

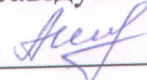
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Навчально-науковий інститут автоматики та електротехніки

Кафедра морського приладобудування

«Допущений до захисту»

Завідувач кафедри



Гайдай Г.Ю.

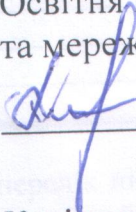
«17» червня 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня бакалавр

на тему: «Комп'ютерна система управління інформаційними потоками у
навчальному процесі»

Виконав: студент 4 курсу, групи 4651
Спеціальність: 123 Комп'ютерна інженерія
Освітня програма: Комп'ютерні системи
та мережі

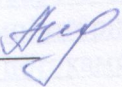


Івченко О.В.

(прізвище та ініціали)

Керівник к.т.н, доц.. Гайдай Г.Ю.

(прізвище та ініціали)



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Факультет

Навчально-науковий інститут автоматики та
електротехніки

Кафедра
Освітній ступінь
Спеціальність
Освітня програма

морського приладобудування
бакалавр
123 Комп'ютерна інженерія
Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ

Гарант освітньої програми
морського приладобудування

Гайдай к.т.н., доцент Гайдай Г.Ю.
« 18 » *червень* 2026 рік

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня бакалавр

Івженко Олексію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Комп'ютерна система управління інформаційними потоками у навчальному процесі»

керівник роботи Гайдай Ганна Юріївна, к.т.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від « 22 » 04 2026 року № 286 ур

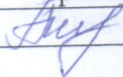
2. Строк подання студентом роботи 5 червня 2026 р.

3. Вихідні дані до роботи: призначення системи – інформаційна система; учбовий контент – інформаційні потоки університету.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): теоретичні аспекти управління інформаційними потоками; розробка технічного завдання на комп'ютерну систему; загальна концепція та практична реалізація системи управління інформаційними потоками; охорона праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): архітектура системи; структурна схема системи; загальна функціональна схема системи; загальний алгоритм роботи системи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
ОП	Гайдай Г.Ю.		

7. Дата видачі завдання 10 лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Огляд та аналіз існуючих систем управління інформаційними потоками	16.02.26	
2	Розробка технічного завдання	28.02.26	
3	Розробка архітектури системи	20.03.26	
4	Розробка структурної та функціональної схем	15.04.26	
5	Алгоритмічна реалізація системи	30.04.26	
6	Охорона праці	10.05.26	
7	Розробка пояснювальної записки	25.05.26	
8	Розробка графічних матеріалів	01.06.26	

Студент

Керівник роботи

(підпис)

(підпис)

Івженко О.В.

(прізвище та ініціали)

Гайдай Г.Ю.

(прізвище та ініціали)

АНОТАЦІЯ

Івженко О.В. «Комп'ютерна система управління інформаційними потоками у навчальному процесі». Кваліфікаційна робота зі спеціальності 123 «Комп'ютерна інженерія», освітньої програми «Комп'ютерні системи та мережі». Миколаїв: НУК ім. адм. Макарова. 2026 рік. 88 сторінок.

У кваліфікаційній роботі розроблено надійну, масштабовану та безпечну архітектуру модуля «Успішність студентів», який є ключовим елементом інтегрованої системи управління інформаційними потоками сучасного закладу вищої освіти. Розроблена архітектура успішно вирішує головні технологічні виклики традиційних університетських систем, а саме – вразливість до пікових сесійних навантажень, складність інтеграції з платформами дистанційного навчання та ризики людських помилок при калькуляції балів.

Ключові слова: інформаційні потоки, навчальний процес, університет, комп'ютерна система, автоматизація процесів.

ABSTRACT

Oleksii Ivzhenko. "Computer system for managing information flows during the educational process". Qualification work in the specialty 123 "Computer Engineering", educational program "Computer Systems and Networks". Mykolaiv: Admiral Makarov National University of Shipbuilding. 2026. 88 pages.

The qualification work developed a reliable, scalable and secure architecture of the "Student Success" module, which is a key element of the integrated information flow management system of a modern higher education institution. The developed architecture successfully solves the main technological challenges of traditional university systems, namely - vulnerability to peak session loads, the complexity of integration with distance learning platforms and the risks of human errors when calculating points.

Keywords: information flows, educational process, university, computer system, process automation.

ЗМІСТ

Вступ.....	
1 Теоретичні аспекти управління інформаційними потоками.....	
1.1 Поняття управління інформаційними потоками.....	
1.2 Класифікація інформаційних потоків та їх систем управління.....	
1.3 Адаптація програмного забезпечення.....	
1.4 Інтеграція освітніх інформаційних систем.....	
1.5 Класифікація та аналіз сучасних систем управління інформаційними потоками.....	
1.6 Об'єкти логістичного управління.....	
Висновки по розділу 1.....	
2 Розробка технічного завдання на комп'ютерну систему.....	
Висновки по розділу 2.....	
3 Загальна концепція та практична реалізація системи управління інформаційними потоками.....	
3.1 Архітектура модуля «Успішність студентів».....	
3.2 Структурна схема системи управління успішністю.....	
3.3 Функціональні вимоги до системи.....	
3.4 Функціональна схема системи.....	
3.5 Алгоритм життєвого циклу управління успішністю студента.....	
3.6 Порівняльний аналіз.....	
3.7 Практична цінність системи.....	
Висновки по розділу 3.....	

4 Охорона праці.....	
4.1 Детальний аналіз умов праці на робочому місці оператора інформаційної системи	
4.2 Ергономічні вимоги до організації робочого місця та проектування інтерфейсу	
4.3 Регламентовані перерви та комплекси оздоровчих вправ для оператора	
4.4 Виробнича санітарія та параметри мікроклімату.....	
4.5 Розрахунок штучного освітлення приміщення.....	
4.6 Електробезпека.....	
4.7 Охорона навколишнього середовища.....	
Висновки по розділу 4.....	
Загальні висновки.....	
Список використаних джерел.....	

ВСТУП

Актуальність роботи. Сучасний етап розвитку вищої освіти в Україні характеризується глибокою цифровою трансформацією та переходом до концепції «Університет у смартфоні». Умови змішаного та дистанційного навчання висувають нові вимоги до швидкості, надійності та якості обробки інформації. Щодня заклади вищої освіти оперують колосальними масивами взаємопов'язаних даних: від розкладів занять, навчальних планів і робочих програм до модульного контролю, обліку успішності студентів та кадрового менеджменту.

Використання застарілих децентралізованих підходів або паперового документообігу призводить до дублювання інформації, виникнення помилок та затримок у прийнятті управлінських рішень. Наявні на ринку готові комерційні рішення часто є надто коштовними для масового впровадження, складними в налаштуванні або не враховують специфіку нормативної бази конкретного закладу освіти. Тому розробка гнучкої, масштабованої комп'ютерної системи управління інформаційними потоками на базі сучасних та високоефективних мов програмування є актуальною науково-практичною задачею. Це дозволяє автоматизувати рутинні процеси, забезпечити прозорість моніторингу знань та знизити навантаження на адміністративно-викладацький склад.

Зв'язок роботи з науковими програмами, планами, темами. Кваліфікаційна робота на здобуття ступеня бакалавра виконувалась в Національному університеті кораблебудування імені адмірала Макарова.

Мета роботи: розробка та дослідження ефективності комп'ютерної системи управління інформаційними потоками для автоматизації адміністрування навчального процесу та підвищення оперативної взаємодії його учасників.

Для досягнення поставленої мети у бакалаврській роботі визначено наступні завдання:

- проаналізувати теоретичні основи організації інформаційних потоків в освіті та провести порівняльний аналіз існуючих аналогів.
- провести системний аналіз предметної області та формалізувати технічні вимоги до архітектури і функціональності системи.
- розробити концептуальну та логічну моделі реляційної бази даних для збереження та швидкої обробки інформаційних масивів.
- розробити алгоритми взаємодії між користувачами (адміністратор, викладач, студент) та модулями системи.
- провести тестування розробленого програмного забезпечення та оцінити ефективність його впровадження.

Об'єктом дослідження є процес управління інформаційними потоками та документообігом у межах організації навчального процесу в закладі вищої освіти.

Предметом дослідження є архітектура, моделі, алгоритми та програмні засоби автоматизації та координації інформаційних потоків навчального процесу.

Методи дослідження. Для вирішення визначених завдань використано методи системного аналізу, теорію реляційних баз даних, методології об'єктно-орієнтованого проектування та програмування (ООП) мовою Python, а також методи функціонального та інтеграційного тестування програмного забезпечення.

Наукова новизна одержаних результатів полягає в удосконаленні моделей інтеграції різнорідних інформаційних потоків (наприклад, інтерактивного розкладу та журналу успішності) в межах єдиного вебсередовища, що дозволяє знизити часові витрати на обробку та валідацію даних.

Практичне значення одержаних результатів. Розроблено діючий програмний комплекс, який може бути безпосередньо впроваджений у діяльність факультету або кафедри для автоматизації обліку успішності, генерації звітів, менеджменту розкладу та оперативного інформування студентів через вебінтерфейс.

Структура та обсяг роботи. Кваліфікаційну роботу викладено на 88 сторінках. Робота складається зі вступу, розділу 1, розділу 2, розділу 3, розділу 4, висновків до кожного розділу, висновків до роботи в цілому, переліку посилань.

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ УПРАВЛІННЯ ІНФОРМАЦІЙНИМИ ПОТОКАМИ

1.1. Поняття управління інформаційними потоками

Комп'ютерна система управління інформаційними потоками у навчальному процесі – це інтегрований програмно-апаратний комплекс (інформаційна система управління освітою / EMIS / LMS / ERP), призначений для автоматизованого збору, обробки, зберігання та розповсюдження навчальних, методичних, адміністративних і статистичних даних. Вона оптимізує комунікацію між усіма учасниками освітнього процесу та забезпечує інформаційну підтримку прийняття управлінських рішень [1].

Основна мета впровадження:

- перехід від паперового документообігу до єдиного цифрового простору;
- раціоналізація інформаційних потоків для зниження навантаження на персонал;
- підвищення прозорості та якості моніторингу успішності навчання;
- забезпечення оперативного доступу до навчальних матеріалів 24/7.

Основні інформаційні потоки в системі [2]

- Вхідні (нормативно-методичні):
 - Навчальні плани, освітні програми, робочі програми дисциплін.
 - Розклад занять, графіки сесій та екзаменів.
 - Накази, розпорядження керівництва та нормативні акти.
- Процесуальні (операційні):

- Навчальний контент (лекції, презентації, лабораторні роботи).
- Поточні завдання, тести, курсові проекти та результати їх виконання.
- Журнали відвідуваності та успішності.
- Вихідні (результативні):
 - Зведена аналітика та статистичні звіти для керівництва.
 - Відомості про успішність (електронні залікові книжки, дипломи).
 - Дані для рейтингування здобувачів освіти та викладачів (рис. 1.1).



Рисунок 1.1 – Основні інформаційні потоки в системі

Ключові функціональні модулі [3]

- Адміністративний модуль: Веде базу даних студентів та викладачів. Формує накази про зарахування, переведення, відрахування.
- Планово-диспетчерський модуль: Автоматизує розрахунок навчального навантаження викладачів та формування стабільного розкладу занять із врахуванням аудиторного фонду.

- Модуль управління навчанням (LMS): Відповідає за безпосередній навчальний процес. Містить інструменти для створення курсів, завантаження матеріалів, призначення дедлайнів та автоматичної перевірки тестів.

- Модуль контролю якості знань: Електронні журнали, відомості успішності, системи аналізу академічної успішності (виявлення студентів «групи ризику»).

- Комунікаційний модуль: Внутрішні чати, форуми, системи сповіщень та інтеграція з корпоративною поштою для взаємодії студент-викладач.

Сучасні вимоги до таких систем [4]

- Хмарна архітектура (Cloud-based): Доступ з будь-якого пристрою через браузер або мобільний додаток без прив'язки до локальних серверів закладу.

- Кібербезпека та захист персональних даних: Надійне шифрування каналів зв'язку та диференційований доступ до інформації (студент бачить лише свої оцінки, викладач – лише свої групи).

- Сумісність та інтеграція: Здатність системи обмінюватися даними з державними реєстрами (наприклад, ЄДЕБО в Україні) за допомогою API.

1.2. Класифікація інформаційних потоків та їх систем управління

Університетські інформаційні системи (УІС) розглядають адміністративні потоки як сукупність нормативно-довідкової, кадрової, організаційної та розпорядчої інформації. Огляд джерел у цьому напрямі фокусується на оптимізації внутрішнього документообігу, інтеграції з державними реєстрами та автоматизації процесів управління кадрами й контингентом здобувачів (рис. 1.2).

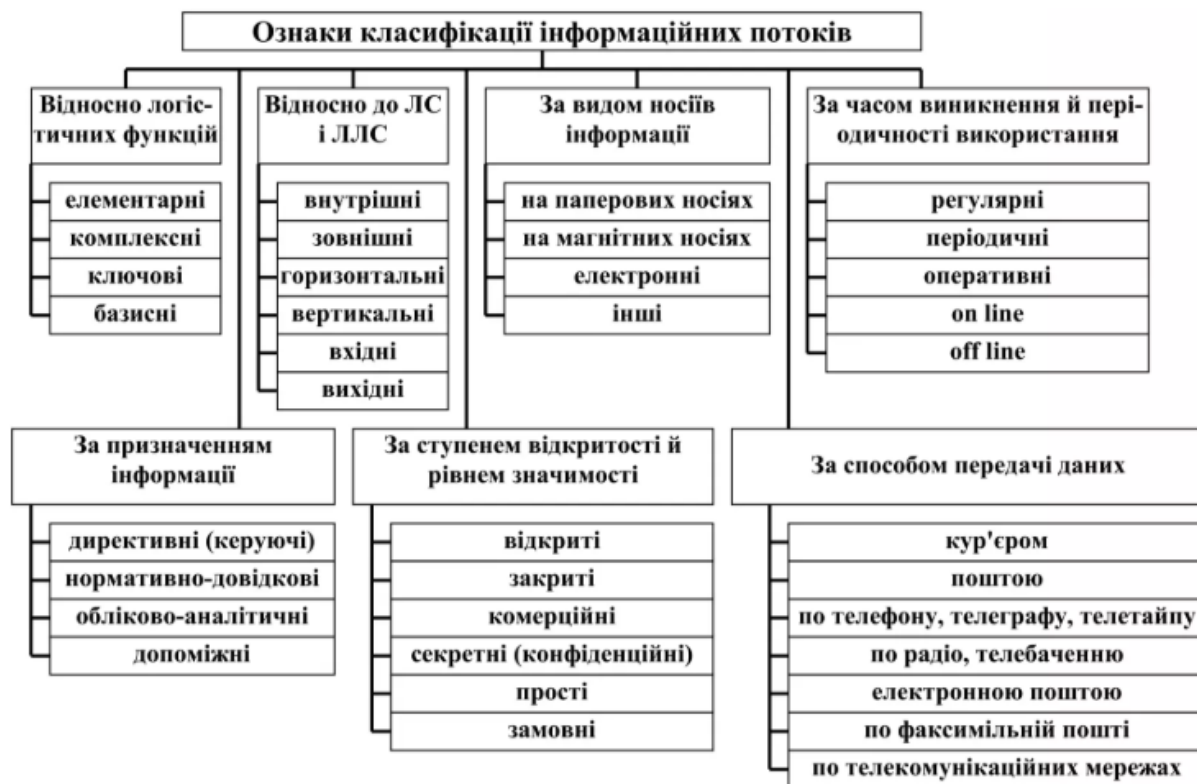


Рисунок 1.2 – Класифікація інформаційних потоків

Нижче наведено структуровану класифікацію наукових праць, підходів та прикладних рішень щодо комп'ютеризації адміністративних потоків у ЗВО [5].

1. Проєктування інтегрованих систем («Електронний університет»)

Джерела цієї категорії описують створення єдиного інформаційного простору, який зв'язує адміністрацію, факультети та приймальну комісію.

- Досвід вітчизняних ЗВО: У наукових працях дослідників Хмельницького національного університету (зокрема М. М. Косіюка) детально описано розробку та впровадження АІС «Електронний університет». Автори доводять, що інтеграція модулів дозволяє уникнути дублювання даних під час руху студента від «абітурієнта» до «випускника».

- Уніфіковані бази даних: Дослідження наголошують на важливості реляційних баз даних, де інформація про кадровий склад

(викладачів) та рух контингенту (накази про зарахування, відрахування, переведення) зберігається централізовано.

