

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
Факультет механтроніки та комп'ютерних наук
Кафедра комп'ютерних наук

Дипломна магістерська робота

на тему: Розробка автоматизованої системи обліку відвідування
занять студентами навчального закладу

Виконав: студент групи МгІТ-21
спеціальності 122 Комп'ютерні науки
освітньої програми Комп'ютерні науки

Анна МАЄВСЬКА

Керівник к.т.н., доц. Тетяна ДЕМКІВСЬКА

Рецензент д.т.н. проф. Віктор ЧУПРИНКА

Київ 2022

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет механіки та комп'ютерних наук

Кафедра комп'ютерних наук

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

_____ Володимир Щербань

“ _____ ” _____ 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНУ МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Маєвській Анні Олексіївні

1. Тема роботи Розробка автоматизованої системи обліку відвідування занять студентами навчального закладу

Науковий керівник роботи Демківська Тетяна Іванівна, к.т.н., доцент

затверджені наказом вищого навчального закладу від «28» вересня 2022 року № 286

2. Строк подання студентом роботи 12.11.2022 р.

3. Вихідні дані до роботи Розробка кафедри комп'ютерних наук.

4. Зміст дипломної роботи (перелік питань, які потрібно розробити)

РОЗДІЛ 1. Постановка задачі; РОЗДІЛ 2. Алгоритмічне забезпечення життєвого циклу програми; РОЗДІЛ 3. Розробка системи обліку; презентація дипломної магістерської роботи з основними результатами дослідження (в роздрукованому вигляді представлена у додатках).

5. Консультанти розділів дипломної магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Вступ	к.т.н., доц. Тетяна ДЕМКІВСЬКА		
Розділ 1	к.т.н., доц. Тетяна ДЕМКІВСЬКА		
Розділ 2	к.т.н., доц. Тетяна ДЕМКІВСЬКА		
Розділ 3	к.т.н., доц. Тетяна ДЕМКІВСЬКА		
Висновки	к.т.н., доц. Тетяна ДЕМКІВСЬКА		

6. Дата видачі завдання 10.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів дипломної магістерської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	02.10.2022	
2	Розділ 1 Постановка задачі	18.10.2022	
3	Розділ 2 Алгоритмічне забезпечення життєвого циклу програми	30.10.2022	
4	Розділ 3 Розробка системи обліку	14.11.2022	
5	Висновки	24.11.2022	
6	Оформлення дипломної магістерської роботи (чистовий варіант)	31.10.2022	
7	Здача дипломної магістерської роботи на кафедру для рецензування (за 14 днів до захисту)	01.11.2022	
8	Перевірка дипломної магістерської роботи на наявність ознак плагіату (за 10 днів до захисту)	04.11.2022	
9	Подання дипломної магістерської роботи на затвердження завідувачу кафедри (з 7 днів до захисту)	07.11.2022	

Студент

Анна МАЄВСЬКА

Науковий керівник роботи

Тетяна ДЕМКІВСЬКА

Директор НМЦУПФ

- _____

Олена ГРИГОРЕВСЬКА

АНОТАЦІЯ

Маєвська А.О. Розробка автоматизованої системи обліку відвідування занять студентами навчального закладу.

Дипломна магістерська робота за спеціальністю 122 - «Комп'ютерні науки» – Київський національний університет технологій та дизайну, Київ, 2022 рік.

Метою даного дослідження є розробка програмного продукту, що допомагає вести облік відвідування студентами навчальних закладів. В роботі проаналізовано діяльність вищих навчальних закладів та розглянуто сучасні засоби розробки автоматизованих систем обліку відвідування занять студентами.

Бази даних використовуються для зберігання інформації про студентів, кафедри, спеціальності, дати відвідування занять. У той же час комп'ютерна техніка, яка використовується для вирішення тієї чи іншої задачі, дозволяє проаналізувати велику кількість прийнятних варіантів і вибрати найкращий.

Підсумовуючи, я розумію актуальність дослідження роботи та її розробки.

Методом дослідження є графічний інтерфейс до бази даних, який розроблено в середовищі Microsoft Visual Studio та системи керування базами даних Borland Database Engine. Ці продукти дозволяють користувачам розробляти консольні програми, програми графічного інтерфейсу користувача та підтримка технології "Windows Forms".

Ключові слова: автоматизована система, автоматизоване робоче місце, база даних, інформаційна система, система обліку, графічний інтерфейс.

A N N O T A T I O N

Majevska A.O. Automated system for recording attendance of classes by students of an educational institution.

Master's thesis in specialty 122 - "Computer Science" - Kyiv National University of Technology and Design, Kyiv, 2022.

The purpose of this study is the development of a software product that helps to keep track of students' visits to educational institutions. The work analyzes the activities of higher educational institutions and considers modern means of developing automated systems for recording attendance of classes by students.

Databases are used to store information about students, departments, majors, dates of class attendance. At the same time, computer technology, which is used to solve this or that problem, allows you to analyze a large number of acceptable options and choose the best one.

In summary, I understand the relevance of job research and its development.

The research method is a graphical interface to the database, which is developed in the Microsoft Visual Studio environment and the Borland Database Engine database management system. These products allow users to develop console applications, graphical user interface applications, and support for Windows Forms technology.

Keywords: automated system, automated workplace, database, information system, accounting system, graphical interface.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	9
Постановка завдання.....	9
Огляд інформаційних джерел та існуючих рішень	9
Обґрунтування мети та визначення завдань	11
РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ	15
Модель життєвого циклу розробки програмного продукту	15
Реалізація програмного забезпечення.....	17
Дослідження роботи баз даних	19
Вибір засобів для документування програмних продуктів	22
Створення інформаційної бази та таблиць.....	22
Створення блок – схем.....	25
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ОБЛІКУ	28
Розробка алгоритмів функціонування системи	28
Програмна реалізація.....	31
Програмне та апаратне забезпечення.....	37
Техніка безпеки при роботі з розробленою системою	38
ВИСНОВКИ.....	40
ЛІТЕРАТУРА.....	41
ДОДАТКИ.....	44

ВСТУП

Актуальність теми. Автоматизація підприємств і організацій дає можливість підвищити ефективність їх роботи і зберегти працездатність співробітників. Розробка автоматизованої інформаційної системи обліку відвідування навчальних закладів є актуальною проблемою сьогодення, оскільки полегшує роботу працівників будь-якого навчального закладу.

Викладач вищого навчального закладу – це особа, яка займається навчальною, методичною, науковою (науково-технічною, художньою) або організаційною діяльністю залежно від основного місця роботи вищого навчального закладу.

Завдання викладача – розкрити потенціал студента, максимально розкрити його здібності і навчити його реалізувати це на практиці.

В рамках даної магістерської роботи моєю задачею є розробка автоматизованої інформаційної системи обліку відвідування занять студентами будь-якого вищого навчального закладу.

Тому метою дослідження є розробка програмного продукту, який допоможе вести облік відвідування учнями навчальних закладів.

У даній магістерській роботі розроблено базу даних вищих навчальних закладів. Ця база даних необхідна для автоматизації роботи всіх викладачів навчального закладу з контролю за відвідуванням учнями та зберігання цих даних. Також розроблено автоматизоване робоче місце для обліку відвідуваності студентів, а також окреме робоче місце для перегляду відвідування лише навчальних курсів. Проаналізовано деталі діяльності вищих навчальних закладів та розглянуто сучасні засоби розробки автоматизованих систем обліку відвідування студентами занять.

Підсумовуючи, я розумію актуальність дослідження відвідування навчальних закладів та розробки системи автоматичного обліку студентів будь-якого навчального закладу з використанням сучасних інтернет-технологій.

Методом дослідження є графічний інтерфейс до бази даних, який у свою чергу розроблено в середовищі "Microsoft Visual Studio" та системі керування базами даних Borland Database Engine. Ці продукти дозволяють користувачам

розробляти консольні програми, програми з графічним інтерфейсом, які підтримують технологію «Windows Forms».

Мета дослідження. Провести аналіз предметної області та розробити програмний продукт, на основі інтерфейсу програмування додатків «Windows Form». За допомогою даного програмного засобу викладачі будуть вести облік студентів та, за потреби, контролювати їх відвідування в реальному часі. Система дозволяє вводити дані про нові групи та їх студентів, переглядати дані про викладачів та студентів, видаляти дані про студентів. Кожен учитель має спільні права доступу для всіх вчителів, що дозволяє їм додавати та видаляти дані, а також переглядати інформацію, введену в базу даних.

Завдання дослідження. Завданням дослідження є вибір середовища і технологій для розробки програмного продукту на основі «Windows Form», також проектування та реалізація програмного продукту, який полегшує роботу відповідальних осіб за відвідування навчального закладу студентами.

Об'єкт дослідження. Забезпечення безпеки даних та конфіденційності інформації за рахунок їх зберігання на сервері. **Предметом дослідження** є структура та реалізація системи обліку.

Практична цінність. Розробка програмного продукту для автоматизованої системи обліку відвідування занять та його самостійна робота без додаткових програм.

Елементи наукової новизни. Розробка програмного продукту на основі програмного продукту у додатку «Windows Form», за допомогою якого можна вести облік відвідування занять та зменшити навантаження на відповідальних осіб.

Практична значущість роботи. Додаток являє собою сервіс, що надає користувачу можливість доступу в різних місцях на різних пристроях, якщо попередньо даний додаток було на них встановлено.

Апробація результатів роботи. Система представлена у вигляді віконного додатку, система має простий у використанні інтерфейс і широкий набір функцій, що дозволяє швидко знайти потрібну інформацію.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Постановка завдання

Завданням магістерської роботи є створення програмного продукту, для ведення обліку відвідування студентами занять в навчальному закладі, проведення аналізу за темою магістерського проєкту, ознайомлення з особливостями ведення обліку відвідування учбових закладів.

Також розглянути існуючі бази даних відвідуваності та дослідити принципи їх роботи та навчання для їх побудови та необхідного функціонування. Розглянути можливість розробки сучасних програмних засобів для систем обліку відвідуваності та дослідити створення бази даних.

Огляд інформаційних джерел та існуючих рішень

Інформаційний процес – це багатоетапний процес, який включає: створення, збір, обробку, зберігання, пошук, передачу, розповсюдження, споживання, отримання, звітування, використання та знищення інформації або даних.

Джерелом інформації зазвичай є об'єкт або суб'єкт, який може надати інформацію. Це може бути будь-який матеріал, з якого отримано певну інформацію чи дані. Наприклад, база даних.

Інформаційна система – це система, в якій здійснюється обробка інформації або в якій відбувається обробка інформації. Інформаційний процес включає пошук, збір, зберігання, передачу, обробку тощо інформації або даних. В інформаційній системі може відбуватися один та/або більше процесів.

Кожна інформаційна система повинна включати три основні функціональні частини:

- Модуль обробки даних;
- Модуль зберігання даних;
- Інтерфейс користувача.

Модель предметної області (SAM) є ключовим компонентом успішно розробленої програми сховища даних або архітектури даних. Як правило, коли

створюються моделі предметної області, вони зазвичай використовуються лише для сегментації моделі даних.

