

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

**ДОПУЩЕНО ДО ЗАХИСТУ**

Завідувач кафедри інженерії програмного  
забезпечення, канд. техн. наук, доцент,  
\_\_\_\_\_ Є. О. Давиденко  
«\_\_\_»\_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ  
РЕЙТИНГОВИХ БАЛІВ В ЗАКЛАДАХ ВИЩОЇ ОСВІТИ**

Спеціальність «Інженерія програмного забезпечення»

121 – КРМ.1 – 608м.21810824

**Здобувач** \_\_\_\_\_ Р. В. Скрипнік  
*підпис*  
«\_\_\_»\_\_\_\_\_ 2024 р.

**Керівник** канд. техн. наук, доцент кафедри інженерії програмного забезпечення  
\_\_\_\_\_ Г. В. Горбань  
*підпис*  
«\_\_\_»\_\_\_\_\_ 2024 р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії програмного  
забезпечення, канд.техн.наук, доцент,

\_\_\_\_\_ Є. О. Давиденко

«\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**на виконання кваліфікаційної роботи магістра**

Видано студенту групи 608м факультету комп'ютерних наук

\_\_\_\_\_ Скрипніку Роману Володимировичу \_\_\_\_\_  
*(прізвище, ім'я, по батькові студента)*

1. Тема кваліфікаційної роботи

«Інформаційна система моніторингу рейтингових балів в закладах вищої освіти»

Затверджена наказом по ЧНУ від «10» листопада 2023 р. № 234

2. Строк представлення кваліфікаційної роботи «\_\_» \_\_\_\_\_ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Вхідні дані до роботи – функціональні та нефункціональні вимоги до інформаційної системи моніторингу рейтингових балів в закладах вищої освіти. Результат – функціонуючий застосунок

4. Перелік питань, що підлягають розробці:

- аналіз предметної області та існуючих аналогів;
- специфікація функціональних та нефункціональних вимог;
- проектування та моделювання системи;
- алгоритмізація та розробка математичної моделі;

– реалізація, тестування та відлагодження застосунка.

5. Перелік графічних матеріалів:

Презентація\_\_\_\_\_.

Керівник роботи \_\_\_\_\_ канд. техн. наук, доцент Горбань Гліб Валентинович  
(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Завдання прийнято до виконання

\_\_\_\_\_ Скрипнік Роман Володимирович  
(прізвище, ім'я, по батькові студента)

\_\_\_\_\_  
(підпис)

Дата видачі завдання «9» листопада 2023 р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

Тема: «Інформаційна система моніторингу рейтингових балів в закладах вищої освіти».

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРМ	10.11.2023	14.11.2023	Виконано
2.	Огляд літератури за темою роботи	15.11.2023	20.11.2023	Виконано
3.	Складання календарного плану КРМ	23.11.2023	24.11.2023	Виконано
4.	Аналіз предметної області	25.11.2023	08.12.2023	Виконано
5.	Розробка проектних рішень	09.12.2023	17.12.2023	Виконано
6.	Моделювання та конструювання ПЗ	18.12.2023	25.12.2023	Виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	26.12.2023	20.01.2024	Виконано
8.	Відгук керівника КРМ	29.01.2024	31.01.2024	Виконано
9.	Оформлення КРМ та презентації	02.02.2024	07.02.2024	Виконано
10.	Попередній захист	08.02.2024	08.02.2024	Виконано
11.	Рецензування	10.02.2024	14.02.2024	Виконано
12.	Завершення оформлення КРМ та презентації	15.02.2024	19.02.2024	Виконано
13.	Захист кваліфікаційної роботи	26.02.2024	27.02.2024	Виконано

Розробив студент Скрипнік Роман Володимирович  
(прізвище, ім'я, по батькові студента) \_\_\_\_\_ (підпис)

«9» листопада 2023 р.

Керівник роботи канд. техн. наук, доцент Горбань Г. В.  
(посада, прізвище, ім'я, по батькові) \_\_\_\_\_ (підпис)

«9» листопада 2023 р.

## АНОТАЦІЯ

до кваліфікаційної роботи магістра

«Інформаційна система моніторингу рейтингових балів в закладах  
вищої освіти»

Студент 608м гр.: Скрипнік Роман Володимирович

Керівник: канд. техн. наук, доцент Горбань Г. В.

Моніторинг рейтингових балів здобувачів вищої освіти дозволяє оцінити якість освітніх послуг та рівень вмотивованості при вивченні різних дисциплін. Аналіз даних про рейтингові бали дозволяє виявити проблемні аспекти у навчальних програмах та розробити відповідну стратегію для підвищення успішності та вмотивованості здобувачів вищої освіти.

Об'єкт кваліфікаційної роботи – процес оцінювання рівня знань здобувачів вищої освіти.

Предметом кваліфікаційної роботи є засоби та методи розробки інформаційної системи для моніторингу рейтингових балів.

Мета кваліфікаційної роботи – оптимізація процесу обліку даних про результати навчання за рахунок розробки інформаційної системи.

В першому розділі здійснено аналіз предметної сфери та існуючих аналогів. У другому розділі наведено результати моделювання та проєктування системи. В третьому розділі виконана алгоритмізація застосунку та розроблено математичну модель аналізу даних. В четвертому розділі охарактеризовано процес реалізації та тестування системи. Також розроблено керівництво користувача.

В результаті виконання кваліфікаційної роботи магістра реалізована інформаційна система моніторингу рейтингових балів.

КРМ викладена на 59 сторінок, вона містить 4 розділи, 35 рисунків, 8 таблиць, 22 джерела в переліку посилань

*Ключові слова: аналіз даних, проєктування баз даних, education technology, Python, MS Power BI.*

## **ABSTRACT**

of the Master's Thesis

«Grades Monitoring System in Higher Education»

Student of group 608: Skrypnyk Roman Volodymyrovych

Supervisor: Dr.Sc., Docent Horban H. V.

Students' grades can be used to assess the quality of educational services and the level of students' motivation. Data analysis allows us to detect current issues to develop better courses in the future.

The object of the work is the process of assessing the level of university students' knowledge.

The subject of this work is the implementation of different algorithms and techniques in grade monitoring system development and deployment.

The purpose of the qualification work is to develop a grade monitoring system to optimize grade data storing and processing

The first section is about the analysis of the subject area and existing solutions. In the second section the process of system modeling and design is accomplished. In the third section application's algorithm and a mathematical model of data analysis were developed. The fourth section describes the system implementation and testing process. A user manual has also been developed.

The grades' monitoring system application is a result of completing the Master's Thesis.

The work is 59 pages long and includes 4 sections, 35 illustrations, 8 tables, and 22 sources.

Keywords: *data analysis, database development, education technology, Python, Power BI.*

## ЗМІСТ

ЗМІСТ .....	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ .....	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ ...	6
1.1 Визначення академічної успішності .....	6
1.2 Підходи до оцінювання у закладах вищої освіти .....	9
1.3 Переваги та недоліки існуючих аналогів .....	11
1.4 Статистичні розрахунки у рейтингах .....	16
1.5 Специфікація вимог.....	17
Висновки до розділу 1 .....	18
2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ .....	19
2.1 Процеси у інформаційній системі.....	19
2.2 Варіанти використання системи .....	22
2.3 Діаграма компонентів.....	30
2.4 Дані в інформаційній системі.....	31
Висновки до розділу 2.....	32
3 АЛГОРИТМИ ОБРОБКИ ДАНИХ У ІНФОРМАЦІЙНІЙ СИСТЕМІ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ .....	33
3.1 Попередня обробка даних.....	33
3.2 Статистична обробка даних.....	34
3.3 Підходи до візуалізації даних .....	38
Висновки до розділу 3 .....	43
4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ .....	44
4.1 Дослідження технологій та інструментів для аналізу даних .....	44
4.2 Програмна реалізація алгоритму для збору та обробки даних .....	46
4.3 Тестування розробленого коду .....	50

4.4 Налаштування інтерактивного звіту для візуалізації результатів аналізу даних .....	51
4.5 Керівництво користувача .....	56
Висновки до розділу 4.....	59
ВИСНОВКИ .....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	61
ДОДАТОК А Текст програмної реалізації алгоритмів читання та обробки даних.....	63
ДОДАТОК Б Текст програмної реалізації unit-тестів .....	66
ДОДАТОК В Результати апробації кваліфікаційної роботи магістра.....	67



## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ**

КРМ	– кваліфікаційна робота магістра
ЄКТС	– європейська кредитно-трансферна система
ЗВО	– заклад вищої освіти
ІС	– інформаційна система
МОН	– Міністерство освіти і науки України

## ВСТУП

Застосування інформаційних технологій в сфері вищої освіти дозволило докорінно змінити підхід до здійснення навчального процесу та оцінювання досягнень здобувачів вищої освіти. Аналіз даних допомагає виявити закономірності та тренди у навчальному процесі. Деякі закономірності можуть вказувати на недостатню якість навчальних послуг або на інші проблеми, а розуміння причин цих явищ допоможе знайти ефективне рішення.

**Мета** полягає у оптимізації процесу обліку даних про результати навчання за рахунок розробки інформаційної системи.

**Об'єкт** – процес оцінювання успішності здобувачів вищої освіти.

**Предметом** є методи та засоби розробки інформаційної системи для моніторингу рейтингових балів.

Для досягнення поставленої мети необхідно виконати наступні **завдання**:

- а) проаналізувати предметну сферу;
- б) розробити модель даних;
- в) розробити модель процесів у системі;
- г) здійснити специфікацію вимог;
- г) обрати алгоритм обробки даних;
- д) реалізувати застосунок для моніторингу рейтингових балів.

Рейтингові бали є показником академічної успішності здобувачі та є підставою для важливих рішень в сфері освіти. Саме тому вдосконалення процесу обліку рейтингових балів є **актуальною** темою.

**Апробація** результатів дослідження полягала у висвітленні окремих аспектів роботи на Всеукраїнській науково-практичній конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія – 2024»:

– Скрипнік Р. В., Горбань Г. В. Метрики академічної успішності у інформаційній системі моніторингу рейтингових балів. Інформаційні технології та інженерія : тези доп. Всеукр. наук.-практ. конф. молодих вчених, аспірантів і студентів. Миколаїв, 31 січня – 2 лютого 2024 р. Чорном. нац. ун-т ім. Петра Могили, 2024. С. 137 – 138.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ

## 1.1 Визначення академічної успішності

**Закладом вищої освіти (ЗВО)** називається юридична особа, яка діє згідно з виданою ліцензією та провадить освітню, наукову, інноваційну та дослідницьку діяльність. ЗВО забезпечує організацію освітнього процесу та надає здобувачам можливість отримати вищу та післядипломну освіту відповідно до їх інтересів та здібностей. Вища освіта визначається сукупністю систематизованих знань, умінь і практичних навичок, способів мислення, професійних, світоглядних і громадянських якостей, морально-етичних цінностей, інших компетентностей, здобутих у закладі вищої освіти (науковій установі) у відповідній галузі знань за певною кваліфікацією на рівнях вищої освіти, що за складністю є вищими, ніж рівень повної загальної середньої освіти [1]. Основними структурними підрозділами закладу вищої освіти є факультети, кафедри та бібліотека. Існують наступні рівні акредитації:

I – технікум, училище та інші прирівняні до них заклади вищої освіти.

II – коледж та інші прирівняні до нього заклади вищої освіти.

III – інститут, консерваторія.

IV – інститут, консерваторія, академія, університет.

**Здобувачі** вищої освіти – це особи, які навчаються у закладі вищої освіти з метою здобуття відповідного наукового ступеня та кваліфікації. В тексті кваліфікаційної роботи терміни «здобувачі» та «студенти» використовуються як синоніми для уникнення повторень, хоча правильним та загальноприйнятим терміном є саме «здобувач».

**Академічна успішність** характеризує не лише досягнення окремого здобувача, а і загальний рівень освітніх послуг в межах певного університету, факультету та кафедри. Основою академічної успішності є відповідність стандартам вищої освіти та конкурентоспроможність на ринку праці.

Дані про досягнення у навчанні використовуються при нарахуванні стипендій та переведенні з контрактної на бюджетну форму навчання. Також

програми обміну для студентів передбачають отримання звіту від університету про успішність окремого студента. Якщо студент не має мотивації або можливості впоратися з навчальною програмою, то саме на основі балів та кількості академічних заборгованостей ухвалюється рішення про відрахування. Також при переведенні на інші спеціальності враховуються результати попереднього навчання. Отже, дані про академічну успішність мають бути об'єктивними. Процес оцінювання повинен бути прозорий. Єдина система оцінювання потрібна для того щоб різні ЗВО могли дійти згоди стосовно шкали оцінок та їх інтерпретації.

На рис. 1.1 представлена концептуальна модель метрик академічної успішності. Бали у числовому форматі є загальноприйнятою формою оцінювання, але не варто забувати і про інші метрики. Деякі з цих показників не можливо виміряти та позначити певним числом, але від цього вони не перестають бути важливим.

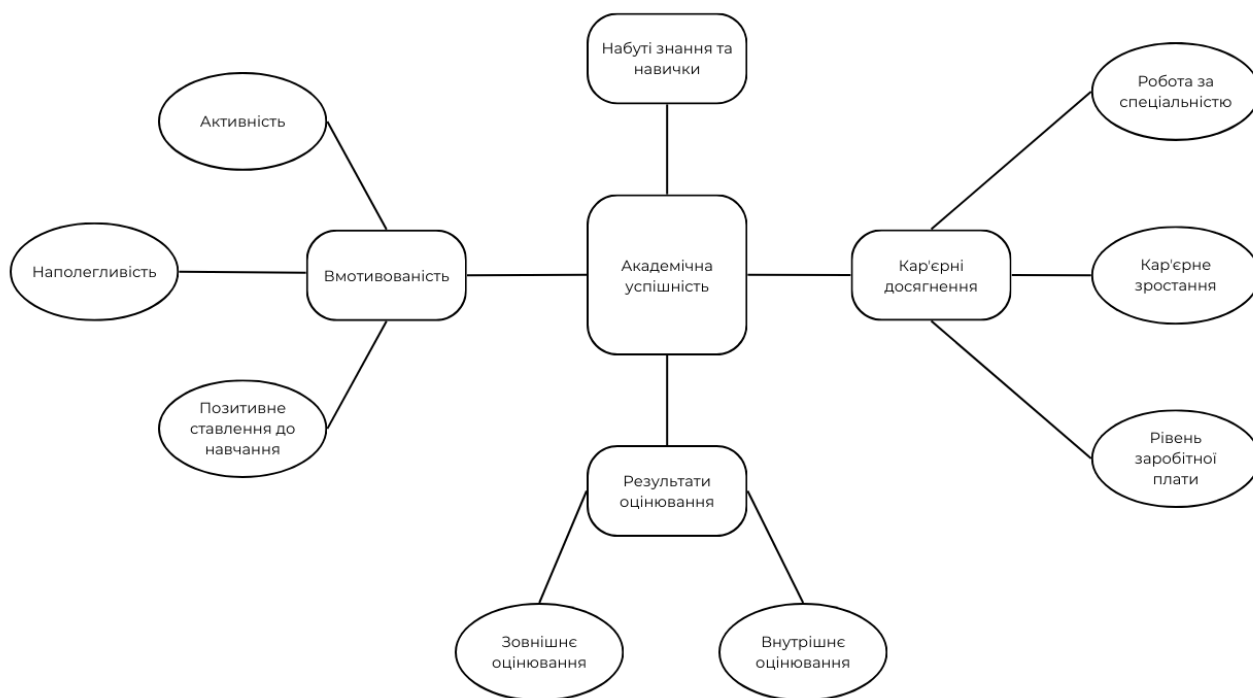


Рисунок 1.1 – Метрики оцінки академічної успішності

Множина знань та навичок є найважливішим показником академічної успішності. Від цієї метрики залежать всі інші наведені показники. Крім того,

Йдеться не лише про безпосередні результати вивчення курсів та набуті «hard skills», а і про розвинуті під час навчання універсальні якості особистості, тобто «soft skills».

Внутрішнє оцінювання передбачає контроль знань з боку університету (наприклад, рейтингові бали). Існують різні системи оцінювання. В Чорноморському національному університеті імені Петра Могили використовується система кредитів ECTS (Європейська кредитно-трансферна накопичувальна система) та відповідна шкала оцінок. Дана система допомагає узгодити принципи розробки та оцінювання курсів у різних університетах. ECTS робить освітню систему більш прозорою та забезпечує академічну мобільність [2]. Результатами зовнішнього оцінювання вважаються, наприклад, успіхи здобувачів на олімпіадах та отримання професійних сертифікатів («CompTIA A+», «Microsoft Fundamentals»), сертифікати про рівень володіння англійською мовою, тощо). Оцінка знань сторонніми організаціями важлива для підтвердження якості освіти.

Кар'єрні досягнення демонструють здатність здобувачів застосовувати знання на практиці. Також це свідчить про загальний розвиток особистості, професійну етику та високий рівень мотивації. Крім того, отримана багатьма здобувачами робота за спеціальністю підтверджує актуальність та корисність навчальної програми. Попри наявність очевидної кореляції з якістю освіти дана метрика досить нестабільна та залежить від багатьох факторів. Наприклад, криза на ринку праці через зовнішні обставини може суттєво змінити ситуацію з працевлаштуванням. Тому некоректно робити висновки про академічну успішність лише за кар'єрними досягненнями.

Вмотивованість можна оцінити за активністю здобувача, тобто наявністю додаткових балів за участь у житті факультету, олімпіадах, конкурсах, тощо. Якщо здобувачу подобається вчитися, то він вчиться краще. Також якщо навчання дається легко, то здобувач має позитивне ставлення до отримання освіти. Отже, формування позитивного враження від навчання є складним процесом та оцінюється досить суб'єктивно. Наприклад, в ЧНУ ім. Петра Могили для

визначення ставлення здобувачів до процесу отримання освіти регулярно поводяться відповідні опитування.

Одним з найважливіших показників академічного успіху є наполегливість здобувача та стабільність рівня зацікавленості навчанням. Завдяки інформаційній системі моніторингу рейтингових балів можливо оцінити зміну рейтингових балів з часом.

Розрахунки для визначення динаміки академічної успішності передбачають об'єднання різних файлів з рейтингами в одну базу даних з наступною розробкою запитів для обчислення статистичних показників. Важливо вказати роль рейтингових балів серед інших метрик успішності, а отже оцінити об'єктивність та межі застосування отриманих результатів. В будь-якому випадку саме оцінки у числовому форматі є загальноприйнятим способом оцінювання академічної успішності.

## 1.2 Підходи до оцінювання у закладах вищої освіти

Застосування ECTS передбачено Законом України «Про вищу освіту» [1] та деякими іншими законами, що стосуються освіти. В табл. 1.1 наведено відповідність між національною шкалою та ECTS.

