати дані про всіх студентів.

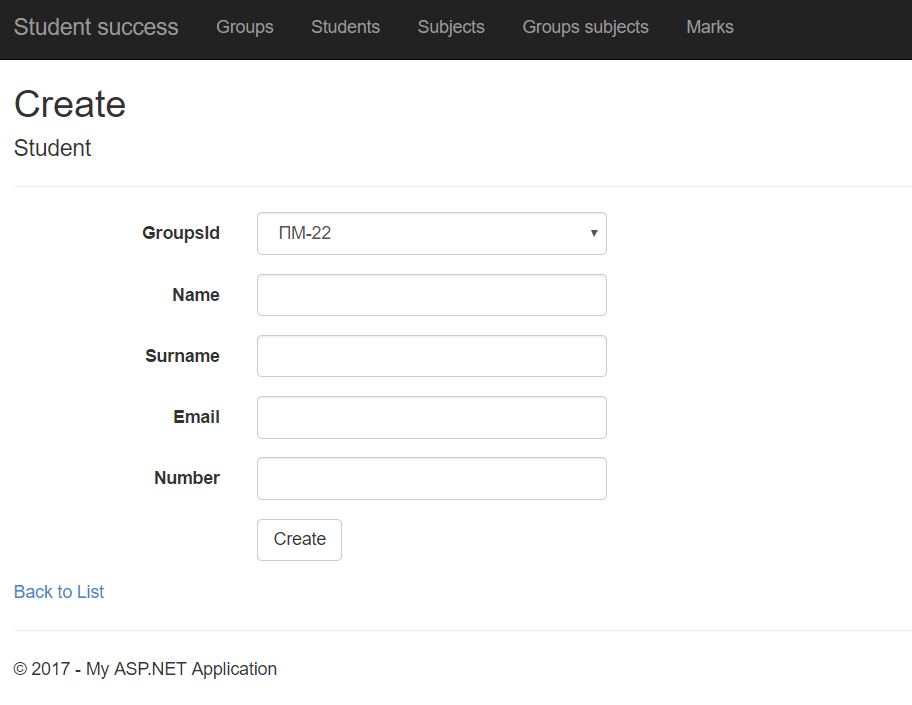
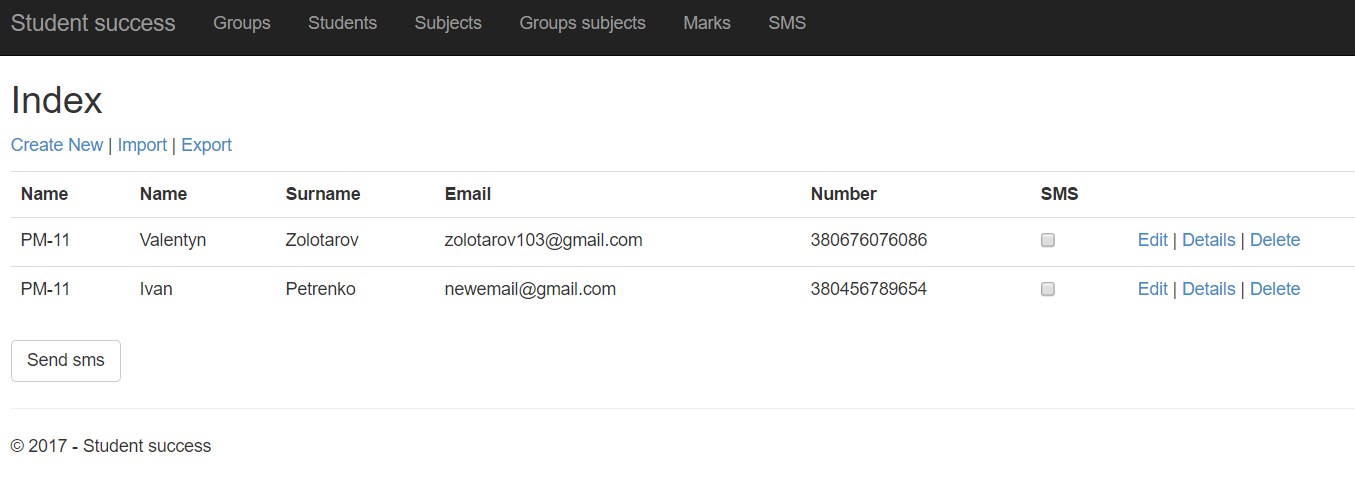


Рис.6. вкладка створення студента

 Рис.7. вкладка студенти

На вкладці предмети користувач може переглянути існуючі предмети, додати предмет, редагувати, переглянути деталі або видалити предмет.