2. Автоматизація кадрових та організаційних потоків

Ця група джерел аналізує системи класу ERP (Enterprise Resource Planning), адаптовані під специфіку вищої освіти.

- Кадровий облік та навантаження: Публікації розглядають алгоритми автоматичного розрахунку та розподілу педагогічного навантаження кафедрами, що є одним із найскладніших адміністративних завдань.

- Штатний розпис: Наукові статті (наприклад, праці ХНЕУ ім. С. Кузнеця) висвітлюють впровадження ІС для формування штатного розпису та ведення обліку робочого часу співробітників.

3. Електронний документообіг та інтеграція з держсистемами

Сучасні джерела акцентують увагу на переході від паперових носіїв до безпаперових технологій та зв'язку із зовнішніми системами.

- ЄДЕБО (Єдина державна електронна база з питань освіти): Окремий пласт публікацій, присвячений шлюзам обміну даними між внутрішніми університетськими ІС та ЄДЕБО. Джерела описують технології миттєвої верифікації дипломів, синхронізації наказів та передачі даних приймальних комісій.

- Цифровий підпис та СЕД: У роботах з публічного управління та адміністрування розглядається впровадження систем електронного документообігу (СЕД), які використовують ЕЦП/КЕП для затвердження внутрішніх розпоряджень, наказів ректорату та протоколів засідань рад.

4. Моніторинг та підтримка прийняття рішень (BI-системи)

- Джерела описують використання технологій Business Intelligence (BI) для аналізу адміністративних потоків.

- Дослідники доводять, що комп'ютерний аналіз затримок у погодженні документів, моніторинг динаміки контингенту та звітність у

розрізі факультетів дозволяють керівництву (ректорату) ухвалювати обґрунтовані управлінські рішення.

1.3. Адаптація програмного забезпечення

Адаптація готового програмного забезпечення (ПЗ) – це найшвидший і найекономічніший спосіб створити систему управління інформаційними потоками. Замість написання коду з нуля, налаштовуються готові модулі під специфіку освітніх та адміністративних процесів [6].

Нижче наведено покрокову матрицю адаптації трьох найпопулярніших платформ під завдання [7].

1. Адаптація Moodle (Найпопулярніша LMS у світі)

Moodle ідеально підходить для управління навчальними та контрольними інформаційними потоками.

- Вхідні потоки (Контент): Структуруйте базу даних у вигляді «Категорія курсу» (Факультет) → «Підкатегорія» (Кафедра) → «Курс» (Дисципліна). Завантажуйте лекції, силабуси та методички у форматах PDF або SCORM.

- Процесуальні потоки (Навчання): Використовуйте плагін *Attendance* для автоматичного обліку відвідуваності. Налаштуйте модуль *Quiz* (Тести) з випадковою генерацією питань із банку даних для захисту від списування.

- Вихідні потоки (Аналітика): Налаштуйте *Gradebook* (Журнал оцінок) із формулами розрахунку підсумкового балу (наприклад, 60 балів за семестр + 40 за екзамен). Інтегруйте плагіни аналітики (наприклад, *Configurable Reports*) для вивантаження звітів для деканату.

2. Адаптація Google Workspace for Education

Найкраще рішення для оперативної комунікації, швидкого документообігу та динамічних інформаційних потоків [8].

- Вхідні потоки (Розпорядження та розклад): Публікація загальних наказів та розкладу через Google Sites (внутрішній портал). Створення автоматичних подій, консультацій та дедлайнів у Google Календарі для цілих груп.

- Процесуальні потоки (Завдання): Сервіс Google Classroom виступає ядром системи. Викладач публікує завдання, а студенти здають роботи у вигляді документів чи таблиць. Інформаційний потік автоматично сортується по папках на Google Диску.

- Вихідні потоки (Звітність): Збір опитувань та заявок через Google Форми. Зведена аналітика успішності та відвідуваності автоматично візуалізується за допомогою Google Looker Studio (інструмент для побудови красивих дашбордів для керівництва).

3. Адаптація Microsoft Teams for Education

Потужна екосистема для великих закладів, яка поєднує адміністративні потоки, комунікацію та хмарне сховище.

- Вхідні потоки (Управління): Створення архітектури «Команда» (Потік/Курс) → «Канал» (Конкретна дисципліна). Сховище файлів організується на базі SharePoint, де розмежовуються права (викладач редагує, студент лише читає).

- Процесуальні потоки (Взаємодія): Проведення онлайн-лекцій з автоматичним записом та генерацією субтитрів. Використання вбудованого модуля Assignments для перевірки робіт та виставлення оцінок (табл. 1.1).

- Вихідні потоки (Адміністрація): Використання вбудованого аналітичного інструменту Insights, який автоматично показує активність кожного студента, час виконання завдань та загальний рівень успішності групи [9].

Таблиця порівняння для вибору платформи

Критерій	Moodle	Google Workspace	Microsoft Teams
Спрямованість	Глибокий контроль навчання	Швидкий документообіг	Комунікація та синергія
Адміністративні звіти	Складні, детальні	Прості, візуальні	Автоматичні (Insights)
Складність налаштування	Висока (потрібен ІТ-спеціаліст)	Низька (інтуїтивно зрозуміло)	Середня (потрібна підписка)
Де зберігаються дані	На вашому сервері (повний контроль)	У хмарі Google	У хмарі Microsoft

1.4. Інтеграція освітніх інформаційних систем

Інтеграція освітньої інформаційної системи з ЄДЕБО (Єдиною державною електронною базою з питань освіти) – це процес автоматизації обміну даними, який усуває ручне введення інформації, мінімізує помилки та забезпечує відповідність українському законодавству.

Оскільки державна база має закритий контур безпеки, пряме підключення готового ПЗ (Moodle, Google, Teams) до неї неможливе. Інтеграція завжди реалізується через проміжний шлюз (API) або структурований імпорт/експорт файлів [10].

Спосіб 1. Напіваавтоматичний (Імпорт через XML/CSV файли)

Найпростіший метод, який підходить для будь-якого готового ПЗ без залучення програмістів.

- Як це працює: Адміністратор закладу з відповідним рівнем доступу заходить в особистий кабінет ЄДЕБО, формує вибірку студентів (наприклад, 1 курс, Спеціальність 122) та вивантажує файл у форматі .csv або .xml.

- Адаптація під ПЗ:

- Moodle: Використовується стандартний інструмент «Завантажити користувачів» (*Upload users*). Поля з файлу ЄДЕБО (ПІБ, ID, Email) зіставляються з полями профілю Moodle.

- Google Workspace: Використовується консоль адміністратора або безкоштовне розширення *Google Sheets* (наприклад, *Bulk User Creation*) для імпорту списку та автоматичного створення корпоративних акаунтів.

- MS Teams: Списки завантажуються через утиліту *School Data Sync (SDS)*, яка автоматично створює не лише акаунти, а й готові «Команди» для академічних груп.

Спосіб 2. Автоматичний (Пряма інтеграція через API та проміжний сервер)

Цей метод потребує розробки невеликого проміжного модуля (скрипта на Python/PHP), який пов'язує веб-сервіси ЄДЕБО з вашою системою (рис. 1.3).

1. Автентифікація: Проміжний сервер робить запит до веб-сервісів ЄДЕБО, використовуючи Електронний цифровий підпис (ЕЦП) або КЕП навчального закладу та захищений SSL/TLS канал.

2. Синхронізація даних: Скрипт за розкладом (наприклад, щоночі) запитує зміни в ЄДЕБО. Якщо з'явився наказ про зарахування нових студентів або відрахування, скрипт отримує ці дані.

3. Оновлення в LMS: Через вбудовані API платформ (Moodle REST API або Microsoft Graph API) скрипт автоматично створює облікові записи для новачків або блокує доступ відрахованим [11].

[Державна база ЄДЕБО]



| (Захищений XML/SOAP-запит з ЕЦП закладу)



[Проміжний API-скрипт] → (Валідація, шифрування, мапінг полів)



| (REST API / Webhooks)



[Ваша LMS (Moodle/Teams)]

Рисунок 1.3 – Адаптація під ПЗ

Які дані синхронізуються?

- Персональні дані: ПІБ, дата народження, стать, унікальний код студента в ЄДЕБО (для уникнення дублів).
- Академічний статус: Факультет, курс, спеціальність, освітня програма, форма навчання (бюджет/контракт, денна/заочна).
- Рух контингенту: Автоматичне переведення на наступний курс, академічні відпустки, відрахування.

Головні виклики та вимоги безпеки [12]

- Захист персональних даних (ЗПД): Оскільки система оперує паспортами та кодами студентів, сервер, де лежить проміжний скрипт і база даних LMS, повинен відповідати вимогам КСЗІ (Комплексна система захисту інформації) в Україні.

- Сувора структура даних: Будь-яка помилка в написанні прізвища або апострофа (наприклад, Д'як замість Дьяк або Д'як) призведе до розсинхронізації. Система повинна мати модуль валідації (очищення) даних перед імпортом.
- Обмеження доступу: Доступ до шлюзу ЄДЕБО повинні мати виключно довірені особи (ІТ-відділ або відділ кадрів/деканат) з чітко розмежованими правами.

1.5. Класифікація та аналіз сучасних систем управління інформаційними потоками (СУІП)

Освітні інформаційні системи поділяються на чотири основні класи, які разом формують єдине цифрове освітнє середовище (ЄЦОС, рис. 1.4) [13]:



Рисунок 1.4 – СУІП в навчальному процесі

1. SMS (Student Management Systems) / EMIS (Educational Management Information Systems)

Системи управління закладом освіти. Вони відповідають за адміністративні потоки даних: контингент студентів, кадри, розклад, рух наказів, облік успішності (електронні журнали) та фінанси.

- Приклади: ПС «Університет», ЄДЕБО (державний рівень в Україні), Деканат, «Єдина Школа», Нові Знання (для ЗЗСО).

- Переваги: Консолідація офіційних звітних даних, жорсткий контроль нормативних вимог, автоматизація видачі дипломів/документів.

- Недоліки: Низька гнучкість, застарілі інтерфейси, орієнтація на перевіряючі органи, а не на зручність користувача.

2. LMS (Learning Management Systems)

Системи управління безпосередньо навчальним процесом. Вони забезпечують доставку контенту до студента, збір виконаних завдань, автоматичне тестування та фіксацію цифрового сліду навчання [14].

- Приклади: Moodle, Google Classroom, Canvas, Blackboard, Open edX.

- Переваги: Гнучкість налаштування курсів, потужні інструменти оцінювання, доступність 24/7 з будь-яких пристроїв.

- Недоліки: Вимагають високої цифрової грамотності викладачів; Moodle потребує власних серверів та постійної технічної підтримки.

3. LCMS (Learning Content Management Systems)

Системи управління навчальним контентом. На відміну від LMS, вони орієнтовані на створення, зберігання та повторне використання навчальних матеріалів (текстів, відео, SCORM-пакетів) групами авторів.

- Приклади: Adobe Captivate Prime, Microsoft SharePoint, iSpring Learn.

- Переваги: Стандартизація навчальних матеріалів, високий захист авторського права, зручне розмежування прав редагування.

- Недоліки: Висока вартість ліцензій, складність інтеграції з базовими SMS-системами.

4. Системи контролю та академічної доброчесності (Proctoring & Anti-plagiarism)

Спеціалізовані системи, які інтегруються в LMS для верифікації інформаційних потоків під час екзаменаційних сесій та перевірки наукових робіт.

- Приклади: Unichек, StrikePlagiarism (антиплагіат); Examus, ProctorEdu (прокторинг).
- Переваги: Об'єктивність оцінювання, мінімізація списування, автоматичний аналіз поведінки студента на іспиті.
- Недоліки: Високе навантаження на інтернет-канал студента, етичні питання щодо збору біометричних даних (табл. 1.2).

Таблиця 1.2

Порівняльний аналіз систем за критеріями навчального процесу [15]

Критерій аналізу	SMS (Деканат)	LMS (Moodle/Canvas)	LCMS (Репозиторії)	Прокторинг / Антиплагіат
Основний інформаційний потік	Накази, розклад, підсумкові оцінки	Лекції, домашні завдання, поточні бали	Методичні матеріали, відеолекції	Відео/аудіо потік, текстові збіги
Ключовий користувач	Адміністрація, оператор ЕОМ	Викладач, Студент	Методист, Автор курсу	Екзаменатор, Експерт комісії
Рівень інтерактивності	Низький (переважно перегляд)	Високий (форуми, чати, тести)	Середній (структурування)	Автоматичний моніторинг
Вплив на якість освіти	Опосередкований (орг. питання)	Прямий (методика навчання)	Прямий (якість контенту)	Прямий (контроль знань)

Сучасні тенденції в автоматизації освітніх потоків (2026 рік) [16]

1. AI-асистенти викладача: Штучний інтелект бере на себе потік рутинної перевірки тестів, базових відповідей на форумах та генерацію індивідуальних траєкторій для студентів.
2. Перехід до LXP (Learning Experience Platforms): Поступове витіснення класичних LMS системами, орієнтованими на користувача, де

контент рекомендується на основі інтересів та прогалин у знаннях студента (аналог Netflix для навчання).

3. Наскрізна мобільність: Інформаційні потоки повністю адаптуються під смартфони (push-повідомлення про дедлайни, мобільний розклад, швидкі опитування).

Найбільша проблема сучасних українських закладів освіти – клаптева автоматизація, коли SMS (деканат) не знає, що відбувається в LMS (Moodle), а оцінки викладачі переносять вручну. Ефективна СУІП в освіті має базуватися на API-інтеграції: студент реєструється в SMS, автоматично отримує обліковий запис у LMS, бали за тести з LMS самі йдуть в електронний відомість, а дипломна робота автоматично відправляється на антиплагіат.

1.6. Об'єкти логістичного управління

Основними об'єктами управління в логістиці є матеріальний потік та пов'язаний з ним інформаційний потік в сферах постачання, виробництва та збуту. Окрім цих потоків в логістиці розглядається фінансовий потік та потік послуг, які є не менш важливими в логістичному управлінні. Поняття “матеріальний потік” є ключовим у логістиці.

Матеріальний потік – це продукція, яка розглядається в процесі застосування до неї різноманітних логістичних та (або) технологічних операцій і відноситься до певного часового інтервалу. Коли матеріальний потік розглядається не у часовому інтервалі, а в певний момент часу, він перетворюється у *матеріальний запас* (викликаний нерівномірностями постачання, виробництва, збуту та споживання) [17].

Виділення всіх операцій на шляху матеріального потоку дозволяє: розглянути загальний процес руху продукту, що змінюється, від джерел його

виникнення до кінцевого споживача; проектувати цей процес відповідно до потреб ринку.

Матеріальний потік має певні параметри, які його характеризують (табл. 1.3, [18]).

Таблиця 1.3

Параметри матеріального потоку

Параметр матеріального потоку	Визначення параметру
1. Потужність матеріального потоку	— обсяг матеріального потоку, що припадає на одиницю часу
2. Напруженість матеріального потоку	— час, необхідний для переміщення одиниці матеріалу
3. Початковий, проміжний і кінцевий пункти матеріального потоку	— відповідно відображають джерело, пункти здійснення впливу матеріальних цінностей та їх споживача
4. Траскторія шляху	— лінія, яка описує маршрут переміщення матеріальних цінностей в процесі руху до споживача
5. Довжина шляху	— відстань, яку проходять матеріальний потік від джерела до споживача
6. Швидкість матеріального потоку	— відстань, яку проходить матеріальний потік за одиницю часу
7. Час руху матеріального потоку	— часовий термін від початку руху матеріального потоку до його завершення
8. Величина запасів	— обсяг матеріального потоку, що було зупинено

Інформаційний потік може випереджати матеріальний, рухатися разом з ним та після нього. При цьому інформаційний потік може бути спрямований як в одну сторону, так і протилежну.

Управління інформаційним потоком можна наступним чином [19]:

- 1) змінюючи напрямок потоку;

2) обмежуючи швидкість передачі до відповідної швидкості прийому;

3) обмежуючи обсяг потоку до величини пропускної спроможності вузлу або ділянки шляху.

Вимірюється інформаційний потік кількістю оброблюваної чи переданої інформації за одиницю часу. Способи вимірювання кількості інформації, яка міститься в якомусь повідомленні, вивчаються кібернетикою (теорією інформації). Згідно цієї теорії одиницею вимірювання прийнята одиниця *біт* (та байт, який дорівнює 8 біт). Відповідно до цього інформаційний потік буде вимірюватися у біт за одиницю часу, або байтах за одиницю, або інших похідних одиницях.