Таблиця 1.1 – Оцінювання згідно з ECTS

Бали	Національна шкала	ЄКТС
		Оцінка
90 – 100	Відмінно	A
82 – 89	Добре	B
75 – 81		C
67 – 74	Задовільно	D
60 – 66		E
35 – 59	Незадовільно	FX
1 – 34		F

ECTS було запроваджено у 1989 році в програмі ЄС «Erasmus» для перезарахування кредитів, отриманих під час навчання за програмами обміну для здобувачів. Як було зазначено раніше, дана система дозволяє підвищити академічну мобільність. Завдяки ECTS різні університети узгодили шкалу оцінок. Також вказана система дозволяє нараховувати не лише бали, а і так звані «кредити». *Кредити* є основою для перезарахування дисциплін при переведенні з одного університету в інший чи при зміні в межах одного ЗВО.

Наприклад, в ЧНУ ім. Петра Могили рейтинговий бал студента факультету комп'ютерних наук рахується як сума зваженого середнього підсумкових балів за кожен окрему дисципліну та додаткових балів. Різна вага дозволяє врахувати складність дисципліни та її роль для підвищення рівня професіоналізму майбутнього фахівця. Важливою складовою рейтингового балу є так звані «додаткові бали». Вони дозволяють врахувати вмотивованість, активність та участь здобувача у громадському житті. Крім того, досягнення у олімпіадах, конкурсах та змаганнях теж підвищують позицію здобувача у рейтингу.

Узгодження принципів оцінювання між викладачами та здобувачами є важливою частиною навчального процесу. На початку кожного окремого курсу (дисципліни) викладач пояснює свої вимоги, критерії оцінювання та розподіл балів між різними видами діяльності. В програмі курсу обов'язково є практичні заняття, лекції та підсумковий контроль (атестація, іспит або залік). Також може бути курсовий проєкт, додаткові завдання та тести для поточного оцінювання. В сучасній вищій освіті заохочується дослідницька діяльність та самостійний пошук інформації, тому здобувачі можуть працювати над власним дослідженням (індивідуальним завданням) за обраною темою в межах певної дисципліни. Відвідування додаткових курсів (наприклад, на «Coursera», «Udemy», «EdEra» тощо) оцінюється при наданні відповідного сертифіката.

Певна варіативність у підходах до поточного оцінювання та у способах ведення звітності зберігається попри регулювання нормативними документами. Навіть в межах одного ЗВО відрізняється спосіб моніторингу рейтингових балів

для різних факультетів. Отже, необхідно обрати певний спосіб ведення документації в якості прикладу для більш детального аналізу.

*В наступних розділах роботи йдеться про запровадженій факультетом комп'ютерних наук ЧНУ ім. Петра Могили спосіб моніторингу рейтингових балів.*

### 1.3 Переваги та недоліки існуючих аналогів

Використання платформ для дистанційного навчання суттєво полегшило облік оцінок та ведення відповідної звітності. В більшості LMS (Learning Management System) є вбудований алгоритм для додавання, редагування та збереження балів. Компоненти LMS представлено на рис. 1.2.

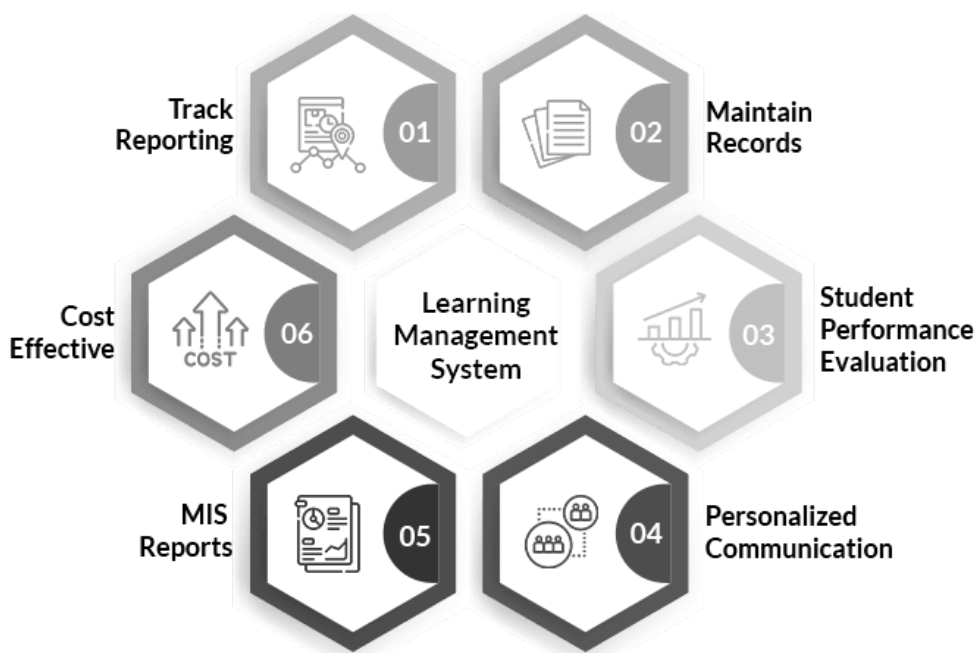


Рисунок 1.2 – Компоненти LMS [3]

Серед переваг можна назвати те що викладач може встановити обмеження для кількості балів за певний вид діяльності або створити завдання, у яких можливе автоматизоване оцінювання. Також система перевіряє введену оцінку і попереджає про помилку (наприклад, випадково надрукували 55 балів замість 5 балів). «MOODLE» надсилає сповіщення про те що певна робота перевірена та оцінена. Розподіл прав доступу дозволяє викладачам, дорадникам та



іншим зацікавленим особам переглядати оцінки окремих студентів з різних дисциплін та отримувати звіти про навчальний процес.

«MOODLE» не формує рейтинговий бал з урахуванням кредитів та додаткових балів. Отже, автоматизувати розробку рейтингів за допомогою «MOODLE» неможливо. Це суттєвий недолік. Таким чином, здобувач дізнається результати лише поточного оцінювання [4]. Складання рейтингів передбачає використання великої кількості додаткових даних про здобувача та затвердження з боку багатьох підрозділів університету. Інформація про кількість стипендій, наявність у здобувача пільг для отримання соціальної стипендії, кредити тощо перевантажила би і без того складний застосунок. Тому для моніторингу рейтингових балів необхідна окрема інформаційна підсистема.

У «Coursera» та інших подібних освітніх платформах підхід до оцінювання відрізняється від «MOODLE». Здобувач отримує інформацію про те скільки відсотків матеріалів курсу опрацьовано в кожному модулі (рис. 1.3).




Item	Status	Due	Weight	Grade
 Are You Connected to Mode Analytics? Quiz	Passed	Dec 24, 2023 11:59 PM EET	2%	100%
 Module 1 Quiz Quiz	Overdue	Dec 24, 2023 11:59 PM EET	20%	--
 Module 2 Quiz Quiz	Overdue	Dec 31, 2023 11:59 PM EET	20%	--

Рисунок 1.3 – Оцінки на «Coursera» [5]

Крім того, «Coursera» нагадує про дедлайни та необхідність виконати певні завдання якнайшвидше. Оцінки виставляються або автоматично, або в результаті так званого «peer review» (коли здобувачі перевіряють роботи одне одного). Студент переглядає лише свої оцінки, та, звісно, жодні рейтинги не складаються. Демонстрація того скільки відсотків курсу пройдено є досить корисною. Окрім мотивації та наочності ця проста інфографіка дозволяє оцінити приблизну кількість часу та зусиль для завершення курсу.



Використання електронних таблиць для формування рейтингу є стандартною практикою для факультету комп'ютерних наук ЧНУ ім. Петра Могили (рис. 1.5).

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК						5	3	3	4	3	Сер.зв жений бал за навчанн я	Σ додаткови х балів (від 0 до 10)	Рейтинг загальний
608 ГРУПА, 121 Інженерія програмного забезпечення 11 СЕМЕСТР						150	90	90	120	90			
№	№	Номер залікової к	Ф.Н.	Група	Мето ди Data	Анал із та візуа	Дідж италі зація	Осно ви наук	Сцен арне моде				
1	5	Бондаренко Стефанія Віталіївна	21810904	ДЗ	608	100	100	100	100	100	100	4,8	94,8
2	2	Андрєва Юлія Андріївна	21810802	ДЗ	608	100	97	100	100	100	99,5	2	91,55
3	9	Гінжол Аліна Іванівна	21810809	ДЗ	608	100	100	100	100	100	100	0	90
4	13	Кіяшко Максим Сергійович	21920801	ДЗ	608	90	90	100	100	95	94,72	0	85,25
5	15	Лопушанський Кирило Андрійович	21810817	ДЗ	608	98	76	97	96	90	92,39	0	83,15
6	18	Постриган Олексій Олексійович	21810823	ДЗ	608	87	82	100	98	81	89,78	0	80,8
7	11	Забелєнков Максим Дмитрович	21810814	ДЗ	608	72	75	91	91	76	80,56	0	72,5
8	21	Фінік Владислав Юрійович	21810826	ДЗ	608	68	83	92	91	72	80,28	0	72,25
К-ть звичайної						17*0,35=5,95=5							
К-ть підвищеної													
<b>СОЦ. СТИПЕНДИЯ</b>													
***	18	Постриган Олексій Олексійович	21810823	ДЗ	608	87	82	100	98	81	89,78	0	80,8
<b>БОРЖНИКИ-Бюджетники</b>													
9	4	Бечка Дмитро Романович	21810804	ДЗ	608	94	91	95	100	н			0
10	6	Бочков Максим Володимирович	21810305	ДЗ	608	70	75	н	86	71			0
11	10	Жлуктарьов Антон Андрійович	21810912	ДЗ	608	н	н	н	н	н			0
12	12	Кисса Олег Петрович	21810915	ДЗ	608	50	72	82	77	н			0
13	14	Ларченко Ігор Анаголійович	21810117	ДЗ	608	н	н	75	н	н			0
14	16	Лукіна Анастасія Вячеславівна	21920903	ДЗ	608	н	61	70	60	68			0
15	17	Парахненко Ростислав Андрійович	21810821	ДЗ	608	60	61	65	н	н			0
16	19	Рєва Ігор Андрійович	21810120	ДЗ	608	64	63	100	н	н			0
17	22	Хоменко Віктор Максимович	21810124	ДЗ	608	68	60	75	н	н			0
<b>КОНТРАКТНИКИ</b>													
18	1	Андрєв Андрій Андрійович	22250825	К	608	87	77	90	91	87	86,72	0	78,05
19	3	Барішніков Вадим Олександрович	21610801	К	608	69	68	85	88	77	77,06	0	69,35
20	7	Вальковський В'ячеслав Сергійович	21810806	К	608	67	70	85	н	н			0
21	8	Гагін Ярослав Русланович	21810808	К	608	84	67	89	н	н			0
22	20	Скрипник Роман Володимирович	21810824	К	608	73	62	80	86	68	74,39	0	66,95

Рисунок 1.5 – Приклад рейтингу

У даного підходу є наступні переваги:

1) *Доступність та сумісність.* Для створення файлу з рейтингом використовується стандартне офісне програмне забезпечення, яке або заздалегідь встановлене на більшості комп'ютерів («Excel» для Windows, «Numbers» для macOS тощо), або доступне онлайн («Google Sheets»). Отже, провідний фахівець не витрачає час на встановлення та вивчення додаткових застосунків. Також використання загальноприйнятого програмного забезпечення робить файл з рейтингом сумісним з більшістю платформ, тобто можна сподіватися на те що всі зацікавлені особи зможуть безпроблемно відкрити та переглянути вміст файлу.

2) *Просте поширення.* Можливо завантажити рейтинг на «Google Drive» та надати доступ до нього всім зацікавленим особам. Також файл можна відправити за допомогою електронної пошти та месенджерів, прикріпити посилання на нього в «MOODLE» або в чаті навчальної групи тощо.

3) *Полегшення комунікації.* Навчальний процес у ЗВО взагалі та моніторинг рейтингових балів зокрема забезпечується великою кількістю підрозділів. В ході роботи колектив співробітників університету розробив домовленість (явну або неявну) стосовно формату ведення документації, умовних позначень та інших аспектів робочого процесу. Використання звичного та традиційного підходу до ведення звітності значно полегшує комунікацію між окремими підрозділами.

У використання електронних таблиць є наступні недоліки:

1) *Неузгодженість.* При перегляді файлів з рейтингами за різні роки можна побачити як змінювався підхід до ведення документації та впорядкування даних.

2) *Недостатня автоматизація.* При формуванні рейтингу велика кількість розрахунків, виправлень та інших маніпуляцій з даними все ще виконується вручну. Це зайве навантаження на провідного фахівця.

3) *Обмежені можливості для пошуку, фільтрації та статистичних розрахунків.* Наприклад, без використання додаткових джерел інформації важко зрозуміти в якому файлі шукати дані про навчання певного студента, оскільки неможливо здійснити пошуку за прізвищем по всьому масиву даних.

4) *Відсутність візуалізації.* Переглядаючи звіт по сесії легко помітити що якщо представити наведену інформацію у візуальному форматі (наприклад, у вигляді кругової діаграми), то вона стала би більш простою для сприйняття.

5) *Відсутність зв'язку.* Враховуючи те що дані про кожну окрему сесію зберігаються в окремому файлі важко прослідкувати динаміку академічної успішності здобувачів за всі роки навчання. Також проблематичним завданням є формування статистичного звіту про навчання на факультеті за різні роки або за весь час існування певної спеціальності. Така статистика була би не лише цікавою, а і корисною для оцінки якості освіти.

Враховуючи всі наведені переваги та недоліки варто розробити рішення, яке поєднає в собі доступність та простоту з широким спектром можливостей для фільтрації, пошуку та обробки даних. Дані про рейтингові бали є основою для створення статистичного звіту про діяльність факультету. Розроблена статистика може бути використана як для діагностики, так і для прогнозування академічної успішності. Варто реалізувати автоматичну валідацію введених даних та доступний інтерфейс, до якого всі зацікавлені особи можуть отримати доступ онлайн.

#### 1.4 Статистичні розрахунки у рейтингах

Розрахунок рейтингового балу здійснюється за формулою 1.1. У даній формулі передбачене врахування кількості кредитів, оцінки з дисципліни та додаткових балів, які студент отримав за семестр.

$$R = \frac{\sum_{k=1}^m G * c}{\sum_{k=1}^m c} + \sum_{k=1}^n A \quad (1.1)$$

де:  $R$  – рейтинговий бал  
 $G$  – оцінка за дисципліну;  
 $c$  – кількість кредитів для дисципліни;  
 $A$  – додаткові бали;  
 $m$  – кількість дисциплін;  
 $n$  – кількість додаткових занять.

Крім розрахунку рейтингу надається звіт по сесії у вигляді статистики, яка включає в себе дані про успішність, якість та розподіл оцінок різних категорій для кожної дисципліни. Успішність є відсотком здобувачів, які закрили дисципліну без перескладань. Якість визначається відсотком здобувачів, які склали предмет на оцінку «добре» та «відмінно».

Варто додатково дослідити як відрізняються рейтингові бали здобувачів різних форм навчання, як змінюється успішність в ході навчання від першого курсу до завершення магістратури тощо. Демонстрація прогресу могла би стати гарною мотивацією для студентів. Також можливо робити узагальнену звітність про рейтингові бали студентів різних спеціальностей або про успіхи у вивченні певної дисципліни за тривалий проміжок часу.

Отже, збір та раціональна організація даних про академічну успішність надає широкий спектр можливостей для досліджень та діагностики якості навчального процесу. Розроблені інтерактивні звіти варто доповнити візуалізацією даних.

### 1.5 Специфікація вимог

На основі здійсненого аналізу предметної звели можна сформулювати більш детальну специфікацію **функціональних** вимог:

1. *Здійснення CRUD-операцій над даними про академічну успішність.* Користувач повинен мати можливість створювати, переглядати, оновлювати та видаляти записи про академічну успішність (результати іспитів, оцінки, академічні досягнення тощо).

2. *Розрахунок рейтингових балів.* Система повинна автоматично обчислювати рейтингові бали на основі наданих даних про академічну успішність. Це може включати суму балів за кожен предмет, середній бал, вагові коефіцієнти для різних типів оцінок тощо.

3. *Фільтрація даних.* Користувач повинен мати можливість фільтрувати дані про академічну успішність за різними критеріями, такими як предмет, період часу, тип оцінки тощо.

4. *Пошук даних.* Система має надавати можливість шукати конкретні дані про академічну успішність за певними параметрами або ключовими словами.

5. *Формування статистики про рейтингові бали.* Система повинна надавати можливість генерувати звіти та статистику про рейтингові бали, включаючи середні значення, медіани, діаграми та інші візуальні засоби аналізу.

Ці вимоги є базовими і можуть бути додатково розширені або уточнені в залежності від конкретних потреб користувачів та вимог до системи. Окрім функціональних вимог існують **нефункціональні** вимоги до системи:

1. *Продуктивність*. Система повинна швидко відповідати при виконанні операцій (завантаження, оновлення та видалення даних).

2. *Надійність*. Система повинна бути стійкою до помилок та відновлюватися в разі виникнення виняткових ситуацій або відмови системи. Необхідно забезпечити механізми резервного копіювання та відновлення даних.

3. *Безпека*. Доступ до конфіденційних даних про академічну успішність повинен бути обмежений та захищений механізмом авторизації. Дані повинні бути захищені від несанкціонованого доступу, модифікації або втрати.

4. *Usability*. Інтерфейс користувача повинен бути інтуїтивно зрозумілим та легким у використанні. Система повинна бути легко масштабованою та придатною до розширення для відповіді на зміни.

5. *Сумісність та портативність*. Система повинна бути сумісною з різними пристроями та платформами щоб забезпечити доступ до даних з різних пристроїв.

## Висновки до розділу 1

У першому розділі кваліфікаційної роботи магістра здійснено аналіз предметної сфери, а саме визначено академічну успішність та особливості оцінювання в закладах вищої освіти. Переваги та недоліки існуючих аналогів варто використати як основу для специфікації вимог до власної системи. Оцінки будуть представлені та інтерпретовані згідно з ECTS. В якості зразка обрано процес моніторингу рейтингових балів здобувачів факультету комп'ютерних наук Чорноморського національного університету імені Петра Могили.

Отже, маючи дані для аналізу та уточнивши розуміння поставленого завдання можливо перейти до проєктування інформаційної системи моніторингу рейтингових балів в закладі вищої освіти.

## 2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ

### 2.1 Процеси у інформаційній системі

Методологія IDEF0 використовується для моделювання послідовності процесів у системі з метою подальшої специфікації вимог та визначення підходів до реалізації продукту [7]. Відповідна діаграма для інформаційної системи моніторингу рейтингових балів представлена на рис. 2.1. Дана діаграма має найвищий рівень абстракції та зображує головний процес (призначення) ІС.