Модель предметної області найефективніше визначити як визначений бізнесом сегмент, який охоплює області даних високого рівня рішення, необхідного для його створення. Хоча найчастіше використовується для визначення домену даних рішення сховища даних, у новій організації або організації, що розробляє програму формальної архітектури даних, його слід використовувати як основу для відображення всіх даних у всій організації.

Основна мета полягає в тому, щоб пов'язані з ним терміни та визначення були орієнтованими на бізнес і зрозумілими з першого погляду.

Модель сутність-зв'язок — це модель, результатом якої є систематичний процес, який описує та визначає предметну область. Він не визначає сам процес, він лише візуалізує його.

ER-моделі зазвичай реалізуються у формі баз даних. У випадку реляційних баз даних дані зберігаються в таблицях, кожен рядок кожної таблиці представляє екземпляр сутності. Деякі поля даних у цих таблицях вказують на індекси в інших таблицях. Ці поля є індикаторами фізичної реалізації зв'язків між сутностями.

Атрибут — це будь-яка характеристика сутності, яка є критичною для предметної області, що аналізується, і призначена для кваліфікації, класифікації, ідентифікації, кількісної оцінки або відображення стану сутності. Властивість представляє характеристику або властивість, пов'язану з набором реальних або абстрактних об'єктів. Екземпляр сутності повинен мати єдине визначене значення для пов'язаної властивості.

Кожна сутність повинна мати мінімальний набір унікальних властивостей, який називається первинним ключем.

Програмне забезпечення, яке зберігається на сервері, відповідає за збереження даних. На стороні клієнта зберігається вся інформація про інтерфейс, і обробка даних повинна бути розподілена між клієнтською та серверною частинами.

Інформаційна система обліку студентів для вищих навчальних закладів — програмний засіб, призначений для обліку навчальних занять, перегляду, додавання та редагування даних про студентів, групи та відвідування за датами.

Система обліку передбачає:

- Різні рівні доступу. Студентам групи надається обмежений доступ лише для перегляду відвідуваності навчального курсу своєї групи, викладачам надається розширений доступ, а викладачам надаються всі дозволи та доступ до даних;
- Ведення бази даних: додавання, оновлення, видалення, редагування інформації;
- Повідомлення про результати підсумкового контролю на основі робочих сесій.

Використання автоматизованих інформаційних систем може спростити облік відвідування студентами викладачів будь-якого вищого навчального закладу. Крім того, цей спосіб більш надійний і практичний у використанні, а також підтримує конфіденційність інформації.

Обґрунтування мети та визначення завдань

Дійшовши висновку, що автоматизація робочого місця є невід'ємною частиною діяльності агентства, стало зрозумілим, що така автоматизація звільнить співробітників від повсякденних завдань, пов'язаних із веденням журналу та обліком, які забирають у них багато часу.

Метою дипломного проєкту є створення автоматизованої інформаційної системи відвідування вищих учбових закладів, за допомогою якої викладачі будуть вести облік студентів та, за потреби, контролювати їх відвідування в реальному часі. Крім того, система дозволить вводити дані про нові групи та їх студентів, переглядати дані про викладачів та студентів, видаляти дані про студентів. Кожен учитель має спільні права доступу для всіх вчителів, що дозволяє їм додавати та видаляти дані, а також переглядати інформацію, введену в базу даних.

В системі можливий пошук такої інформації:

- стосовно будь-якого студента чи учня, який навчається у вищому навчальному закладі;
- про відвідування студентами або учнями занять;
- за назвою навчальної групи чи класу;
- За рівнем освіти.

У цій автоматизованій інформаційній системі можна роздрукувати звітну інформацію формату А4. Є всі необхідні поля, заповнені новими даними про студента або учня. Також створюється окремий користувач – студент або учень, який може перевіряти доступ лише до своєї групи чи класу.

За допомогою спеціальних методів та інструментів можна отримати доступ до редагованої інформації про відвідування учнями або студентами аудиторій навчальних закладів та швидко зберегти зміни.

Всі дані зберігаються в базі даних.

База даних — це насамперед набір таблиць, але також є процедури, представлення, тригери та багато інших об'єктів, які є не менш важливі. Таблицю бази даних можна розглядати як двовимірну таблицю з певною кількістю властивостей об'єкта.

Ідентифікатор — це елемент у таблиці, на який можна посилатися. Стовпці таблиці відповідають різним характеристикам об'єктів - полів. Кожне поле можна охарактеризувати назвою та типом збережених даних. Тип поля характеризує тип даних, що зберігаються в полі. Це можуть бути рядки, стовпці, числа, функції, логічні значення, формули, великі текстові повідомлення, зображення тощо.

Кожен рядок таблиці відповідає об'єкту. Він називається записом і містить значення всіх полів, що характеризують об'єкт.

Ключові поля в таблиці не обходяться, тому інформація між різними таблицями не конфліктує. Оскільки база даних може містити багато таблиць, дані в ній можуть дублюватися та суперечити один одному. Тому введіть ключові поля, щоб підтвердити унікальність кожного запису. У пов'язаній таблиці зазвичай одна є основною, а інша – допоміжною, якою керує основна

таблиця. Первинна і вторинна таблиці з'єднані між собою ключами. Деякі поля, які існують в обох таблицях, можна використовувати як ключі.

База даних містить дев'ять таблиць: Відвідування, Групи, Групи_связь, Дисципліни, Кафедри, Спеціальності, Студенти, Ступінь_Освіти, Факультети. Ці форми містять інформацію про студентів вищих навчальних закладів. Окремі таблиці корисні, але більше інформації можна отримати з сукупності пов'язаних таблиць, тому деякі з цих таблиць використовуються лише для належного зв'язку в базі даних.

Сутність у базі даних — це будь-який об'єкт у базі даних, який можна диференціювати відповідно до характеру предметної області, у якій розробляється база даних. Розробники баз даних повинні вміти правильно визначати сутності.

Бази даних можна створювати за допомогою різних програмних засобів: Oracle Database Standard Edition, Microsoft Office Access, Microsoft SQL Server, IBM DB2.

Oracle Database Standard Edition 2 — доступна, повнофункціональна база даних, створена для безпрецедентної простоти використання, потужності та продуктивності.

Oracle Database Standard Edition 2 містить усі функції, необхідні для розробки робочих груп, відділів і веб-додатків. Oracle Database Enterprise Edition забезпечує продуктивність, доступність, масштабованість і безпеку для розробки додатків, таких як додатки для обробки великих обсягів онлайн-транзакцій (OLT), сховища даних і вимогливі Інтернет-додатки. Oracle Database Personal Edition підтримує середовище розробки та розгортання для одного користувача. Oracle Database Express Edition легко встановлювати та керувати, що дозволяє розробляти, розгортати та поширювати програми. Продукти Oracle Database Standard Edition орієнтовані на організації середнього розміру або відділи великих організацій (робочі групи). "персональний Oracle " призначений для особистого використання.

Db2 — це сімейство продуктів керування даними, розроблене IBM, включаючи сервери баз даних. Спочатку вони підтримують реляційну модель,

але були розширені для підтримки об'єктно-реляційних функцій і нереляційних структур, таких як JSON і XML.

База даних DB2 — це реляційна база даних, яка забезпечує розширене керування даними та аналітичні можливості для транзакційних робочих навантажень. Розроблена для високої продуктивності, ефективності даних, доступності та надійності даних, ця операційна база даних підтримує операційні системи Linux, Unix і Windows.

Програмне забезпечення бази даних Db2 включає розширені функції, такі як технологія in-memory (IBM BLU Acceleration), розширені інструменти керування та розробки, оптимізація зберігання, керування робочим навантаженням, ефективне стиснення та постійна доступність даних (IBM pureScale).

Важливою особливістю комп'ютерних програм Db2 є обробка помилок. Раніше платформа SQL Communication Area (SQLCA) була призначена для Db2, щоб повертати програмі інформацію про помилки після виконання кожного оператора SQL. Основна, але не особливо корисна діагностика помилок виконується в полі SQLCODE блоку SQLCA.

РОЗДІЛ 2. АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ЖИТТЄВОГО ЦИКЛУ ПРОГРАМИ

Модель життєвого циклу розробки програмного продукту

Загалом, життєвий цикл визначається моделлю та описується у формі методології. Модель або парадигма життєвого циклу визначає всю організацію, а часто її основні фази та принципи переходу між ними. Методологія визначає складність робіт, їх детальний зміст, а також ролі та обов'язки експертів на різних етапах обраної моделі.

Сформулюйте задачу, алгоритм розв'язання задачі. Послідовність програми, логіка, вона працює. Чому таким чином, як і всі, перевага і їх надія.

Життєвий цикл програмного забезпечення супроводжується розробкою, обігом і використанням програмної документації.

Модель життєвого циклу — це структура процесів, робіт і завдань, яка включає розробку, експлуатацію та підтримку програмних продуктів; охоплює життєвий цикл системи, від визначення системних вимог до виведення з експлуатації. Найбільшого розповсюдження наразі досягли дві моделі:

- Каскадна модель;
- Спіральна модель.

Перша — це каскадна модель, де кожна фаза роботи виконується лише один раз. На кожному етапі робота виконується настільки ретельно, що немає необхідності повертатися до попереднього етапу. Перед переходом до наступного етапу необхідно перевірити результати кожного етапу.

Спіральна модель — це генератор моделі процесу управління ризиками для програмних проєктів. Базуючись на унікальній моделі ризику проєкту, спіральна модель допомагає командам прийняти елементи однієї або кількох моделей процесів, таких як інкрементальний, водоспадний або еволюційний архетипи.

Створена спіральна модель життєвого циклу на основі можливості внесення змін у процес і проміжні продукти. Метою внесення змін є задоволення потреб користувача, як тільки буде встановлено, що створений артефакт або елемент документації не відповідає фактичному стану розробки. Дана модель

життєвого циклу дозволяє проводити аналіз, перевірку, оцінку коректності продукту циклу розробки та приймати рішення про перехід до наступного циклу або повернення до попереднього циклу для доопрацювання проміжного продукту.

Ця модель відрізняється від каскадної моделлю можливістю багаторазово повертатися до процесу формулювання вимог і переробляти версію системи з будь-якого процесу моделі.

