

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Бабій Ростислав Степанович

**Програмний засіб оптимізації роботи куратора
групи / The software for the group curator work
optimization**

напрямок підготовки: 123 Комп'ютерна інженерія
фахове спрямування - Комп'ютерна інженерія
Бакалаврська робота

Виконав студент групи КСМ-43/2
Р.С. Бабій

Науковий керівник:
Івасьєв С.В.

Тернопіль - 2018

РЕЗЮМЕ

Дипломний проект містить 77 сторінок пояснюючої записки, 32 рисунок, 12 таблиць, 2 додатки та 2 аркуші формату А3.

Метою цього дипломного проекту є створення бази даних електронний журнал куратора академічної групи використовуючи систему управління базами даних Microsoft Access, та програмного додатку для її управління. Який повинен буде автоматизувати роботу куратора у ведені журналу роботи з групою.

Microsoft Access, за допомогою якої була створена база даних - це функціонально повна реляційна система управління базою даних. В ній передбачені всі необхідні засоби для визначення і обробки даних, а також для управління ними при роботі з великими об'ємами інформації.

Розроблена база даних з програмним додатком є ефективним засобом з простим інтерфейсом, що дозволяє значно спростити роботу куратора при ведені електронного журналу роботи з групою.

Розроблений програмний додаток для роботи з базою даних електронний журнал куратора академічної групи може використовуватись у роботі наставників академічних, спортивних та творчих груп. Для прослідковування виконання плану роботи і успіхів учасників групи.

Ключові слова: БАЗА ДАНИХ, РОЗРОБКА, СЕРЕДОВИЩЕ DELPHI, ОБ'ЄК, МОВА SQL, ВІЗУАЛЬНІ КОМПОНЕНТИ, ЕЛЕКТРОННИЙ ЖУРНАЛ.

RESUME

Diploma project contains 77 pages of explanatory notes, 32 figures, 12 tables, 2 additions and 2 sheets of A3 .

The aim of this diploma project is to create a database of e-zine curator academic group using a database management system Microsoft Access, and software applications for its management who will have to automate the work supervisor to work on the magazine led group.

Microsoft Access, with which was created database - a functionally complete relational database management system . It provides all the necessary tools to determine i data , a well as management when dealing with large amounts of information.

Developed a database of software application is an effective tool with a simple interface that can significantly simplify the curator at the electronic journal led work group.

The developed software application for use with electronic journal database curator academic group may be used in the Mentor academic , sporting and creative groups. To trace the execution of the work plan and progress of the group.

Keywords: DATABASE, DEVELOPMENT, ENVIRONMENT DELPHI, OBJECTIVITY, LANGUAGE SQL, VISUAL COMPONENTS, ELECTRONIC JOURNALS.

ЗМІСТ

Перелік умовних скорочень.....	9
Вступ.....	10
1 Аналіз предметної області.....	11
1.1 Обов'язки куратора групи.....	11
1.2 Аналіз існуючих проектних рішень.....	16
1.3 Постановка завдання	19
2 Дослідження методів проектування.....	21
2.1 Обґрунтування вибору проектних рішень.....	21
2.2 Методи роботи з ODBC.....	24
2.3 Проектування бази даних.....	28
3 Проектування та розробка програмного засобу.....	35
3.1 Розробка графічного інтерфейсу	35
3.2 Розробка функцій для роботи з БД.....	37
3.3 Тестування та верифікація програмного засобу.....	42
4 Техніко-економічний розділ.....	45
4.1 Розрахунок витрат на розробку програмного додатку.....	45
4.2 Визначення експлуатаційних витрат.....	50
4.3 Визначення економічної ефективності і терміну окупності капітальних вкладень	54
Висновки	56
Список використаних джерел	57
Додаток А Код програмного засобу.....	61
Додаток Б Довідка про використання.....	72

					ДП.КСМ.07236/16.00.00.000.ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата			
Розробив		Бабій Р.С.			Літ.	Арк.	Акрушів
Перевір.		Івасьєв С.В.			8	61	
Консульт.		Паздрій І.Р.			ТНЕУ.ФКІТ. КСМ-43/2		
Н. Контр.		Гураль І.В.					
Затвердив		Березький О.М.					

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- ВНЗ – Вищий навчальний заклад.
БД – База даних.
СУБД – Система управління базами даних.
ЕОМ – Електронна обчислювальна машина.
ПК – Персональний комп'ютер.
SQL – Структурована мова запитів.
КС – Комп'ютерна система.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Основні ідеї сучасної інформаційної технології базуються на концепції, згідно якої дані повинні бути організовані в бази даних з метою адекватного відображення реального світу, що змінюється, і задоволення інформаційних потреб користувачів. Ці бази даних створюються і функціонують під управлінням спеціальних програмних комплексів, званих системами управління базами даних.

Збільшення об'єму і структурної складності даних, що зберігаються, розширення круга користувачів інформаційних систем привели до широкого розповсюдження найбільш зручних і порівняно простих для розуміння реляційних (табличних) СУБД.

Таким чином, на сьогодні розробник не зв'язаний рамками якого-небудь конкретного пакету, а залежно від поставленого завдання може використовувати самі різні додатки. Тому, важливішим представляється загальний напрям розвитку СУБД і інших засобів розробки додатків в даний час.

Темою мого дипломного проекту є розробка бази даних електронний журнал куратора академічної групи. Створена програма володіє рядом переваг:

- автоматизація ведення журналу;
- легкий спосіб введення або видалення інформацію про роботу куратора;
- зручний дизайн програми із зрозумілим інтерфейсом і оригінальними кнопковими формами;
- можливість перегляду або висновку на друк звітів роботи куратора.

Метою цього дипломного проекту є створення бази даних в СУБД Microsoft Access та програмного додатку, який повинен буде автоматизувати роботу куратора в веденні журналу роботи з групою. Що дасть змогу покращити роботу куратора і прослідкувати виконання запланованого плану та успіхів студентів.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обов'язки куратора групи

Основна мета виховання у вищому професійному навчальному закладі - це формування «високоморальної, духовно розвиненої і фізично здорової особистості, здатного до високоякісної професійної діяльності та відповідальності за прийняті рішення» [6]. У роботах, присвячених дослідженню виховної складової діяльності вузу, ключовою фігурою, здатною об'єднати викладачів і студентів при вирішенні виховних завдань, а також забезпечити ефективне формування загальнокультурних компетенцій однозначно визнається куратор академічної групи [1].

Сьогодні інститут кураторства переживає період активного наукового теоретичного осмислення і впровадження в практику.

У рекомендаціях щодо організації виховного процесу в вузі прописані основні напрямки діяльності кураторів студентських груп:

- спрямованість на успішну адаптацію студентів молодших курсів (I-II) до умов навчання у вузі,
- реалізація прав і обов'язків студентів,
- надання сприяння в духовно-моральному і в професійному становленні особистості майбутнього фахівця.

Основні обов'язки куратора групи:

1. вивчення та аналіз соціологічних і психологічних даних про студентів, їх здібностях і індивідуальні особливості;
2. планування та реалізація спільно зі студентським активом виховних заходів, формування організаторських умінь і навичок, обрання старости навчальної групи, а також представників в органи студентського самоврядування, "співуправління", студентські громадські організації;
3. вивчення та аналіз соціально-психологічного клімату в студентській групі, створення атмосфери довіри, взаємодопомоги та співробітництва в навчальній групі;

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

4. забезпечення реалізації завдань виховної діяльності в групі [6].

Перелічені напрямки і обов'язки є лише загальні положення діяльності куратора. Очевидно, що цього недостатньо для організації ефективної роботи куратора з групою на місцях. В даний час немає загального для всіх українських вузів положення про кураторства. Кожен вуз, покладаючись на нормативні документи про освіту, самостійно розробляє «Положення про кураторської діяльності», в якому конкретизує завдання і прописує повноваження куратора студентської групи.

З метою отримання відповідей на поставлені питання, ми провели якісно-кількісний аналіз «Положень про кураторської діяльності» 27-та вибраних навчання українських вузів. В результаті було виділено 29 найбільш часто зустрічаються приписів, що представляють собою зміст роботи куратора академічної групи. На наступному етапі ми класифікували виділені обов'язки куратора за напрямками і вираховали в процентному співвідношенні частоту народження кожної обов'язки в положеннях про кураторської діяльності в різних вузах. Отримані дані представлені в таблицях 1.1-1.2.

Більшість вузів, в першу чергу, ставлять перед кураторами завдання, пов'язані з реалізацією прав і обов'язків студентів, а також успішної адаптацією до умов навчання.

Таблиця 1.1 - Спрямованість на реалізацію прав та обов'язків студентів

Обов'язки куратора академічної групи	% вузів
Ознайомлення студентів з основними локальними документами щодо організації навчально-виховного процесу у ВНЗ, (Статут університету, Правила внутрішнього розпорядку вищого навчального закладу, Правила проживання в гуртожитку та ін.).	96,3
Контроль успішності студентів групи (поточної і семестрової), аналіз причин і допомогу в усуненні незадовільних результатів навчання.	100
Відвідування гуртожитків, моніторинг умов проживання студентів групи (не рідше одного разу на місяць). Сприяння у поліпшенні житлових умов у гуртожитку.	63

Продовження таблиці 1.1.

Обов'язки куратора академічної групи	% вузів
Надання допомоги студентам в планування самостійної роботи, виконання ними навчального графіка.	29,6
Інформування дітей-сиріт, інвалідів, малозабезпечених студентів, що навчаються в групі про їхні права та надання допомоги в їх реалізації.	7,4

Всі вузи зобов'язують кураторів ознайомити студентів з Статутом вузу і правилами організації навчально-виховного процесу (96,3%), здійснювати контроль за академічною успішністю в групі (100%), сформувати згуртований дружний колектив (88,9%), організовувати культурно-масові (88,9%) і науково-дослідні (77,8%) заходи (таблиця 1.2).

Таблиця 1.2 - Спрямованість на успішну адаптацію студентів до умов навчання у вузі

Обов'язки куратора академічної групи	% вузів
Допомога в згуртуванні колективу групи.	88,9
Надання допомоги в підготовці, проведенні та прийнятті безпосередньої участі в культурно-масових та фізкультурно-оздоровчих заходах (суботниках, недільниках).	88,9
Здійснення організаційної роботи по залученню студентів до науково-дослідницької роботи.	77,8
Підбір і призначення активу групи, безпосередню участь і допомогу в роботі старост.	70,4
Регулярне підтримання зв'язку з батьками студентів.	55,6
Надання допомоги у розвитку студентського самоврядування.	55,6
Сприяння сприятливим діловим відносинам між викладачами і студентами. У разі виникнення конфліктних ситуацій зуміти їх дозволити.	44,4

Створення атмосфери довіри, взаємодопомоги, співробітництва в групі, вибір активу групи, врегулювання конфліктних ситуацій вимагає вивчення

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

індивідуально-психологічних особливостей кожного студента. Однак лише 55,6% вузів ставлять перед кураторами цю задачу (таблиця 1.3).

Таблиця 1.3 - Вивчення та аналіз соціологічних і психологічних даних про студентів, їх здібностях і індивідуальні особливості

Обов'язки куратора академічної групи	% вузів
Вивчення та аналіз індивідуальних психолого-соціологічних особливостей студентів.	55,6
Знання умов життя і побуту кожного студента: його матеріальне, сімейний стан, здоров'я, забезпеченість житлом.	44,4

Проведене дослідження обов'язків куратора в різних вузах України дозволяє зробити наступні висновки:

1. вимоги державної політики в сфері виховної діяльності у вузі знаходять своє відображення в положеннях про кураторської діяльності. однак, користуючись правом самостійного визначення обов'язків куратора, вузи по-різному трактують пріоритетність тих чи інших напрямків у вихованні студентів;

2. до найбільш часто зустрічається в положеннях про кураторської діяльності обов'язків куратора відносяться (у напрямку зниження):

- контроль успішності студентів групи (100%);
- ознайомлення студентів з основними локальними документами вузу (96,3%);
- надання допомоги в підготовці та проведенні культурно-масових та фізкультурно-оздоровчих заходах (88,9%);
- допомога в згуртуванні колективу групи (88,9%);
- робота по залученню студентів до науково-дослідної роботи (77,8%).
- підбір і призначення активу групи (70,4%);
- відвідування гуртожитків (63,0%);
- регулярне проведення кураторських годин (63,0%);
- інформування деканату про становище в академічній групі (55,6%);

- вивчення та аналіз індивідуальних психолого-соціологічних особливостей студентів (55,6%);

- надання допомоги в розвитку студентського самоврядування (55,6%);

- регулярне підтримування зв'язку з батьками студентів (55,6%).