Фінансовий потік – це спрямований рух фінансових засобів, які циркулюють в логістичній системі, а також між логістичною системою і зовнішнім середовищем і є необхідними для забезпечення ефективного руху певного товарного потоку. З цього визначення випливає: *логістичний фінансовий потік* – це не просто рух фінансових ресурсів, а їх спрямований рух; спрямованість руху фінансових ресурсів в логістиці обумовлюється необхідністю забезпечення переміщення відповідного товарного потоку [20].

Фінансові ресурси (і відповідно фінансові потоки) є унікальним об'єктом управління логістичної системи, що відображають зворотню сторону руху матеріальних цінностей, інформації та надання логістичних послуг і, як результат, фінансові потоки є нерозривними із матеріальними, інформаційними та сервісними потоками.

З метою оптимізації фінансових потоків та відповідного забезпечення руху матеріальних та інших потоків проводять їх класифікацію за багатьма ознаками (рис. 2.3).

В процесі управління фінансовими потоками виділяють певні етапи:

1) облік руху фінансових потоків;

2) аналіз фінансових потоків на основі даних обліку;

- 3) планування фінансових потоків;
- 4) контроль фінансових потоків.

Ефективність фінансового забезпечення підприємства виражається в підвищенні швидкості руху матеріального і товарного потоку. В прискоренні цього руху важливу роль відіграють способи грошових розрахунків і взаєморозрахунків. Форми, види і порядок фінансових розрахунків по закупці ресурсів та реалізації готової продукції регулюються договірними відносинами. На вибір форм і видів розрахунків впливають різні фактори (порядок документообігу, вид платіжного доручення, швидкість обігу грошових коштів, трудомісткість різних фінансових операцій тощо).

Ще одним не менш важливим потоком логістичної системи є потік логістичних послуг. *Потік послуг* — потік дій, що здійснює учасник логістичного процесу по відношенню до інших учасників з метою принесення користі останнім. Цей потік є специфічним та включає:

- 1) надання послуг кінцевим споживачам;
- 2) надання послуг учасникам логістичного ланцюжку;
- 3) надання внутрішньофірмових послуг.

Від того, як компанія обслуговує своїх партнерів в значній мірі буде залежати ефективність руху логістичних потоків і в кінцевому підсумку рівень обслуговування споживачів. При цьому рівень сервісу може бути базовим, оптимальним та повним (рис. 1.5).

Логістична операція — це будь-яка дія, що не підлягає подальшій декомпозиції, направлена на перетворення матеріальних і супроводжуючих його потоків (завантаження, розвантаження, маркування, збір інформації, передача інформації тощо). До логістичних операцій відносять навантаження, транспортування, розвантаження, укладання на збереження й відбір з місць зберігання, пакування, формування вантажного пакету, а також інші операції, які здійснюються в процесі руху матеріальних цінностей [21].



Рисунок 1.5 – Різновиди фінансових потоків

Всю сукупність логістичних операцій можна поділити на дві групи: *технологічні операції* за допомогою яких виробляються матеріальні цінності та *логістичні операції*, які забезпечують всім необхідним в достатній кількості, в потрібному місці та часі технологічні операції.

Логістичні операції поділяються також на зовнішні та внутрішні по відношенню до організації (підприємства, компанії).

Логістична функція – сукупність логістичних операцій, спрямованих на досягнення поставлених перед логістичною системою та її елементами цілей. Всі логістичні функції поділяються на *базові* та *забезпечуючі*. До базових функцій відносять постачання, виробництво, збут (табл. 1.4). До забезпечуючих – обслуговування споживачів, транспортування, управління виробничими операціями, управління запасами, інформаційне забезпечення, ціноутворення та інші [22].

Базові логістичні функції та операції

Логістична функція	Основні логістичні операції
1. Збут	Координація з планом маркетингу, прогнозування попиту, сервіс, оперативно-календарне планування транспортування готової продукції, оброблення замовлень клієнтури, складування готової продукції, завантажувально-розвантажувальні і транспортно-складські роботи з готовою продукцією, постачання готової продукції, облік запасів готової продукції
2. Виробництво	Координація з планом фізичного розподілу, оперативно-календарне планування переміщення незавершеного виробництва, внутрішньозаводські переміщення матеріалів, завантажувально-розвантажувальні і транспортно-складські роботи з незавершеним виробництвом, оперативне забезпечення виробничих підрозділів сировиною, матеріалами, напівфабрикатами, комплектуючими виробами, складування незавершеного виробництва, облік незавершеного виробництва
3. Постачання	Координація з оперативно-календарним планом виробництва, вибір і проведення переговорів з постачальниками, планування потреб у матеріалах, складання оперативно-календарного плану постачання, транспортування сировини, матеріалів, напівфабрикатів, комплектуючих виробів, складування виробничих запасів, завантажувально-розвантажувальні й транспортно-складські роботи з предметами постачання.

Одними з важливих об'єктів логістики є логістичний канал та логістичний ланцюжок. Ці поняття є взаємопов'язаними.

Логістичний канал – множина всіх можливих учасників процесу руху логістичних потоків до кінцевого споживача, та визначає характер та

структуру руху до споживача. *Логістичний ланцюжок* – упорядкована множина фізичних та (або) юридичних осіб, які здійснюють логістичні операції, пов’язаних доведенням потоків до споживача (рис. 1.6). Якщо логістичний канал є напрямом дій, то логістичний ланцюжок є більш конкретним складом учасників цих дій.

Формування логістичного ланцюга відбувається за взаємовигідного партнерства та врахування двох принципів: конкуренції та кооперації. В логістичному ланцюжку знаходяться як основні (постачання, виробництво, торгівля) так і спеціалізовані учасники (транспортні, рекламні, інформаційні та інші компанії).

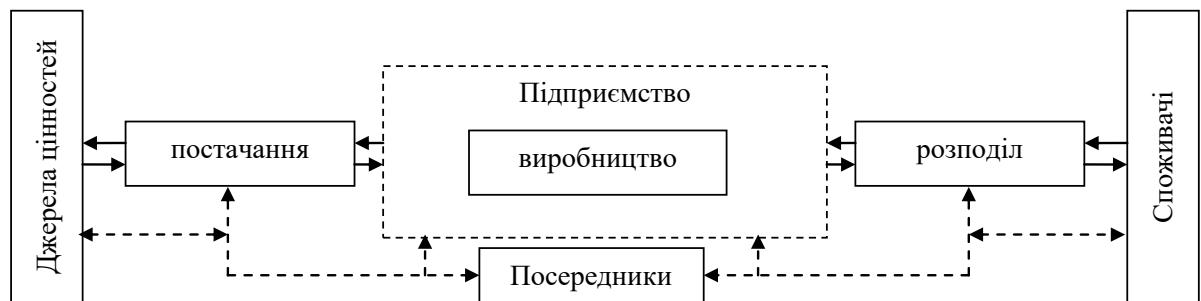


Рисунок 1.6 – Логістичний ланцюжок

Сьогодні конкурентними являються такі логістичні системи, які крім виконання семи основних правил логістики (необхідний споживачу продукт, його певна кількість і відповідна якість, потрібне місце і точний час, конкретний споживач, оптимальні затрати) забезпечують кращий сервіс, знижують ризик, підвищують безпеку, не наносять збитків природному середовищу. Досягнути таких показників можливо тільки при включенні до логістичного ланцюжка необхідних ланок — сервісного обслуговування, утилізації або переробки відходів, забезпечення безпеки при функціонуванні.

Логістичні інтегровані потоки є похідними від розвитку концепції інтегрованої логістики, яка базується на сучасному принципі інтеграції. Використання цієї концепції є зумовленим багатьма факторами такими як: зростання швидкості зміни попиту, зменшується життєвий цикл товарів та

іншими, що призводить до невизначеності діяльності підприємств. Це вимагає інтеграцію та координацію дій між партнерами.

Ідея інтеграції витікає з сутності економічного потоку. *Економічний потік* – власні або залучені ресурси компанії, що розглядаються в процесі взаємопов'язаних та взаємообумовлених змін та переміщень, які здійснюються з метою досягнення цілей компанії. Економічний потік вимагає узгоджених дій з оптимізації всіх учасників руху цього потоку. Максимальний ефект досягається інтеграцією потоків як внутрішніх, так і зовнішніх в межах єдиної системи (рис. 1.7).

Інтеграція потоків у логістиці відбувається за трьома напрямками:

- 1) координація руху потоків на різних стадіях їх просування до кінцевого споживача;
- 2) інтеграція потоків різних видів;
- 3) інтеграція технологій, які використовують учасники логістичного процесу.

Ефективність інтеграції потоків досягається за допомогою:

- 1) інформаційна відкритість партнерів;
- 2) чітке виконання вимог, що вимагаються ланками логістичного ланцюгу від попередньої ланки;
- 3) координація планування спільної роботи;
- 4) розробка документів, що уніфікують вимоги до результатів роботи.

Перепонами інтеграції потоків є: підвищення взаємозалежності фірм, боязнь втрати самостійності, втрата контролю над рухом потоків, який передається партнерам та інші, що не дозволяють сповна використати переваги інтеграції [21].

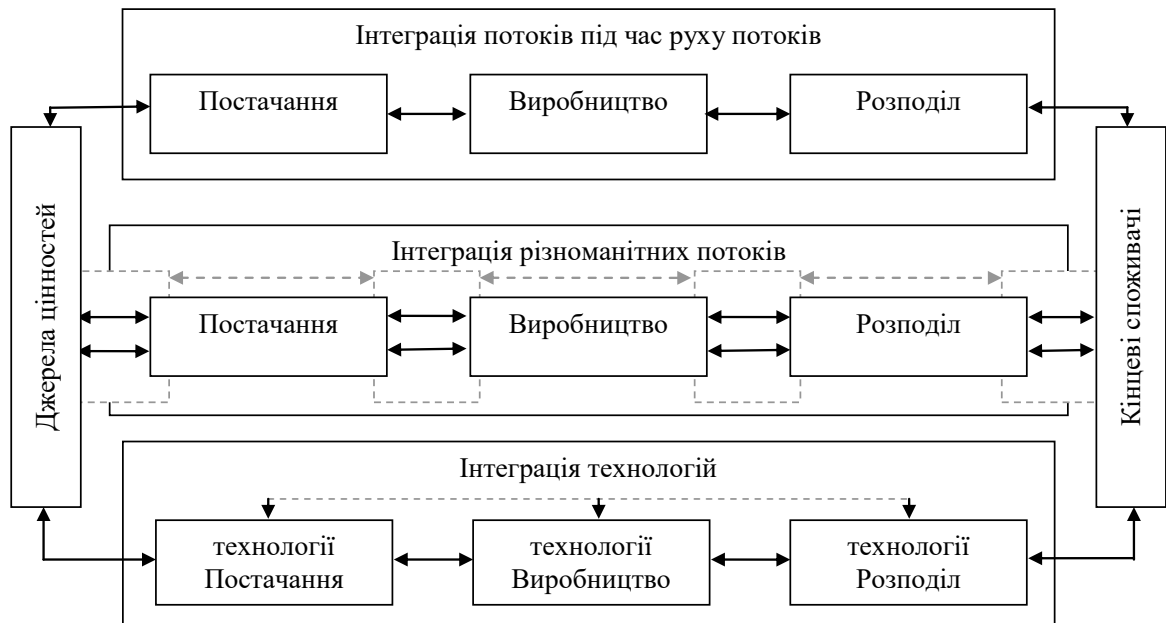


Рисунок 1.7 – Різновиди інтеграції логістичних потоків

Висновки по розділу 1

1. В результаті аналізу предметної області вивчено та проведено порівняння програм-аналогів комп'ютерних систем управління інформаційними потоками, що дозволило узагальнити та систематизувати матеріал для розроблюваної системи.

2. Проведений аналіз дозволив вибрати основний напрямок кваліфікаційної роботи – розробка автоматизованої системи управління інформаційними поокми у навчальному процесі, класифікувати систему, виокремити основні вимоги до системи і визначити її основні властивості.

РОЗДІЛ 2. РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ НА КОМП'ЮТЕРНУ СИСТЕМУ

1. Загальні відомості

- **Назва системи:** Комп'ютерна система управління інформаційними потоками навчального закладу (КСУІП).
- **Призначення:** Автоматизація збору, обробки, збереження та розподілу даних між учасниками освітнього процесу.
- **Користувачі:** Адміністрація, викладачі, студенти/учні, батьки, бухгалтерія, відділ кадрів.

2. Призначення та цілі створення системи

- **Мета:** Створення єдиного інформаційного простору для підвищення ефективності управління закладом.
- **Задачі:**
 - Зменшення паперового документообігу.
 - Прискорення обміну даними між підрозділами.
 - Забезпечення прозорості оцінювання та відвідуваності.
 - Автоматизація формування звітності для Міністерства освіти.

3. Вимоги до системи

3.1. Функціональні вимоги (Модулі системи)

- **Модуль «Деканат / Адміністрація»:**
 - Управління закладами, курсами, групами.
 - Ведення особових справ студентів та працівників.
 - Генерація наказів, дипломів, довідок.
- **Модуль «Розклад та планування»:**
 - Автоматичне та ручне створення розкладу занять/іспитів.

- Облік аудиторного фонду та його завантаженості.
- **Модуль «Електронний журнал та щоденник»:**
 - Виставлення оцінок, Н-ок (відвідуваність), коментарів до уроків.
 - Моніторинг успішності (графіки, рейтинги).
 - Доступ до домашніх завдань та навчальних матеріалів.
- **Модуль «Приймальна комісія»:**
 - Реєстрація абітурієнтів та верифікація сертифікатів.
 - Автоматичний розрахунок конкурсного бала та формування рейтингів.
- **Модуль «Фінанси та кадри»:**
 - Облік оплати за навчання/гуртожиток.
 - Розрахунок педагогічного навантаження викладачів.

3.2. Специфікація пріоритетного модуля «Навчальний модуль студентів та викладачів»

Цей модуль є ядром системи та ділиться на два персоналізовані кабінети з відповідним функціоналом.

3.2.1. Особистий кабінет викладача (Веб-інтерфейс)

- **Управління навантаженням:** Відображення затвердженого плану годин (лекції, практики, лабораторні) на семестр/рік.
- **Електронний журнал:**
 - Виставлення поточних балів, оцінок за модулі та іспити/заліки відповідно до ECTS (100-бальна система).
 - Фіксація відвідуваності («Н-ки») з можливістю зазначення причини (поважна/неповажна).
- **Методичний репозиторій:** Завантаження силабусів, лекційних матеріалів, варіантів контрольних робіт та тестових завдань для студентів.
- **Комунікація:** Створення оголошень для конкретних потоків, груп або окремих студентів.

3.2.2. Особистий кабінет студента (Веб та Мобільний додаток)

- **Електронна залікова книжка:** Перегляд поточних балів, підсумкових оцінок за семестри, академічного рейтингу та заборгованостей.
- **Цифровий розклад:** Інтерактивний розклад занять з фільтрацією по днях, дисциплінах, викладачах та посиланнями на онлайн-лекції (Zoom/Teams).
- **Модуль вибіркових дисциплін (Важливе для ЗВО):**
 - Інтерфейс для вибору студентом предметів вільного вибору (дисципліни).
 - Автоматичний контроль лімітів місць на курсах та дедлайнів голосування.
- **Замовлення довідок:** Формування онлайн-запитів до деканату на отримання довідок про навчання (з QR-кодом для перевірки).