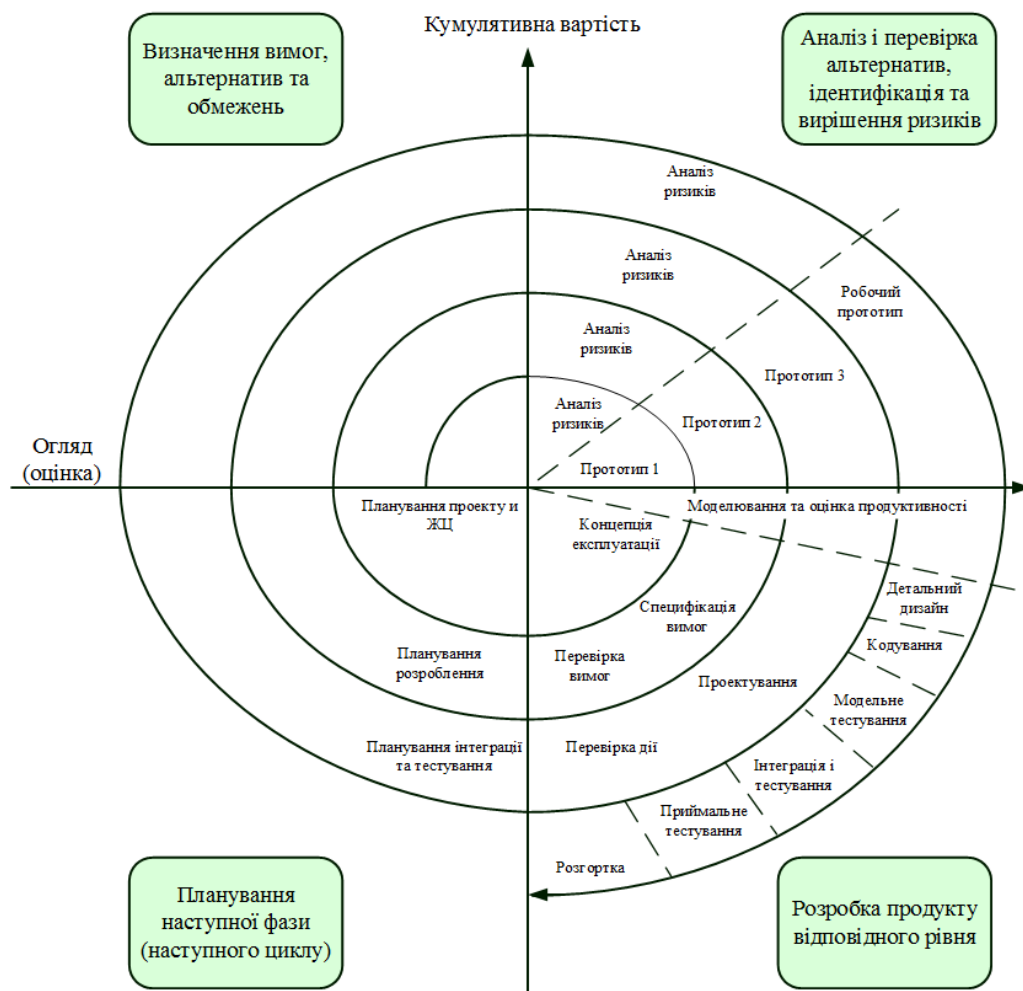


Рисунок 2.1. Спіральна модель

Найбільш помітними є такі методи розробки програмного забезпечення:

Rational Unified Process (RUP), який забезпечує ітераційну модель розробки з чотирма фазами: ініціювання, дослідження, створення та впровадження. Проходячи чотири основні фази, які називаються циклами розробки, кожен цикл завершується створенням робочої версії системи. Під час технічного обслуговування продукт продовжує розвиватися і знову проходить ті

самі фази. Розробка програмного забезпечення на основі RUR базується на створенні та підтримці моделей на основі UML.

Microsoft Solutions Framework (MSF) — це підхід, призначений для підвищення керованості процесу розробки незалежного підприємства (Microsoft), який потім використовують розробники, що використовують продукти Microsoft. Подібно до RUR, підхід також включає чотири фази: аналіз, проектування, розробка, стабілізація, є ітераційним і передбачає використання об'єктно-орієнтованого моделювання. У порівнянні з RUR, MSF в основному зосереджується на розробці бізнес-додатків.

eXtreme Programming (XP) — це підхід, зосереджений на ефективній комунікації між клієнтами та виконавцями в рамках проекту розробки ІС для відстеження змін у вимогах. Підхід заснований на командній роботі та постійному тестуванні прототипів.

Гнучка розробка програмного забезпечення (Agile) — це клас методів розробки програмного забезпечення, заснований на ітераційній моделі розробки програмного забезпечення, у якій вимоги та рішення розробляються шляхом співпраці між самоорганізованими командами.

Реалізація програмного забезпечення

Автоматизоване програмне забезпечення (CASE) — це сфера програмних засобів, які використовуються для розробки та впровадження програм. Інструменти CASE схожі на інструменти автоматизованого проектування (CAD), які використовуються для проектування апаратних продуктів, і частково натхненні ними. Інструменти CASE використовуються для розробки високоякісного програмного забезпечення без помилок і зручного обслуговування.

Сучасні інструменти CASE охоплюють широкий спектр підтримки численних методів проектування ІБ: від простих інструментів аналізу та документування до комплексних інструментів автоматизації, що охоплюють весь життєвий цикл програмного забезпечення.

Мета полягає в тому, щоб створити повністю автоматизоване програмне забезпечення на основі технічних описів.

Інструменти CASE — це програми, які допомагають розробникам програмного забезпечення планувати, проектувати та документувати результати їхньої роботи. Важливою частиною CASE-Tools є графічна нотація, яка використовується для візуалізації архітектури програмної системи.

Інструменти CASE часто інтегровані в інтегроване середовище розробки (IDE); іноді існують окремі програми, які повністю присвячені CASE (без надання інших типових елементів середовища розробки).

На додаток до сучасної об'єктно-орієнтованої графічної нотації UML, деякі інструменти CASE також підтримують так звані методи моделювання даних структурованого аналізу та структурованого проектування (SA/SD) і моделювання зв'язків сутностей (ERM/SERM).

Програмне забезпечення CASE часто асоціюється з методами розробки інформаційних систем і засобами автоматизації, які можна використовувати в процесі розробки програмного забезпечення.

Для створення автоматизованої інформаційної системи обліку відвідуваності учнями навчальних закладів обрано Microsoft SQL Server.

Microsoft SQL Server — це програмний продукт, розроблений компанією Microsoft. Цю програму можна використовувати для дослідження, розробки та обслуговування класичних серверних додатків мови SQL і веб-додатків. SQL Server містить розширену версію SQL Management Studio.

Основною мовою запитів, що використовується, є Transact - SQL, спільно створений Microsoft і Sybase. Transact-SQL — це стандартна реалізація ANSI/ISO мови структурованих запитів (SQL) із розширеннями. Використовується для обробки широкого спектру баз даних від особистих до великих корпоративних баз даних, вона конкурує з іншими СУБД у цьому сегменті ринку.

Microsoft SQL Server і Sybase ASE під час взаємодії з мережею використовують протокол прикладного рівня під назвою «Потік табличних даних» (TDS, Tabular Data Transfer Protocol). Протокол TDS також реалізовано в

проекті FreeTDS, щоб надати різноманітним програмам можливість взаємодії з базами даних Microsoft SQL Server і Sybase.

Редакція Express містить лише ядро бази даних MS SQL Server. Це корисно, якщо вам потрібно підключитися віддалено або потрібно виконати віддалене адміністрування.

SQL Server Management Studio Express — це пакет без бази даних, лише інструменти для керування екземпляром SQL Server.

Дослідження роботи баз даних

База даних — це систематично відформатований набір пов'язаних даних, які можна шукати на вимогу. Американський комп'ютерний вчений Джим Грей писав про бази даних: «Коли люди вживають слово «база даних», вони в основному кажуть, що дані повинні бути індивідуальними та мати схематичну структуру. Це саме те, що описує база даних».

База даних — це область, де зберігається структурована інформація або дані. Бази даних створюються та керуються через систему керування базами даних. Для цих систем приклади включають Microsoft Access, MySQL, IBM DB2, Informix, Interbase, Microsoft SQL Server, PostgreSQL, Oracle і Sysbase.

Є можливість організувати та індексувати дані в таблиці, рядки та стовпці, щоб полегшити пошук відповідної інформації. Обробники баз даних створюють бази даних таким чином, що лише один набір програмного забезпечення забезпечував доступ до даних для всіх користувачів.

Основне призначення бази даних — обробка великих обсягів інформації шляхом зберігання, пошуку та керування даними.

За моделлю організації даних розрізняють такі бази даних:

- Ієрархічна база даних може бути представлена як дерево, що складається з об'єктів різних рівнів. Між об'єктами існують зв'язки типу «предок-нащадок».
- Мережева база даних є подібною до ієрархічної, за винятком того, що кожен об'єкт може мати більше одного предка.

- Реляційна база даних зберігає дані у вигляді таблиць. Найчастіше використовують БД використовують реляційну модель даних.
- Об'єктно-орієнтовану оформляють у вигляді моделей об'єктів.

Якщо говорити більш конкретно, в інформатиці та повсякденному використанні комп'ютерів база даних відноситься до організованої самостійної колекції пов'язаних даних, що зберігаються в електронному та цифровому вигляді, програмного забезпечення, яке обробляє таку колекцію (система керування базами даних або СУБД), а також когнітивні області, в яких вони вивчаються. На додаток до властивої здатності зберігати дані, бази даних також забезпечують, завдяки своєму дизайну та пріоритезації даних, так звані системи керування вмістом, здатність швидко отримувати та оновлювати дані. Добре структурована база даних надає актуальну та точну інформацію. Оскільки правильна структура має вирішальне значення для досягнення цілей бази даних, варто витратити деякий час на вивчення принципів правильної структури. Зрештою, це збільшить ваші шанси отримати базу даних, яка відповідає вашим потребам і легко адаптується до змін.

Процес розробки структури баз даних полягає в діях, описаних далі. Для початку необхідно визначити призначення **бд (?)** - це допоможе підготуватися перед наступними кроками. Наступним кроком буде пошук необхідних даних та їх впорядкування. Потрібно підготувати всі назви таблиць, їх поля та типи даних. Наступним кроком буде перенесення цих даних в утиліту "SQL Server Management Studio" за допомогою коду і конструктору.

Після цього необхідно указати первинні ключі - елементи таблиці які в подальшому будуть використовуватися як унікальний ідентифікатор кожного рядка.

Кожна таблиця повинна містити стовпець або набір стовпців, які однозначно ідентифікують кожен рядок, що зберігається в таблиці. Зазвичай це унікальний ідентифікаційний номер, наприклад, номер паспорта або серійний номер працівника. У термінології баз даних ця інформація називається первинним ключем таблиці.

Первинний ключ завжди повинен містити значення. Якщо в якийсь момент значення в стовпці може стати непризначеним або невідомим (без значення), його не можна використовувати як частину первинного ключа.

Завжди потрібно вибирати первинний ключ, значення якого не змінюється. У базі даних із кількома таблицями первинний ключ однієї таблиці можна використовувати як посилання на інші таблиці. Якщо первинний ключ змінюється, зміна також має бути застосована скрізь, де на неї є посилання. Використовуючи первинний ключ, який не змінюється, ви зменшуєте ймовірність його несинхронізації з іншими таблицями, які на нього посилаються.

Щоб представити зв'язок «один-до-багатьох» у вашій власній структурі бази даних, вам потрібно взяти первинний ключ на одній стороні зв'язку та вставити його як додатковий стовпець у таблицю на стороні багатьох зв'язків.

Створення зв'язків між таблицями потрібні для їх зв'язку в роботі. Якщо все було зроблено правильно, то залишається лише заповнити таблиці даними і це буде останнім кроком в створення даної бази даних.

