ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційно-комп’ютерних технологій

Кафедра Інженерія програмного забезпечення

Освітній рівень: бакалавр

Спеціальність: 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Зав. Кафедри ІПЗ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Панішев

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 р.

ЗАВДАННЯ

НА КУРСОВУ РОБОТУ СТУДЕНТУ

Верьовка Дмитро Вікторович

1. Тема роботи: Інформаційно-пошукова система «Міста».

керівник роботи: .Морозов Андрій Васильович

1. Строк подання студентом: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 р.
2. Вихідні дані до роботи: Розробити програму «Міста».
3. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)
4. Технічне завдання
5. Аналіз аналогічний розробок
6. Аналіз структурних і функціональних особливостей системи
7. Алгоритми роботи програми
8. Опис роботи програми
9. Програмне дослідження інформаційної системи складу\_\_\_\_\_\_\_\_ \_\_\_\_\_\_
10. Інструкція користувача \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
11. Перелік графічного матеріалу (з точним зазначенням обов‘язкових креслень)

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
| Завдання видав | Завдання прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання «\_\_» \_\_\_\_\_\_2018 р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № з\п | Назва етапів випускної роботи | Строк виконання етапів роботи | Примітка |
| 1 | Постановка задачі |  |  |
| 2 | Пошук, огляд та аналіз аналогічних розробок |  |  |
| 3 | Формулювання технічного завдання |  |  |
| 4 | Опрацювання літературних джерел |  |  |
| 5 | Проектування |  |  |
| 6 | Написання програмного коду |  |  |
| 7 | Відлагодження |  |  |
| 8 | Написання пояснювальної записки |  |  |
| 9 | Захист |  |  |

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_  (підпис) | Верьовка Д. В.  (прізвище та ініціали) |
| Керівник роботи | \_\_\_\_\_\_\_\_  (підпис) | **Морозов А.В.**  (прізвище та ініціали) |

# [РЕФЕРАТ](#_РЕФЕРАТ)

Курсовий проект, 30 с., 8 мал., 3 табл., 5 джерел, 3 додатків.

Дисципліна: “ОБЄКТНО ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ”

Тема: “ ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА «Міста» ”.

У проекті проаналізовані сучасні засоби та технологічні підходи до розробки інтерфейсу користувача прикладної програми, а також методи візуалізації даних та реалізації додаткових сервісів на прикладі системи обліку міст України.

В процесі розробки реалізовані функції, визначені завданням на курсовий проект.

Результат розробки оформлений у вигляді програмного пакету і комплекту супроводжувальної документації.

Для реалізації проекту було застосоване середовище розробки Microsoft Visual Studio 2015.

Практична цінність роботи полягає у демонстрації можливості підвищення якісних характеристик програми в області надійності обробки даних за рахунок розробки ефективного інтерфейсу користувача.

ЗМІСТ

[РЕФЕРАТ 3](#_Toc17190540)

[ВСТУП 5](#_Toc17190541)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ 6](#_Toc17190542)

[2 ОПИС СТРУКТУРИ ТА МЕТОДІВ ОБРОБКИ ДАНИХ 8](#_Toc17190543)

[2.1 Обґрунтування вибору методів та засобів обробки даних 8](#_Toc17190544)

[2.2 Опис структур даних 8](#_Toc17190545)

[3 ОПИС СТРУКТУРИ ТА АЛГОРИТМУ ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 10](#_Toc17190546)

[3.1 Модульна структура програмного засобу 10](#_Toc17190547)

[3.2 Інформаційна модель системи 11](#_Toc17190548)

[4 ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧА ТА СТРУКТУРИ ПРОГРАМНИХ МОДУЛІВ 12](#_Toc17190549)

[4.1 Опис програмних модулів 12](#_Toc17190550)

[4.1.1 Модуль “Misto ” 12](#_Toc17190551)

[4.1.2 Модуль “Pereglad” 13](#_Toc17190552)

[4.1.3 Модуль “ Users ” 14](#_Toc17190553)

[4.2 Опис інтерфейсу користувача 17](#_Toc17190554)

[ВИСНОВОК 18](#_Toc17190555)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 19](#_Toc17190556)

[ДОДАТКИ 20](#_Toc17190557)

[ДОДАТОК 1 20](#_Toc17190558)

[ДОДАТОК 2 23](#_Toc17190559)

[ДОДАТОК 3 27](#_Toc17190560)

# ВСТУП

Сучасне суспільство диктує все нові правила життя. Основою впровадження стали саме комп'ютерні технології. В наш час комп'ютери стали невід'ємною частиною нашого життя, які використовується всюди, це і стало приводом створення програмного забезпечення різного рівня складності та різного за галузями застосування. Однією із основних вимог до систем такого типу є надійність обробки даних і запобігання виникненню помилок на всіх етапах обробки інформації.