3. менш ніж в 50% локальних документах вузів прописується в якості обов'язків для кураторів:

- сприяння сприятливим діловим відносинам між викладачами і студентами. У разі виникнення конфліктах ситуацій зуміти їх дозволити;

- знання умов життя і побуту кожного студента: його матеріальне, сімейний стан, здоров'я, забезпеченість житлом;

- попередження девіантної поведінки і пропаганда здорового способу життя;

- формування загальнолюдських норм гуманістичної моралі (доброти, взаєморозуміння, професійної етики, толерантності і т.д.), культури спілкування;

- проведення індивідуальної роботи зі студентами у вигляді бесід.

4. Такий напрямок у виховній діяльності у вузі як сприяння в професійному становленні особистості майбутнього фахівця має бути прописано в документах всіх вузів. Однак, лише третина вищих шкіл ставить це завдання перед кураторами. Підкреслюючи той факт, що основна стратегічна задача вищої професійної освіти - допомогти студентові стати висококваліфікованим фахівцем, що володіє всіма досягненнями обраної професії, ми вважаємо, що роль куратора в рішенні цієї задачі недооцінена.

Отримані результати дозволяють виділити основні завдання, які стоять перед куратором академічної групи та виділити сутності для подальшого проектування та розробки системи автоматизованої обробки та зберігання даних для журналу куратора.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

1.2 Аналіз існуючих проектних рішень

При проектуванні нового програмного продукту необхідно розглянути ряд існуючих програмних засобів, які широко застосовуються для виконання завдань та цілей поставлених для автоматизації роботи куратора академічної групи. Серед найбільш розповсюджених є:

- «BioTime EDU»;
- «MyAttendanceTracker»;
- «Attendance», «Attendance Tracker».

Наведені програмні засоби відрізняються як структурою програмного забезпечення так і архітектурними особливостями і набором спеціальних функцій.

Серед відомих засобів моніторингу відвідування є система «BioTime EDU». Розробником програмного засобу є відома компанія «МЗ Biometrics». Система дозволяє проводити повну автоматизацію стеження за відвідуваністю студентами вищого навчального закладу

Система «BioTime EDU» базується на використанні біометричного сканера відпечатків пальця при вході в аудиторію вищого навчального закладу. Модулі системи забезпечують співставлення аудиторій з розкладом та навчальним планом студента. Завдяки інтелектуальному аналізу та побудові зв'язків система може виконувати контроль відвідуваності занять згідно індивідуального плану кожного студента. Вигляд біометричного сканера наведено на рисунку 1.1.



Рисунок 1.1 - Біометричний сканер системи «BioTime EDU»

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Система дозволяє автоматизувати процес складання індивідуального розкладу та співставляє розклад з відвідуванням кожним студентом власного індивідуального плану навчання чим забезпечує спрощення роботи викладача та куратора. Приклад роботи з розкладами наведено на рисунку 1.2.

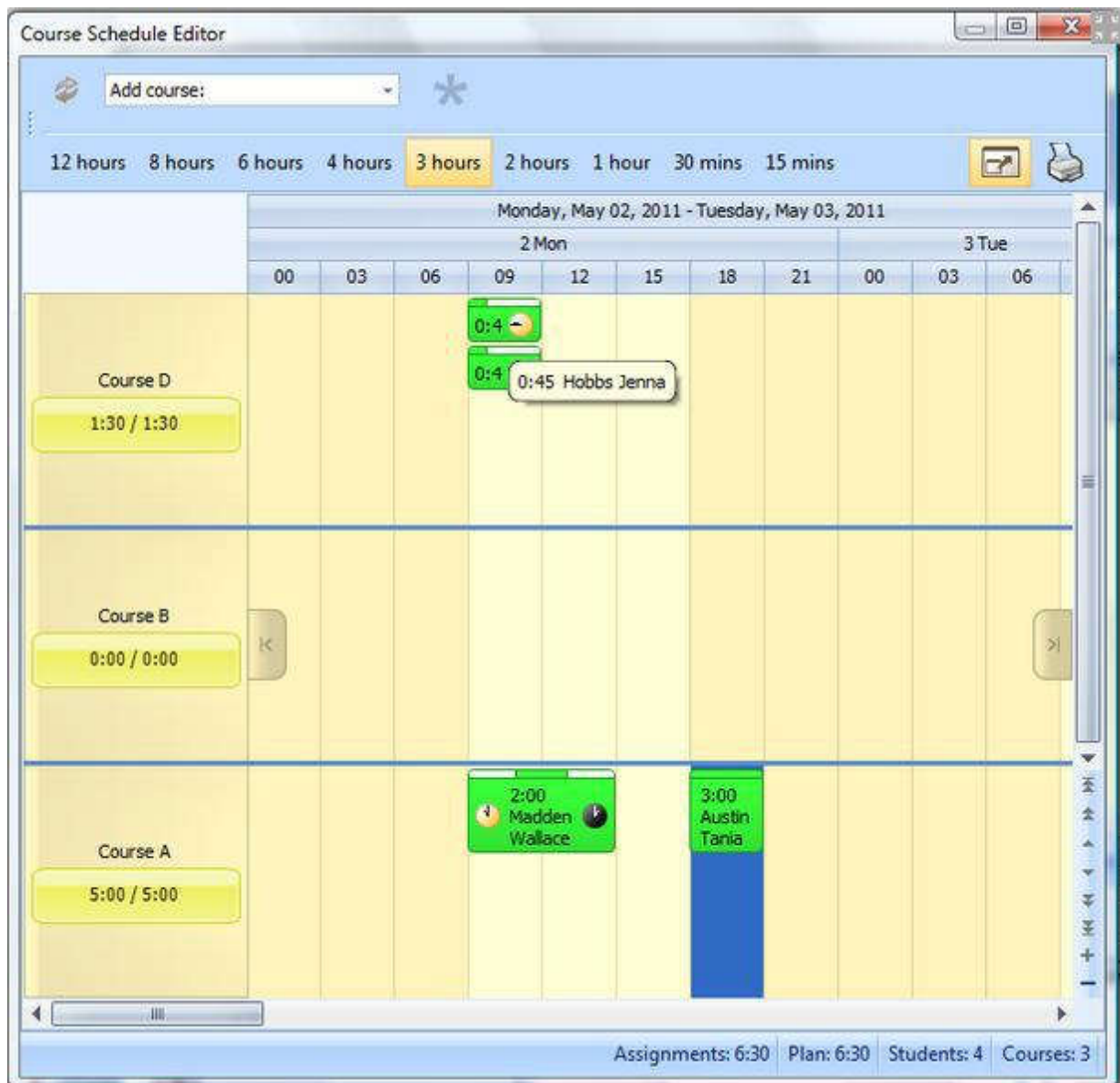


Рисунок 1.2 - Робота з розкладом

Система дозволяє вести облік особових справ студентів та при виявленні великої кількості пропусків автоматично генерує емейл попередження, як студенту так і відповідальним особам. Робота з особовими справами студентів наведена на рисунку 1.3.



Рисунок 1.3 - Редагування особових справ студентів

Необхідно також звернути увагу на досить поширену, веб-орієнтовану систему «MyAttendanceTracker». Система створена для ведення обліку відвідуваності занять студентами.

Після запуску програмного продукту відбувається ініціалізація майстра – помічника, який дозволить настроїти використання спеціальних функцій та модулів за необхідністю. Веб – орієнтовані системи мають значну перевагу над платформи залежними, оскільки можуть використовуватись на різноманітних мобільних пристроях. Головне вікно додатку приведене на рисунку 1.4.

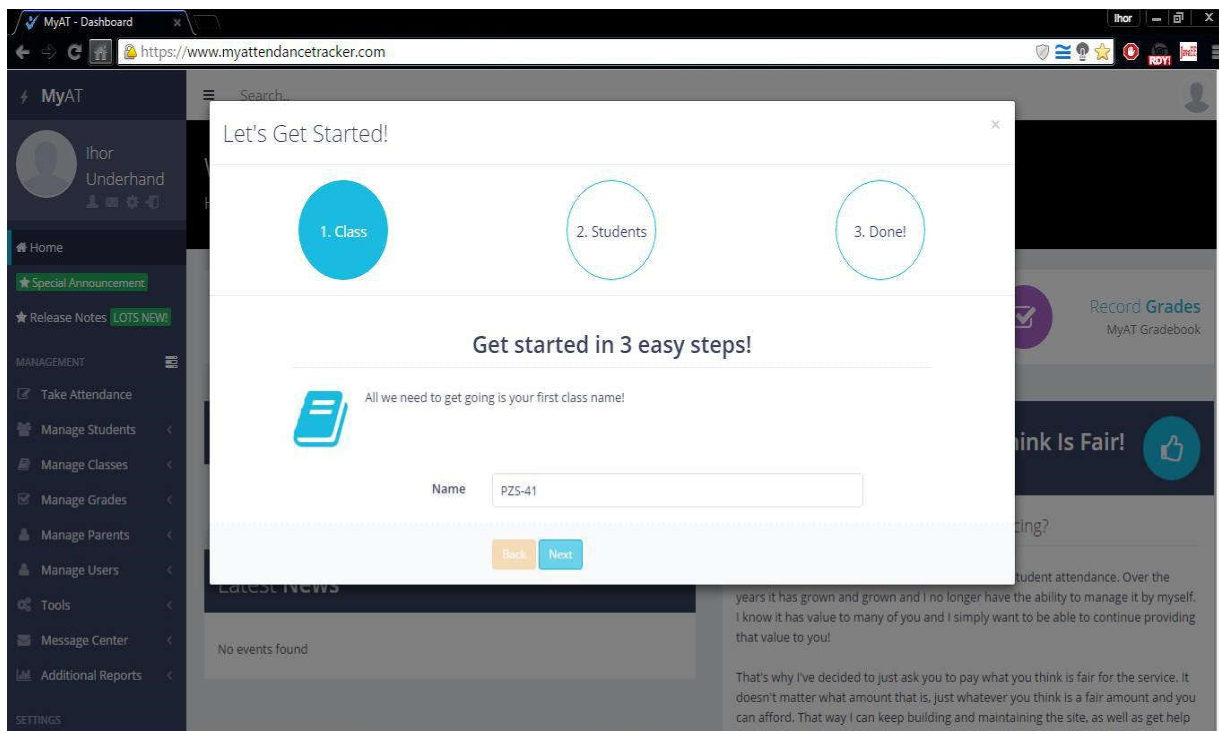


Рисунок 1.4 - Майстер початку роботи з веб-системою

Система дозволяє вносити інформацію про опікунів та відповідальних осіб, що забезпечує зворотній зв'язок, викладача та куратора групи з батьками студентів.

1.3 Постановка завдання

У роботі повинна бути розроблена інформаційна система аналізу успішності студентів. Ця тема є актуальною в даний час, так як багато навчальних закладів проводять аналіз успішності студентів для відстеження рівня підготовки і виявлення можливих недоліків у освітньому процесі.

Система аналізу успішності студентів повинна забезпечувати:

1. Пошук і сортування інформації:

- по кожному студенту;
- по групах;
- по факультетах;

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

- з кожної дисципліни.

2. Додавання даних про студентів.

3. Видалення даних.

4. Редагування наявних даних.

5. Формування карток і списків.

6. Побудова діаграм.

На практиці це означає виконання автоматизованої обробки аналізу успішності студентів; звільнення співробітників від ряду рутинних робіт з підготовки звітної документації; створення бази даних про студентів; здійснення операцій по створенню архівів успішності студентів. Завдяки автоматизації дозволяє шукати інформацію за запитом викладачів.

Для реалізації даної програми в рамках дипломного проектування, було обрано метод комп'ютерного проектування. Даний проект має оригінальну структуру і компонування.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

2 ДОСЛІДЖЕННЯ МЕТОДІВ ПРОЕКТУВАННЯ

2.1 Обґрунтування вибору проектних рішень

Використовуючи програми Delphi і Access для створення інформаційно-довідкових систем ми можемо відзначити як позитивні сторони цих програм, так і негативні. Але вибираючи зі списку існуючих програм, для програмування і БД, ми вибрали саме ці. Далі ми опишемо позитивні і негативні сторони даних програмних продуктів.

Базовий курс програмування орієнтований на роботу в Borland Delphi. Вибір середовища швидкої розробки додатків (RAD) Delphi обумовлений як тим, що це найбільш популярне середовище програмування на просторах Росії, так і безсумнівними об'єктивними достоїнствами Delphi. Серед цих переваг:

- простий і потужний мову програмування Pascal;
- зручна і повна об'єктна модель;
- досить зручне середовище розробки;
- велика об'єктна бібліотека VCL;
- потужні засоби розробки додатків баз даних.

Розглянемо ці особливості Delphi докладніше.

