

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка
до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
ОРГАНІЗАЦІЇ ДОКУМЕНТООБІГУ НАВЧАЛЬНОЇ ЧАСТИНИ
ІНСТИТУТУ МОВОЮ C#**»

Виконав: студент 4 курсу, групи ПД–41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Зверінський А.Ю.
(прізвище та ініціали)

Керівник Бондарчук А.П.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ _____ ” _____ 2022 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

Зверінський Андрій Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для організації документообігу навчальної частини інституту мовою С#»

Керівник роботи: Бондарчук Андрій Петрович, професор, доктор технічних наук

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «18» лютого 2022 року №.

2. Строк подання студентом роботи «3» червня 2022 року

3. Вхідні дані до роботи:

3.1 Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо організації документообігу.

3.2 Офіційна документація Microsoft Visual Studio.

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити):

4.1 Аналіз актуальності та проблематики інформаційної системи інституту.

4.2 Аналіз та вибір інструментів для реалізації ПЗ.

4.3 Проектування програми.

4.4 Висновки.

5. Перелік демонстраційного матеріалу (назва основних слайдів):

- 5.1 Титульний слайд.
 - 5.2 Мета, об'єкт, предмет, наукова новизна дослідження.
 - 5.3 Актуальність.
 - 5.4 Аналіз аналогів.
 - 5.5 Технічні завдання.
 - 5.6 Програмні засоби та інструменти реалізації.
 - 5.7 Діаграма діяльності.
 - 5.8 Діаграма класів.
 - 5.9 Висновки.
 - 5.10 Кінцевий слайд.
6. Дата видачі завдання «11» квітня 2022р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04.2022	Виконано
2	Дослідження аналогів та актуальності додатку	13.04.2022	Виконано
3	Аналіз та вибір інструментів для розробки додатку	14.04.2022	Виконано
4	Проектування та реалізація	02.05.2022	Виконано
5	Вступ, висновки, реферат	17.05.2022	Виконано
6	Розробка обов'язкових демонстраційних матеріалів	18.05.2022	Виконано
7	Попередній захист роботи	29.05.2022	Виконано
8	Здача роботи	3.06.2022	Виконано

Студент _____ Зверінський А.Ю.
 (підпис) (прізвище та ініціали)

Керівник роботи _____ Бондарчук А.П.
 (підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 79 с., 10 табл., 27 рис., 18 джерел.

Ключові слова: документообіг, інформаційна система, звіт, база даних.

Об'єкт дослідження – створення та обробка звіту успішності студентів навчальних закладів.

Предмет дослідження – автоматизація організації документообігу інституту.

Мета роботи – розробка програмного забезпечення для спрощення організації документообігу інституту.

Наукова новизна – створення нових та перетворення відомих ігрових механік, притаманних для ігор жанра три в ряд; створення інтуїтивного та привабливого дизайну гри, що зачепить гравця.

У дипломному проєкті був проведений аналіз інформаційної системи інституту та додатків-аналогів. Проаналізовано та виявлено переваги і недоліки програмних інструментів розробки. Проаналізовано особливості розробки інформаційних систем та баз даних.

Програмне забезпечення було створено за допомогою середовища розробки Microsoft Visual Studio Community 2019 на мові програмування C#. В якості платформи було обрано операційну систему Windows. Для створення інформаційної бази даних використовувався Microsoft Access. Для створення елементів користувацького інтерфейсу було використано Microsoft Forms.

Даний додаток може слугувати ефективним інструментом, що дозволить автоматизувати процеси формування та редагування звітів, що пов'язані з відслідковуванням успішності окремих студентів. Програмне забезпечення дозволяє автоматизувати процеси документообігу інституту, тим самим полегшує та оптимізує роботу працівників.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПОТОЧНОГО СТАНУ ІНФОРМАЦІЙНОЇ СИСТЕМИ В ОБЛАСТІ ДОКУМЕНТООБІГУ НАВЧАЛЬНОЇ ЧАСТИНИ ІНСТИТУТУ	11
1.1 Особливості діяльності у сфері документообігу навчальної частини інституту	11
1.2 Аналіз сучасних програм із визначенням їх переваг та недоліків.....	13
1.2.1 Програмний продукт "ІС.Університет"	14
1.2.2 Програма "Деканат"	16
1.2.3 Програмний продукт АС «Навчальний відділ»	17
1.2.4 Висновок по проведеному аналізу аналогічних систем	19
1.3 Необхідність автоматизації бізнес-процесів у даній сфері.....	20
1.4 Опис технології обробки інформації.....	22
1.5 SWOT-аналіз існуючої інформаційної системи	22
1.6 Заходи та рекомендації для вдосконалення інформаційної системи	24
1.7 Характеристика організації вирішення проекту.....	24
1.8 Обґрунтування проектних рішень.....	25
1.9 Постановка задачі.....	25
1.10 Висновок.....	27
2 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ РОЗРОБКИ.....	28
2.1 Аналіз вимог. Use-case діаграми. Основні прецеденти.....	28
2.2 Архітектура проекту	38
2.3 Особливості розробки бази даних. ERD діаграма з описанням сутностей	39
2.4 Особливості розробки рівня BLL	42
2.5 Особливість реалізації бізнес логіки – діаграма домена.....	44
2.6 Особливості розробки рівня UI	45
2.7 Особливості розробки DAL	47
2.8 Висновок.....	49
3 РЕАЛІЗАЦІЯ ПОСТАВЛЕНИХ ЗАДАЧ ТА ТЕСТУВАННЯ.....	51
3.1 Вибір технологій	51
3.2 Результати функціонального тестування розробленого додатку	52
3.3 Інструкція користувачеві програми	63
3.4 Висновок.....	64

4 ЕКОНОМІЧНА ЧАСТИНА	65
4.1 Розрахунок витрат на розробку програмного забезпечення	65
4.2 Визначення експлуатаційних витрат	69
4.3 Розрахунок ціни споживання проектного рішення	72
4.4 Визначення показників економічної ефективності	73
4.5 Висновки	74
ВИСНОВКИ	75
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	77

ВСТУП

Проблема автоматизації виробничих процесів та процесів управління як засобу підвищення продуктивності праці завжди була і залишається актуальною у будь-якій сфері людської діяльності, сфера освіти також не є винятком. Необхідність автоматизації процесів управління у сфері освіти пояснюється завданнями полегшення праці персоналу, а отже, підвищення ефективності робіт зі складання різних звітів та відомостей та їх обробки, а також обробки та зберігання різних даних.

На етапі автоматизації найперспективнішим є автоматизація планово – управлінських функцій з урахуванням персональних ЕОМ, встановлених безпосередньо робочих місцях спеціалістів. Ці системи набули широкого поширення в організаційному управлінні під назвою автоматизованих робочих місць (АРМ). Це дозволить використовувати систему людям, які не мають спеціальних знань у галузі програмування, і одночасно дозволить доповнювати систему в міру потреби.

У разі модернізації освіти процес інформатизації стає дедалі актуальним. Використання інформаційних технологій адміністрацією закладів освіти дозволяє збільшити ефективність управління. Автоматизовані системи дозволяють працювати з різними шаблонами документів, витрачаючи найменшу кількість часу на пошук та систематизацію необхідної інформації.

В даній дипломній роботі необхідно розробити програмне забезпечення для організації документообігу навчальної частини інституту.

Для досягнення зазначеної мети було поставлено такі завдання:

- Провести аналіз стану інформаційної системи в області документообігу навчальної частини інституту;
- Розглянути сучасні програми, що використовуються для документообігу навчальних частин інституту із визначенням їх переваг та

недоліків;

- На основі отриманих в результаті аналізу даних зробити постановку завдання до майбутньої системи;
- Побудувати use-case діаграму із списком вимог до майбутньої системи;
- Описати архітектуру проекту;
- Здійснити розробку програмного забезпечення для організації документообігу навчальної частини інституту мовою C#;
- Провести функціональне тестування розробленої системи;
- Розробити інструкцію для користувача програми;
- Розрахувати витрати на розробку програмного забезпечення.

1 АНАЛІЗ ПОТОЧНОГО СТАНУ ІНФОРМАЦІЙНОЇ СИСТЕМИ В ОБЛАСТІ ДОКУМЕНТООБІГУ НАВЧАЛЬНОЇ ЧАСТИНИ ІНСТИТУТУ

1.1 Особливості діяльності у сфері документообігу навчальної частини інституту

Управління інформаційними ресурсами має для діяльності будь-якої навчальної установи особливе значення. У світі установи стикаються з необхідністю обробки колосального обсягу інформації. У більшості установ зі складною структурою важливе значення має рівень організації взаємодії підрозділів та порядок обміну інформацією. Більшість інформації передається у вигляді документів на паперовому носії (обмін службовою документацією та звітністю).

Документообіг установи - це сукупність взаємозалежних процедур, що забезпечують рух документів в установі з моменту їх створення або надходження до завершення виконання або відправлення.

З метою раціональної організації документообігу всі документи розподіляються на документопотоки: реєстровані та незареєстровані документи, що входять, вихідні та внутрішні документи, документи, що надходять або надсилаються до вищестоящих підрозділів, документи, що надходять або надходять з підвідомчих установ та ін. Під документопотоком розуміється сукупність документів, що виконують певне цільове призначення у процесі документообігу. Надходячи в установу документи утворюють потік вхідної документації, які, пройшовши необхідні інстанції обробки, розбиваються на підтоки як конкретні доручення.

У підрозділах з документації, що формується співробітниками, також утворюються документопотоки, які в результаті зливаються в єдиний потік

вихідної документації. А документи, створені в організації та не призначені до виходу за її межі, становить потік внутрішньої документації.

Важливою характеристикою документообігу є його обсяг. Під обсягом документообігу розуміється кількість документів, що надійшли в організацію та створених нею протягом певного періоду часу (як правило, року). Обсяг документообігу - важливий показник, що використовується як критерій при виборі організаційної форми діловодства, організації інформаційно-пошукової системи за документами установи, встановлення структури служби діловодства, її штатного складу та інші.

Поряд з організацією документообігу входить зберігання документів та їх використання у діяльності установи. Система зберігання документів - це сукупність коштів, способів та прийомів обліку та систематизації документів з метою їх пошуку та використання у поточній діяльності установи.

Управлінські документи становлять ядро установчої документації. Саме вони забезпечують керованість об'єктів, як у межах держави, і у окремої організації. Ці документи представлені комплексом систем, основними у тому числі є такі системи документації:

1. Організаційно-правова документація. Організаційна діяльність установи виявляється у розробці та затвердженні комплексу організаційно-правових документів, що містять правила, норми, положення, що встановлюють статус організації, структуру, штатну чисельність та посадовий склад, функціональний зміст діяльності організації в цілому, її підрозділів та працівників, їх права, обов'язки, відповідальність та інші аспекти. До організаційно-правових документів відносяться: статут організації, положення про організацію; положення про структурні підрозділи; Посадові інструкції працівникам, правила, пам'ятки та ін. Організаційно-правові документи, з погляду терміну дії, відносяться до безстрокових і діють до їх скасування або до затвердження нових.

2. Розпорядча документація. Основне призначення розпорядчих документів - регулювання діяльності, що дозволяє органу управління забезпечувати реалізацію поставлених перед ним завдань, отримувати максимальний ефект від своєї діяльності та діяльності установ, що входять до його системи. Розпорядчі документи містять рішення, що йдуть зверху вниз у системі управління: від керуючого органу до керованого, тобто від вищого до підвідомчого, від керівника організації до структурних підрозділів та працівників.

3. Інформаційно-довідкова та довідково-аналітична документація. Інформація про фактичний стан справ у системі управління міститься в різних джерелах, але найважливіше місце серед них займають довідково-інформаційні та довідково-аналітичні документи: акти, довідки, доповідні записки, листування та ін. Документи цієї групи повідомляють відомості, що спонукають приймати певні рішення, Інакше висловлюючись, - ініціюють управлінські рішення, дозволяють вибрати той чи інший спосіб дії.

4. Звітна документація. Звітну документацію складають документи, що містять відомості про результати діяльності установи за певний період часу – рік, півріччя, квартал, місяць, декада. Звітні документи можуть складатися за різними напрямками діяльності установи: про результати функціональної діяльності підрозділів та загалом про установу, про результати науково-дослідної або науково-впроваджувальної діяльності, про результати рекламної діяльності (рекламних кампаній), про виконання окремих, разових заходів.

1.2 Аналіз сучасних програм із визначенням їх переваг та недоліків

З появою перших засобів автоматизації було розроблено перші програмні продукти, кожен ВНЗ намагається автоматизувати свій навчальний процес. Ця автоматизація відбувається силами самого ВНЗ або за рахунок залучення сторонніх компаній, що займаються розробкою та впровадженням програмних продуктів.

Проте кількість продуктів, які абсолютно підходять для впровадження в будь-який ВНЗ дуже мало. Пов'язано це з тим, що вузів значно менше і в більшості з них використовують власні розробки «заточені» під їх специфіку та потреби. У багатьох вузах взагалі не користуються будь-яким спеціалізованим програмним забезпеченням, обмежуючись паперовим обліком та печаткою через текстові редактори з можливістю «підгонки» тексту під потрібні поля на бланку. Однак деякі вузи все ж таки користуються стороннім програмним забезпеченням, найбільш популярне представлено нижче.

1.2.1 Програмний продукт "1С.Університет"

"1С:Університет" - дозволяє автоматизувати облік, зберігання, обробку та аналіз інформації про основні процеси вищого навчального закладу: вступ до вузу, навчання, оплата за навчання, випуск та працевлаштування випускників, розрахунок та розподіл навантаження професорсько-викладацького складу, діяльність навчально- методичних відділів та деканатів та рівневої системи підготовки (бакалавр, спеціаліст, магістр) на рівні навчальних планів та документів державного зразка про закінчення вузу, формування звітності. Інтерфейс програми зображено на рисунку 1.1.

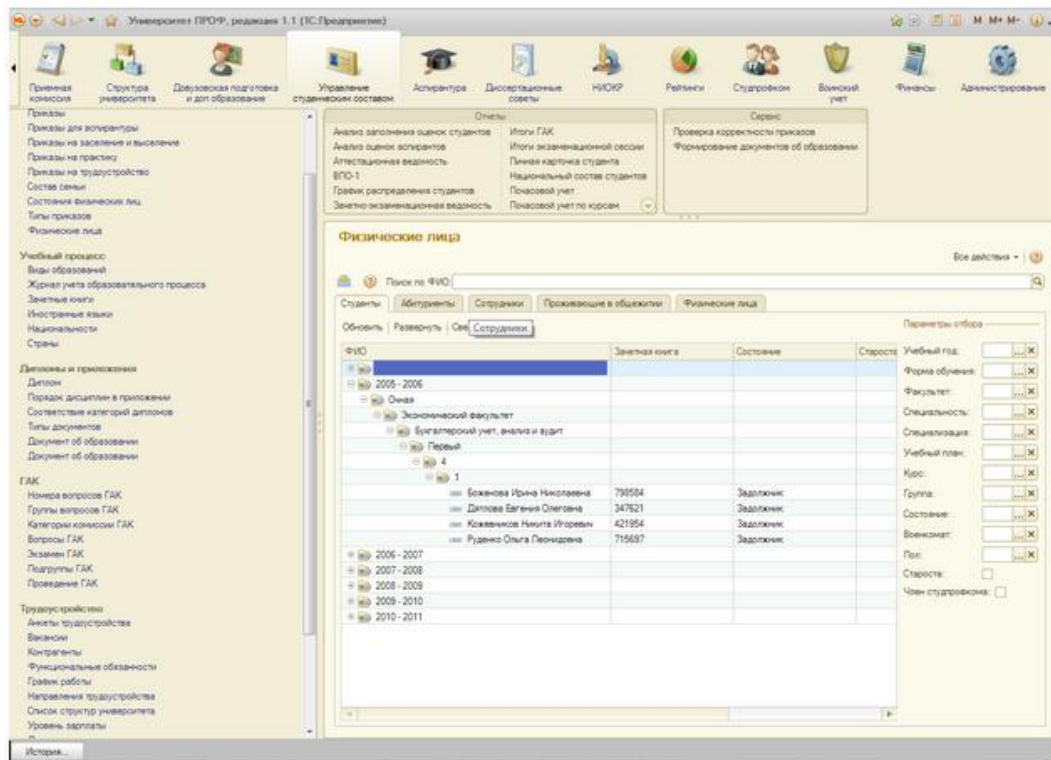


Рисунок 1.1 – Интерфейс программы "1С:Университет"

Преваги:

- об'єднує у собі всі модулі для повноцінної роботи ВНЗ;
- формування наказів та розпоряджень;
- можливість самостійного визначення нових типів наказів та зміни існуючих;
- гнучке настроювання вихідних друкованих форм розпорядчих документів;
- врахування коректності проведення документів;
- облік відвідуваності;
- Можливість налаштування та створення шаблонів друкованих форм документів.

Недоліки:

- перевантажений інтерфейс;

- висока ціна (40000 грн.).

1.2.2 Програма "Деканат"

Програма обліку студентів «Деканат» є засобом автоматизації управління контингентом студентів (рис. 1.2). Основними функціями програми є:

- Редагування списку груп, спеціальностей та відомостей про факультет;
- Створення та заповнення особових справ студентів;
- Переміщення студентів (довільне переміщення студента із групи до групи, на наступний курс, переміщення у категорії «Відраховані», «Закінчили навчання», «Академічна відпустка» та відновлення з них);
- Формування друкованих звітів, списків груп та статистики;
- Пошук студентів у базі даних, отримання зведень за кількістю студентів;
- Підготовка звітів форми ЗНК.

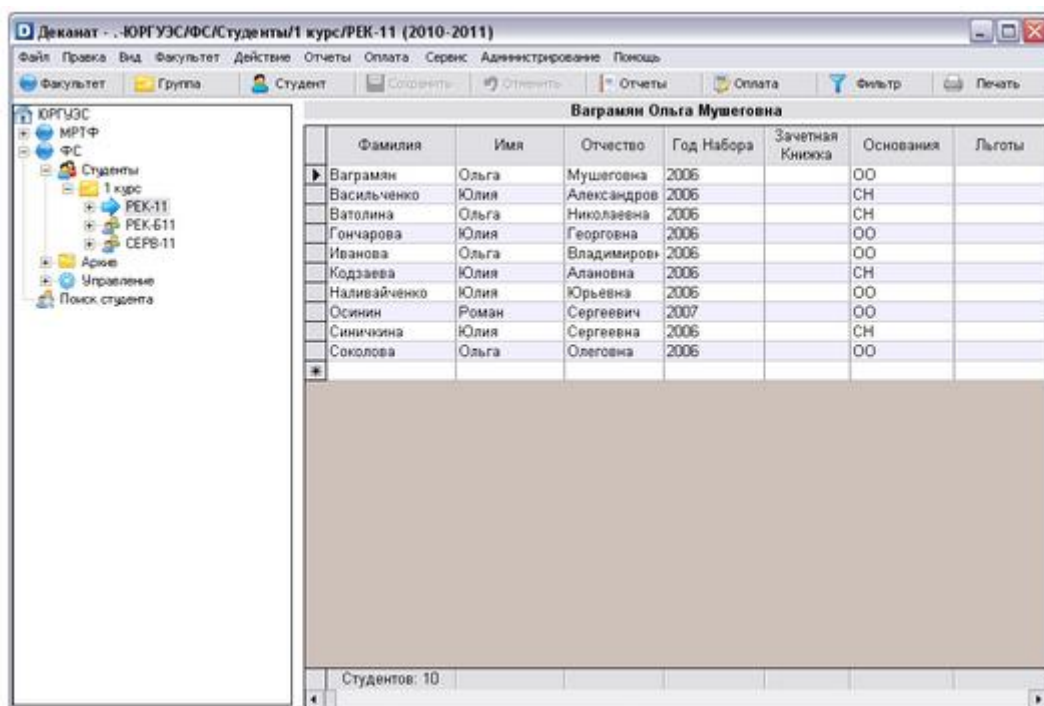


Рисунок 1.2 – Интерфейс програми "Деканат"

Переваги:

- Облік даних щодо оплати навчання студентом;
- Гнучкий розподіл прав користувачів;
- Введення оцінки або вибір значення зі списку натисканням цифри;
- Формування наказів та розпоряджень.