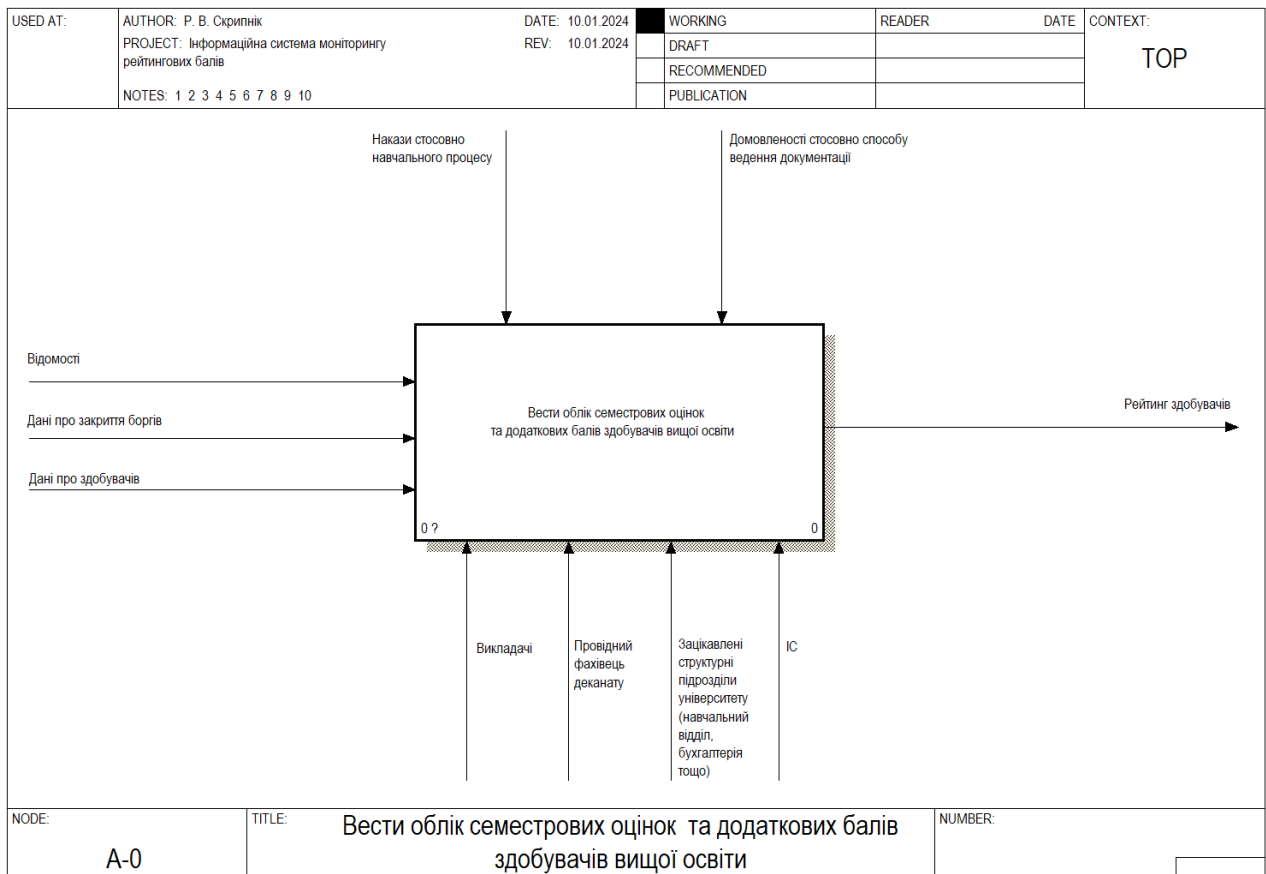


Рисунок 2.1 – Головний процес в інформаційній системі (IDEF0)

Згідно з розробленою моделлю рейтинг здобувачів вищої освіти складається за участю викладачів, провідного фахівця деканату та зацікавлених у рейтингових балах структурних підрозділів університету. Для розробки рейтингу потрібно об'єднати дані з відомостей (які надають викладачі), персональні дані здобувачів



(форма навчання, номер заліковки, право на соціальну стипендію тощо) та дані про закриття академічних заборгованостей (вже після завершення сесії).

Процес ведення обліку оцінок стосується не лише провідного фахівця та викладачів, а і інших структурних підрозділів університету. Бухгалтерія використовує дані про нарахування стипендії, навчальний відділ здійснює моніторинг успішності тощо. При розробці рейтингу варто керуватися домовленостями стосовно способу ведення документації. На рис. 2.2 представлено результат декомпозиції головного процесу інформаційної системи.

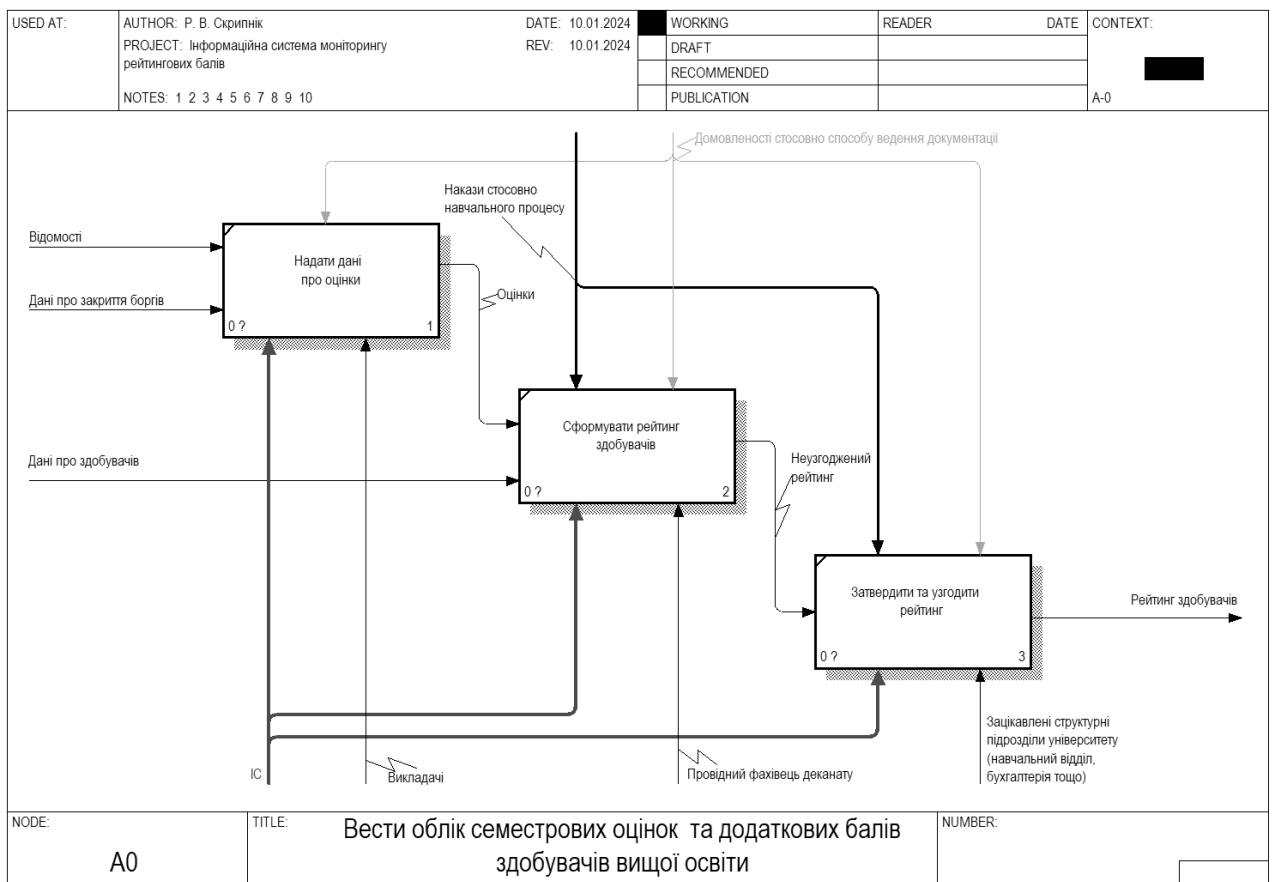


Рисунок 2.2 – Результат декомпозиції головного процесу (IDEF0)

Провідний фахівець використовує дані про форму навчання (контракт або бюджет) та наявність права на соціальну стипендію при складанні рейтингів. Накази від МОН та керівництва університету регулюють основні аспекти процесу моніторингу успішності та складання рейтингів (наприклад, визначають скільки стипендій може бути призначено та які умови мають бути виконані для отримання академічної та соціальної стипендії).

Отже, викладач надає дані про оцінки. Провідний фахівець деканату складає рейтинг. Інші структурні підрозділи затверджують та узгоджують отриманий рейтинг.

IDEF3 – це методологія моделювання, яка дозволяє задокументувати послідовність процесів в системі та визначити логічні зв'язки між ними за допомогою операторів [8]. На відміну від IDEF0 дана методологія зосереджується не на ресурсах і результатах для певного процесу, а саме на послідовності процесів та логічних зв'язках між ними, які виражені за допомогою операторів (AND, OR, XOR, Sync AND, Sync OR). В даній роботі IDEF3 використовується для декомпозиції процесу формування рейтингу здобувачів (рис. 2.3).

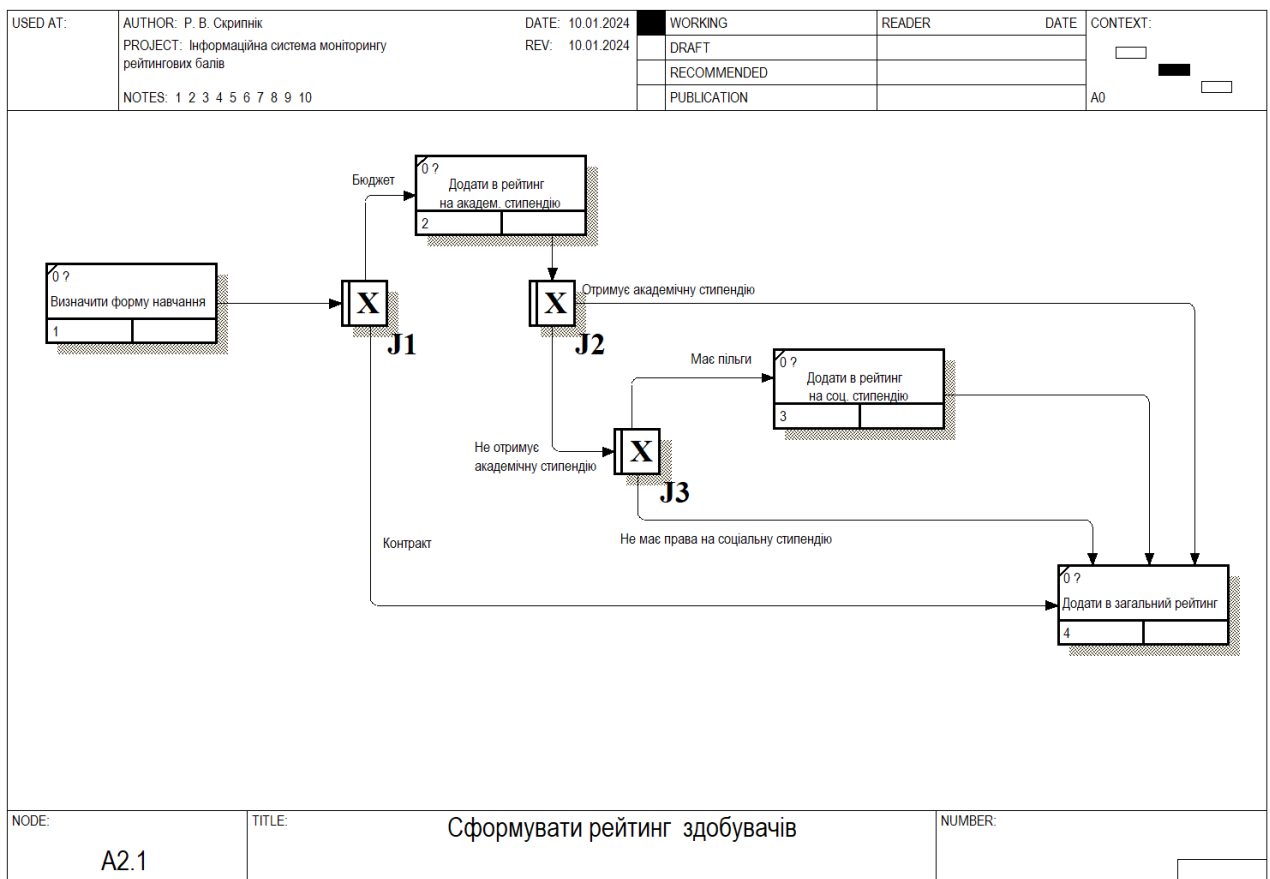


Рисунок 2.3 – Декомпозиція процесу формування рейтингу (IDEF3)

Отже, в процесі моделювання процесів вдалося поглибити розуміння вимог до інформаційної системи моніторингу рейтингових балів в закладі вищої освіти.

## 2.2 Варіанти використання системи

Мова моделювання *UML (Unified Modeling Language)* є стандартом в галузі програмного забезпечення для візуального представлення архітектури, структури та поведінки систем [9]. UML надає широкий спектр графічних символів і концепцій, які дозволяють розробникам та аналітикам ефективно моделювати складні системи. Однією з ключових переваг UML є його універсальність та стандартизованість, що робить його важливим інструментом для комунікації між різними учасниками проєкту, від розробників до клієнтів та менеджменту. Використання UML сприяє поліпшенню розуміння системи, спрощує процес розробки та забезпечує однорідність у визначенні вимог і архітектури проєкту.

Специфікація варіантів використання системи – це процес формалізованого опису різних сценаріїв можливих взаємодій користувачів з системою. Цей процес може бути виконаний наступним чином:

1. Визначення переліку акторів акторів. Спочатку необхідно визначити всіх акторів, які будуть взаємодіяти з системою. Актори – це будь-які сутності, які використовують систему або впливають на неї.

2. Визначення варіантів використання. Потім слід ідентифікувати різні сценарії або варіанти використання, які можуть виникнути під час взаємодії акторів з системою.

3. Створення діаграм варіантів використання. Після ідентифікації акторів і варіантів використання можна побудувати діаграму варіантів використання, що візуалізує взаємодію між акторами та системою.

4. Опис варіантів використання. Кожен варіант використання потрібно докладно описати, використовуючи специфікацію у певному форматі.

5. Визначення пріоритетів. На завершення, варіанти використання можуть бути відсортовані за пріоритетами або іншими критеріями для визначення того, які з них є найважливішими або найбільш критичними для реалізації.

*Діаграма варіантів використання* є потужним інструментом для опису функціональних вимог до системи з точки зору її користувачів [10]. Ця діаграма дозволяє ідентифікувати різні сценарії використання системи, ролі, які

взаємодіють з системою, та їх взаємодії. Кожен варіант використання представляє собою конкретну послідовність подій, які відбуваються між користувачами і системою для досягнення певної мети. *Use case diagrams* дозволяють стейкхолдерам взаємодіяти та уточнювати вимоги до системи на ранніх етапах розробки, що сприяє створенню більш якісних застосунків. Специфікація вимог здійснена на основі варіантів використання. Відповідна UML-діаграма представлена на рис. 2.4.

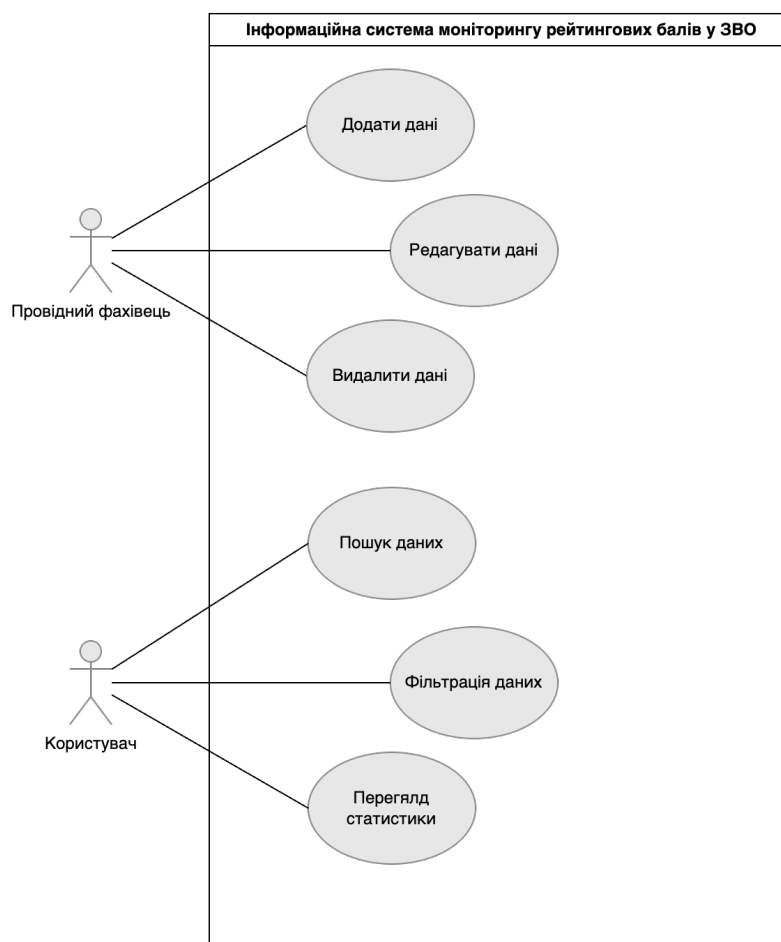


Рисунок 2.4 – Діаграма варіантів використання

1. *Провідний фахівець*. Мається на увазі провідний фахівець деканату, який збирає дані та має право вносити зміни в рейтинг.

2. *Користувач*. Викладач, студент, співробітник навчального відділу та будь-яка інша зацікавлена особа з відповідними правами доступу, якій потрібні дані про успішність здобувачів вищої освіти.

Таблиця 1.1 – Варіант використання «Пошук даних»

<b>UC-01: Пошук даних</b>	
<b>Діючі особи:</b>	Користувач, інформаційна система.
<b>Мета:</b>	Знайти довільні дані про одну з сутностей (студент, група, дисципліна тощо) в системі.
<b>Передумова:</b>	Користувач має доступ до інтерактивного звіту про рейтингові бали.
<p>Успішний сценарій:</p> <ol style="list-style-type: none"> <li>1. Користувач переходить до інтерактивного звіту про рейтингові бали.</li> <li>2. Користувач переходить до форми пошуку.</li> <li>3. Система здійснює пошук у завантажених з БД даних та їх відображення згідно до шаблону інтерактивного звіту.</li> </ol>	
<b>Результат:</b>	Користувач знайшов необхідні дані.
<b>Розширення:</b>	
<b>*a</b>	Помилка підключення до БД. Користувач отримує повідомлення про помилку.
<b>1a</b>	Шукані дані відсутні. Користувач отримує відповідне повідомлення замість пустих рядків у таблиці.

Таблиця 1.2 – Варіант використання «Фільтрація даних»

<b>UC-02: Фільтрація даних</b>	
<b>Діючі особи:</b>	Користувач, інформаційна система
<b>Мета:</b>	Обмежити кількість відображених даних згідно до заданого правила.
<b>Передумова:</b>	Користувач має доступ до інтерактивного звіту про рейтингові бали.
<p>Успішний сценарій:</p> <ol style="list-style-type: none"> <li>1. Користувач переходить до інтерактивного звіту про рейтингові бали.</li> <li>2. Користувач переходить до доступних фільтрів.</li> <li>3. Система здійснює фільтрацію завантажених з БД даних та їх відображення згідно до шаблону інтерактивного звіту.</li> </ol>	
<b>Результат:</b>	Користувач здійснив фільтрацію даних.
<b>Розширення:</b>	
<b>*a</b>	Помилка підключення до БД. Користувач отримує повідомлення про помилку.
<b>1a</b>	При здійсненні фільтрації жоден запис з БД не відповідає всім правилам фільтрації. Користувач отримує відповідне повідомлення.