Мова програмування Паскаль (Pascal) був створений Н. Віртом в 1968 році спеціально для навчання студентів. Мова Паскаль був квінтесенцією технологій програмування середини 60-х - епохи створення структурного програмування. Мова був популярний в університетському середовищі Європи (в тому числі і в Радянському Союзі). Однак бідна і не цілком вдала система введення / виведення і відсутність підтримки розбиття програм на модулі не дозволили в той час Паскалю вийти за межі навчальних аудиторій. Однак прозорий і потужна мова послужив основою для багатьох експериментів в області мов програмування. На початку 80-х років з'явилася система програмування Turbo Pascal фірми Borland, в якій використовувався розширений діалект мови Паскаль з удосконаленою системою введення /

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

виводу і підтримкою модулів. Саме цей діалект мови і став найбільш поширеним, так як міг бути використаний в створенні сучасних великих додатків. В кінці 80-х років в діалект мови Паскаль від Borland були додані об'єктно-орієнтовані розширення, а 1992 році Borland Pascal став підтримувати програмування для MS Windows. При цьому Паскаль зберіг простоту і виразну потужність навчальної мови.

Об'єктно-орієнтовані засоби мови Паскаль пройшли великий шлях від найпростіших об'єктних розширень в TurboPascal 5.5 до сучасної об'єктної моделі Delphi, яка не поступається за своїми потенційними можливостями жодному об'єктно-орієнтованому мови. Об'єктна модель Delphi підтримує всі кошти які можуть знадобитися при об'єктно-орієнтованому програмуванні. Крім того, Delphi підтримує об'єктні технології Microsoft і, отже, може використовуватися для створення додатків і компонент COM / DCOM, а в останніх версіях і dotNET.

Але основною перевагою Delphi є його середовище програмування з візуальним конструктором програм. Це середовище (разом з об'єктною бібліотекою VCL) дозволяє ефективно програмувати під MS Windows, не відволікаючись на з'ясування всіх деталей Win API, а працювати над логікою програми.

З точки зору професійного програміста не менш важливим є те, що Delphi володіє високорозвиненими засобами для роботи з базами даних. Бази даних дозволяють ефективно управляти інформацією. Delphi дозволяє ефективно створювати додатки, що працюють з базами даних.

Вибір системи управління баз даних (СУБД) являє собою складну багатопараметричну завдання і є одним з важливих етапів при розробці додатків баз даних. Обраний програмний продукт повинен задовольняти як поточним, так і майбутнім потребам підприємства, при цьому слід враховувати фінансові витрати на придбання необхідного обладнання, самої системи, розробку необхідного програмного забезпечення на її основі, а також навчання персоналу. Крім того, необхідно переконатися, що нова СУБД здатна принести підприємству реальні вигоди.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

Під час написання дипломної роботи розглянуті основні особливості СУБД MS Access, її функціональні можливості і на підставі цієї інформації можна визначити сферу ефективного застосування цієї СУБД.

Основним фактором є використання платформи фірми Microsoft - операційної системи Windows. Хоча MS Access застосовується тільки під Windows, широке розповсюдження цієї ОС не є перешкодою для масового використання.

Найбільш значущі характеристики сфери застосування Microsoft Access полягають в наступному:

- наявність деякого обсягу інформації, який необхідно систематизувати і зберігати централізовано. Цей обсяг інформації не повинен перевищувати певної межі, наприклад, великі корпоративні системи, сховища даних OLTP або системи OLAP - це не варіант використання Access; обмеженість матеріальних і трудових ресурсів для розробки системи зберігання даних. В цьому плані Access надає досить великі можливості за невелику вартість. Ціна розробки та супроводу даної СУБД цілком доступна навіть малим компаніям або підприємцям; використання даних переважно в режимі одного користувача доступу. Можливо, звичайно, збільшення числа одночасно працюючих користувачів, за рахунок можливостей ядра Access, проте при великих кількостях використання Access недоцільно, зважаючи на пропорційного збільшення мережевого трафіку, і зниження швидкості обробки даних;

- відсутність жорстких вимог по захисту інформації. Access дозволяє захищати дані лише на призначеному для користувача рівні. Чого, однак, цілком вистачає для захисту від перегляду або необережних дій в середовищі співробітників невеликої компанії. Звідси зробимо висновок, що довіряти MS Access секретні або дуже цінні дані не варто; наявність потреби публікувати дані в мережі Intranet або Internet, або редагувати їх за допомогою браузера.

- фактори швидкості роботи програм не є критичними. Якщо програма створена в Access не варто очікувати від неї високої продуктивності. Однак, існує величезне коло завдань, в яких основний час при роботі з програмою доводиться на очікування дій користувача. В такому випадку невисока

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

швидкість програми абсолютно не помітна; при використанні MS Access виключається необхідність мати таку посаду, як адміністратор БД. Всі адміністративні операції прості і автоматизовані; поширеність ОС Windows, простота і стандартизованість інтерфейсу, наявність великої кількості зручних Майстрів роблять Access дуже дружній для більшості користувачів. І з іншого боку можна покладатися на стабільність виробника, компанія Microsoft є одним зі світових лідерів у виробництві ПО і забезпечує свої продукти докладною документацією, технічною підтримкою і локалізацією.

2.2 Методи роботи з ODBC

ODBC - Open DataBase Connectivity (вільний переклад - відкрита система зв'язку з базами даних). Open (відкрита) в даному випадку означає "розширювана", "легко нарощувана", "відкрита для модифікацій". Архітектура додатку з використанням ODBC приведена на рисунку 2.1.

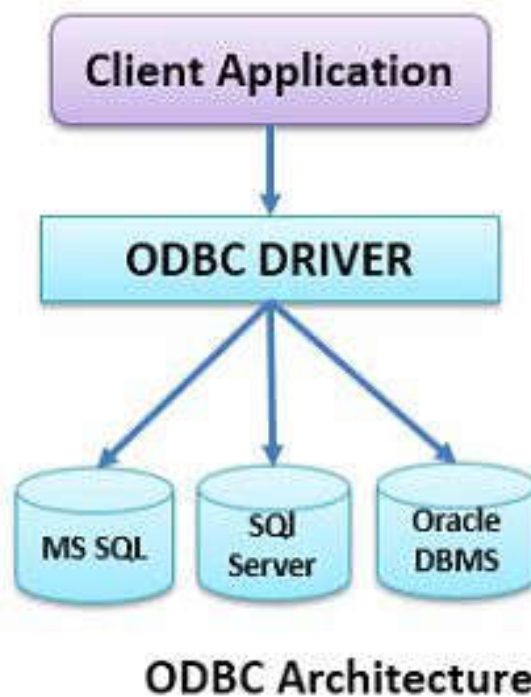


Рисунок 2.1 - Архітектура ODBC драйвера

Для додавання нового типу БД потрібно написати драйвер і зареєструвати його. Ще одна перевага, що впливає з такого побудови ODBC - користувальницький додаток спілкується з фізичною БД через менеджер драйверів, фактично нічого не знаючи про тип використовуваної БД (спілкування менеджера драйверів і самих драйверів відбувається стандартно, так що з точки зору користувача всі джерела даних мають практично однакові властивості). Таким чином, можна поміняти фізичний тип бази даних, а додаток не доведеться переписувати (звичайно, існують винятки з-за особливостей підтримки мови SQL різними типами БД, але вони несуттєві). ER – діаграма розробленої бази даних приведена на ДП.КСМ.07236/16.00.00.000 С1.

Фізично ODBC являє собою набір динамічних бібліотек DLL, які обслуговують підключення і роботу з конкретним типом бази даних. При запиті на підключення до певної, заздалегідь описаної бази "активізується" певний DLL - драйвер цього типу БД. Звернення до певної бази даних відбувається по імені так званого джерела даних ODBC (або DSN - data source name). Можна представити DSN як свого роду оголошення БД на даному комп'ютері, причому DSN може бути призначеним для користувача, тобто знаходиться в області видимості тільки одного користувача, в сеансі якого був створений цей DSN, і системним, тобто видимим всіма користувачами даного комп'ютера.

Для практичної задачі досить буде всього використовувати лише чотири системні функції:

1. Функція SQLDataSources повертає список всіх джерел даних ODBC.

RETCODE SQLDataSources (HENV hEnv, UWORD fDirection, UCHAR * szDSN,

SWORD cbDSNMax, SWORD * pcbDSN, UCHAR * szDescription,

SWORD cbDescriptionMax, SWORD * pcbDescription);

– hEnv - вказівник, що повертається функцією SQLAllocEnv ();

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

- fDirection - задає режим роботи. Може приймати значення SQL_FETCH_FIRST - повернути перший запис і SQL_FETCH_NEXT - повернути наступний запис. Логіка роботи аналогічна функцій FindFirst і FindNext;

- szDSN - покажчик на буфер довжиною не менше cbDSNMax, куди буде записано ім'я джерела даних;

- cbDSNMax - довжина буфера szDSN;

- pcbDSN - покажчик на ціле, куди записується реальна довжина (без урахування завершального знаку \0) прийнятого імені джерела даних;

- szDescription - покажчик на буфер довжиною як мінімум 255 символів, куди записується опис прийнятого джерела даних;

- cbDescriptionMax - довжина буфера szDescription;

- pcbDescription - покажчик на ціле, куди записується реальна довжина опису джерела даних.

2. Функція SQLConfigDataSources додає, змінює або видаляє вказаний джерело даних.

BOOL SQLConfigDataSources (HWND hwndParent, WORD fRequest, LPCSTR lpszDriver, LPCSTR lpszAttributes)

- hwndParent - хендл батьківського вікна. Якщо не дорівнює NULL, відображається стандартний для даного типу джерела даних діалог настройки.

- fRequest - тип запитуваної операції. Може приймати наступні значення:

- ODBC_ADD_DSN (ODBC_ADD_SYS_DSN) - запит на створення нового призначеного для користувача (системного) джерела даних;

- ODBC_CONFIG_DSN (ODBC_CONFIG_SYS_DSN) - запит на зміну існуючого призначеного для користувача (системного) джерела даних;

- ODBC_REMOVE_DSN (ODBC_REMOVE_SYS_DSN) - видалення користувача (системного) джерела даних.

										ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
											26
Змн.	Арк.	№ докум.	Підпис	Дата							

- lpszDriver - покажчик на рядок, що містить ім'я драйвера, асоційованого з конкретним типом БД;

- lpszAttributes - покажчик на рядок атрибутів в формі ключове слово - значення. Кожна така пара повинна закінчуватися символом \0, в кінці рядка атрибутів повинен стояти подвійний символ \0.

Наведемо кілька атрибутів:

"DSN = MySource \0UID = MyLogin \0PWD = MyPass" - джерело даних з ім'ям MySource, логін - MyLogin, пароль – MyPass

Кожен драйвер "має" свої ключові слова, найчастіше застосовуються:

PWD - пароль

UID - логін

DSN - ім'я джерела даних

DATABASE - ім'я бази даних

SERVER - ім'я сервера, на якому знаходиться база даних

Для баз даних Microsoft Access для вказівки імені БД замість DATABASE використовується ключове слово DBQ.

3. Функція SQLAllocEnv створює новий хендл для використання в інших функціях і ініціалізує інтерфейс для роботи з ODBC

RETCODE SQLAllocEnv (HENV * hEnv)

hEnv - покажчик на змінну, яка буде зберігати створений хендл в разі вдалого виконання функції

4. Функція SQLFreeEnv видаляє заданий хендл і звільняє всю пам'ять, асоційовану з ним.

RETCODE SQLFreeEnv (HENV hEnv)

- hEnv - хендл, який потрібно видалити

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Управління джерелами даних ODBC (та й взагалі налаштуванням всієї системи ODBC) здійснюється за допомогою спеціальної програми - ODBC-адміністратора. Його можна знайти в різних версіях Windows:

- Windows 9x - сам виконуваний файл odbcad32.exe знаходиться в каталозі Windows \ System. Запустити його можна прямо або через Панель управління (значок "Джерела даних ODBC (32-біт)"). Якщо на комп'ютері встановлено BDE, необхідно запустити адміністратор BDE і з меню Object вибрати пункт ODBC Administrator.

- Windows 2000 - виконуваний файл odbcad32.exe лежить в каталозі WinNT \ System32, а запускати його можна через Панель управління -> Адміністрування -> Джерела даних ODBC.

2.3 Проектування бази даних

Для зберігання атрибутивної інформації був обраний формат MS Access, який має наступну особливість, всі таблиці зберігаються в одному файлі (на відміну від Paradox і Dbase) і не вимагає при цьому запущеного сервера, як, наприклад, Interbase. Необхідний також зв'язок з файлами форматів dbf і db для завантаження / вивантаження даних в / з БД.

Для роботи з BLOB полями розроблено наступну процедуру:

```
var
  AStream: TBLOBStream;
  Data: Integer;
begin
  // Відкриваємо таблицю (звичайний TTable)
  ATable.Open;
  // Створюємо потік.
```

									ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
										28
Змн.	Арк.	№ докум.	Підпис	Дата						


```

AStream:      =      TBLOBStream      (ATable.CreateBLOBStream
(ATable.FieldName ( 'Поле')));

// Що-небудь читаємо з нього.
AStream. Read (Data, SizeOf (Data));

// Звільняємо потік і закриваємо таблицю.

AStream.Free;
ATable.Close;
end;

```

При проектування було обрано компоненти ADO 2.6.