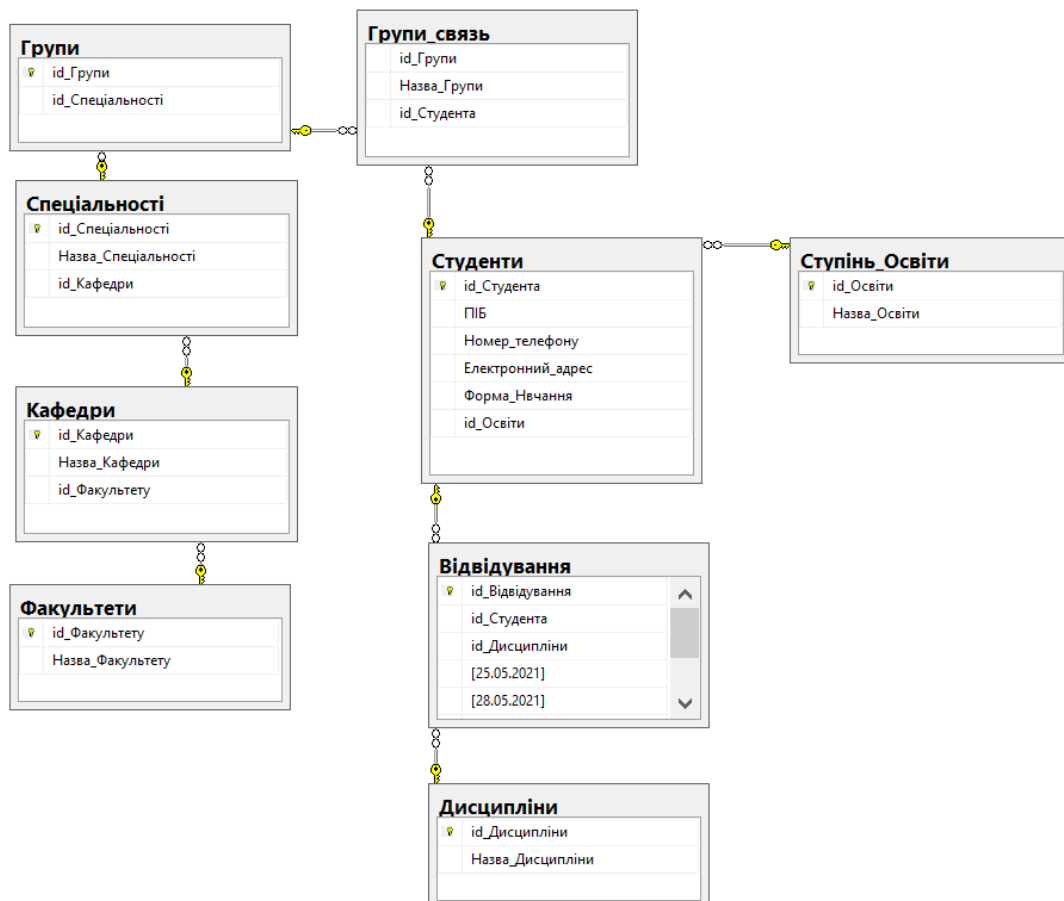


Рис.2.2. Схема даних

Вибір засобів для документування програмних продуктів

Робочий документ — це набір документів, які містять необхідну інформацію для розробки, виробництва, підтримки та виконання певної програми. Для документування текстових документів я обрала програму Microsoft Word, яка входить до пакету офісних програм Microsoft Office.

Microsoft Office Access — це реляційна система керування базами даних (СУБД) корпорації Microsoft. Система є частиною пакету Microsoft Office. Він має широкий спектр можливостей, включаючи відповідні запити, зв'язок із зовнішніми таблицями та базами даних. Завдяки вбудованій мові VBA можна писати програми для роботи з базами даних у самому Access.

Текстовий редактор Microsoft Word — текстовий редактор, призначений для створення, перегляду, редагування та форматування текстових документів. Microsoft Word - один з найпоширеніших текстових редакторів.

Це пов'язано з його багатьма перевагами, в основному включаючи широкий спектр функцій. Широке використання Word також сприяє його вбудованим інструментам для перетворення файлів, створених іншими текстовими редакторами, у файли формату Word і навпаки.

Існує кілька версій Microsoft Word для Windows, і кожна наступна версія, як правило, сумісна з попередньою версією та має додаткові функції.

Створення інформаційної бази та таблиць

Для створення бази даних використовується мова запитів SQL. Створення бази даних показано далі на рисунку 2.3.

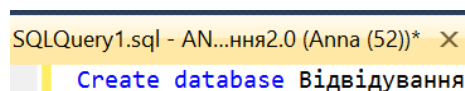


Рис.2.3. Створення бази даних

Для створення таблиць також було використано мову запитів SQL.

Нижче зображено створену першу таблицю «Відвідування».

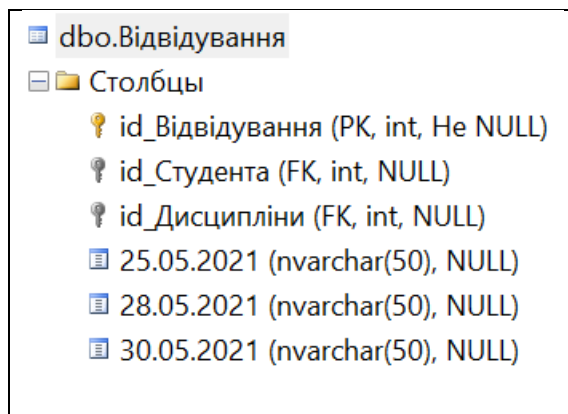


Рис. 2.4. Створення таблиці «Відвідування»

Далі було створено таблицю «Групи», її зображено на Рис.2.5:

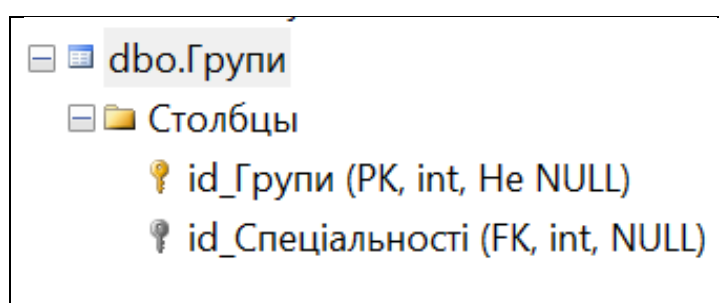


Рис. 2.5. Створення таблиці «Групи»

Далі були створення таблиці «Групи_связь», її зображено на Рис. 2.6:

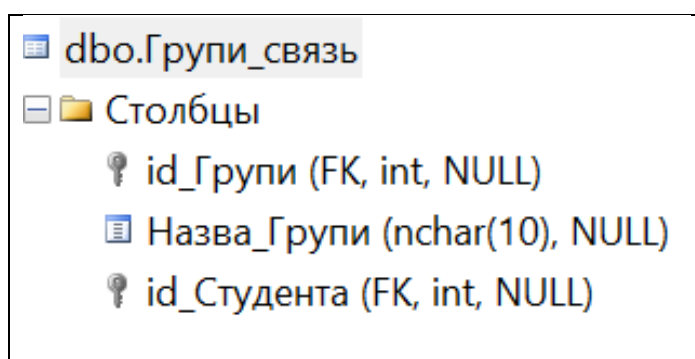


Рис. 2.6. Таблиця «Групи_связь»

Далі на наступному рисунку зображено таблицю «Дисципліни»:

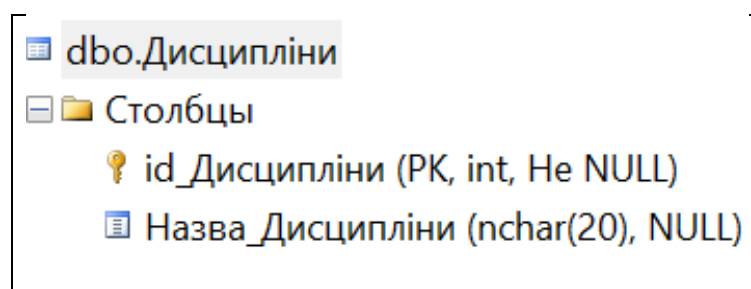


Рис. 2.7. Таблиця «Дисципліни»

Потім було створення таблиці «Кафедри», її зображено на Рис. 2.8.

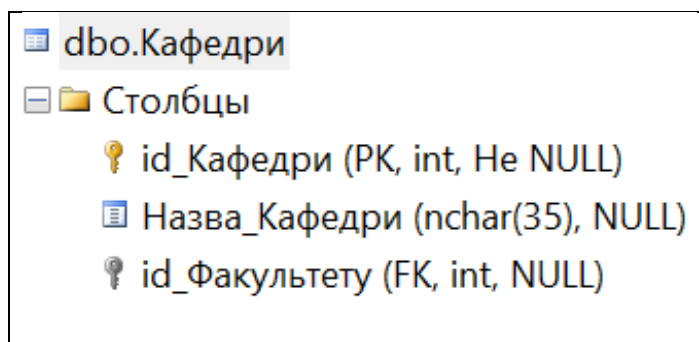


Рис.2.9. Таблица «Кафедри»

На наступному рисунку зображено таблицю «Спеціальності».

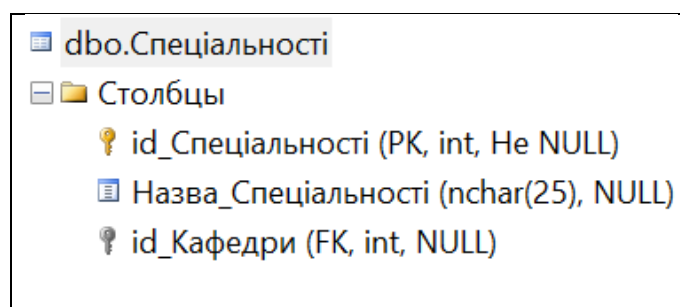


Рис.2.10. Таблица «Спеціальності»

Далі було створення таблиці «Студенти», її зображено на Рис. 2.11:

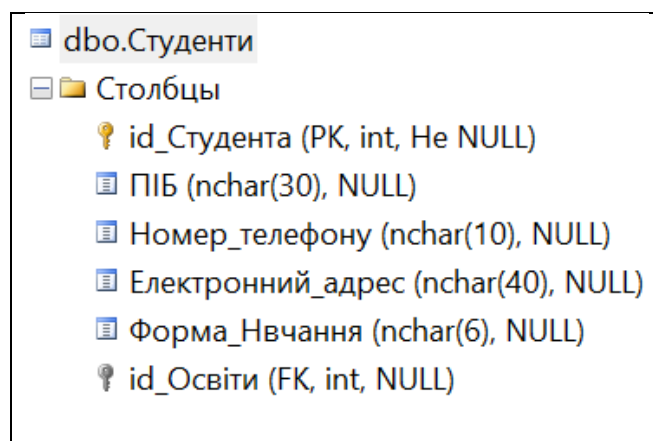


Рис.2.11. Таблица «Студенти»

На наступному рисунку зображено таблицю «Ступінь_Освіти».