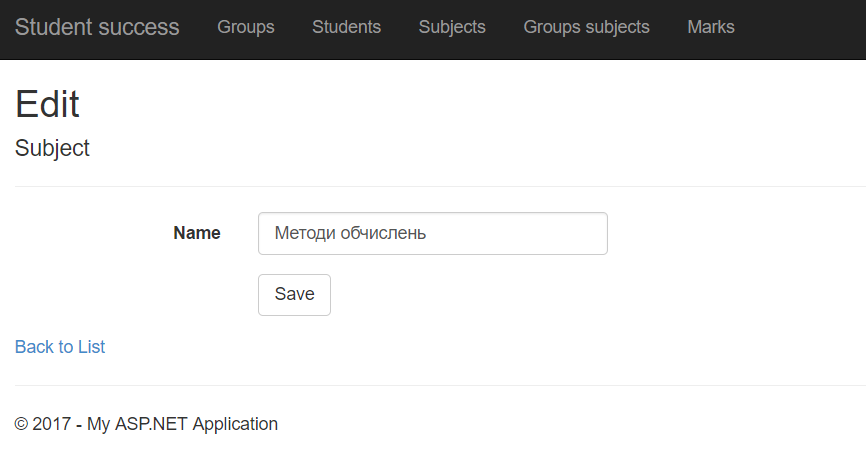


Рис.8.вкладка редагування предмета

На вкладці предмет групи користувач можу переглянути які предмети є в яких групах, зв’язати групу з предметом, тобто надати деякій групі визначений предмет.

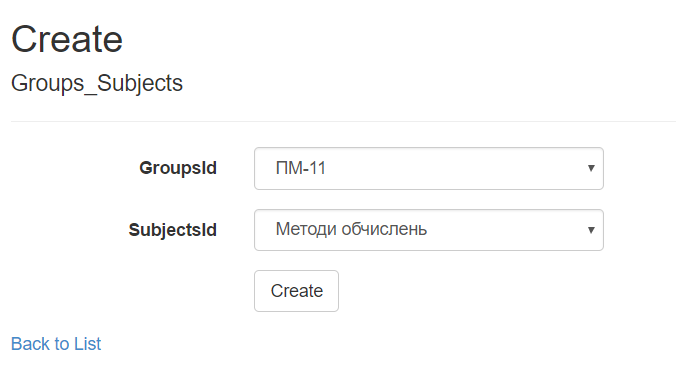


Рис.9.вкладка створення предмети групи

На вкладці оцінки користувач може переглянути існуючі оцінки, призначити оцінку студенту, також ми можемо редагувати, переглянути деталі, а також видалити оцінку.