Також в програмному продукті використані таблиці dBase, які є одними з перших форматів БД, що з'явилися на ПК, завдяки чому з ними можуть взаємодіяти з багатьма СУБД - якщо і не на пряму, то по крайній мірі, маючи можливість імпортувати і експортувати дані в цьому форматі. Більш того, з деякими застереженнями, їх можна переглядати і редагувати навіть в електронних таблицях, наприклад, в Excel.

При проектуванні бази даних було виділено сутності, як це показано на ДП.КСМ.07236/16.00.00.001 С1. В результаті було створено структури основних таблиць. Структура таблиці, де зберігаються логіни та паролі приведена на рисунку 2.2.

Имя поля	Тип данных
id	Счетчик
user	Текстовый
password	Текстовый
id_role	Числовой

Рисунок 2.2 - Структура таблиці «authentification»

В базі даних для підвищення ступеня нормалізації створено таблицю, яка містить інформацію про групи, спеціальність та ідентифікатор куратора. Структура розробленої таблиці приведена на рисунку 2.3.

autentification		groups
Имя поля	Тип данных	
id	Счетчик	
groupe_name	Текстовый	
specialiti_id	Числовой	
id_teacher	Числовой	

Рисунок 2.3 - Структура таблиці «groups»

Для ведення обліку успішності та відвідування студентами занять згідно навчального плану розроблено структуру таблиці jurnal (рисунок 2.4).

autentification		groups	jurnal
Имя поля	Тип данных		
id	Счетчик		
mark	Числовой		
was_in	Логический		
description	Текстовый		
remark	Числовой		
id_studet	Числовой		
id_teacher	Числовой		
id_groupe	Числовой		

Рисунок 2.4 - Структура таблиці «jurnal»

Системою передбачено різні рівні доступу користувачів. Опис рівнів доступу зберігається в окремій таблиці та дозволяє зберігати в собі опис прав наданих для конкретного користувача (рисунок 2.5).

autentification		groups	jurnal	permissions_level_description_table
Имя поля	Тип данных			
id	Счетчик			
permissions_level_id	Числовой			
permissions_level_description	Текстовый			

Рисунок 2.5 - Структура таблиці «permissions_level_description_table»

На рисунку 2.6 приведена структура таблиці «permissions_table», яка дозволяє встановити доступ до конкретної таблиці та є службовою.

permissions_table	
Имя поля	Тип данных
id	Счетчик
id_role	Числовой
table_name	Текстовый
permissions_level	Числовой

Рисунок 2.6 - Структура таблиці «permissions_table»

Особисті дані кураторів та вчителів зберігаються в окремій таблиці person, структура якої приведена на рисунку 2.7.

permissions_table		person	
Имя поля	Тип данных		
id	Счетчик		
user_id	Числовой		
name	Текстовый		
famili	Текстовый		
father	Текстовый		
phone	Текстовый		
adress	Текстовый		

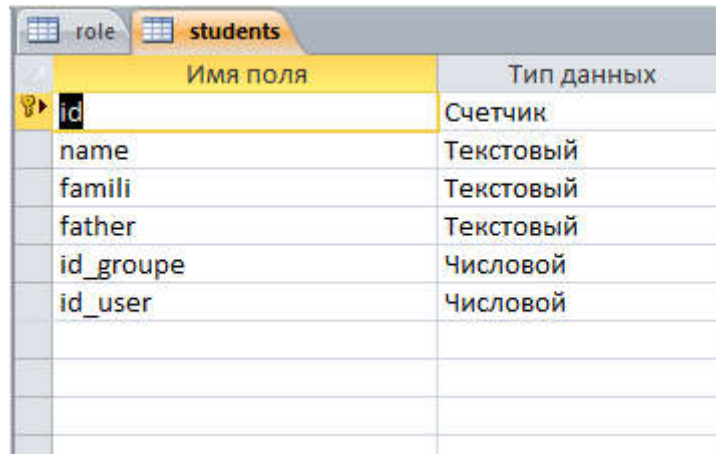
Рисунок 2.7 - Структура таблиці «person»

В системі передбачено систему рівнів користувачів, яка зв'язана з таблицями доступів та таблицею авторизації (рисунок 2.8).

role	
Имя поля	Тип данных
id	Счетчик
role_name	Текстовый
role description	Текстовый

Рисунок 2.8 - Структура таблиці «role»

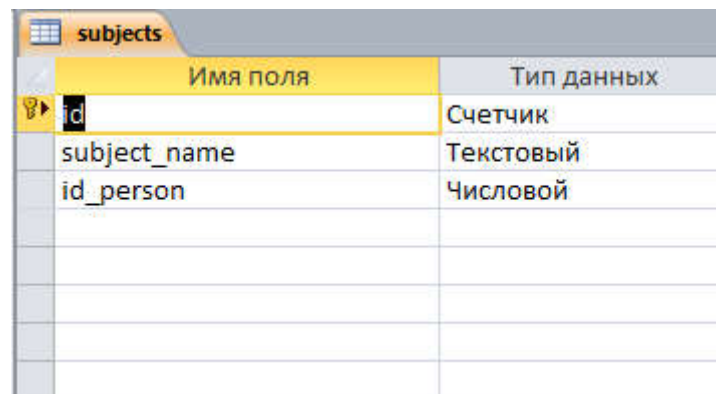
Інформація про студентів зберігається в таблиці students, та містить персональну інформацію про студентів. Структура розробленої таблиці приведена на рисунку 2.9.



Имя поля	Тип данных
id	Счетчик
name	Текстовый
famili	Текстовый
father	Текстовый
id_groupe	Числовой
id_user	Числовой

Рисунок 2.9 - Структура таблиці «students»

Для надання можливості кожному з студентів мати власний набір предметів та викладачів створена таблиця subjects, структура котрої приведена на рисунку 2.10.



Имя поля	Тип данных
id	Счетчик
subject_name	Текстовый
id_person	Числовой

Рисунок 2.10 - Структура таблиці «subjects»

У ранніх версіях Delphi замість dbExpress використовувалися SQL Links, але, починаючи з Delphi 7, ця система вважається застарілою. Як і інші SQL-орієнтовані компоненти, dbExpress і SQL Links є тільки у версіях Enterprise і Architect.

Незважаючи на те, що останнім часом компанія Borland посилено просуває SQL-орієнтований підхід для розробки БД і агітує за використання відповідних компонентів (зокрема, dbExpress), BDE залишається популярним засобом для розробки щодо нескладних БД. Такому стану речей сприяє не тільки зручність і простота цієї технології, але і велика кількість додатків, створених з використанням BDE, підтримку і розвиток яких потрібно здійснювати і сьогодні.

Для налаштування параметрів БД, підтримуваних BDE, а так само для створення псевдонімів (Aliases) до баз даних, використовують спеціальну утиліту, що входить в поставку Delphi - BDE Administrator. Робоча область цієї програми є блокнот з 2 сторінок - Databases і Configuration (рисунок 2.1).

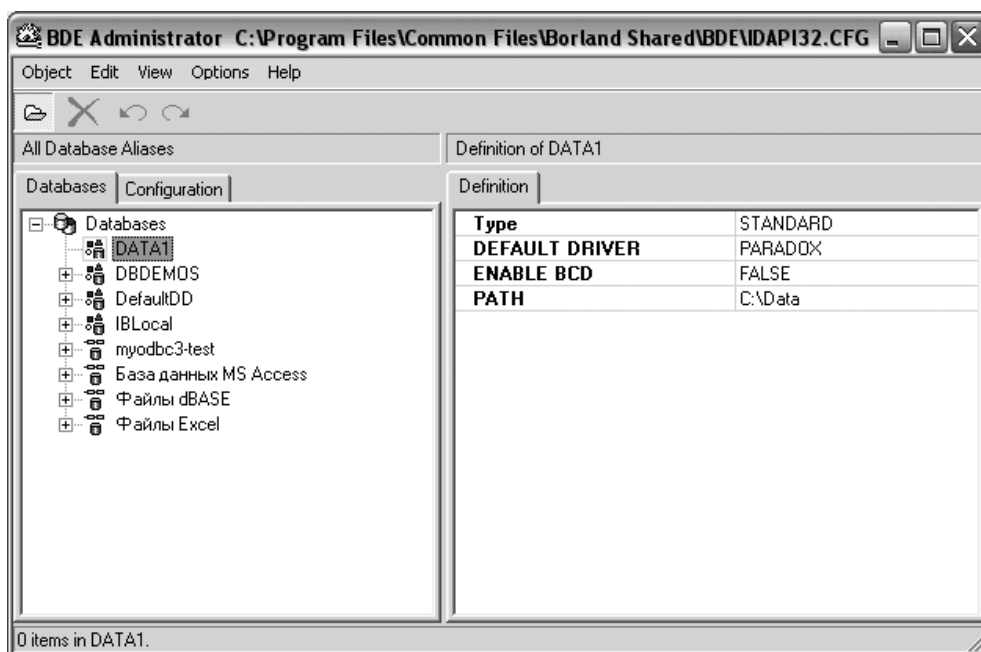


Рисунок 2.1 - Вікно BDE Administrator поточної сторінки Databases

На сторінці Databases розташований список наявних псевдонімів до БД. Відразу після установки Delphi і BDE створюється кілька псевдонімів, зокрема, DBDEMOS, DefaultDD і IBLocal. А щоб створити новий псевдонім, досить з меню Object вибрати пункт New. В результаті відкриється вікно, в якому ви зможете вибрати тип СУБД. За замовчуванням пропонується Standard, що зазвичай має на увазі використання Paradox як драйвер для створюваної БД. Після цього в списку з'явиться новий елемент, який буде названий

Standard1, а праворуч, в області визначення властивостей (Definition), будуть виведені всі доступні для редагування параметри. Стосовно до стандартного для Delphi типу це буде власне тип (Type), драйвер СУБД (Default driver), режим перетворення дійсних чисел (Enable BCD) і каталог, в якому розташовані файли БД (Path). Оскільки Delphi в якості "свого" формату підтримує не тільки Paradox, то в якості драйвера можна вказати так само формати dBase, FoxPro і ASCII (текстовий формат).

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

3.1 Розробка графічного інтерфейсу

При проектуванні програмного засобу було створено діаграму потоків даних системи, декомпозиція якої наведена на рисунку 3.1.

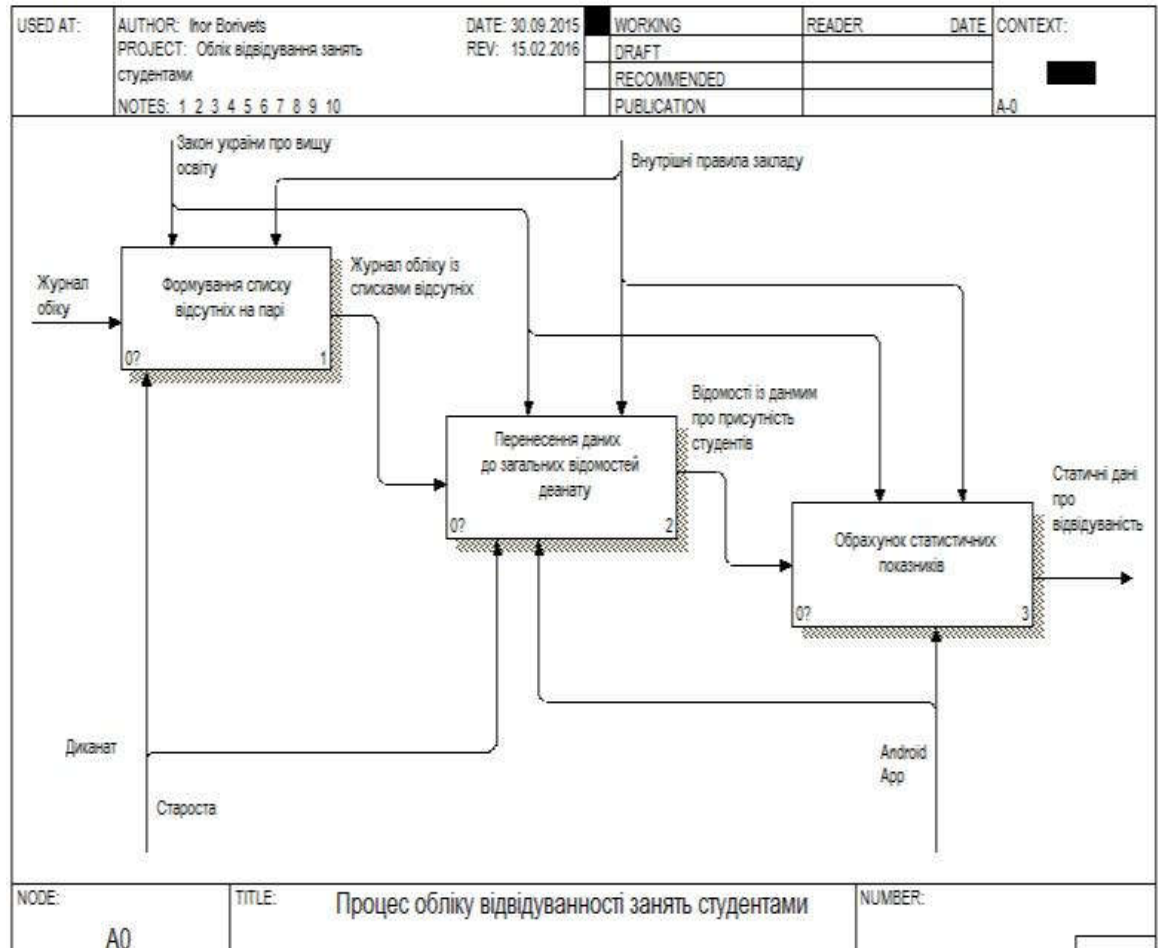


Рисунок 3.1 - Декомповована діаграма IDEF0

При проектуванні програмного засобу були використані наступні модулі: Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, DB, DBTables, Grids, DBGrids, Menus, ComCtrls;

Н рисунку 3.2 приведено головне вікно додатку на етапі проектування. Рисунок наведено з інтерфейсом інструменту розробки Delphi RAD Studio, яке надає широкі та гнучкі можливості створення програмних засобів.