Таблиця 1.3 – Варіант використання «Перегляд статистики»

<b>УС-03: Перегляд статистики</b>	
<b>Діючі особи:</b>	Користувач, інформаційна система
<b>Мета:</b>	Переглянути статистику про якість навчального процесу.
<b>Передумова:</b>	Користувач має доступ до інтерактивного звіту про рейтингові бали.
<p>Успішний сценарій:</p> <ol style="list-style-type: none"> <li>1. Користувач переходить до інтерактивного звіту про рейтингові бали.</li> <li>2. Користувач переходить до розділу зі статистикою.</li> <li>3. Користувач задає параметри фільтрації (за потреби).</li> <li>4. Система здійснює обчислення актуальної статистики.</li> </ol>	
<b>Результат:</b>	Користувач здійснив фільтрацію даних.
<b>Розширення:</b>	
<b>*a</b>	Помилка підключення до БД. Користувач отримує повідомлення про помилку.
<b>1a</b>	При розрахунку статистики виникла помилка через нестачу даних для розрахунку. Користувач отримує повідомлення про помилку.

Таблиця 1.4 – Варіант використання «Додати дані»

<b>UC-04: Додати дані</b>	
<b>Діючі особи:</b>	Провідний фахівець, інформаційна система
<b>Мета:</b>	Додати дані в БД.
<b>Передумова:</b>	Провідний фахівець має доступ до застосунку для роботи з базою даних.
<p>Успішний сценарій:</p> <ol style="list-style-type: none"> <li>1. Провідний фахівець переходить до застосунку для управління даними в БД.</li> <li>2. Провідний фахівець обирає які саме дані хоче додати.</li> <li>3. Провідний фахівець вводить дані (вручну, з файлу тощо).</li> <li>4. Система здійснює валідацію та збереження даних.</li> </ol>	
<b>Результат:</b>	Провідний фахівець додав дані в БД.
<b>Розширення:</b>	
<b>*a</b>	Помилка підключення до БД. Провідний фахівець отримує повідомлення про помилку.
<b>1a</b>	Нові дані порушують обмеження цілісності БД. Провідний фахівець отримує повідомлення про помилку.



Таблиця 1.5 – Варіант використання «Редагувати дані»

<b>UC-05: Редагувати дані</b>	
<b>Діючі особи:</b>	Провідний фахівець, інформаційна система
<b>Мета:</b>	Відредагувати дані в БД.
<b>Передумова:</b>	Провідний фахівець має доступ до застосунку для роботи з базою даних.
<p>Успішний сценарій:</p> <ol style="list-style-type: none"> <li>1. Провідний фахівець переходить до застосунку для управління даними в БД.</li> <li>2. Провідний фахівець обирає які саме дані хоче відредагувати.</li> <li>3. Провідний фахівець вводить дані (вручну, з файлу тощо).</li> <li>4. Система здійснює валідацію та збереження даних.</li> </ol>	
<b>Результат:</b>	Провідний фахівець відредагував дані в БД.
<b>Розширення:</b>	
<b>*a</b>	Помилка підключення до БД. Провідний фахівець отримує повідомлення про помилку.
<b>1a</b>	Нові дані порушують обмеження цілісності БД. Провідний фахівець отримує повідомлення про помилку. В БД залишилася стара версія даних.

Таблиця 1.6 – Варіант використання «Видалити дані»

<b>UC-06: Видалити дані</b>	
<b>Діючі особи:</b>	Провідний фахівець, інформаційна система
<b>Мета:</b>	Видалити дані з БД.
<b>Передумова:</b>	Провідний фахівець має доступ до застосунку для роботи з базою даних.
<p>Успішний сценарій:</p> <ol style="list-style-type: none"> <li>1. Провідний фахівець переходить до застосунку для управління даними в БД.</li> <li>2. Провідний фахівець обирає які саме дані хоче видалити.</li> <li>3. Система здійснює видалення обраного запису та всіх пов'язаних даних.</li> </ol>	
<b>Результат:</b>	Провідний фахівець додав дані в БД.
<b>Розширення:</b>	
<b>*a</b>	Помилка підключення до БД. Провідний фахівець отримує повідомлення про помилку.
<b>1a</b>	Видалення обраного запису призведе до каскадного видалення пов'язаних даних. Провідний фахівець отримує повідомлення з попередженням про видалення великої кількості взаємопов'язаних записів та повторний запит про підтвердження операції.

## 2.3 Діаграма компонентів

Діаграма компонентів – це важливий інструмент моделювання архітектури програмного забезпечення, який дозволяє візуалізувати структуру системи та її компоненти, а також залежності між ними [11]. На цій діаграмі компоненти системи представлені у вигляді прямокутників або боксів, які відображають фізичні або логічні модулі програмного забезпечення. Взаємодії між компонентами показуються за допомогою стрілок або ліній, що вказують на напрямок зв'язку. Діаграма компонентів дозволяє розробникам та архітекторам системи краще розуміти її структуру, розділити систему на модулі для подальшого розроблення та тестування, а також виявити можливі точки розширення або покращення. Вона є важливим інструментом для проектування та управління складними програмними проєктами, дозволяючи забезпечити ефективність, масштабованість та стабільність системи.

Компоненти можуть надавати та використовувати інтерфейс інших компонентів. Наприклад, база даних надає інтерфейс, який використовується як для здійснення CRUD-операцій над даними системи, так і для інтерактивної візуалізації. На рис. 2.5 представлено діаграму компонентів інформаційної системи моніторингу рейтингових балів у закладі вищої освіти.

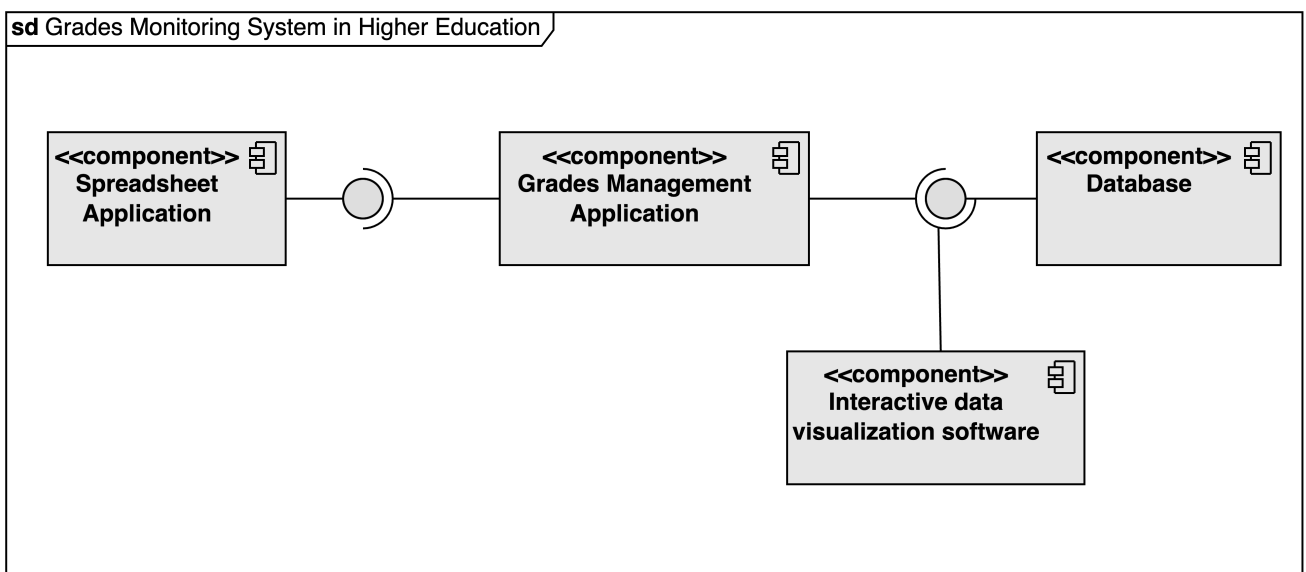


Рисунок 2.5 – Діаграма компонентів

Код для управління даними використовує провідний фахівець деканату для ведення обліку даних (Grades Management Application).

Сховище даних (Database) містить дані про рейтингові бали, дисципліни, здобувачів, навчальні групи тощо.

Інтерактивна візуалізація (Interactive data visualization software) є саме тим інтерфейсом, з яким взаємодіє користувач для перегляду рейтингів, пошуку та фільтрації даних. Також завдяки вказаному компоненту можна розраховувати статистичні показники, робити візуалізацію даних та аналізувати рейтингові бали здобувачів відповідно до поставленого завдання.

## **2.4 Дані в інформаційній системі**

Діаграма сутність-зв'язок (ERD) є ключовим інструментом у моделюванні баз даних, де вона використовується для візуалізації та організації взаємозв'язків між сутностями. У цій діаграмі сутності відображаються у вигляді прямокутників, а зв'язки між ними – у вигляді ліній з маркерами, які вказують на тип взаємодії. ERD допомагає розробникам та аналітикам даних зрозуміти структуру і взаємозв'язки між різними сутностями в системі. Вона дозволяє визначити ключові сутності, їх атрибути та взаємозв'язки, що допомагає у розробці ефективної бази даних.

Крім того, ERD може використовуватися для виявлення потреб у додаткових таблицях чи зв'язках, а також для визначення нормалізації даних та оптимізації структури бази даних. Таким чином, ERD є невід'ємною частиною процесу розробки баз даних і допомагає забезпечити консистентність, ефективність та правильність даних в системі.

Сутностями у системі є викладач, курс, оцінка, програма (тобто план навчання), студент, додаткові бали та група. Дані про дисципліну відокремлені від даних про кількість кредитів, викладача та семестр оскільки одну і ту саму дисципліну можуть викладати різним групам протягом різних семестрів різні викладачі. Оцінка міститься у зв'язувальній таблиці, тобто дана таблиця містить

складений первинний ключ з зовнішніх ключів інших таблиць (таблиця Student та таблиця Curriculum).

На рис. 2.6 представлено діаграму «сутність-зв'язок» (ERD) для інформаційної системи моніторингу рейтингових балів. Діаграми даного типу використовуються для моделювання реляційних баз даних [12], отже на основі створеної ERD можна задати схему таблиць в БД.

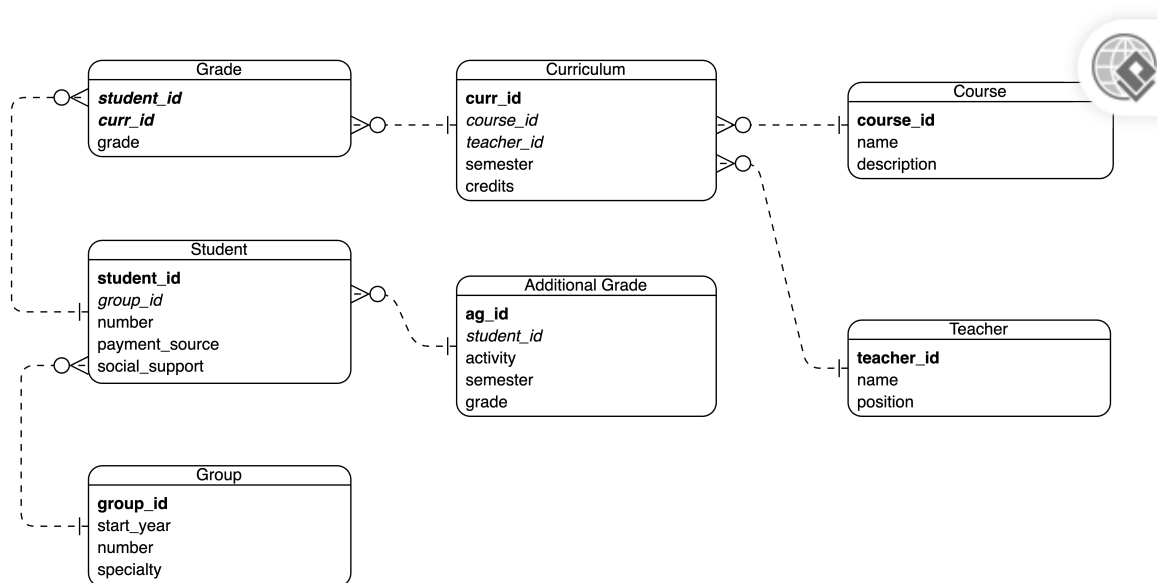


Рисунок 2.6 – Діаграма «сутність-зв'язок» (ERD)

Система керування базами даних (СКБД) – це програмне забезпечення для управління даними в межах бази даних. СКБД управляє даними, форматами даних, метаданими та структурою таблиць. Перш ніж заповнювати базу даних необхідно задати структуру таблиць згідно з розробленою моделлю.

## Висновки до розділу 2

В ході розробки другого розділу кваліфікаційної роботи здійснено проєктування системи. Для моделювання процесів використано методології IDEF0 та IDEF3. Для представлення функціональних вимог застосовано діаграму варіантів використання. Архітектура системи представлена за допомогою діаграми компонентів. Структура бази даних охарактеризована з використанням ERD.

## 3 АЛГОРИТМИ ОБРОБКИ ДАНИХ У ІНФОРМАЦІЙНІЙ СИСТЕМІ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ

При виконанні аналізу даних про рейтингові бали необхідно розраховувати статистичні показники та створити візуалізацію даних для наочного представлення отриманих результатів.

### 3.1 Попередня обробка даних

Перш ніж переходити до обробки даних варто пересвідчитися у тому що датасет не містить дублікати, пусті комірки та помилкові значення. Більшість мов програмування (Python, R тощо) містять вбудовані функції або бібліотеки для виявлення дублікатів та пропущених значень (пусті комірки, тип даних None тощо). Помилки у даних бувають наступних типів:

1) Помилка у форматі даних. Наприклад, недотримання вимог до формату номера телефона або дати. Зазвичай подібні помилки виявляються при валідації введених користувачем даних ще до запису в БД.

2) Помилка у типі даних. У наборі даних або у таблиці БД кожен атрибут має заздалегідь визначений тип. Наприклад, ціна товару має бути числом з плаваючою крапкою. Але користувач може спробувати ввести це значення разом з символом валюти. Це перетворює введені дані на рядок. Подібні проблеми вирішує валідація або використання регулярних виразів для отримання лише тим даних, які відповідають певному формату.

3) Логічна помилка. Правильне за типом та форматом значення порушує певні правила, які є характерними для предметної сфери. Такі помилки важко виявити. Наприклад, кількість кредитів ECTS для певної дисципліни може бути введена у правильному форматі та мати відповідний тип даних. Але якщо подивитися на навчальну програму взагалі, то стане зрозуміло що настільки велика кількість кредитів означає що лише на цю одну дисципліну доведеться витратити неможливу кількість годин (порушення норм стосовно навантаження як для викладачів, так і для здобувачів вищої освіти). В такій ситуації велику цінність має консультація від експерта.

4) Помилки через людський фактор. Якщо дані вводяться вручну, то можливі як помилки через неухважність, так і свідоме викривлення для досягнення певної мети. Наприклад, при друкуванні прізвища студента можливо випадково пропустити літеру або переплутати ініціали. Зазвичай подібні ситуації вирішуються за допомогою повторного перегляду введених даних або перевірки іншою людиною.

Отже, перед статистичною обробкою варто трансформувати датасет у такий спосіб щоб не залишилося помилкових значень, дублікатів та пустих комірок. Зазвичай процес попередньої обробки даних складається з наступних етапів:

1. Дослідження набору даних. Важливо перевірити джерело та якість даних. Необхідно переконатися в тому що методи збору даних та структура датасету відповідають поставленому завданню.

2. Трансформація даних. Видалення або виправлення неправильних, відсутніх або дубльованих значень. Можна застосовувати як автоматизовані, так і напівавтоматизовані методи.

3. Валідація даних. Варто визначити контрольні точки під час обробки даних, на основі яких можна легко побачити хибні елементи або неточності.

4. Документування процесу обробки даних. Детальна документація для процесу очищення та аналізу даних повинна включати в себе історію всіх внесених змін та обґрунтування прийнятих рішень.

Боротьба з помилками в даних є постійним процесом, і важливо підтримувати високу якість даних на кожному етапі їх обробки та аналізу.

### **3.2 Статистична обробка даних**

Статистична обробка даних – це процес аналізу та інтерпретації даних за допомогою статистичних методів і технік.

Статистична обробка даних може включати в себе такі процеси як сортування та фільтрація даних, розрахунок основних статистичних характеристик (наприклад, середнє значення, медіана, стандартне відхилення),

визначення взаємозв'язків між змінними, використання статистичних моделей для прогнозування або класифікації та візуалізацію результатів у вигляді діаграм.

Статистична обробка даних важлива в багатьох галузях (включаючи науку, бізнес, медицину, соціологію тощо), оскільки вона дозволяє робити обґрунтовані висновки та приймати рішення на основі даних (так звані data-driven decisions).

Важливим аспектом статистичної обробки даних є **нормалізація**. У випадку оцінок можна говорити про те що складність дисципліни та підхід до оцінювання створюють різний розподіл оцінок. Наприклад, у деяких випадках ніхто у навчальній групі не отримує більше 90-та балів, тому нормалізація дозволить масштабувати вектор оцінок. Для деяких алгоритмів аналізу даних та машинного навчання нормалізація є важливим етапом. Нормалізація значень виконується за формулою 3.1:

$$v_i = \frac{v_i}{||\bar{v}||} \quad (3.1)$$

де:  $v_i$  – елемент вектора.

$||\bar{v}||$  – норма вектора.

Для розрахунку норми вектора існують різні підходи. Наприклад, Евклідова норма розраховується за формулою 3.2.

$$||\bar{v}|| = \sqrt{\sum_{i=1}^n v_i^2} \quad (3.2)$$

Наприклад, дано вектор оцінок у системі ECTS  $\bar{v} = (85, 70, 60, 87, 75)$ . У випадку нормалізації отримуємо наступну норму вектора:

$$||\bar{v}|| = \sqrt{85^2 + 70^2 + 60^2 + 87^2 + 75^2} \approx 170$$

Після розрахунку норми вектора можна здійснити нормалізацію. Отже, вектор оцінок трансформується наступним чином:

$$\bar{v} = \left( \frac{85}{170}; \frac{70}{170}; \frac{60}{170}; \frac{87}{170}; \frac{75}{170}; \right) = (0,5; 0,41; 0,35; 0,51; 0,44)$$



Існує безліч метрик для дослідження набору даних. Якщо йдеться про програмну реалізацію, то зазвичай існують заздалегідь розроблені та вбудовані в обраний інструмент функції. Однією з найпопулярніших метрик є *середнє арифметичне*, тобто сума всіх елементів набору даних поділена на їх кількість. Вказана метрика розраховується за формулою 3.3.

$$\mu = \frac{1}{N} \sum_{i=1}^N v_i \quad (3.3)$$

де:  $v_i$  – елемент вектора.

$N$  – кількість елементів у векторі.