Людський фактор є однією із причин виникнення помилок, які мають ненавмисний характер та проявляються в процесі введення інформації, але можуть призвести до непередбачуваних наслідків. Помилки можуть бути пов’язані із неуважність, недостатнім рівнем кваліфікації користувача, невірним представленням даних у документах та іншим.

Підвищити якість обробки даних можна за рахунок організаційних заходів, наприклад, впровадженням уніфікованої системи підготовки даних та ведення документообігу, підвищенням кваліфікації персоналу, створенням комфортних умов для роботи, але більшої ефективності у даному напрямку можна досягнути за рахунок програмного запобігання виникнення помилок та контролю правильності введення даних.

Сучасні технології розробки інтерфейсу користувача надають різні можливості реалізації такого захисту. Аналіз існуючих засобів і технологічних підходів для рішення даної задачі, вибір із них найбільш доцільних, з наступною реалізацією на прикладі побудови реального інтерфейсу користувача для введення інформації у базу даних є основною метою даного курсового проекту.

В процесі виконання курсового проекту будуть розроблені додаткові елементи програмної системи, які розширюють її функціональність та підвищують якісні показники.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ

У відповідності до завдання на курсовий проект необхідно розробити інтерфейс введення та обробки даних для системи обліку Міст України.

Мета даного курсового проектування – спроектувати графічний інтерфейс користувача програмної системи із застосуванням сучасних технологій та інструментальних засобів, на основі методів структурного та об’єктно – орієнтованого підходу.

У якості об’єкту дослідження визначено міста. В результаті дослідження повинні бути визначені основні інформаційні об’єкти, їх характеристики, встановлені зв’язки між ними, джерела надходження даних та обмеження, що на них накладаються.

Предметом дослідження виступають засоби візуалізації даних та додаткові сервіси для забезпечення необхідної функціональності програмного проекту.

У якості методів дослідження обрано вивчення і узагальнення матеріалів із заданої предметної області, порівняльний аналіз існуючих технологічних та проектних підходів, інструментальних засобів реалізації програмного проекту.

Окрім вказаного в процесі виконання проекту потрібно розробити специфікацію вимог, які повинні бути реалізовані в процесі виконання проекту, описати загальні технологічні принципи та проектні рішення, структури і алгоритми обробки даних, визначити технології і методи їх обробки, розробити модульну структуру програми у відповідності до її функціональності, виконати проектування та кодування програмних модулів, розробити інтегровану довідкову систему та інсталяційний пакет.

Актуальність даної розробки полягає у тому, що для будь – якого інтерфейсу користувача, пов’язаного із обслуговуванням баз даних, є важливим контроль введення даних та запобігання помилкам операторів.

Контроль даних, що вводяться користувачем, на відповідність допустимому типу, діапазону значень, загальною довжиною тощо для даної операції, а також припинення спроб порушити його роботу шляхом введення свідомо невірної інформації.

Впровадження такої програмної системи дозволить автоматизувати у більшості випадків облік даних міст, за рахунок чого збільшити швидкість обробки інформації, зменшити кількість помилок при заповненні типових документів та підвищити якість їх оформлення, ефективно формувати статистичні та інші звіти із мінімальними витратами часу.

В результаті аналізу предметної області було встановлено:

Місто — це тип поселення, зазвичай значного за чисельністю та густотою населення, мешканці якого зайняті, як правило, поза сільським господарством. У багатьох країнах статус міста визначається і закріплюється законодавчо, при цьому може висуватися критерій чисельності населення.

Міста складається з назви, які пропонують користувачеві.

Обсяг, міст зумовлені насамперед можливостями матеріально-технічної бази і рівнем країни.

Як уже згадувалось, місто – це засіб розміщення людей, котрі перебувають на обліку. Основною з них є назва.

Зміст міста стосується:

* отримання назви.

Внаслідок використання програми користувачем, найголовніша функція стосується забезпечення збереження назв в достатній кількості.

При продажі товару вибирається товар і зазначається кількість яку було продано.