Недоліки:

- Висока ціна (15000 грн).

1.2.3 Програмний продукт АС «Навчальний відділ»

«Навчальний відділ» - програма призначена на вирішення різноманітних завдань, що з урахуванням складу учнів осіб (слухачів), успішності і розрахунком тих чи інших параметрів із них у навчальних закладах (рис. 1.3). Може

застосовуватись для автоматизації діяльності працівників приймальної комісії, деканатів, відділу кадрів, військово-облікового столу навчальних закладів.

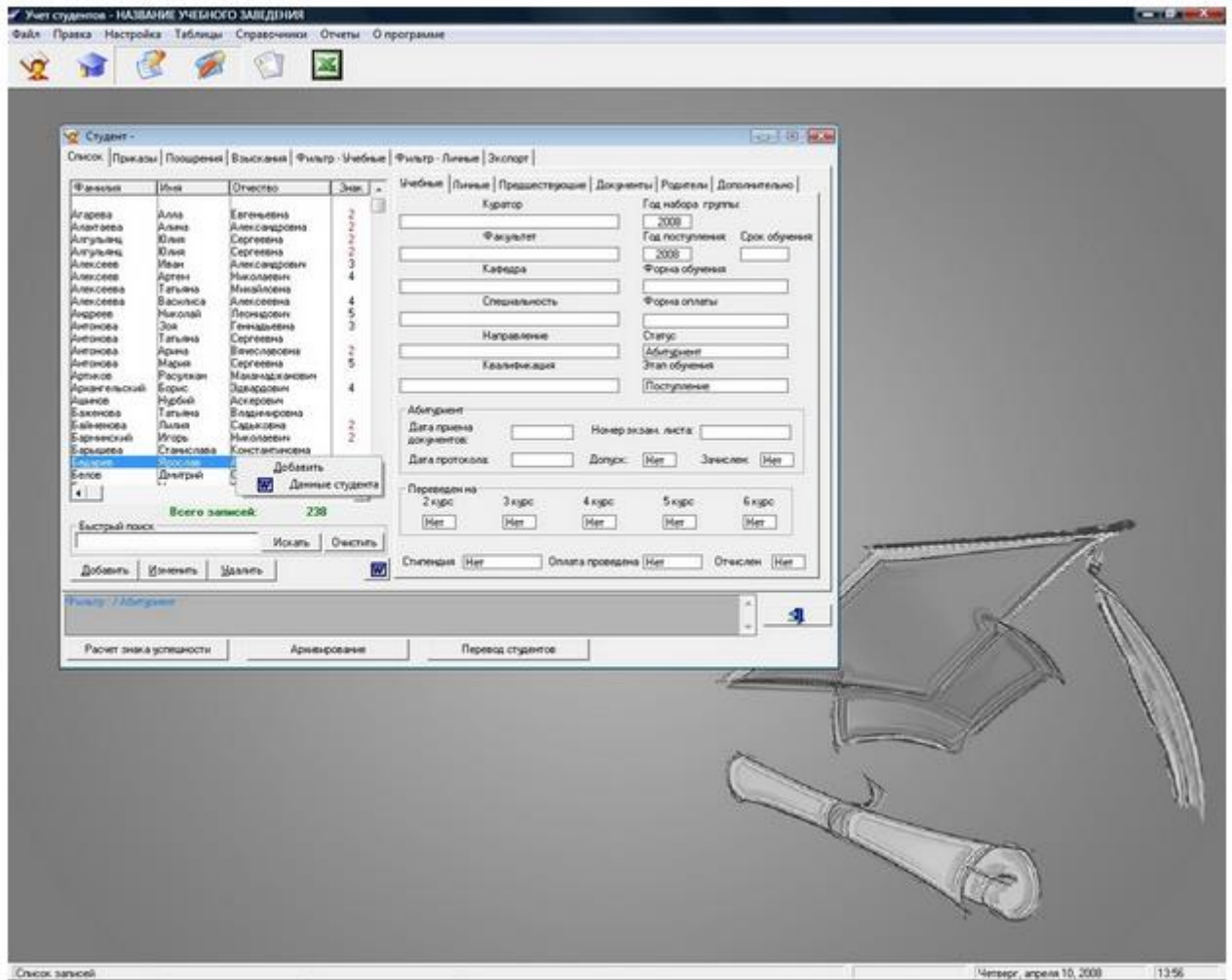


Рисунок 1.3 – Інтерфейс програми «Навчальний відділ»

Функції програми Навчальний відділ:

- Облік індивідуальних та навчальних ознак абітурієнтів, студентів та інших, які задаються користувачем, типів учнів навчальних закладів;
- Облік наказів про прийом, переклад та відрахування, заохочення та стягнення студентів, виконані курсові роботи та проходження практики;
- Облік виданих дипломів про повну та неповну вищу освіту та академічні довідки, а також зіпсованих бланків;

- Облік викладацького складу, предмети, які вони читають, та їх обсяг у годинах;
- Облік успішності кожного зі студентів окремо та навчальних груп загалом;
- Формування навчальних планів для навчальних груп;
- Розрахунок коефіцієнта успішності здачі поточної сесії для оперативного контролю учнів;
- Розрахована на багато користувачів робота з розмежуванням прав доступу;
- Гнучка система фільтрації учнів та викладацького складу;
- Генерація звітів для приймальної комісії, деканату та відділу кадрів у форматі Microsoft Excel;
- Друк дипломів, додатків до дипломів та академічних довідок;
- Можливість редагування базових шаблонів та додавання звітів користувача;
- Імпорт та експорт усієї або частини бази даних.

Переваги:

- гнучка система фільтрації учнів та викладацького складу;
- генерація звітів для приймальної комісії, деканату та відділу кадрів у форматі Microsoft Excel;
- можливість редагування базових шаблонів і додавання звітів користувача;

Недоліки:

- відсутність великої кількості необхідних звітів;
- відсутність імпорту-експорту елементів бази.

1.2.4 Висновок по проведеному аналізу аналогічних систем

Провівши аналіз існуючих розробок для організації документообігу навчальної частини інституту, було зроблено висновок, що вони мають свої переваги та недоліки. Головним недоліком став великий функціонал, який не потрібний, а також вартість для неї дуже велика, тому виникла необхідність у розробці нового програмного забезпечення. У таблиці 1.1 представлена зведена інформація аналізу існуючих програмних продуктів.

Таблиця 1.1 – Аналіз існуючих програмних продуктів.

Характеристики	ІС. Університет	Деканат	Навчальний відділ	Розроблене ПЗ
Зручний інтерфейс	-	+	-	+
Формування звітів успішності	+	-	+	+
Доступність для українських ВНЗ	-	+	-	+
Невисокі системні вимоги	-	+	+	+
Можливість безкоштовного користування	-	-	-	+

Для зручності роботи потрібне програмне забезпечення, що враховує її особливості, специфіку функціонування, що дозволяє виконувати певні функції обліку, актуальні на даний момент, а також проводити гнучке налаштування та розширення функціоналу у разі необхідності у процесі використання програмного продукту.

1.3 Необхідність автоматизації бізнес-процесів у даній сфері

На сьогоднішній день на ринку представлено безліч систем для автоматизації документообігу навчальної частини інституту.

Завдяки автоматизації відбувається:

- підвищення продуктивності праці;
- зведення до мінімуму негативного впливу людського фактору на найважливіші бізнес-процеси;
- безпечне зберігання інформації;
- швидке отримання звітів;
- швидка підготовка документів;
- швидкий пошук необхідної інформації.

Автоматизація необхідна в тих ділянках діяльності, де її поява збільшить швидкість виконання завдань і, відповідно, підвищить ефективність праці [4].

Автоматизація діяльності навчальної частини інституту є проектною роботою, розбитою на фази, метою якої зрештою є підвищення ефективності роботи співробітників, підвищення якості виконання роботи, збільшення точності даних, швидкості виконання операцій тощо [5].

Автоматизація діяльності завжди є унікальним рішенням та робота над цим проектом завжди індивідуальна. Але робота з проектами побудована на однакових засадах.

Для того, щоб спростити процес автоматизації, проект поділяють на кілька етапів:

- проектне дослідження;
- розробка і впровадження;
- супровід.

На першому етапі вивчається предметна область. Ставиться ціль, яка допоможе вирішити поставлені завдання. Також збирається інформація про те, як працює дана установа. Після перерахованого вище визначається вартість та терміни, необхідні для реалізації проекту.

На другому етапі робота поділяється на фази:

- написання технічного завдання;

- розробка;
- тестування;
- дослідна експлуатація;
- навчання користувачів;
- введення в експлуатацію.

На третьому етапі здійснюється супровід автоматизованої системи:

- виправлення помилок;
- доопрацювання для покращення роботи програми, консультація.

1.4 Опис технології обробки інформації

На момент розробки програмного продукту у більшості навчальних частини інститутів не було програмного забезпечення для організації документообігу, складання документації, виведення звітів та статистики. Запис студентів здійснювався ручним способом у журналі. А для обліку та складання звітів використовувалась програма Microsoft Excel. Але при великому потоці інформації, збільшився і час їх виконання.

Виявлено такі недоліки:

1. Великі часові витрати на опрацювання даних про студентів.
2. Небезпечна технологія зберігання документації.
3. Великі часові витрати на пошук необхідної інформації.
4. Проблеми з формуванням звітності.

Провівши аналіз поточної інформаційної системи, прийнято автоматизувати робочі процеси деканату.

1.5 SWOT-аналіз існуючої інформаційної системи

SWOT-аналіз — це аналіз, який показує сильні та слабкі сторони, можливості та загрози.

Стан документообігу навчальної частини інституту також залежить від успішного реагування на різні дії ззовні.

При аналізі зовнішньої ситуації необхідно враховувати суттєві фактори на даний час. Розгляд цих чинників з можливостями установи дозволяє вирішувати проблеми.

SWOT-аналіз — це необхідний елемент досліджень, обов'язковий попередній етап при складанні будь-якого рівня стратегічних і організаційних планів установи.

В результаті SWOT-аналізу отримано дані, які необхідні при розробці стратегічних цілей та завдань. У таблиці 1.2 представлено результати SWOT-аналізу для документообігу навчальної частини інституту.

Таблиця 1.2 – SWOT-аналіз документообігу навчальної частини інституту

Потенційні внутрішні сильні сторони (S)	Потенційні внутрішні слабкості (W)
1. Застосовується проста форма для формування заявок, звична для працівників (Журнал, MS Excel) 2. Не потрібно часто використовувати ПК.	1. Використання паперових носіїв інформації. 2. Великі витрати ручної праці. 3. Довготривале та складне формування звітності
Потенційні зовнішні сприятливі можливості (O)	Потенційні зовнішні погрози (T)
1. Можливість запровадження нового програмного забезпечення.	1. Можливість втрати інформації на паперових носіях. 2. Слабка система забезпечення безпеки інформації. 3. Відсутність актуальної інформації.

Сильні та слабкі сторони є факторами внутрішнього середовища, які впливають на стан установи. Можливості та загрози — зовнішні фактори, що впливають на установу ззовні та при цьому не контролюються нею.

Переваги SWOT-аналізу полягають у тому, що він достатньо просто дозволяє розглянути становище установи, послуги у галузі, і тому він є популярним інструментом в управлінні ризиками та прийнятті управлінських рішень при організації документообігу навчальної частини інституту.

1.6 Заходи та рекомендації для вдосконалення інформаційної системи

Основною метою для автоматизування варіанту вирішення завдання є створення програмного забезпечення для організації документообігу навчальної частини інституту.

Для реалізації проекту автоматизації мають забезпечуватися такі можливості:

1. Ведення електронних відомостей успішності студентів з оцінками .
2. Виконання розрахунків та видача різних звітів та статистик.

Переваги автоматизованого процесу виражаються у:

- скороченні часу на процедуру ведення електронних відомостей успішності студентів;
- скороченні часу на процес формування різних звітів;
- оптимізації роботи співробітників, з метою швидкого та надійного способу ведення, пошуку, обробки, безпечного зберігання інформації про результати діяльності.

1.7 Характеристика організації вирішення проекту

Розроблене програмне забезпечення передбачається використовувати у навчальної частини інституту для роботи спеціаліста з метою вдосконалення

процесу організації документообігу навчальної частини інституту, а також можливості формування звітності.

Перевагою пропонованої автоматизованої системи є простота та інтерфейс. У цій програмі реалізується весь необхідний для роботи функціонал. Також програма забезпечує зручність зберігання та обробки інформації. Також спеціаліст може отримувати різні види звітів та статистику.

1.8 Обґрунтування проектних рішень

Важливим фактором, який необхідно врахувати при розробці програмного забезпечення є потреба в ресурсах, які є в установі.

Мінімальні вимоги для запуску та функціонування розробленого програмного забезпечення:

1. Процесор PentiumI V/Хеон 2.4 ГГц.
2. Оперативна пам'ять: 1024 Мб і вище.
3. Вільний дисковий простір не менше 120 Мб.
4. Мережева картка.
5. Миша.
6. Клавіатура.
7. Монітор.

З опису видно, що до технічного забезпечення програми потреби не є високими.

1.9 Постановка задачі

Дипломна робота припускає:

- розробку схеми БД;
- розробку та реалізацію додатка, що включає такі основні модулі:
 - 1) модуль вводу/редагування та видалення інформації в БД;

- 2) модуль формування звітності;
- 3) модуль забезпечення захисту інформації.

Реалізація додатка виконується з використанням технологій .NET на мові програмування C#. Результат – Windows додаток, що представляє собою **Систему організації документообігу навчальної частини інституту**, яка містить:

Сутності:

- Студенти (інформація про студента): прізвище, ім'я, № телефону, домашня адреса, електронний адрес, ідентифікатор групи;
- Група (інформація про групу): назва та опис;
- Дисципліна (інформація про дисципліну) : назва та опис;
- Відомість (інформація про відомість): номер відомості, ідентифікатор групи, ідентифікатор дисципліни, дата заповнення;
- Рейтинг (інформація про успішність студентів): ідентифікатор студента, ідентифікатор відомості, ідентифікатор дисципліни, оцінка за предмет;
- Користувачі системи (інформація про користувачів та їхні облікові записи): прізвище, ім'я, нікнейм облікового запису, пароль, загальна інформація;
- Логи (інформація про активність користувачів системи) ідентифікатор користувача, подія в системі, дата/час події.

Введення:

Вводиться і редагується: студенти, групи, дисципліни.

Формування відомості про успішність студентів.

Звітність:

- Зведена відомість за групу (студент , оцінка);
- Рейтинг студентів спеціальності по курсам (Група , ПІБ, Середній бал за 100 шкалою);
- Стипендія (Група , ПІБ, оцінки по 5 бальній шкалі, середній бал);

- Виписка за результатами навчання за семестр (ПБ, навчальна дисципліна, оцінка, № відомості) для навчальної картки;
- Додаток до диплому про вищу освіту (Назва предмету, оцінка).

1.10 Висновок

У ході аналізу поточного стану інформаційної системи в області організації документообігу навчальної частини інституту були розглянуті можливі проблеми та помилки при організації документообігу, причини виникнення та способи їхнього відстеження. Було зроблено опис технологій обробки інформації. Зроблено SWOT-аналіз існуючої інформаційної системи. На основі отриманих даних було проведено постановку задачі на створення інформаційної системи, основне завдання якої полягає у організації документообігу навчальної частини інституту.

2 РОЗРОБКА

2.1 Аналіз вимог. Use-case діаграми. Основні прецеденти

Згідно з поставленою задачею можна висунути функціональний список вимог до проекту (табл. 2.1.).

Таблиця 2.1 – Функціональні вимоги до додатку «Організація документообігу навчальної частини інституту»

Вимоги	Опис
REQ-1	Система повинна дозволяти здійснювати реєстрацію користувача в програмі
REQ-2	Система повинна вести журнал подій, що відбулися в програмі
REQ-3	Система повинна дозволяти користувачеві додати та редагувати інформацію про облікові записи користувачів
REQ-4	Система повинна дозволяти користувачеві додати та редагувати інформацію про студентів
REQ-5	Система повинна дозволяти користувачеві додати та редагувати інформацію про дисципліни
REQ-6	Система повинна дозволяти користувачеві додати та редагувати інформацію про групи
REQ-7	Система повинна дозволяти користувачеві формувати відомості про успішність
REQ-8	Система повинна дозволяти формувати різноманітну звітність

Таблиця 2.2 – Нефункціональні вимоги до додатка «Організація документообігу навчальної частини інституту»

Вимоги	Опис
REQ-9	Додаток повинен мати простий дизайн та зручну навігації
REQ-10	Поля повинні бути не порожніми, унікальними відносно вже існуючих записів

Таблиця 2.3 – Актори та цілі додатка «Організація документообігу навчальної частини інституту»

Актори	Цілі
Користувач	Мета користувача полягає в роботі з системою.
База даних	Мета бази даних полягає у зберігання інформації

Таблиця 2.4 – Опис варіантів використання додатка «Організація документообігу навчальної частини інституту»

Варіант використання	Ім'я	Опис
UC1	Ідентифікація в системі	Дозволяє користувачу пройти ідентифікацію в системі
UC2	Вивід каталогу користувачів	Дозволяє користувачеві з правами системного адміністратора вивести каталог всіх користувачів системи

UC3	Додати користувача	Дозволяє користувачеві додати нового користувача системи
UC4	Редагувати користувача	Дозволяє користувачеві редагувати інформацію вибраної із списку користувача
UC5	Видалити користувача	Дозволяє користувачеві видалити вибраного із списку користувача
UC6	Вивід каталогу студентів	Дозволяє користувачеві вивести каталог всіх студентів
UC7	Додати студента	Дозволяє користувачеві додати нового студента
UC8	Редагувати студента	Дозволяє користувачеві редагувати інформацію вибраного із списку студента
UC9	Видалити студента	Дозволяє користувачеві видалити вибраного із списку студента
UC10	Вивід каталогу дисципліни	Дозволяє користувачу вивести каталог всіх зареєстрованих у системі дисципліни
UC11	Додати дисципліну	Дозволяє користувачу додати інформацію про дисципліну
UC12	Редагувати дисципліну	Дозволяє користувачеві редагувати інформацію вибраної із списку дисципліни
UC13	Видалити дисципліну	Дозволяє користувачеві видалити вибрану дисципліну
UC14	Вивід каталогу груп	Дозволяє користувачу вивести список всіх груп
UC15	Додати групу	Дозволяє користувачу додати нову групу
UC16	Редагувати групу	Дозволяє користувачеві редагувати інформацію вибраної із списку групи

UC17	Видалити групу	Дозволяє користувачеві видалити вибрану із списку групу
UC18	Формування відомості	Дозволяє здійснювати формування відомості успішності студентів
UC19	Формування звітності	Дозволяє користувачеві програми формувати різноманітну звітність
UC20	Вивід системних подій	Дозволяє вивести всі події, які відбулися в системі

З поставлених вимог (див. розділ 1) тепер можна виставити повний опис вимог із сценаріями, що будуть входними даними для візуального моделювання мовою UML.

UC1 Ідентифікація в системі

Актор: користувач.

Ціль актора: пройти ідентифікацію в системі.

Задіяний актор: база даних.

Передумова: користувач ввівши нікнейм на пароль натискає кнопку «Підтвердити».

Післяумова: система відображає головне вікно програми з меню.

UC2 Вивід каталогу користувачів

Актор: користувач.

Ціль актора: вивести інформацію про всіх користувачів системи.

Задіяний актор: база даних.

Передумова: користувач натиснув на пункт меню «Користувачі».

Післяумова: система відображає екран для виведення списку всіх зареєстрованих користувачів системи.

