

Міністерство освіти і науки України
Державний університет «Одеська політехніка»
Навчально-науковий інститут штучного інтелекту та робототехніки
Кафедра комп'ютерних систем

Піщурнікова Валерія Олегівна,
студентка групи УК-162

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Дослідження методів підвищення ефективності системи обліку успішності
студентів

Спеціальність:

123 - Комп'ютерна інженерія

Освітня програма:

Спеціалізовані комп'ютерні системи

Керівник:

Ступень П.В.,

к.т.н., доцент

Одеса – 2021

Міністерство освіти і науки України
Державний університет «Одеська політехніка»
Навчально-науковий інститут штучного інтелекту та робототехніки
Кафедра комп'ютерних систем

Рівень вищої освіти: магістерський
Спеціальність: 123 - Комп'ютерна інженерія
Спеціалізація,
освітня програма: Спеціалізовані комп'ютерні системи

ЗАТВЕРДЖУЮ
Завідувач кафедри

Ситніков В.С.
2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Піщурнікової Валерії Олегівни, студентки групи УК-162

1. Тема роботи: Дослідження методів підвищення ефективності системи обліку успішності студентів

Керівник роботи: Ступень П.В., к.т.н., доцент

затверджені наказом ректора від _____ р. № _____

2. Зміст роботи: Аналіз минулих досліджень, аналіз методів обліку, побудова універсальної математичної моделі системи обліку успішності студентів, проектування алгоритмів основних функцій, аналіз методів оптимізації баз даних, , проектування бази даних, проектування системи обліку успішності студентів, дослідження ефективності системи методом Т. Саати.

3. Перелік ілюстративного матеріалу: Додаток А. Концептуальна модель бази даних, Додаток Б. Діаграма варіантів використання системи, Додаток В. Діаграма класів системи, Презентація.

4. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	доц. Ступень П.В.		
2	доц. Ступень П.В.		
3	доц. Ступень П.В.		

5. Дата видачі завдання: _____ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	Огляд минулих досліджень	30.08 - 15.09	
2	Розробка методів обліку	16.09 – 05.10	
3	Проектування алгоритмів обліку	06.10 – 16.10	
4	Проектування бази даних	16.10 – 25.10	
5	Проектування системи обліку	25.10 – 04.11	
6	Додатки №1, 2, 3	04.11 – 08.11	
7	Реалізація системи	08.11 – 15.11	
8	Перевірка ефективності алгоритмів	15.11 – 20.11	
9	Оформлення пояснювальної записки	20.11 – 26.11	
10	Оформлення презентації	27.11 – 30.11	
11	Захист кваліфікаційної роботи		

Здобувач вищої освіти

Піщурнікова В.О.

Керівник роботи

Ступень П.В.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІТИЧНИЙ ОГЛЯД	7
1.1 Аналіз попередніх досліджень	8
1.2 Аналіз методів обліку	15
Висновки до першого розділу	19
2 ДОСЛІДЖЕННЯ МЕТОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОБЛІКУ УСПІШНОСТІ.....	20
2.1 Моделі обліку навчальної роботи	20
2.2 Побудова універсальної математичної моделі системи обліку успішності студентів	25
2.3 Проектування алгоритмів обліку	30
2.4 Методи оптимізації баз даних	37
2.4.1 Проектування бази даних	39
Висновки до другого розділу	43
3 ПЕРЕВІРКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНИХ РІШЕНЬ	44
3.1 Проектування системи обліку успішності	44
3.2 Дослідження ефективності системи методом Т. Саати	61
Висновки до третього розділу	65
ВИСНОВКИ	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	69
ДОДАТОК А. Концептуальна модель бази даних	
ДОДАТОК Б. Діаграма варіантів використання системи	
ДОДАТОК В. Діаграма основних класів системи	

ВСТУП

Нині відбувається прискорення темпу життя, що дуже впливає людину. Нам доводиться орієнтуватися в інформаційному потоці, що постійно оновлюється, використовувати різні методи для прискорення вирішення нових завдань.

Інформаційні технології активно впроваджуються в усі сфери діяльності людини, в тому числі і в сферу освіти. У школах повсюдно використовуються електронні шкільні журнали і щоденники. Величезна кількість програмних комплексів надають можливість ведення шкільних журналів і щоденників. Однак в більшості вузів ця практика поки не впроваджена, хоча потреба в цьому існує.

В умовах розвитку дистанційного навчання можливість його модернізації вимагають наявності в вузі системи оцінки якості підготовки студентів. Однією з форм контролю є бально-рейтингова або модульно-рейтингова система.

На сьогодні в умовах карантину все гостріше стає тема переходу від паперових носіїв до електронних систем обліку. Впроваджуючи бально-рейтингові системи, багато вузів вимагають від викладачів використовувати журнали успішності (електронні журнали) як інструмент кількісної оцінки знань, навичок і умінь студентів[1].

Актуальною задачею є надати можливість студентам отримувати інформації, щодо їх успішності в реальному часі та в зручному для них вигляді. Вимоги до рівня підготовки студентів і процедури їх оцінки повинні бути відкритими і зрозумілими для всі учасників навчального процесу. Перехід до електронного обліку дає можливість зручної систематизації даних, спрощує процеси моніторингу та контролю учбової активності для викладачів та адміністрації вищих навчальних закладів. Перехід від разових іспитів в кінці навчального курсу до оцінки навчальних досягнень студента на всіх проміжних етапах навчання за курсом істотно підвищить об'єктивність оцінки цих досягнень і зробить практично неможливим будь-який обман.

Дослідженнями даної тематики займалися представники харківського авіаційного університету, полтавського державного медичного університету, ВНМУ ім. Пирогова, КНЕУ ім. В. Гетьмана, НПУ ім. М. Драгоманова. У кваліфікаційній роботі використані нароби кандидатів технічних наук хабаровської державної академії економіки і права Чуйко О.И. та Белозерова С.И, Васильєва В.М - підсистема ІАС Університет «Електронний журнал», О. П. Мулярчук та Ю. С. Антонова.

Мета дослідження полягає в тому, що на основі аналізу існуючих рішень в області впровадження електронних журналів контролю успішності і відвідування студентів, в тому числі і впровадження бально-рейтингової системи в інших вузах, а також розробки інформаційної системи, що представляє собою електронний журнал обліку відвідування та успішності студентів на основі теоретичного обґрунтування сутності, змісту та структури обліку виявити шляхи та умови підвищення його ефективності, а також провести їх експериментальне апробування.

Наукове завдання дослідження полягає у підвищенні ефективності обліку навчальної діяльності студентів вузів на основі методу врахування найважливіших факторів, що впливають на успішність.

Гіпотеза дослідження: можливість визначити, які критерії є найважливішими для процесу обліку успішності та розробити методи їх врахування дозволить систематизувати та полегшити процес обліку під час навчального семестру.

Для досягнення мети необхідно вирішити наступні завдання:

- Обґрунтувати прикладні засади обліку навчальної діяльності студентів вузів.
- У ході експериментального дослідження перевірити ефективність розробленої програми підвищення ефективності обліку навчальної діяльності студентів вузів та перевірити її експериментально.
- Виявити основні шляхи та умови підвищення ефективності контролю навчальної діяльності студентів вузів.

Об'єктом дослідження є процес обліку успішності студентів. Метод обробки підсумкових результатів студенту та формування рейтингу успішності визначені як предмет дослідження.

Покращення алгоритмів системи моніторингу успішності студентів полегшить оцінювання якості знань та дозволить студенту наочно побачити кількість пропусків, кількість захищених лабораторних або практичних робіт, бали за контрольні роботи і інші види робіт, оцінити свої досягнення в навчанні, порівняти їх з успіхами одногрупників і вжити заходів до виправлення ситуації.

Вирішення завдань дослідження здійснювалося за допомогою комплексу теоретичних (тестування, спостереження, порівняльний аналіз результатів діяльності та розбір труднощів) та практичних (експеримент, педагогічний моніторинг) методів.

Розробка системи оцінювання основана на використанні технологій веб-програмування - JS, HTML/CSS та платформа ASP.NET MVC для розробки веб-застосувань.

Новизна роботи полягає у вдосконаленні методів для полегшення керування обліком академічної успішності для всіх учасників навчально-виховної діяльності. Система створена на основі вже існуючих результатів досліджень, доопрацюванні та впровадженні запропонованих в них методів.

Система створена для забезпечення ефективності обліку успішності. Базується на сучасних результатах попередніх досліджень та запропонованих в роботі методах. Запропонована система має практичний інтерес і рекомендована до використання у закладах освіти.

1 АНАЛІТИЧНИЙ ОГЛЯД

Стрімкий розвиток інформаційних технологій та необхідність в підвищенні якості освіти обумовлюють необхідність модернізації освітнього процесу за допомогою сучасних технологій програмування. Одним із важливих аспектів навчального процесу є облік успішності студентів. Для цього викладачі можуть використовувати різні засоби від паперових журналів до електронних таблиць та баз даних, але саме за допомоги комп'ютерних технологій можна спростити та систематизувати цей процес. Розробка та покращення ефективності систем обліку академічної успішності студентів полегшить моніторинг та оцінювання якості знань студентів, зменшить трудовитрати викладачів, пришвидшить отримання даних для звітності.