# 2 ОПИС СТРУКТУРИ ТА МЕТОДІВ ОБРОБКИ ДАНИХ

## **2.1 Обґрунтування вибору методів та засобів обробки даних**

Оскільки, у відповідності до технічного завдання і аналізу предметної області, розробка інтерфейсу введення і обробки даних для системи обліку товарів на складі проводиться на основі принципів та елементів систем управління базами даних, то логічним є - збереження даних здійснювати у вигляді файлів таблиць бази даних певного типу, вибір і обробку інформації виконувати на основі запитів, для введення і відображення даних використовувати діалогові екранні форми, а вихідні паперові документи формувати у вигляді звітів.

Однією із сучасних універсальних інструментальних систем програмування є Microsoft Visual Studio 2015., яка має розвинені засоби для програмування та обробки баз даних, і в той же час підтримує майже всі відомі формати даних, що при необхідності дозволить перевести створені файли даних на іншу програмну платформу.

Створення файлу бази даних та визначення його структури здійснюється за допомогою СУБД Access, яка входить до складу професійної версій офісного пакету Microsoft Office.

Розповсюджується як умовно-безкоштовне програмне забезпечення (англ. shareware), версія для платформи Pocket PC як безкоштовне програмне забезпечення.

**2.2 Опис структур даних**

Розглянемо структуру бази даних для заданої предметної області. Основними сутностями цієї структури є:

* Назва – характеризує найменування населеного пункту.

На Рис.2.1. наведено докладну ER – діаграму (сутність – зв’язок) системи за нотацією Баркера.

Нижче наведено структуру файлів таблиць бази даних.

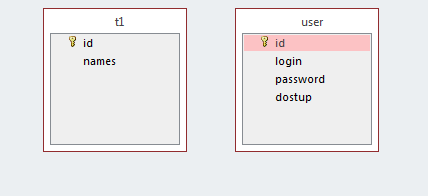


Рисунок 2.1 - ER - діаграма даних системи

t1 – таблиця у якій зберігається інформація про міста.

Структура полів файлу t1 наведено у таблиці 2.1.

Таблиця 2.1 - Структура полів таблиці t1

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип** | **Інформаційний зміст** |
| id | Лічильник | Номер міста |
| names | Текстовий | Назва міста |

User – таблиця у якій міститься інформація про постачальників.

Структура полів файлу user наведено у таблиці 2.3.

Таблиця 2.3 - Структура полів таблиці supplier

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип** | **Інформаційний зміст** |
| id | Лічильник | Номер користувача |
| login | Текстовий | Логін користувача |
| password | Текстовий | Пароль користувача |
| dostup | Текстовий | Доступ користувача |

**3 ОПИС СТРУКТУРИ ТА АЛГОРИТМУ ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

## **3.1 Модульна структура програмного засобу**

Програма складається з набору програмних модулів, кожний з яких виконує одну чи декілька функцій. Модульна структура програми зображена на рис. 3.1. У таблиці 3.1. наведено призначення програмних модулів.

Login

Pereglad

Users

Рисунок 3.1 – Модульна структура програмної системи

Таблиця 3.1 - Призначення програмних модулів

|  |  |  |
| --- | --- | --- |
| №  з/п | Позначення | Призначення |
| 1 | Misto | Головний модуль програми. Містить головну форму для авторизації у системі і забезпечує перехід з однієї форми на інші. |
| 2 | Pereglad | Модуль призначений для перегляду товарів. |
| 3 | Users | Модуль реєстрації користувачів у системі, та встановлення їхнього рівня доступу. |

## **3.2 Інформаційна модель системи**

Інформаційна модель відображує загальний склад об’єктів автоматизованої системи, інформаційні потоки та зв’язки між ними. Структура інформаційної моделі представлена у вигляді функціональної схеми, наведеної на рис. 3.2.

**Збереження та обробка даних про міста**

**Загальний список міст**

**Формування та обробка оперативних даних про міста**

**Інформація по обраному запиту**

**Введення та редагування даних про міста**

**Формування та обробка оперативних даних про міста**

**Оперативні дані**

**Реєстрація та обробка даних про користувачів системи**

**Додавання нових міст**

**TAB\_**

**ADMIN**

**TAB\_MISTO**

**Формування довідників**

Рисунок 3.2 – Функціональна схема програмної системи

**4 ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧА ТА СТРУКТУРИ ПРОГРАМНИХ МОДУЛІВ**

## **4.1 Опис програмних модулів**

### **4.1.1 Модуль “Misto ”**

Модуль **Misto** є головним модулем програми і призначений для ідентифікації користувачів та визначення їх прав доступу при вході у систему. Окрім цього модуль виконує перехід до іншого модуля програми.