UC3 Додати користувача

Актор: користувач.

Ціль актора: додати нового клієнта.

Задіяний актор: база даних.

Передумова: користувач ввівши всю необхідну інформацію про користувача натискає на кнопку «Додати».

Післяумова: система додає нового користувача та відображає екран із списком всіх користувачів.

UC4 Редагувати користувача

Актор: користувач.

Ціль актора: редагувати інформацію про вибраного користувача.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку користувача.

Післяумова: система відображає екран для редагування інформації про вибраного користувача.

UC5 Видалити користувача

Актор: користувач.

Ціль актора: видалити користувача із бази даних.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку користувача.

Післяумова: система відображає екран з можливістю видалення даних про користувача.

UC6 Вивід каталогу студентів

Актор: користувач.

Ціль актора: вивести інформацію про студентів.

Задіяний актор: база даних.

Передумова: користувач натиснув на пункт меню «Студенти».

Післяумова: система відображає екран для виведення списку всіх студентів.

UC7 Додати студента

Актор: користувач.

Ціль актора: додати нового студента.

Задіяний актор: база даних.

Передумова: користувач ввівши всю необхідну інформацію про студента натискає на кнопку «Додати».

Післяумова: система додає нового студента і відображає екран із списком всіх студентів.

UC8 Редагувати студента

Актор: користувач.

Ціль актора: редагувати вибраного із списку студента.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку студента.

Післяумова: система відображає екран для редагування інформації про вибраного студента.

UC9 Видалити студента

Актор: користувач.

Ціль актора: видалити вибраного студента із бази даних.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку студента.

Післяумова: система відображає екран з можливістю видалення даних студента.

UC10 Вивід каталогу дисципліни

Актор: користувач.

Ціль актора: вивести інформацію про всі дисципліни.

Задіяний актор: база даних.

Передумова: користувач натиснув на пункт меню «Дисципліни».

Післяумова: система відображає екран для виведення списку всіх дисципліни.

UC11 Додати дисципліну

Актор: користувач.

Ціль актора: додати інформацію про нову дисципліну.

Задіяний актор: база даних.

Передумова: користувач ввівши всю необхідну інформацію про дисципліну натискає на кнопку «Додати».

Післяумова: система додає нову дисципліну і відображає екран із списком всіх дисциплін.

UC12 Редагувати дисципліну

Актор: користувач.

Ціль актора: редагувати вибрану із списку дисципліну.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку дисципліну.

Післяумова: система відображає екран для редагування інформації про вибрану дисципліну.

UC13 Видалити дисципліну

Актор: користувач.

Ціль актора: видалити дисципліну із бази даних.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку дисципліну.

Післяумова: система відображає екран з можливістю видалення даних про дисципліну.

UC14 Вивід каталогу груп

Актор: користувач.

Ціль актора: вивести інформацію про всі групи.

Задіяний актор: база даних.

Передумова: користувач натиснув на пункт меню «Групи».

Післяумова: система відображає екран для виведення списку всіх груп.

UC15 Додати групу

Актор: користувач.

Ціль актора: додати нову групу.

Задіяний актор: база даних.

Передумова: користувач ввівши всю необхідну інформацію про групу натискає на кнопку «Додати».

Післяумова: система додає нову групу та відображає екран із списком всіх груп.

UC16 Редагувати групу

Актор: користувач.

Ціль актора: редагувати вибрану із списку групу.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідну із списку групу.

Післяумова: система відображає екран для редагування інформації про вибрану групу.

UC17 Видалити групу

Актор: користувач.

Ціль актора: видалити групу із бази даних.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідну із списку групу.

Післяумова: система відображає екран з можливістю видалення даних про групу.

UC18 Формування відомості

Актор: користувач.

Ціль актора: здійснити формування нової відомості про успішність студентів.

Задіяний актор: база даних.

Передумова: користувач натискає на пункт меню «Відомість».

Післяумова: система відображає екран з можливістю формування нової відомості.

UC19 Формування звітності

Актор: користувач.

Ціль актора: формування різноманітної звітності для навчальної частини інституту.

Задіяний актор: база даних.

Передумова: користувач користуючись меню «Звіти» має можливість формування будь-якої звітності.

Післяумова: система відображає екран з можливістю формування звітності за необхідними критеріями.

UC20 Вивід системних подій

Актор: користувач.

Ціль актора: вивести список всіх подій в системі.

Задіяний актор: база даних.

Передумова: користувач вибирає пункт меню «Системний журнал».

Післяумова: система відображає екран із списком всіх системних подій.

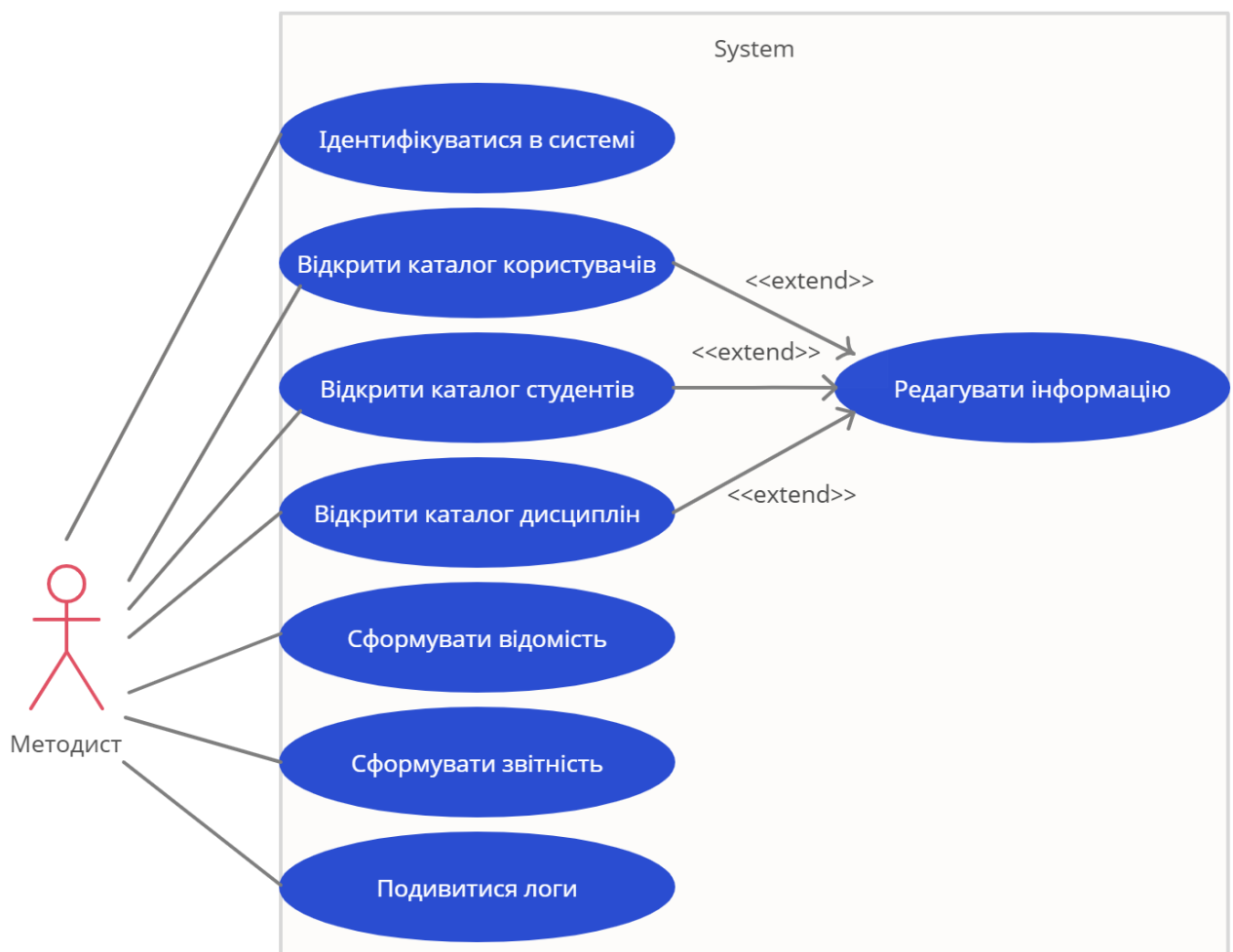


Рисунок 2.1 – Діаграма use-case

2.2 Архітектура проекту

Розроблений продукт повинен відповідати характеристикам якості, таким як: стійкість, корисність, доступність, масштабованість, відкритість, гнучкість, можливість тестування. Це вимагає від процесу розробки додаткові обмеження/правила, а саме:

- дотримання шаблонів і стилів;
- документування розробки на різних рівнях;
- тестування компонентів, окремих модулів, підсистем;
- управління проектами, процесами.

З урахуванням вимог до забезпечення стійкості та гнучкості системи при її розробці було обрано шаблон Layers, який розбиває систему на дві частини: клієнт та сервер.

Проектування системи буде покладатись на предметну область (DDD підхід) та принципи SOLID [5].

При розробці клієнта був обраний користувацький інтерфейс Windows Forms.

Сервер, в свою чергу, буде складатись з таких модулів:

- 1) BLL (англ. Business Logic Layer) – логіка та всі необхідні обчислення додатку на мові бізнесу;
- 2) DAL (англ. Data Acces Layer) – рівень доступу до даних.
- 3) DB (англ. Data Base) – база даних для зберігання даних.

З урахування всіх вище перерахованих шаблонів структура проекту буде виглядати наступним чином:

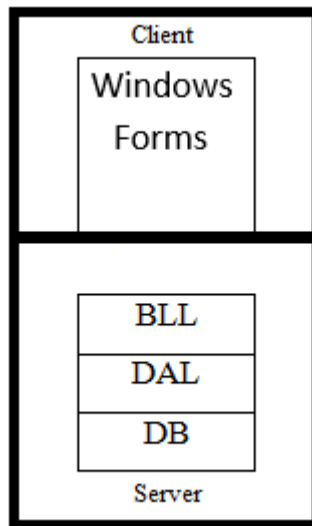


Рисунок 2.2 – Структура проекту

Кожен компонент буде реалізувати контракт (інтерфейс), який надає на гнучкість компоненту.

Оскільки бізнес постійно і відносно швидко змінюється, то кожен із компонентів повинен швидко адаптуватись під ці зміни. Для цього використаємо шаблон Dependency Injection (DI).

2.3 Особливості розробки бази даних. ERD діаграма з описанням сутностей

Схема "сутність-зв'язок" (також ERD або ER-діаграма) - це різновид блок-схеми, де показано, як різні "сутності" (люди, об'єкти, концепції і так далі) пов'язані між собою всередині системи. ER-діаграми найчастіше застосовуються для проектування та налагодження реляційних баз даних у сфері освіти, дослідження та розробки програмного забезпечення та інформаційних систем для бізнесу.

ER-діаграми (або ER-моделі) покладаються на стандартний набір символів, включаючи прямокутники, ромби, овали та сполучні лінії для відображення сутностей, їх атрибутів та зв'язків. Ці діаграми влаштовані за тим самим

принципом, як і граматичні структури: сутності виконують роль іменників, а зв'язку — дієслів.

ER-діаграми - "родичі" схем структури даних (DSD), де замість зв'язків між самими сутностями відображається відношення між елементами всередині них. ER-діаграми часто використовуються у поєднанні з діаграмами DFD, що схематично показують рух потоків інформації в рамках процесу або системи.

У ER-моделях та моделях даних зазвичай виділяють до трьох рівнів деталізації:

Концептуальна модель даних – схема найвищого рівня з мінімальною кількістю подробиць. Перевага цього підходу полягає у можливості відобразити загальну структуру моделі та всю архітектуру системи. Менш масштабні системи можуть обійтися без цієї моделі. І тут можна відразу переходити до логічної моделі.

Логічна модель даних: містить більш детальну інформацію, ніж концептуальна модель. На цьому рівні визначаються докладніші операційні та транзакційні сутності. Логічна модель залежить від технології, у якій застосовуватиметься.

Фізична модель даних: на основі кожної логічної моделі даних можна становити одну або дві фізичні моделі. В останніх має бути достатньо технічних подробиць для складання та впровадження самої бази даних.

Можна звернути увагу на той факт, що схожі рівні масштабу та деталізації зустрічаються і в інших видах схем (наприклад, у діаграмах DFD), проте дана класифікація відрізняється від трьохсхемного підходу у розробці ПЗ, де розподіл інформації здійснюється за дещо іншим принципом. Щоправда, іноді розробники застосовують ER-діаграми з додатковими ієрархіями, якщо дизайн бази даних потребує більше інформаційних рівнів. Наприклад, розробник може додати нові групи за принципом розширення вгору (суперкласи) та вниз (підкласи). А саме:

- ✓ Лише реляційні дані. Слід чітко розуміти, що мета ER-діаграм – показати зв'язки та відносини між елементами, тому вони відображають лише реляційну структуру.
- ✓ Лише для структурованих даних. Дані мають бути чітко розбиті на поля, стовпці та рядки, інакше користі від ER-діаграми буде мало. Це стосується і частково структурованих даних, оскільки лише деякі з них будуть придатними для роботи.
- ✓ Складність інтеграції з базою даних. Застосування ER-моделей для інтеграції з існуючою базою даних — непросте завдання через різницю в архітектурі.

Згідно завдання будемо ERD діаграму.

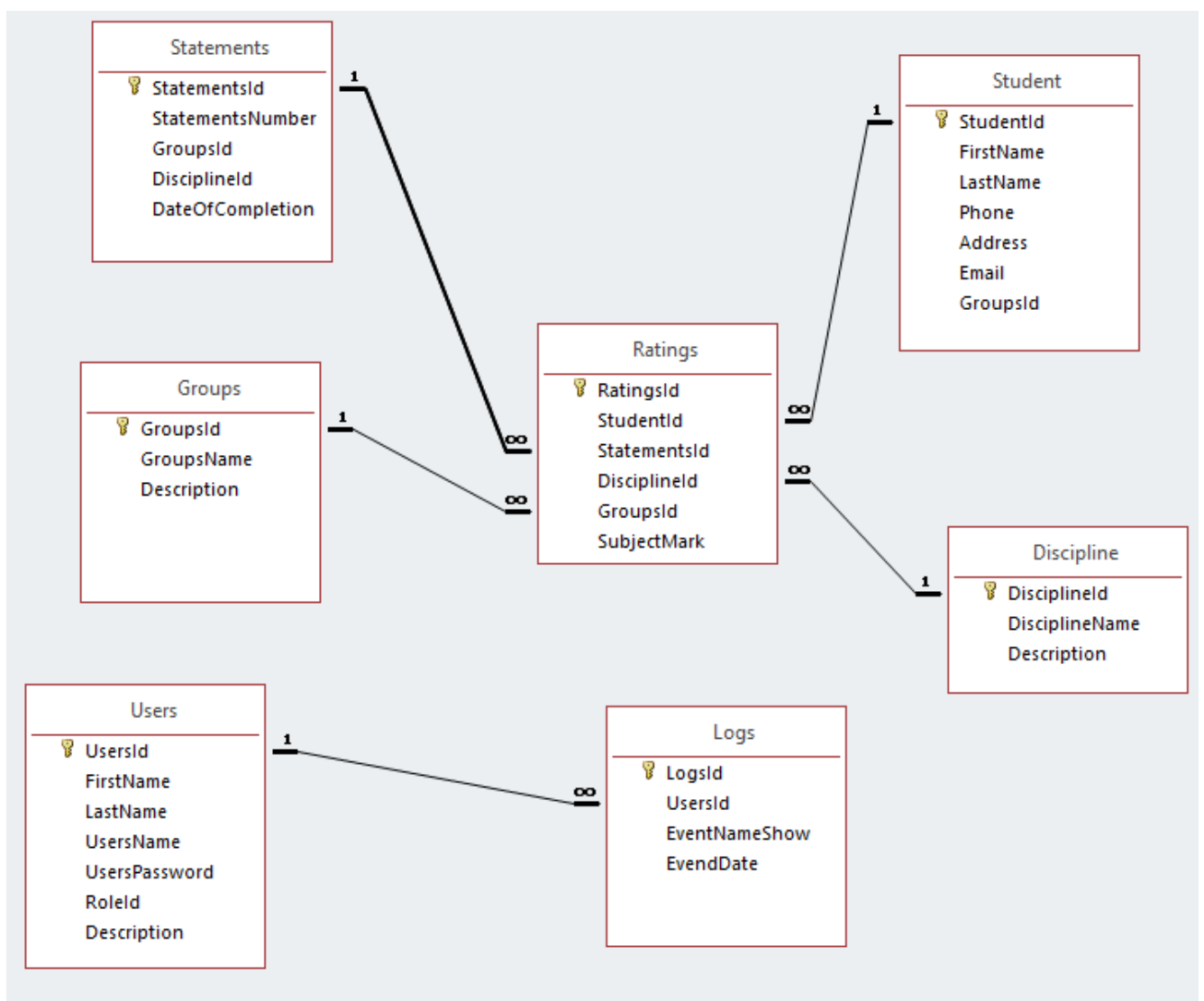


Рисунок 2.3 – ERD діаграма.

Як ми бачимо на рис. 2.3 наша база даних складається із 7 сутностей. Кожна сутність має свою таблицю, а саме:

1. Таблиця «Discipline» – зберігає інформацію про дисципліни.
2. Таблиця «Student» - зберігає інформацію про студентів.
3. Таблиця «Logs» - зберігає інформацію про активність користувачів системи і їхні дії.
4. Таблиця «Groups» - зберігає інформацію про навчальні групи.
5. Таблиця «Ratings» - зберігає інформацію про всі виставлені у відомість оцінки.
6. Таблиця «Statements» - зберігає інформацію про відомості успішності навчання студентів.
7. Таблиця «Users» - містить інформацію про всіх користувачів системи та їхні облікові дані.

2.4 Особливості розробки рівня BLL

Бізнес-логіка – у розробці інформаційних систем – сукупність правил, принципів, залежностей поведінки об'єктів предметної галузі (області людської діяльності, яку система підтримує). Інакше можна сказати, що бізнес-логіка — це реалізація правил та обмежень операцій, що автоматизуються. Є синонімом терміну "логіка предметної області" (англ. domain logic). Бізнес-логіка визначає правила, яким підпорядковуються дані предметної області.

Простіше кажучи, бізнес-логіка – це реалізація предметної галузі в інформаційній системі. До неї відносяться: формули розрахунку щомісячних виплат з позик (у фінансовій індустрії), автоматизована відправка повідомлень електронної пошти керівнику проекту після закінчення частин завдання всіма

підлеглими (у системах управління проектами), відмова від готелю при скасуванні рейсу авіакомпанією (у туристичному бізнесі)) і т.д.

У фазі бізнес-моделювання та розробки вимог бізнес-логіка описується у вигляді:

- ✓ тексту;
- ✓ концептуальних аналітичних моделей предметної галузі (онтології);
- ✓ бізнес-правил;
- ✓ різноманітних алгоритмів;
- ✓ діаграм діяльності;
- ✓ графів та діаграм переходу станів;
- ✓ моделей бізнес-процесів.

У фазі аналізу та проектування системи бізнес-логіка втілюється в різних діаграмах мови UML або подібних до неї. У фазі програмування бізнес-логіка втілюється в коді класів та їх методів, у разі використання об'єктно-орієнтованих мов програмування, або процедур та функцій у разі застосування процедурних мов.

На жаргоні розробників програмного забезпечення «бізнес-логікою» також називаються програмні модулі, що її реалізують, та рівень системи, на якому ці модулі знаходяться (англ. business logic layer).

У багаторівневих (багатошарових) інформаційних системах цей рівень взаємодіє з нижчим рівнем інфраструктурних сервісів (англ. infrastructure layer), наприклад, інтерфейсом доступу до бази даних або файлової системи (англ. data-access layer, DAL) і рівнем сервісів додатка (англ. application services layer), який, своєю чергою, взаємодіє з рівнем користувальницького інтерфейсу (англ. user interface layer) чи зовнішніми системами.