Облік успішності – невід'ємний бік процесу навчального навчання.

Об'єкти оцінки успішності – це знання, вміння та навички учнів, їхня повнота, правильність, точність, міцність, зв'язок із життям, уміння застосовувати їх на практиці, а також усна, письмова, графічна, практична форми їх вираження.

Електронний журнал успішності – це автоматизована система обліку результатів навчальної діяльності студентів та відвідування ними занять. Журнал є складовою системи автоматизації управління університетом в сфері організації навчальної діяльності. Система забезпечує генерацію аналітичної інформації, яка може використовуватися як підґрунтя для прийняття управлінських рішень, спрямованих на покращення якості освіти[1].

Процес обліку успішності студентів займає значний час викладача під час проведення навчальних занять. Також непростим є процес підбиття підсумків, що є важливим при оцінці знань з будь-якої дисципліни. Для автоматизації таких процесів доцільно використовувати систему обліку успішності студентів[2].

Попередній аналіз показав, що дуже мало вищих навчальних закладів в Україні використовують автоматичні системи або підсистеми для контролю навчального процесу. Такі системи використовують харківський авіаційний університет, полтавський державний медичний університет, ВНМУ ім. Пирогова, КНЕУ ім. В. Гетьмана, НПУ ім. М. Драгоманова. На жаль, ці системи в більшості є громіздкими, незвучними для використання, що робить їх неефективними для впровадження.

Дослідження ефективності таких систем та надання рекомендацій щодо подальшого розвитку даної галузі є основною метою даної роботи.

1.1 Аналіз попередніх досліджень

Автоматизацією та розробкою програм для обліку успішності студентів та дослідженням методів обліку та виведення підсумкових оцінок займалися як вітчизняні так і зарубіжні спеціалісти. Для визначення подальшого розвитку систем обліку академічної активності та постановки чітких задач розглянемо роботи попередніх дослідників.

Кандидати технічних наук хабаровської державна академія економіки і права Чуйко О.И. та Белозерова С.И запропонували систему на основі хмарних технологій. Авторами хмарні технології обрані, перш за все, через простоти і зручності їх використання. Для створення електронного журналу був обраний Google.Диск, тому що на відміну від інших, в ньому є можливість розмежування прав доступу, створення спільних папок, створення і редагування файлів різних форматів через інтерфейс, а також сервіс Google Apps.

Система надає викладачу можливість відмічати відвідуваність студентів. За допомогою функції реалізовано рахунок заповнених клітинок, помножене на 2, тому що одне заняття становить дві академічні години (Рис 1.1)[3].

Математическое обеспечение АИС (к.т.н., доцент Чуико О.И.)																					
Лекции																					
№ г/п	ФИО	февраль			март			аттестация	апрель			май			пропуски						
1	Алесева А.А.	11	18	25	4	11	18	25	пропуски	1	8	15	22	29	6	13	25	27	лекций	аттестация	
2	Богданов А.А.				н				2											пропуски	атт.
3	Васильева В.В.					н			2	н										2	атт.
4	Галина Г.Г.								0											0	атт.
5	Денисова Д.Д.								0						н					0	атт.
6	Захаров И.С.								0					н	н	н				0	атт.
7	Иванов И.И.					н			2		н	н	н	н	н	н		н		8	н/ат
8	Петров П.П.		н	н		н	н		8											16	н/ат
9	Сидоров С.С.								0						н					0	атт.
10	Фомин И.Ф.								0						н					0	атт.
Лабораторные работы																					
№ г/п	ФИО	февраль			март			аттестация	апрель			май			пропуски						
1	Алесева А.А.	14	21	28	7	14	21	28	пропуски	4	11	18	25	16	23	30				лаб	
2	Богданов А.А.		н						2											2	
3	Васильева В.В.								0	н									н	4	
4	Галина Г.Г.		н						2	н	н									6	
5	Денисова Д.Д.								0		н			н	н					6	
6	Захаров И.С.								0					н	н	н	н			8	
7	Иванов И.И.		н	н		н			6	н	н	н	н	н	н	н				18	
8	Петров П.П.		н	н	н	н			8										н	10	
9	Сидоров С.С.								0						н					2	
10	Фомин И.Ф.								0		н	н								4	

Рисунок 1.1 – Электронный журнал обліку відвідуваності студентів

У роботі було реалізовано облік захисту лабораторних робіт студентами. При цьому для наочності було побудовано діаграму «Кількість захищених лабораторних робіт», щоб студент міг побачити свої досягнення порівняно з одногрупниками[3].

В журналі облік ведеться окремо по лекціях і лабораторних робіт, а також розроблена бально-рейтингова система. Для реалізації бально-рейтингової системи в кожній таблиці було створено по три листа: відвідуваність, успішність і рейтинг.

На аркуші «успішність» створено таблиці обліку захисту лабораторних робіт та оцінок за контрольні роботи. Для виключення помилок та описок при заповненні таблиць було створено обмеження на дані, що вводяться в комірки. Це реалізовано за допомогою функції «Перевірка даних», у діалоговому вікні якої обрано вид даних – число, та заданий діапазон від 1 до 5, вказано дію під час введення невірних даних – показувати попередження та під час введення відображати підказку[3].

У таблиці рейтингу в якості підсумку представлені: сума балів, середня оцінка за захист лабораторних робіт і кількість захищених робіт. Для наочності за кількістю

захищених робіт і кількості набраних балів побудовані діаграми, проаналізувавши які студент може оцінити свої досягнення в порівнянні з досягненнями своїх одногрупників, оцінити своє становище і обсяг майбутніх робіт.

За сумою балів у кінці семестру може бути виставлена оцінка за відповідними критеріями. Для зручності перегляду даних за допомогою Google site був створений сайт. Для кожної групи була створена окрема сторінка, а для кожної дисципліни підсторінка, в яку вставлені таблиці з Google Диска (Рис 1.2)[3].

The screenshot shows a web page for the 'Кафедра ИСТ' (Department of Information Systems). The main heading is 'Математическое обеспечение АИС'. Below it, there is a table with the following data:

№ п/п	ФИО	баллы за посещения	баллы за лаб. работы	баллы за контр. работу	доп. баллы (доклад, реферат)	сумма баллов за семестр	оценка за экзамен
1	Алесева А.А.	30	45	15	10	100	отлично
2	Богданов А.А.	28	22	9	0	59	
3	Васильева В.В.	26	33	14	8	81	хорошо
4	Галина Г.Г.	27	15	7	3	52	
5	Денисова Д.Д.	26	20	9	2	57	
6	Захаров И.С.	23	28	11	0	62	
7	Иванов И.И.	14	16	7	0	37	
8	Петров П.П.	21	18	8	2	49	
9	Сидоров С.С.	28	36	13	8	85	хорошо
10	Фомин И.Ф.	27	42	14	10	93	отлично

Below the table, there is a legend for the grades:

баллы	оценка
до 60	неуд
61-75	удовл
76-89	хорошо
90-100	отлично

Рисунок 1.2 – Вигляд системи розробленої за допомогою Google Apps

Проаналізувавши дане рішення було виявлено недоліки та можливості для подальшої розробки. Алгоритм враховує лише бали за відвідування лекцій, лабораторні роботи та модульній контроль знань. Рейтинг балів зіставляється лише для окремо взятого курсу та формується в алфавітному порядку, а не у порядку зменшення сумарної оцінки.

Ще одна розробка висвітлена у роботі Васильєва В.М - підсистема ІАС Університет «Електронний журнал». У підсистемі передбачено шість рівнів доступу[4]:

1. «Студент» має право на перегляд успішності своєї групи і на проходження тестів.
2. «Викладач» має право на зміну оцінок в журналі з предметів, які він викладає, на створення і редагування тестів зі своїх предметів, планування проходження тестів для окремих груп. Для викладача передбачена функція генерації звіту із навчального навантаження на основі даних отриманих в процесі заповнення електронного журналу, які включають дату, номер пари, зміст теми та кількість годин. Система дозволяє вносити свої корективи, в процес заповнення. Отриманий звіт імпортується у формат PDF.
3. «Староста» має право на зміну оцінок та відвідуваність в журналі з предметів своєї групи.
4. «Ректорат» має право на перегляд успішності студентів всіх груп і факультетів.
5. «Куратор» має право на перегляд успішності студентів своєї групи.
6. «Адміністратор» має необмежений доступ до всіх описаних сервісів, і до додаткових модулів адміністрування бази даних.

Таке рішення немає алгоритму автоматичного розрахунку підсумкової оцінки та можливості формування рейтингів. Система також має надлишок рівней доступу.