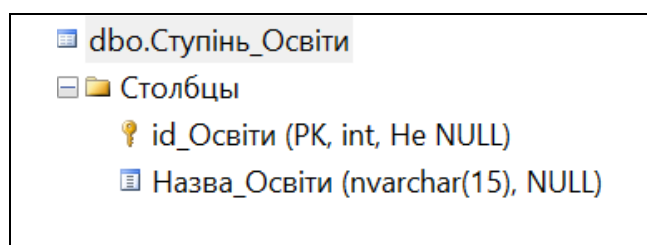


Рис.2.12. Таблица «Ступінь_Освіти»

Далі було створення таблиці «Факультети», її зображено на Рис. 2.13:

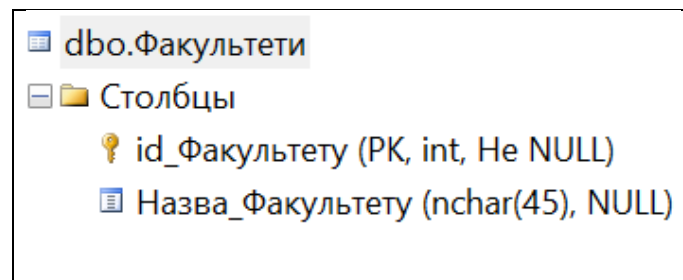


Рис.2.13. Таблиця «Факультети»

Створення блок – схем

Блок-схема алгоритму — це схема, що представляє технічний процес. Блок-схему також можна визначити як принципову схему алгоритму, покроковий підхід до вирішення проблеми.

Блок-схеми використовуються при проектуванні та документуванні простих процесів або програм. Основною структурою блок-схеми є блок-схема, яка абстрагує типи вузлів, їхній вміст та іншу допоміжну інформацію.

Форма доступу показана нижче з трьома кнопками: «Отримати дані», «Зберегти дані» та «Додати стовпець дати». Нижче буде наведена блок-схема цих кнопок.

Рис. 2.14. Форма «Відвідування»

На наступних рисунках буде зображено ряд блок – схем до кнопок «Отримати дані», «Додати колонку з датою» та «Зберегти дані», які відповідають за певні методи.

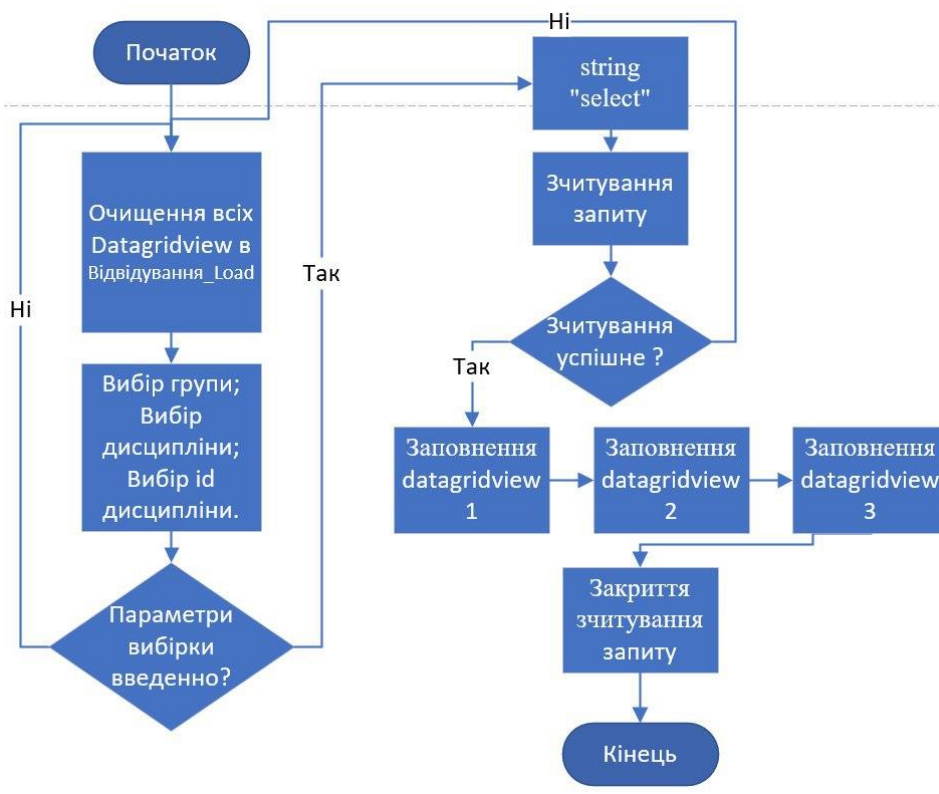


Рис. 2.15. Блок – схема до кнопки «Отримати дані»

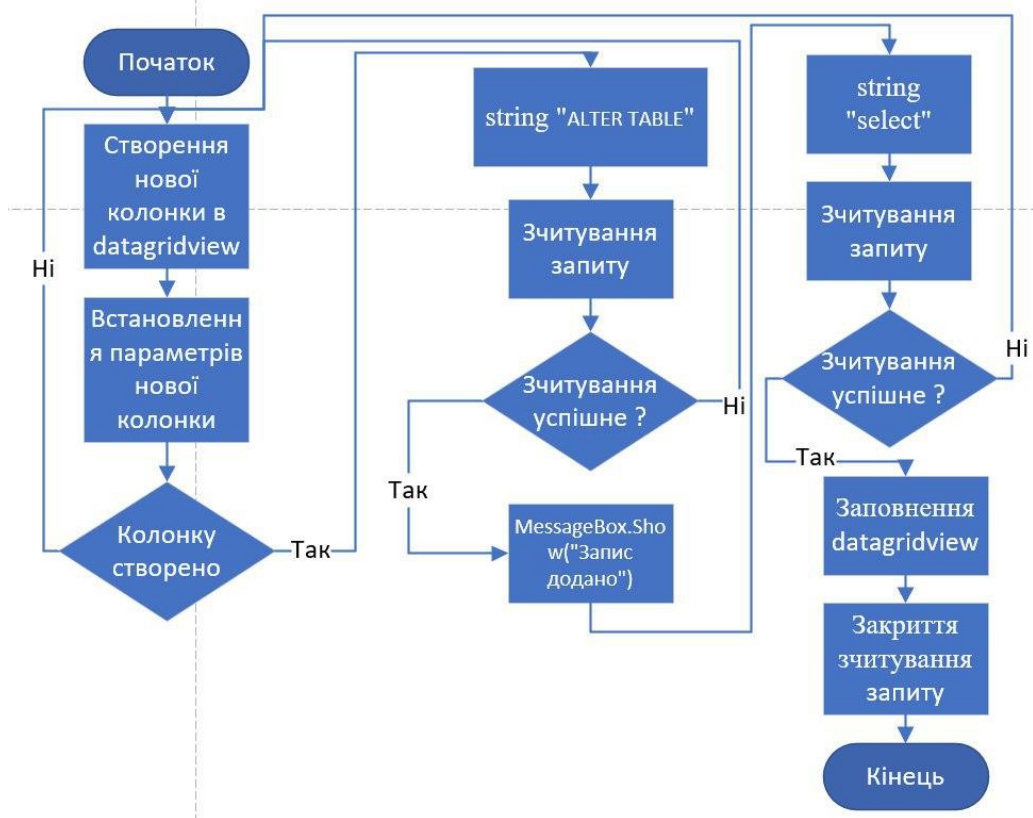


Рис. 2.16. Блок – схема до кнопки «Додати нову колонку»

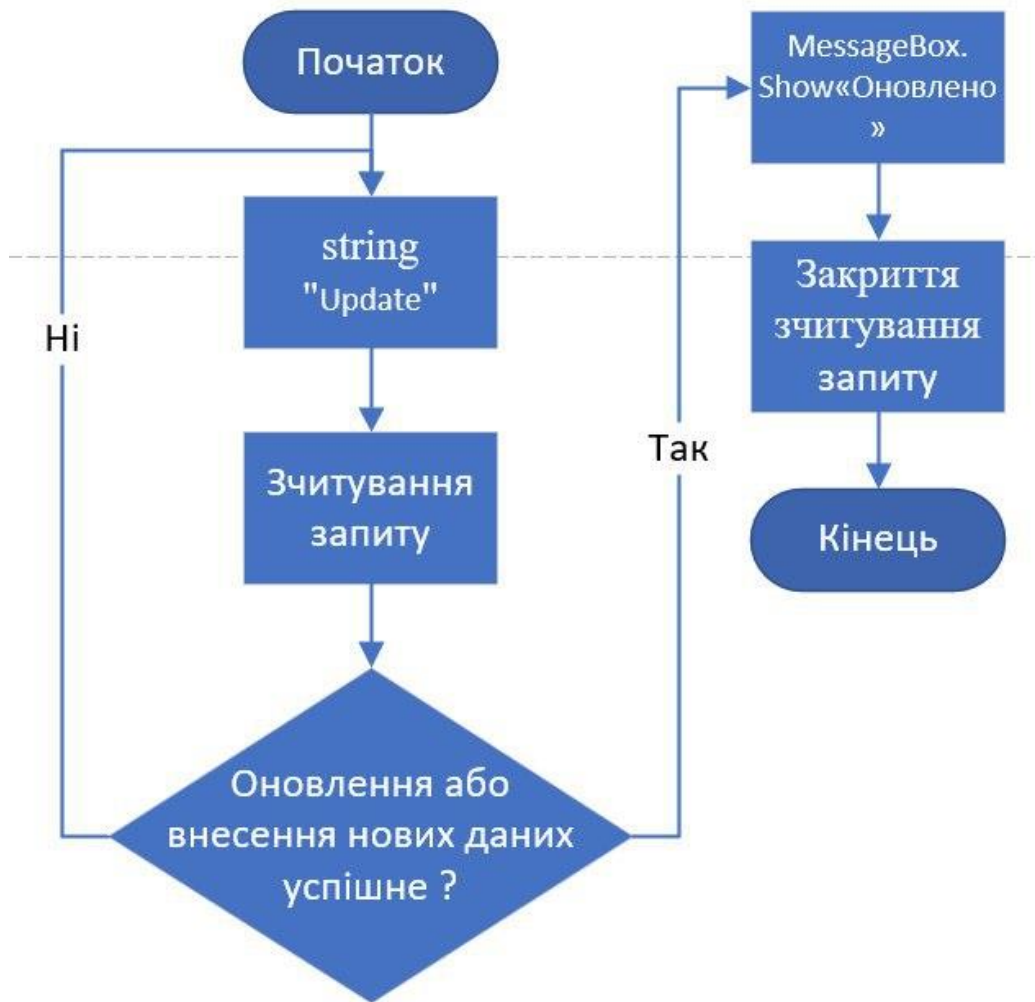


Рис. 2.17. Блок – схема до кнопки «Зберегти дані»

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ОБЛІКУ

Розробка алгоритмів функціонування системи

Для створення програмного забезпечення макета була обрана мова структурованих запитів - SQL (Structured Query Language) — декларативна мова програмування для взаємодії користувача з базами даних для запитів, оновлення та керування реляційними базами даних.

Створення, налаштування та керування модулями зберігання даних здійснюється за допомогою утиліти «SQL Server Management Studio» — інтегрованого середовища для створення, керування, налаштування та керування компонентами SQL Server.

Крім того, у цьому дипломному проекті розроблено графічний інтерфейс для бази даних, який, у свою чергу, розроблено в середовищі Microsoft Visual Studio, серії продуктів Microsoft, включаючи інтегровані середовища розробки програмного забезпечення та багато інших інструментів. Ці продукти дозволяють користувачам Розробляти консольні програми, тобто програми з графічним інтерфейсом, які підтримують технологію Windows Forms.