Інтерфейс реалізації функції ідентифікації користувачів побудований на взаємодії із таблицею бази даних User, в цій таблиці зберігаються логіни і паролі користувачів, а також визначені для них права доступу.

Перехід на інший програмний модуль виконується з використанням програмного коду.

Екранна форма модуля в режимі конструювання зображена на рис. 4.1.

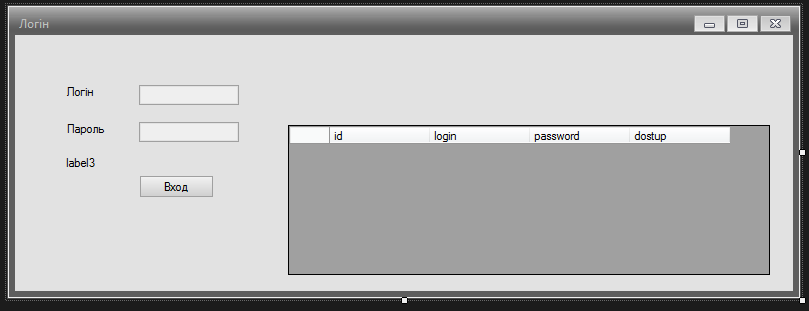


Рисунок 4.1 - Екранна форма модуля **Misto** в режимі конструювання

Програмна специфікація модуля **Misto** наведена у таблиці 4.1.

Таблиця 4.1 - Програмна специфікація модуля Misto

|  |  |  |
| --- | --- | --- |
| **Компонент програми** | **Позначення** | **Зміст** |
| **1** | **2** | **3** |
| Стандартні модулі | using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms; | Стандартні модулі, з якими встановлюється зв’язок |
| Візуальні компоненти | Label1 | Надпис «Логін» |
| Label2 | Надпис «Пароль» |
| Label3 | Надпис «Невірні дані» |
| TextBox1 | Поле для введення логінів |
| TextBox2 | Поле для введення паролю |
| Button1 | Кнопка авторизації у системі |
| Datagridview1 | Доступ до БД для перевірки логіну і паролю |
| Процедури | private void button1\_Click(object sender, EventArgs e) | Перевірка введених даних на існування, та авторизація у системі із встановленням функціоналу |
| public Form2() | Відкриття форми авторизації та розподілення функціоналу |

Програмний код модуля **Misto** наведено у Додатку 1

### **4.1.2 Модуль “Pereglad”**

Модуль **Pereglad** є модулем програми і призначений для додавання нових міст в систему. Окрім цього модуль виконує навігаційні функції для перегляду існуючих даних.

Інтерфейс реалізації функції додавання та обробки даних про міста, побудований на взаємодії із таблицею бази даних t1, в цій таблиці зберігаються данні про існуючі міста.

Екранна форма модуля в режимі конструювання зображена на рис. 4.2.

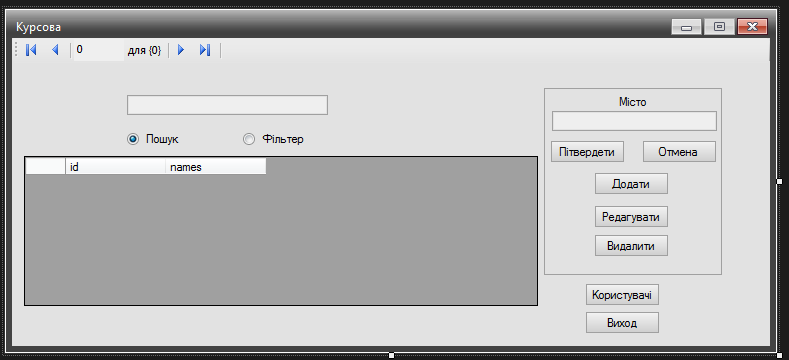


Рисунок 4.2 - Екранна форма модуля **Pereglad** в режимі конструювання

Програмна специфікація модуля **Pereglad** наведена у таблиці 4.2.