BLL буде обробляти речі, які є частиною бізнес-сфери, а не частиною бази даних і не частиною користувальницького інтерфейсу.

В даному проекті реалізовано клас: один - для формування звітності по виконаних заявах надання гуманітарної допомоги: кількість видач по кожному предмету гуманітарної допомоги за вибраний період, а також інформацію про дисципліну.

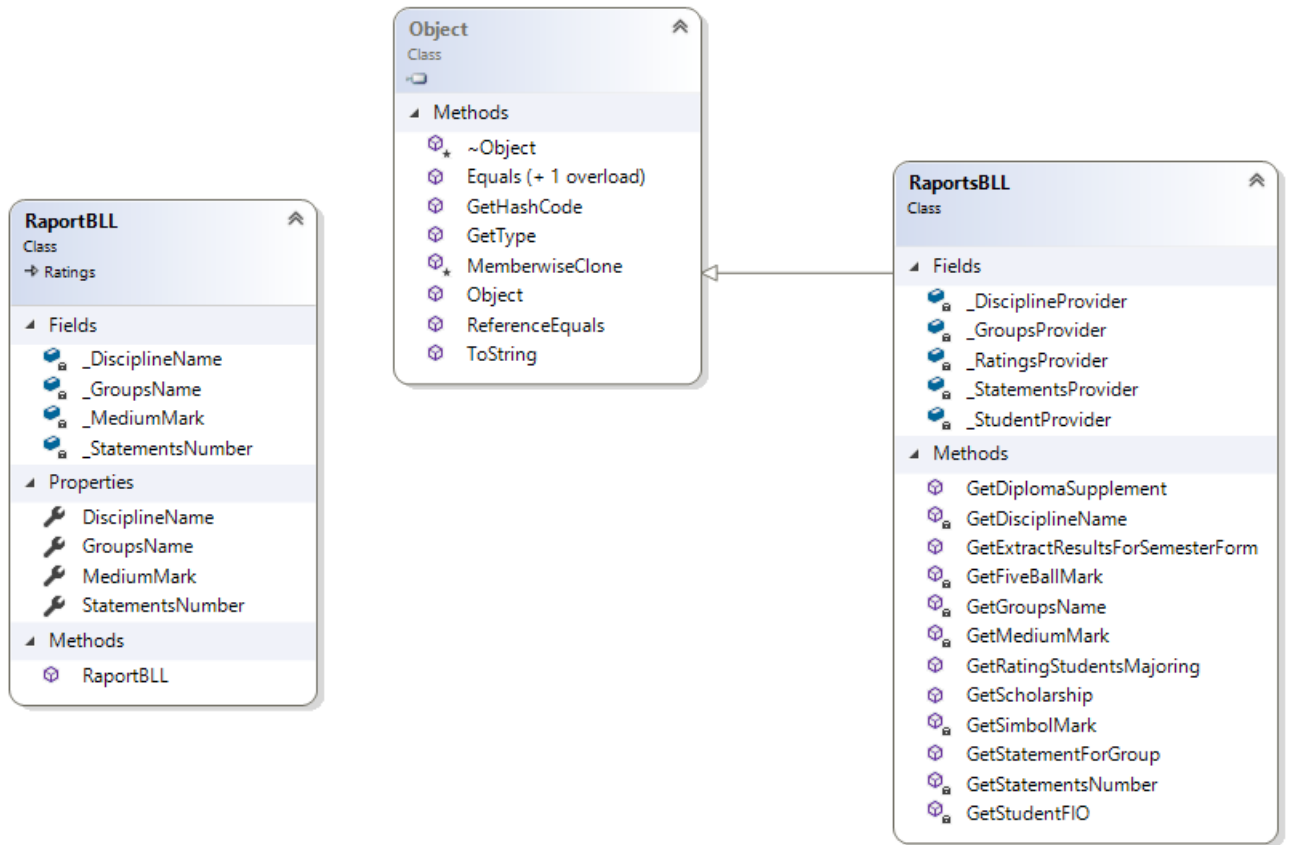


Рисунок 2.4 – Діаграма класу BLL «Організація документообігу навчальної частини інституту»

2.5 Особливість реалізації бізнес логіки – діаграма домена.

Будуємо діаграму класів домена. Кожен клас описує конкретну таблицю бази даних для зручного опрацювання даних (рис. 2.5).

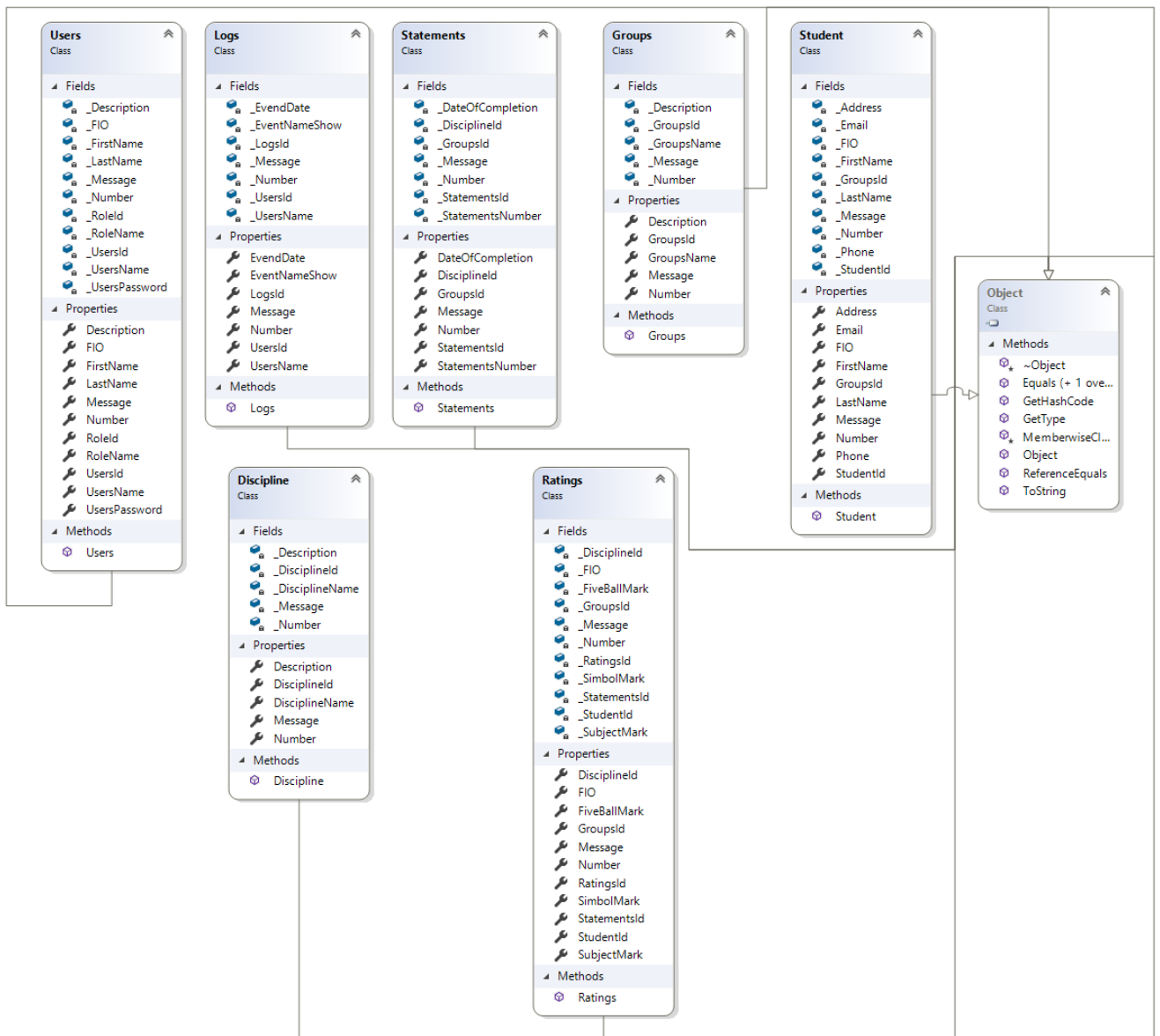


Рисунок 2.5. Діаграма класів домена

2.6 Особливості розробки рівня UI

User interface (UI) елементи – це частини, які дизайнери використовують для створення програм або веб-сайтів. Вони додають інтерактивність в інтерфейс користувача, надаючи користувачеві точки зіткнення при навігації по них. Кнопки, смуги прокручування, пункти меню та чекбокси.

Інтерфейсу користувача (UI) використовує UI елементи для створення візуальної мови і забезпечення узгодженості продукту, що робить його зручним для користувача і простим у навігації без особливих зусиль з боку користувача.

UI елементи зазвичай поділяються на одну з наступних чотирьох груп:

- Елементи керування введенням (Input Controls) – дозволяють користувачам вводити інформацію до системи.
- Компоненти навігації (Navigation Components) – допомагають користувачам переміщатися продуктом або веб-сайтом. Загальні навігаційні компоненти включають панелі вкладок та головне меню програми.
- Інформаційні компоненти (Informational Components) – діляться інформацією з користувачем.
- Контейнери (Containers) – містять зв'язаний контент разом.

В даному проєкті для розробки користувацького інтерфейсу я використав Windows Forms.

Windows Forms - це платформа користувача інтерфейсу для створення класичних додатків Windows. Вона забезпечує один з найефективніших способів створення класичних додатків за допомогою візуального конструктора в Visual Studio.

На рис. 2.6 показана діаграма програми «Організація документообігу навчальної частини інституту» з правами адміністратора системи.

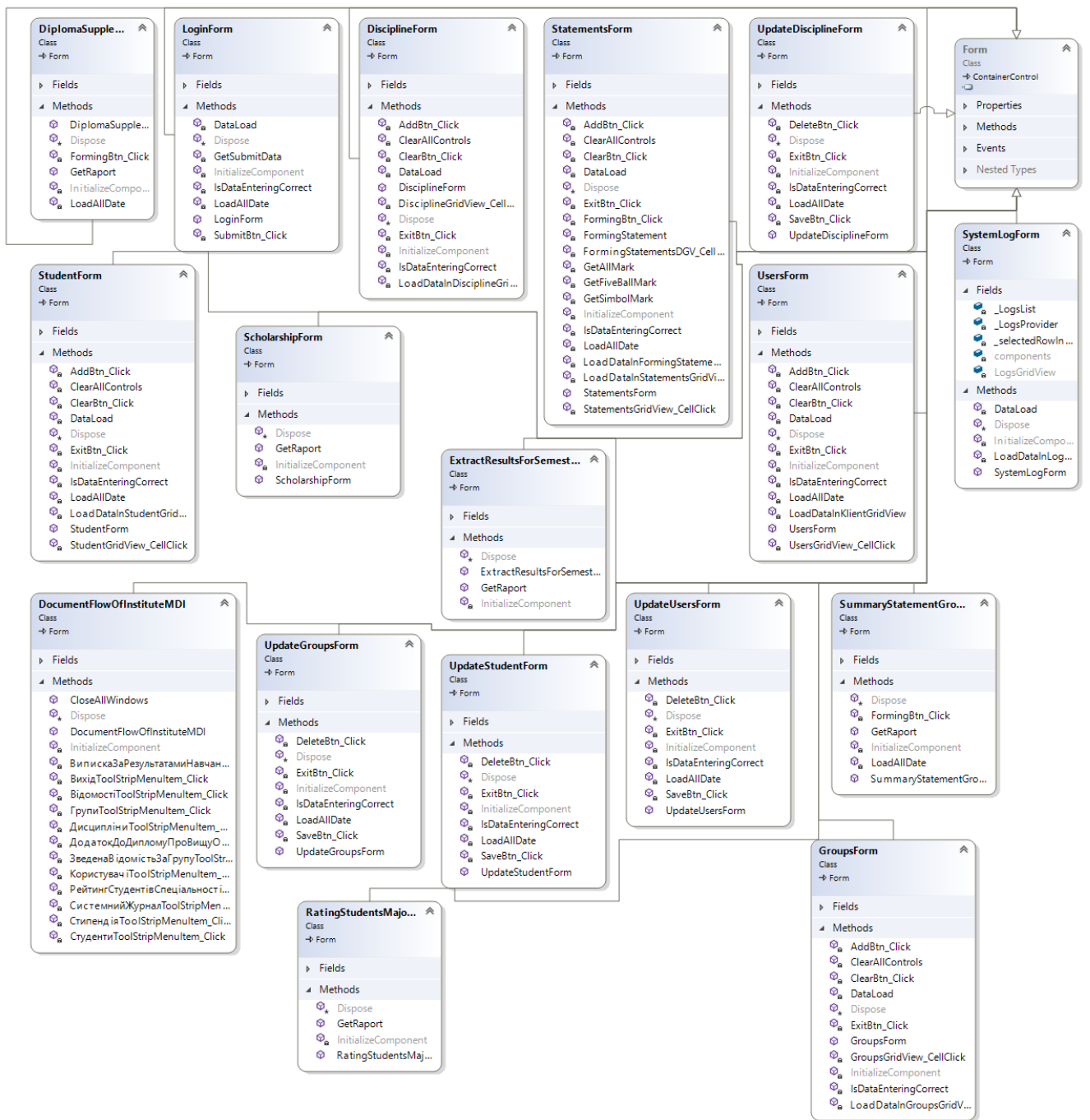


Рисунок 2.6 – Діаграма інтерфейсу програми «Організація документообігу навчальної частини інституту»

2.7 Особливості розробки DAL

Шар доступу до даних (Data Access Layer – DAL) у програмному забезпеченні – це шар комп'ютерної програми, який надає спрощений доступ до даних, що

зберігаються у постійному сховищі якогось типу, такому як реляційна база даних. Цей акронім в основному використовується в оточенні Microsoft.NET.

DAL може повертати посилання на об'єкт (у термінах об'єктно-орієнтованого програмування) з його атрибутами замість рядків полів із таблиці бази даних. Це дозволяє створювати клієнтські (або модулі користувача) модулі з більш високим рівнем абстракції. Така модель може бути реалізована шляхом створення класу з методами доступу до даних, які безпосередньо посилаються на відповідний набір процедур бази даних. Інша реалізація може потенційно отримувати або записувати записи або з файлової системи. DAL приховує складність сховища даних, що лежить в основі даних.

Замість використання таких команд як «створити», «видалити» або «оновити» в певній таблиці в базі, клас і кілька процедур, що зберігаються, можуть бути створені в базі. Ці процедури можуть викликатися методом усередині класу, який поверне об'єкт, що містить запитані значення. Або команди створення, видалення та оновлення можуть бути виконані всередині простих функцій, що зберігаються у шарі доступу до даних.

Також методи бізнес-логіки із програми можуть бути співвіднесені до шару доступу до даних.

Для роботи з базою даних було реалізовано 7 класів. Назва кожного класу починається із відповідній їй назві таблиці в базі даних та закінчується приставкою «Provider».

На рис. 2.7 приведено діаграму класів з методами для роботи з базою даних.

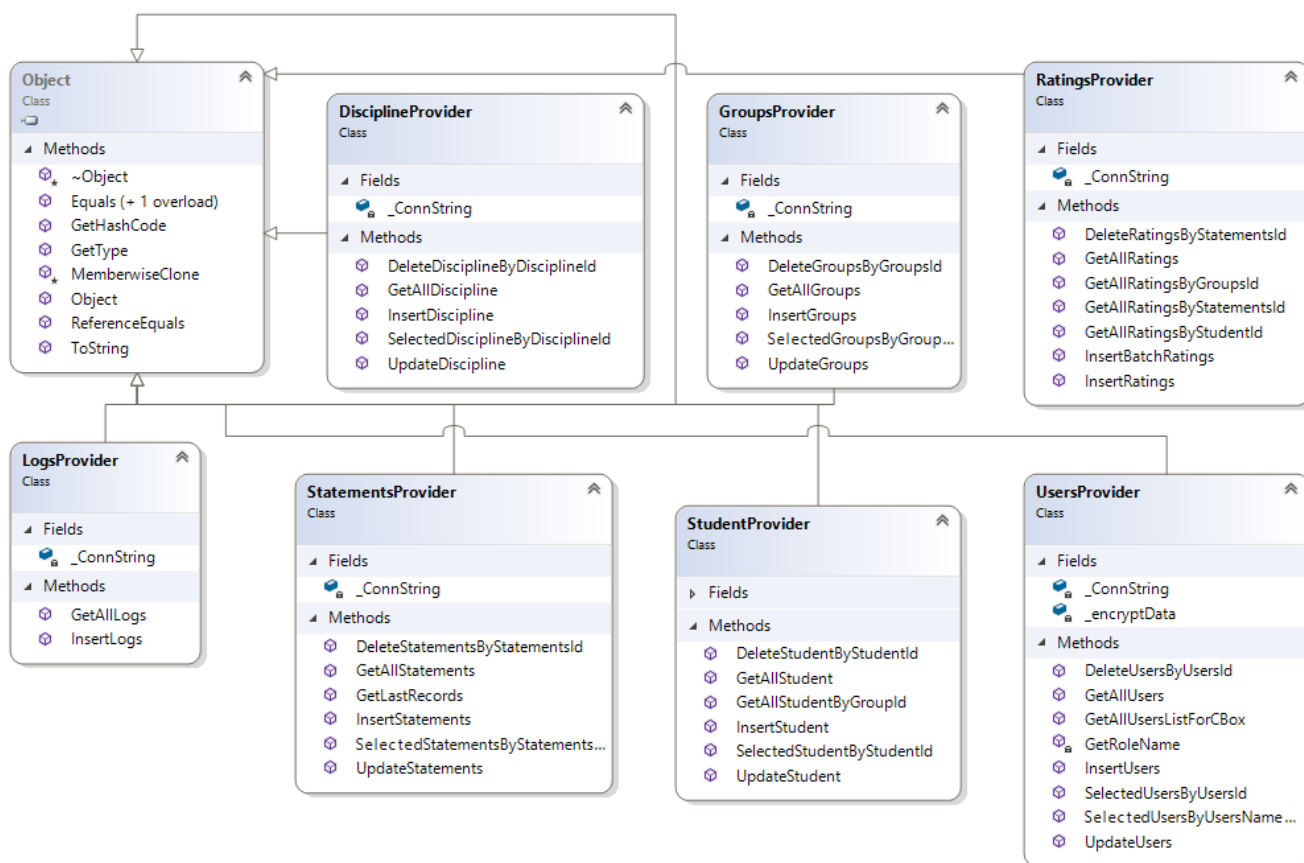


Рисунок 2.7 – Діаграма класів з методами для роботи з базою даних

2.8 Висновок

Побудова сучасних інформаційних систем займає дуже багато часу. Вона починається з аналізу предметної області, детального планування системи, описання вимог, моделювання її поведінки за допомогою uml діаграм. Визначаються шаблони, які будуть використовуватись при розробці.

Після планування починається стадія розробки. Розробка починається зі створення користувацького інтерфейсу і закінчується базою даних.

При розробці слід враховувати, що вимоги змінюються швидко і потрібно будувати систему так, щоб вона була гнучкою.

Розробка системи виконується по окремим компонентам. Кожний створений компонент потрібно детально тестувати, щоб мінімізувати помилки на наступних етапах розробки.

Якщо дотримуватися всіх вищеперерахованих вимог, то можна побудувати гнучку і стійку інформаційну систему.

3 РЕАЛІЗАЦІЯ

3.1 Вибір технологій

Для побудови користувацького інтерфейсу по шаблону будемо використовувати WinForms.

Для розробки інформаційної бази використовувався Microsoft Access. система управління базами даних, програма, що входить до складу пакету офісних програм Microsoft Office. Має широкий спектр функцій, включаючи зв'язані запити, сортування по різних полях, зв'язок із зовнішніми таблицями і базами даних.