База даних — це інструмент для збору й упорядкування інформації. Щоб створити базу даних, спочатку потрібно визначити предметні області потрібної бази даних, а потім створити потрібні таблиці та зв'язки між ними. Усі вони перераховані нижче.

Файл основних даних містить інформацію, необхідну для запуску бази даних, і посилання на інші файли в базі даних. Дані користувача та об'єкти можуть зберігатися в цьому файлі або у файлах допоміжних даних. Кожна база даних має головний файл даних. Рекомендується використовувати розширення MDF як назву файлу основних даних.

Файли журналу транзакцій містять інформацію для відновлення бази даних. Для кожної бази даних має існувати принаймні один файл журналу. Для файлів журналу транзакцій рекомендується використовувати розширення LDF.

Після створення бази даних до неї можна отримати доступ за допомогою команди «use».

Кожна база даних містить таблиці, які зберігають певні дані.

Таблиця - це об'єкт бази даних, який зберігає дані в певній одиниці пам'яті, а також це об'єкт, який може бути представлений традиційною прямокутною таблицею.

База даних «Відвідування» розроблювалася паралельно з даним ПЗ. Було розроблено наступні таблиці: «Відвідування», «Групи», «Групи_св'язь», «Дисципліни», «Кафедри», «Спеціальності», «Студенти», «Ступінь_Освіти», «Факультети», до яких вже були внесені певні дані.

У головному меню програми є основна форма для перегляду відвідуваності студентів. Крім того, є кнопка з випадаючим списком «Функція», який складається з трьох пунктів: «Додати відвідуваність», «Додати студента», «Видалити студента».

Для кожного з цих пунктів є окрема таблиця. Форма «Додати відвідування» дозволяє вам додати дати курсу, які відвідують ваші студенти. Ви також можете додати певні теги (лише один символ).

У формі «Додати Студента» є можливість створювати нового студента. Додавати певну інформацію про нього попередньо вибравши групу.

Форма «Видалити Студента» містить в собі опцію видалення студента з групи, якщо є необхідність.

В усіх формах є можливість оновлення, редагування, додавання, видалення, перегляд та збереження даних. Всі дані містяться в БД «Відвідування».

Графічний інтерфейс користувача – це система засобів, за допомогою яких користувач взаємодіє з комп'ютером. Його основу складають усі системні об'єкти та функції, представлені користувачеві у вигляді екранних графічних компонентів: вікон, кнопок, розкритих полів, значків, списків, меню.

Під час реалізації проекту розроблено графічний інтерфейс з використанням мови програмування C#, та систему побудови інтерфейсу «Windows Forms» для Windows-додатків.

Перед тим, як користувач увійде в головне меню програми, він повинен пройти через меню авторизації - ввести логін і пароль реального облікового запису. Якщо користувач введе правильні дані, то він перейде в головне меню,

але якщо користувач введе неправильні дані при введенні логіна або пароля, то він отримає повідомлення про помилку і може повторити спробу.

Програмні засоби дипломної бакалаврської роботи розроблено в оболонці програмного додатку Microsoft Visual Studio. Цей ППІ необхідний для розробки прикладних програм і є системою випадків.

Microsoft Visual Studio — це сімейство програмних продуктів Microsoft, що включає інтегроване середовище розробки програмного забезпечення. За допомогою даної серії програмних продуктів можна самостійно розробляти прості консольні програми, а також програми з простим графічним інтерфейсом, у тому числі використані при створенні даної дипломної програми з підтримкою Windows Forms.

WPF — це система, необхідна для створення клієнтських додатків Windows із візуально привабливими можливостями взаємодії з користувачем, і це графічна (презентаційна) підсистема в .NET Framework (починаючи з версії 3.0) з використанням мови XAML. За допомогою WPF ви можете створювати широкий спектр автономних додатків і програм, що запускаються в браузері.

За своєю суттю WPF — це система векторної візуалізації, яка не залежить від роздільної здатності пристрою виведення та створена з урахуванням можливостей сучасних графічних пристроїв. WPF надає інструменти для створення візуальних інтерфейсів, включаючи XAML (Extensible Application Markup Language), елементи керування, зв'язування даних, макети, 2D і 3D графіку, анімацію, стилі, шаблони, документацію, текст, мультимедіа та дизайн.

Базовою графічною технологією WPF є DirectX, який, на відміну від Windows Forms, використовує GDI/GDI+. Продуктивність WPF вища, ніж GDI+, завдяки використанню апаратного прискорення графіки через DirectX.

Пропонований додаток дозволяє оптимізувати та автоматизувати облік. Програма дозволяє вводити, редагувати та видаляти дані у формі. Програмний продукт може бути вдосконалений і розроблені нові шаблони документів, необхідних для бухгалтерського обліку.

Основною вимогою для запуску РР є наявність окремого комп'ютера з встановленим VS та SQL Server 2014 Management Studio, оскільки може знадобитися додаткова настройка комп'ютерної системи.

Програмна реалізація

Робота програми розпочинається з запуску файлу .sln. Нижче на рисунку зображено сам файл та необхідні папки для коректної роботи програми:

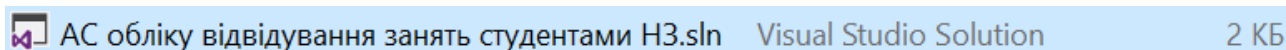


Рисунок 3.1. Файл для запуску програми

Після запуску програми відкривається титульна сторінка програми з назвою магістерської роботи та прізвищем. Як показано нижче:

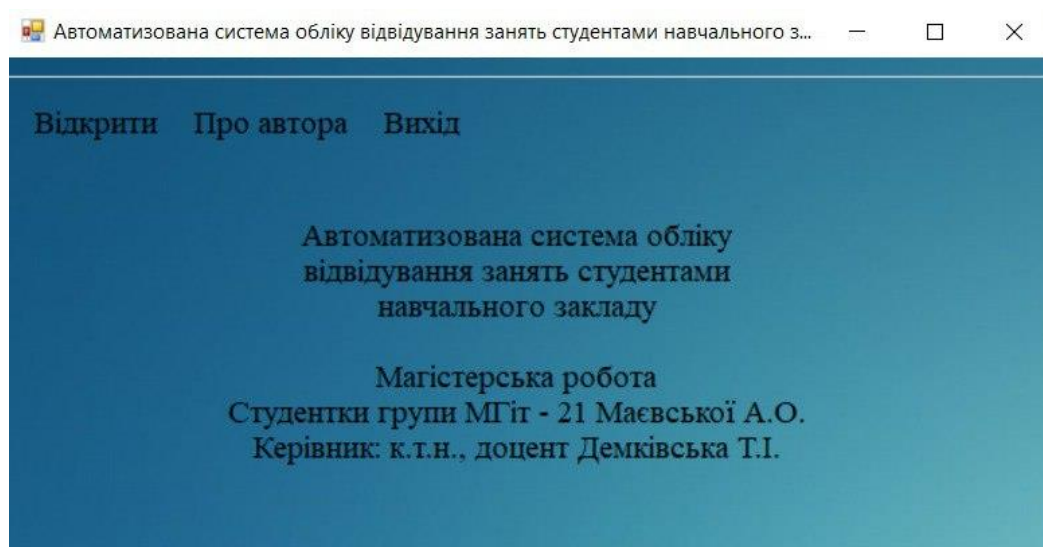


Рис. 3.2. Головна сторінка програми

Якщо натиснути на вкладку «Про автора» з'явиться коротка інформація, яку зображено на рисунку нижче:

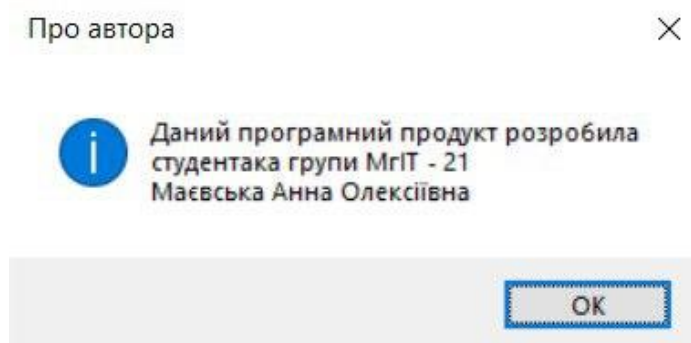


Рис. 3.3. Вкладка «Про автора»

Після натискання на вкладку «Відкрити» з'являється вікно «Авторизації», на якому потрібно ввести логін та пароль потрібного користувача для входу в систему. Рисунок даного вікна зображено нижче:



Рисунок 3.4. Авторизація

Якщо авторизація не пройшла успішно, то на екрані з'явиться спливаюче вікно. Воно зображено на наступному рисунку нижче:

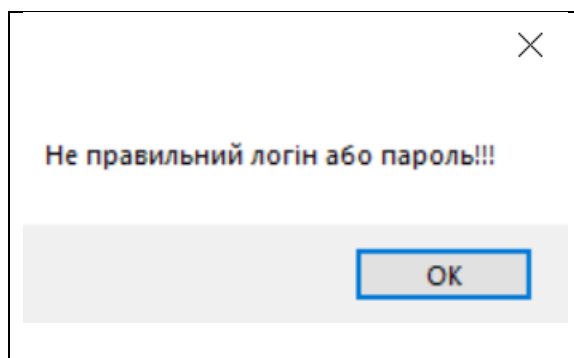


Рисунок 3.5. Помилка при авторизації

Після успішної авторизації з'являється головне вікно «Автоматизована система обліку відвідування занять студентами навчального закладу»:

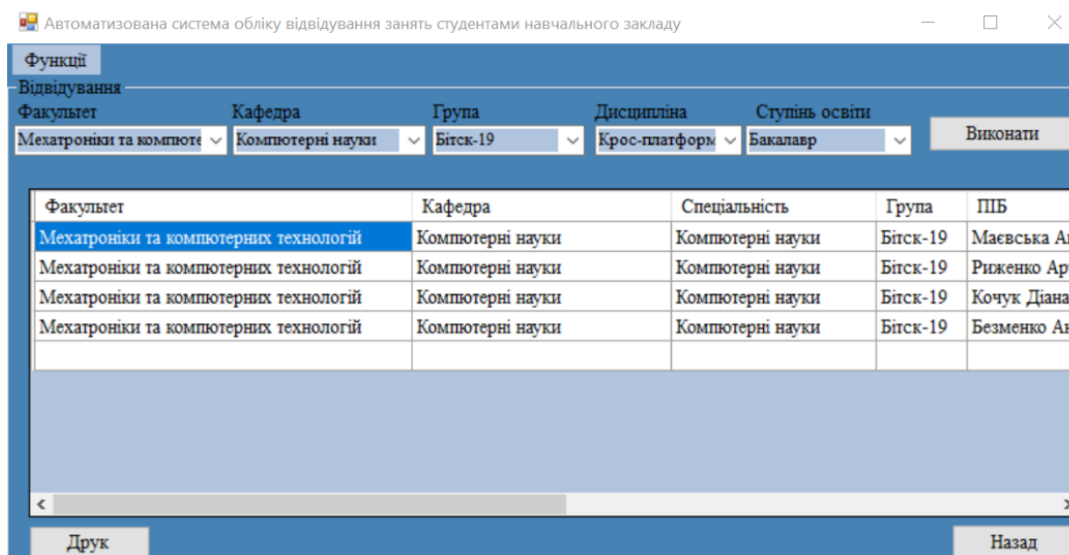


Рисунок 3.6. Головне вікно

Головне вікно програми включає в себе 3 наступні вкладки: «Додати Відвідування», «Додати Студента», «Видалити Студента». Дані вкладки необхідні в системі для додавання та видалення нових даних, та для їх перегляду та редагування.