Комп'ютерна система «Деканат» ПП «Політек-софт» - пакет програм, що призначеній для планування та обліку навчального процесу. Основною метою пакету є скорочення часу, який витрачають працівники вищих навчальних закладів на вирішення повсякденних завдань та спрощення процедури роботи з даними. Пакет побудований за клієнт-серверною технологією, що дозволяє встановлювати його на безліч комп'ютерів, які об'єднані в локальну мережу та працюють із єдиною базою даних (Рис 1.3)[5]. Використання додаткових Web-сценаріїв забезпечує можливість доступу до бази даних в рамках окремих програм Пакету з Інтернету[5].

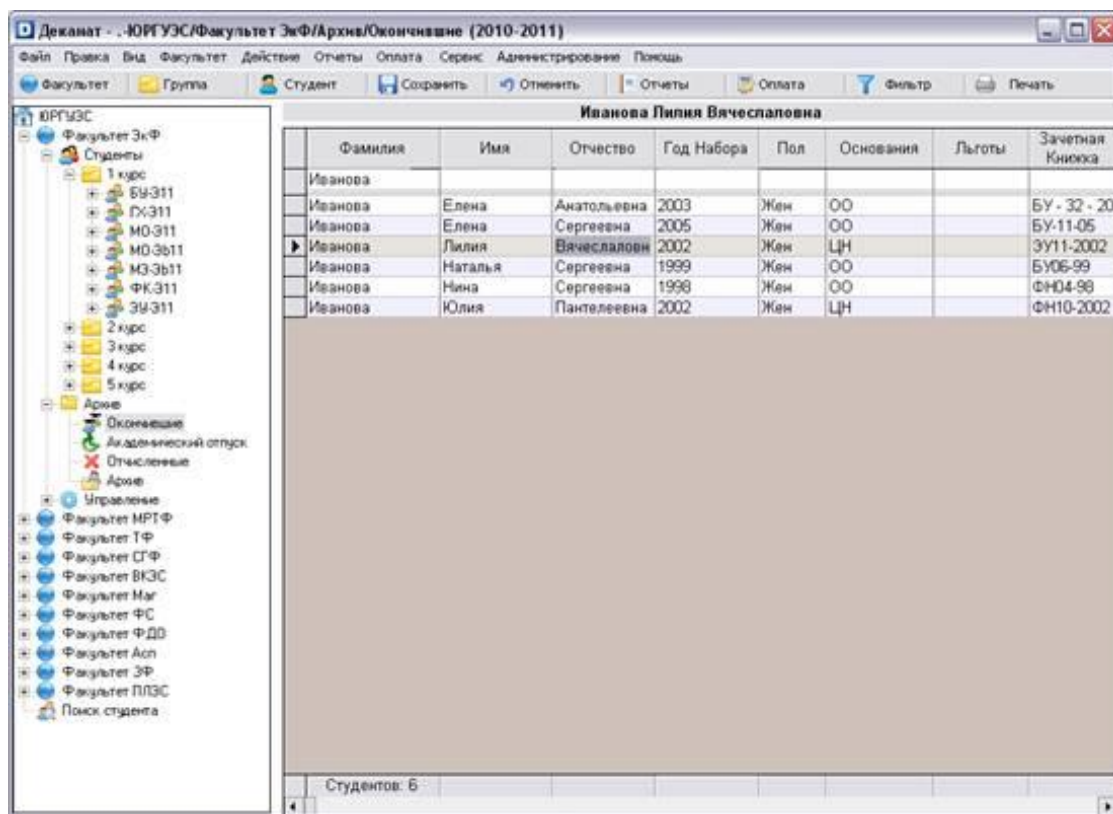


Рисунок 1.3 – Интерфейс комп'ютерної системи «Деканат»

Пакет має конструктор звітів, який дозволяє створювати та редагувати вже існуючі звітні документи, використовуючи HTML – мову розмітки гіпертексту. Звітні документи, які генерує Пакет, можна переглядати перед відправкою на друк у програмах MS Word, MS Excel, інтернет-браузері та додатково редагувати[5].

Система зосереджена більше на планування та розподілені навантаження для викладачів кафедри і немає чітких алгоритмів автоматизації підрахунків. В програмі можна контролювати відвідування занять та формувати великий обсяг даних про всіх студентів та їх успішності за весь період навчання. Проте недоліком оглянутої системи є те, що вона не відображає динаміку успішності студента. А також її надмірна складність в управлінні та введенні даних.

Особливості розробки підсистеми обліку академічної успішності студентів висвітлено у роботі О. П. Мулярчук та Ю. С. Антонова. Приведено результати розробки підсистеми обліку та оцінювання академічної успішності, яка складається

із чотирьох етапів: аналіз предметної галузі, аналіз даних, створення СУРБД, розробка веб-застосування[2]. Дана підсистема побудована з урахуванням принципів об'єктивності, неупередженості, захищеності гнучкості, доступності та зручності.

У цій роботі перевагу було надано саме розробці відкритого електронного журналу[2]. Під відкритим електронним журналом у цій роботі будемо розуміти таку систему, що дозволяє здійснювати перегляд успішності без будь якої авторизації/автентифікації[6].

Система має три рівні доступу (студент, викладач, гість) та зосередження саме на наданні вичерпної для кожного рівня інформації (Рис 1.4)[5].

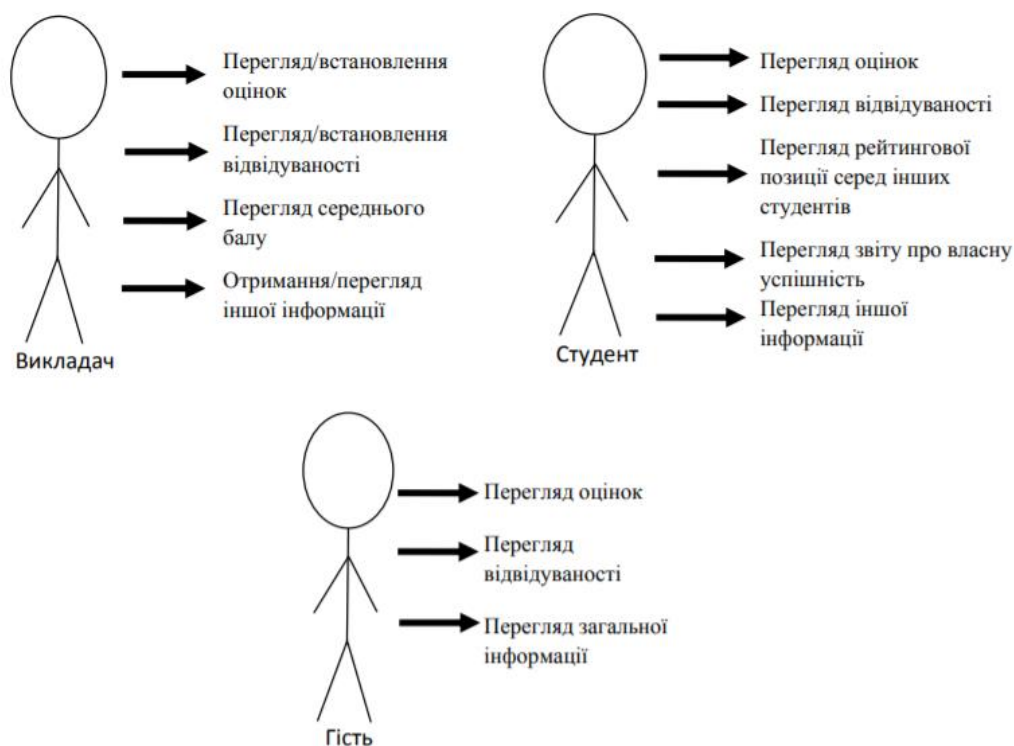


Рисунок 1.4 – Можливості системи в роботі О. П. Мулярчук та Ю. С.

Антонова

Система використовує облік відвідуваності, але немає розподіл на більшу кількість учбових активностей, таких як лекції, семінарські заняття, лабораторні та практичні роботи.

В роботі І. М. Кульчицького, Т.В. Магеровської, В.В. Сеника – «Автоматизація оцінювання знань студентів у кредитно-модульній системі болонського процесу» запропоновано розробку системи з такими характеристиками[7]: успішність з кожної навчальної дисципліни для кожної академічної групи обліковуємо окремою книгою; в межах книги для обліку створено низку документів, кожен з яких – таблиця на окремому аркуші; семестр триває 18 тижнів; кожного тижня може бути прочитана лекція, проведене практичне заняття чи лабораторна робота, студент може виконати індивідуальне завдання чи витратити час на самостійну роботу; за кожну тему практичного чи лабораторного заняття, а також за виконання індивідуального завдання чи завдання самостійної роботи студент отримує певну кількість балів; контроль за правильним розподілом балів та їхньою кількістю покладається на викладача; кожна тема практичного чи лабораторного заняття, індивідуального завдання чи завдання для самостійної роботи оцінюється для зручності у 100-бальній системі; під час оцінювання сумарного результату суму балів округлюємо до найменшого цілого, більшого від набраної суми[7].

Проаналізувавши попередні дослідження з розробки та підвищення ефективності комп'ютерних систем для автоматизації моніторингу та контролю академічної успішності студентів виявлено питання, які потребують подальшого доопрацювання.