Microsoft Access, володіючи всіма рисами класичної СУБД, надає і додаткові можливості. Access - це не тільки потужна, гнучка і проста у використанні СУБД, але і система для розробки працюючих з базами даних додатків. За допомогою Access можна створити додаток, що працює в середовищі Windows і повністю відповідає потребам по управлінню даними. Використовуючи запити, є можливість вибирати і обробляти що зберігається в таблицях інформацію.

Для роботи з Базою даних реалізуємо класи (рис.3.1). В кожному класі є методи, за допомогою яких можна: вибирати, вставляти, редагувати та видаляти дані. Також є методи, які працюють з вибіркою даних із декількох таблиць.

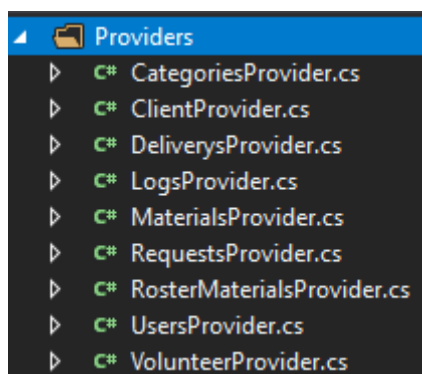


Рисунок 3.1 – Класи для роботи з базою даних

3.2 Результати функціонального тестування розробленого додатку

Для того, щоб провести тестування необхідно запустити програму із назвою «DocumentFlowOfInstitute». Після запуску програми буде виведено вікно, де користувачу буде запропоновано ввести ім'я та пароль (рис.3.2.).

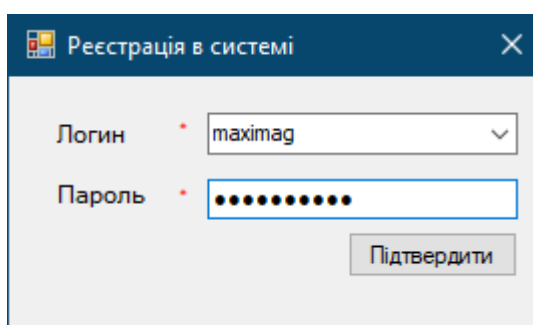


Рисунок 3.2. Реєстрація користувача в системі.

Якщо ж дані буде введена інформація буде не коректною, програма буде попереджувати про це користувача відповідним повідомленням (рис.3.3.).

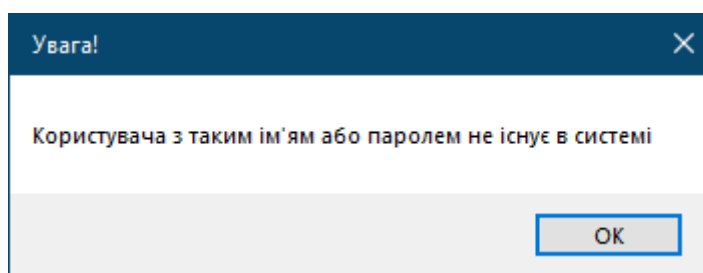


Рисунок 3.3. Попередження про неправильне введення даних.

Для швидшого пошуку можна вибрати ім'я із випадаючого списку або під час введення даних програма поставить у верх списку те ім'я, яке буде вводиться з клавіатури.

Після успішної ідентифікації користувача системи буде відкрите головне вікно програми з основним меню (рис.3.4.).

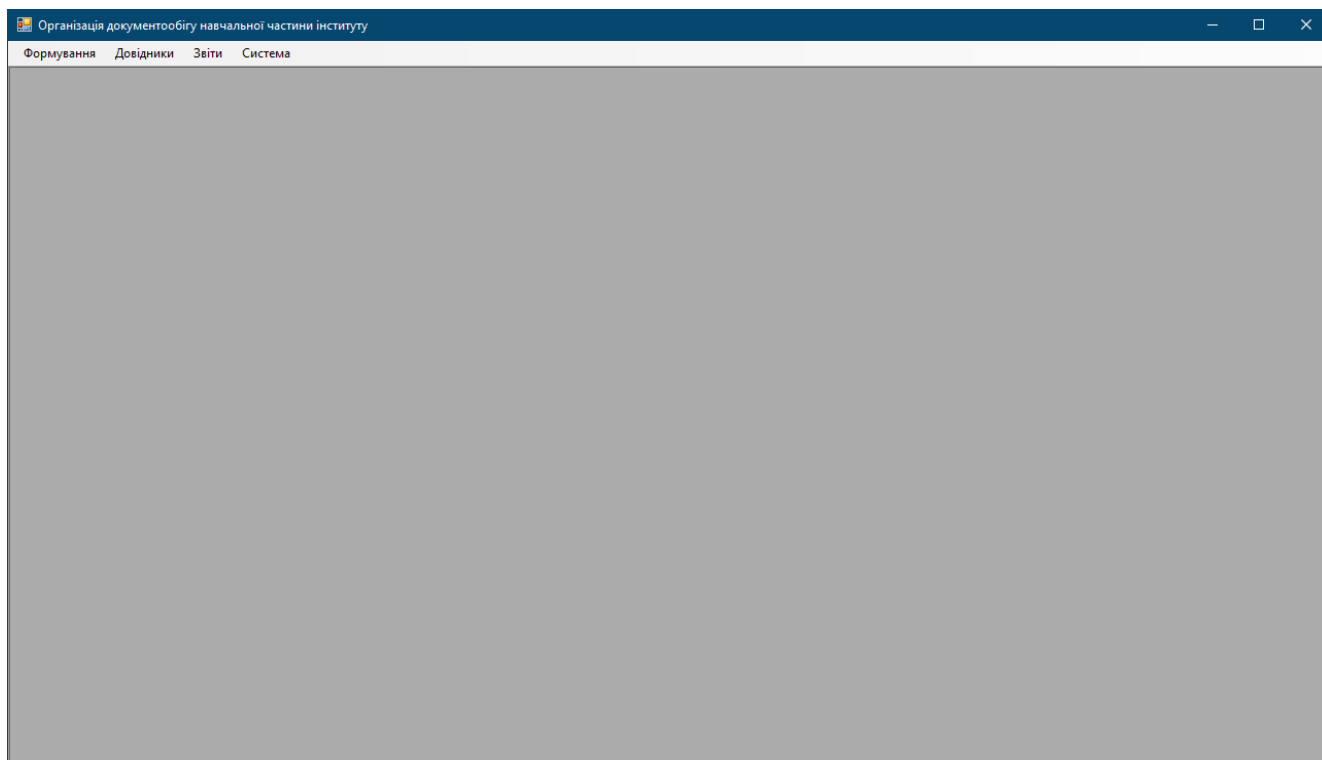


Рисунок 3.4 – Головне вікно програми

Так, як ми здійснили вхід з правами системного адміністратора, для нас відкриті всі можливі функції системи.

Для продовження роботи створимо дисципліну, та введемо всі необхідні дані про неї (рис 3.5.). Також в даному вікні відображається список всіх дисциплін, які вже додані в системі.

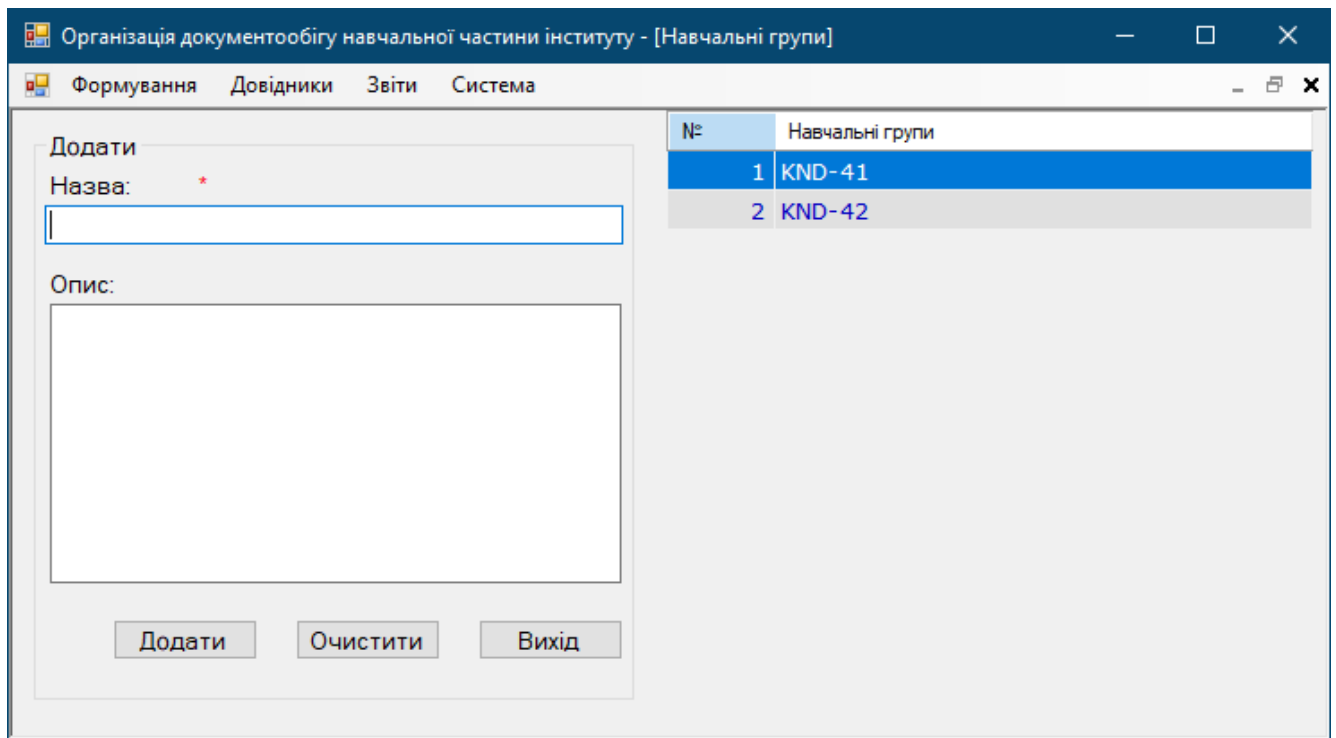


Рисунок 3.5 – Вікно для введення даних про дисципліну

Система перевіряє всі обов'язкові поля на коректність вводу інформації. Обов'язкові поля позначаються зірочкою. В даному випадку недопустимим буде введення пустого поля «Назва».

Також реалізована можливість редагування та видалення даних у випадку якщо це є необхідним. На рис 3.6 зображено вікно для редагування даних про навчальну групу.

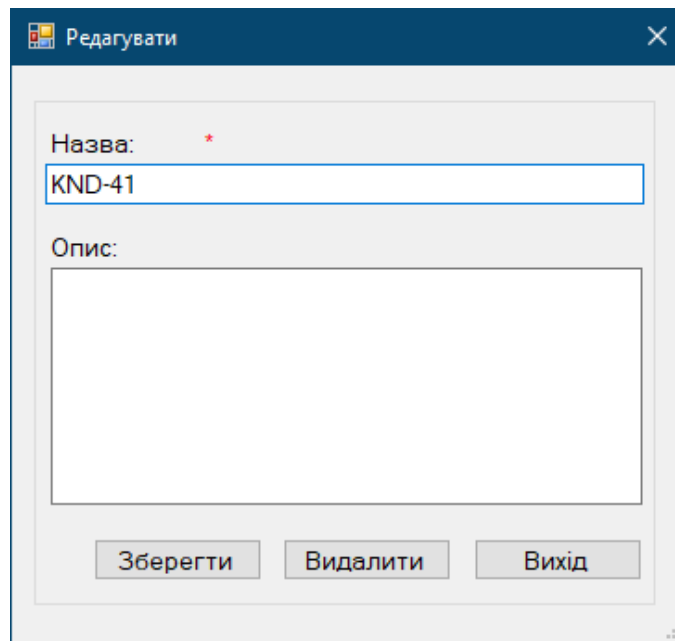


Рисунок 3.6 – Вікно для редагування даних вибраної групи

У системі є можливість переглянути список навчальних дисциплін. Для цього необхідно перейти по головному меню «Довідники» → «Дисципліни», після чого відкриється відповідне вікно. Також в цьому вікні є можливість додати нову дисципліну (рис. 3.7.).

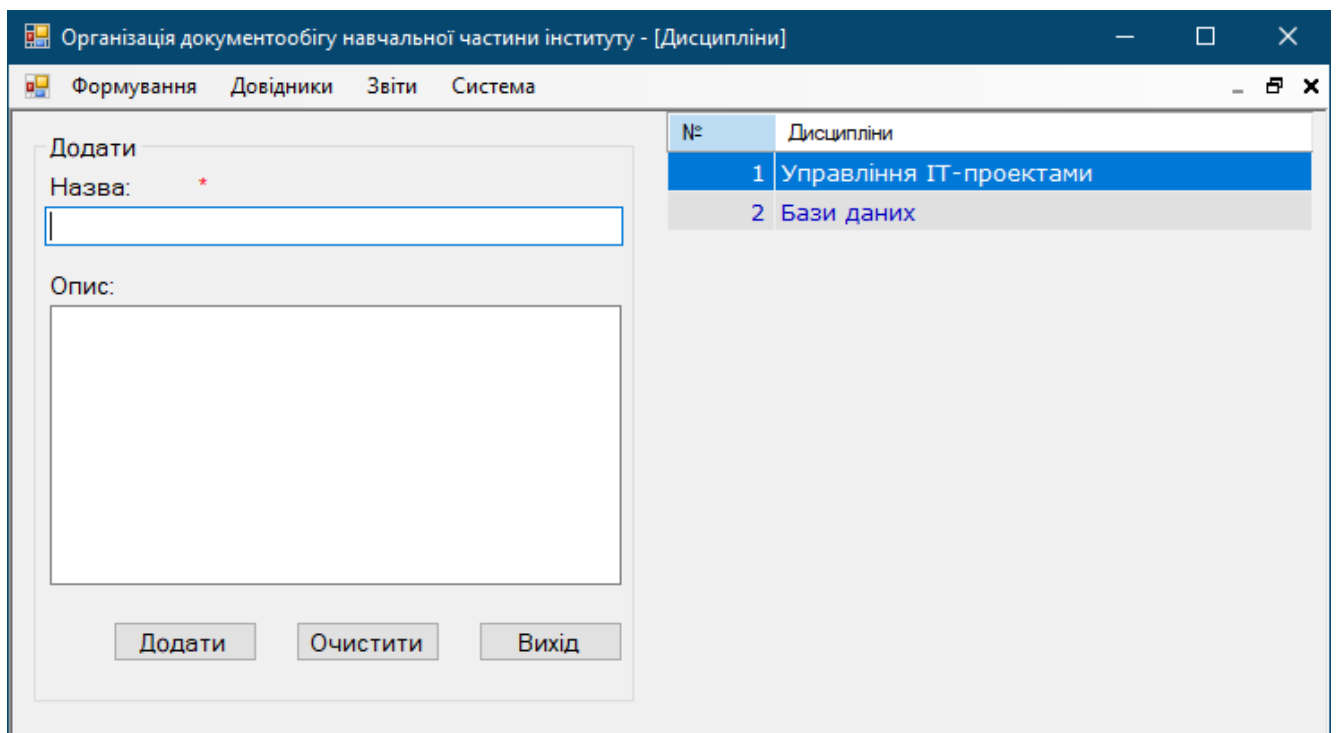


Рисунок 3.7 – Вікно для додавання нової дисципліни

Також у випадку помилки, дані про введені дані навчальної дисципліни можна редагувати, або якщо вона є не актуальною, то її можна видалити з системи (рис. 3.8).

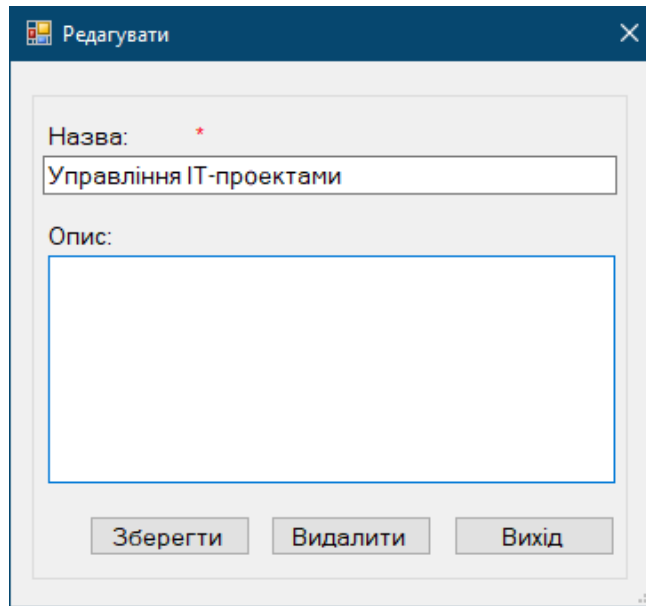


Рисунок 3.8 – Вікно для редагування вибраної дисципліни

Для ведення інформаційної бази студентів реалізована можливість додавання та переглядання інформації про них. Щоб це зробити необхідно перейти по меню «Довідники» → «Студенти». Після цього виведеться вікно зі списком всіх зареєстрованих студентів (рис. 3.9).

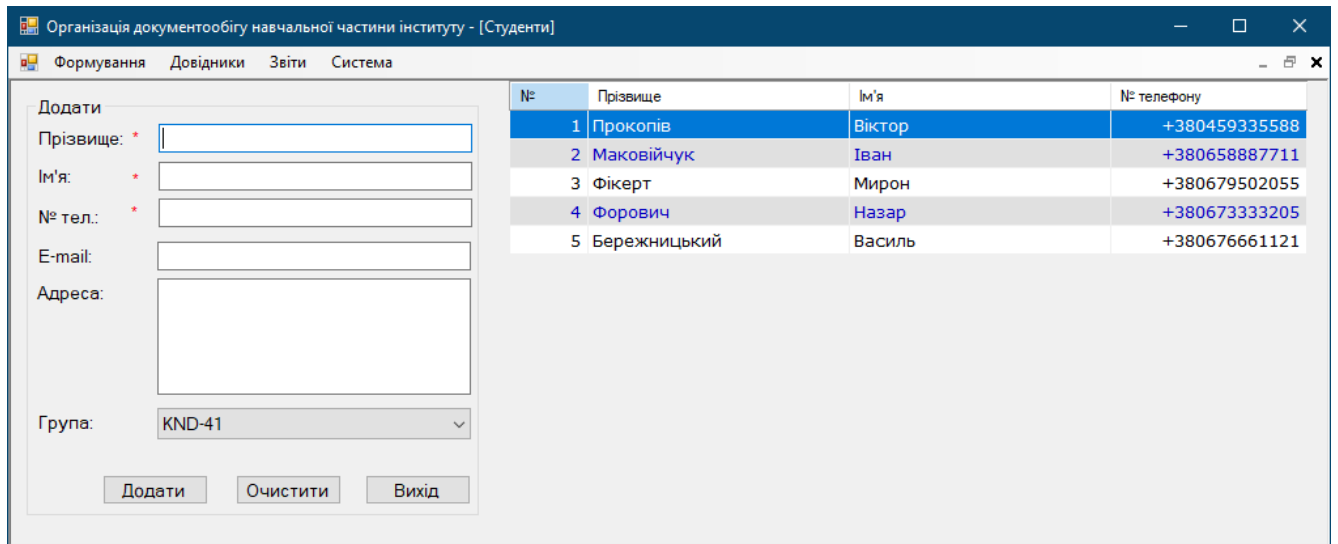


Рисунок 3.9. Вікно для додавання інформації про студентів

В разі необхідності програму можна відредагувати або видалити. Для цього у списку справа потрібно натиснути лівою кнопкою на по необхідному запису, після чого з'явиться вікно з можливістю редагування/видалення інформації про студента (рис. 3.10).