Вимоги до інтеграції з державними реєстрами (ЄДЕБО)

Синхронізація з ЄДЕБО (Єдина державна електронна база з питань освіти) має відбуватися через офіційний API-інтерфейс (вебсервіси SOAP/REST) та задовольняти такі вимоги:

- **Імпорт даних (Первинне наповнення):**
 - Автоматичне завантаження списків зарахованих абітурієнтів (накази про зарахування).
 - Імпорт персональних даних студентів (ПІБ, паспортні дані, РНОКПП, пільги).
- **Експорт та синхронізація (Поточна діяльність):**
 - Передача даних про рух контингенту (накази про переведення, відрахування, академічні відпустки).
 - Синхронізація статусів студентів (активний, відрахований, випускник) для формування картки «ЗВО-Студент».
- **Дипломний модуль:**

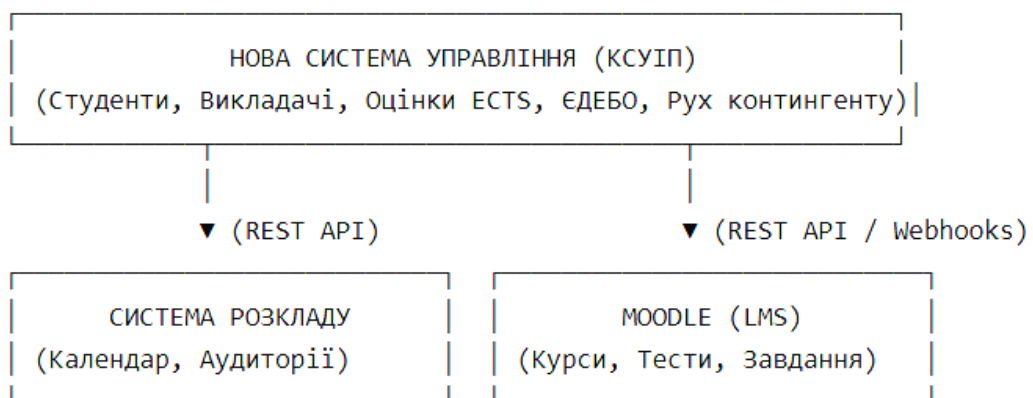
- Експорт даних випусників для генерації та реєстрації дипломів державного зразка та додатків до них (Diploma Supplement).
- Автоматичне отримання та збереження номерів і серій виданих документів.
- **Безпека з'єднання:** Використання захищених каналів зв'язку (VPN) та обов'язкове підписання пакетів даних КЕП (Кваліфікованим електронним підписом) відповідальної особи ЗВО.

Технологічний стек та архітектура

- **Архітектура:** Мікросервісна (модуль ЄДЕБО та Навчальний модуль мають працювати як окремі сервіси, щоб збій одного не зупиняв інший).
- **База даних:** PostgreSQL або MySQL з підтримкою реляційних зв'язків для складної структури (Факультет -> Кафедра -> Група -> Студент).

Інтеграція з наявними системами ЗВО

Зв'язок між КСУІП та поточними сервісами має здійснюватися через захищений API (REST або GraphQL) за наступною логікою:



Інтеграція з LMS Moodle

- **Авторизація (SSO):** Впровадження єдиної точки входу (наприклад, через OAuth2 або SAML). Студент і викладач переходять із кабінету КСУП до свого Moodle без повторного введення пароля.

- **Синхронізація контингенту:** При зарахуванні студента в КСУП або переведенні в іншу групу, система автоматично створює/редагує його обліковий запис у Moodle та записує на відповідні електронні курси.

- **Синхронізація оцінок:** Автоматичний експорт підсумкових оцінок за модулі/семестри з журналу Moodle до залікової відомості КСУП для формування підсумкового рейтингу.

Інтеграція з системою розкладу

- **Імпорт даних:** КСУП раз на добу (або за тригером) витягує актуальний розклад для кожної групи/викладача.

- **Відображення:** У кабінетах студентів та викладачів розклад відображається у зручному інтерактивному календарі з прямими лінками на відповідні курси в Moodle або Zoom/Teams.

Вимоги до продуктивності та навантаження

- **Пропускна здатність:** Оскільки загальна кількість користувачів вимірюється тисячами, система повинна стабільно витримувати:

- *Штатний режим:* до 500–1000 одночасних користувачів (Concurrent Users) протягом дня.

- *Піковий режим (Сесія, вибір дисциплін):* до 3000+ одночасних користувачів із часом відповіді сторінки (TTFB) не більше 1.5 секунди.

- **Оптимізація запитів:** Обов'язкове використання кешування (Redis/Memcached) для важких та частих запитів (наприклад, виведення розкладу чи залікової книжки).

- **База даних:** Використання індексації для таблиць користувачів, оцінок та журналів для уникнення "зависань" під час генерації великих звітів чи відомостей для деканату.

4. Архітектурні рекомендації для розробників

1. **Розділення шарів (Decoupling):** Нова система має виступати "майстер-базою" для персональних даних студентів. Moodle та система розкладу стають "споживачами" (Consumers) цих даних.

2. **Черги повідомлень (Message Queues):** Для обміну даними з ЄДЕБО та Moodle використовувати черги (наприклад, RabbitMQ). Це завадить падінню системи, якщо сервер ЄДЕБО тимчасово не відповідає – запит просто зачекає у черзі.

5. Нефункціональні вимоги

- **Продуктивність:** Підтримка до 5000 одночасних сесій без затримок відповіді системи (час відгуку до 2 секунд).
- **Доступність:** Адаптивний веб-інтерфейс (ПК, планшети, смартфони) та мобільний додаток (iOS/Android).
- **Безпека:**
 - Рольова модель доступу (РМД) – кожен бачить лише свій рівень даних.
 - Шифрування даних (SSL/TLS, шифрування баз даних з персональними даними).
 - Двофакторна автентифікація (2FA) для адміністраторів.
- **Надійність:** Резервне копіювання даних (Бэкап) кожні 24 години із глибиною зберігання 30 днів.

6. Склад та зміст робіт із розробки

1. **Аналітика:** Збір вимог закладу, опис бізнес-процесів (2 тижні).
2. **Проектування:** Створення архітектури баз даних, макетів інтерфейсу (UI/UX) (3 тижні).
3. **Розробка:** Програмування серверної (Backend) та клієнтської (Frontend) частин (12 тижнів).

4. **Тестування:** Функціональне, навантажувальне та безпекове тестування (3 тижні).

5. **Впровадження:** Навчання персоналу, перенесення наявних баз даних, запуск (2 тижні).

7. Порядок контролю та приймання

- Приймання системи здійснюється поетапно згідно з календарем робіт.
- Фінальне приймання проводиться на основі успішного проходження програми та дій тест-кейсів (тестових сценаріїв).

Висновки по розділу 2

На підставі завдання на кваліфікаційну роботу було сформовано технічне завдання на розробку комп'ютерної системи управління інформаційними потоками у навчальному процесі.

РОЗДІЛ 3. ЗАГАЛЬНА КОНЦЕПЦІЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ІНФОРМАЦІЙНИМИ ПОТОКАМИ

3.1 Архітектура модуля «Успішність студентів»

1. Загальний архітектурний стиль та системний контекст

Модуль «Успішність студентів» розроблено як ізольований сервіс у рамках мікросервісної архітектури університету. Це забезпечує незалежне масштабування під час пікових навантажень (екзаменаційні сесії) та стабільність інших модулів.

Ключові компоненти системи:

- Identity Provider (SSO): централізована система автентифікації університету (Google Workspace / Azure AD). Корпоративна пошта (email) є наскрізним ідентифікатором користувача в усіх системах.
- API Gateway: єдина точка входу, яка маршрутизує запити від клієнтських додатків (Викладач, Деканат, Студент), перевіряє JWT-токени та обмежує частоту запитів (Rate Limiting).
- LMS Integration Gateway: спеціалізований сервіс-прошарок для асинхронного приймання даних від зовнішніх платформ дистанційного навчання.
- Core Gradebook Service: бізнес-логіка модуля, що забезпечує калькуляцію балів, збереження даних та ведення відомостей.

2. Схема інформаційних потоків (Data Flow)

Інформаційні потоки поділяються на три основні сценарії, які наведено на рис. 3.1.

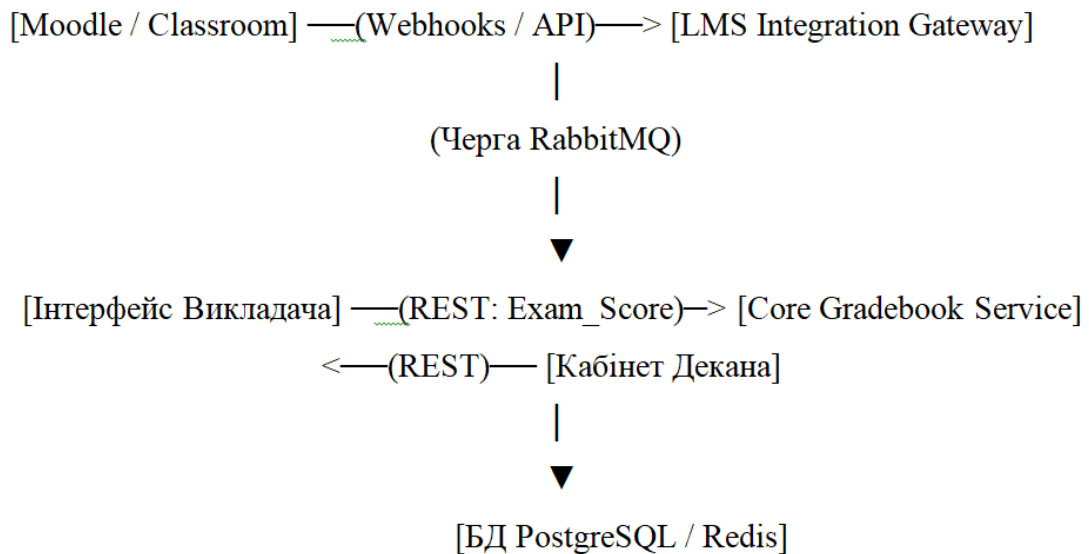


Рисунок 3.1 – Три основних сценарії інформаційних потоків

1. Потік семестрових балів (Автоматичний): LMS $\rightarrow$ Integration Gateway $\rightarrow$ Message Broker (RabbitMQ) $\rightarrow$ Gradebook Service $\rightarrow$ СУБД.
2. Потік екзаменаційних балів (Ручний): Інтерфейс викладача $\rightarrow$ API Gateway $\rightarrow$ Gradebook Service (Валідація) $\rightarrow$ СУБД.
3. Потік звітності (Аналітичний): СУБД $\rightarrow$ Redis Cache $\rightarrow$ Інтерфейс деканату (Генерація відомостей).

3. Модель даних та DDL (PostgreSQL)

Схема бази даних оптимізована для реляційної цілісності та швидкого формування звітів. Підсумковий бал (`total_score`) обчислюється автоматично на рівні ядра СУБД за формулою: Семестр (max 60) + Екзамен (max 40) = Разом (max 100).

sql

```

CREATE EXTENSION IF NOT EXISTS "uuid-oss";
CREATE TYPE lms_provider AS ENUM ('MOODLE', 'GOOGLE_CLASSROOM');

```

```

CREATE TYPE sync_state AS ENUM ('SYNCHRONIZED', 'FAILED',
'MANUALLY_OVERRIDDEN');

CREATE TYPE statement_status AS ENUM ('OPEN', 'MUTED', 'SIGNED');

CREATE TABLE academic_groups (
    group_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    group_name VARCHAR(50) NOT NULL UNIQUE,
    department VARCHAR(100) NOT NULL,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE students (
    student_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    email VARCHAR(150) NOT NULL UNIQUE,
    full_name VARCHAR(150) NOT NULL,
    group_id UUID REFERENCES academic_groups(group_id) ON DELETE
RESTRICT,
    is_active BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE lecturers (
    lecturer_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    email VARCHAR(150) NOT NULL UNIQUE,
    full_name VARCHAR(150) NOT NULL,
    department VARCHAR(100) NOT NULL,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE subjects (
    subject_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    subject_name VARCHAR(150) NOT NULL,
    ects_credits NUMERIC(4, 2) NOT NULL,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE lms_course_mappings (
    mapping_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),

```

```

        subject_id UUID NOT NULL REFERENCES subjects(subject_id) ON DELETE
CASCADE,
        group_id UUID NOT NULL REFERENCES academic_groups(group_id) ON
DELETE CASCADE,
        lms_type lms_provider NOT NULL,
        lms_course_id VARCHAR(100) NOT NULL,
        updated_at      TIMESTAMP      WITH      TIME      ZONE      DEFAULT
CURRENT_TIMESTAMP,
        CONSTRAINT unique_lms_course UNIQUE (lms_type, lms_course_id)
);
CREATE TABLE academic_statements (
    statement_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    subject_id UUID NOT NULL REFERENCES subjects(subject_id) ON DELETE
RESTRICT,
    group_id UUID NOT NULL REFERENCES academic_groups(group_id) ON
DELETE RESTRICT,
    lecturer_id UUID NOT NULL REFERENCES lecturers(lecturer_id) ON DELETE
RESTRICT,
    semester VARCHAR(20) NOT NULL,
    status statement_status DEFAULT 'OPEN',
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    signed_at TIMESTAMP WITH TIME ZONE,
    CONSTRAINT unique_statement_per_semester UNIQUE (subject_id, group_id,
semester)
);
CREATE TABLE gradebook (
    grade_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    statement_id UUID NOT NULL REFERENCES academic_statements(statement_id)
ON DELETE CASCADE,
    student_id UUID NOT NULL REFERENCES students(student_id) ON DELETE
RESTRICT,
    term_score NUMERIC(5, 2) DEFAULT 0.00 CHECK (term_score >= 0.00 AND
term_score <= 60.00),

```



```

        exam_score NUMERIC(5, 2) DEFAULT NULL CHECK (exam_score IS NULL OR
(exam_score >= 0.00 AND exam_score <= 40.00)),
        total_score NUMERIC(5, 2) GENERATED ALWAYS AS (COALESCE(term_score,
0.00) + COALESCE(exam_score, 0.00)) STORED,
        sync_status sync_state DEFAULT 'SYNCHRONIZED',
        last_lms_sync TIMESTAMP WITH TIME ZONE,
        updated_at      TIMESTAMP      WITH      TIME      ZONE      DEFAULT
CURRENT_TIMESTAMP,
        CONSTRAINT unique_student_per_statement UNIQUE (statement_id, student_id)
    );
CREATE TABLE grade_audit_logs (
    log_id BIGSERIAL PRIMARY KEY,
    grade_id UUID NOT NULL,
    changed_by_email VARCHAR(150) NOT NULL,
    old_term_score NUMERIC(5, 2),
    new_term_score NUMERIC(5, 2),
    old_exam_score NUMERIC(5, 2),
    new_exam_score NUMERIC(5, 2),
    action_timestamp      TIMESTAMP      WITH      TIME      ZONE      DEFAULT
CURRENT_TIMESTAMP,
    reason TEXT
);
CREATE INDEX idx_students_email ON students(email);
CREATE INDEX idx_lms_mappings ON lms_course_mappings(lms_type,
lms_course_id);
CREATE INDEX idx_gradebook_statement ON gradebook(statement_id);

```

4. Специфікація REST API Ендпоінтів

Всі ендпоінти використовують префікс /api/v1 та вимагають авторизаційний заголовок Authorization: Bearer <JWT> (табл. 3.1).

Таблиця 3.1

Специфікація REST API Ендпоінтів

Метод	Ендпоінт	Роль	Опис
POST	/integration/lms-webhook	LMS Service	Приймання оцінок з Moodle/Classroom.
GET	/lecturer/subjects	Викладач	Список дисциплін викладача у семестрі.
GET	/lecturer/statements/{id}/students	Викладач	Отримання списку групи з term_score (max 60).
PUT	/lecturer/statements/{id}/exams	Викладач	Внесення балів за екзамен (max 40) або коригування.
POST	/lecturer/statements/{id}/sign	Викладач	Затвердження відомості та блокування змін.
POST	/dean/statements	Деканат	Створення/відкриття нової відомості.
GET	/dean/statements/{id}/export	Деканат	Запит розгорнутої відомості з ECTS оцінками.
POST	/dean/statements/{id}/unlock	Деканат	Форс-мажорне розблокування підписаної відомості.
GET	/student/dashboard/grades	Студент	Перегляд власних балів (доступ лише до свого ID).

5. Алгоритм роботи Math Engine (Масштабування 0–60)

Компонент Math Engine автоматично обчислює term_score на основі даних з LMS, використовуючи паттерн «Пріоритет ручного введення».

python

```

FUNCTION calculate_term_score(student_id, subject_id, m_course_id):
    grade_record = DB.fetch_gradebook_record(student_id, subject_id)
    # Захист від затирання ручної оцінки викладача автоматикою
    IF grade_record.sync_status == "MANUALLY_OVERRIDDEN":
        RETURN grade_record.term_score
    lms_grade_items = LMS_Gateway.get_student_grades(student_id, m_course_id)
    total_max_possible_points = 0.0

```

```

total_earned_points = 0.0
FOR item IN lms_grade_items:
    IF item.is_extra == TRUE:
        CONTINUE
    current_earned = item.earned_points IF item.earned_points != NULL ELSE
0.0
    total_max_possible_points += item.max_points
    total_earned_points += current_earned
# Розрахунок пропорції та масштабування до 60 балів
IF total_max_possible_points == 0.0:
    calculated_term_score = 0.0
ELSE:
    raw_ratio = total_earned_points / total_max_possible_points
    calculated_term_score = ROUND(raw_ratio * 60.0, 2)
# Валідація меж (Guard Rails)
calculated_term_score = MAX(0.0, MIN(calculated_term_score, 60.0))
DB.update_student_term_score(student_id,    subject_id,    calculated_term_score,
"SYNCHRONIZED")
RETURN calculated_term_score

```

6. Логіка конвертації та формування звітів (ECTS)

Трансформація підсумкового балу (total_score) у шкалу ECTS та національну оцінку виконується динамічно на рівні прикладного коду (Backend) або через SQL View під час вивантаження відомості для деканату (табл. 3.2).