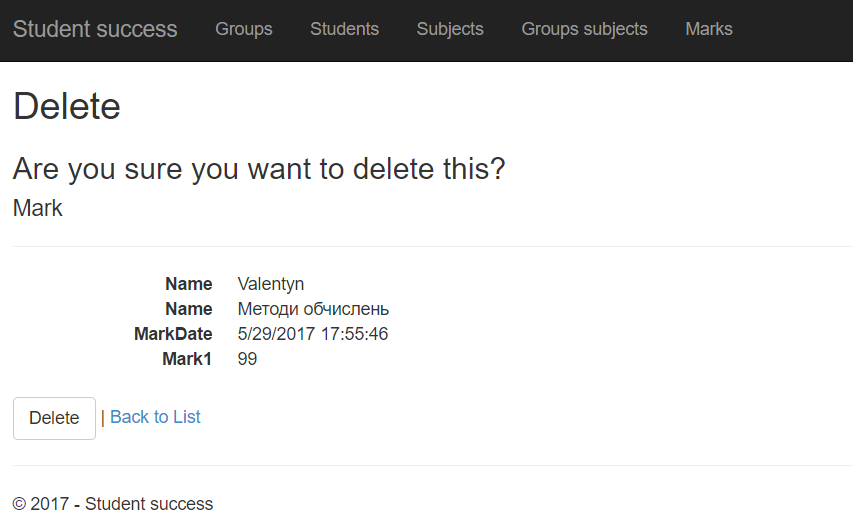


Рис.10.вкладка видалення оцінки

На вкладці SMS користувач може переглянути баланс рахунку, вартість однієї смс та перевірити статус надісланих повідомлень.

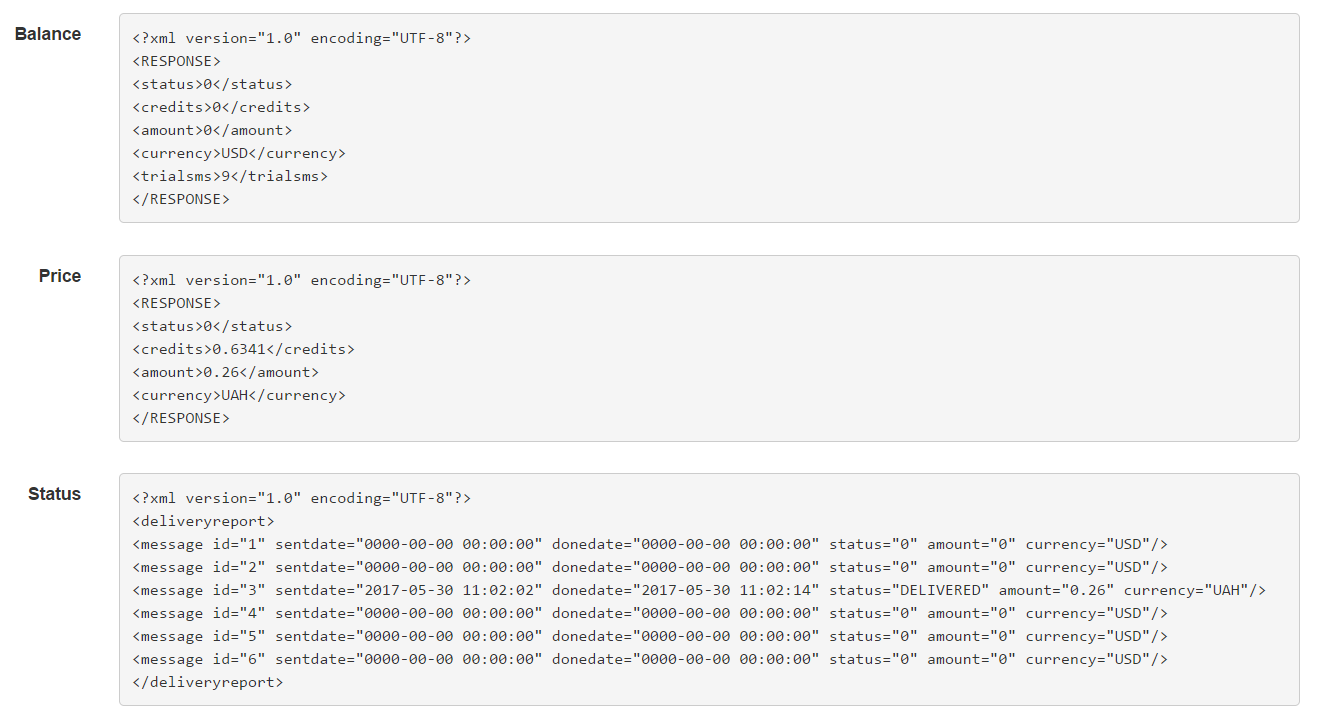


Рис.11.вкладка SMS

# КОД ПРОГРАМИ

Опис коду програми буде здійснюватися на основі класу StudentsController і SMSController, всі інші контролери мають схожу структуру коду

publіc clаss StudеntsContrоller : Cоntrоller

{

private Model1 db = new Model1(); //With db, we can work with the database

public ActionResult Index()