Системи чи підсистеми, що були висвітлені вище зосередженні на зберіганні даних академічної успішності та надані звітів для адміністрації вищих навчальних закладів. Декілька з них мають алгоритми, щодо підрахування остаточних оцінок та урахування балів за різні активності, що виконуються під час навчального процесу. Значними недоліками таких систем є незручність налаштування оцінювання та застарілі методи надання інформації.

Перспективами для подальшого дослідження стали методи обліку різних академічних активностей з урахуванням всіх можливих критеріїв, які здатні вплинути на підсумкову оцінку студента для полегшення роботи викладачів. Тобто, створення зрозумілої системи налаштування оцінювання кожної дисципліни або курсової роботи (проекту).

Ще однією з перспектив стане розробка методу для підрахунку та складання підсумкового рейтингу з усіх предметів та робіт для даного семестру для кожного потоку.

1.2 Аналіз методів обліку

Оцінка успіхів орієнтує учнів як щодо рівня їх досягнень у навчальній діяльності, так і у розвитку необхідних для досягнення високої успішності морально-вольових якостей.

Правильно організований і проведений облік успішності має бути, з одного боку, всебічним, тобто охоплювати всі сторони навчально-виробничої діяльності учнів, з іншого - диференційованим, тобто здійснюватися стосовно кожної теми, кожного вузлого питання навчальної програми, до кожної виконаної навчально-виробничої роботи.

Контроль та оцінка навчальних успіхів учнів як частини обліку успішності повинні також відповідати вимогам валідності. Валідність контролю та оцінки передбачає: відповідність способів та засобів контролю змісту навчання; відповідність виявленого рівня знань та умінь поставленим цілям процесу навчання; чітка відповідність оцінки заданому обсягу знань та рівню умінь контрольованого; діагностичність, тобто об'єктивна достовірність, точність виміру, відтворюваність результатів контролю; Оптимальність процедури контролю за часу.

Поточний облік - це вид обліку, який забезпечує своєчасне засвоєння та закріплення навчального матеріалу на кожному етапі навчання[8]. Перехід до наступного етапу здійснюється за умови, якщо буде виконано попереднє завдання. Вчитель безперервно веде облік результатів, спираючись на регулярні свої спостереження за роботою учнів. Поточний облік сприяє успішному вирішенню завдань правильного, ґрунтовного та своєчасного засвоєння знань, розвитку пізнавальних здібностей учнів.

Поточний контроль знань може мати такі види:

- індивідуальна відповідь;
- усне опитування на лекціях, практичних заняттях;
- перевірка виконання письмових домашніх завдань, практичних та розрахунково-графічних робіт;
- виконання та захист практичних та лабораторних завдань;
- контрольні роботи;
- тестування, у т.ч. комп'ютерне;
- виконання самостійних робіт;
- контроль самостійної роботи (у письмовій чи усній формі);
- участь у семінарі;
- захист реферату чи творчої роботи;
- термінологічний диктант;
- тестування в інтернет-тренажері.

Тематичний облік. Закінчивши вивчення всієї теми, викладач ще раз повертається до неї надалі, щодо інших, оскільки раніше засвоєні знання нерідко органічно входять у новий матеріал як його складова частина.

Закріплення знань проводиться весь час, поки вивчається тема, але воно має підсумкове значення: учні оглядають тему загалом, усвідомлюють її структуру, систематизують засвоєння, встановлюють нові зв'язки між знаннями, простежують розвиток явищ, понять, ідей.

Періодичний облік. Це облік знань, який проводиться за певний період навчального семестру – по неділях та за модуль. При правильно поставленому поточному та тематичному обліку модулю бали можна виводити без спеціальної перевірки. Але коли рівень підготовки якоїсь частини студентів на момент виведення підсумкового балу викликає у викладача сумніви, необхідна спеціальна перевірка знань цих учнів.

Підсумковий облік. Цей вид обліку проводиться під час підсумкового повторення наприкінці навчального року. Тобто це відтворення найважливіших питань курсу, найстисліший огляд пройдених тем, розділів навчального матеріалу, осмислення та поглиблення засвоєних знань на вищому рівні.

Семестровий облік з певної дисципліни проводиться відповідно до навчального плану у вигляді семестрового екзамену або заліку в терміни, встановлені графіком навчального процесу та в обсязі навчального матеріалу, визначеного робочою програмою дисципліни[8].

Залік - одна з форм узагальнюючого підсумкового повторення. Мета заліку - систематизувати пройдений матеріал, створити в учнів цілісне уявлення про якийсь блок матеріалів, що вивчається[8]. Ця форма може бути використана контролю знань учнів. Вона дає можливість як перевірити рівень засвоєння нового матеріалу, а й показує реальні можливості учнів, сприяє самоперевірці знань, привчає до самостійної роботи, підвищує відповідальність учнів за навчання.

Семестровий іспит – це форма підсумкового контролю засвоєння студентами теоретичного та практичного матеріалу з навчальної дисципліни за семестр, що проводитиметься як контрольний захід під час екзаменаційної сесії[9].

Методи перевірки знань, умінь та навичок – це коли викладач має всі підстави вимагати від студентів звіту в освоєнні матеріалу, що вивчається, і оцінювати рівень засвоєного у кожного учня. Розрізняють декілька методів перевірки знань.

Усне опитування

На заняттях контроль знань студентів здійснюється у вигляді фронтальної та індивідуальної перевірки.

При фронтальному опитуванні за короткий час перевіряється стан знань учнів класу з певного питання чи групи питань.

Індивідуальне усне опитування дозволяє виявити правильність відповіді змісту, його послідовність, самостійність суджень і висновків, ступінь розвитку логічного мислення, культуру мови учнів. Ця форма застосовується для поточного та тематичного обліку, а також для відпрацювання та розвитку експериментальних умінь учнів.

Семінар – одна з активних форм контролю знань. Дозволяє, спираючись на знання, що учня, розширити їх і поглибити, формує вміння доводити, переконувати, відстоювати свою думку. У ході семінарського заняття відбувається узагальнення наявних у студентів знань та закріплення їх. Семінарські заняття завжди передбачають вступне слово висновок викладача, колективне обговорення.

Письмовий облік

Письмова перевірка дозволяє за короткий час перевірити знання великої кількості учнів одночасно[9]. Використовується письмовий контроль знань учнів з метою діагностики вміння застосовувати знання у навчальній практиці та здійснюється у вигляді диктантів, контрольних, перевірочних та самостійних робіт, тестів, рефератів.

Самостійна робота. Традиційна форма контролю знань, яка за своїм призначенням ділиться на навчальну самостійну роботу та контролюючу. Самостійна робота є необхідним етапом будь-якої теми. Як правило, вона проводиться після колективного вирішення чи обговорення завдань нової теми та обов'язково передують контрольній роботі з цієї теми.

Практична робота. Для закріплення теоретичних знань та відпрацювання навичок та умінь, здатності застосовувати знання при вирішенні конкретних завдань

використовується практична робота, яка пов'язана не лише із завданням на комп'ютері, а й, наприклад, може включати завдання побудови схеми, таблиці, написання програми тощо.

Лабораторна робота. Лабораторна робота - досить незвичайна форма контролю, вона вимагає від учнів як наявності знань, а й умінь застосовувати ці знання у нових ситуаціях, кмітливості[9]. Використовується лабораторна робота для закріплення певних навичок із програмними засобами, коли крім алгоритмічних розпоряджень у завданні учень може отримувати консультації викладача.

Тест являє собою короткочасне технічно порівняно просто складене випробування, що проводиться в рівних для всіх випробуваних умовах і має вигляд такого завдання, рішення якого піддається якісному обліку і є показником ступеня розвитку на даний момент відомої функції даного випробуваного.

Результат контролю навчальної діяльності учнів виявляється в оцінці. Умовним відображенням оцінки є відмітка, зазвичай виражена в балах. Найважливішою характеристикою результатів навчання є навченість, тобто. досягнутий рівень засвоєння знань, умінь та навичок, розвитку психічних процесів, якостей особистості.

Висновки до розділу

В першому розділі визначено актуальність тематики обліку академічної успішності студентів. Проведено аналіз вже існуючих рішень, концепцій, теорій та результатів попередніх досліджень, виявлено слабкі та сильні сторони запропонованих методів.

В цьому розділі в ході вивчення існуючих методів поставлено задачу про побудову універсального методу обліку успішності студентів вищих навчальних закладів. Для перевірки його ефективності на основі запропонованих рішень спроектувати систему автоматизованого обліку.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОБЛІКУ УСПІШНОСТІ

2.1 Моделі обліку навчальної роботи

Зниження трудомісткості обробки інформації про успішність студентів можна використати підхід для розробки. У рамках цієї роботи буде розглянуто поточний та підсумковий контроль, оскільки в сукупності дані форми контролю дозволяють всебічно оцінити успішність освоєння студентами навчальної дисципліни.

Існує безліч математичних моделей, що дозволяють враховувати поточну успішність студентів, кожна з них має свої переваги та недоліки.