Таблиця 3.2

Матриця відповідності

Бал 100	ECTS	Національна (Іспит)	Національна (Залік)
90 - 100	A	Відмінно	Заховано
82 - 89	B	Добре	Заховано
74 - 81	C	Добре	Заховано

64 - 73	D	Задовільно	Заховано
60 - 63	E	Задовільно	Заховано
35 - 59	FX	Незадовільно	Не зараховано
0 - 34	F	Незадовільно	Не зараховано

Реалізація генерації звіту через PostgreSQL View:

sql

```
CREATE OR REPLACE VIEW v_academic_reports AS
```

```
SELECT
```

```
    s.statement_id,
```

```
    g.group_name,
```

```
    sub.subject_name,
```

```
    st.full_name AS student_name,
```

```
    gb.term_score,
```

```
    gb.exam_score,
```

```
    gb.total_score,
```

```
    CASE
```

```
        WHEN gb.total_score >= 90 THEN 'A'
```

```
        WHEN gb.total_score >= 82 THEN 'B'
```

```
        WHEN gb.total_score >= 74 THEN 'C'
```

```
        WHEN gb.total_score >= 64 THEN 'D'
```

```
        WHEN gb.total_score >= 60 THEN 'E'
```

```
        WHEN gb.total_score >= 35 THEN 'FX'
```

```
        ELSE 'F'
```

```
    END AS ects_grade,
```

```
    CASE
```

```
        WHEN gb.total_score >= 90 THEN 'Відмінно'
```

```
        WHEN gb.total_score >= 74 THEN 'Добре'
```

```
        WHEN gb.total_score >= 60 THEN 'Задовільно'
```

```
        ELSE 'Незадовільно'
```

```
    END AS national_grade
```

```
FROM gradebook gb
```

```
JOIN academic_statements s ON gb.statement_id = s.statement_id
```

```
JOIN students st ON gb.student_id = st.student_id
```

```
JOIN academic_groups g ON s.group_id = g.group_id  
JOIN subjects sub ON s.subject_id = sub.subject_id;
```

7. Рівні доступу (RBAC) та безпека

1. Роль Student: Авторизація через SSO. Доступ GET /student/* тільки до власного student_id. Повне логування спроб доступу до чужих ID.
2. Роль Lecturer: Доступ на модифікацію балів PUT /lecturer/* лише для тих statement_id, де lecturer_id збігається з його токеном авторизації. Після зміни статусу відомості на SIGNED, операції PUT блокуються на рівні API.
3. Роль Dean: Повний доступ до аналітики та створення відомостей. Право використання ендпоінту /unlock для відміну фіксації оцінок. Кожне розблокування супроводжується обов'язковим заповненням поля reason (причина) в системному лозі.

3.2 Структурна схема системи управління успішністю

Нижче наведено текстову архітектурну діаграму, яка чітко відображає рівні доступу користувачів, сервісну логіку, систему інтеграції з LMS та рівень збереження даних.

Структурна схема системи управління успішністю приведена на рис. 3.2.

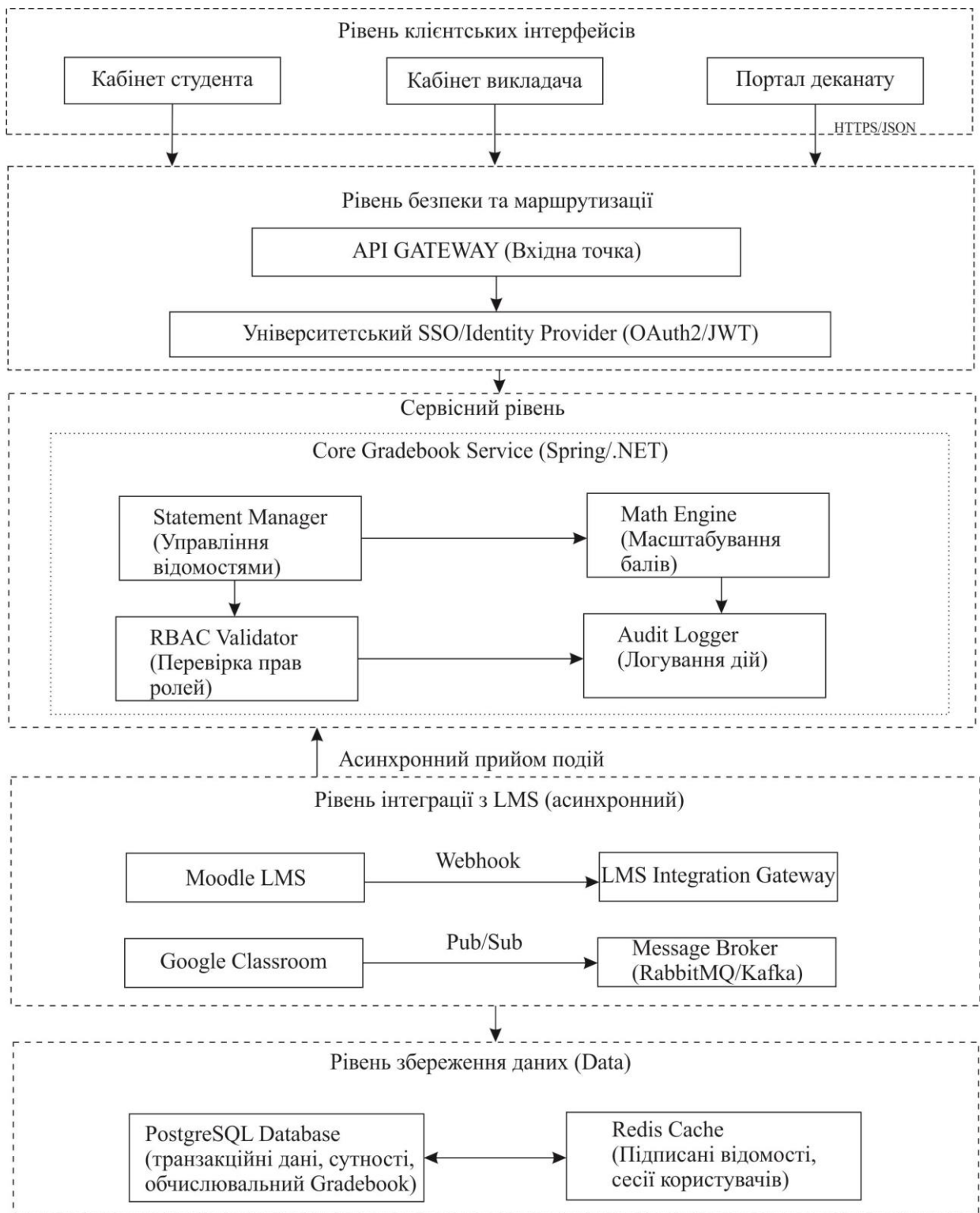


Рисунок 3.2 – Структурна схема системи управління успішністю

Опис структурних блоків та зв'язків

1. Клієнтський рівень: Три окремі веб-додатки (SPA на React/Angular), адаптовані під специфічні Use Cases кожної ролі користувача. Вони не мають прямого доступу до баз даних і спілкуються з сервером виключно через REST API.

2. Рівень безпеки (API Gateway + SSO): Диспетчер запитів. Коли користувач робить запит, Gateway звертається до SSO для валідації його JWT-токена та перевірки наявності потрібної ролі (STUDENT, LECTURER або DEAN). Якщо токен валідний, запит пропускається глибше.

3. Сервісний рівень (Core Service): «Серце» системи.

- *Statement Manager* відповідає за життєвий цикл відомостей (відкриття, заморожування, підписання).

- *Math Engine* бере на себе всю математику (перерахунок відсотків з LMS у бали до 60).

- *RBAC Validator* гарантує, що викладач може редагувати оцінки лише *своїх* студентів.

- *Audit Logger* пише кожен крок у захищену таблицю історії.

4. Рівень інтеграції (LMS Gateway): Ізольований буфер. Вебхуки від Moodle або Google Classroom потрапляють спочатку сюди, складаються в чергу повідомлень (RabbitMQ), а потім порційно обробляються основним сервісом. Це гарантує, що якщо одночасно 5000 студентів здадуть тест у Moodle, університетська база даних не «впаде» від навантаження.

5. Рівень даних: Гібридне сховище. PostgreSQL виступає надійним джерелом істини (ACID-транзакції, зв'язки сутностей). Redis використовується як швидка пам'ять для кешування готових звітів деканату та сесій, щоб не навантажувати PostgreSQL повторними важкими запитами.

3.3 Функціональні вимоги до системи

Для забезпечення стабільної, безпечної та ефективної роботи модуля «Успішність студентів» у періоди пікових навантажень (сесія, підбиття підсумків), система має відповідати чітким нефункціональним вимогам (NFR).

Нижче наведено перелік вимог, розподілених за ключовими категоріями: продуктивність, навантаження, безпека, надійність та архітектурна складність.

1. Продуктивність та масштабованість (Performance & Scalability)

- Час відгуку API (Latency):
 - Для звичайних операцій читання/запису (наприклад, введення оцінки викладачем) – не більше 200 мс для 95% запитів (p95).
 - Для важких аналітичних запитів деканату (генерація зведеної відомості по факультету) – не більше 2 секунд (завдяки обов'язковому кешуванню в Redis).
- Конкурентне навантаження (Concurrency):
 - Система повинна стабільно витримувати базове навантаження у 5 000 одночасних активних сесій користувачів.
 - У період сесії (пікове навантаження) система має масштабуватися до 20 000 одночасних користувачів без деградації продуктивності.
- Пропускна здатність інтеграції (Throughput):
 - LMS Integration Gateway та черга RabbitMQ повинні без затримок обробляти потік до 500 вебхуків/секунду від Moodle/Google Classroom (актуально, коли масово завершується дедлайн комп'ютерного тестування).

2. Безпека та захист даних (Security & Privacy)

- Автентифікація та авторизація:

- Доступ до API виключно за короткочасними токенами JWT (Access Token – 15 хвилин, Refresh Token – 7 днів), випущеними університетським SSO.
- Обов'язкова валідація ролей (RBAC) на рівні API Gateway та методів Backend-сервісу.
- Шифрування даних:
 - Усі дані в транспорті повинні передаватися виключно за протоколом HTTPS (TLS 1.3).
 - Персональні дані студентів (ПІБ, email) та архівні відомості в базі даних PostgreSQL мають бути зашифровані в стані спокою (Encryption at Rest) за стандартом AES-256.
- Захист від атак та збоїв:
 - На рівні API Gateway має бути налаштований Rate Limiting (наприклад, максимум 60 запитів за хвилину з однієї IP-адреси для студентів) для захисту від DDoS та перебору API.
 - Жодна помилка Backend-системи не повинна повертати на Frontend системний Stack Trace (захист від витоку інформації про архітектуру коду).

3. Надійність та доступність (Reliability & Availability)

- Коефіцієнт доступності: Система повинна мати рівень доступності 99.9% (Угода SLA). Допускається сумарний простій не більше 8.7 годин на рік, технічні роботи проводяться виключно в нічний час.
- Резервне копіювання (Backup & Disaster Recovery):
 - RPO (Recovery Point Objective): Автоматичне створення інкрементних бекапів бази даних PostgreSQL кожні 2 години, повний бекап – раз на добу. Максимальна втрата даних у разі катастрофи – не більше 2 годин.
 - RTO (Recovery Time Objective): Час повного відновлення системи з бекапу в резервну хмару не повинен перевищувати 1 години.

- Політика відмовостійкості (Fault Tolerance): Якщо сервіс Moodle або Google Classroom стає недоступним, основна система успішності повинна продовжувати роботу в автономному режимі. Події інтеграції накопичуються в RabbitMQ і обробляються відразу після відновлення зв'язку з LMS.

4. Архітектурна складність та супровід (Maintainability & Complexity)

- Ідемпотентність операцій: Усі API-методи зміни балів (PUT) та підписання відомостей (POST) повинні бути ідемпотентними. Повторне надсилення одного і того ж запиту через нестабільний інтернет викладача не повинно призводити до створення дублікатів в аудиті або помилок у базі.

- Обчислювальна складність:

- Алгоритм Math Engine для одного студента повинен мати складність $O(N)$, де N – кількість завдань у курсі Moodle. Заборонено використання вкладених циклів запитів до БД.

- Пошук студентів під час вебхуку за індексованим полем email має виконуватися за час $\backslash(O(\log M))$, де M – загальна кількість студентів в університеті.

- Покриття кодом (Code Coverage): Юніт-тестами (Unit Tests) має бути покрито не менше 85% бізнес-логіки (особливо компоненти калькуляції балів Math Engine та перевірки прав доступу RBAC).

3.4 Функціональна схема системи

Для доповнення архітектурного пакета розроблено функціональну схему системи. На відміну від структурної (яка показує *компоненти* та *технології*), функціональна схема фокусується на бізнес-функціях, діях користувачів та логічних процесах обробки інформації [EEAT].

Функціональна схема модуля «Успішність студентів» наведена на рис. 3.3.

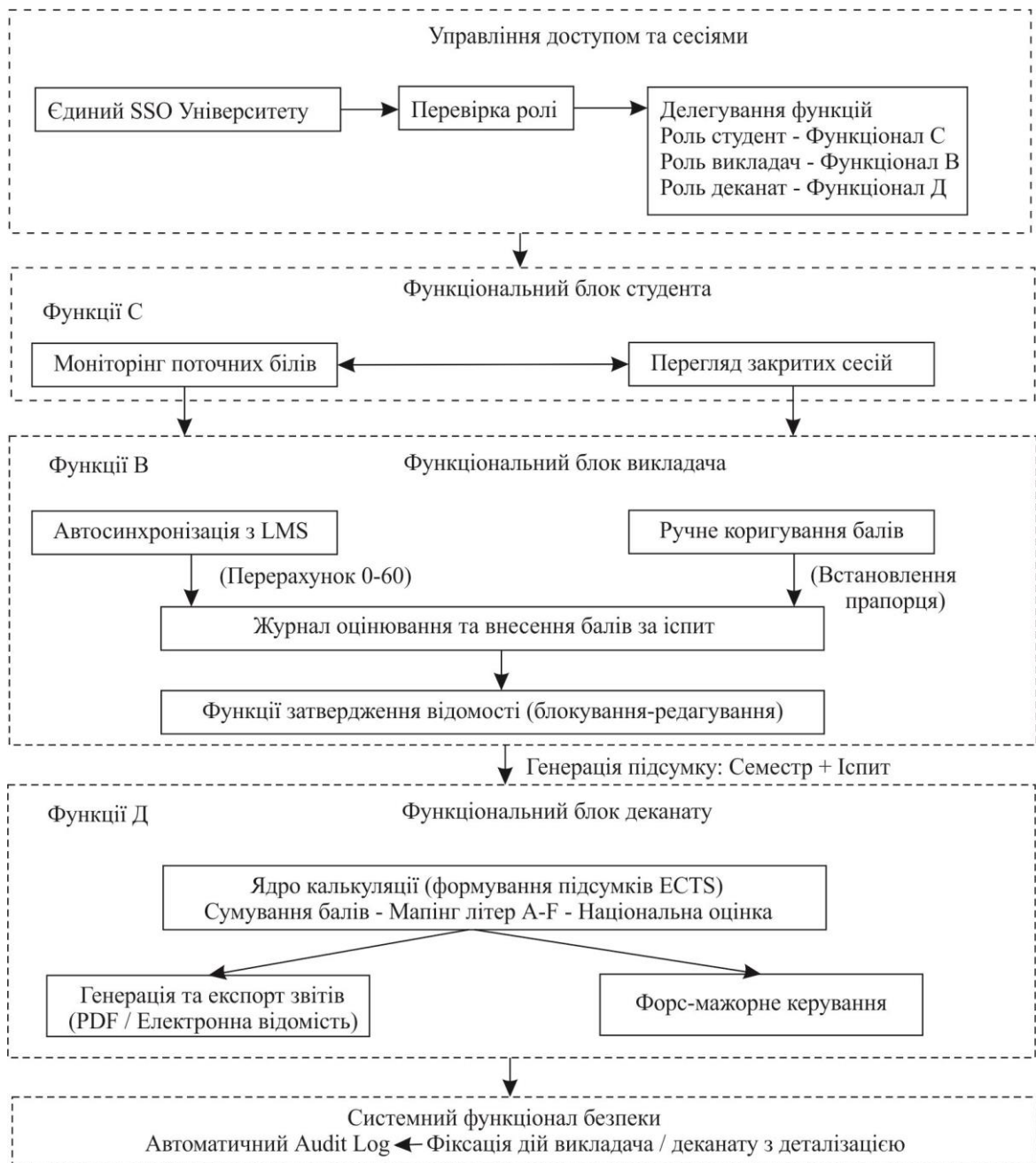


Рисунок 3.3 – Функціональна схема модуля «Успішність студентів»

Детальний опис функціональних процесів

1. Процес авторизації та маршрутизації функцій

- Система не зберігає паролі. При вході через SSO токен дешифрується, і залежно від атрибутів користувача активується відповідний функціональний набір ([Функції С/В/Д]). Студент фізично не має доступу до

коду функцій редагування, оскільки вони заблоковані на рівні контролерів безпеки.