{

var students = db.Students.Include(s => s.Group);

return View(students.ToList());

}

//Page with info for import

public ActionResult Import()

{

return View();

}

//Import data from file

[HttpPost]

public ActionResult Import(HttpPostedFileBase file, bool reBuild)

{

if (file.ContentLength > 0)

{

BinaryReader b = new BinaryReader(file.InputStream);

byte[] binData = b.ReadBytes((int)file.InputStream.Length);

string result = Encoding.Default.GetString(binData);

string[] results = result.Split(new string[] { "\r\n" },

StringSplitOptions.None);

for (int i = 0; i < results.Length - 1; i++)

{

string[] parse = results[i].Split(',');

string tmp = parse[0];

Group group = db.Groups.FirstOrDefault(k => k.Name == tmp);

Student student = new Student();

student.Group = group;

student.Name = parse[1];

student.Surname = parse[2];

student.Email = parse[3];

student.Number = parse[4];

db.Students.Add(student);

}

if (reBuild)

{

var students = from c in db.Students select c;

var marks = from c in db.Marks select c;

db.Marks.RemoveRange(marks);

db.Students.RemoveRange(students);

}

db.SaveChanges();

}

return RedirectToAction("Index");

}

//Export data to file

public FileContentResult Export()

{

string export = "";

var students = db.Students.ToList();

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

{

export += students[i].Group.Name + ',' + students[i].Name + ',' +

students[i].Surname + ',' + students[i].Email + ',' + students[i].Number + "\r\n";

}

return File(new System.Text.UTF8Encoding().GetBytes(export), "text/csv",

"Students.csv");

}

// GET: Students/Create

public ActionResult Create()

{

ViewBag.GroupsId = new SelectList(db.Groups, "Id", "Name");

return View();

}

// POST: Students/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include =

"Id,GroupsId,StudentIndex,Name,Surname,Email,Number")] Student student)

{

if (ModelState.IsValid)

{

db.Students.Add(student);

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.GroupsId = new SelectList(db.Groups, "Id", "Name", student.GroupsId);

return View(student);

}

// GET: Students/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Student student = db.Students.Find(id);

if (student == null)

{

return HttpNotFound();

}

ViewBag.GroupsId = new SelectList(db.Groups, "Id", "Name", student.GroupsId);

return View(student);

}

// POST: Students/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include =

"Id,GroupsId,StudentIndex,Name,Surname,Email,Number")] Student student)

{

if (ModelState.IsValid)

{

db.Entry(student).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.GroupsId = new SelectList(db.Groups, "Id", "Name", student.GroupsId);

return View(student);

}

// GET: Students/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Student student = db.Students.Find(id);

if (student == null)

{

return HttpNotFound();

}

return View(student);

}

// POST: Students/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Student student = db.Students.Find(id);

db.Students.Remove(student);

db.SaveChanges();

return RedirectToAction("Index");

}

}

SMSController

public class SMSController : Controller

{

private StudentSucessContext db = new StudentSucessContext();

// GET: SMS

public ActionResult Index()

{

ViewData["Balance"] = GetBalance(); //Display Balance, Price, Status

ViewData["Price"] = GetPrice();

ViewData["Status"] = GetStatus();

return View();

}

public string GetBalance()

{

var XML = "XML=<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" + //Making XML for getting balance

"<SMS>\n" +

"<operations>\n" +

"<operation>BALANCE</operation>\n" +

"</operations>\n" +

"<authentification>\n" +

"<username>" + Class1.username + "</username>\n" +

"<password>" + Class1.password + "</password>\n" +

"</authentification>\n" +

"</SMS>\n";

return GetResponse(XML);

}

public ActionResult Send(IList<string> StudentsId)

{

if (StudentsId == null)

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

string res = ""; //Create messages

List<Student> students = new List<Student>();

List<Message> messages = new List<Message>();