Розглянемо деякі моделі обліку та оцінювання навчальної роботи студентів протягом семестру.

1. Дихотомічна система

Для організації оцінювання результатів навчальної діяльності студентів у вищій школі викладачами часто практикується система, яка ґрунтується на використанні дихотомічної шкали оцінки («залік-незалік»).

При використанні даної системи завдання вважається зарахованим (значення 1), якщо студент виконав правильно все завдання або є незначні похибки, інакше, за невиконання завдання або наявність грубих помилок завдання вважається не зарахованим (значення 0) [10]. Підсумкова відмітка з дисципліни виставляється учню виходячи з суми зарахованих завдань чи з частці виконання всіх завдань, у своїй викладачем заздалегідь задається і оголошується граничне значення, необхідне успішної здачі предмета.

Головною перевагою використання дихотомічної системи оцінювання є те, що дозволяє визначити загальну картину успішності засвоєння навчального матеріалу, а

також сформованості практичних навичок у учнів. До недоліків даної системи можна віднести:

- відсутнє уявлення про справжній рівень отриманих знань, умінь та навичок у учнів;
- не враховується різний рівень складності завдань, всі завдання мають однакову значимість;
- не простежується динаміка успіхів студентів у різних галузях пізнавальної діяльності [11].

Система з дихотомічною шкалою оцінки застосовується, коли як форма контролю навчальним планом встановлено залік, оскільки його метою є перевірка досягнення студентами рівня обов'язкової підготовки, проте дана система не підходить для використання у випадку, якщо формою контролю є іспит[10].

2. Накопичувальна система балів

Бальна система є зведенням правил і положень, згідно з якими стає можливою ефективна та всебічна оцінка знань, умінь, навичок, а також інших результатів навчальної діяльності студентів протягом усього терміну вивчення загальноосвітньої дисципліни. У цілому нині можна назвати такі основні особливості бальної системи:

- результати навчальної діяльності студентів за кожним поточним завданням оцінюються у балах;
- викладач самостійно розробляє методику та критерії оцінювання;
- викладач обирає форму проведення оцінної процедури на контрольних точках (самостійна робота, лабораторна робота, робота на семінарі тощо);
- викладач може розробляти та формувати власну шкалу балів, яку повідомляють студенти на початку вивчення навчальної дисципліни [12].

Протягом семестру студенти заробляють бали за виконані завдання. Виконання всіх завдань у рамках вивчення дисципліни дозволяє студентам набрати певну кількість балів за деякою шкалою.

Виконання завдань протягом семестру студент накопичує певну кількість балів, та за набраною сумою балів або за часткою виконання робіт, заздалегідь встановленою викладачем, залежно від форми проведення підсумкового контролю, виставляється залік або оцінка за іспит. У той самий час дана система передбачає можливість отримати додаткові бали: виконати додаткові завдання, взяти індивідуальну роботу, підготувати доповідь тощо. Тому можна сказати, що бальна система є засобом підвищення мотивації у студентів щодо навчальної дисципліни.

Безліч різноманітних самостійних та індивідуальних робіт, включених до бальної системи оцінювання, спрямовано на активізацію самостійної роботи студентів протягом усього семестру [13]. У той самий час дана система дозволяє активізувати пізнавальну діяльність в учнів, стимулювати їх більш якісне виконання завдань.

Однак це стосується тільки тих студентів, які насправді зацікавлені в досягненні високих результатів навчання та придбанні знань та навичок, інакше це перетворюється лише на прагнення набрати необхідну кількість балів для здачі предмета.

3. Бально-рейтингова система

Використання бально-рейтингової системи оцінювання у навчальному процесі покликане реалізовувати такі завдання:

- отримувати інтегровану оцінку навчальних здобутків студентів;
- сприяти підвищенню рівня самоорганізації самостійної навчальної діяльності студента;
- сприяти створенню духу змагань між студентами;

- надавати актуальну інформацію про рейтинг студентів потенційним роботодавцям, зацікавленим у пошуку висококваліфікованих спеціалістів.

Будь-яка бально-рейтингова система (БРС) передбачає поєднання форм поточного та підсумкового контролю з усіх навчальних дисциплін протягом кожного семестру.

У будь-якій БРС індивідуальний рейтинг навчальних досягнень студента (семестровий або підсумковий) формується як побудова під час та за результатами освоєння кожної окремої дисципліни рейтингової оцінки (показника) за встановленою бальною шкалою, як правило, 100-бальною (рейтинговий показник дисципліни)[14].

Впровадження бальної рейтингової системи дозволяє забезпечити ефективніший контроль результатів поточної навчальної діяльності студентів. У свою чергу, поточний контроль призначений для оцінки якості засвоєння студентами теоретичного матеріалу навчальної дисципліни, отриманих внаслідок практичної діяльності знань, умінь та навичок, а також рівня самостійної підготовки студентів[14].

Аналіз існуючих моделей обліку поточної успішності дозволив зробити вибір на користь змішування накопичувальної та бально-рейтингової системи оцінювання, оскільки саме можливість самоорганізації та самоврядування власною навчальною діяльністю студента має виступити як одна з основних цілей запровадження системи обліку успішності у ВНЗ.

Аналіз існуючих систем автоматизованого обліку дозволив виділити ряд істотних, на наш погляд, недоліків.

По-перше, громіздкість систем, що реалізують БРС, які викликають незручності для викладача, будучи вкрай складними в роботі, оскільки для їх функціонування потрібне налаштування дуже великої кількості параметрів.

По-друге, великий обсяг супровідної документації (планової, поточної, звітної) на паперових носіях, що спричиняє невиправдане збільшення навантаження викладача.

По-третє, різні підходи до побудови моделі поточної успішності не дуже зручно для студентів, оскільки кожен викладач пред'являє власні певні вимоги до системи. Тому виникає потреба у побудові узагальненої уніфікованої моделі обліку поточної успішності, заснованої на єдиному підході, яка також мала можливість задовольняти різні індивідуальні запити.

Таким чином, є актуальною проблема вдосконалення БРС на сучасній технологічній основі.

Для вирішення проблем зазначених вище, а також для ефективного впровадження та реалізації методів обліку бально-рейтингової системи в основу проектування мають бути покладені такі принципи:

- Алгоритм повинен мати універсальний характер, тобто, в його основі лежить узагальнена математична модель оцінювання, що охоплює різні оціночні ситуації, передбачені навчальними планами та робочими програмами дисциплін.
- Нормативний рейтинг дисципліни визначається часткою виконання всіх передбачених видів навчальної діяльності. Звідси випливає, що викладач може встановлювати бали за різні види діяльності на власний розсуд, не орієнтуючись при цьому на заздалегідь задану і певну їх кількість. Немає необхідності також пов'язувати бальну оцінку різних видів діяльності у загальній сумі балів.
- Повинна бути передбачена можливість співвідношення результатів навчання з міжнародною шкалою ECST grading scale.
- Система має бути досить універсальною та гнучкою, щоб забезпечити налаштування викладача, що передбачають облік специфічних особливостей дисципліни чи контингенту учнів.

- Створення конкретної БРС викладачем має проводитися під час інтерактивного взаємодії з електронною системою, у якому він встановлює необхідні значення початкових параметрів. Після цього система має генерувати журнал поточної успішності, підсумкову рейтингову відомість та підсумкову відомість вузівських оцінок. Усі документи створюються, використовуються та зберігаються в електронному форматі.
- Документи БРС – журнал успішності та підсумкові відомості – мають бути доступні студенту на рівні перегляду для самоорганізації та самоврядування власною навчальною діяльністю.

2.2 Побудова універсальної математичної моделі системи обліку успішності студентів

З проведеного аналітичного огляду методів обліку навчальної діяльності та порівняльне дослідження даних програмних товарів та їх застосування для вирішення задачі ведення електронного журналу успішності, можна виділити основні критерії, які впливають на проведення обліку. Для вдалого проведення обліку необхідно враховувати такі критерії:

- 1) Облік відвідування студентів;
- 2) Облік успішності теоретичної частини курсу;
- 3) Облік практичної частини курсу;
- 4) Облік успішності окремих проектів;
- 5) Облік заборгованості з дисциплін;
- 6) Підсумковий облік успішності;
- 7) Візуалізація журналу успішності;
- 8) Формування звітів журналу успішності.

Облік відвідування є невід’ємною частиною моніторингу навчально-виховного процесу. В свою чергу він може як впливати, так і не впливати на підсумкову оцінку

студента. За відвідування занять студент може отримувати бали. Найчастіше бали зараховують за відвідування лекційних занять, тоді як бали за практику нараховуються лише за виконання завдань. Викладач сам вирішує чи буде критерій впливати на успішність студента чи ні. Для позначення цього впливу введемо коефіцієнт – j , що буде приймати значення 0, якщо відвідування не впливає на успішність студента і 1, якщо впливає. Тоді критерій обліку відвідування може бути врахований, як (2.1):

$$V = j(v_1 + v_2 + \dots + v_n), \quad (2.1)$$

де v_i - облік відвідування за одне заняття.