Таблиця 4.2 - Програмна специфікація модуля Pereglad

|  |  |  |
| --- | --- | --- |
| **Компонент програми** | **Позначення** | **Зміст** |
| **1** | **2** | **3** |
| Стандартні модулі |  | Стандартні модулі, з якими встановлюється зв’язок |
| Візуальні компонети | DatagriGview1 | Таблиця відображення записів міст |
| TextBox2 | Текстове поле призначене для пошукових запитів |
| RadioButton1, RadioButton2 | Кнопки для вибору «Пошук» або «Фільтр» |
| Процедури | private void textBox2\_TextChanged(object sender, EventArgs e) | Призначена для «Пошуку» або «Фільтрації» під час введення значення |

Програмний код модуля **Pereglad** наведено у додатку 2

продовження Таблиця 4.5

### **4.1.3 Модуль “** **Users ”**

Модуль Users є модулем програми і призначений для реєстрації та перегляду користувачів в системі, визначення їхнього рівня доступу.

Інтерфейс реалізації функції додавання та обробки даних про користувачів системи, побудований на взаємодії із таблицею бази даних Users, в цій таблиці зберігаються данні про існуючих користувачів системи та їхні рівні доступу.

Екранна форма модуля в режимі конструювання зображена на рис. 4.3.

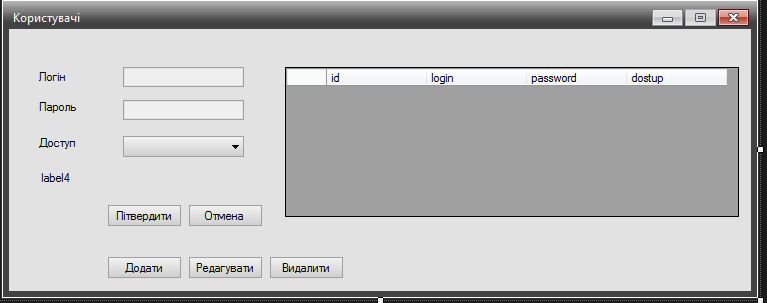


Рисунок 4.3 - Екранна форма модуля **Users** в режимі конструювання

Програмна специфікація модуля **Users** наведена у таблиці 4.3.

Таблиця 4.3 - Програмна специфікація модуля Users

|  |  |  |
| --- | --- | --- |
| **Компонент програми** | **Позначення** | **Зміст** |
| **1** | **2** | **3** |
| Стандартні модулі | using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms; | Стандартні модулі, з якими встановлюється зв’язок |
| Візуальні компоненти | Label1 | Текстовий напис «Логін» |
| Textbox1 | Текстове поле призначене для відображення та введення логінів користувачів |
| Label2 | Текстовий напис «Пароль» |
| Textbox2 | Текстове поле призначене для відображення та введення паролів користувачів |
| Label3 | Текстовий напис «Рівень доступу» |
| Combobox1 | Текстове поле призначене для вибору рівня доступу користувачів |
| Datagridview1 | Таблиця відображення записів користувачів |

Програмний код модуля Users наведено у додатку 3

## **4.2 Опис інтерфейсу користувача**

Програма представляється користувачеві у вигляді вікна зображеному на рисунку 4.4. Інтерфейс користувача являє собою форму з полями, вибором і текст, таблиці з іформацією про міста (користувачів) зображеному на рисунку 4.5. З можливість пошуку даних в таблицях.

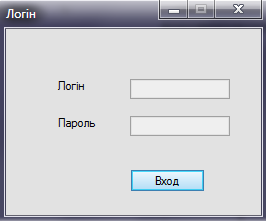


Рисунок 4.7 – Форма входу

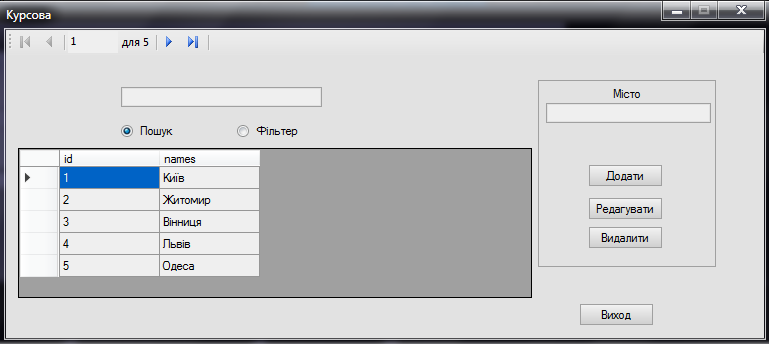


Рисунок 4.8 – Форма перегляду

# ВИСНОВОК

В результаті виконання курсового проекту було розроблено інтерфейс введення і обробки даних для системи обліку міст.