На основі середнього арифметичного можна розрахувати *стандартне відхилення*. Значення стандартного відхилення дозволяє оцінити варіативність вибірки. Також на основі стандартного відхилення можливо розрахувати дисперсію. Розрахунок стандартного відхилення здійснюється за формулою 3.4.

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (v_i - \mu)^2}{N}} \quad (3.4)$$

де:  $v_i$  – елемент вектора.

$N$  – кількість елементів у векторі.

$\sigma$  – середнє арифметичне вектора.

*Розмахом* називається різниця між мінімальним та максимальним значенням в наборі даних. *Медіана* – елемент який знаходиться посередині впорядкованого набору даних. Медіана не чутлива до так званих «outliers». Наприклад, медіана набору даних  $x = (60; 70; 75; 85; 87)$  – це число 75. *Квантили* розподіляють впорядкований датасет на рівні частини. Якщо йдеться про третій квантиль (Q3), то можна стверджувати що 75% елементів менші за третій квантиль.

Наприклад, при дослідженні оцінок з певної дисципліни дані про те що 65 балів є третім квантилем розшифровуються так: 75% студентів отримали оцінку нижчу за 65 балів, отже академічна успішність групи не найкраща.

**Коефіцієнт кореляції** характеризує взаємозв'язок між змінними. Значення цього коефіцієнта варіюється від 1 до -1. Якщо значення наближається до 1, то йдеться про наявність кореляції. Якщо коефіцієнт кореляції близький до -1, то має місце зворотна кореляція. Наближене до 0 значення свідчить про відсутність кореляції. Коефіцієнт кореляції розраховується за формулою 3.5.

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{(n \sum x^2 - (\sum x)^2)(n \sum y^2 - (\sum y)^2)}} \quad (3.5)$$

де:  $n$  – кількість змінних.

$x, y$  – змінні між якими обчислюється кореляція.

Набір даних містить  $x = (1; 2; 3; 4; 5; 6)$  та  $y = (2; 4; 7; 9; 12; 14)$ . Необхідно розрахувати коефіцієнт кореляції між цими змінними. У табл. 3.1 представлено попередні розрахунки.

Таблиця 3.1 – Розрахунок необхідних параметрів для коефіцієнта кореляції

$x$	$y$	$xy$	$x^2$	$y^2$
1	2	2	1	4
2	4	8	4	16
3	7	21	9	49
4	9	36	16	81
5	12	60	25	144
6	14	84	36	196
<b>21</b>	<b>48</b>	<b>211</b>	<b>91</b>	<b>490</b>

Останній рядок табл. 3.1 містить суму по стовпцях. Ці значення будуть використані у наступних розрахунках.

$$r = \frac{6 \cdot 211 - 21 \cdot 48}{\sqrt{(6 \cdot 91 - 21^2)(6 \cdot 490 - 48^2)}} = 0,998$$

Отже, за допомогою коефіцієнта кореляції можливо з'ясувати взаємозв'язки між різними змінними в наборі даних. Зазвичай розраховують коефіцієнт кореляції між всіма наявними змінними та представляють результат у вигляді матриці кореляції датасету.

### 3.3 Підходи до візуалізації даних

Візуалізація даних використовується для наочного представлення результатів аналізу даних. Для візуалізації можна застосовувати різні типи графіків та діаграм в залежності від поставленого завдання.

Використання графіків та діаграм в якості візуалізації даних є важливою складовою аналізу даних. Графіки дозволяють зображувати великі обсяги даних у зрозумілій і зручній формі, що полегшує сприйняття інформації. Вони можуть використовуватися для відображення трендів, порівняння даних, виявлення взаємозв'язків та ідентифікації аномалій. Графіки можуть мати різні форми (лінійні, кругові, стовпчикові тощо), що дає можливість вибрати найбільш підходящий тип для конкретного завдання. Крім того, вони можуть бути легко побудовані за допомогою спеціалізованих застосунків.

Зазвичай графіки та діаграми використовуються на етапі демонстрації результатів здійсненого аналізу даних для демонстрації знайдених трендів та аномалій.

**Кругова діаграма** показує співвідношення між частинами та цілим. Наприклад, співвідношення між кількістю студентів які отримують та які не отримують академічну стипендію в межах однієї групи (рис. 3.1). Вказана діаграма складається з круга, який розділений на сектори кожен з яких представляє певну категорію даних. Розмір сектора пропорційний частці або відсотку, який відповідає кількості або обсягу даних у відповідній категорії. Тому кругові діаграми часто використовуються для візуалізації розподілу даних, підкреслення відносних часток або порівняння різних категорій. Вони є ефективним інструментом для спрощення аналізу даних та передачі інформації завдяки своїй зрозумілості та лаконічності. Однак важливо враховувати, що

кругові діаграми можуть бути менш ефективними, коли вони містять багато категорій або коли розподіл даних відносно рівномірний, що може призвести до складнощів у сприйнятті інформації [13].

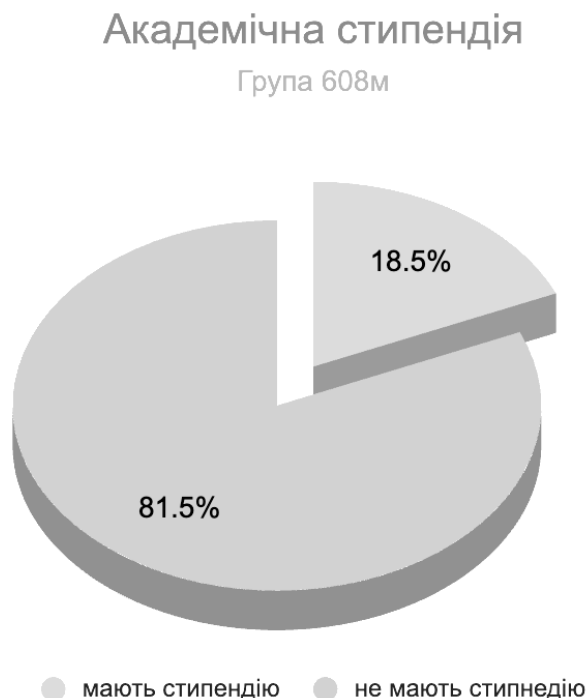


Рисунок 3.1 – Приклад кругової діаграми

**Стовпчикова діаграма** дозволяє порівняти значення того самого показника для різних категорій.

Стовпчикова діаграма (гістограма) є потужним інструментом візуалізації даних, який допомагає зрозуміти та порівняти кількісні показники. Її ефективність полягає в тому, що вона надає можливість швидко візуалізувати розподіл даних за різними категоріями або змінними. Ця діаграма складається з вертикальних стовпчиків, де висота кожного стовпчика відображає значення відповідної категорії. Вона дозволяє наглядно порівняти значення між різними категоріями або спостерігати за змінами в часі. Завдяки своїй простоті та зрозумілості, стовпчикові діаграми широко використовуються в наукових дослідженнях, бізнесі, соціальних науках та інших галузях для аналізу даних і прийняття управлінських рішень.

Гістограми допомагають візуалізувати складні концепції та зробити інформацію більш доступною та зрозумілою. Наприклад, за допомогою цієї діаграми можливо порівняти кількість студентів для кожної зі спеціальностей факультету комп'ютерних наук (рис. 3.2).

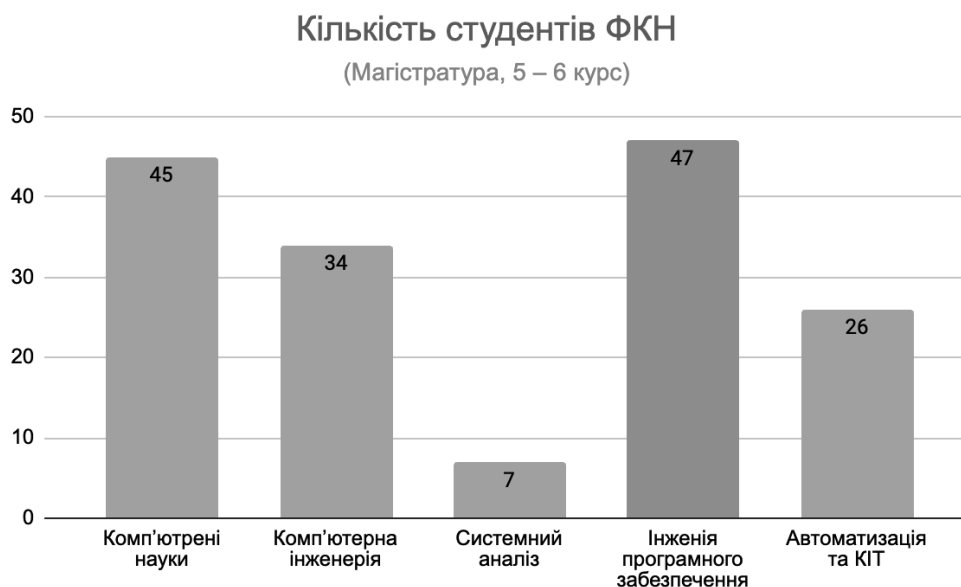


Рисунок 3.2 – Приклад стовпчикової діаграми

Графіки (лінійні діаграми) використовується для відображення залежності між двома змінними. Цей тип діаграми особливо ефективний для візуалізації тенденцій та змін в часі, таких як зміни температури, надходження або витрати коштів, кількість переглядів сайту тощо.

Лінійні діаграми дозволяють швидко виявити зв'язок між даними та визначити тренди, що допомагає в прийнятті управлінських рішень та прогнозуванні майбутніх подій. Вони також можуть бути використані для порівняння кількох наборів даних та виявлення взаємозв'язків між ними. Завдяки своїй простоті та зрозумілості лінійні діаграми залишаються одним із найпопулярніших інструментів в аналізі даних та візуалізації інформації.

Графік може вказувати на характер взаємозв'язку між змінними. За напрямком лінії можна визначити, наприклад, чи є залежність між змінними прямо пропорційною чи обернено пропорційною.

Наприклад, за допомогою графіку можна показати як змінювалася кількість абітурієнтів спеціальності «Інженерія програмного забезпечення» за період з 2020 по 2023 роки (рис.3.3). Отриманий тренд підтверджується емпіричним досвідом, оскільки події останніх років вплинули в тому числі і на освітній процес в ЗВО. В якості  $x$  виступає рік, а  $y$  – кількість заяв від абітурієнтів.



Рисунок 3.3 – Приклад графіка

**Діаграма розсіювання** (*scatter slot*) є важливим інструментом візуалізації даних, який дозволяє вивчати взаємозв'язок між декількома змінними. У цьому типі діаграми кожна точка на графіку представляє собою одне спостереження з даних і позначається відповідно до значень двох змінних на горизонтальній та вертикальній осі. Це дозволяє швидко виявити наявність або відсутність кореляції між цими змінними, а також визначити її напрямок та силу.

Діаграми розсіювання широко використовується в наукових дослідженнях, статистичному аналізі, економічних та соціальних науках, медицині та інших галузях для вивчення взаємозв'язків між різними змінними. Вона дозволяє візуалізувати велику кількість даних у простий і зрозумілий спосіб, що допомагає виявити закономірності та зробити висновки, спрямовані на подальший аналіз чи прийняття рішень.

Наприклад, можна побудувати діаграму розсіювання для дослідження зв'язку між балами за вступні іспити та рейтинговими балами за перший навчальний семестр в університеті (рис. 3.4). Можна зробити висновок про те що на основі високих оцінок на вступних виступах можна спрогнозувати академічну успішність здобувача вищої освіти.

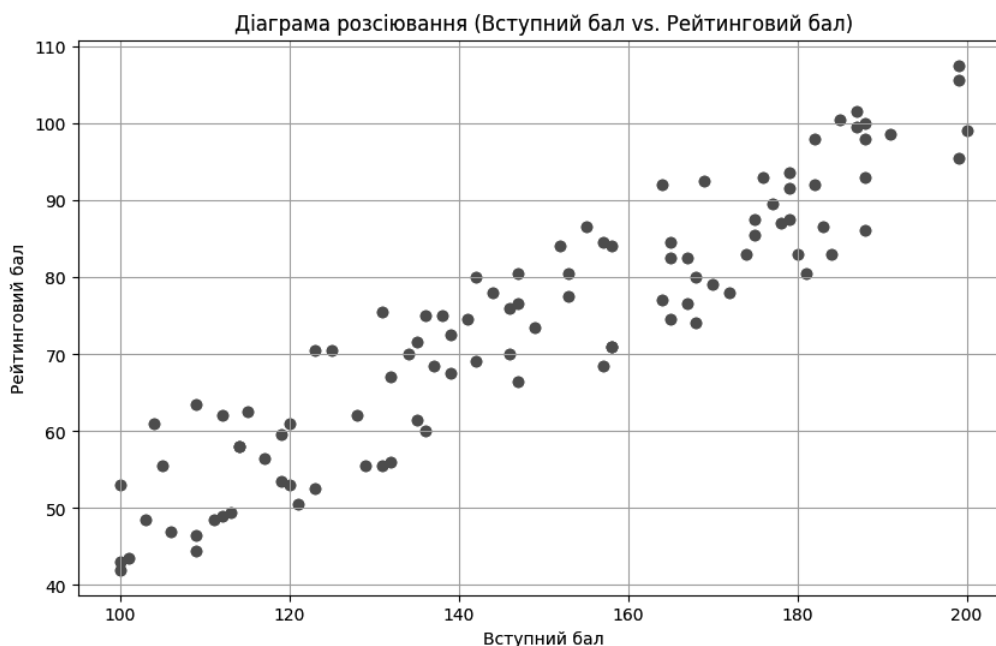


Рисунок 3.4 – Приклад діаграми розсіювання

Отже, для кожного завдання існує відповідний тип діаграми, який допоможе відобразити знайдені в ході аналізу закономірності та тренди.

Вимоги до візуалізації даних варіюються в залежності контексту та потреб користувачів. Проте існують деякі загальні принципи та критерії, які допомагають забезпечити якість та ефективність візуалізації даних. Ось деякі з них:

1. *Зрозумілість*. Візуалізація даних повинна бути зрозумілою для цільової аудиторії. Необхідно враховувати спеціалізацію та інтереси користувачів при розробці звітів. Варто акцентувати лише важливі та цікаві знахідки, а не перевантажувати всією доступною інформацією та деталями процесу аналізу даних.

2. *Інклюзивність*. Шрифти, кольори та розміри інших візуальних елементів мають бути підібрані у такий спосіб щоб люди зі слабим зором мали можливість побачити та зрозуміти зображення та текст.

3. *Ефективне використання простору.* Важливо уникати перенасиченості даними та забезпечити зручність перегляду. У дизайні існують певні правилами стосовно побудови композиції та взаємного розташування різних елементів.

4. *Доцільний вибір типу графіку.* Варто обирати виходячи з поставленого завдання. Детальний аналіз різних типів графіків та їх призначення були представлені вище.

5. *Інтерактивність.* У візуалізації можуть бути використані елементи інтерактивності, такі як вибір параметрів фільтрації даних, динамічне оновлення, пошук за ключовими словами. Це дозволяє користувачам активно взаємодіяти з даними та отримувати більш деталізовану інформацію.

6. *Адаптивність.* Є висока ймовірність того що розроблений звіт будуть переглядати з телефону. Саме тому візуалізація повинна якісно відображатися на будь-якому пристрої.

Ці вимоги допомагають створювати ефективні та інформативні візуалізації, які сприяють кращому зрозумінню предметної сфери та прийняттю обґрунтованих управлінських рішень.

### **Висновки до розділу 3**

В третьому розділі досліджено статистичні метрики, які застосовуються для опису, порівняння та дослідження датасетів. Також проаналізовано різні типи діаграм та особливості їх використання. Визначені метрики можуть бути застосовані не лише до всього набору даних. При розділенні датасету на окремі фрагменти за певною ознакою можливо порівняти академічну успішність різних категорій здобувачів вищої освіти. Діаграми та графіки мають бути частиною звіту по сесії, який використовується користувачами інформаційної системи для отримання інформації про успішність здобувачів вищої освіти.



## 4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ

### 4.1 Дослідження технологій та інструментів для аналізу даних

Аналіз даних – це процес виявлення та інтерпретації закономірностей та взаємозв'язків в даних. В табл. 4.1 наведено класифікацію інструментів для аналізу даних.

Таблиця 4.1 – Класифікація технологій для аналізу даних

Тип	Приклади	Призначення
Мови програмування	Python, R, SQL	Для роботи з датасетами великого обсягу та для застосування складних алгоритмів обробки даних та машинного навчання. Використовуються програмістами, фахівцями з аналізу даних та науковцями.
Електронні таблиці	MS Excel, Google Sheets	Для датасетів малого та середнього обсягу Широко використовуються в багатьох організаціях та установах спеціалістами різного профілю.
Інструменти для візуалізації даних	Matplotlib, Seaborn, ggplot2	Є бібліотеками певних мов програмування. Генерують окремі зображення.
Інструменти для бізнес-аналізу	Power BI, Tableau	Надають широкий спектр можливостей для аналізу даних (від трансформації та очищення до публікації звіту). Використовуються для створення інтерактивних звітів.
Алгоритми ШІ	TensorFlow, scikit-learn, PyTorch	Надають реалізацію поширених алгоритмів машинного навчання та всю необхідну документацію. Дозволяють гнучко налаштувати відповідну модель та досягти високого рівня точності.
Інструменти для обробки BigData	Hadoop, Spark, Apache Flink	Зазвичай йдеться про розподілене зберігання великих обсягів даних та можливості для ефективної обробки та аналізу цих даних.
Інструменти для аналізу природних мов	NLTK	Містять реалізації алгоритмів для обробки природних мов. Також зазвичай є бібліотеками мов програмування.

Не варто забувати про супутні інструменти, такі як *Jupyter Notebook* або *Google Colaboratory*. Також історію змін набору даних та розробку відповідних

звітів та алгоритмів відстежують та документують у системі контролю версій (наприклад, *Git*). Код мовами Python або R відлагоджують у спеціальних IDE.

Виконання аналізу даних передбачає використання великого стеку технологій, оскільки найчастіше використовують одразу декілька інструментів для виконання різних завдань.

Мови програмування у аналізі даних досить сильно відрізняються між собою. Наприклад, *Python* – це універсальна технологією, яка надає можливість створювати як вебзастосунки та ігри, так і автоматизувати задачі, виконувати аналіз даних та web scraping. В той самий час *R* є стандартною мовою програмування для наукових досліджень та візуалізації даних у наукових виданнях, але в той самий розробка застосунків на *R* не є популярним явищем. Зазвичай використання цієї мови програмування передбачає створення скриптів. *SQL* взагалі використовується лише для маніпуляцій з даними в межах бази даних. Він також вирізняється існуванням великої кількості діалектів (залежить від СКБД) та складається з багатьох компонентів (рис. 4.1). Вибір мови програмування залежить від поставленої задачі та вимог.