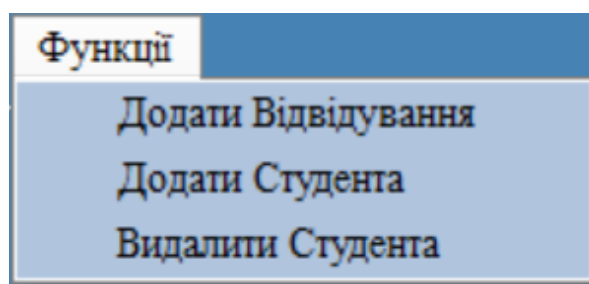


Рисунок 3.7. Вкладки програми

На наступному рисунку зазначено наступну форму «Відвідування». Після натискання на попередній формі в випадаючому списку на кнопку «Додати Відвідування» відкриється нове вікно для додавання відвідування студентів.

Відвідування

Оберіть вашу дисципліну та групу

Групи: Бітск-19 | Дисципліни: Крос-платформи | 3 | Отримати дані

id_Студента	ПІБ	Номер_
1	Маєвська Анна Олексіївна	0504283
2	Риженко Артем Сергійович	0681675
1003	Кочук Діана Вікторівна	0515115
1005	Безменко Анастасія Ігорівна	0681234

id_Відвідування	id_Студента	id_Дисципліни	25.05.202
1	1	3	п
2	2	3	в
3	1003	3	п
5	1005	3	

id_Відвідування	id_Студента	ПІБ	Назва_Групи	Назва_Дисципліни	id_Дисципліни
1	1	Маєвська Анна Олексіївна	Бітск-19	Крос-платформи	3
2	2	Риженко Артем Сергійович	Бітск-19	Крос-платформи	3
3	1003	Кочук Діана Вікторівна	Бітск-19	Крос-платформи	3
5	1005	Безменко Анастасія Ігорівна	Бітск-19	Крос-платформи	3

Додати колонку

Додати колонку з датою | Зберегти дані

"Дату вводити в лапках"

Рис. 3.8. Вікно «Відвідування»

Наступною формою є «Додати Студента». Воно необхідне для реєстрації нового студента в групі та додавання певних його даних. За допомогою запиту на редагування є можливість вносити зміни даних і зберігати їх за допомогою окремої кнопки. Дане вікно зображено на наступному рисунку:

Додати студента

Перегляд

Додати студента Оновити дані Додати студента в групу

id Студента	ПІБ	Номер телефону	Електронний адрес	Форма Нвчання	id Освіти
1	Маєвська Анна Олексіївна	0504283158	annamaevskaya99@gmail.com	Денна	1
2	Риженко Артем Сергійович	0681675817	artem.rizhenko@gmail.com	Денна	1
1003	Кочук Діана Вікторівна	0515115151	k.diana@gmail.com	Денна	1
1005	Безменко Анастасія Ігорівна	0681234567	bezm.a@gmail.com	Денна	1

Перегляд груп
Назва групи

Пошук

id Групи	Назва Групи
1	Бітск-19
1	Бітск-19
1	Бітск-19
1	Бітск-19

Перегляд Ступенів Освіти
Ступінь освіти

Пошук

id Освіти	Назва Освіти
1	Бакалавр
2	Магістр
3	Доктор наук

Додати студента в групу

Вивести інформацію про студентів

id Групи	Назва Групи	id Студента
1	Бітск-19	1
1	Бітск-19	2
1	Бітск-19	1003
1	Бітск-19	1005

Рис. 3.9. Вікно «Додавання Студента»

Останньою формою є «Видалити студента». Вона необхідна для видалення певного студента з певної групи для необхідності. Всі дані видаляються з усієї БД.

Видалення

Студенти
Група

Бітск-19

Пошук

id студента

Видалити

id Студента	ПІБ	Номер телефону	Електронн
1	Маєвська Анна Олексіївна	0504283158	annamaevsk
2	Риженко Артем Сергійович	0681675817	artem.rizhen
1003	Кочук Діана Вікторівна	0515115151	k.diana@gm
1005	Безменко Анастасія Ігорівна	0681234567	bezm.a@gm

Рис. 3.10. Вікно «Видалити Студента»

На наступних рисунках зображено як працює кнопка «Друк»:

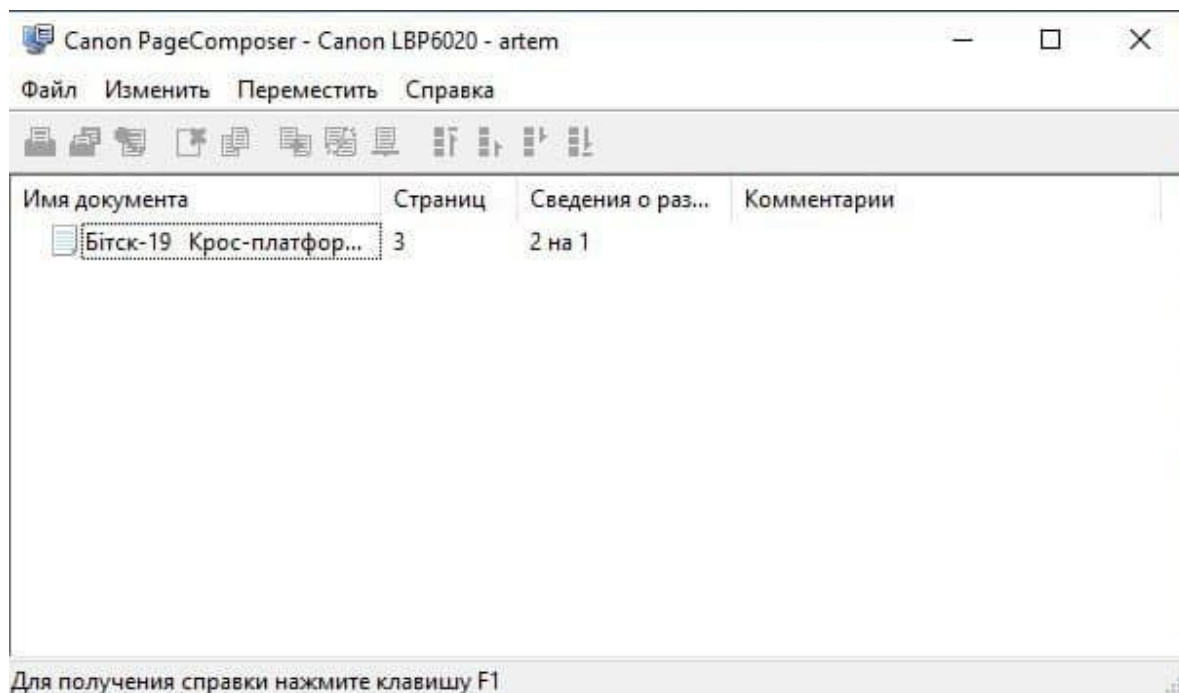


Рис. 3.11. Работа кнопки «Друк»

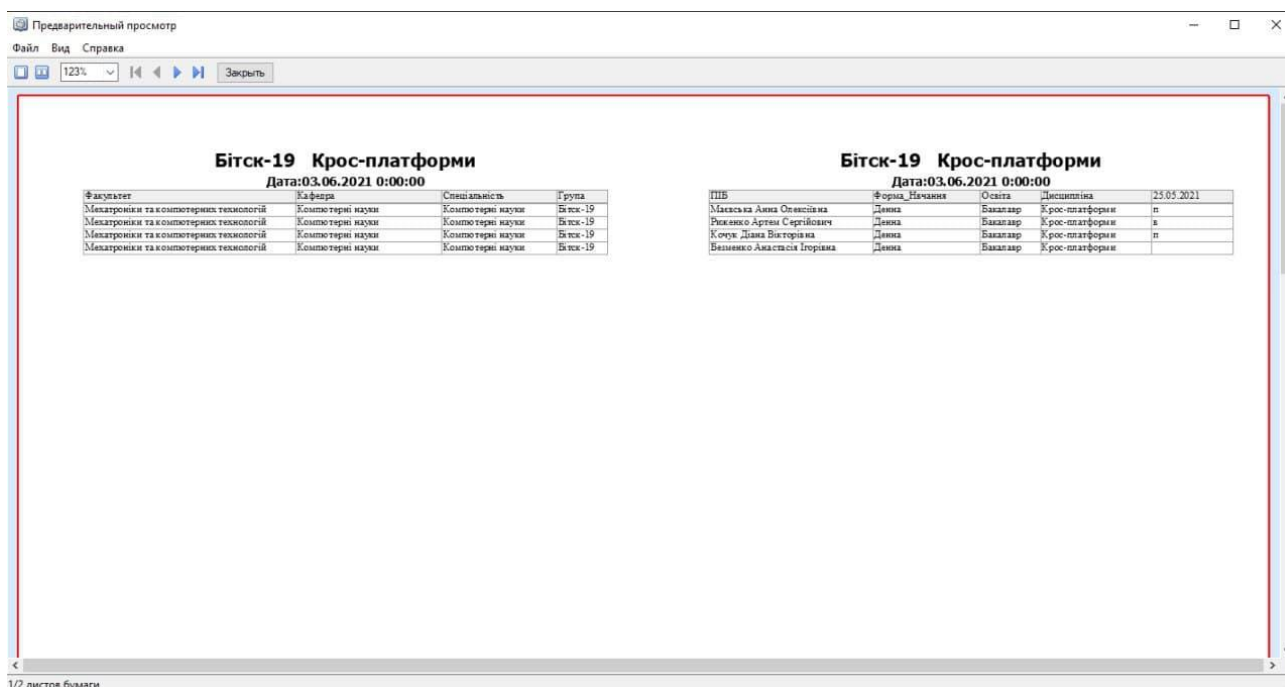


Рис. 3.12. Работа кнопки «Друк»

Програмне та апаратне забезпечення

Тестування системи — це процес технічного дослідження, призначений для отримання інформації про якість продукції. Методики тестування включають процес пошуку помилок, дефектів і тестування програмних компонентів для оцінки.

Прикладне програмне забезпечення – це програмна система, призначена для реалізації конкретних завдань обробки даних, які вирішують користувачі під час своєї діяльності.

Програмне забезпечення — це сукупність системних програм обробки інформації та програмної документації, які взаємопов'язані та необхідні для використання різних програм.

Кожна інформаційна система повинна включати три основні функціональні частини:

- Модуль обробки даних;
- Модуль зберігання даних;
- Інтерфейс користувача.

Програмне забезпечення, яке зберігається на сервері, відповідає за збереження даних. На стороні клієнта зберігається вся інформація про інтерфейс, і обробка даних повинна бути розподілена між клієнтською та серверною частинами.