У відповідності до завдання здійснено аналіз предметної області, визначені інформаційні об’єкти та їх властивості, засоби та методи обробки даних, опис інтерфейсу та алгоритмів функціонування програми. Також реалізовані додаткові функції:

* контроль послідовності заповнення полів при введенні даних;
* контроль заповнення обов’язкових полів;
* контроль даних на відповідність заданому діапазону значень;
* введення даних шляхом підстановки значень;
* сортування та пошук даних за різними критеріями;
* доступ до функцій програми у відповідності до прав користувачів.

Подальший розвиток проекту можливий у напрямку покращення інтерфейсу та його інтеграції у існуючі WEB – технології із розробкою відповідних інтерфейсів і сервісів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Джон Шарп. Microsoft Visual C#. Подробное руководство. 8-е издание.
2. Д. Грин Э. Стиллмен. Изучаем C#.
3. Джеффри Р. Windows для профессионалов. Создание эффективных Win32-пpилoжeний с учетом специфики 64-разрядной версии Windows / Рихтер Джеффри. – Питер, 2001. – 752 с.
4. Эндрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5.
5. Герберт Шилдт. C# 4.0: полное руководство.

ДОДАТКИ

ДОДАТОК 1

Програмний код модуля **Sclad**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kursDataSet2.user". При необходимости она может быть перемещена или удалена.

this.userTableAdapter.Fill(this.kursDataSet2.user);

}

private void button1\_Click(object sender, EventArgs e)

{

if(textBox1.Text=="1111" & textBox2.Text=="1111" )

{

label3.Visible = true;

label3.Text = "Правельні дані.";

Form1 form1 = new Form1();

form1.button7.Visible = true;

form1.FormClosed += formClosed;

form1.Show();

this.Hide();

cle();

return;

}

if (dataGridView1.DataSource == userBindingSource)

{

userBindingSource.Filter = " [login] LIKE'" + textBox1.Text + "%'";

//где Искать - название столбца в DatagridView

}

string login, password;

int dostup = 0;

string passwords = textBox2.Text;

int index = dataGridView1.Rows.Count;

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

{

login = (string)dataGridView1.Rows[i].Cells[1].Value;

password = (string)dataGridView1.Rows[i].Cells[2].Value;

if (password == passwords)

{

dostup = (int)dataGridView1.Rows[i].Cells[3].Value;

break;

}

}

switch (dostup)

{

case 0:

label3.Visible = true;

label3.Text = "Помилкові дані. Не правильний логін або пароль!";

break;

case 1:

label3.Visible = true;

label3.Text = "Правельні дані.";

Form1 form1 = new Form1();

form1.button7.Visible = true;

form1.FormClosed += formClosed;

form1.Show();

this.Hide();

cle();

break;

case 2:

label3.Visible = true;

label3.Text = "Правельні дані.";

Form1 form1\_ = new Form1();

form1\_.FormClosed += formClosed;

form1\_.Show();

this.Hide();

cle();

break;

default:

label3.Visible = true;

label3.Text = "Помилкові дані. Не правильний логін або пароль!";

break;

}

}

void formClosed(object sender, FormClosedEventArgs e)

{

this.Show();

this.userTableAdapter.Fill(this.kursDataSet2.user);

}

void cle()

{

textBox1.Text = "";

textBox2.Text = "";

label3.Text = "";

}

}

}

ДОДАТОК 2

Програмний код модуля **Pereglad**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void kursDataSetBindingSource\_CurrentChanged(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kursDataSet1.t1". При необходимости она может быть перемещена или удалена.

this.t1TableAdapter.Fill(this.kursDataSet2.t1);

}

private void button1\_Click(object sender, EventArgs e)

{

string Wname;

Wname = textBox1.Text;

this.t1TableAdapter.Insert(Wname);

this.t1TableAdapter.Fill(this.kursDataSet2.t1);

textBox1.Text = "";

}

private void button2\_Click(object sender, EventArgs e)

{

int index;

string WName;

// отримати позицію виділеного рядка в dataGridView1

index = dataGridView1.CurrentRow.Index;

// отримати дані рядка

WName = (string)dataGridView1.Rows[index].Cells[1].Value;