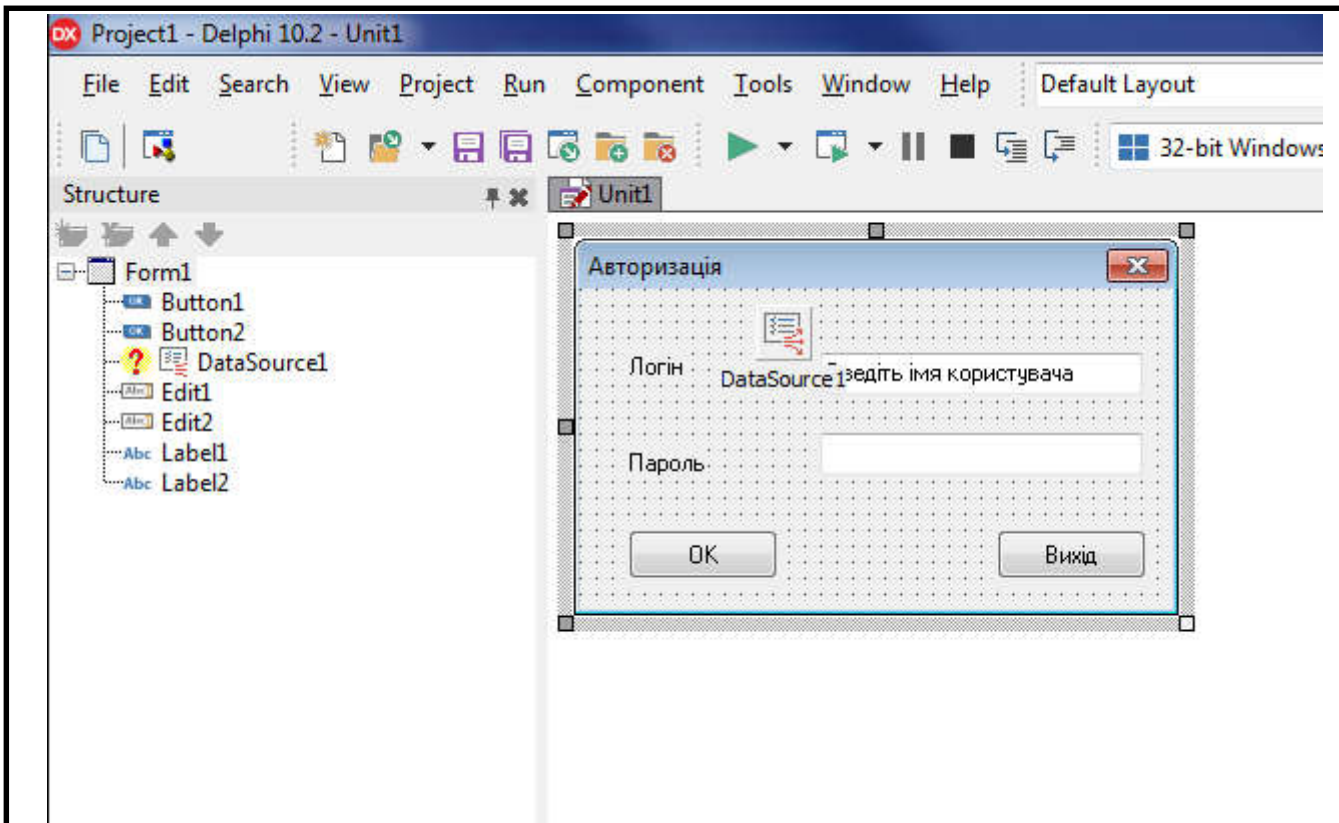


Рисунок 3.2 – Вікно авторизації на етапі проектування

Для технічних операцій головний адміністратор має доступ до усіх таблиць та дозвіл на повне їхнє редагування. Для роботи з журналом оцінок створена форма що на рисунку 3.3.

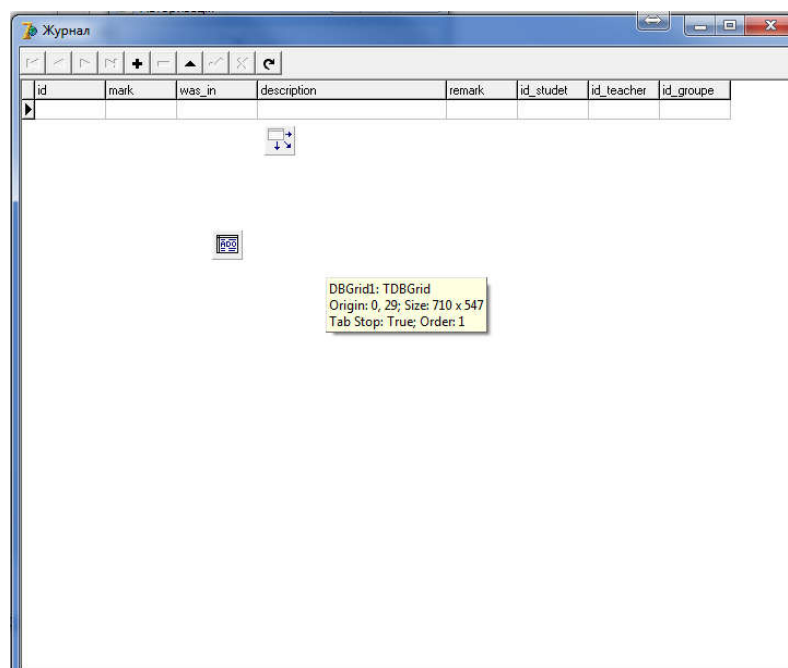
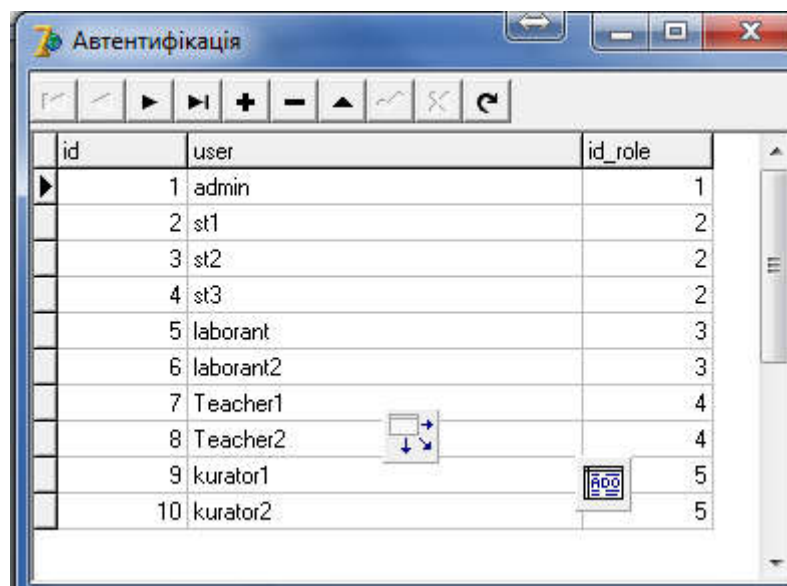


Рисунок 3.3 – Екранна форма для роботи з журналом оцінок

Для редагування даних про користувачів адміністратору розроблена форма, що на рисунку 3.4.

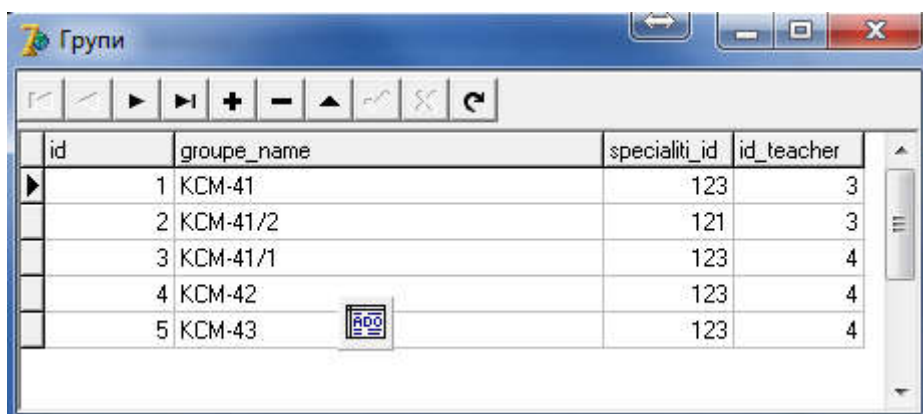


The screenshot shows a window titled "Автентифікація" with a table containing 10 rows of user data. The table has three columns: 'id', 'user', and 'id_role'. The data is as follows:

id	user	id_role
1	admin	1
2	st1	2
3	st2	2
4	st3	2
5	laborant	3
6	laborant2	3
7	Teacher1	4
8	Teacher2	4
9	kurator1	5
10	kurator2	5

Рисунок 3.4 – Таблиця авторизації

Для редагування груп створено форму, що на рисунку 3.5.

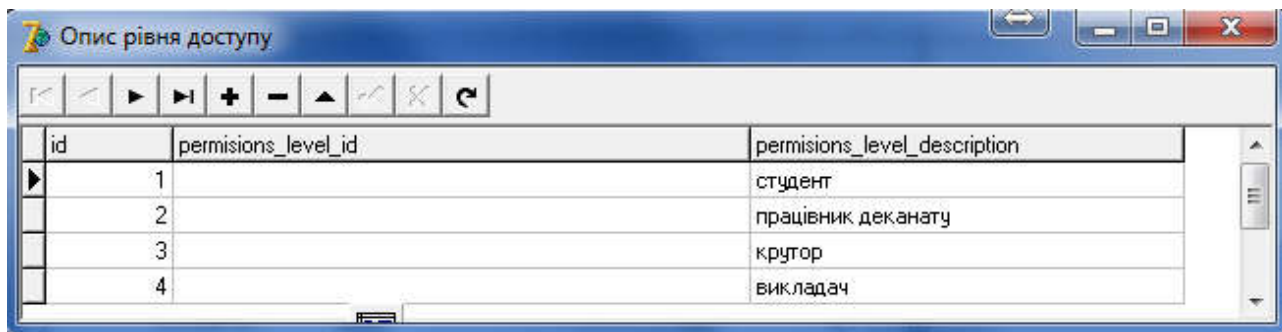


The screenshot shows a window titled "Групи" with a table containing 5 rows of group data. The table has four columns: 'id', 'groupe_name', 'specialiti_id', and 'id_teacher'. The data is as follows:

id	groupe_name	specialiti_id	id_teacher
1	KCM-41	123	3
2	KCM-41/2	121	3
3	KCM-41/1	123	4
4	KCM-42	123	4
5	KCM-43	123	4

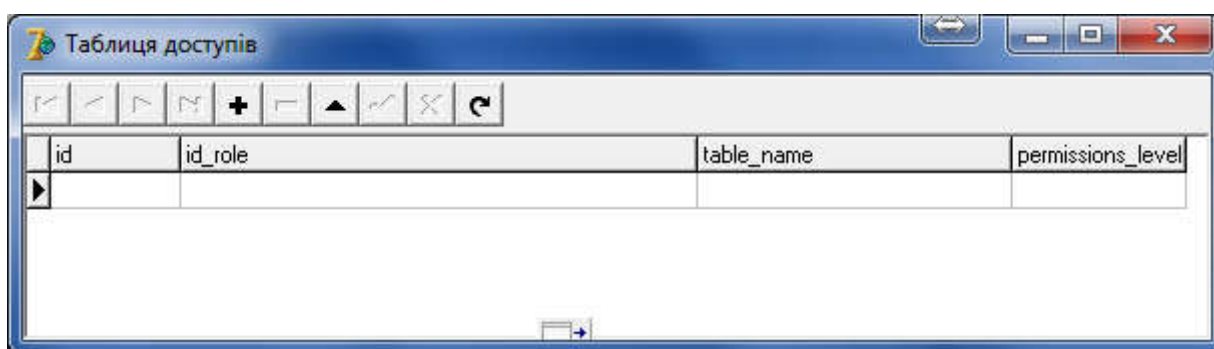
Рисунок 3.5 – Таблиця групи

Для роботи з рівнями доступу створено форму, що на рисунку 3.6 та 3.7. Ці форми дозволяють працювати з описами рівнів груп та встановленням доступів до окремих таблиць. Головний адміністратор при початковому налаштуванні може встановлювати права доступу до окремих таблиць для певних типів користувачів.