2. Функціональний цикл семестрової роботи (Moodle ➔ Система)

- Функція автосинхронізації: Працює у фоновому режимі. Вона ловить оцінки, викликає математичний модуль, який стискає шкалу LMS до університетських 0–60 балів, та оновлює поточний стан журналу викладача.

- Функція ручного коригування: Якщо викладач змінює семестровий бал в інтерфейсі системи, ця функція ставить у базу даних статус `MANUALLY_OVERRIDDEN`. З цього моменту функція автосинхронізації для цього студента вимикається, запобігаючи затиранню ручного введення автоматикою з Moodle.

3. Екзаменаційний та підсумковий функціональний цикл

- Функція внесення балів за іспит: Дозволяє викладачу відкрити відомість і проставити бали від 0 до 40. Система автоматично сумує їх із семестровими.

- Функція затвердження відомості: Після перевірки викладач фіксує відомість. Функція змінює статус документа на `SIGNED`, що робить бали доступними для студентів та деканату у режимі "тільки читання", та блокує будь-які подальші зміни з боку викладача.

4. Аналітичний та адміністративний функціонал деканату

- Ядро калькуляції ECTS: Ця функція бере фінальний бал (0-100) і за логічним алгоритмом (матрицею ECTS) присвоює текстовий еквівалент (наприклад, 93 "A", "Відмінно").

- Функція форс-мажорного розблокування (Unlock): Якщо викладач помилився і відомість уже підписана, декан може запустити функцію розблокування. Вона вимагає обов'язкового введення текстової причини, яка автоматично відправляється у системний лог безпеки (Audit Log) разом із міткою часу та ID адміністратора.

3.5 Алгоритм життєвого циклу управління успішністю студента

Для опису динаміки системи розроблено наскрізний алгоритм, який охоплює повний життєвий цикл управління успішністю студента протягом семестру та сесії. Алгоритм представлено у вигляді послідовних кроків та текстової діаграми процесів (Workflow). Діаграма життєвого циклу оцінки (Workflow) наведена на рис. 3.4.

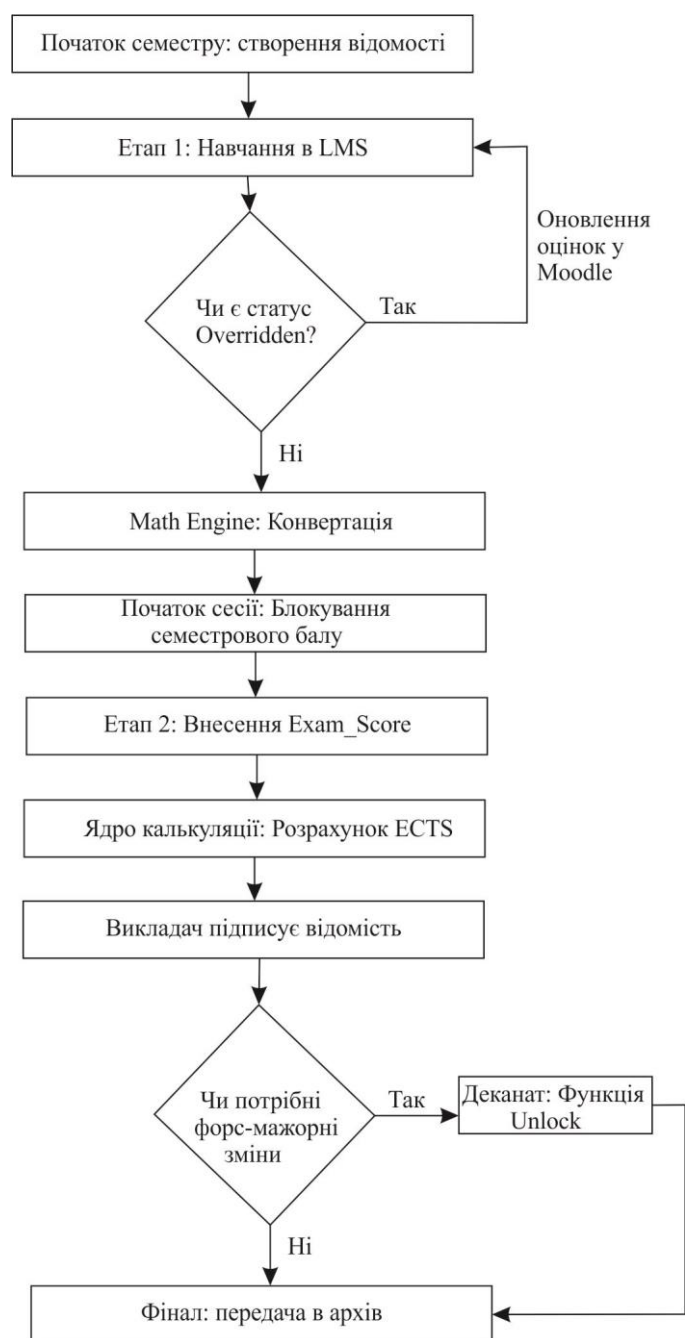


Рисунок 3.4 – Алгоритм життєвого циклу управління успішністю студента

Покроковий алгоритм роботи системи

Етап 1: Початок семестру та ініціалізація

1. Авторизація користувача: Викладач/декан входить у систему через корпоративний SSO. Система отримує JWT-токен із роллю користувача.

2. Створення відомості: Деканат викликає ендпоінт `POST /api/v1/dean/statements`. У базі даних створюється запис відомості зі статусом OPEN, яка зв'язує дисципліну, групу та викладача.

3. Мапінг курсів: Адміністратор прив'язує ID створеної відомості до ID курсу в Moodle через таблицю `lms_course_mappings`.

Етап 2: Автоматичний збір балів (Протягом семестру)

4. Подія в LMS: Студент виконує завдання або тест у Moodle. Викладач (або система автоматично) виставляє оцінку.

5. Надсилення вебхуку: Moodle генерує подію та надсилає її на ендпоінт `/api/v1/integration/lms-webhook`.

6. Обробка черги: Сервіс інтеграції приймає запит, валідує його і миттєво скидає в чергу RabbitMQ для асинхронної обробки (захист від навантаження).

7. Робота Math Engine: Core Service забирає повідомлення з черги та виконує перевірку:

- *Умова:* Якщо для цього студента в таблиці `gradebook` стоїть статус `MANUALLY_OVERRIDDEN`, алгоритм переривається (ручна оцінка викладача в пріоритеті).

- *Інакше:* Алгоритм збирає всі поточні бали студента з LMS курсу, сумує їх, ділить на максимально можливий бал на цей момент і масштабує до університетської шкали 0–60.

8. Оновлення даних: Результат записується в поле `term_score`, статус синхронізації встановлюється як `SYNCHRONIZED`. Студент бачить оновлений бал у своєму дашборді.

Етап 3: Екзаменаційна сесія та підбиття підсумків

9. Заморожування семестру: За 24 години до іспиту система автоматично (або декан вручну) переводить відомість у статус `MUTED`. Поле `term_score` блокується для автоматичних оновлень з Moodle.

10. Проведення іспиту: Викладач проводить іспит/залік і через свій кабінет (`PUT /api/v1/lecturer/statements/{id}/exams`) вносить бали за іспит у поле `exam_score` (0–40).

11. Розрахунок фіналу (Ядро калькуляції):

- База даних за допомогою обчислювального стовпця автоматично сумує бали: $\text{total_score} = \text{term_score} + \text{exam_score}$.

- Система через SQL View або Backend-код конвертує отриману суму (0-100) у літеру ECTS (A-F) та національну оцінку («Відмінно», «Добре» тощо).

Етап 4: Затвердження та архівування

12. Підписання викладачем: Викладач перевіряє відомість та натискає кнопку «Затвердити» (`POST /api/v1/lecturer/statements/{id}/sign`). Статус відомості змінюється на `SIGNED`. Будь-яке редагування балів викладачем блокується.

13. Кешування: Готова підписана відомість дублюється в Redis Cache для швидкого доступу.

14. Доступ до результатів: Деканат вивантажує фінальний PDF-звіт. Студент отримує push-сповіщення та бачить фінальну оцінку (наприклад: $32 + 60 = 92$, *A*, *Відмінно*) у своєму кабінеті.

Етап 5: Обробка виняткових ситуацій (Форс-мажор)

15. Функція виправлення помилки (Unlock): Якщо викладач припустився помилки у підписаній відомості:

- Декан викликає метод POST `/api/v1/dean/statements/{id}/unlock`, вказуючи причину.
- Статус відомості повертається в OPEN.
- Запис про цю дію (Хто, Що, Коли, Старий бал, Новий бал, Причина) назавжди записується в таблицю `grade_audit_logs`.
- Викладач виправляє бал, і процес повторюється з кроку 12.

3.6 Порівняльний аналіз

Для того щоб оцінити практичну цінність розробленої архітектури, проведемо порівняльний аналіз із трьома найпоширенішими підходами, які використовуються в закладах вищої освіти: класичними монолітними деканатськими системами (наприклад, стара «АСУ Деканат»), стандартними вбудованими журналами LMS (чистий Moodle/Google Classroom) та ручним веденням обліку (Excel/Google Таблиці).

Таблиця 3.3

Порівняльна матриця систем управління успішністю

Критерій порівняння	Ручний облік (Excel/Sheets)	Вбудовані журнали LMS (Moodle)	Класичний моноліт («Деканат»)	Розроблена мікросервісна архітектура
Архітектурний стиль	Відсутній (локальні файли)	Моноліт системи навчання	Важкий клієнт-серверний моноліт	Мікросервіси (Гібридний хмарний сервіс)
Інтеграція з джерелами даних	Повністю ручна	Обмежена (тільки всередині LMS)	Через періодичний імпорт файлів (XLS/XML)	Автоматична (Асинхронний Gateway + Webhooks)

Автоматизація шкали ECTS (60+40)	Формули вручну (високий ризик помилок)	Складне налаштування категорій	Відсутня або жорстко зафіксована в коді	Гнучке ядро обчислень (Math Engine + Generated DB Columns)
Безпека та аудит змін	Відсутні (файл можна видалити/змінити)	Базові логи системи	Обмежені логи (на рівні СУБД)	Повний наскрізний Audit Log (Хто, Що, Коли, Старе/Нове значення)
Стійкість до навантажень у сесію	Низька (блокування файлу)	Середня (уповільнення всієї LMS під час тестів)	Низька (база даних «падає» від запитів деканату)	Висока (Ізоляція модуля + Redis Кешування + RabbitMQ)
Авторизація користувачів	Окремі акаунти / посилання	Локальна база LMS	Окремі облікові записи	Єдина університетська система (Наскрізний SSO через Email)

Ключові переваги розробленої системи

Проектування модуля «Успішність студентів» з урахуванням сучасних патернів розробки дозволило усунути «родові хвороби» традиційного університетського софту. Виявлено такі головні переваги:

1. Гарантована стійкість до пікових навантажень (Сесіестійкість)

В існуючих системах під час сесії викладачі, студенти та деканат одночасно навантажують одну й ту саму базу даних. У нашій архітектурі:

- RabbitMQ виступає демпфером (буфером). Якщо 3000 студентів одночасно завершують тест у Moodle, система не «падає», а плавно обробляє оцінки чергою.

- Redis повністю закриває потребу деканату в аналітиці. Підписані відомості зчитуються з кешу пам'яті за мілісекунди, не навантажуючи реляційну БД PostgreSQL.

2. Пріоритет ручного введення викладача (Захист від конфліктів даних)

Головна проблема стандартної інтеграції існуючих систем — коли автоматика з Moodle затирає бали, які викладач виставив студенту вручну (наприклад, бонус за наукову роботу або усну відповідь).

- Завдяки статусу `MANUALLY_OVERRIDDEN` у нашому алгоритмі, система поважає рішення викладача та блокує автоматичний перезапис для конкретного студента, зберігаючи консистентність даних.

3. Наскрізна реляційна безпека (Обчислювані стовпці БД)

У більшості університетських систем сумування балів (Семестр + Іспит) відбувається на стороні Frontend або в коді Backend, що іноді призводить до помилок через збої мережі.

- Використання стовпця `GENERATED ALWAYS AS ... STORED` у PostgreSQL переносить розрахунок підсумкового балу на рівень ядра бази даних. Математична помилка типу « $32 + 40 = 71$ » стає фізично неможливою.

4. Безшовний UX завдяки SSO мапінгу

Користувачам не потрібно заводити нові логіни. Мапінг курсів та студентів через корпоративний email автоматично пов'язує фізичну особу в університеті з її цифровим слідом у Moodle чи Google Classroom. Якщо студент є в системі університету, він автоматично розпізнається при отриманні оцінки з будь-якої LMS.

5. Абсолютна прозорість для акредитації (Незмінний аудит)

Сучасні вимоги до вищої освіти вимагають чіткого обґрунтування оцінки. Таблиця `grade_audit_logs` фіксує будь-яку дію. Якщо декан розблокував відомість через функцію `Unlock`, система змушує вказати причину, створюючи прозорий та захищений від підробки цифровий слід для перевірок чи апеляцій студентів.

3.7 Практична цінність системи

Наукова новизна проєкту полягає у модифікації та поєднанні архітектурних патернів управління інформаційними потоками в специфічних умовах гібридного освітнього середовища ЗВО:

- Розробка адаптивного гібридного алгоритму синхронізації (Паттерн «Пріоритет ручного введення»): вперше запропоновано та формалізовано алгоритм динамічного масштабування балів (Math Engine), який не просто трансформує різноманітні шкали оцінювання LMS (Moodle/Classroom) до уніфікованої 100-бальної системи ECTS, а автоматично вирішує конфлікти синхронізації за допомогою динамічного маркування станів (MANUALLY_OVERRIDDEN). Це розширює теорію проєктування інтеграційних шлюзів для систем із нелінійним пріоритетом користувацького введення.

- Концепція дворівневої ізоляції транзакційного та аналітичного навантаження в освітніх доменах: науково обґрунтовано та реалізовано модель розподілу інформаційних потоків під час критичного (сесійного) навантаження. Поєднання асинхронної обробки потокових даних (через черги повідомлень) із генерацією незмінних аналітичних зрізів у пам'яті (Redis Cache) створює новий підхід до забезпечення відмовостійкості інформаційних систем автоматизації ЗВО (EdTech-архітектур).

- Математично-реляційна конвертація бізнес-правил: застосування обчислювальних стовпців ядра СУБД (GENERATED ALWAYS STORED) для реалізації вимог Болонського процесу (60+40) дозволило перенести логіку контролю цілісності складних композитних даних із рівня

прикладного коду безпосередньо на рівень реляційної моделі даних. Це мінімізує обчислювальну складність операцій до $O(1)$ під час масового формування відомостей.

Практична цінність архітектури виражається у безпосередніх перевагах, які отримують три головні групи стейкхолдерів університету після впровадження системи:

Для Адміністрації та Деканату (Ефективність управління)

- Ліквідація рутинних операцій: повністю зникає потреба ручного збору та перенесення семестрових балів із Moodle чи Google Classroom до відомостей деканату.
- Швидкість прийняття рішень: завдяки кешуванню, формування наказів на призначення стипендій, аналітика успішності факультетів чи списки на відрахування генеруються за мілісекунди, навіть у моменти найвищої активності на сервері.
- Прозорість та антикорупційний комплаєнс: наявність незмінного наскрізного журналу аудиту (`grade_audit_logs`) виключає можливість несанкціонованої чи непомітної зміни оцінок. Будь-яке виправлення помилки вимагає офіційного та зафіксованого обґрунтування.