// заповнити поля

textBox1.Text = WName;

textBox2.Enabled = false;

button1.Visible = false;

button2.Visible = false;

button3.Visible = false;

button4.Visible = true;

button5.Visible = true;

dataGridView1.Enabled = false;

bindingNavigator2.Enabled = false;

}

private void button3\_Click(object sender, EventArgs e)

{

int id\_worker;

string WName;

int index;

// взяти номер поточного (виділеного) рядка в dataGridView1

index = dataGridView1.CurrentRow.Index;

// заповнити внутрішні змінні з поточного рядка dataGridView1

id\_worker = Convert.ToInt32(dataGridView1[0, index].Value);

WName = Convert.ToString(dataGridView1[1, index].Value);

// сформувати інформаційний рядок

string massege = "Дійсно видалити місто "+ WName + " ?";

DialogResult dialogResult = MessageBox.Show(massege, "Видалити", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes)

{

this.t1TableAdapter.Delete(id\_worker, WName); // метод Delete

this.t1TableAdapter.Fill(this.kursDataSet2.t1);

}

else if (dialogResult == DialogResult.No)

{

//do something else

}

}

private void button4\_Click(object sender, EventArgs e)

{

int index;

string WName;

int id\_worker;

if (dataGridView1.RowCount <= 1) return;

// отримати позицію виділеного рядка в dataGridView1

index = dataGridView1.CurrentRow.Index;

if (index == dataGridView1.RowCount - 1) return; //

// отримати дані рядка

id\_worker = (int)dataGridView1.Rows[index].Cells[0].Value;

WName = (string)dataGridView1.Rows[index].Cells[1].Value;

string nWName;

// отримати нові (змінені) значення з форми

nWName = textBox1.Text;

string massege = "Дійсно змінити місто " + WName + " на "+ nWName +" ?";

DialogResult dialogResult = MessageBox.Show(massege, "Змінит", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes)

{

// змінити в адаптері

this.t1TableAdapter.Update(nWName, id\_worker, WName);

this.t1TableAdapter.Fill(this.kursDataSet2.t1);

textBox1.Text = "";

textBox2.Enabled = true;

dataGridView1.Enabled = true;

bindingNavigator2.Enabled = true;

button1.Visible = true;

button2.Visible = true;

button3.Visible = true;

button4.Visible = false;

button5.Visible = false;

}

else if (dialogResult == DialogResult.No)

{

//do something else

}

}

private void button5\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox2.Enabled = true;

dataGridView1.Enabled = true;

bindingNavigator2.Enabled = true;

button1.Visible = true;

button2.Visible = true;

button3.Visible = true;

button4.Visible = false;

button5.Visible = false;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

if (radioButton1.Checked == true)

{

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox2.Text))

{

dataGridView1.FirstDisplayedScrollingRowIndex = i;

dataGridView1.Rows[i].Selected = true;

break;

}

}

}

else

{

if (dataGridView1.DataSource == t1BindingSource)

{

t1BindingSource.Filter = " [names] LIKE'" + textBox2.Text + "%'";

//где Искать - название столбца в DatagridView

}

}

}

private void button6\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button7\_Click(object sender, EventArgs e)

{

Form3 form3 = new Form3();

form3.Show();

}

}

}

ДОДАТОК 3

Програмний код модуля **Users**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void Form3\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kursDataSet2.t1". При необходимости она может быть перемещена или удалена.

this.userTableAdapter1.Fill(this.kursDataSet2.user);

}

private void button1\_Click(object sender, EventArgs e)

{

int x = comboBox1.SelectedIndex;

if (x == -1)

{

label4.Visible = true;

label4.Text = "Не вибраний рівень доступу";

}

else

{

if (dataGridView1.DataSource == userBindingSource)

{

userBindingSource.Filter = " [login] LIKE'" + textBox1.Text + "%'";

//где Искать - название столбца в DatagridView

}

string logins;

string login = textBox1.Text;

bool isn = false;

int index = dataGridView1.Rows.Count;

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

{

logins = (string)dataGridView1.Rows[i].Cells[1].Value;

if (login == logins)

{

isn = true;

break;

}

}

if (isn == false)

{

string Wlogin, Wpassword;

int Wdostup;

Wlogin = textBox1.Text;

Wpassword = textBox2.Text;

Wdostup = comboBox1.SelectedIndex + 1;

this.userTableAdapter1.Insert(Wlogin, Wpassword, Wdostup);

this.userTableAdapter1.Fill(this.kursDataSet2.user);

textBox1.Text = "";

textBox2.Text = "";

comboBox1.SelectedIndex = -1;

label4.Visible = false;