id	permissions_level_id	permissions_level_description
1		студент
2		працівник деканату
3		крутор
4		викладач

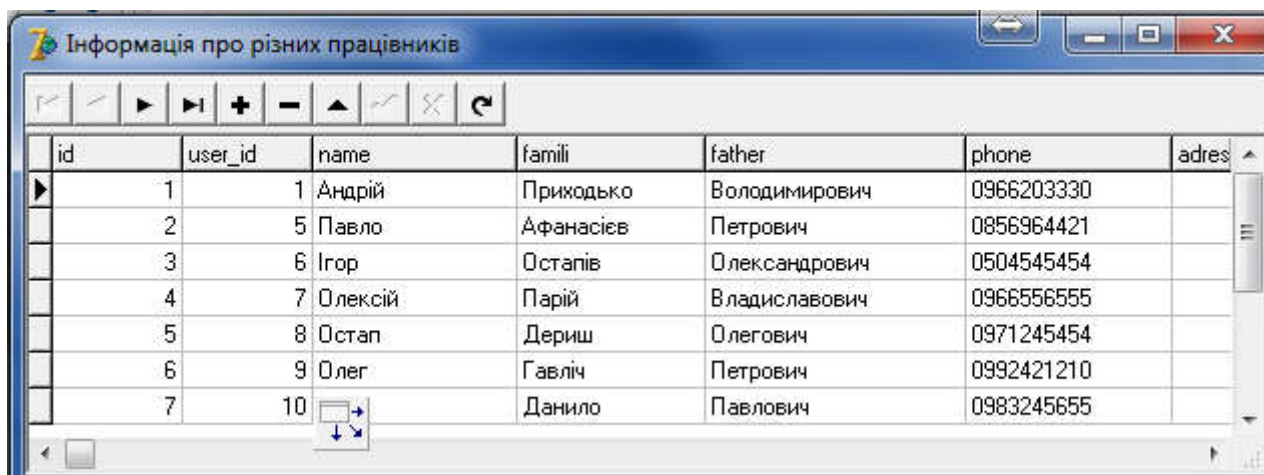
Рисунок 3.6 – Таблиця Опис рівня



id	id_role	table_name	permissions_level
----	---------	------------	-------------------

Рисунок 3.7 – Таблиця доступів

Для роботи з персональною інформацією про зареєстрованих в системі користувачів передбачено форму, що на рисунку 3.8.



id	user_id	name	famili	father	phone	adres
1	1	Андрій	Приходько	Володимирович	0966203330	
2	5	Павло	Афанасієв	Петрович	0856964421	
3	6	Ігор	Остапів	Олександрович	0504545454	
4	7	Олексій	Парій	Владиславович	0966556555	
5	8	Остап	Дериш	Олегович	0971245454	
6	9	Олег	Гавліч	Петрович	0992421210	
7	10		Данило	Павлович	0983245655	

Рисунок 3.8 – Робота з персональними даними користувачів системи

В системі встановлено ролі користувачів, для розподілу прав доступу та зручності адміністрування бази даних. Редагувати інформацію про ролі користувачів можна за допомогою форми, що на рисунку 3.9.

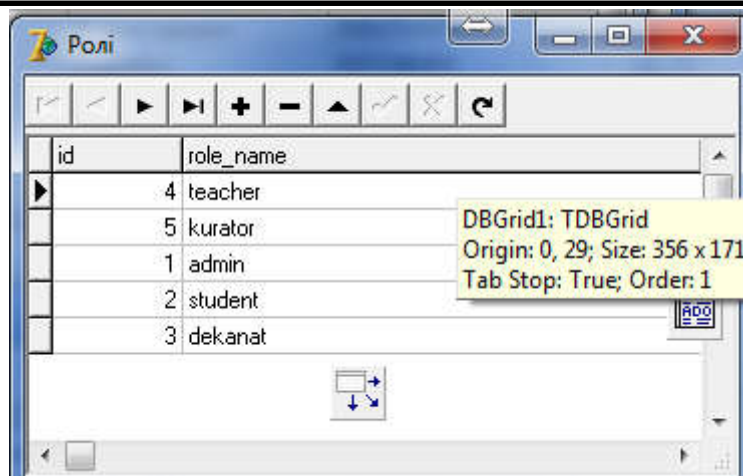


Рисунок 3.9 – Ролі користувачів

Інформація про самих студентів заповнює викладач або адміністратор, який має відповідні права доступу. Форма що це дозволяє приведена на рисунку 3.10.



Рисунок 3.10 – Інформація про студентів

Для відображення інформації про розробників та про сам програмний засіб створено форму, що приведена на рисунку 3.11.

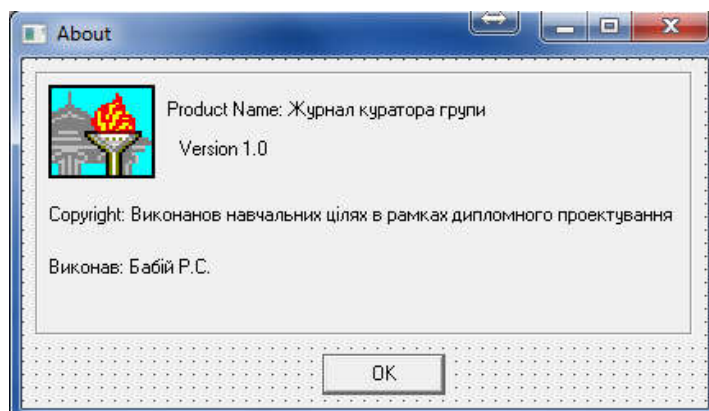


Рисунок 3.11 – Відомості про авторів

Для реалізації графічного додатку було розроблено форму, що представлена на рисунку 3.12.

Рисунок 3.12 - Головне вікно програмного засобу на етапі проектування

Для перегляду результатів здачі екзаменаційних відомостей було розроблено форму, що на рисунку 3.13.

Рисунок 3.13 – Форма екзаменаційних відомостей на етапі проектування

Для внесення екзаменаційних відомостей було розроблено форму, що на рисунку 3.4. Приведена форма дозволяє відображати екзаменаційну відомість розроблену для спеціального типу користувачів таких, як опікуни та батьки.

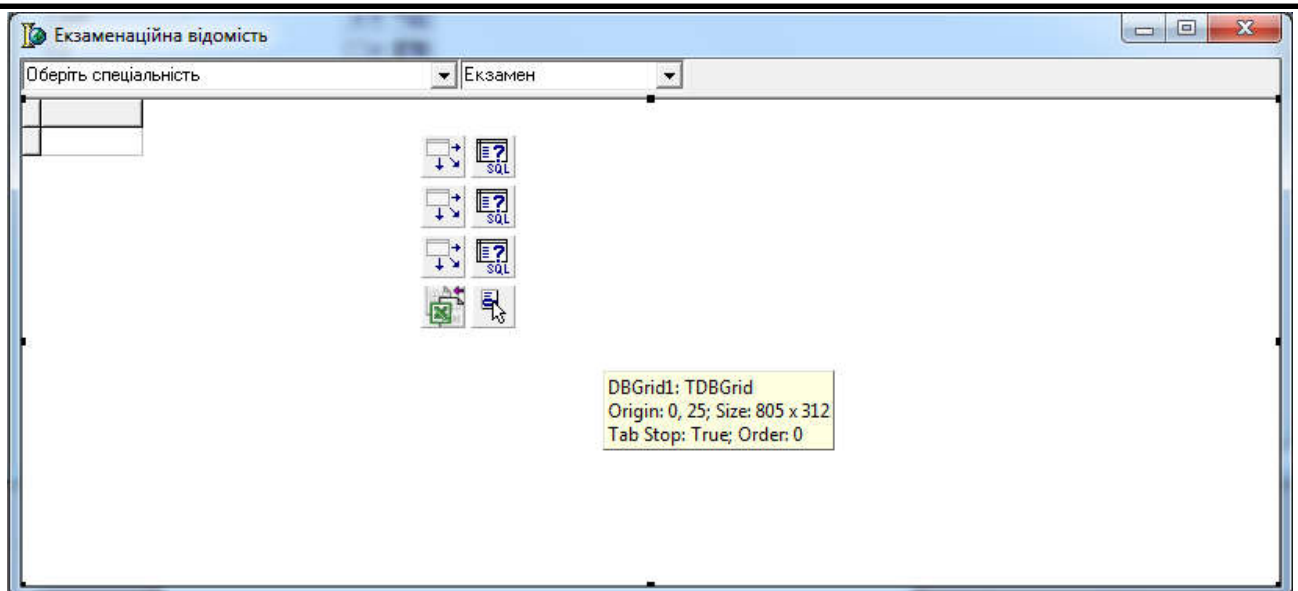


Рисунок 3.14 – Форма для внесення екзаменаційних відомостей на етапі проектування

Для внесення даних довідників та зв'язування таблиць для фільтрації та пошуку даних розроблена форма, що на рисунку 3.15.

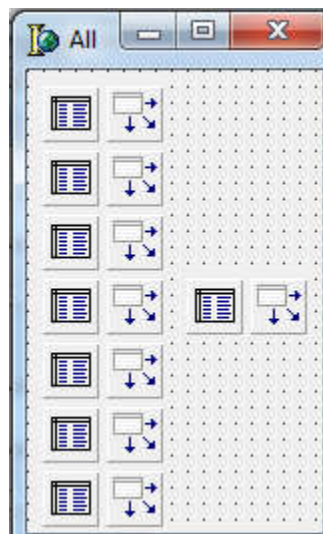


Рисунок 3.15 – Форма для внесення екзаменаційних відомостей на етапі проектування

Для створення додатків, що працюють з БД через BDE, в VCL передбачена група компонентів, розташована на закладці BDE палітри компонентів. З представлених 8 компонентів для нас зараз найбільший інтерес представляють 2, а саме Database і Table.

3.2 Розробка функцій для роботи з БД

Компонент Database представляє собою компонент, що інкапсулює в собі базу даних в цілому. З його допомогою можна контролювати наявність доступу до бази даних з програми, стежити за ходом транзакцій, а так само підключатися до клієнт-серверних БД. Властивості цього компонента наведені в таблиці 3.2.

Таблиця 3.2. - Властивості компонента Database

Властивість	Тип	Опис
AliasName	String	Визначає псевдонім BDE для підключення
Connected	Boolean	Визначає, встановлено чи ні з'єднання з БД
DatabaseName	String	Визначає ім'я БД, асоційоване з даним компонентом
Directory	String	Визначає робочий каталог для БД Paradox або dBase
DriverName	String	Визначає ім'я драйвера BDE для даної БД
Exclusive	Boolean	Включає монопольний доступ до БД
InTransaction	Boolean	Вказує, чи виконується в даний момент транзакція
KeepConnection	Boolean	Дозволяє вказати, чи додаток залишатися підключеним до БД, коли активних сполук немає
Params	TStrings	Визначає список додаткових параметрів для псевдоніма BDE
SessionName	String	Визначає ім'я сесії для даного компонента
ReadOnly	Boolean	Включає режим доступу до даних тільки на читання

Мабуть, найбільш важливою властивістю є DatabaseName: вказавши в якості значення цієї властивості один з уже визначених в BDE псевдонімів, ми отримаємо готовий до використання компонент. Ще одна властивість - Connected - відповідає за пряму лінію зв'язку з БД. Таким чином, вказавши в якості значення властивості DatabaseName "DATA1", і встановивши властивість Connected в істину, ми підключимося до БД.

Досить цікавою особливістю компонента Database є можливість створення псевдонімів БД, що діють в рамках створюваного додатка. Перш за

Змн.	Арк.	№ докум.	Підпис	Дата

все, можна вказати в якості значення властивості Alias те ж значення DATA1, вибравши його зі списку, а в якості DatabaseName вказати будь-яке довільне значення (наприклад, MyData). Таким чином, для інших компонент, мають властивість DatabaseName можна буде вказувати MyData як значення цієї властивості.