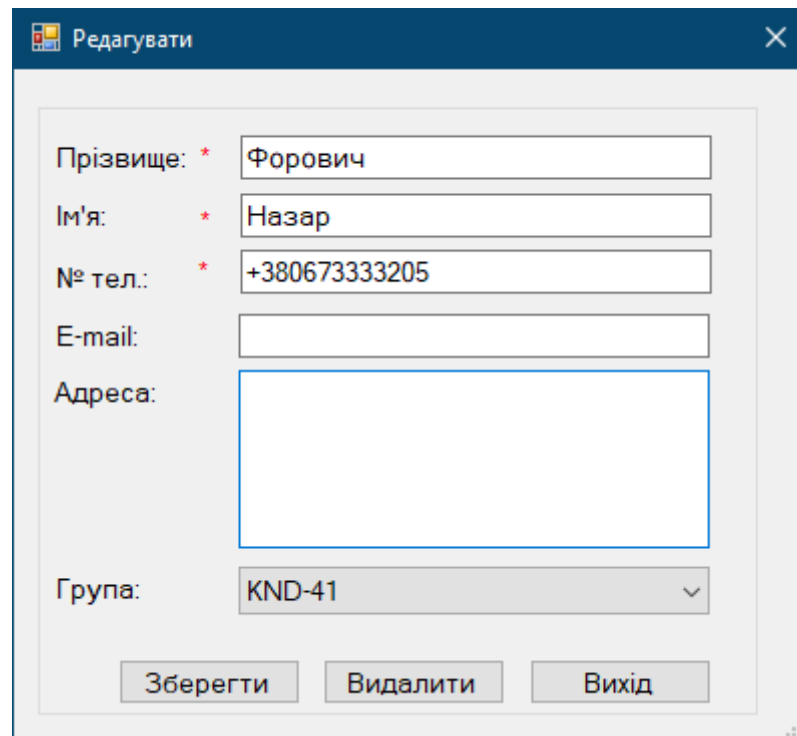


Рисунок 3.10 – Вікно для редагування інформації про студента

Для формування нової відомості про успішність навчання студентів необхідно перейти по головному меню програми «Формування» → «Відомості», після чого відкриється вікно з можливістю формування нової відомості. (рис. 3.11).

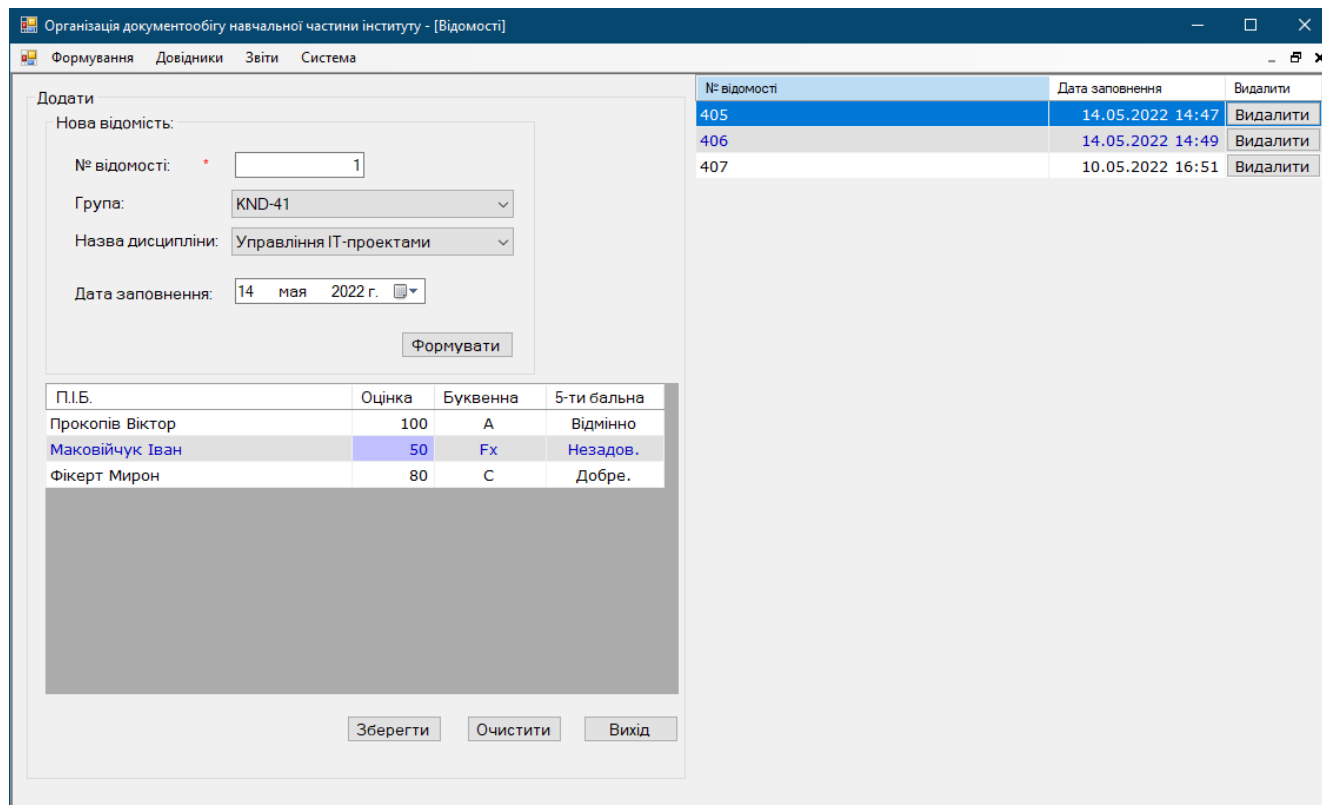


Рисунок 3.11 – Вікно для формування нової відомості

Важливою функцією системи є формування звітності, за яке відповідає меню «Звіти». В даному меню реалізовано 5 різних типів звітності.

На рис. 3.12 зображене вікно для формування зведеної відомості за групу.

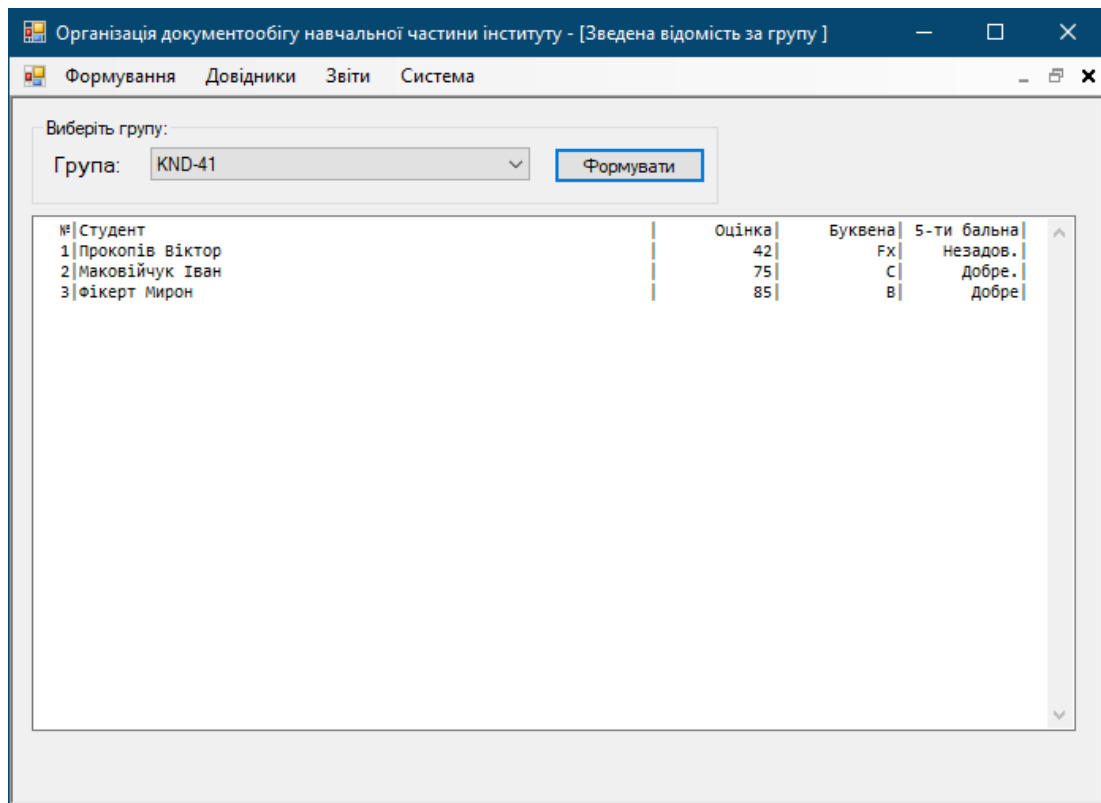


Рисунок 3.12 – Вікно для формування зведеної відомості за групу

Для того, щоб взнати рейтинг студентів спеціальності по курсам необхідно перейти по меню «Звіти» → «Рейтинг студентів спеціальності по курсам», після чого відкриється відповідне вікно із сформованим рейтингом (рис. 3.13).

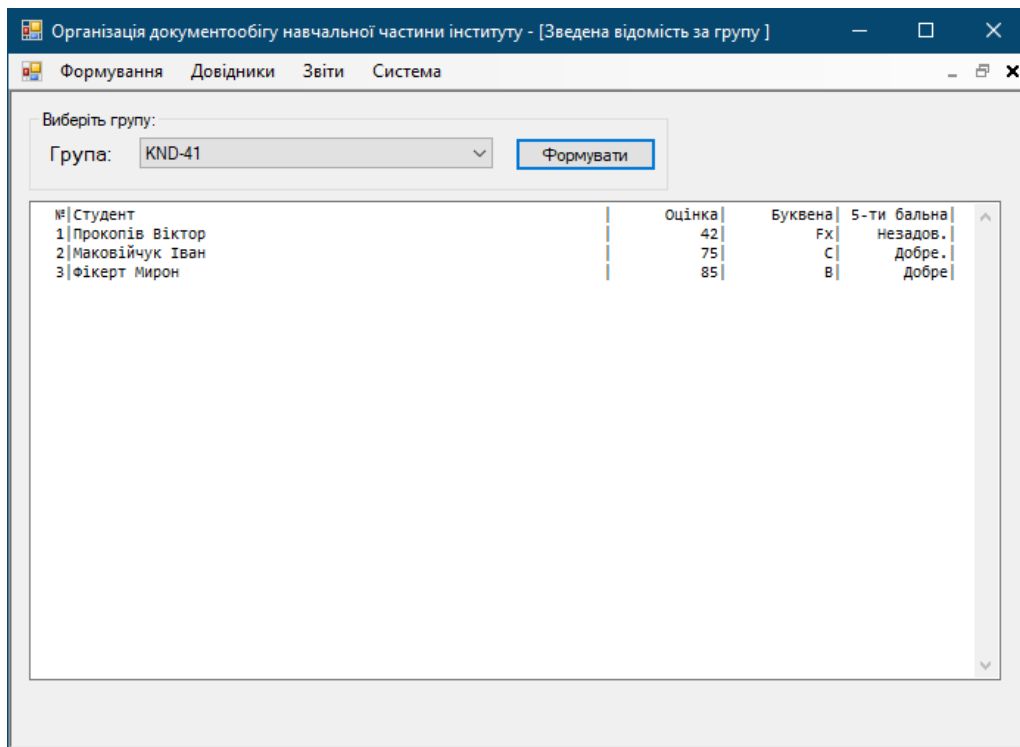


Рисунок 3.13 – Вікно для формування рейтингу студентів спеціальності по курсам

Для нарахування стипендії для студентів можна скористатись звітністю «Стипендія» (рис. 3.14).

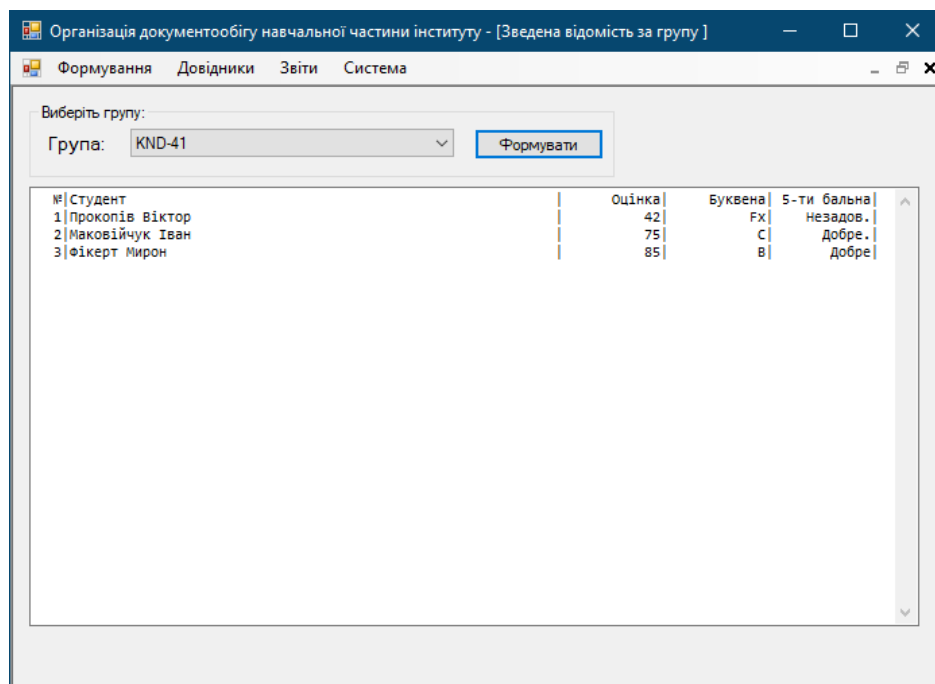
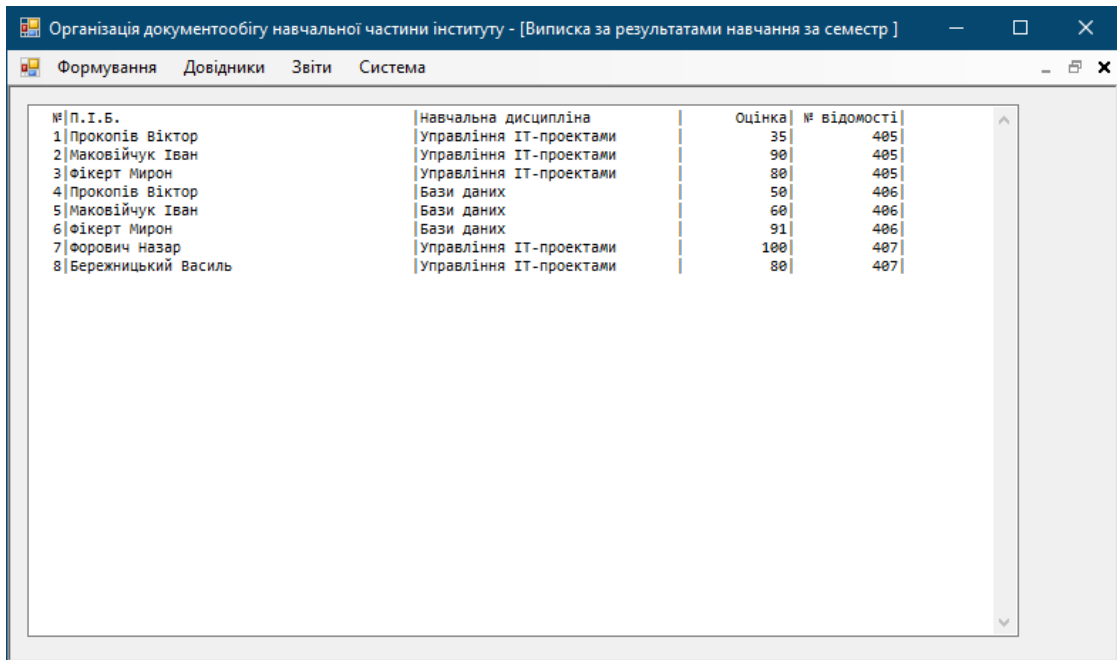


Рисунок 3.14 – Вікно для нарахування стипендії

Щоб отримати виписку за результатами навчання за семестр необхідно перейти по меню «Звіти» → «Виписка за результатами навчання за семестр», після чого відкриється відповідне вікно із випискою (рис. 3.15).



№ П.І.Б.	Навчальна дисципліна	Оцінка	№ відомості
1 Прокопів Віктор	Управління ІТ-проектами	35	405
2 Маковійчук Іван	Управління ІТ-проектами	90	405
3 Фікерт Мирон	Управління ІТ-проектами	80	405
4 Прокопів Віктор	Бази даних	50	406
5 Маковійчук Іван	Бази даних	60	406
6 Фікерт Мирон	Бази даних	91	406
7 Форович Назар	Управління ІТ-проектами	100	407
8 Бережницький Василь	Управління ІТ-проектами	80	407

Рисунок 3.15 – Виписка за результатами навчання за семестр

Для створення додатку до диплому про вищу освіту можна скористатись меню «Додаток до диплому про вищу освіту» (рис. 3.16).

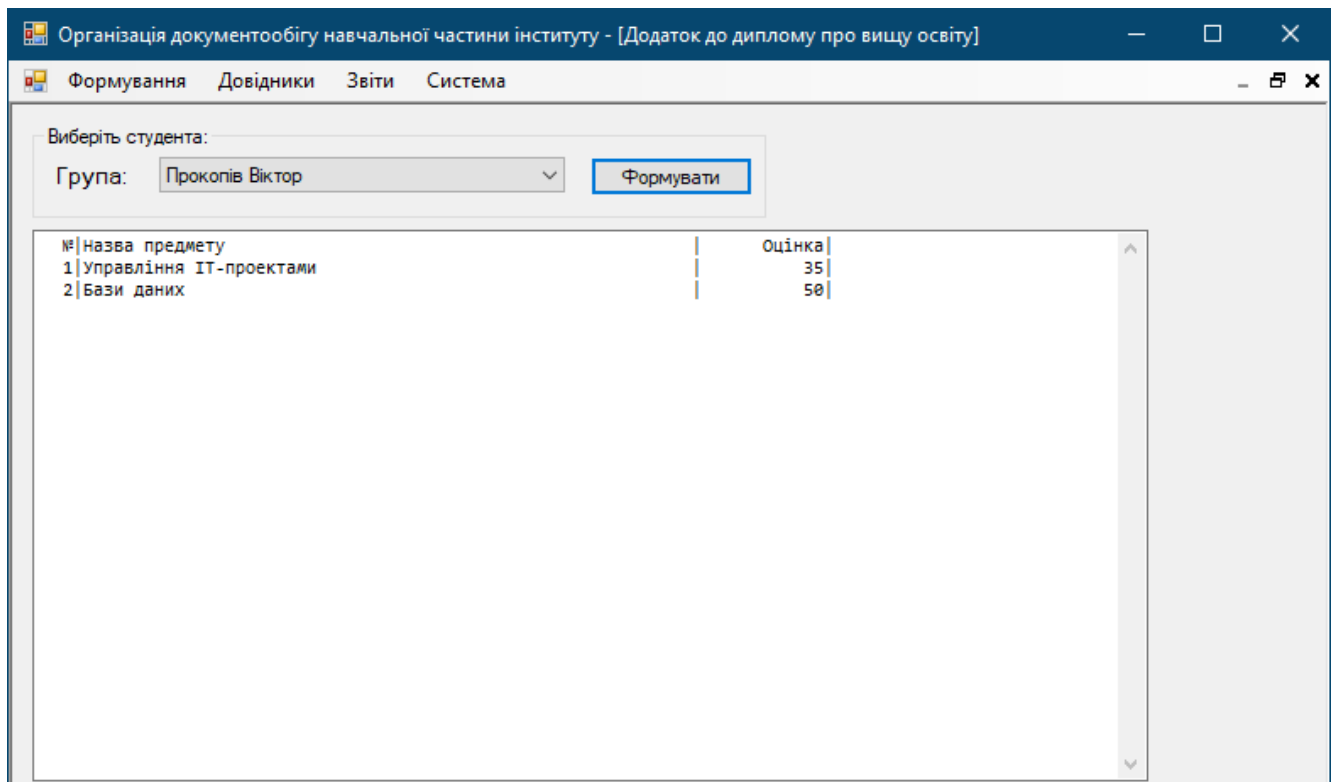


Рисунок 3.16 – Додаток до диплому про вищу освіту

Також в системі можна бачити активність користувачів, та те, що вони робили в системі. Це можливо зробити за допомогою облікового запису, який має права системного адміністратора. Для цього перейдемо по меню «Система» → «Системний журнал». В цьому вікні виводяться всі події системи (3.17).