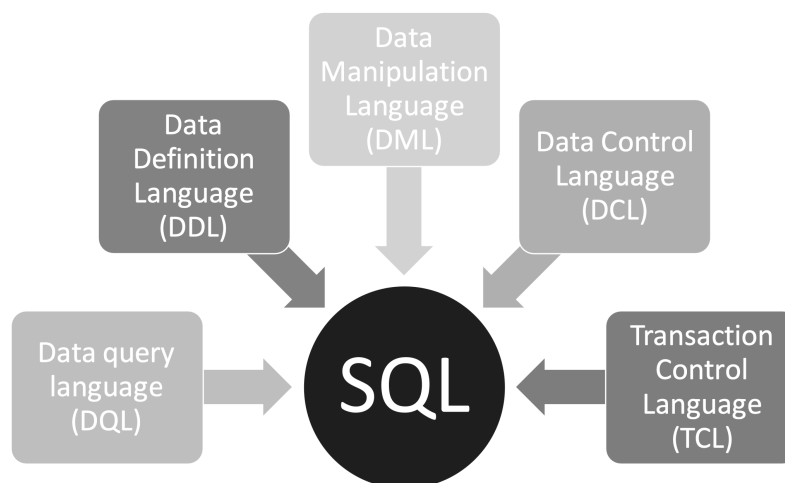


Рисунок 4.1 – Компоненти SQL

Для виконання поставленого завдання необхідно поєднати можливості мов програмування *Python* та *SQL* з інтерактивними звітами *MS Power BI*. Отримані фрагменти коду мають бути протестовані, а інтерактивний звіт налаштований таким чином щоб користувач отримав всю необхідну інформацію.

## 4.2 Програмна реалізація алгоритму для збору та обробки даних

В попередніх розділах було зазначено що ЧНУ ім. Петра Могили обрано в якості об'єкта інформатизації. За зразок взято підхід до ведення документації на факультеті комп'ютерних наук. Отже, приклад скриптів для читання датасетів має бути реалізований саме на основі даних про рейтингові бали студентів цього факультету. Рейтинги надано у форматі **.xls**.

Наступним кроком є «витягування» лише необхідних записів з цих файлів та збереження результатів в базі даних. Перед записом в базу даних варто зробити зведений .csv файл з усією необхідною інформацією для побудови інтерактивного звіту про діяльність факультету комп'ютерних наук. З часом формат відповідних .xls файлів набув узгодженості, тому розроблений код можна застосувати до всіх файлів які сформовано відповідно до шаблону. До 2014 року документація велася у дещо іншому форматі, тому до даних з того часу застосувати розроблений скрипт не вдасться. Приклад зібраних даних представлено на рис. 4.2. Текст відповідного скрипта наведено у додатку А.

	student_name	code	payment	group	subject	grade	credits	hours	type	teacher_name	additional_grade	study_year_start	study_year_end	semester	specialty	course
0	Агарков Артем Юрійович	21710101	ДЗ	101	Філософія	92	3	90	i	Гавелля В.П.	0.4	2017	2018	1	Комп'ютерні науки	1
1	Агарков Артем Юрійович	21710101	ДЗ	101	Українська мова (за професійним спрямуванням)	78	3	90	i	Понкратова В.М.	0.4	2017	2018	1	Комп'ютерні науки	1
2	Агарков Артем Юрійович	21710101	ДЗ	101	Історія та культура України.	93	6	180	i	Підберезних І.Є.	0.4	2017	2018	1	Комп'ютерні науки	1
3	Агарков Артем Юрійович	21710101	ДЗ	101	Інформаційні технології	93	3	90	з	Ніколенко С.Г. Дворецька С.В.	0.4	2017	2018	1	Комп'ютерні науки	1
4	Агарков Артем Юрійович	21710101	ДЗ	101	Вища математика	63	4	120	з	Дихта Л.М.	0.4	2017	2018	1	Комп'ютерні науки	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
359	Юрченко Дарина Сергіївна	21310325	ДЗ	571	Прикладне програмування	97	5.5	165	з	Сіделев, Беліков	5	2017	2018	2	Автоматизація та КІТ	5
360	Юрченко Дарина Сергіївна	21310325	ДЗ	571	Основи наукових досліджень	100	4	120	з	Щесюк	5	2017	2018	2	Автоматизація та КІТ	5
361	Юрченко Дарина Сергіївна	21310325	ДЗ	571	Патентознавство	95	4	120	з	Прищепов	5	2017	2018	2	Автоматизація та КІТ	5
362	Юрченко Дарина Сергіївна	21310325	ДЗ	571	Програмне забезпечення комп'ютерно-інтегровани...	90	3	90	з	Жук	5	2017	2018	2	Автоматизація та КІТ	5
363	Юрченко Дарина Сергіївна	21310325	ДЗ	571	Англійська мова (за професійним спрямуванням)	97	5	150	а	Біла	5	2017	2018	2	Автоматизація та КІТ	5

Рисунок 4.2 – Приклад даних зі сформованого датасету

На жаль, рейтинги у форматі .xls не сумісні з інструментами для бізнес-аналізу (наприклад, MS Power BI). Хоч ці інструменти і пропонують підключення до excel-таблиць, але структура рейтингів виявилася неприйнятною для завантаження та аналізу. Саме тому виникла необхідність змінити формат та структуру датасету за допомогою скрипта.

Читання та аналіз даних відбувається за допомогою бібліотеки `pandas`. Використано функцію `ExcelFile` [14], яка дозволяє завантажити всі аркуші відповідного файлу та покроково читати дані з кожного аркуша. Особливість обраних файлів полягає у представленні даних про два навчальні семестри на одному аркуші. Тому в ході читання визначено комірки, які є маркером початку та закінчення (так звані «cut points») таблиці з даними про рейтингові бали за певний семестр. Розроблений код містить наступні функції:

1. *`collect_subject_metadata`* – це функція для збору даних про дисципліни (кількість годин, прізвище викладача тощо). Дана інформація розташована в заголовку таблиці з рейтинговими балами.

2. *`collect_student_data`* – функція для збору оцінок з усіх дисциплін, які здобувач вивчав протягом навчального року. Якщо здобувач не з'явився на іспит чи залік, то оцінка вважається рівній 0.

3. *`collect_additional_grades`* – функція для збору даних про додаткові бали. Ці бали ставляться не за кожну окрему дисципліну, а за весь семестр взагалі. Тому даний факт необхідно враховувати при обчисленні суми додаткових балів.

4. *`detect_specialty`* – функція для визначення спеціальності за номером групи. Алгоритм розроблено з урахуванням того факту що у присвоєнні номерів групам є логіка. Тобто перший символ означає номер курсу, а два наступні – код спеціальності. Для старших курсів (магістратура) також додається літера «м» для позначення ступеня вищої освіти.

5. *`detect_course`* – функція для визначення номеру курсу на основі номера групи. Даний алгоритм також спирається на домовленість стосовно принципу іменування навчальних груп, яка розроблена керівництвом закладу вищої освіти.

6. *`xls_to_dataframe`* – функція, яка призначена для застосування всіх наведених вище функцій до кожного окремого аркуша в межах файлу. Файл обирає користувач. Також необхідно вказати до якого навчального року відноситься цей файл.

Отже, поєднання всіх розроблених функцій дозволяє зібрати та обробити дані, які будуть використані на наступних етапах реалізації системи.

Скрипт змінює формат даних з *wide* на *long*. У «широкому» форматі представлено рейтинги в їх оригінальному вигляді. В даному випадку дисципліни є заголовками стовпчиків. У «довгому» форматі заголовки всіх стовпчиків є категоріями. У *pandas* застосовується функція **melt** для зміни формату [15].

Також на деяких етапах роботи алгоритму відбувається об'єднання декількох датасетів за допомогою **декартового добутку** (функція **merge** з *pandas* [16]). Принцип роботи даного алгоритму представлено на рис. 4.3.

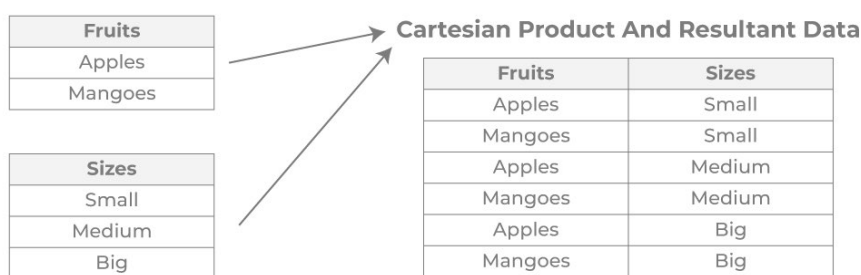


Рисунок 4.3 – Декартовий добуток множин

**Конкатенація** наборів даних представляє собою об'єднання декількох наборів даних у один набір (у *pandas* – функція **concat** [17]). Це об'єднання відбувається вертикально. На рис. 4.4 продемонстровано принцип конкатенації. Очевидно що об'єднують датасети з однаковою структурою.

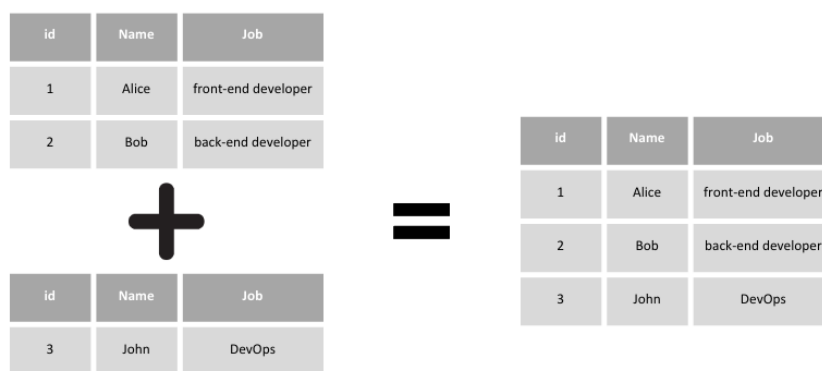


Рисунок 4.4 – Приклад конкатенації наборів даних

Отже, результуючий набір даних формується завдяки застосуванню конкатенації та декартового добутку.

Алгоритм формування набору даних представлено за допомогою блок-схеми на рис. 4.5.

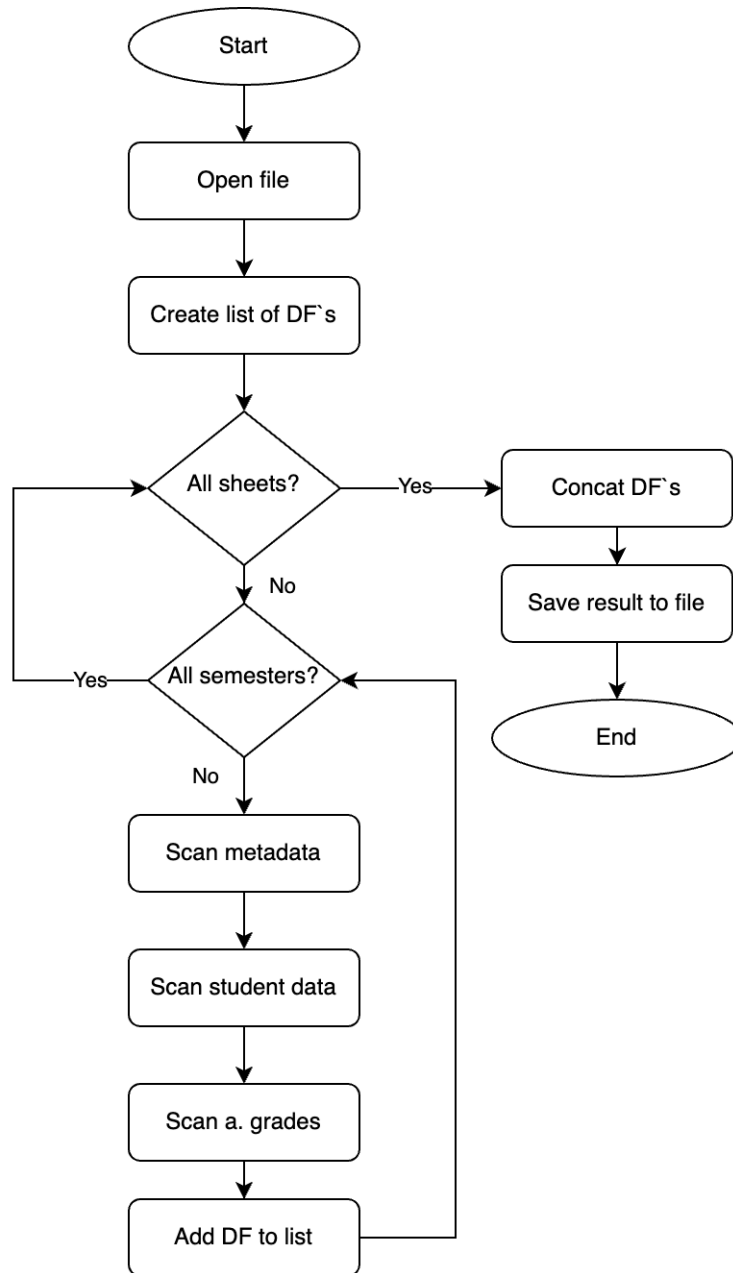


Рисунок 4.5 – Блок-схема алгоритму формування набору даних

На блок-схемі (рис. 4.5) «DF» є скороченням від «DataFrame», а «a. grades» є позначенням для «additional grades». Отже, розроблений алгоритм передбачає покроковий аналіз кожного аркуша в межах обраного файлу та кожного семестру на обраному аркуші.

### 4.3 Тестування розробленого коду

Тестування дозволяє пересвідчитися у коректності функціонування розробленого застосунка. В даній системі використано автоматизоване тестування за допомогою **unittest**. Вказана бібліотека дозволяє створювати unit-тести.

Unit-тестування – це процес перевірки коректності функціонування окремих компонентів програмного забезпечення (зазвичай функцій або методів). Основна ідея полягає в тому щоб визначити чи співпадають очікування з реальними результатами роботи функції. Переваги unit-тестів:

1. Автоматизація. Немає необхідності проводити тестування вручну. Отже, ймовірність помилок через людський фактор та витрати часу скорочуються. Також це суттєво полегшує повторне виконання тестів в разі потреби.

2. Ізольованість. Кожен unit-тест перевіряє лише одну конкретну функцію або метод, ізолюючи її від інших частин програми. Тобто при негативних результатах тестування можна легко встановити що саме є джерелом проблеми. Результати одного unit-тесту не повинен впливають на результати інших тестів.

В межах даної інформаційної системи протестовано функції для визначення спеціальності та курсу за номером групи (*detect\_specialty* та *detect\_course*). Результати тестування наведено на рис. 4.6.

```
===== test session starts =====  
collecting ... collected 2 items  
  
automated_testing_example.py::TestEdApp::test_detect_course PASSED [ 50%]  
automated_testing_example.py::TestEdApp::test_detect_specialty PASSED [100%]  
  
===== 2 passed in 0.04s =====
```

Рисунок 4.6 – Результати виконання тестів

Програмна реалізація unit-тестів представлена у додатку Б. Розроблені тести пройшли, отже протестовані функції можуть бути застосовані для доповнення створеного набору даних.

#### **4.4 Налаштування інтерактивного звіту для візуалізації результатів аналізу даних**

Для візуалізації даних та створення інтерактивного звіту використано MS Power BI. Microsoft Power BI є інструментом для візуалізації та аналізу даних. Він дозволяє користувачам збирати, перетворювати, візуалізувати та аналізувати дані з різних джерел (бази даних, хмарні сервіси, Excel-файли тощо).

Існує декілька окремих компонентів Power BI, а саме Power BI Desktop (для створення звітів та візуалізацій), Power BI Service (хмарна платформа для публікації звітів) та Power BI Mobile (мобільний застосунок для перегляду та спільного користування звітами) [18].

Power BI дозволяє користувачам ефективно перетворювати необроблені дані на корисну інформацію за допомогою створення динамічних та інтерактивних звітів. Інтерактивність звіту надає йому ще більшу інформативність, оскільки користувач сам обирає які саме дані та у якому контексті бажає отримати. Всі візуальні елементи (діаграми та графіки) є дуже гнучкими завдяки великій кількості опцій для налаштування. Ці налаштування стосуються не лише дизайну, а і способу організації даних.

Окремо варто відзначити Power Query, який застосовано для групування та агрегації даних у розробленому датасеті. Power Query – це потужний інструмент для збору, трансформації та очищення даних в середовищі Power BI [19]. Він дозволяє підключатися до різних джерел даних та здійснювати ряд операцій з цими даними для підготовки їх до аналізу та візуалізації. Основні функції Power Query включають в себе:

1. *Підключення до джерел даних.* Power Query має широкий спектр підтримуваних джерел даних, включаючи реляційні бази даних, файли Excel, текстові файли, CSV, XML, JSON, директорії тощо.

2. *Трансформація даних.* Power Query дозволяє здійснювати фільтрацію, сортування, групування, об'єднання таблиць, додавання стовпців тощо.

3. *Очищення даних.* Power Query надає інструменти для очищення даних від дублікатів, недійсних значень, помилок та інших непотрібних елементів.



4. *Автоматизація операцій.* Power Query дозволяє автоматизувати процеси трансформації даних шляхом створення спеціальних скриптів. Також зберігається історія внесених змін. У будь-який момент можна повернутися до попередніх етапів обробки даних. Наприклад, на рис. 4.7 зображена історія змін, які внесено у завантажений датасет.

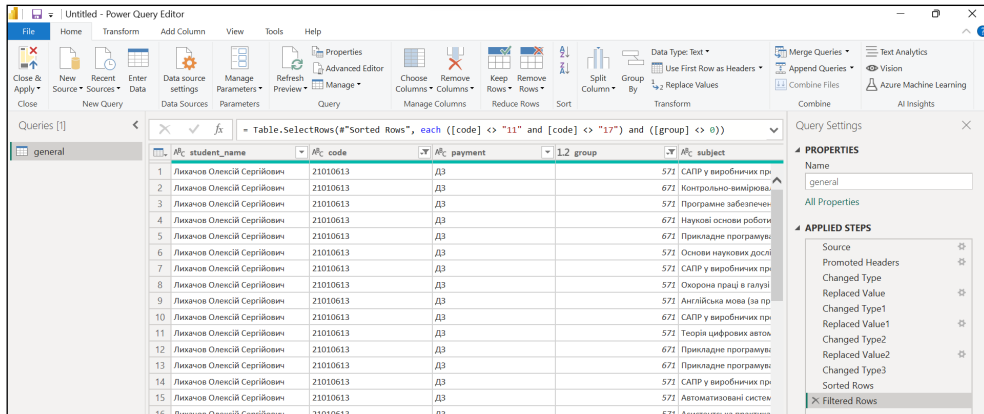


Рисунок 4.7 – Приклад інтерфейсу PowerQuery

Приклад додавання нового стовпчик у датасет представлено на рис. 4.8. Цей стовпчик дозволяє створити категоріальну ознаку у форматі оцінки ECTS для розрахованих рейтингових балів.