До обліку теоретичної частини курсу віднесемо реферати або доповідь (E), розрахунково-графічні роботи (G), самостійні роботи (B), модульні контрольні роботи (M). Виходячи з того, що система контролю є бально-рейтинговою ми знаємо, що необхідно враховувати дві модульні контрольні роботи.

Розподілення по модулям можна вважати елементом підсумкового контролю. Розрахунково-графічна робота у семестрі може бути лише одна, але враховуватися може, як в одному модулі, так і частинами. Рефератів або доповідей може бути декілька та вони теж розподіляються по модулям. Так само і самостійні роботи, які дозволяють проводити тематичний облік.

До обліку практичної частини належить облік практичних (P) та семінарських занять (T), лабораторних робіт (L). Як правило, таким роботам відповідають заняття. Такі роботи є елементами поточного та часто тематичного обліку в системі контролю успішності у вищих навчальних закладах. Кількість таких робіт найчастіше, але не завжди відповідає кількості виділених програмою занять. На кожний модуль припадає приблизно половина всіх робіт семестру. Тоді для одного модулю підрахунок кількості балів за лабораторні роботи буде виглядати, як (2.2):

$$L_k = l_1 + l_2 + \dots + l_n, \quad (2.2)$$

де l_i - облік успішності роботи студента за одну роботу;

k – номер модуля.

Так само вираховується бал за модуль для семінарських та практичних робіт.

Так як, система контролю є модульною треба враховувати проміжні результати семестру, це дозволить проводити підсумковий облік більш детально. Для підрахунку балу підсумкової оцінки введемо позначення, що приведені в таблиці 2.1.

Таблиця 2.1 – Позначення видів контролю

№	Позначення	Вид контролю
1	<i>M</i>	Модульна контрольна робота
2	<i>V</i>	Відвідуваність
3	<i>E</i>	Реферат або доповідь
4	<i>G</i>	Розрахунково-графічна робота
5	<i>B</i>	Самостійні роботи
6	<i>P</i>	Практичні роботи
7	<i>T</i>	Семінарські заняття
8	<i>L</i>	Лабораторні роботи
9	<i>A</i>	Успішність за модуль
10	<i>F</i>	Успішність за семестр
11	<i>D</i>	Додаткові завдання

До курсу дисципліни можуть входити лише декілька видів контролю. Тоді для правильного розрахування модульної оцінки введемо коефіцієнт – j_q , який показує чи

є в даному курсі певний вид контролю (q) чи ні. Обов'язковою є лише модульна контрольна робота.

Важливим елементом обліку є значущість впливу певного виду контролю на підсумкову оцінку. Часто кількість виділених балів на модуль та кількість робіт не дозволяють встановити ціле число для оцінювання. Наприклад, в першому модулі виконуються три лабораторні роботи, в другому – чотири, кількість балів за лабораторні роботи часто повинна бути однакою. Для зручності бали за одну роботу можна визначити цілим числом та визначити, яка максимальна кількість балів визначена на цей вид контролю в модулі. В такому випадку доречно ввести коефіцієнт значущості – z , який визначається автоматично для зручності обліку. Такий коефіцієнт доречно визначати лише для видів контролю, які включають в себе більше одної роботи за модуль.

Додаткові роботи, як можливість підвищення модульної або підсумкової оцінки. До таких можуть відноситися презентації, відповіді на лекціях, участь у конференціях, виконання індивідуальних завдань. Вони враховуються окремо та не залежать від суми балів за основні види робіт.

В такому разі підсумкова оцінка за модуль може бути винайдена як суми балів усіх робіт, що належать заданому модулю помножена на коефіцієнт обліку (2.3).

$$A_k = M_k + j_v V_k + z_{RjR} R_k + j_G G_k + z_{SjS} S_k + z_{PjP} P_k + z_{TjT} T_k + z_{LjL} L_k, \quad (2.3)$$

де k – номер модулю, приймає значення 1 або 2.

В такому разі підсумкова оцінка є сумою оцінок за обидва модулі та додаткових балів, як необов'язкової складової (2.4):

$$F = M_1 + M_2 + (D) \quad (2.4)$$

Сума балів за всі обов'язкові види контролю не може перевищувати 100 балів.

Підсумковий облік залежить від обраного виду підсумкового контролю. Якщо це залік, підсумкова оцінка визначається за формулою 2.4. В тому разі, якщо це екзамен, тоді підсумкова оцінка винайдена за формулою порівнюється із 90 балами. Якщо вона більше, тоді студент претендує на автомат, якщо менше тоді остаточний бал визначається на екзамені.

Оцінювання проектів, що не враховують в курсі, а вимагають окремого оцінювання, таких як курсова робота або проект також може проводитися по етапам. Обов'язковим елементом обліку є захист роботи. Обліку балів за такий вид контролю може бути винайдений за формулою (2.5).

$$K = k_1 + k_2 + \dots + k_n + Z, \quad (2.5)$$

де K – курсова робота чи проект,

k_i – етап виконання роботи,

Z – етап захисту роботи.

Облік заборгованостей не впливає на підсумковий бал, але є важливим аспектом для звітності та для формування рейтингу студента.

Усі види навчальної діяльності, які передбачені у плані вивчення дисципліни, оцінюються часткою виконання навчальних завдань у рамках цієї діяльності або часткою успішності проходження контрольних точок (S_i). При цьому для кожного виду діяльності, що оцінюється, встановлюється критична частка або порогове значення $(Scr)_i$ – проходження даного виду навчальної діяльності не зараховується,

якщо фактична частка виконання, набрана k -м студентом $(S_k)_i$, виявиться нижчою за критичну, тобто. за $(S_k)_I < (S_{cr})_i$ вважається $S_{ji} = 0$.

Рейтинговий показник (R), що характеризує успішність освоєння навчального навантаження за поточний семестр (2.6):

$$R = \frac{\sum_{i=1}^N Fi + \sum_{i=1}^M Ki}{N+M} \quad (2.6)$$

У всіх видах навчальної діяльності, передбачених планом вивчення дисципліни, має бути досягнуто встановленого граничного значення, якщо $S_i = 0$, то і $R = 0$, ця умова виключає можливість компенсувати один вид діяльності за рахунок іншого.

Крім простоти та універсальності перевагами даного методу є:

- можливість для викладача встановлювати бали за різні види навчальної діяльності на свій розсуд, не орієнтуючись при цьому на якесь задане число;
- можливість зміни переліку завдань або їх змісту безпосередньо під час вивчення дисципліни, що супроводжується зміною набору.

2.3 Проектування алгоритмів обліку

На основі описаної вище універсальної моделі розробимо алгоритми для проведення автоматизованого обліку успішності.

На початку спроектуємо алгоритм налаштування навчальної діяльності, яка враховує виділені критерії обліку та надає можливість викладачу керувати програмою курсу.

На першому етапі викладач обирає створюється курс дисципліни чи курсової роботи.

Курс дисципліни вимагає обрати види учбової діяльності, які можливі в його рамках та які будуть оцінюватися. На вибір пропонуються:

- Лекції;
- Практичні заняття;
- Лабораторні роботи;
- Семінарські заняття;
- Самостійні роботи;
- Розрахунково графічні роботи;
- Реферати;
- Модульний контроль знань;
- Додаткові види діяльності, які будуть обов'язкові в рамках проходження даного курсу (повідомлення, презентації, індивідуальні завдання).

Лекційні заняття мають додаткові налаштування – наявність чи відсутність можливості отримання балів відвідування занять.

На цьому етапі налаштовується можливість отримання балів за діяльності, що необов'язкові для даного курсу, але надаються можливість отримати додаткових балів в кінці семестру.

Викладач обирає метод виставлення підсумкової оцінки – залік чи екзамен. В разі першого варіанту в кінці семестру бали, що набрані суються з урахуванням всіх коефіцієнтів. В другому випадку бали також підраховуються, але виставляються в кінцеву таблицю лише ті, що вище 90 балів, інші виставляються вручну викладачем.

Сума з урахуванням коефіцієнтів повинна дорівнювати 100 балам. Бали за додаткові завдання враховуються окремо.

Блок-схему алгоритму налаштування представлено на рисунку 2.1.