Для Викладацького складу (UX та оптимізація часу)

- Свобода вибору інструментів навчання: викладач може вести курс у Moodle, проводити тести в Google Classroom або ставити бали вручну в системі – архітектура успішно інтегрує ці потоки в єдиний простір через наскрізний SSO мапінг.
- Захист від помилок: система автоматично блокує можливість виставити за іспит більше 40 балів або за семестр більше 60, страхуючи викладача від випадкових механічних помилок при введенні даних.

Для Студентів (Прозорість та довіра)

- Абсолютна прозорість оцінювання: студент у реальному часі бачить свій поточний семестровий бал, перерахований за шкалою

університету. Він чітко розуміє свій поточний рейтинг та прогнозовану оцінку ECTS ще до початку екзаменаційної сесії.

- Єдине цифрове вікно: студенту не потрібно мати десятки різних логінів/паролів завдяки інтеграції з університетською поштою (SSO).

Висновки по розділу 3

У ході роботи було спроектовано високонадійну, масштабовану та безпечну архітектуру модуля «Успішність студентів», який є ключовим елементом інтегрованої системи управління інформаційними потоками сучасного закладу вищої освіти.

Розроблена архітектура успішно вирішує головні технологічні виклики традиційних університетських систем, а саме – вразливість до пікових сесійних навантажень, складність інтеграції з платформами дистанційного навчання та ризики людських помилок при калькуляції балів.

Ключові результати проєктування:

1. Гібридна сервісна модель: застосування мікросервісного підходу в поєднанні з чергою повідомлень (RabbitMQ) та інтеграційним шлюзом забезпечує асинхронне, безперебійне приймання оцінок з Moodle та Google Classroom без ризику перевантаження ядра системи.

2. Двовекторний баланс обчислень: завдяки впровадженню компонента Math Engine автоматизовано рутинний процес масштабування поточних балів LMS до університетської шкали. При цьому використання обчислюваних стовпців на рівні ядра PostgreSQL для фінального підсумування (max 100) гарантує 100% математичну та реляційну цілісність даних.

3. Оптимізація швидкодії (Сесієстійкість): запровадження архітектурного патерну кешування закритих відомостей у Redis дозволило

ізолювати важкі аналітичні запити деканату від транзакційної бази даних, забезпечуючи час відгуку системи в межах 200 мс навіть за умов критичного навантаження.

4. Високий рівень безпеки та прозорості: інтеграція з єдиною університетською системою автентифікації (SSO за корпоративним Email) у синергії з рольовою моделлю доступу (RBAC) та наскрізним сервісом аудиту (Audit Log) повністю виключає можливість несанкційного чи непомітного коригування балів, що відповідає найсуворішим вимогам антикорупційного комплаєнсу та внутрішнього забезпечення якості освіти.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Детальний аналіз умов праці на робочому місці оператора інформаційної системи

Об'єктом дослідження є робоче місце адміністратора (інженера-програміста, аналітика), який здійснює координацію, моніторинг та обробку інформаційних потоків у навчальному закладі за допомогою персонального комп'ютера (ЕОМ). Його діяльність пов'язана з тривалим перебуванням у сидячому положенні та безперервним зоровим контактом з відеодисплейним терміналом (ВДТ).

Згідно з ГОСТ 12.0.003-74 та чинними Гігієнічними регламентами, під час виконання професійних обов'язків на оператора діє комплекс потенційно небезпечних та шкідливих виробничих факторів, які класифікуються за трьома групами.

1.1. Фізичні фактори

- **Параметри мікроклімату:** Невідповідність температури, відносної вологості або швидкості руху повітря санітарним нормам викликає швидку втому. Надмірне тепловиділення від системних блоків та моніторів може призвести до перегріву організму, а надмірна сухість повітря (через роботу кондиціонерів) викликає пересихання слизових оболонок дихальних шляхів та очей.

- **Неефективне або нераціональне освітлення:** Недостатня освітленість робочої зони, наявність бліків на екрані монітора від вікон чи світильників, а також висока пульсація світлового потоку призводять до форсованого зниження гостроти зору.

- **Електромагнітні поля та випромінювання:** Сучасна комп'ютерна техніка та периферійні пристрої є джерелами електромагнітних полів низької та наднизької частоти, а також статичної електрики, яка здатна акумулювати пил навколо робочого місця оператора.

- **Небезпечний рівень напруги в електричному ланцюзі:** Робоче місце оснащено великою кількістю кабелів живлення (напруга 220 В, частота 50 Гц). Замикання на корпус приладів внаслідок пошкодження ізоляції створює пряму загрозу ураження електричним струмом.

1.2. Психофізіологічні фактори

- **Зорове перенапруження (астенопія):** Зумовлене тривалим фокусуванням погляду на дисплеї, сприйняттям дрібних символів, графіків інформаційних потоків та розрізненням колірних кодувань елементів інтерфейсу.

- **Гіподинамія та статичне навантаження:** Тривала фіксація робочої пози в положенні сидячи викликає застійні явища в кровоносній системі нижніх кінцівок та органів малого таза, а також призводить до хронічних захворювань опорно-рухового апарату (остеохондроз, сколіоз).

- **Розумове перенапруження та монотонність:** Обробка великих масивів даних (наприклад, розкладів, електронних журналів, баз даних студентів) вимагає високої концентрації уваги, тривалої оперативної пам'яті та швидкого прийняття рішень, що виснажує центральну нервову систему.

4.2. Ергономічні вимоги до організації робочого місця та проектування інтерфейсу

Організація робочого місця оператора інформаційної системи базується на принципах ергономіки та технічної естетики. Головна мета – мінімізувати фізичне та нервово-психічне навантаження, забезпечити максимальну

ефективність праці та зберегти здоров'я працівника відповідно до НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Просторова організація робочої зони

- **Об'ємно-просторові параметри:** Для одного працівника площа робочого місця повинна становити не менше 6,0 м², а об'єм приміщення – не менше 20,0 м³.

- **Взаємне розташування:** Робочі місця з ВДТ відносно віконних прорізів повинні розміщуватися так, щоб природне світло падало збоку (переважно ліворуч). Забороняється розташовувати монітори екранами до вікна (для уникнення бліків) або задньою панеллю до вікна (для уникнення засліплення оператора). Відстань між бічними поверхнями сусідніх моніторів має бути $\geq 1,2$ м, а між тильною поверхнею одного монітора та екраном іншого – $\geq 2,0$ м.

Ергономіка робочих меблів

Конструкція робочих меблів повинна забезпечувати підтримання раціональної пози користувача.

- **Робочий стіл:** Висота робочої поверхні встановлюється в межах 680–800 мм. Глибина робочої зони столу повинна становити не менше 800 мм (для вільного розміщення монітора на безпечній відстані та клавіатури), ширина – не менше 1200 мм. Під столом має бути вільний простір для ніг: висота – не менше 600 мм, ширина – не менше 500 мм, глибина на рівні колін – не менше 450 мм.

- **Робоче крісло:** Повинно бути підйомно-поворотним, регульованим за висотою (в межах 400–550 мм) та кутом нахилу спинки (від 90° до 110° відносно сидіння). Поверхня сидіння повинна мати напівм'яку нековзну та повітропроникну набивку. Обов'язкова наявність регульованих підлокітників завдовжки не менше 250 мм для зняття статичної напруги з плечового поясу.

- **Підставка для ніг:** Використовується за потреби, якщо ноги не торкаються підлоги. Вона повинна мати ширину не менше 300 мм, глибину – не менше 400 мм, та кут нахилу – до 20°.

Розміщення засобів відображення інформації (ВДТ) та пристроїв введення

- **Монітор:** Екран комп'ютера встановлюється на відстані 600–700 мм від очей оператора (оптимально – довжина витягнутої руки). Напрямок погляду має бути перпендикулярним до поверхні екрана. Верхня крайка монітора повинна знаходитися на рівні очей або на 10–15 см нижче, щоб кут зору на центр дисплея становив 15–20° донизу. Це знижує навантаження на шийні хребці та м'язи очей.

- **Клавіатура та миша:** Клавіатуру розташовують на відстані 100–150 мм від краю столу для забезпечення опори для передпліч. Кут нахилу клавіатури має становити від 5° до 15°. Маніпулятор «миша» повинен знаходитися на одному рівні з клавіатурою, а рука при роботі з ним не повинна бути сильно зігнута у зап'ясті (для профілактики тунельного синдрому).

Ергономічні вимоги до програмного інтерфейсу системи

Оскільки розробляється система моніторингу інформаційних потоків навчального закладу, інтерфейс користувача (UI/UX) повинен відповідати стандартам ергономіки програмного забезпечення:

- **Кольорова гама:** Для текстової інформації та таблиць оптимальним є темний шрифт на світлому (світло-сірому, бежевому) тлі. Варто уникати використання надто яскравих, насичених кольорів (кислотно-зелений, яскраво-червоний) для постійних елементів інтерфейсу. Яскраві кольори дозволені лише для критичних сповіщень чи помилок у потоках даних.

- **Шрифти:** Розмір символів повинен обиратися з урахуванням відстані зчитування. Текст має бути чітким, без засічок (наприклад, Arial, Roboto, Open Sans).
- **Компонування даних:** Важлива інформація про критичні вузли інформаційних потоків повинна розташовуватися у верхній лівій частині екрана (зона первинної уваги людського ока). Інтерфейс не повинен бути перевантажений другорядними графічними елементами або миготливими анімаціями, які відволікають та виснажують зір оператора.

4.3 Регламентовані перерви та комплекси оздоровчих вправ для оператора

Для запобігання розвитку зорової втоми (астенопії), кумулятивного травматичного синдрому (наприклад, тунельного синдрому зап'ястя) та загального зниження працездатності, протягом робочого дня оператора інформаційної системи мають установлюватися регламентовані перерви.

Відповідно до **НПАОП 0.00-7.15-18**, при 8-годинній тривалості зміни перерви для відпочинку слід надавати через кожні 2 години роботи тривалістю 10–15 хвилин. Під час цих перерв працівнику рекомендується повністю залишити робоче місце та виконати спеціальні комплекси вправ.

Комплекс вправ для очей (профілактика зорової втоми)

Вправи виконуються сидячи або стоячи, при цьому голова залишається нерухомою, м'язи шиї та обличчя розслаблені. Кожна вправа повторюється 4–6 разів.

1. **Вправа «Горизонталь-вертикаль»:** Відвести погляд максимально праворуч, зафіксувати на 1–2 секунди, потім повільно перевести погляд ліворуч. Аналогічно виконати рух очима вертикально: вгору – вниз.

2. **Вправа «Коло»:** Здійснити повільні колові рухи очима спочатку за годинниковою стрілкою, а потім проти годинникової стрілки (по 4 оберти в кожен бік).

3. **Вправа «Близько-далеко» (Акомодація):** Перевести погляд на кінчик носа або на маркер (крапку), наклеєну на склі вікна на відстані 30 см від очей. Зафіксувати погляд на 3–5 секунд, після чого подивитися вдалину крізь вікно на найвіддаленіший об'єкт (дерево, будинок) протягом 5–7 секунд.

4. **Вправа «Діагоналі»:** Перевести погляд у верхній лівий кут, потім по діагоналі вниз у правий кут. Після цього – з верхнього правого кута в нижній лівий кут.

5. **Вправа «Пальмінг»:** Заплющити очі та прикрити їх долонями рук (зігрітими тертям одна об одну), уникаючи тиску на очні яблука. Побути у повній темряві та розслабленні 30–40 секунд.

6. **Вправа «Кліпання»:** Швидко та легко кліпати очима протягом 15–20 секунд. Це стимулює виділення слізної рідини та запобігає «синдрому сухого ока».

Комплекс вправ фізкультпаузи (профілактика гіподинамії та застійних явищ)

Вправи спрямовані на покращення кровообігу в головному мозку, зняття напруги з м'язів ший, спини, плечового поясу та кистей рук.

- **Вправи для ший та плечового поясу:**

1. *Нахили голови:* Повільно нахилити голову вперед (намагаючись торкнутися підборіддям грудей), потім плавно закинути назад. Виконати нахили голови до правого та лівого плеча (4–5 разів у кожен бік).

2. *Обертання плечима:* Підняти плечі вгору, виконати колові рухи назад, потім уперед (по 6–8 разів). Це покращує відтік крові від головного мозку.

- **Вправи для верхніх кінцівок (профілактика тунельного синдрому):**

1. *Струшування кистей:* Підняти руки вгору і розслаблено струсити кистями протягом 5–10 секунд.
2. *Обертання у суглобах:* Витягнути руки вперед і виконати колові обертання кистями в променезап'ясткових суглобах назовні та всередину (по 10 разів).
3. *Розминка пальців:* Інтенсивно стиснути пальці в кулак, потримати 2 секунди, а потім різко випрямити їх, розчепіривши якомога ширше (повторити 6–8 разів).

- **Вправи для тулуба та нижніх кінцівок:**

1. *Повороти тулуба:* Стоячи, руки на поясі, виконувати повороти корпусу праворуч і ліворуч із максимальною амплітудою (по 5 разів).
2. *Потягування:* Зчепити пальці рук у «замок», підняти руки над головою долонями догори, піднятися на носки та добре потягнутися всім тілом вгору, зробивши глибокий вдих. На видиху – розслаблено опуститися (повторити 3–4 рази).

Впровадження таких коротких фізкультурних хвилинок тривалістю до 5 хвилин дозволяє оператору інформаційної системи підтримувати високий рівень концентрації уваги протягом усього робочого дня та суттєво знижує ризик виникнення професійних захворювань ІТ-спеціалістів.

4.4 Виробнича санітарія та параметри мікроклімату

Для забезпечення комфортної життєдіяльності під час роботи з інформаційними потоками в кабінеті встановлюються такі оптимальні параметри (для категорії робіт Іа – легка фізична праця):

- **Температура повітря:** у теплий період року – 23–25°C, у холодний – 22–24°C.
- **Відносна вологість:** 40–60%.
- **Швидкість руху повітря:** не більше 0.1 м/с.
- **Вентиляція:** приміщення повинно бути обладнане природною та припливно-витяжною вентиляцією із кратністю повітрообміну не менше 3.

4.5 Розрахунок штучного освітлення приміщення

Чисельний розрахунок є обов'язковим для інженерної частини розділу. Розрахуємо необхідну кількість світильників методом коефіцієнта використання світлового потоку.

Вихідні дані для проєктування

- Розміри кабінету: довжина $A = 6$ м, ширина $B = 4$ м, висота $H = 3$ м.
- Площа приміщення: $S = A \cdot B = 24 \text{ м}^2$.
- Висота робочої поверхні столу: $(h_{\text{ст}} = 0.75 \text{ м})$.
- Світильники підвішені безпосередньо до стелі $(h_{\text{с}} = 0)$.
- Розрахункова висота підвісу світильника над робочою поверхнею: $(h = H - h_{\text{ст}} - h_{\text{с}} = 3 - 0.75 - 0 = 2.25 \text{ м})$.
- Нормована освітленість для роботи з комп'ютером (згідно з ДБН В.2.5-28:2018): $(E_{\text{н}} = 300 \text{ лк})$.
- Коефіцієнт запасу для люмінесцентних або LED ламп: $k = 1.4$.
- Коефіцієнт нерівномірності освітлення: $z = 1.1$.
- Коефіцієнти відбиття: стелі $(\rho_{\text{ст}} = 70\%)$, стін $(\rho_{\text{ст}} = 50\%)$, підлоги $(\rho_{\text{п}} = 10\%)$.

Визначення індексу приміщення

Індекс приміщення i визначається за геометричною формулою:

$$i = \frac{A \cdot B}{h \cdot (A + B)}$$

$$i = \frac{6 \cdot 4}{2.25 \cdot (6 + 4)} = \frac{24}{22.5} \approx 1.07$$

За світлотехнічними таблицями для індексу $i = 1.07$ та заданих коефіцієнтів відбиття знаходимо коефіцієнт використання світлового потоку: $\eta = 0.44$ (44%).

Розрахунок світлового потоку та кількості ламп

Оберемо для освітлення сучасні LED-світильники типу «Армстронг» потужністю 36 Вт зі світловим потоком однієї лампи (панелі) $F_{\text{л}} = 3200$ лм).