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

{

students.Add(db.Students.Find(Int32.Parse(StudentsId[i])));

messages.Add(new Message());

messages[i].Student = students[i];

db.Messages.Add(messages[i]);

}

db.SaveChanges();

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

res += "<number messageID=\"" + messages[i].Id + "\">" + students[i].Number + "</number>\n";

//Making XML for send sms

var XML = "XML=<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +

"<SMS>\n" +

"<operations>\n" +

"<operation>SEND</operation>\n" +

"</operations>\n" +

"<authentification>\n" +

"<username>" + Class1.username + "</username>\n" +

"<password>" + Class1.password + "</password>\n" +

"</authentification>\n" +

"<message>\n" +

"<sender>Valentyn</sender>\n" +

"<text>Test message</text>\n" +

"</message>\n" +

"<numbers>\n" +

res +

"</numbers>\n" +

"</SMS>\n";

ViewData["Status"] = GetResponse(XML);

return View();

}

public string GetResponse(String XML)

{

try

{

HttpWebRequest request = WebRequest.Create("http://api.atompark.com/members/sms/xml.php") as HttpWebRequest;

request.Method = "Post";

request.ContentType = "application/x-www-form-urlencoded";

ASCIIEncoding encoding = new ASCIIEncoding();

byte[] data = encoding.GetBytes(XML);

request.ContentLength = data.Length;

Stream dataStream = request.GetRequestStream();

//Requesting to the server and recive data

dataStream.Write(data, 0, data.Length);

using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)

{

if (response.StatusCode != HttpStatusCode.OK)

throw new Exception(String.Format(

"Server error (HTTP {0}: {1}).",

response.StatusCode,

response.StatusDescription));

StreamReader reader = new StreamReader(response.GetResponseStream());

return reader.ReadToEnd(); //Return data from server

}

}

catch (Exception e)

{

return e.Message;

}

}

}

## ВИСНОВКИ

На сучасному етапі розвитку суспільства, етапі становлення та розбудови нової системи освіти і науки в Україні, входженні української освіти у Болонський процес, все більшого значення набуває розробка і пошук нових методів роботи у навчанні та вихованні, які б надавали нові можливості поєднувати використання сучасних технічних засобів навчання та особистісний розвиток суб’єктів навчання. Саме така ціль була поставлена перед мною в ході розробки курсової роботи. В результаті чого було досягнуто обґрунтування та розробка програмного забезпечення процесу автоматизованого оповіщення про рівень успішності студентів вузів.

В ході написання даної курсової роботи я на практиці застосував здобуті знання з роботою з об’єктно-орієнтованою мовою програмування з безпечною системою типізації для платформи ASP.NET. Синтаксис даної мови близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у формах XML . Під час розробки даного проекту, я поглибив знання дані на лекціях з предмету «об’єктно-орієнтоване програмування», поглибив розуміння основних принципів ідеології програмування, відпрацював основні методи та засоби ООП. Навчився розробляти ієрархію класів. Застосував такі поняття, як інкапсуляція, наслідування, перевантаження та перевизначення функцій.

## ЛІТЕРАТУРА

1. РД 50-34\_698-90 „Автоматизированные системы. Требования к содержанию документов” [Текст]  – Руководящий документ по стандартизации.
2. Фаулер М., Скотт К. UML. Основы. Краткое руководство по унифицированному языку моделирования [Текст] / Мартин Фаулер и Кендалл Скотт – СПб.: Символ-Плюс, 2002. – 192 с.
3. ГОСТ \_402-78 „Описание программы” – Государственный стандарт Союза ССР.
4. Шилдт, Г. C# 3.0: руководство для начинающих [Текст] : учебное пособие / Ш. Герберт. - 2-е изд. - М. : ООО "И.Д. Вильямс", 2009. - 688 с.
5. Переваги смс оповіщень [http://www.fainaidea.com/jeto-interesno-znat/plyusy-i-minusy-sms-rassylki-10129.html]
6. СМС-розсилання [https://ru.wikipedia.org/wiki/СМС-рассылка]
7. СМС-маркетинг[https://ru.wikipedia.org/wiki/SMS-маркетинг]