Апаратне забезпечення — це взаємопов'язана система технічних пристроїв, які виконують функції обробки, зберігання, введення та виведення інформації.

Мінімальні характеристики, необхідні для програми:

- Процесори з тактовою частотою вище 1 ГГц;
- Об'єм оперативної пам'яті не менше 512 Мб;
- Об'єм вільного місця на жорсткому диску не менше 1 Гб;
- Швидкість підключення до Інтернету вище 200 Кбіт/с.

Програмне забезпечення, необхідне для використання цієї програми:

- операційні системи Windows 8, 8.1, 10;
- Microsoft SQL Server;

- .Net Framework 4.5.

Техніка безпеки при роботі з розробленою системою

Основна вимога до техніки безпеки – дотримання правил безпеки комп'ютера, тобто підключення до електромережі. Підключення до живлення комп'ютера має проходити через джерело безперебійного живлення, щоб, якщо щось піде не так з індикаторами, користувач міг зберегти в програмі та вимкнути комп'ютер.

Комп'ютери не слід розміщувати в дуже холодних або жарких приміщеннях, оптимальною температурою для комп'ютера вважається від 5 оС до 30 оС. Крім того, комп'ютер потрібно поставити на рівну поверхню, щоб забезпечити рівномірне охолодження системи.

У процесі використання розробленого програмного забезпечення необхідно дотримуватись загальних правил техніки безпеки при роботі з комп'ютером. При використанні комп'ютера діють такі небезпечні та шкідливі фактори:

- фізичний;
- Психофізика.

Стаття 18. Працівники зобов'язані дотримуватись вимог нормативних актів про охорону праці працівники зобов'язані:

а) розуміти і виконувати вимоги відповідних нормативних актів з охорони праці, правил поведження з машинами, установами, обладнанням та іншими засобами виробництва, використання засобів колективного та індивідуального захисту;

б) виконувати обов'язки з охорони праці, передбачені колективним договором (угодою, трудовим договором) і правилами внутрішнього трудового розпорядку підприємства;

в) проводити в установленому порядку попередні та періодичні медичні огляди;

г) Співпрацювати з власником для організації безпечних і нешкідливих умов праці, особисто вживати необхідних заходів для усунення будь-якої виробничої ситуації, яка становить загрозу його життю чи здоров'ю або

оточуючим людям і навколишньому природному середовищу, повідомляти про небезпеку для його безпосередній керівник або інші посадові особи.

ВИСНОВКИ

В даній дипломній магістерській роботі було досліджено і розроблено структури бази даних та програмний продукт для автоматизованого обліку відвідування занять студентами навчального закладу за допомогою програмної бази «Windows-Forms».

Проаналізовано роботу введення та виведення інформації з бази даних та автоматизованої інформаційної системи.

Проаналізовано роботу баз даних в програмному засобі «Microsoft SQL Server» та розроблено БД під назвою «Відвідування» на мові структурованих запитів SQL.

На об'єктно-орієнтованій мові програмування C# було розроблено конкретні класи, які було по-різному реалізовано, але виконували ті самі дії, результат було описано в звіті та відображено в інтерфейсі.

Також було розроблено алгоритм ведення обліку відвідування вищих навчальних закладів студентами. Розроблено блок-схеми основних функцій у програмі.

Було оформлено звітність по даним в програмному продукті з можливістю їх друку.

Всі помилки в програмі опрацьовані та виправлені.

Розроблений програмний продукт може бути реально введений в експлуатацію в будь-яких навчальних закладах, після індивідуального коригування від структуру конкретного учбового закладу.

ЛІТЕРАТУРА

1. Організація баз даних: практичний курс: Навч. посіб. для студ. / А. Ю. Берко, О. М. Верес; Нац. ун-т «Львівська політехніка» - 2003.
2. Резніченко, В.А. (2021). 60 років базам даних. «Проблеми програмування» (№ 3) - 2022 р.
3. Silberschatz, Abraham; Sudarshan, S. (2011). Database system concepts (вид. 6). New York: McGraw-Hill – 2011 р.
4. Копанєва В. О. Формати опису мережевих інформаційних ресурсів. Інформаційна діяльність: Проблеми науки, освіти, практики, Київ, 17-19 травня 2011 р.
5. Воройський Ф. С. – Інформатика. Новий систематизований словник-довідник (Вступний курс з інформатики та обчислювальної техніки у термінах) - М.: Ліберія, 2001.
6. Кормен, Томас; Лейзерсон, Чарльз; Рівест, Рональд; Стайн, Кліфорд Частина 3: Структури даних. Вступ до алгоритмів - 2019 р.
7. Лука Карделлі, Пітер Вегнер. Про розуміння типів, абстракції даних і поліморфізму – 2006 р.
8. Що нового у Visual Studio. Архів оригінал – 27 березня 2016.
9. Оголошуємо загальну доступність Visual Studio – 19 березня 2017.
10. Solidity Ethereum у Microsoft Visual Studio - CCN: Фінансові новини біткойнів і криптовалют. CCN: Фінансові новини біткойнів і криптовалют – 1 червня 2016 р.
11. Visual Studio – 22 серпня 2016 р.
12. Айвор Хортон - Microsoft Visual C++ 2005 р.
13. Базовий курс Beginning Visual C++ — М.: «Діалектика», 2007 р.
14. Microsoft's Visual C++ Developer Center Community - 19 жовтня 2009 р.
15. Building Win32 Applications Using Visual C++ 2005 Express – 24 січня 2010 р.
16. Крейн, Річард; Резнік, Стів; Боуен, Кріс «Essential Windows Communication Foundation» - 11 лютого 2008 р.

17. WCF Client is Open Source. - 22 травня 2015 р.
18. Announcing the .NET Framework 4.7 – 19 травня 2017 р.
19. Сведения про життєвий цикл різних версій .NET Framework - 21 січня 2012 р.
20. Visual Studio 2010 and .NET FX 4 - 21 січня 2012 р.
21. Перелік мов програмування для Java Virtual Machine - 21 серпня 2011 р.
22. Gosling and McGilton The Java Language Environment - 12 травня 2013 р.
23. Джон Скит. С# для професіоналів — М.: «Вільямс», 2014 р.
24. Крістіан Нейгел і др. С# 5.0 і платформа .NET 4.5 для професіоналів 2013 р.
25. А. Хейлсберг, М. Торгерсен, С. Вілтамут, П. Золото. Язык програмування С#. Классика Информатика. 4-е издание - 2012 р.
26. Э. Стілмен, Дж. Гринь. Вивчаємо С#. 2-е издание Head First С#, 2ed. - 2012 р.
27. Ендрю Троелсен. Язык програмування С# 5.0 і платформа .NET Pro С# 5.0 і .NET 4.5 Framework 2013 р.
28. Джозеф Албахарі, Бен Албахарі. С# 6.0. Коротко про С# 6.0: повний довідник. — 2018 р.
29. Герберт Шилдт С# 4.0 Повний довідник. — М.: «Вільямс», 2010 р.
30. Крістіан Нейгел, Карлі Уотсон Visual С# — М.: Діалектика, 2010 р. Microsoft нескінченно розчаровує новими програмами для Windows Phone - 29 вересня 2017 р.
31. Сумісність із Visual Studio 2019 | Microsoft Docs – 3 червня 2020 р.
32. Blog post on Xamarin Releases. Xamarin – березня 2018 р.
33. Desktop Operating System Market Share - 6 жовтня 2014 р.
34. Шеховцов В. А. Операційні системи К, Видавнича група BHV. 2005 р.
35. Andrew S. Tanenbaum, Herbert Bos. Modern Operating Systems на «Google Books» 2015 р.

36. Апаратне і програмне забезпечення персонального комп'ютера, HTML+RDFa - 16 вересня 2012 р.

37. Керівництво для настільних персональних комп'ютерів [Електронний ресурс] - Режим доступу: <https://docs.microsoft.com/enus/dotnet/desktop/wpf/overview/?view=netdesktop-5.0>.

38. Гуманітарний портал [Електронний ресурс] - Режим доступу <https://gtmarket.ru/concepts/7091> – 2012 р.

39. 4. C# для початківців [Електронний ресурс] - Режим доступу: <http://www.cyberforum.ru/csharp-beginners/thread2471856.html>.

40. Цик Бен - Ган - Microsoft SQL Server 2008. Основи T-SQL.

41. Карвіш Б.К 21 Програмування баз даних SQL. Типові помилки та їх усунення Б. Карвін. - М: Рід Груп, 2012 р.

42. 1. Троелсен, Ендрю. Т70 Мова програмування C# 2010 та платформа .NET 4.0, 5-е вид. : Пров. з англ. - М.: ТОВ "І.Д. Вільямс", 2011 р.

ДОДАТКИ

Додаток А

Команди створення таблиць БД

```
CREATE TABLE [dbo].[Відвідування](
    [id_Відвідування] [int] IDENTITY(1,1) NOT NULL,
    [id_Студента] [int] NULL,
    [id_Дисципліни] [int] NULL,
    [25.05.2021] [nvarchar](50) NULL,
    [28.05.2021] [nvarchar](50) NULL,
    [30.05.2021] [nvarchar](50) NULL,
    [ 03.06.2021] [nvarchar](50) NULL,
```

```
REATE TABLE [dbo].[Групи](
    [id_Групи] [int] IDENTITY(1,1) NOT NULL,
    [id_Спеціальності] [int] NULL,
```

```
CREATE TABLE [dbo].[Групи_связь](
    [id_Групи] [int] NULL,
    [Назва_Групи] [nchar](10) NULL,
    [id_Студента] [int] NULL
```

```
CREATE TABLE [dbo].[Дисципліни](
    [id_Дисципліни] [int] IDENTITY(1,1) NOT NULL,
    [Назва_Дисципліни] [nchar](20) NULL,
```

```
CREATE TABLE [dbo].[Кафедри](
    [id_Кафедри] [int] IDENTITY(1,1) NOT NULL,
    [Назва_Кафедри] [nchar](35) NULL,
    [id_Факультету] [int] NULL,
```

```
CREATE TABLE [dbo].[Спеціальності](
    [id_Спеціальності] [int] IDENTITY(1,1) NOT NULL,
```

```
[Назва_Спеціальності] [nvarchar](25) NULL,  
[id_Кафедри] [int] NULL,
```

```
CREATE TABLE [dbo].[Студенти](  
    [id_Студента] [int] IDENTITY(1,1) NOT NULL,  
    [ПІБ] [nvarchar](30) NULL,  
    [Номер_телефону] [nvarchar](10) NULL,  
    [Електронний_адрес] [nvarchar](40) NULL,  
    [Форма_Нвчання] [nvarchar](6) NULL,  
    [id_Освіти] [int] NULL,
```

```
CREATE TABLE [dbo].[Ступінь_Освіти](  
    [id_Освіти] [int] IDENTITY(1,1) NOT NULL,  
    [Назва_Освіти] [nvarchar](15) NULL,
```

```
CREATE TABLE [dbo].[Факультети](  
    [id_Факультету] [int] IDENTITY(1,1) NOT NULL,  
    [Назва_Факультету] [nvarchar](45) NULL,
```