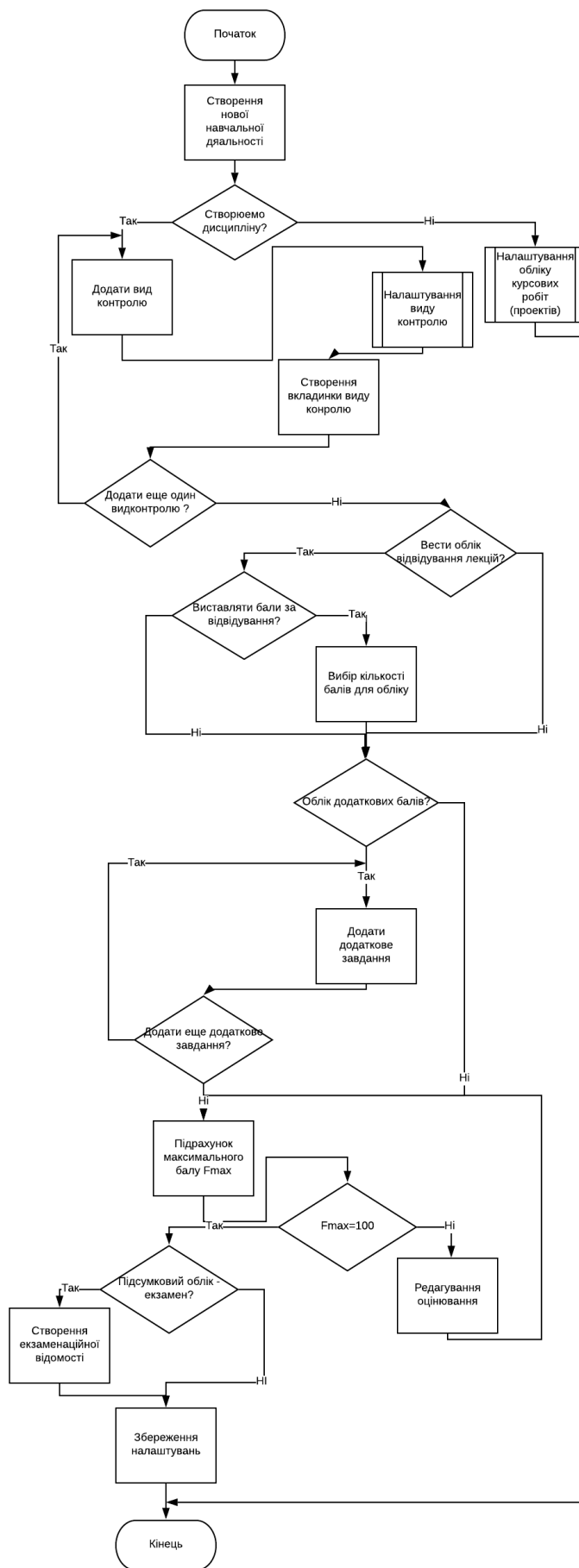


Рисунок 2.1 – Блок-схема алгоритму налаштування

Як видно з рисунку алгоритм містить підпрограму налаштування обліку певного виду контролю та підпрограму налаштування обліку курсових робіт (проектів).

На етапі налаштування обліку певного виду робіт визначається кількість балів за кожну окрему роботу, кількість робіт в кожному модулі та вираховується коефіцієнт, якщо навчальне навантаження розподілено нерівномірно. Для цього визначається кількість балів, яка надається на один модуль. Метод обліку залежить також до якого типу робіт відноситься даний вид контролю. Вигляд алгоритму представлено на рисунку 2.2.

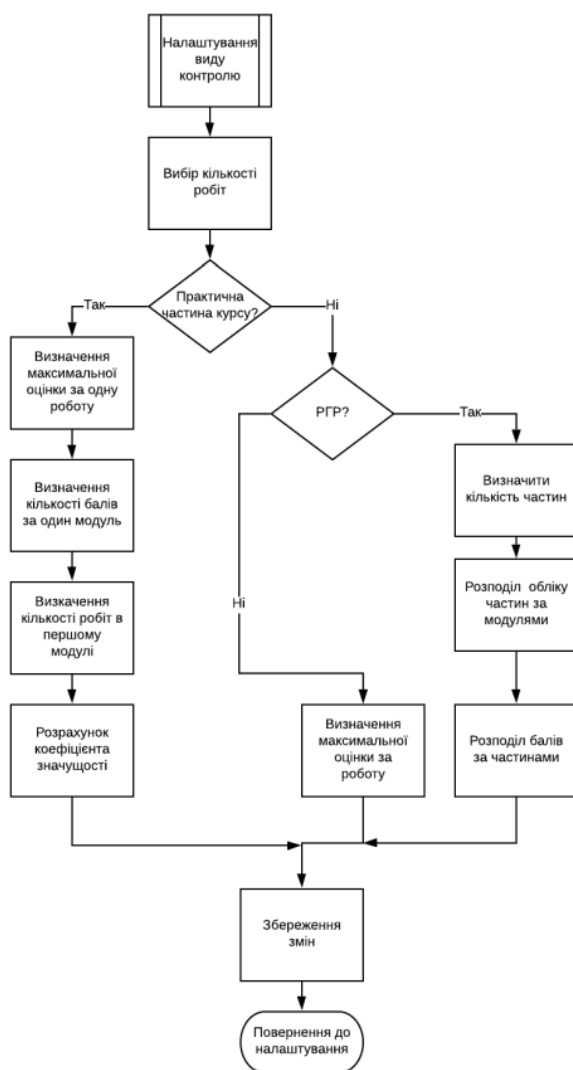


Рисунок 2.2 – Блок-схема алгоритму налаштування обліку виду контролю

Для оцінювання курсової роботи викладач додає окрему сторінку, при налаштуванні якої можна обрати кількість етапів на яку поділяється робота, бали, що студент може отримати за виконання кожної з частин. Підрахунок балів за курсову роботу або проект виконується автоматично після заповнення колонки «Захист», що є обов'язковою. Алгоритм налаштування обліку курсових робіт чи проектів показано на рисунку 2.3.

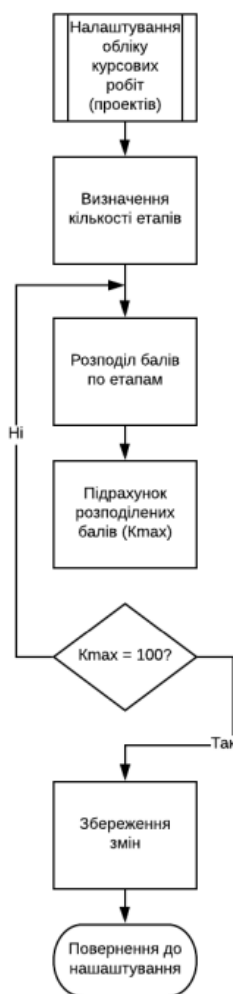


Рисунок 2.3 – Блок-схема алгоритму налаштування обліку курсових робіт (проектів)

Облік успішності студента проводиться за модулями. Протягом навчального семестру викладач виставляє бали в системі. Бали додаються в кінці кожного модулю та семестру. В кінці кожного семестру проводиться автоматичний підрахунок балів

та виставляється підсумкова оцінка. Для пришвидшення підрахунку спочатку проводиться облік за модулями, а потім за модульними оцінками визначається підсумкова. Алгоритм знаходження підсумкової представлено на рисунку 2.4.

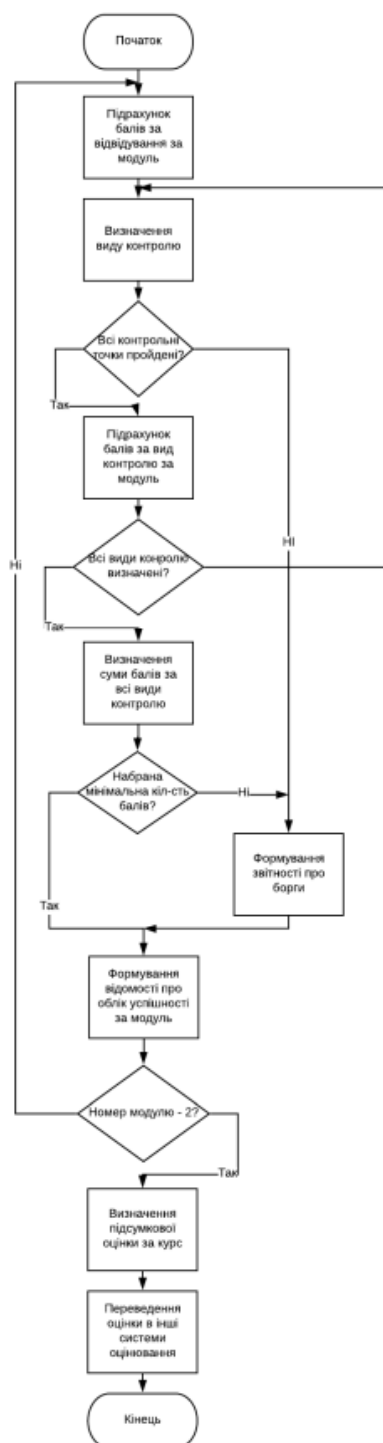


Рисунок 2.4 – Блок-схема алгоритму підрахунку підсумкових балів

Для кожного потоку зіставляється рейтинг успішності. Важливою частиною обліку є визначення рейтингового балу студента для формування рейтингу успішності. Блок-схема алгоритму винайдення рейтингового балу представлено на рисунку 2.5.