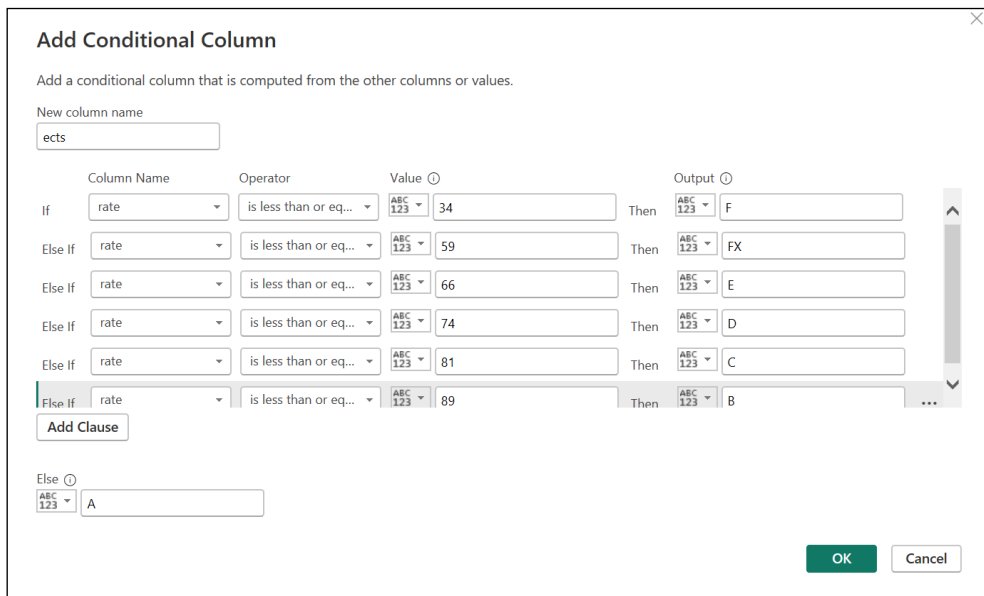


Рисунок 4.8 – Приклад додавання conditional column [20]

Дизайн та структура інтерактивного звіту налаштовані відповідно до вимог та поставленого завдання. При налаштуванні використано стандартні опції [21].

Для зручності подальшої обробки варто об'єднати дані про навчальний семестр у один рядок. Формат цього рядка варто задати як номер семестру, рік початку навчального року та рік закінчення навчального року. Все одно ці дані завжди використовуються разом. Новий формат даних про навчальний семестр представлено на рис. 4.9.

ABC payment	ABC semester	ABC specialty
Контракт	1_2017_2018	Системний аналіз
Держ. замовлення	2_2017_2018	Комп'ютерні науки
Держ. замовлення	1_2018_2019	Комп'ютерна інженерія
Держ. замовлення	1_2018_2019	Автоматизація та КІТ
Держ. замовлення	1_2017_2018	Комп'ютерні науки
Держ. замовлення	1_2018_2019	Автоматизація та КІТ
Держ. замовлення	1_2018_2019	Автоматизація та КІТ
Держ. замовлення	2_2018_2019	Комп'ютерні науки
Держ. замовлення	2_2017_2018	Комп'ютерна інженерія

Рисунок 4.9 – Оновлені дані про навчальний семестр

Наступним кроком необхідно розрахувати рейтинговий бал за формулою з пункту 1.4. Для цього необхідно виконати групування разом з агрегацією (рис. 4.10). Варто зазначити що в даному випадку можна прибрати з запиту дані про назву дисципліни, кількість годин та кредитів, ім'я викладача та спосіб здійснення контролю знань (іспит, залік, атестація, курсова робота тощо).

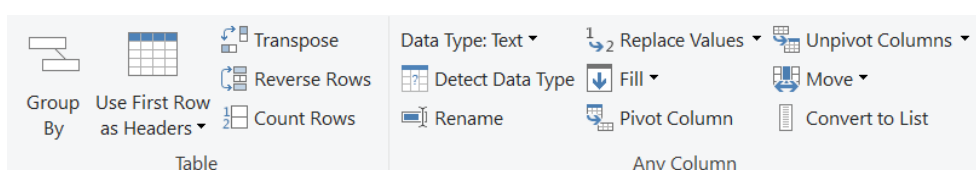


Рисунок 4.10 – Функція для здійснення групування

В даному випадку необхідно здійснювати процес розрахунку покроково. Перш за все необхідно порахувати добуток балів та кредитів. Далі варто виконати групування на основі даних про здобувача та про номер семестру. В ході групування можливо вказати правило для агрегації. В даному випадку це буде сума для всіх інших стовпчиків крім одного, який містить додаткові бали.

Додаткові бали нараховуються не за кожену дисципліну окремо, а взагалі за весь семестр. Це потрібно врахувати при агрегації. Отже, налаштовані правила для групування представлено на рис. 4.11.

New column name	Operation	Column
grades_credit_sum	Sum	grades_credit
credits_sum	Sum	credits

Рисунок 4.11 – Правила групування даних

Результат множення балу за дисципліну на кредит цієї дисципліни позначено як `grades_credit`. Для швидкого розрахунку значень арифметичних операцій над полями набору даних існують спеціальні функції (рис. 4.12).

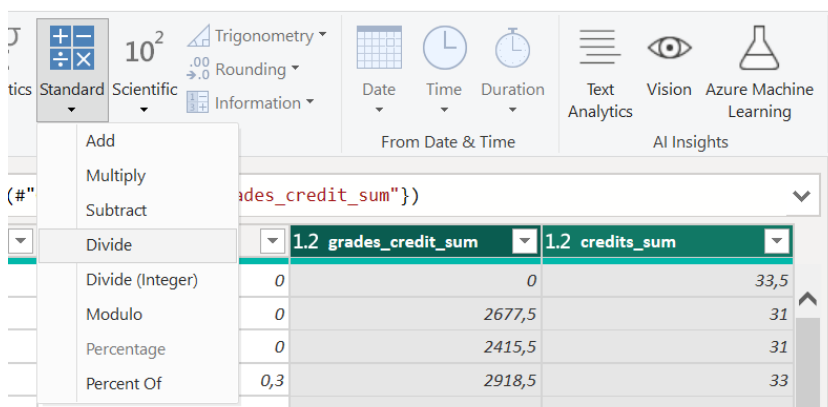


Рисунок 4.12 – Варіанти розрахунків для стовпців

Отже, після розрахунку рейтингових балів можливо переходити до налаштування візуальної складової звіту.

Майже для всіх діаграм з інтерактивного звіту виконується розрахунок кількості здобувачів у певній категорій. Важливо зазначити що в такій ситуації варто використовувати `Count(Distinct)` для того щоб рахувати без дублікатів (рис. 4.13) [22].

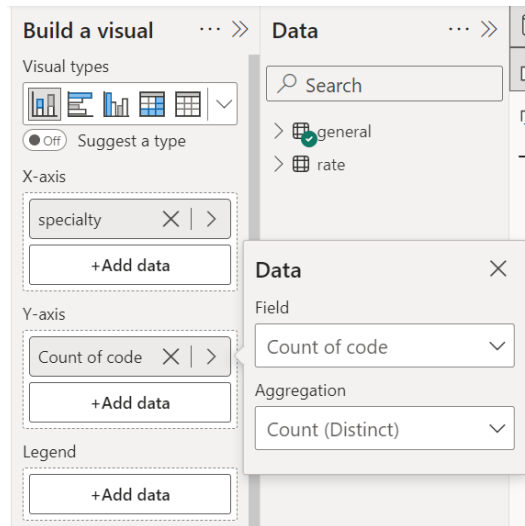


Рисунок 4.13 – Розрахунок без дублікатів

Для налаштування дизайну існує великий вибір заздалегідь налаштованих тем звіту. Але в даній роботі прийнято рішення розробити дизайн самостійно. Всі необхідні опції знаходяться на вкладці «Format». Приклад налаштованої діаграми представлено на рис. 4.14.



Рисунок 4.14 – Приклад діаграми

Отже, інтерактивний звіт налаштовано. Фільтрація даних відбувається при взаємодії з елементами доданих до звіту діаграм.

## 4.5 Керівництво користувача

Розроблений звіт складається з двох частин: аналізу структури факультету комп'ютерних наук (дослідження кількості здобувачів за спеціальністю, формою навчання, номером курсу тощо) та аналізу даних про академічну успішність різних категорій здобувачів. На рис. 4.15 представлено приклад звіту про кількість здобувачів вищої освіти ФКН (вигляд за замовчуванням).

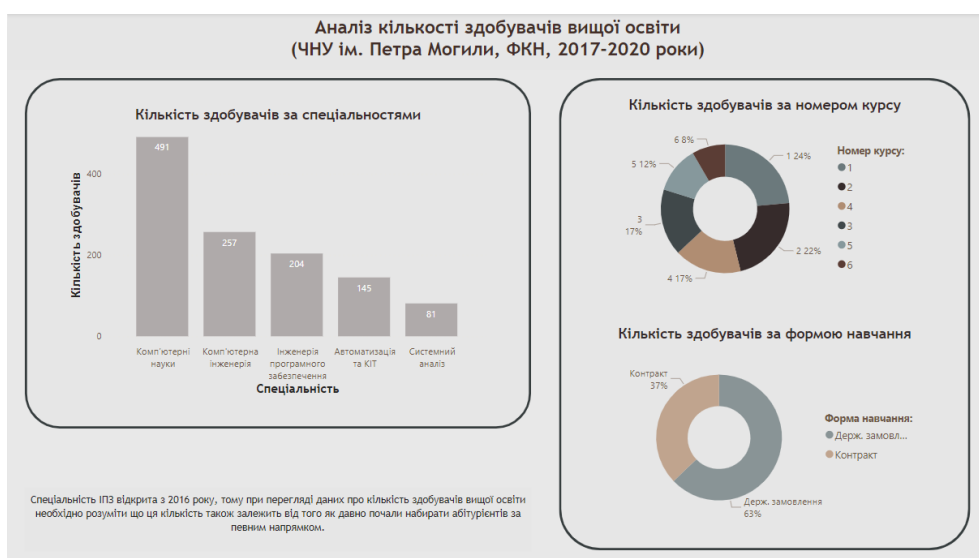


Рисунок 4.15 – Звіт про кількість здобувачів вищої освіти

Якщо користувач обере певну форму навчання, то візуалізація буде перебудована для надання статистики про обрану категорію (рис. 4.16).

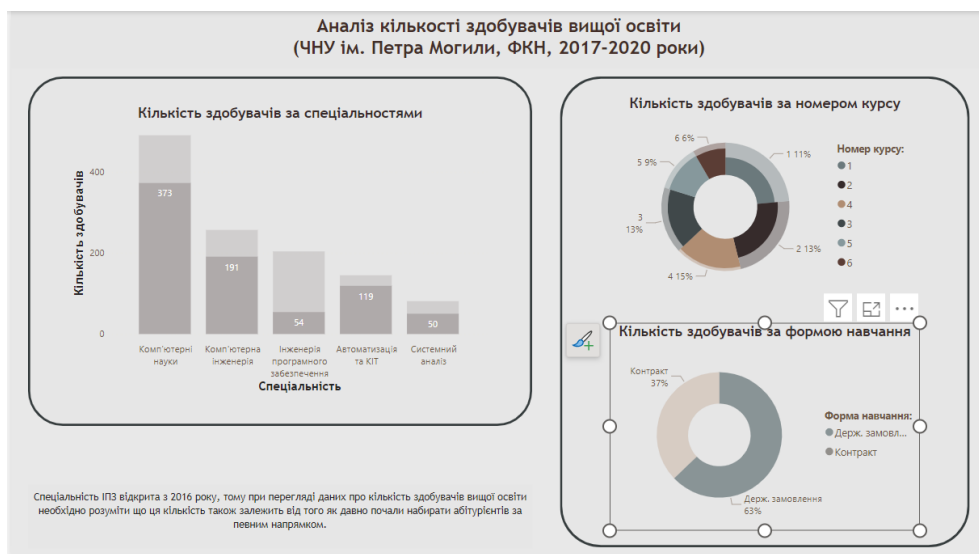


Рисунок 4.16 – Статистика про обрану категорію здобувачів

Також можна відфільтрувати дані за номером курсу. Приклад отриманих результатів представлено на рис. 4.17. На цьому прикладі наведено звіт про кількість першокурсників за спеціальностями та співвідношення державного замовлення та контракту.

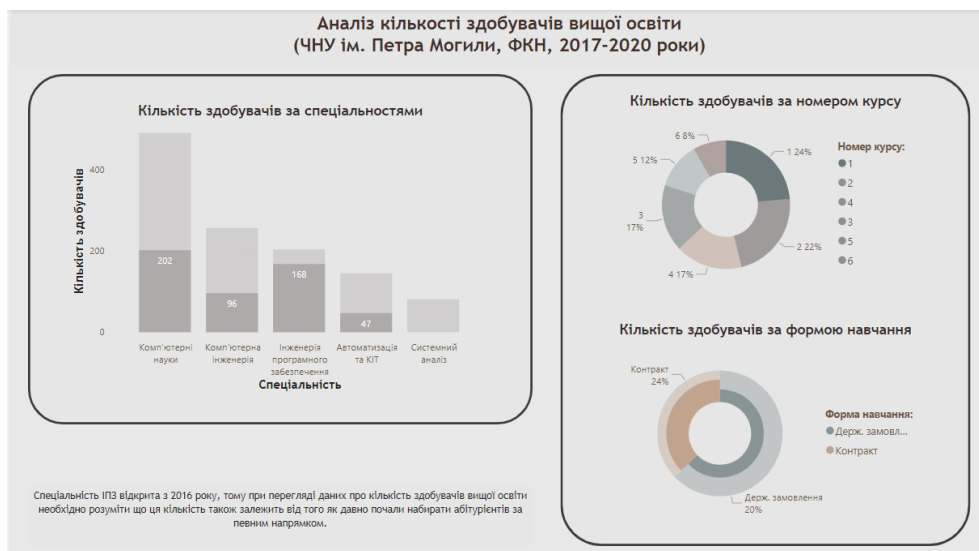


Рисунок 4.17 – Дані про кількість першокурсників

При дослідженні академічної успішності цікаво подивитися розподіл балів для «бюджетників» та «контрактників». За відсутності фільтрації бали підпорядковуються нормальному закону розподілу. Але якщо обрати «Держ. замовлення» в якості параметра фільтрації, то почнуть переважати оцінки «А+» (рис. 4.18).

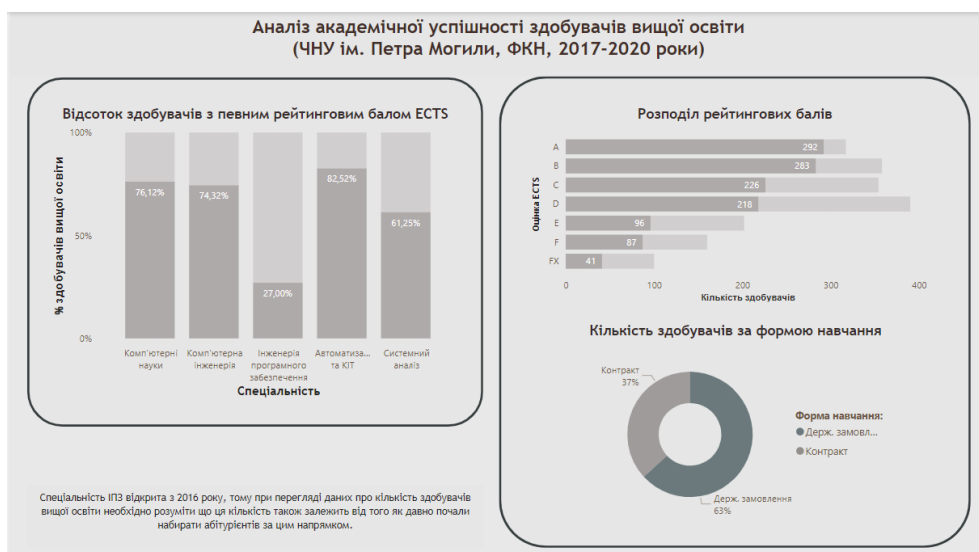


Рисунок 4.18 – Розподіл балів для «Держ. Замовлення»

Якщо обрати здобувачів які навчаються за контрактом, то ситуація змінюється радикальним чином. Найпоширенішою оцінкою є «D». Також видно що спеціальність «Інженерія програмного забезпечення» має найменше «бюджетників» серед усіх інших спеціальностей (рис. 4.19).

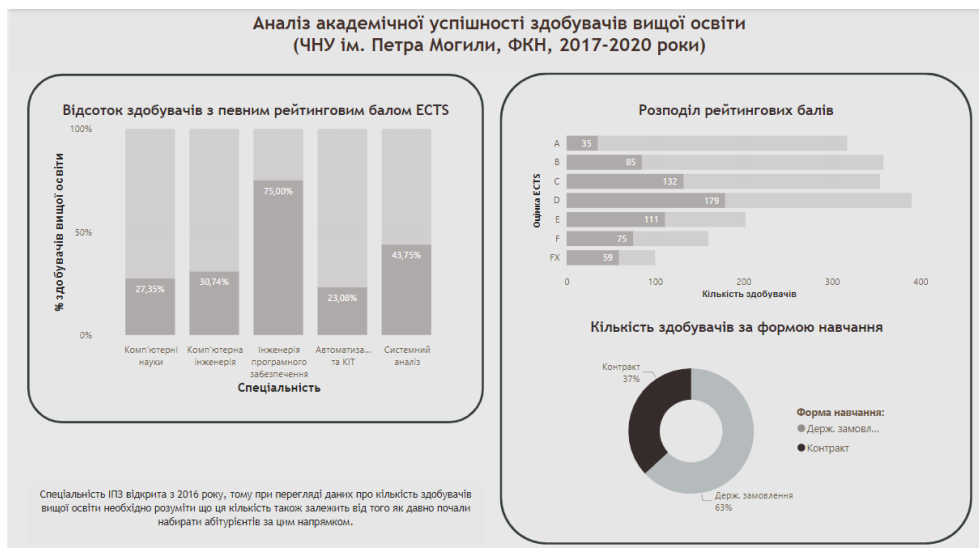


Рисунок 4.19 – Розподіл балів для «Контракт»

Можливо дослідити відсоток здобувачів з певними балами в межах кожної спеціальності. Наприклад, на рис. 4.20 представлено відсоток «відмінників». Спеціальність «Автоматизація та КІТ» має найкращі показники в цьому випадку. Найбільше студентів з оцінками «A» навчається на бюджеті.