№	Користувач	Подія	Дата
1	maximag	Користувач ввійшов в систему	14.05.2022 19:11
2	maximag	Користувач вийшов із системи	14.05.2022 16:53
3	maximag	Користувач додав відомість із номером 408	14.05.2022 16:52
4	maximag	Користувач ввійшов в систему	14.05.2022 16:52
5	maximag	Користувач додав відомість із номером 407	14.05.2022 16:51
6	maximag	Користувач ввійшов в систему	14.05.2022 16:51
7	maximag	Користувач вийшов із системи	14.05.2022 16:50
8	maximag	Користувач ввійшов в систему	14.05.2022 16:49
9	maximag	Користувач ввійшов в систему	14.05.2022 16:49
10	maximag	Користувач вийшов із системи	14.05.2022 16:49
11	maximag	Користувач ввійшов в систему	14.05.2022 16:48
12	maximag	Користувач ввійшов в систему	14.05.2022 16:48
13	maximag	Користувач вийшов із системи	14.05.2022 16:46
14	maximag	Користувач ввійшов в систему	14.05.2022 16:46
15	maximag	Користувач ввійшов в систему	14.05.2022 16:45
16	maximag	Користувач ввійшов в систему	14.05.2022 16:43
17	maximag	Користувач вийшов із системи	14.05.2022 7:29
18	maximag	Користувач ввійшов в систему	14.05.2022 7:28
19	maximag	Користувач вийшов із системи	14.05.2022 7:24
20	maximag	Користувач ввійшов в систему	14.05.2022 7:23
21	maximag	Користувач вийшов із системи	14.05.2022 7:21
22	maximag	Користувач ввійшов в систему	14.05.2022 7:21
23	maximag	Користувач вийшов із системи	09.05.2022 21:08
24	maximag	Користувач ввійшов в систему	09.05.2022 21:08
25	maximag	Користувач вийшов із системи	09.05.2022 20:52

Рисунок 3.17 – Вікно «Системний журнал»

Треба сказати, що дана система є не сильно функціональною, але вона є досить простою в користуванні. Її інтерфейс є інтуїтивно зрозумілим для користувача.

Тестування програми успішно проведено та не було виявлено жодних помилок системи.

3.3 Інструкція користувачеві програми

Розроблена програма відкриває широкі можливості для організації документообігу навчальної частини інституту мовою.

На початку необхідно запустити додаток «Організація документообігу навчальної частини інституту».

Для формування відомості про успішність студентів необхідно наповнити інформаційну систему, а саме: інформацію про студентів, дисципліни та групи.

Програма дозволяє також редагувати та видаляти будь-які дані системи. Всі проведені операції фіксуються у базі даних та записуються як події системи.

Для того, щоб здійснити формування відомості необхідно перейти по меню «Формування» - > «Відомості», після чого виведеться дочірнє вікно, в якому можна створити нову відомість про успішність навчання студентів. Далі, щоб сформувати відомість необхідно: ввести № відомості; вибрати групу та назву дисципліни; та вказати дату заповнення. Після цього натиснути кнопку «Формувати». Програма сформує список студентів для вибраної групи та дисципліни, після чого користувач зможе вводити дані успішності студентів. Закінчивши введення даних, необхідно натиснути кнопку «Зберегти».

Однією із важливих функцій програми є можливість створення звітів про успішність студентів. У розробленому програмному забезпеченні можна формувати таку звітність:

- Зведену відомість за групу (студент , оцінка);

- Рейтинг студентів спеціальності по курсам (Група , ПБ, Середній бал за 100 шкалою);
- Стипендію (Група , ПБ, оцінки по 5 бальній шкалі, середній бал);
- Виписку за результатами навчання за семестр (ПБ, навчальна дисципліна, оцінка, № відомості) для навчальної картки;
- Додаток до диплому про вищу освіту (Назва предмету, оцінка).

3.4 Висновок

В ході виконання роботи мовою C# в середовищі Visual Studio 2019 реалізовано систему розподілу гуманітарної допомоги.

Реалізовано такі функціональні вимоги:

- можливість додавати, редагувати та видаляти облікові записи користувачів;
- можливість додавати, редагувати та видаляти дані про студентів;
- можливість додавати, редагувати та видаляти дані про дисципліни;
- можливість додавати, редагувати та видаляти дані про групи;
- можливість формування відомості;
- фіксування активності користувачів системи. Ведення логів;
- можливість формування звітності.

4 ЕКОНОМІЧНА ЧАСТИНА

В даному розділі дипломної роботи проводиться економічне обґрунтування доцільності розробки програмного забезпечення. Зокрема розраховується комплексний показник якості проектного рішення, який показує його переваги в порівнянні з аналогами. А також на основі показника якості та ціни споживання проектного рішення та його аналога визначається коефіцієнт конкурентноздатності, який показує спроможність даного проектного рішення конкурувати з аналогами.

Програмне забезпечення призначене для організації документообігу навчальної частини інституту.

4.1. Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів (К) включають:

$$K = K_1 + K_2 \quad (4.1)$$

де K_1 - витрати на розробку програмних засобів, грн.

K_2 - витрати на відлагодження і дослідну експлуатацію програми рішення задачі на ПК, грн.

- витрати на розробку програмних засобів включають:

- ✓ - витрати на оплату праці розробників;
- ✓ - витрати на відрахування у спеціальні державні фонди (Вф,);
- ✓ - витрати на куповані вироби (Кв);

- витрати на придбання спецобладнання для експериментальних робіт (Об);

- ✓ - накладні витрати (Н);
- ✓ - інші витрати (Ів).

Витрати на оплату праці розробників проекту визначаються за формулою:

$$Z = \sum_{i=1}^N \sum_{j=1}^M n_{ij} t_{ij} C_{ij} \quad (4.2)$$

де n_{ij} - чисельність розробників і-ої спеціальності j-го тарифного розряду, які приймають участь в проектуванні, чол.;

t_{ij} - час, який затрачений на розробку проекту співробітника і-ої спеціальності j-го тарифного розряду, днів;

C_{ij} - денна заробітна плата і-ої спеціальності j-го тарифного розряду, грн.;

$$C_{ij} = \frac{C_{ij}^0 (1+h)}{p} \quad (4.3)$$

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

h - коефіцієнт, що визначає розмір додаткової заробітної плати;

p - середня кількість робочих днів у місяці (21 день).

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

№	Посада виконавців	Місячний оклад, грн.	Середньоденна ставка, грн./дні
1	Доцент	8000	380,95
2	Консультант з економіки	8000	380,95
3	Консультант з охорони праці	8000	380,95
4	Студент	1600	76,19

Таблиця 4.2. Розрахунок витрат на оплату праці

№	Спеціальність розробника	Час розробки, дні	Денна заробітна	Витрати на розробку, грн.
1	Доцент	5	380,95	1904,75
2	Консультант з економіки	1	380,95	380,95
3	Консультант з охорони праці	1	380,95	380,95
4	Студент	65	76,19	4952,35
	Разом			7619,00

Величину відрахувань у спеціальні державні фонди визначають у процентному співвідношенні від суми основної та додаткової заробітної плати. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 22 % від суми заробітної плати:

$$Вф = 22,0: 100 * 3 \quad (4.4)$$

$$Вф = 0,22 * 7619,0 = 1676,18 \text{ грн.}$$

Таблиця 4.3 – Розрахунок витрат на куповані вироби

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна за одиницю	Кількість купованих	Сума, грн.
1	Папір (формат А4)	500 листів	85,00	1	85,00
3	Зошит	Шт.	16,70	1	16,70
6	Диск (CD-RW)	Шт.	16,15	2	32,30
Всього	134,00				

При розробці даного програмного забезпечення спеціальне обладнання не використовувалось, тому витрати на спеціальне обладнання відсутні.

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими процентами до витрат на оплату праці:

$$H = \frac{30}{100} Z \quad (4.5)$$

$$H = 0,3 * 7619,0 = 2285,7 \text{ (грн.)}$$

Інші витрати відображають видатки, які не враховані в інших статтях витрат. Вони розраховуються за встановленими процентами до витрат на оплату праці:

$$I_B = 10:100 * Z \quad (4.6)$$

$$I_B = 0,1 * 7619,0 = 761,9 \text{ (грн.)}$$

Витрати на розробку програмного забезпечення розраховуються за формулою:

$$K_1 = Z + Вф + K_B + Об + H + I_B \quad (4.7)$$

$$K_1 = 7619,0 + 1676,18 + 134,00 + 2285,7 + 761,9 = 12476,78 \text{ грн.}$$

Витрати на відлагодження і дослідну експлуатацію програмного забезпечення визначаються за формулою:

$$K_2 = S_{Mг} * t_{Від} \quad (4.8)$$

де $S_{Mг}$ - вартість однієї машино-години роботи конкретного типу ПК, грн./год.;

$t_{Від}$ - машинний час, витрачений на відлагодження і дослідну експлуатацію програмних засобів, год.

Загальна кількість днів роботи на ПК рівна 60 днів. Середній щоденний час роботи на ПК - 2 год., тому:

$$t_{Від} = 60 * 2 = 120 \text{ (год.)}. \text{ Для ПК типу IBM PC/AT } S_{Mг} = 20,0 \text{ (грн.)}$$

Отже:

$$K_2 = 20,0 * 120 = 2400 \text{ (грн.)}$$

Таблиця 4.4 – Кошторис витрат на розробку програмного забезпечення

№	Найменування елементів витрат	Сума витрат, грн.
1	Витрати на оплату праці	7619,00
2	Відрахування у спеціальні державні фонди	1676,18
3	Витрати на куповані вироби	134,00
4	Накладні витрати	2285,70
5	Інші витрати	761,90
6	Витрати на відлагодження і дослідну експлуатацію програмного забезпечення	2400,00
	Всього	14876,78

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість машино-годин роботи ПК (за час дії програми):

$$E_{\text{п}} = E_{1\text{п}} + E_{2\text{п}} \quad (4.9)$$

де $E_{\text{п}}$ - одноразові експлуатаційні витрати на проектне рішення (аналог), грн.;

$E_{1\Pi}$ - вартість підготовки даних для експлуатації проектного рішення (аналогу), грн.;

$E_{2\Pi}$ - вартість машино-годин роботи ПК для виконання проектного рішення (аналогу), грн.

Річні експлуатаційні витрати $B_{\text{еп}}$ визначаються за формулою:

$$B_{\text{еп}} = E_{\Pi} * N_{\Pi} \quad (4.10)$$

де N_{Π} - періодичність експлуатації проектного рішення (аналогу), раз/рік.

Вартість підготовки даних для роботи на ПК визначається за формулою:

$$E_{\Pi\Pi} = \sum_{l=1}^L n_l t_l c_l \quad (4.11)$$

де l - номери категорій персоналу, який приймає участь у підготовці даних ($l=1,2,\dots,L$);

n_l , - чисельність співробітників l -ої категорії, чол.;

t_l , - трудомісткість роботи співробітників l -ої категорії по підготовці даних, год.;

c_l — середнього динна ставка співробітника l -ої категорії з врахуванням додаткової заробітної плати та відрахувань у спеціальні державні фонди, грн./год.

$$c_l = \frac{c_l^0 (1+b)}{m} \quad (4.12)$$

де c_l^0 - основна місячна заробітна плата працівника l -ої категорії, грн.;

b - коефіцієнт, який враховує додаткову заробітну плату і відрахування у спеціальні державні фонди;

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: $c^o = 10000$ грн.

Тоді:

$$c_1 = 10000 (1+0,57) / 21*8 = 93,5 \text{ (грн./год)}$$

Трудомісткість працівника по підготовці даних для проектного рішення складає 1 год., для аналога 1,5 год.

Таблиця 4.7 – Розрахунок витрат на підготовку даних та реалізацію проектного рішення на ПК

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати , грн.
Проектне рішення			
1	1	93,5	93,5
Аналог			
1	1,5	93,5	140,3

Витрати на експлуатацію ПК визначається за формулою:

$$E_{2П} = t * S_{МГ} \quad (4.13)$$

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

$S_{МГ}$ - вартість однієї машино-години роботи ПК, грн./год.

$$E_{2П} = 1 * 20,0 = 20 \text{ (грн.)}$$

$$E_{2a} = 1,5 * 20,0 = 30,0 \text{ (грн.)}$$

$$E_{П} = 93,5 + 20,0 = 113,5 \text{ (грн.)}$$

$$E_a = 140,3 + 30,0 = 170,3 \text{ (грн.)}$$

$$B_{еп} = 113,5 * 252 = 28602,0 \text{ (грн.)}$$

$$B_{са} = 170,3 * 252 = 42915,6 \text{ (грн.)}$$

4.3 Розрахунок ціни споживання проектного рішення

Ціна споживання - це витрати на придбання і експлуатацію проектного рішення за весь строк його служби:

$$Ц_{сп} = Ц_{п} + B_{епrv} \quad (4.14)$$

де $Ц_{п}$ - ціна придбання проектного рішення, грн.:

$$Ц_{п} = K \left(1 + \frac{Pr}{100}\right) + K_0 + K_k \quad (4.15)$$

де Pr - норматив рентабельності;

K_0 - витрати на прив'язку та освоєння проектного рішення на конкретному об'єкті, грн.;

K_k - витрати на доукомплектування технічних засобів на об'єкті, грн.;

$$Ц_{п} = 14876,78 * (1 + 0,3) = 19339,8 \text{ грн.}$$

$B_{епrv}$ - теперішня вартість витрат на експлуатацію проектного рішення (за весь час його експлуатації), грн.:

$$B_{епrv} = \sum_{t=0}^T \frac{B_{еп}}{(1+R)^t} \quad (4.16)$$

де $B_{еп}$ - річні експлуатаційні витрати, грн.;

T - строк служби проектного рішення, років;

R - річна ставка дисконтування (приймається у розмірі 12%).

$$B_{епrv} = \sum_{t=1}^5 \frac{28602,0}{(1+0,12)^t} = 28602,0 * 3,605 = 103110,21 \text{ (грн)}$$

$$B_{епрв} = \sum_{t=1}^5 \frac{42915,6}{(1+0,12)^t} = 42915,6 \times 3,605 = 154710,73 \text{ (грн.)}$$

$$\Pi_{сп} = 14876,78 + 103110,21 = 117986,99 \text{ (грн.)}$$

$$\Pi_{са} = 6000,00 + 154710,73 = 160710,73 \text{ (грн.)}$$

4.4 Визначення показників економічної ефективності

Економічний ефект в сфері проектування рішення:

$$E_{пр} = \Pi_{а} - \Pi_{п} \quad (4.17)$$

$$E_{пр} = 6000,00 - 14876,78 = -8876,78 \text{ (грн.)}$$

Річний економічний ефект в сфері експлуатації:

$$E_{кв} = B_{са} - B_{сп}$$

$$E_{кв} = 42915,6 - 28602,0 = 14313,6 \text{ (грн.)}$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{екв} = \sum_{t=1}^T E_{екв} (1+R)^{T-t}$$

$$\Delta E_{екв} = \sum_{t=1}^5 14313,6 (1+0,12)^{5-t} = 14313,6 \times 6,3528 = 90931,44 \text{ (грн.)}$$

Сумарний ефект складає:

$$E = E_{пр} + \Delta E_{екв} = -8876,78 + 90931,44 = 82054,66 \text{ (грн.)}$$

Таблиця 4.8 – Показники економічної ефективності проектного рішення

№	Найменування		Значення показників
---	--------------	--	---------------------

		Одиниці вимірювання	Базовий варіант	Новий варіант
1	Капітальні вкладення	грн.	-	19339,8
2	Ціна придбання	грн.	6000,00	14876,78
3	Річні експлуатаційні витрати	грн.	42915,60	28602,00
4	Ціна споживання	грн.	160710,73	117986,99
5	Економічний ефект в сфері проектування	грн.	-	- 8876,78
6	Економічний ефект в сфері експлуатації	грн.	-	14313,6
7	Додатковий ефект в сфері експлуатації	грн.	-	90931,44
8	Сумарний ефект	грн.	82054,66	

4.5 Висновки

В даному розділі проведено розрахунок витрат на розробку проектного рішення. Здійснено порівняння з існуючим аналогом, і цим показано, що дане проектне рішення має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, гнучкість, зручність. Згідно проведеного економічного обґрунтування дане проектне рішення є конкурентоздатним. Крім того, отримано додатній економічний ефект у розмірі 82054,66 грн. і тому розробка і впровадження цього проектного рішення є економічно доцільними.

ВИСНОВКИ

У ході виконання роботи було вивчено процес організації документообігу навчальної частини інституту. Були розглянуті можливі проблеми та помилки при організації документообігу навчальної частини інституту, причини виникнення та способи їхнього відстеження.

Дану програму можна використовувати для організації документообігу навчальної частини інституту, що значно полегшить роботу.

У першій частині роботи було проведено поточного стану інформаційних систем в області документообігу навчальної частини інституту. Було зроблено опис технологій обробки інформації. Зроблено SWOT-аналіз існуючої інформаційної системи. На основі отриманих даних було проведено постановку задачі на створення інформаційної системи.

У другій частині роботи розглянуто дослідження системи та описано архітектуру системи розподілу гуманітарної допомоги. Зроблено аналіз вимог до програмного забезпечення та побудовано use-case діаграми основних прецедентів.

У третій частині проведено тестування розробленого продукту та аналіз отриманих результатів. Також приведений загальний опис програми розробленої системи. Результати тестування були успішними, помилок не було виявлено. Закінчується розділ інструкцією користувачеві та висновками, що вдалося розробити в даній системі.

В результаті проведеної роботи вирішено актуальне науково-технічне завдання для організації документообігу навчальної частини інституту. У процесі вирішення завдання розроблено інженерну методику автоматизованої процедури проведення організації документообігу навчальної частини інституту, і таким чином поставленої мети досягнуто. У ході досліджень отримано такі основні наукові та практичні результати:

1. Розроблено гнучку систему, призначену для організації документообігу навчальної частини інституту.