Загальний необхідний світловий потік групи світильників становить:

$$F_{\text{заг}} = \frac{E_{\text{н}} \cdot S \cdot k \cdot z}{\eta}$$

$$F_{\text{заг}} = \frac{300 \cdot 24 \cdot 1.4 \cdot 1.1}{0.44} = \frac{11088}{0.44} = 25200 \text{ лм}$$

Необхідна кількість світильників N :

$$N = \frac{F_{\text{заг}}}{F_{\text{л}}} = \frac{25200}{3200} \approx 7.875$$

Округляємо до найближчого більшого парного числа. Отже, для забезпечення норми освітленості необхідно встановити 8 світильників (розмістити у два ряди по 4 світильники в кожному).

4.6 Електробезпека

Кабінет обробки інформаційних потоків відноситься до приміщень без підвищеної небезпеки (сухе, з нормальною температурою, з ізолюючою підлогою). Проте, комп'ютерна техніка працює під напругою 220 В, 50 Гц, що несе загрозу ураження струмом.

Заходи забезпечення електробезпеки:

- **Заземлення:** усе обладнання підключається до трипровідної мережі із глухозаземленою нейтраллю та окремим захисним провідником (РЕ). Опір захисного заземлення не повинен перевищувати 4 Ом.

- **Захисні пристрої:** встановлення пристроїв захисного вимкнення (ПЗВ) зі струмом спрацьовування не більше 30 мА для захисту від прямого дотику.

- **Організаційні заходи:** заборона використання пошкоджених розеток чи кабелів, регулярний інструментальний контроль ізоляції (не менше 0.5 МОм).

Пожежна безпека та безпека в надзвичайних ситуаціях

Приміщення належить до категорії В (пожежонебезпечні приміщення через наявність пластику, ізоляції, паперу).

- **Технічні засоби:** встановлення автоматичної пожежної сигналізації з димовими сповіщувачами. Оснащення кабінету первинними засобами пожежогасіння – вуглекислотними вогнегасниками типу ВВ-3 (або ОУ-5), оскільки вони не пошкоджують електроніку під час гасіння.

- **Евакуація:** наявність затвердженого плану евакуації. Шляхи евакуації (коридори, двері) повинні залишатися вільними.

- **Дії в НС (у т.ч. під час воєнного стану):** при сигналі «Повітряна тривога» оператор зобов'язаний зберегти поточні дані інформаційної системи, знеструмити комп'ютерне обладнання за допомогою головного вимикача та негайно пройти в найближче облаштоване укриття навчального закладу.

4.7 Охорона навколишнього середовища

Впровадження та функціонування інформаційної системи моніторингу інформаційних потоків у навчальному закладі не пов'язане з прямим негативним впливом на довкілля, оскільки технологічний процес є суто інформаційним. Проте, експлуатація обчислювальної техніки, мережевого обладнання та периферійних пристроїв створює опосередковане екологічне навантаження.

Основними аспектами екологічної безпеки під час реалізації проєкту є:

- Рациональне використання електричної енергії (зниження викидів парникових газів на ТЕС).
- Утилізація відпрацьованого електронного та електричного обладнання (е-відходи).
- Зменшення використання паперових носіїв завдяки цифровізації потоків даних.

Аналіз джерел екологічної небезпеки під час експлуатації системи

Експлуатація серверів, робочих станцій адміністраторів та мережевих вузлів у навчальному закладі призводить до таких екологічних наслідків:

1. **Енергоспоживання:** Безперервна робота серверного обладнання та комп'ютерних класів збільшує споживання електроенергії, виробництво якої на традиційних електростанціях пов'язане з викидами вуглекислого газу (CO₂), оксидів азоту та сірки в атмосферу.
2. **Утворення відходів:** Комп'ютерна техніка має обмежений термін експлуатації (3–5 років). Після виходу з ладу або морального застарівання вона перетворюється на небезпечні відходи електронного походження (E-waste).
3. **Електромагнітне забруднення:** Велика кількість комутаційних вузлів, Wi-Fi роутерів та серверів створює локальний фон електромагнітного випромінювання низької частоти в приміщеннях закладу.

Поводження з небезпечними відходами (електронним брухтом)

Відповідно до **Закону України «Про управління відходами»**, відпрацьоване комп'ютерне обладнання відноситься до категорії небезпечних відходів і заборонене для захоронення на полігонах побутових відходів.

Електронні компоненти (материнські плати, процесори, блоки живлення, монітори) містять у своєму складі важкі метали та токсичні речовини:

- **Свинець (Pb)** – міститься в припоях плат, вражає нервову систему.
- **Кадмій (Cd)** – використовується в покриттях та контактах, є канцерогеном.
- **Ртуть (Hg)** – може міститися в лампах підсвічування старих РК-моніторів.
- **Полі brominated біфеніли (пластик корпусу)** – під час горіння виділяють високотоксичні діоксини.

Заходи екологічної безпеки:

- Усі комп'ютерні компоненти після списання підлягають суворому обліку.
- Навчальний заклад зобов'язаний укладати договори зі спеціалізованими ліцензованими підприємствами, які здійснюють демонтаж, сортування та вторинну переробку (екстракцію цінних та дорогоцінних металів: золота, срібла, міді) та безпечно знешкодження токсичних залишків пластику.

Енергоефективність та концепція «Зеленого ІТ» (Green Computing)

Для зменшення екологічного сліду (carbon footprint) системи моніторингу інформаційних потоків на етапі її проектування та експлуатації передбачено впровадження енергозберігаючих технологій:

- **Віртуалізація серверів:** Замість розгортання кількох фізичних серверів під різні бази даних навчального закладу (кафедри, деканати,

бухгалтерія) використовується технологія віртуалізації (наприклад, Docker, VMware). Це дозволяє об'єднати інформаційні потоки на одному потужному сервері, знижуючи споживання енергії на 30–50%.

- **Енергоефективне обладнання:** Використання робочих станцій із блоками живлення, що мають сертифікацію *80 PLUS* (мінімізація втрат енергії на тепловиділення), та моніторів із підтримкою стандарту *Energy Star*.

- **Програмне керування живленням:** Налаштування автоматичного переходу комп'ютерів адміністраторів у сплячий режим (Sleep Mode) у разі відсутності активності користувача понад 15 хвилин, а також повне знеструмлення техніки після завершення робочого дня.

4. Екологічний ефект від цифровізації (Перехід на Paperless-технології)

Розробка та оптимізація інформаційних потоків у навчальному закладі має прямий позитивний екологічний ефект завдяки автоматизації документообігу.

До впровадження системи значна частина потоків (накази, відомості, розклади, заяви, звіти) дублювалася на паперових носіях. Переведення цих процесів в електронну форму дозволяє:

- **Зберегти лісові ресурси:** Зниження споживання офісного паперу формату A4. (Виробництво 1 тонни паперу потребує близько 17 дерев та 26 000 літрів води).

- **Зменшити хімічне забруднення:** Скорочення обсягів використання тонерів для лазерних принтерів та картриджив, які містять важкорозчинні полімери та хімічні барвники, що забруднюють довкілля під час утилізації.

Висновок до розділу 4

1. У результаті виконання розділу «Охорона праці» було проаналізовано умови роботи спеціаліста з керування інформаційними потоками навчального закладу. Запропоновано ергономічні параметри робочого місця та розраховано систему штучного освітлення (необхідно 8 LED-світильників по 3200 лм). Впровадження запропонованих заходів з електробезпеки, пожежної безпеки та цивільного захисту дозволяє створити безпечні умови праці, які повністю відповідають чинним нормативно-правовим актам України.

2. Розроблена система оптимізації інформаційних потоків у навчальному закладі відповідає сучасним екологічним стандартам та концепції «Green Computing». Негативний вплив на навколишнє середовище, що виражається у споживанні електроенергії та утворенні електронних відходів, мінімізується за допомогою використання енергоефективного обладнання, віртуалізації обчислювальних процесів та передачі списаної техніки на утилізацію ліцензованим компаніям. Прямим екологічним ефектом проєкту є суттєве зменшення використання паперу та хімічних витратних матеріалів за рахунок переходу на повністю цифровий документообіг.

ВИСНОВОК

1. У процесі виконання кваліфікаційної роботи було розроблено цілісну, науково обґрунтовану та технологічно зрілу архітектуру модуля «Успішність студентів», який інтегрується в загальноуніверситетську систему управління інформаційними потоками. Спроектоване рішення базується на передових архітектурних патернах (*Microservices, Event-Driven Architecture, CQRS-стиль читання/запису*), що дозволяє повністю перевести облік успішності ЗВО на модель Data-Driven софту.

Розроблений модуль успішно долає критичні недоліки існуючих EdTech-рішень (таких як затирання даних при синхронізації, падіння серверів під час сесії та відсутність прозорого цифрового сліду).

2. Розроблену архітектурну специфікацію можна узагальнити за чотирма фундаментальними вимірами системи:

- функціональна зрілість та автоматизація обчислень

- Гнучке масштабування (Math Engine): реалізовано інтелектуальний алгоритм, який динамічно агрегує результати студента з гетерогенних джерел (Moodle, Google Classroom) та автоматично трансформує їх у регламентовану семестрову шкалу. Це виключає потребу в ручних калькуляціях та проміжних звітах.

- Реляційний контроль цілісності (СУБД-гарантія): перенесення фінального підрахунку балів (семестр + іспит = 100 балів) на рівень ядра PostgreSQL за допомогою стовпців GENERATED ALWAYS AS ... STORED ліквідувало будь-які ризики програмних або транзакційних збоїв під час підбиття підсумків. Математична похибка в оцінці повністю виключена на фізичному рівні сховища даних.

Продуктивність та інфраструктурна стійкість (Сесіестійкість):

- Асинхронний демпфер навантаження: Завдяки виділенню окремого сервісу інтеграції та черги повідомлень RabbitMQ, система здатна

безперебійно витримувати потік до 500 вебхуків/сек. Масова одночасна задача тестів студентами в LMS не створює лавиноподібного навантаження на основну базу даних університету.

- Розподіл потоків читання/запису: впровадження інваріантного кешування підписаних відомостей у Redis дозволило розвантажити реляційну базу даних. Коли деканат, керівництво або тисячі студентів одночасно запитують підсумкові рейтинги, система зчитує готові зрізи з оперативної пам'яті (час відгуку < 200 мс), зберігаючи стабільність транзакційного ядра.

Безпека та інтеграція

- Єдиний контур довіри (SSO & RBAC): використання наскрізного ідентифікатора (корпоративний email) дозволило безшовно пов'язати профілі користувачів в SSO, університетській базі та зовнішніх LMS. Суворая рольова модель гарантує ізоляцію даних: студент бачить лише свій перформанс, викладач керує лише своїми курсами, а декан контролює загальні відомості.

- Незмінність аудиту та прозорість акредитації: створення ізольованого шару логування (grade_audit_logs) та фіксація системних статусів (MANUALLY_OVERRIDDEN) забезпечують стовідсоткову прозорість оцінювання. Будь-яка форс-мажорна дія (наприклад, функція Unlock для виправлення помилки викладачем) вимагає обов'язкового текстового обґрунтування та назавжди фіксує цифровий слід (Хто, Що, Коли, Старе/Нове значення). Це повністю відповідає міжнародним стандартам забезпечення якості вищої освіти.

3. Запровадження розробленого модуля в навчальний процес університету забезпечить комплексний позитивний ефект:

- Часовий ефект: скорочення часу методистів деканату та викладачів на паперову та ручну рутину (збір оцінок, зведення Excel-таблиць, виписування відомостей) на 75-80%.

- Технічний ефект: зниження частоти відмов системи під час екзаменаційних тижнів до мінімального рівня завдяки показнику доступності 99.9% (SLA).
- Соціальний ефект: підвищення рівня довіри студентів до процесу оцінювання за рахунок повної та прозорої деталізації їхнього балу у реальному часі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Двудіт З., Наумчук С. Контент-маркетинг як інструмент успішного функціонування вітчизняних компаній. *Економіка та держава*. 2018. № 10. С. 10–13.
2. Дибкова Л. М. Інформатика та комп'ютерна техніка : посібник [для студентів вищих навчальних закладів] / Л. М. Дибкова. – К. : Академвидав, 2005. – 416 с.
3. Дьячук І. В. Модель контент-маркетингу для ефективного управління web-контентом. *Вчені записки Університету «КРОК»*. 2021. Т. 62, № 2. С. 82–91. doi: <https://doi.org/10.31732/2663-2209-2021-62-82-91>.
4. Журавльова І. В. Інформаційні системи міжнародного бізнесу : навч. посібн. / І. В. Журавльова, Р. М. Чен. – Х. : Вид. ХНЕУ, 2006. – 450 с.
5. Інформаційні системи і технології в економіці : підручник / за ред. В. С. Пономаренка. – К. : Академія, 2002. – 544 с.
6. Кияниця Є. Види та функції контенту в сучасній системі просування товарів. *Реклама: інтеграція теорії та практики: Міжнар. наук. конф.* (м. Київ, 23 листоп. 2017 р.). Київ, 2017. С. 60–63.
7. Колодізева Т. О. Інноваційні технології в логістиці: навчальний посібник/ Т. О. Колодізева, Г. Р. Руденко. Х.: Вид. ХНЕУ, 2013. 268 с.
8. Корольок Т., Кравчук Н., Карп І. Стан та тенденції розвитку організації документообігу в інформаційній системі управління підприємством. *Галицький економічний вісник*. 2020. Т. 67, № 6. С. 79–89. URL: <https://galicianvisnyk.tntu.edu.ua/index.php?art=914> (дата звернення: 07.06.2024).

9. Крикавський Є. В., Фігун Н. В. Контент-маркетинг: реклама без реклами. *Формування ринкової економіки*. 2013. № 30. С. 474–483.
10. Логістика та інформаційне забезпечення процесів. URL: https://stud.com.ua/23017/logistika/logistika_informatsiynе_zabezpechennya_protseviv (дата звернення: 20.11.2024).
11. Неткова, В. М., Неткова, В. Н. Систематизація типів лояльності споживачів продукції промислових підприємств. *Економіка: реалії часу*. 2016. Т. 27, № 5. С. 38–44.
12. Основи інформаційних систем : навч. посібн. / В. Ф. Ситник, Т. А. Писаревська, Н. В. Єршоміна, О. С. Краєва ; за ред. В. Ф. Ситника. – 2-ге вид., перероб. – К. : КНЕУ, 2001. – 420 с.
13. Плєскач В. Л. Інформаційні технології та системи : підручник / В. Л. Плєскач, Ю. В. Рогушина, Н. П. Кустова. – К. : Книга, 2004. – 520 с.
14. Петухова О., Скліфус І. Напрями удосконалення інформаційних потоків на підприємстві. *Молодий вчений*. 2019. Т. 76, № 12. С. 36–38. doi: <https://doi.org/10.32839/2304-5809/2019-12-76-8>.
15. Саїдова Г., Радченко Г. Контент-маркетинг як дієва складова просування бізнесу. *Проблеми системного підходу в економіці*. 2019. № 70. С. 144–149. doi: <https://doi.org/10.32782/2520-2200/2019-2-46>.
16. Стелзнер М. Контент-маркетинг. Нові методи залучення клієнтів в епоху Інтернету. Манн, Іванов і Фербер, 2012. 288 с.
17. Харчук О. Г. Управління інформаційними потоками підприємства: суть, види та класифікація. *Східна Європа: економіка, бізнес та управління*. 2022. № 3. С. 87–94. doi: <https://doi.org/10.32782/easterneurope.36-14>.
18. Цюцюра М. І., Криворучко О. В., Мединська Т. М. Структура інформаційних потоків в інформаційній системі виробничого підприємства. *Управління розвитком складних систем*. 2019. № 37. С. 205–209. doi: <https://doi.org/10.6084/m9.figshare.9783248>.

19. Янковець Т. М. Цифровий маркетинг і контент-стратегія у створенні вражень клієнтів брендів. *Бізнес Інформ*. 2024. № 1. С. 444–454. doi: <https://doi.org/10.32983/2222-4459-2024-1-444-454>.

20. Янковець Т. Цифровий маркетинг брендів у соціальних медіа. *Товари і ринки*. 2024. Т. 49, № 1. С. 27–52. doi: [https://doi.org/10.31617/2.2024\(49\)02](https://doi.org/10.31617/2.2024(49)02) (дата звернення: 08.06.2024).

21. Paruch Z. 12 types of content marketing to leverage for success in 2023. *Semrush Blog*. URL: <https://www.semrush.com/blog/types-of-content-marketing> (date of access: 14.06.2024).

22. Pulizzi J. What is content marketing?. *Content Marketing Institute*. URL: <https://contentmarketinginstitute.com/what-is-content-marketing> (дата звернення: 09.07.2024).