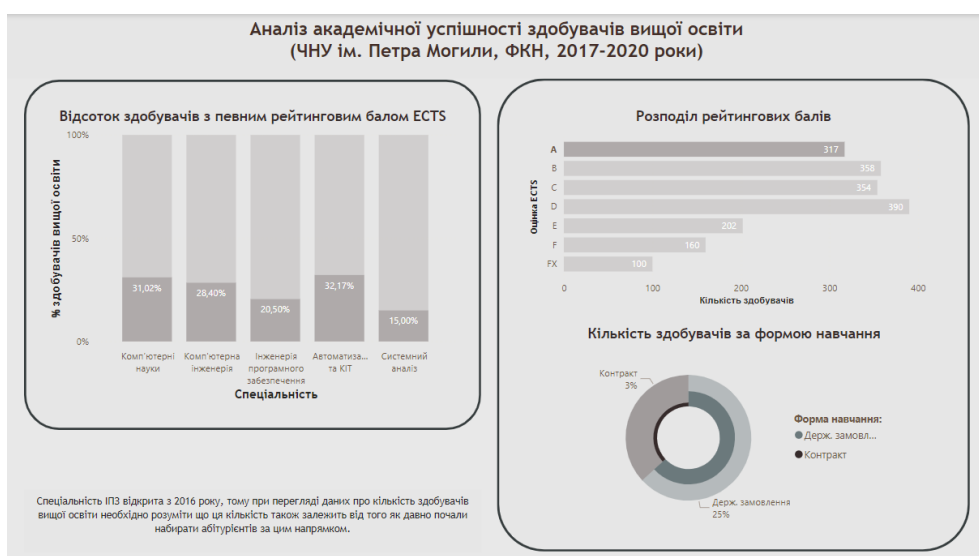


Рисунок 4.20 – Відсоток здобувачів з оцінкою «A»

Отже, розроблений звіт може бути використаний для подальшого аналізу академічної успішності здобувачів вищої освіти. При інтерпретації результатів варто покладатися на застереження стосовно меж застосування обраного підходу. Рейтингові бали є важливим, але не єдиним підходом до дослідження успішності здобувачів та якості закладу вищої освіти. Також існують специфічні аспекти, які визначають зміст отриманих результатів. Наприклад, в створеному датасеті «Інженерія програмного забезпечення» не характеризується найбільшою кількістю здобувачів серед інших спеціальностей. При інтерпретації цього факту варто врахувати що лише у 2016 році з'явилася можливість поступити на цю спеціальність, тому на момент формування використаних у роботі рейтингових таблиць кількість здобувачів на відповідній спеціальності не могла бути надто великою.

Розроблений скрипт дозволяє завантажити ще більшу кількість рейтингових таблиць (якщо виникне така потреба). Розроблений звіт містить всі необхідні інструменти для трансформації даних (CRUD) та може бути адаптований до датасету з подібною структурою та інакшим змістом. Отже, вдалося автоматизувати процес створення зведеної статистики для факультету комп'ютерних наук ЧНУ ім. Петра Могили.

#### **Висновки до розділу 4**

У четвертому розділі кваліфікаційної роботи магістра здійснено дослідження технологій та інструментів для аналізу даних, які включають в себе використання мови програмування *Python* для реалізації алгоритмів збору даних та *MS Power BI* для створення звітів. В ході програмної реалізації враховано швидкість та ефективність мови *Python*, що дозволило оптимально виконувати завдання з обробки великого обсягу даних. Також виконано тестування розробленого коду. Налаштовано візуалізацію результатів аналізу даних у *MS Power BI*. Вдалося продемонструвати комплексний підхід до аналізу даних.



## ВИСНОВКИ

Застосування інформаційних технологій в сфері вищої освіти змінило підхід до навчального процесу та відкрило нові можливості у аналізі та оптимізації освітніх процесів. Діджиталізація стала ключовим фактором у створенні ефективних інструментів для моніторингу академічної успішності студентів та розробки стратегій управління освітнім процесом. Аналіз даних, як необхідний етап в цьому процесі, виявився надзвичайно корисним для розуміння потреб і проблем закладу вищої освіти.

Виявлення закономірностей та трендів у навчальних процесах за допомогою аналізу даних стало важливим інструментом для закладів вищої освіти. Ці закономірності можуть бути важливими індикаторами проблем, які вимагають уваги та вирішення. Аналіз даних дозволяє глибше зрозуміти природу та причини виявлених недоліків, що в свою чергу допомагає у розробці та впровадженні ефективних стратегій для вирішення цих проблем.

Мета кваліфікаційної роботи полягала у вдосконаленні процесу обліку та аналізу академічної успішності здобувачів за рахунок розробки та впровадження інформаційної системи. Об'єктом дослідження є процес оцінювання успішності здобувачів вищої освіти, а предметом – методи та засоби розробки інформаційної системи для моніторингу рейтингових балів. Для досягнення поставленої мети виконано комплекс завдань: аналіз предметної сфери, розробка моделей даних та процесів, специфікація вимог, вибір алгоритмів обробки даних та програмна реалізація інформаційної системи.

Рейтингові бали визначають академічну успішність здобувачів та відіграють ключову роль у визначенні їх подальшої кар'єри та можливостей для отримання освіти. Отже, вдосконалення процесу обліку та аналізу рейтингових балів є актуальним завданням та має практичне значення. Розроблений скрипт та інтерактивний звіт можуть бути застосовані для оптимізації процесу створення звітності у закладі вищої освіти.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про вищу освіту: Закон України від 28 вересня 2017 року № 1556-18. URL: <https://zakon.rada.gov.ua/laws/show/1556-18#Text> (дата звернення: 26.12.2023).
2. European Credit Transfer and Accumulation System (ECTS): European Education Area URL: <https://education.ec.europa.eu/> (дата звернення: 26.12.2023).
3. Learning Management System Software (LMS) URL: <https://www.iitms.co.in/learning-management-system/> (дата звернення: 26.12.2023).
4. MOODLE3 ЧНУ імені Петра Могили URL: <https://moodle3.chmnu.edu.ua/> (дата звернення 26.12.2023).
5. Coursera URL: <https://www.coursera.org/> (дата звернення: 26.12.2023).
6. Електронні класні журнали та щоденники з можливостями дистанційного навчання URL: <https://nz.ua/#> (дата звернення: 26.12.2023).
7. Li Q., Chen Y.-L. IDEF0 Function Modeling. Modeling and Analysis of Enterprise and Information Systems: From Requirements to Realization. Berlin, Heidelberg, 2009. P. 98 – 122.
8. Menzel C., Mayer R. The IDEF family of languages. 2006. P. 215-249. URL: [https://doi.org/10.1007/3-540-26661-5\\_10](https://doi.org/10.1007/3-540-26661-5_10) (дата звернення: 15.02.2024)
9. What is Unified Modeling Language (UML)?: Visual Paradigm URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> (дата звернення 27.12.2023).
10. UML Use Case Diagram Tutorial: Lucidchart URL: <https://www.lucidchart.com/pages/uml-use-case-diagram> (дата звернення: 28.12.2023).
11. What is Component Diagram?: Visual Paradigm URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/> (дата звернення: 28.12.2023).
12. What is an Entity Relationship Diagram (ERD)? URL: <https://www.lucidchart.com/pages/er-diagrams> (дата звернення: 27.12.2023).

13. What Makes a Good Visualization?: David McCandless URL: <https://informationisbeautiful.net/visualizations/what-makes-a-good-data-visualization/> (дата звернення: 17.02.2024).

14. Working with Excel files using pandas: GeeksForGeeks URL: <https://www.geeksforgeeks.org/working-with-excel-files-using-pandas/> (дата звернення: 17.02.2024).

15. Melt function documentation URL: <https://pandas.pydata.org/docs/reference/api/pandas.melt.html> (дата звернення: 18.02.2024).

16. Merge function documentation URL: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html> (дата звернення: 18.02.2024).

17. Concat function documentation URL: <https://pandas.pydata.org/docs/reference/api/pandas.concat.html> (дата звернення: 18.02.2024).

18. Power BI: Microsoft URL: <https://www.microsoft.com/en-us/power-platform/products/power-bi#Product-overview> (дата звернення: 18.02.2024).

19. What is Power Query?: Microsoft Learn URL: <https://learn.microsoft.com/en-us/power-query/power-query-what-is-power-query> (дата звернення: 18.02.2024).

20. Add a conditional column (Power Query): Microsoft URL: <https://support.microsoft.com/en-au/office/add-a-conditional-column-power-query-f2422ed1-f565-4e64-ba3f-4f8d4616254e> (дата звернення: 18.02.2024).

21. Power BI design: Microsoft URL: <https://learn.microsoft.com/en-us/power-bi/create-reports/service-dashboards-design-tips> (дата звернення: 18.02.2024)

22. DISTINCTCOUNT function (DAX): Microsoft Learn URL: <https://learn.microsoft.com/en-us/dax/distinctcount-function-dax> (дата звернення: 18.02.2024)

## ДОДАТОК А

### Текст програмної реалізації алгоритмів читання та обробки даних

```
# Функція для збору даних про дисципліни та викладачів
def collect_subject_metadata(df):
    # Detect the cut points
    mask = df == "Прізвище, ім'я студента"
    semester_data_start = np.where(mask.any())[0]

    mask = df == "Сер.зважений бал за навчання"
    semester_data_end = np.where(mask.any())[0]

    metadata_all = list()
    for i in range(0, len(semester_data_start)): # для 2-х семестрів

        if len(semester_data_start) != len(semester_data_end):
            continue

        semester = df.iloc[:, semester_data_start[i]:semester_data_end[i]]

        # Detect rows with subject metadata
        credits_row_index = semester[semester.eq("Кредити").any(axis=1)].index.tolist()[0]
        teachers_row_index = credits_row_index + 5

        # Drop extra rows
        metadata = semester.loc[credits_row_index:teachers_row_index].dropna(axis=1)
        metadata = metadata.iloc[:, 1:]
        metadata = metadata[:-1]

        # Rename row indexes
        new_index_names = ["credits", "hours", "type", "subject", "teacher_name"]
        metadata = metadata.rename(index=dict(zip(metadata.index, new_index_names)))

        metadata_all.append(metadata)

    return metadata_all

# Функція для збору даних про оцінки
def collect_student_data(df):
    # Detect the cut points
    mask = df == "Прізвище, ім'я студента"
    semester_data_start = np.where(mask.any())[0]

    mask = df == "Сер.зважений бал за навчання"
    semester_data_end = np.where(mask.any())[0]

    student_data_all = list()
    for i in range(0, len(semester_data_start)): # для 2-х семестрів

        if len(semester_data_start) != len(semester_data_end):
            continue

        semester = df.iloc[:, semester_data_start[i]:semester_data_end[i]]

        # If some row is not about student it has NaN in "Номер заліковки"
        semester = semester.dropna(subset=[semester.columns[1]])

        # Числове позначення академічних заборгованостей
        semester.fillna(0, inplace=True)
        semester.replace("н", 0, inplace=True)

        # Індекс на основі першого рядка
        semester.columns = semester.iloc[0]
        semester = semester[1:]

    student_data = semester.melt(id_vars = semester.columns[0:4])
```

```
student_data.columns = ["student_name", "code", "payment", "group", "subject",
"grade"]

student_data_all.append(student_data)

return student_data_all

# Функція для збору даних про додаткові бали
def collect_additional_grades(df):
    # Detect the cut points
    mask = df == "Прізвище, ім'я студента"
    semester_data_start = np.where(mask.any())[0]

    mask = df == "Σ додаткових балів (від 0 до 10)"
    semester_data_end = np.where(mask.any())[0]

    additional_grades_data_all = list()
    for i in range(0, len(semester_data_start)): # для 2-х семестрів

        if len(semester_data_start) != len(semester_data_end):
            continue

        semester = df.iloc[:, semester_data_start[i]:semester_data_end[i] + 1]

        # If some row is not about student it has NaN in "Номер заліковки"
        semester = semester.dropna(subset=[semester.columns[1]])

        # Select only surname and additional grades
        semester = semester.iloc[:, [0, -1]].dropna()

        # Rename cols
        semester.columns = ["student_name", "additional_grade"]

        additional_grades_data_all.append(semester)

    return additional_grades_data_all

# Функція для визначення спеціальності за номером групи
def detect_specialty(group_number_str):
    if isinstance(group_number_str, str) and len(group_number_str) > 2:
        specialty_codes = {
            "01": "Комп'ютерні науки",
            "02": "Комп'ютерні науки",
            "03": "Комп'ютерні науки",
            "04": "Комп'ютерні науки",
            "05": "Комп'ютерна інженерія",
            "06": "Комп'ютерна інженерія",
            "07": "Системний аналіз",
            "08": "Інженерія програмного забезпечення",
            "09": "Інженерія програмного забезпечення",
            "10": "Інженерія програмного забезпечення",
            "71": "Автоматизація та КІТ",
        }

        code = group_number_str[1:3]

        if code in specialty_codes:
            return specialty_codes[code]

    return None

# Функція для визначення курсу за номером групи
def detect_course(group_number_str):
    if isinstance(group_number_str, str) and len(group_number_str) > 2:
        code = group_number_str[0]
        if code.isdigit():
            return code
    else:
        return None
```

```
# Функція для послідовного читання всіх аркушів у файлів
def xls_to_dataframe(file, study_year_start, study_year_end):
    groups = [sheet_name for sheet_name in file.sheet_names if re.match(r'^\d+m?$',
sheet_name) ]

    data = list()
    for group in groups:
        df = file.parse(group)

        specialty = detect_specialty(group)
        course = detect_course(group)

        metadata_all = collect_subject_metadata(df)
        student_data_all = collect_student_data(df)
        additional_grades_all = collect_additional_grades(df)

        for semester in range(0, len(metadata_all)):
            metadata = metadata_all[semester]
            student_data = student_data_all[semester]
            additional_grades = additional_grades_all[semester]

            merged_dataset = pd.merge(student_data, metadata.transpose(), on='subject')
            merged_dataset = pd.merge(merged_dataset, additional_grades, on='student_name')

            merged_dataset["study_year_start"] = study_year_start
            merged_dataset["study_year_end"] = study_year_end
            merged_dataset["semester"] = semester + 1
            merged_dataset["specialty"] = specialty
            merged_dataset["course"] = course

            data.append(merged_dataset)

    return pd.concat(data, axis=0)

# Приклад збору даних
file_path = "reiting_19-20.xls"
study_year_start = 2019
study_year_end = 2020

file = pd.ExcelFile(file_path)
result = xls_to_dataframe(file, study_year_start, study_year_end)
result.to_csv(f"grades_data_{study_year_start}-{study_year_end}.csv", index=False)

files = ["grades_data_2017-2018.csv", "grades_data_2018-2019.csv",
"grades_data_2019-2020.csv"]
df = pd.concat(map(pd.read_csv, files), ignore_index=True)
df.to_csv(f"general.csv", index=False)
```

## ДОДАТОК Б

### Текст програмної реалізації unit-тестів

```
import unittest

# Метод для визначення спеціальності за номером групи
def detect_specialty(group_number_str):
    if isinstance(group_number_str, str) and len(group_number_str) > 2:
        specialty_codes = {
            "01": "Комп'ютерні науки",
            "02": "Комп'ютерні науки",
            "03": "Комп'ютерні науки",
            "04": "Комп'ютерні науки",
            "05": "Комп'ютерна інженерія",
            "06": "Комп'ютерна інженерія",
            "07": "Системний аналіз",
            "08": "Інженерія програмного забезпечення",
            "09": "Інженерія програмного забезпечення",
            "10": "Інженерія програмного забезпечення",
            "71": "Автоматизація та КІТ",
        }

        code = group_number_str[1:3]

        if code in specialty_codes:
            return specialty_codes[code]

    return None

# Метод для визначення курсу за номером групи
def detect_course(group_number_str):
    if isinstance(group_number_str, str) and len(group_number_str) > 2:
        code = group_number_str[0]
        if code.isdigit():
            return code
    else:
        return None

# AUTOMATED TESTS
class TestEdApp(unittest.TestCase):

    def test_detect_specialty(self):
        self.assertEqual(detect_specialty("101"), "Комп'ютерні науки")
        self.assertEqual(detect_specialty("108"), "Інженерія програмного забезпечення")
        self.assertEqual(detect_specialty("671м"), "Автоматизація та КІТ")
        self.assertEqual(detect_specialty("67"), None)

    def test_detect_course(self):
        self.assertEqual(detect_course("101"), "1")
        self.assertEqual(detect_course("671"), "6")
        self.assertEqual(detect_course("-30"), None)
        self.assertEqual(detect_course("10"), None)

# RUN TESTS
if __name__ == "__main__":
    unittest.main()
```

## ДОДАТОК В

### Результати апробації кваліфікаційної роботи магістра

УДК.004.67

*Скрипник Р. В., Горбань Г. В.*

*Чорноморський національний університет ім. Петра Могили,  
м. Миколаїв, Україна*

#### МЕТРИКИ АКАДЕМІЧНОЇ УСПІШНОСТІ У ІНФОРМАЦІЙНІЙ СИСТЕМІ МОНІТОРИНГУ РЕЙТИНГОВИХ БАЛІВ

Метою інформаційної системи моніторингу рейтингових балів є оцінка академічної успішності студентів. На рис. 1 представлена концептуальна модель метрик академічної успішності.



Рисунок 1 – Метрики оцінки академічної успішності

Множина знань та навичок є найважливішим показником академічної успішності. Від цієї метрики залежать всі інші наведені метрики. Крім того, йдеться не лише про безпосередні результати вивчення курсів та набуті «hard skills», а і про розвинуті під час навчання універсальні якості особистості, тобто «soft skills».

Внутрішнє оцінювання передбачає контроль знань з боку університету (наприклад, рейтингові бали). Існують різні системи оцінювання. В Чорноморському національному університеті імені Петра Могили використовується система кредитів ECTS (European Credit Transfer and Accumulation System) та відповідна шкала оцінок. Дана система допомагає узгодити принципи розробки та оцінювання курсів у різних університетах. ECTS робить освітню систему більш прозорою та забезпечує академічну мобільність [1]. Результатами зовнішнього оцінювання вважаються, наприклад, успіхи студентів на олімпіадах та отримання професійних сертифікатів



(«CompTIA A+», «Microsoft Fundamentals», сертифікати про рівень володіння англійською мовою, тощо). Оцінка знань сторонніми організаціями важлива для підтвердження якості освіти.

Кар'єрні досягнення демонструють здатність студента застосовувати знання на практиці. Також це свідчить про загальний розвиток особистості, професійну етику та високий рівень мотивації. Крім того, отримана багатьма студентами робота за спеціальністю підтверджує актуальність та корисність навчальної програми. Попри наявність очевидної кореляції з якістю освіти дана метрика досить нестабільна та залежить від багатьох факторів. Наприклад, криза на ринку праці через зовнішні обставини може суттєво змінити ситуацію з працевлаштуванням. Тому некоректно робити висновки про академічну успішність лише за кар'єрними досягненнями.

Вмотивованість можна оцінити за активністю студента, тобто наявністю додаткових балів за участь у житті факультету, олімпіадах, конкурсах, тощо. Якщо студенту подобається вчитися, то він вчиться краще. Також якщо навчання дається легко, то студент має позитивне ставлення до отримання освіти. Отже, формування позитивного враження від навчання є складним процесом та оцінюється досить суб'єктивно. Наприклад, в ЧНУ ім. Петра Могили для визначення ставлення студентів до процесу отримання освіти регулярно проводяться відповідні опитування.

Одним з найважливіших показників академічного успіху є наполегливість студента та стабільність рівня зацікавленості навчанням. Завдяки інформаційній системі моніторингу рейтингових балів можливо оцінити зміну рейтингових балів з часом. В будь-яком випадку розрахунки для визначення динаміки академічної успішності передбачають об'єднання різних файлів з рейтингами у одну базу даних з наступною розробкою запитів для обчислення статистичних показників. Важливо вказати яку роль відіграють рейтингові бали серед інших метрик успішності, а отже оцінити об'єктивність та межі застосування отриманих результатів.

#### **ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. European Credit Transfer and Accumulation System (ECTS): European Education Area URL: <https://education.ec.europa.eu/> (дата звернення 26.12.2023).