}

else

{

label4.Visible = true;

label4.Text = "Користувач з таким логіном вже існує";

}

}

this.userTableAdapter1.Fill(this.kursDataSet2.user);

if (dataGridView1.DataSource == userBindingSource)

{

userBindingSource.Filter = " [login] LIKE'" + textBox1.Text + "%'";

//где Искать - название столбца в DatagridView

}

}

private void button2\_Click(object sender, EventArgs e)

{

int index;

string Wlogin, Wpassword;

int Wdostup;

if (dataGridView1.RowCount < 0) return;

// отримати позицію виділеного рядка в dataGridView1

index = dataGridView1.CurrentRow.Index;

// отримати дані рядка

Wlogin = (string)dataGridView1.Rows[index].Cells[1].Value;

Wpassword = (string)dataGridView1.Rows[index].Cells[2].Value;

Wdostup = (int)dataGridView1.Rows[index].Cells[3].Value;

// заповнити поля

textBox1.Text = Wlogin;

textBox2.Text = Wpassword;

comboBox1.SelectedIndex = Wdostup - 1;

button1.Visible = false;

button2.Visible = false;

button3.Visible = false;

button4.Visible = true;

button5.Visible = true;

dataGridView1.Enabled = false;

}

private void button3\_Click(object sender, EventArgs e)

{

int id\_worker;

string Wlogin, Wpassword;

int Wdostup;

int index;

// взяти номер поточного (виділеного) рядка в dataGridView1

index = dataGridView1.CurrentRow.Index;

// заповнити внутрішні змінні з поточного рядка dataGridView1

id\_worker = Convert.ToInt32(dataGridView1[0, index].Value);

Wlogin = Convert.ToString(dataGridView1[1, index].Value);

Wpassword = Convert.ToString(dataGridView1[2, index].Value);

Wdostup = Convert.ToInt16(dataGridView1[3, index].Value);

// сформувати інформаційний рядок

string massege = "Дійсно видалити користувача " + Wlogin + " ?";

DialogResult dialogResult = MessageBox.Show(massege, "Видалити", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes)

{

this.userTableAdapter1.Delete(id\_worker, Wlogin, Wpassword , Wdostup); // метод Delete

this.userTableAdapter1.Fill(this.kursDataSet2.user);

}

else if (dialogResult == DialogResult.No)

{

//do something else

}

}

private void button4\_Click(object sender, EventArgs e)

{

int index;

string Wlogin, Wpassword;

int Wdostup;

int id\_worker;

if (dataGridView1.RowCount <= 0) return;

// отримати позицію виділеного рядка в dataGridView1

index = dataGridView1.CurrentRow.Index;

// if (index == dataGridView1.RowCount - 1) return; //

// отримати дані рядка

id\_worker = (int) dataGridView1.Rows[index].Cells[0].Value;

Wlogin = (string) dataGridView1.Rows[index].Cells[1].Value;

Wpassword = (string) dataGridView1.Rows[index].Cells[2].Value;

Wdostup = (int) dataGridView1.Rows[index].Cells[3].Value;

string nWlogin, nWpassword;

int nWdostup;

// отримати нові (змінені) значення з форми

nWlogin = textBox1.Text;

nWpassword = textBox2.Text;

nWdostup = comboBox1.SelectedIndex + 1;

string massege = "Дійсно зробити змінити користувача " + Wlogin + " ?";

DialogResult dialogResult = MessageBox.Show(massege, "Змінит", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes)

{

// змінити в адаптері

this.userTableAdapter1.Update(nWlogin, nWpassword, nWdostup, id\_worker, Wlogin, Wpassword, Wdostup);

this.userTableAdapter1.Fill(this.kursDataSet2.user);

textBox1.Text = "";

textBox2.Text = "";

comboBox1.SelectedIndex = -1;

dataGridView1.Enabled = true;

button1.Visible = true;

button2.Visible = true;

button3.Visible = true;

button4.Visible = false;

button5.Visible = false;

}

else if (dialogResult == DialogResult.No)

{

//do something else

}

}

private void button5\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox2.Text = "";

comboBox1.SelectedIndex = -1;

dataGridView1.Enabled = true;

button1.Visible = true;

button2.Visible = true;

button3.Visible = true;

button4.Visible = false;

button5.Visible = false;

}

}

}