У той же час, якби у нас не було визначено псевдонім DATA1, або ж з яких-небудь причин нам не хотілося б його використовувати, то ми могли б "з нуля" створити псевдонім для цього додатка. Для цього буде потрібно вказати ті ж параметри, що і при створенні псевдоніма засобами BDE Administrator, а саме тип драйвера, шлях до БД і власне псевдонім. Код вийде приблизно таким:

```
Database1.DatabaseName:='MyData1'; // псевдонім
Database1.DriverName:='STANDARD'; // драйвер Paradox
Database1.Connected:=true; // активуємо компонент
Database1.Directory:='C:\Data'; // встановлюємо шлях до файлу даних
```

Разом з тим така властивість, як DatabaseName переважно все ж ставити не під час виконання, а в режимі розробки через інспектор об'єкта. Це дозволить посилатися на БД з інших компонент, вказуючи в їх властивості DatabaseName, так само вже на етапі візуального проектування програми.

Одним з таких компонент, що має властивість DatabaseName, як раз і є Table. Якщо Database являє собою базу даних в цілому, то Table - це уявлення одиначної таблиці з БД. Слід відразу зазначити, що Table може звертатися до таблиць БД як за посередництвом компонента Database, посилаючись на певне в ньому властивість DatabaseName, так і безпосередньо, якщо у властивості DatabaseName компонента Table вказати один з визначених у BDE псевдонімів. Зрозуміло, компонент Table має і інші властивості - вони наведені в таблиці 3.3.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.3. Властивості компонента Table

Властивість	Тип	Опис
Active	Boolean	Дозволяє вказати, чи бути встановлено підключення до бази даних
CanModify	Boolean	Вказує, чи може додаток змінювати вміст таблиці Paradox або dBase
DatabaseName	String	Визначає ім'я БД, асоційоване з даним компонентом
DataSource	TDataSource	Визначає ім'я об'єкта джерела даних для використання даної таблиці в якості підпорядкованої при зв'язку типу головний-підлеглий
DefaultIndex	Boolean	Дозволяє вказати, чи буде проводитися упорядкування записів таблиці з первинного індексу
Exclusive	Boolean	Включає монопольний доступ до даної таблиці Paradox або dBase
Exists	Boolean	Вказує, чи існує ця таблиця в БД
FieldDefs	TFieldDefs	Вказує на список полів, що визначають дані
Filter	String	Визначає умова, за яким буде відбуватися вибірка полів для показу
Filtered	Boolean	Визначає, чи є фільтр включеним
FilterOptions	TFilterOptions	Це впливає на доступність прапорів для фільтра
IndexDefs	TIndexDefs	Надає інформацію про індекси таблиці
IndexFieldCount	Integer	Вказує на кількість полів, використаних в поточному індексі
IndexFieldNames	String	Визначає список полів, які використовуються в якості індексу (через кому)
IndexFields	array of Tfield	Список індексів таблиці
IndexFiles	TStrungs	Визначає список файлів з індексами для таблиць dBase
IndexName	String	Визначає вторинний індекс, за яким має здійснюватися упорядкування (замість первинного)

Змн.	Арк.	№ докум.	Підпис	Дата

ДП.КСМ.07236/16.00.00.000.ПЗ

Арк.

44

Продовження таблиці 3.3.

MasterFields	String	Визначає список полів (через кому) в головній таблиці, за якими повинна встановлюватися зв'язок типу головний-підлеглий
MasterSource	TDataSource	Визначає ім'я об'єкта джерела даних для використання даної таблиці в якості головної при зв'язку типу головний-підлеглий
SessionName	String	Визначає ім'я сесії для даного компонента
ReadOnly	Boolean	Визначає режим доступу до таблиці
TableName	String	Визначає ім'я таблиці (назва файлу для таблиць dBase або Paradox)
TableType	TTableType	Визначає тип таблиці. Може приймати значення ttDefault (тип визначається по розширенню файлу), ttParadox, ttDBase, ttFoxPro і ttASCII

Тут слід зробити застереження, що насправді деякі властивості лише успадковані компонентом Table від своїх предків. Відповідно, ми можемо їх зустріти і в інших компонентах, пов'язаних з БД. Зокрема, це властивість Active, успадковане від класу TDataSet - спільного предка всіх БД-компонент, що надають безпосередній доступ до даних. Від цього ж предка відбувається і властивість FieldDefs, що дозволяє самостійно налаштувати список оброблюваних полів. А група властивостей, пов'язаних з відбором даних (Filter, Filtered і FilterOptions) є частиною класу TBDEDataSet - спадкоємця класу TDataSet, орієнтованого на роботу з базами даних за допомогою драйверів BDE.

Зупинимося на властивості FilterOptions, яке містить 2 прапора:

- foCaseInsensitive - рядки будуть порівнюватися без урахування регістру символів;
- foNoPartialCompare - символ "зірочка" (*) в поле фільтра буде інтерпретуватися саме як символ, а не як шаблон підстановки.

Властивості фільтрів використовуються тільки при роботі з BDE, оскільки в SQL-орієнтованих компонентах мається на увазі, що відбір даних

буде здійснюватися за допомогою запитів SQL, що, в общем-то, цілком природно, враховуючи призначення і можливості самого SQL.

Для використання таблиці досить вказати значення для властивостей DatabaseName і TableName. Наприклад, якщо продовжити приклад зі створенням псевдоніма для компонента Database, то після останнього рядка коду, що встановлює значення для властивості Directory, досить додати:

```
Table1.DatabaseName:='MyData1'; // встановлюємо значення БД на псевдонім
```

```
Table1.TableName:='Customer'; // в імені таблиці розширення не обов'язково
```

```
Table1.Active:=true; // підключаємо БД
```

Подальше використання компонента Table зводиться до того, що він, будучи таблицею БД, дозволяє виробляти з неї ряд маніпуляцій, включаючи редагування даних, переміщення по записах і т.д. Для цього використовуються методи, яких у компонента Table, з урахуванням всіх наслідувань, є близько сотні. Втім, багато хто з них не можна назвати повсякденно необхідними. З найбільш затребуваних можна відзначити методи First і Last, Next і Prior, використовувані для навігації по записах таблиці, а так само Append, Delete і Insert, які використовуються для додавання і видалення записів. Слід зазначити, що всі ці методи є успадкованими від класу TDataSet, що говорить про те, що вони є і у безлічі інших призначених для роботи з БД компонент. Весь код програмного засобу приведено в додатку А.

3.3 Тестування та верифікація програмного засобу

Тестування програмного забезпечення - перевірка відповідності між реальним і очікуваним поведінкою програми, що здійснюється на кінцевому

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

наборі тестів, обраному певним чином. У більш широкому сенсі, тестування - це одна з технік контролю якості, що включає в себе активності з планування робіт (Test Management), проектування тестів (Test Design), виконання тестування (Test Execution) і аналізу отриманих результатів (Test Analysis).

Після запуску програмного продукту куратору буде відображено головне вікно додатку, яке приведене на рисунку 3.16.

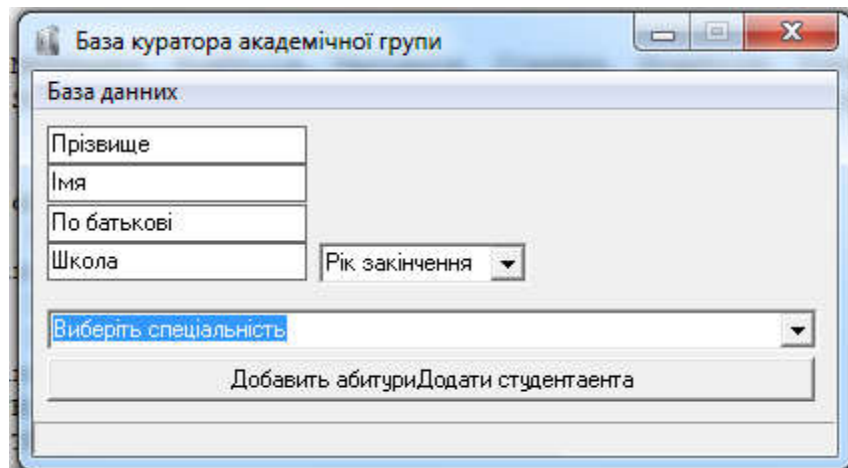


Рисунок 3.16 – Головне вікно програмного продукту на етапі проектування

Якість програмного забезпечення (Software Quality) - це сукупність характеристик програмного забезпечення, що відносяться до його здатності задовольняти встановлені і передбачувані потреби. [ISO 8402: 1994 Quality management and quality assurance]

Верифікація (verification) - це процес оцінки системи або її компонентів з метою визначення чи задовольняють результати поточного етапу розробки умов, сформованим на початку цього етапу [IEEE]. Тобто чи виконуються наші цілі, терміни, завдання по розробці проекту, визначені на початку поточної фази.

Валідація (validation) - це визначення відповідності розробляється ПО очікуванням і потребам користувача, вимогам до системи [BS7925-1].

Процес оцінки відповідності продукту явним вимогам (специфікаціям) і є верифікація (verification), в той же час оцінка відповідності продукту очікуванням і вимогам користувачів - є валідація (validation).

При тестуванні програмного засобу були використані наступні апаратні засоби: 15.6" (1920x1080) Full HD, матовий / Intel Core i5-7200U (2.5 - 3.1 ГГц) / RAM 4 ГБ / HDD 500 ГБ / AMD Radeon 520, 2 ГБ / DVD+/-RW / LAN / Wi-Fi / Bluetooth.

Цілі тестування:

– Підвищити ймовірність того, що додаток, призначене для тестування, буде працювати правильно при будь-яких обставинах.

– Підвищити ймовірність того, що додаток, призначене для тестування, буде відповідати всім описаним вимогам.

– Надання актуальної інформації про стан продукту на даний момент.

Результати бакалаврського проектування підтверженні довідкою про впровадження, яка наведена в додатку Б. Для роботи з системою створено форму, що дозволяє виконувати пошук та фільтрацію

Рисунок 3.17 – Форма для внесення екзаменаційних відомостей на етапі проектування

Головне меню програмного засобу приведене на рисунку 3.18. Та забезпечує можливості роботи головним меню та маніпуляцією даних.

Рисунок 3.18 – Форма для внесення екзаменаційних відомостей на етапі проектування

Головне меню програмного засобу приведене на рисунку 3.18. Проведене тестування програмного засобу та виявлено певні помилки при з'єднанні з базою даних. Помилки були виправлені, а виключні ситуації опрацьован.

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

Метою техніко – економічного розділу дипломної роботи є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності програмного засобу оптимізації роботи куратора групи та прийняття рішення про його подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки. Для проведення даного дослідження необхідно провести ряд розрахунків.

4.1 Розрахунок витрат на розробку програмного додатку

Витрати на розробку і впровадження програмного засобу оптимізації роботи куратора групи (K) включають:

$$K = K_1 + K_2,$$

де K_1 - витрати на розробку апаратного та програмного забезпечення грн.;

K_2 - витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку апаратних та програмних засобів включають:

- витрати на оплату праці розробників ($B_{оп}$);
- витрати на відрахування у спеціальні державні фонди ($B_{ф}$);
- витрати на матеріали та комплектуючі ($П_е$);
- накладні витрати (H);
- інші витрати ($I_е$);
- витрати на використання комп'ютерної техніки ($B_{КТ}$).

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

Розрахунок витрат на оплату праці.

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоемності відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломант; консультант техніко-економічного розділу (таблиця 4.1).

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

№п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, викладач	5286
2	Консультант техніко-економічного розділу, доцент	6026
3	Студент	1100

Витрати на оплату праці розробників проекту визначаються за наступною формулою (4.1):

$$B_{OP} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} , \quad (4.1)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.,

Середньо годинна ставка працівника може бути розрахована за такою формулою (4.2):

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ - місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Коефіцієнт h , який визначає розмір додаткової заробітної плати, для керівника та консультанта техніко-економічного розділу дорівнює 0,47.

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, старший викладач	16	46,25	740
2	Консультант техніко-економічного розділу, доцент	2	88,6	177,2
3	Студент	144	6,55	943,2
Разом				1860,4

Відрахування на соціальні заходи. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати:

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

$$B_{\phi} = \frac{20,5}{100} \cdot 1860,4 = 381,38 \text{ грн.}$$

Розрахунок витрат на матеріали та комплектуючі.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою (4.3):

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

де K_i - витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i - ціна за одиницю i -го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

№ п/п	Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн	
	Допоміжна література	шт	1	600	600	60	660	
	Папір (формат А4)	уп	2	80	160	16	176	
	Ручка кулькова	шт	2	10	20	2	22	
	Олівець простий	шт	2	10	20	2	22	
	Диски CD-R	шт	2	15	30	3	33	
	Зошит, 96 арк	шт	1	50	50	5	55	
	Тонер для принтера	уп	1	90	90	9	99	
	Канцелярські маркери (червоний, зелений)	шт	2	20	40	4	44	
	Р а з о м							1111,00

Витрати на використання комп'ютерної техніки.

Витрати на використання комп'ютерної техніки (V_{KT}) включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 6 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп'ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера грн.
1	Проведення досліджень та оформлення їх результатів	60	360
2	Оформлення техніко-економічного розділу	8	48
3	Оформлення ДП	12	72
Разом		80	480

Накладні витрати.