2. Дана система поки що ніде не використовується. Розроблені методи організації документообігу навчальної частини інституту можуть бути використані для широкого класу завдань, тому можливий подальший розвиток розробленого програмного забезпечення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

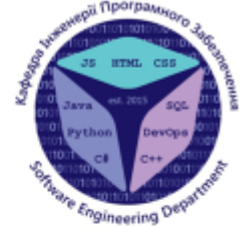
1. Java Servlet [Электронный ресурс] // Wikipedia, the free. – Электрон. дан. – Режим доступа: http://en.wikipedia.org/wiki/Java_Servlet (дата обращения: 15.05.2012). – Загл. с экрана. Архитектура программного обеспечения [Электронный ресурс] // Режим доступа:
2. https://ru.wikipedia.org/wiki/Архитектура_программного_обеспечения_Microsoft_.NET_Remoting [Электронный ресурс]
3. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя = The Unified Modeling Language user guide. — 2-е изд. — М., СПб.: ДМК Пресс, Питер, 2004. — 432 с.
4. Джозеф Шмуллер. Освой самостоятельно UML 2 за 24 часа. Практическое руководство = Sams Teach Yourself UML in 24 Hours, Complete Starter Kit. — М.: Вильямс, 2005. — 416 с.
5. Документація на програмне забезпечення [Електронний ресурс] // Режим доступу:
https://ru.wikipedia.org/wiki/Документація_на_программное_обеспечение
6. Ильязова М.Д. Модель выпускника вуза в рамках компетентностного подхода к целям и результатам ВПО [Текст] // Педагогические науки № 3 2006 г.
7. Литвинов О.А., Карпенко Н.В. Тестування інформаційних систем: модульне, інтеграційне, системне [Текст] – Д.: Ліра, 2016. – 283 с.
8. Литвинов О.А., Герасимов В.В., Карпенко Н.В. Об'єктно-орієнтована розробка інформаційних систем [Текст] – Д.: Ліра, 2018. – 448 с.
9. Мащенко Т. П. Рейтинговая система оценивания знаний учащихся в современном образовательном пространстве [Текст] // XII Всероссийский интернет-педсовет. – Электрон. дан. – Режим доступа :<http://12.pedsovet.org/> (дата обращения: 15.05.2012). – Загл. с экрана.
10. Мелецинек Адольф. Инженерная педагогика [Текст]. - М.: МАДИ(ТУ), 1998. - 185 с. 6. О классном журнале замолвите слово... [Электронный ресурс]. – Электрон. дан. – Режим доступа <http://upr.1september.ru/2005/09/4.htm>

11. Саяпин А. В. О реализации объектной библиотеки для компьютерного тестирования как средства определения уровня знаний студента [Текст] / Международный научно-технический журнал «Информационные технологии моделирования и управления» (ISSN 1813-9744) No 2(61)2010.
12. Саяпин А.В. Использование сетевых технологий для проведения тестирования студентов [Текст] / Актуальные проблемы экономики, информатики и права: Сб. науч. статей X межвузовской научно-практической конференции «Актуальные проблемы экономики, информатики и права». – Красноярск: Красноярский филиал МЭСИ, 2010.
13. Саяпин А.В. Оценка объективной меры сложности компьютерного теста методами имитационного моделирования [Текст]. - Международная научно-практическая конференция «Инновационная интегрированная система профессионального образования. Проблемы и пути развития-2011» (статья) – Красноярск: СибГАУ, 2011.
14. Сидоренко Е.В. Методы математической обработки в психологии [Текст]. – СПб.: ООО «Речь», 2003.
15. Сравнение ASP.NET MVC [Электронный ресурс] . – Электрон. дан. – Режим доступа <http://www.aspnet.com.ua/Blog/sravnenie-Asp-Net-mvc.aspx/203> (дата обращения: 15.05.2012). – Загл. с экрана.
16. Физическая энциклопедия. В 5-ти томах [Текст]. — М.: Советская энциклопедия. Главный редактор А. М. Прохоров. 1988.
17. Чельшкова М.Б. Теория и практика конструирования педагогических тестов [Текст] /- М., 2001
18. Створення архітектури програми [Електронний ресурс] // Режим доступу: <https://habr.com/ru/post/276593/>

ДОДАТОК А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОРГАНІЗАЦІЇ
ДОКУМЕНТООБИГУ НАУКОВОЇ ЧАСТИНИ ІНСТИТУТУ»

Виконав студент 4 курсу
групи ПД-41
Зверінський Андрій Юрійович
Керівник роботи
Професор, доктор технічних наук
Бондарчук Андрій Петрович

Київ-2022

2

Ключові проблеми та актуальність автоматизації предметної галузі

1. Висока ціна користування;
2. Широкий функціонал, який в більшості не буде використовуватися, але є обов'язковим додатком, що підіймає ціну;
3. Високі технічні вимоги, що не завжди можуть собі дозволити деякі комп'ютери, для комфортної роботи;
4. Перевантажений інтерфейс, який на перших етапах може відштовхнути користувачів;

Мета, об'єкт та предмет дослідження

Мета роботи – розробка програмного забезпечення для спрощення організації документообігу інституту

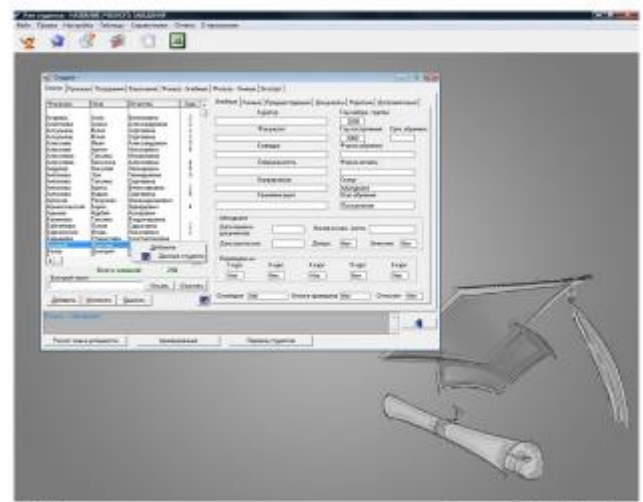
Об'єкт дослідження – створення, обробка та формування звітностей успішності студентів навчальних закладів

Предмет дослідження – автоматизація організації документообігу інституту

Аналіз існуючих програмних продуктів



1С. Університет



Навчальний відділ

Аналіз існуючих програмних продуктів

Характеристики	ІС. Університет	Деланат	Навчальний відділ	Розроблене ПЗ
Зручний інтерфейс	-	+	-	+
Формування звітів успішності	+	-	+	+
Доступність для українських ВНЗ	-	+	-	+
Невисокі системні вимоги	-	+	+	+
Можливість безшовного користування	-	-	-	+
Оцінка	1 (+)	3 (+)	2 (+)	5 (+)

Технічні завдання

На основі отриманої інформації можемо визначити, що програма має виконувати такі функції:

1. додавати, редагувати та видаляти облікові записи користувачів;
2. додавати, редагувати та видаляти дані про студентів;
3. додавати, редагувати та видаляти дані про дисципліни;
4. додавати, редагувати та видаляти дані про групи;
5. формування відомості;
6. фіксування активності користувачів системи. Ведення логів;
7. формування звітності.

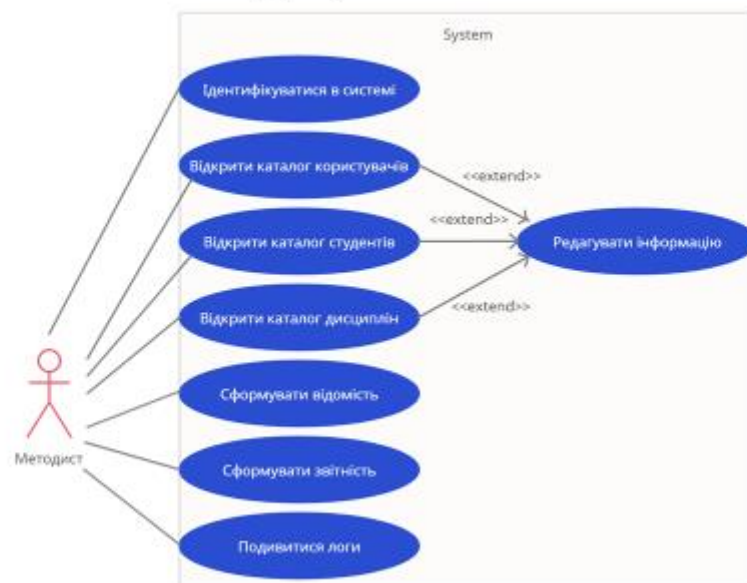
При цьому ПЗ не має навантажувати систему, тому при розробці будемо орієнтуватися на такі системні вимоги:

Процесор Pentium I V/Xeon 2.4 ГГц., Оперативна пам'ять: 1024 Мб і вище.,
Вільний дисковий простір не менше 120 Мб.,

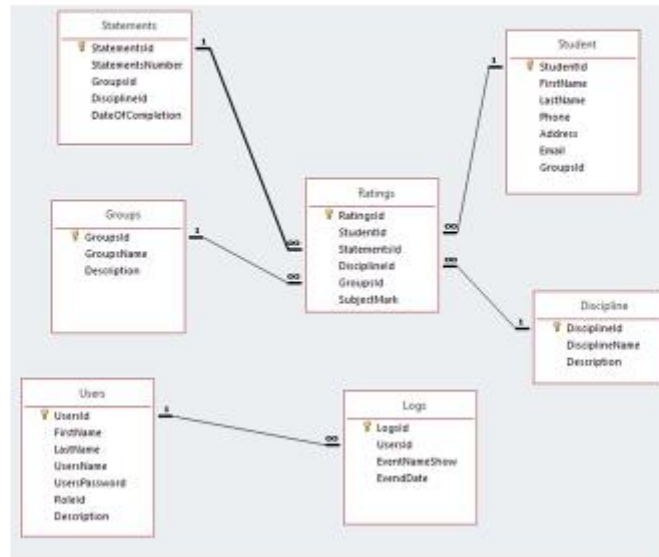
ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



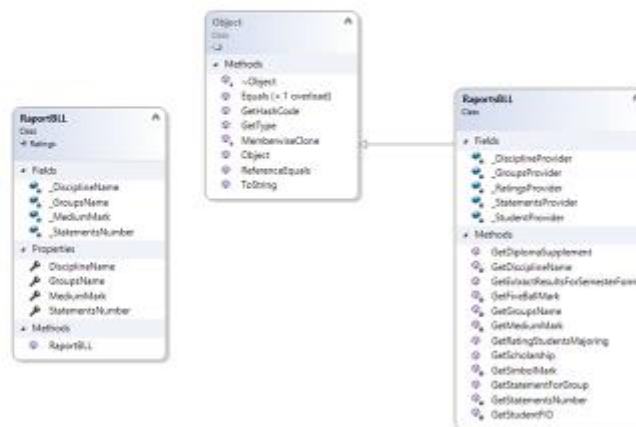
Діаграма use-case



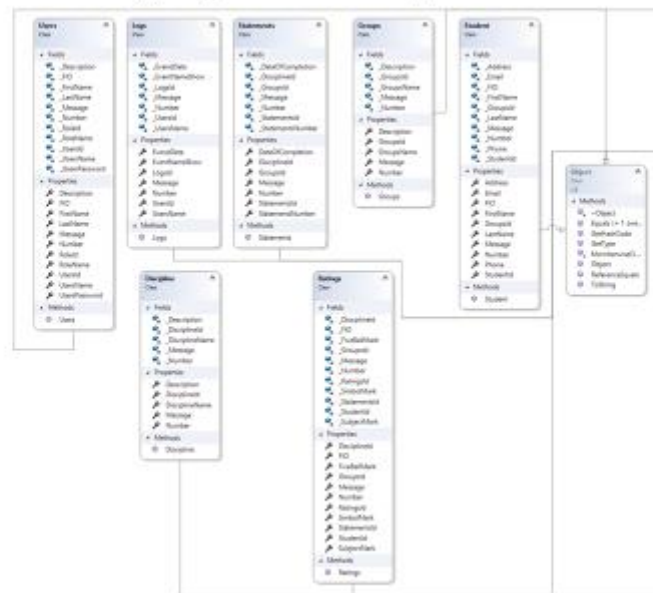
ERD діаграма бази даних



Діаграма класів бізнес логіки

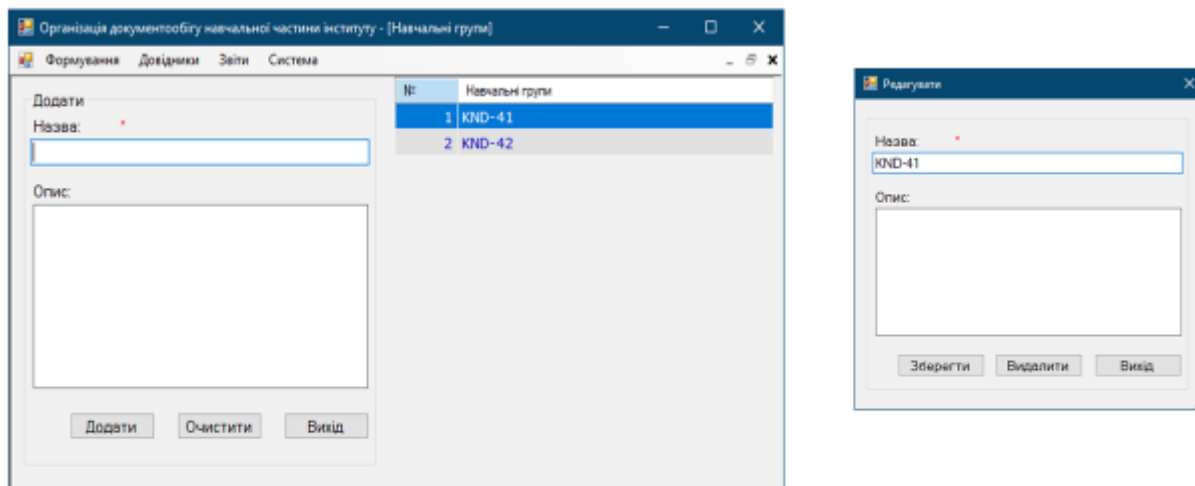


Діаграма класів домена

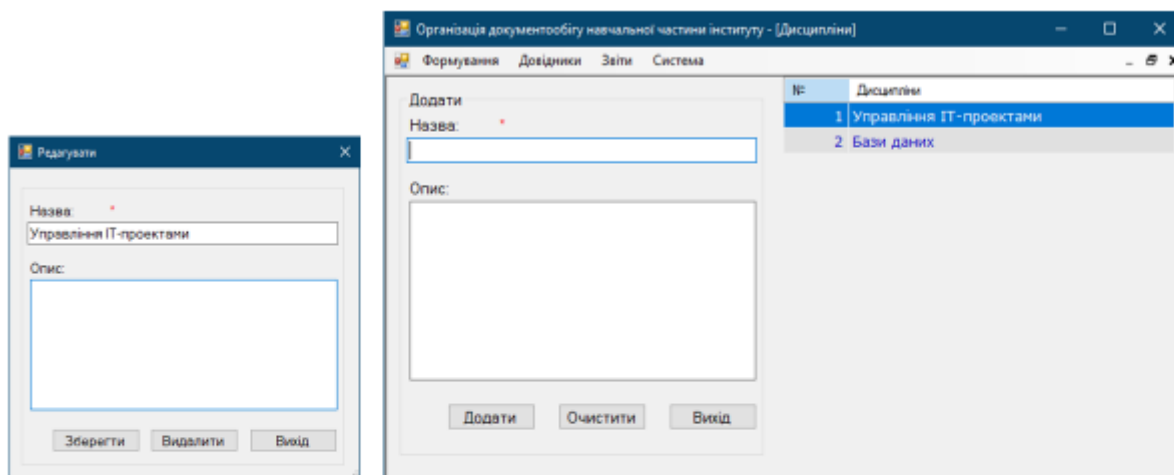


Можливості розробленої програми:

1. додавати, редагувати та видаляти облікові записи користувачів;
2. додавати, редагувати та видаляти дані про студентів;
3. додавати, редагувати та видаляти дані про дисципліни;
4. додавати, редагувати та видаляти дані про групи;
5. формування відомості;
6. фіксування активності користувачів системи. Ведення логів;
7. формування звітності.



Введення даних про групи



Додавання нових дисциплін

Організація документообігу навчальної частини інституту - [Студенти]

Формування Друк Дія Звіт Система

Додати

Прізвище:

Ім'я:

№ тел.:

E-mail:

Адреса:

Група: KND-41

Додати Очистити Вийти

№	Прізвище	Ім'я	№ телефону
1	Прокопа	Віктор	+380450335588
2	Максєйчук	Іван	+380660887711
3	Фікерт	Мирон	+380679502055
4	Фороєнч	Назар	+380673333205
5	Борезюндраій	Василь	+380676661121

Результати

Прізвище:

Ім'я:

№ тел.:

E-mail:

Адреса:

Група: KND-41

Зберегти Виділити Вийти

Додавання інформації про студентів

Організація документообігу навчальної частини інституту - [Підписи]

Формування Друк Дія Звіт Система

Додати

Нової відомості

№ відомості:

Група: KND-41

Новою дисципліною: Управління IT-проєктами

Дата заповнення: 14.05.2022

Формувати

№ відомості	Дата заповнення	Відіслати
435	14.05.2022 14:47	ВІДАВІТИ
436	14.05.2022 14:49	ВІДАВІТИ
407	10.05.2022 16:51	ВІДАВІТИ

П.І.Б.	Оцінка	Буквенна	Б-типальна
Прокопа Віктор	100	A	Відмінно
Максєйчук Іван	80	Fx	Низькоє.
Фікерт Мирон	80	C	Добре.

Зберегти Очистити Вийти

Формування нової відомості

№ студент	Оцінка	Буквена	5-ти бальна
1 прокопів Віктор	42	Fx	незадов.
2 маковічук Іван	78	C	добре.
3 сіхерт Марко	85	B	добре.

Формування зведеної відомості за групу

№ студент	Оцінка	Буквена	5-ти бальна
1 прокопів Віктор	42	Fx	незадов.
2 маковічук Іван	75	C	добре.
3 сіхерт Марко	85	B	добре.

Формування рейтингу студентів спеціальності по курсам

Організація документообігу навчальної частини інституту - [Завдання відомість за групу]

Формування Довідники Звіт Система

Вибірть групу:

Група: KND-41

№Студент	Оцінка	Буквена	5-ти Балова
1)Прокопів Віктор	42	F	Незадов.
2)Наковілюк Іван	75	C	Добре.
3)Коворт Мерон	85	B	Добре.

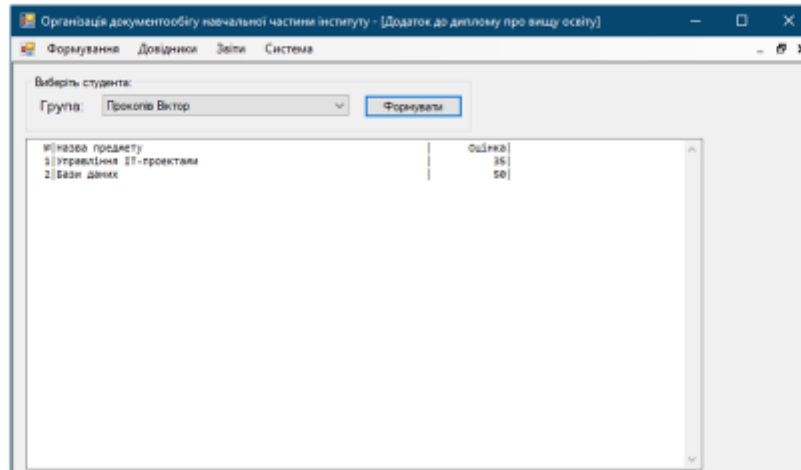
Нарахування стипендії

Організація документообігу навчальної частини інституту - [Виписка за результатами навчання за семестр]

Формування Довідники Звіт Система

№П.І.Ф.	Навчальна дисципліна	Оцінка	№ відомості
1)Прокопів Віктор	Управління IT-проектами	25	485
2)Наковілюк Іван	Управління IT-проектами	90	405
3)Коворт Мерон	Управління IT-проектами	98	485
4)Прокопів Віктор	Базе даних	50	406
5)Наковілюк Іван	Базе даних	68	486
6)Коворт Мерон	Базе даних	91	406
7)Сорокин Матер	Управління IT-проектами	100	487
8)Веремичкаій Василь	Управління IT-проектами	80	487

Виписка за результатами навчання за семестр



Додаток до диплому про вищу освіту

ВИСНОВКИ

ПЗ розроблене в результаті цієї бакалаврської роботи, здатне автоматизувати процеси документообігу інституту та оптимізувати роботу працівників.

Програма має ряд переваг над іншими програмами-аналогами, як наприклад: зручний інтерфейс та низькі системні вимоги.

За допомогою розробленого програмного забезпечення автоматизовані:

1. ведення бази даних студентів;
2. ведення бази даних дисциплін;
3. ведення бази даних груп навчання;
4. формування навчальної відомості;
5. формування звітів діяльності навчальної частини інституту.

Перспективи подальших досліджень:

Розширення функціоналу, шляхом додавання функцій керування групами, потоками та студентами.

Дякую за увагу!