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 1860,4 = 2790,6 \text{ (грн).}$$

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Інші витрати.

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I_B = 1860,4 \cdot 0,1 = 186,04 \text{ (грн).}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_M + H + I_B + B_{KT},$$

$$K_1 = 1860,4 + 381,38 + 1111,00 + 2790,6 + 186,04 + 480,00 = 6809,42 \text{ (грн).}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою (4.4):

$$K_2 = S_{m.g.} \cdot t_{vid} \quad (4.4)$$

де $S_{m.g.}$ - вартість однієї машино-години роботи ПК, грн./год;

t_{vid} - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 6 грн., тому $K_2 = 6 \cdot 60 = 360$ грн.

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблювальної системи моніторингу слід порівняти її з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi},$$

де E_{Π} - одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ - вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

$E_{2\Pi}$ - вартість роботи комп'ютера для виконання проектного рішення (аналог), грн.

Річні експлуатаційні витрати $B_{E\Pi}$ визначаються за формулою:

$$B_{E\Pi} = E_{\Pi} * N_{\Pi},$$

де N_{Π} - періодичність експлуатації ПЗ (аналог), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1\Pi} = \sum_{i=1}^n n_i t_i c_i,$$

де i - категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i - кількість працівників i -ої категорії, осіб.;

t_i - трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i - середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0(1+b)}{m},$$

де c_i^0 - основна місячна заробітна плата працівника i -ої категорії, грн.;

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

b - коефіцієнт, який враховує додаткову заробітну плату (прийmemo 0,57);

m - кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c = 3723$ грн.

Тоді:

$$c_1 = \frac{3723(1 + 0,57)}{22 * 8} = 33,21 \text{ грн/год}$$

Трудомісткість підготовки даних для проектного рішення складає 1 год., для аналога 1,5 год.

Таблиця 4.5- Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
Проектне рішення			
1	1	33,21	33,21
Аналог			
1	1,5	33,21	66,42

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ}$$

де t - витрати машинного часу для реалізації рішення (аналогу), год.;

$S_{МГ}$ - вартість однієї години роботи комп'ютера, грн./год.

$$E_{2П} = 1 * 6 = 6 \text{ грн.}; E_{2А} = 1,5 * 6 = 9 \text{ грн.}$$

$$E_{П} = 33,21 + 6 = 39,21 \text{ грн.}; E_{А} = 66,42 + 9 = 75,42 \text{ грн.}$$

$$B_{ЕП} = 39,21 * 252 = 9880,92 \text{ грн.}; B_{ЕА} = 75,42 * 252 = 19005,84 \text{ грн.}$$

Обчислення накладних витрат.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників.

$$H_B = 0,7 * V_{OP}, \quad (4.7)$$

де H_B – накладні витрати.

$$H_B = 0,7 * 5845,11 = 4091,58 \text{ грн.}$$

Складання кошторису витрат та визначення собівартості.

Результати проведених розрахунків зведемо у таблицю 4.6.

Таблиця 4.6 - Кошторис витрат

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	1860,4
2	Відрахування у спеціальні державні фонди	381,38
3	Витрати на матеріали та комплектуючі	1111,00
4	Накладні витрати на розробку	2790,6
5	Інші витрати	186,04
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	360
7	Накладні витрати експлуатацію	4091,58
8	Річні експлуатаційні витрати	18005,84
Разом		27786,09

Розрахунок ціни проекту.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

Договірна ціна (C_D) для проектних рішень розраховується за формулою (4.8):

$$C_D = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.8)$$

де $B_{КС}$ – кошторисна вартість, грн.;

p - середній рівень рентабельності, % (приймаємо 26% за погодженням з керівником).

$$C_D = 29786,09 \cdot (1 + 0,26) = 37530,47 \text{ грн.}$$

4.3 Визначення економічної ефективності і терміну окупності капітальних вкладень

Економічна ефективність (E_ϕ) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_\phi = \frac{П}{B_{КС}}, \quad (4.9)$$

де $П$ – прибуток, грн.;

$B_{КС}$ – кошторисна вартість, грн..

$$E_\phi = 7812,27 \text{ грн.} / 29786,09 \text{ грн.} = 0,35.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (Tr):

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

$$T_P = \frac{1}{E_P} . \quad (4.10)$$

Тобто: $T_P = 1/0,25 = 2,8$ р.

Прийнятним вважається термін окупності близький до 7 років.

Розраховані економічні показники проекту занесемо до таблиці 4.7.

Таблиця 4.7 - Економічні показники розробки

№ п/п	Показник	Значення
1.	Собівартість, грн.	39786,09
2.	Плановий прибуток, грн.	6744,38в
3.	Ціна, грн.	26530,47
4.	Економічна ефективність	0,35
5.	Термін окупності, рік	2,8

Враховуючи основні економічні показники з таблиці 4.7, можна зробити висновок, що при економічній ефективності 0,25 та терміні окупності – 4 роки проводити роботи по впровадженню даного програмного модуля є доцільним та економічно вигідним.

ВИСНОВКИ

1. Було проаналізовано та досліджено предметну область, що дало можливість виконати постановку завдання та провести оцінку існуючих програмних засобів.

2. Проведене дослідження середовища розробки програмного засобу та проектування бази дани, що дало можливість максимально ефективно використати інструментарій розробки та маніпуляції даними.

3. На основі аналізу технічного завдання та недоліків існуючих систем сформовано вимоги до програмного забезпечення та функцій, які ним передбачені.

4. В результаті оцінки поставлених вимог до програмного засобу виконано проектування бази даних та основних таблиць, що забезпечать ефективне збереження даних та дозволять максимально ефективно опрацьовувати внесені дані.

5. Розроблено програмний засіб доступу до бази даних, який характеризується зручним користувачьким інтерфейсом та ергономічністю роботи користувача, що дозволить оптимізувати роботу куратора групи.

6. Проведена верифікація та тестування програмного засобу, яке довело коректність його роботи, та можливість його впровадження.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Васильська М. В. Узагальнена імітаційна модель розвитку системи мобільного зв'язку [Текст] / М. В. Васильська, В. М. Кичак, В. А. Северілов // Вісник Вінницького політехнічного інституту. – 2011. – №3. – С. 166-172.

2. Дибкова Л. М. Інформатика і комп'ютерна техніка [Текст] : навч. посіб. / Л. М. Дибкова. – 4-те вид., стер. – К. : Академвидав, 2012. – 464 с. – (Альма-матер).

3. Зыкин С. В. Формирование гиперкубического представления реляционной базы данных [Текст] / С. В. Зыкин // Программирование. – 2006. – ноябрь-декабрь (№6). – С.71-79.

4. Козловський А. В. Комп'ютерна техніка та інформаційні технології [Текст] : навч. посіб. / А. В. Козловський, Ю. М. Паночишин, Б. В. Погріщук. – 2-ге вид., стер. – К. : Знання, 2012. – 464 с. – Режим доступу : [http://library.tneu.edu.ua/images/stories/zmist/2014/літк/Ком'ютерна техніка та інформаційні технології.pdf](http://library.tneu.edu.ua/images/stories/zmist/2014/літк/Ком'ютерна_техніка_та_інформаційні_технології.pdf).

5. Антюфеев С. В. Нечеткий решающий алгоритм [Текст] / С. В. Антюфеев // Программирование. – 2006. – ноябрь-декабрь (№6). – С.27-35.

6. Беспроводная свобода : [идеальная домашняя сеть] [Текст] // CHIP+DVD. Россия. – 2009. – №9. – С. 128-131.

7. Бокс Дон Разработка и применение подключенных систем на основе Indigo: [Программирование] [Текст] / Дон Бокс // MSDN Magazine, /Русская редакция. – 2004. – №1. – С.39-46.

8. Мамченко С. Д. Основи інформатики та обчислювальної техніки: практикум [Текст] : навч. посіб. / С. Д. Мамченко, В. А. Одинець. – К. : Знання, 2007. – 292 с. – (Вища освіта XXI століття). – Режим доступу : [http://library.tneu.edu.ua/images/stories/zmist/2012/літо/основи інформатики мамченко 2007.pdf](http://library.tneu.edu.ua/images/stories/zmist/2012/літо/основи_інформатики_мамченко_2007.pdf).

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

9. Мельниченко А. Обучение вместо программирования [Текст] / А. Мельниченко // Электронные компоненты и системы. – 2004. – №12. – С.36-40.

10. Мінухін С. В. Комп'ютерні мережі. Загальні принципи функціонування комп'ютерних мереж [Текст]: навч. посіб. / С. В. Мінухін, С. В. Кавун, С. В. Знахур. – Х.: ХНЕУ, 2008. – 208 с. – Режим доступу: http://library.tneu.edu.ua/images/stories/zmist/2012/літк/компютерні_мережі_мінухін_2008.pdf.

11. Наливайко Н. Я. Інформатика [Текст]: навч. посіб. / Н. Я. Наливайко. – К.: ЦУЛ, 2011. – 577 с. – Режим доступу: http://library.tneu.edu.ua/images/stories/zmist/2013/літі/Інформатика_Наливайко_Н.Я..pdf.

12. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы [Текст]: учеб. пособ. / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб.: Питер, 2012. – 944 с. – (Учебник для вузов. Стандарт третьего поколения). – Режим доступу: http://library.tneu.edu.ua/images/stories/zmist/2012/літк/компьютерные_сети_олифер_2012.pdf.

13. Сейдаметова З. Нова версія стандарту Computer Science Curricula: еволюція базисного корпусу знань за чверть століття [Текст] / З. Сейдаметова, В. Темненко // Вища школа. – 2012. – № 12. – С. 54-64.

14. Теслюк В. М. Дослідження і проектування комп'ютерних систем та мереж [Електронний ресурс]: консп. лекцій / В. М. Теслюк. – Тернопіль: ТНЕУ, 2012. – 62 с. – Режим доступу: <http://library.tneu.edu.ua/images/stories/predmety/літд/дослідження%20і%20проектування%20комп'ютерних%20систем%20та%20мереж/Досл%20і%20проектув%20комп%20сист%20та%20мереж.pdf>.

15. Хассел Д. Администрирование Windows server 2003 [Текст] / Д. Хассел. – СПб.: Питер, 2006. – 576 с. – Режим доступу: http://library.tneu.edu.ua/images/stories/zmist/2012/літа/администрирование_виндовс_хассел_2006.pdf.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

16. Шалыто А. А. Новая инициатива в программировании [Текст] / А. А. Шалыто // Информационно-управляющие системы. – 2003. – №4. – С.52-57.
17. Шляхтина С. Обзор решений для работы с системным реестром [Текст] / С. Шляхтина // Компьютер Пресс + CD. – 2007. – №6. – С. 158-163.
18. Стенли Б. С++ для начинающих / Б. Стенли, Липпман: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
19. Фаронов В. В. Система программирования Delphi [Текст] / В. В. Фаронов. – С-Пб. : БХВ-Петербург, 2006. – 888 с.
20. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. - Тернопіль: ТНЕУ, 2016.–65 с.
21. Паздрій І.Р. Методичні вказівки до написання техніко-економічного розділу для дипломних проектів на здобуття освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» / І.Р.Паздрій. - Тернопіль: ТНЕУ, 2015.– 36 с.
22. Шалыто А. А. Новая инициатива в программировании [Текст] / А. А. Шалыто // Информационно-управляющие системы. – 2003. – №4. – С.52-57.
23. Фомин А. В. Расширения реалиционной модели для обеспечения безопасности в базах данных [Текст] / А. В. Фомин // Информационно-управляющие системы. – 2003. – №4. – С.24-33.
24. Рогожкин, И. Средства вывода информации [Текст] / И. Рогожкин. – С.96-98.
25. Средства Java от компании Sun Microsystems [Текст] // PC MAGAZINE. RUSSIAN EDITION. – 2003. – №11.
26. Гурский Д. Кто виноват и что делать?: Извечные вопросы в контексте борьбы со спамом: [Компьютерная сеть] [Текст] / Д. Гурский, Ю. Стрельченко // Hard'n'soft ua. – 2003. – №10. – С.74-82.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

27. Шалыто А. А. Автоматное и синхронное программирование [Текст] / А. А. Шалыто // Искусственный интеллект. – 2003. – №4. – С.82-89.

28. Штейнберг Р. Б. Вычисление задержки в стартах конвейеров для суперкомпьютеров со структурно-процедурной организацией вычислений [Текст] / Р. Б. Штейнберг // Искусственный интеллект. – 2003. – №4. – С.105-144.

29. Белозерский Л. А. Совместимость обучаемого классификатора и систем автоматического распознавания [Текст] / Л. А. Белозерский // Искусственный интеллект. – 2003. – №4. – С.185-195.

30. Дариев, И. Что принес Windows Vista Service Pack 1 [Текст] / И. Дариев // Компьютерное обозрение. – 2008. – 26 февраля (№8). – С. 60-61.

					ДП.КСМ.07236/16.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65