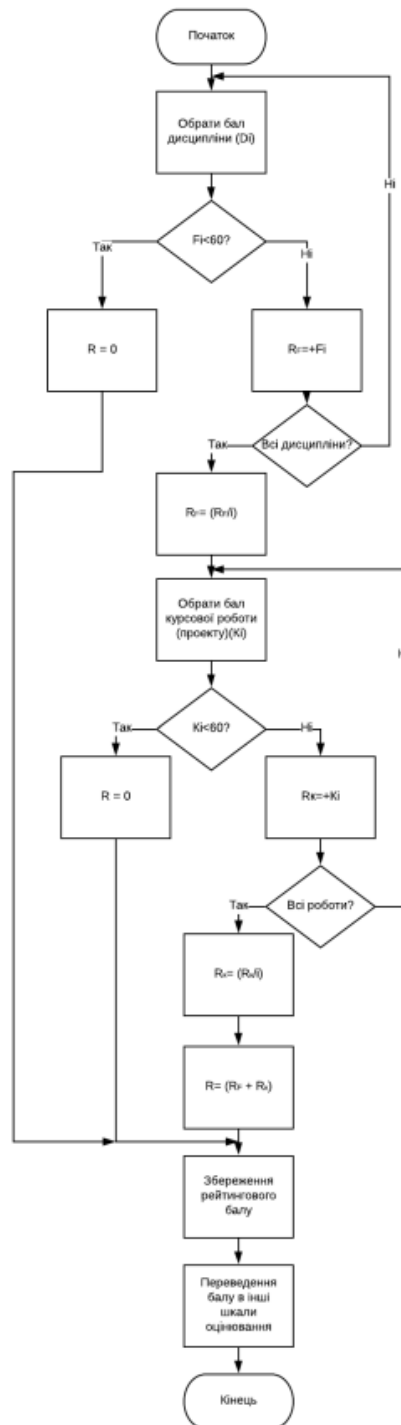


Рисунок 2.5 - Блок-схема алгоритму винайдення рейтингового балу

2.4 Методи оптимізації баз даних

Досить часто розробники зустрічаються із проблемами продуктивності баз даних або систем управління базами даних загалом. У цій роботі будуть дані загальні рекомендації з налаштування оптимізації як бази даних, так і всієї системи управління базами даних в цілому.

База даних та система управління базами даних є складними системами з кількома рівнями взаємодії з користувачем та адміністратором бази даних.

Цілком логічно, що можливості оптимізації таких складних систем також мають на увазі багаторівневність. Насправді оптимізація може бути розбита кілька етапів чи рівнів[15]:

1. Оптимізація під час проектування бази даних.
2. Оптимізація під час роботи з реалізованою базою даних (оптимізація запитів).
3. Оптимізація взаємодії програми та MS SQL Server (далі розглядатимемо взаємодію програми .NET і MS SQL Server)

До першого рівню можна віднести індексацію баз даних та нормалізацію. Індексація - це спосіб сортування кількох записів по кількох полях. Індекс - об'єкт бази даних, створений з метою підвищення продуктивності пошуку даних. Таблиці в базі даних можуть мати велику кількість рядків, які зберігаються в довільному порядку, та їх пошук за заданим критерієм шляхом послідовного перегляду таблиці рядок за рядком може тривати багато часу[16]. Індекс формується із значень одного або кількох стовпців таблиці та покажчиків на відповідні рядки таблиці і, таким чином, дозволяє шукати рядки, які відповідають критерію пошуку. Прискорення роботи з використанням індексів досягається насамперед за рахунок того, що індекс має структуру, яка оптимізована під пошук.

Для оптимальної продуктивності запитів індекси зазвичай створюються на стовпцях таблиці, які часто використовуються в запитах. Для однієї таблиці може

бути створено кілька індексів. Однак збільшення числа індексів уповільнює операції додавання, оновлення, видалення рядків таблиці, оскільки при цьому доводиться оновлювати індекси[16]. Крім того, індекси займають додатковий обсяг пам'яті, тому перед створенням індексу слід переконатися, що запланований вигаиш у продуктивності запитів перевищить додаткову витрату ресурсів комп'ютера на супровід індексу.

Нормалізація представляє процес поділу даних за окремими таблицями. Нормалізація усуває надмірність даних (data redundancy) і цим уникає порушення цілісності даних за її зміни, тобто уникає аномалій зміни (update anomaly).

Як правило, нормалізація переважно застосовується при висхідному підході проектуванні бази даних, тобто коли всі атрибути, які треба зберегти в базі даних, групуємо по сутності, для яких потім створюються таблиці[17]. Однак при низхідному підході, коли спочатку виявляються сутності, а потім їх атрибути та зв'язки між ними, нормалізація також може застосовуватися, наприклад, для перевірки коректності спроектованих таблиць.

Нормалізація передбачає застосування нормальних форм структури даних. Існує 7 нормальних форм. Кожна нормальна форма (за винятком першої) має на увазі, що до даних вже було застосовано попередню нормальну форму[17].

На рівні роботи з реалізованою базою даних можна розглядати оптимізацію запитів до бази даних. Оптимізація запитів - це функція СУБД, що здійснює пошук оптимального плану виконання запитів з усіх можливих для заданого запиту, процес зміни запиту та/або структури БД з метою зменшення використання обчислювальних ресурсів під час виконання запиту. Один і той же результат може бути отриманий СУБД у різний спосіб (планами виконання запитів), які можуть істотно відрізнятися як за витратами ресурсів, так і за часом виконання. Завдання оптимізації полягає у знаходженні оптимального способу.

У реляційній СУБД оптимальний план виконання запиту - це така послідовність застосування операторів реляційної алгебри до вихідних та проміжних відносин, яка

для конкретного поточного стану БД (її структури та наповнення) може бути виконана з мінімальним використанням обчислювальних ресурсів[18].

Як ефективний метод оптимізації також можна розглядати використання збережених процедур та функцій.

Збережені процедури є набором команд SQL, які можуть компілюватися і зберігатися на сервері. Таким чином, замість того, щоб зберігати часто використовуваний запит, клієнти можуть посилатися на відповідну процедуру, що зберігається. Це забезпечує кращу продуктивність, оскільки цей запит має аналізуватися лише один раз і зменшується трафік між сервером та клієнтом. Концептуальний рівень можна підвищити за рахунок створення на сервері бібліотеки функцій[19].

Тригер є збереженою процедурою, яка активізується при настанні певної події.

Процедури, що зберігаються, і функції (підпрограми) можуть забезпечити кращу продуктивність тому, що менше інформації потрібно для пересилання між клієнтом і сервером[19]. Вибір збільшує навантаження на сервер БД, але знижує витрати за клієнта.

Такі методи можуть бути використані для проектування бази даних, що буде використана для підвищення ефективності процесу обліку та моніторингу успішності для системи автоматизації.

2.4.1 Проектування бази даних

Для підвищення ефективності збереження та управління даними обліку успішності спроектовано базу даних із використанням методів оптимізації, які зазначені вище.

За допомогою інструментів SQL Server Management Studio, розроблено базу даних для зберігання інформації про всіх учасників навчально-виховного процесу, видів та форм контролю відвідуваності та успішності. Розроблена база даних складається з 9 таблиць, дві з яких генеруються автоматично та створені для роботи з asp.net identity.

Під час проектування бази даних в основу покладена нормалізація за третьою формою. Третя нормальна форма передбачає, що кожен стовпець, що не є ключем, повинен залежати тільки від стовпця, який є ключем, тобто має бути відсутня транзитивна функціональна залежність. У кожного запису є свій ідентифікатор, кожне поле містить тільки один запис та усі зв'язки бази побудовані через ідентифікаційний номер.

Для зберігання даних про студентів створена таблиця Students, що містить такі поля:

- Id – ідифікаційний номер студента, первиний ключ;
- GroupId – ідифікаційний номер групи, до якої належить студент;
- ApplicationUserId – ідифікаційний номер студента, як користувача системи.

Інфомрація про викладачів містить таблиця Teachers:

- Id - ідифікаційний номер викладача, первиний ключ;
- Position – посада викладача;
- AcademicTitle – наукове звання викладача;
- ApplicationUserId - ідифікаційний номер викладача, як користувача системи.

В таблиці Groups зберігається інформацію, що описує групи, до яких належать студенти:

- Id - ідифікаційний номер групи, первиний ключ;
- Course – курс до якого належить група;
- Name – назва групи;

- DirectoinId - ідифікаційний номер спеціальності, до якої належить група;
- TeacherId - ідифікаційний номер викладача, що веде певну дисципліну у відповідній групі;

Таблиця GroupStudyItem службова для підтримання зв'язку між групами студентів та дисциплінами:

- StudyItemId - ідифікаційний номер навчальної діяльності (курсу дисципліни чи курсової роботи/проекту), яка виконується групою.
- GroupsId - ідифікаційний номер потоку груп студентів.

Таблиця StudyItems містить інформацію, щодо навчальної діяльності, тобто курсу дисципліни або курсової роботи чи проекту:

- Id - ідифікаційний номер навчальної діяльності, первинний ключ;
- Name – назва навчальної діяльності (текст);
- StudyType – тип навчальної діяльності (дисципліна чи курсова робота/проект);
- TeacherId - ідифікаційний номер викладача, що веде навчальну діяльність;

Інформація щодо напрямку підготовки зберігається в таблиці Directions, що містить такі поля:

- Id – ідифікаційний номер напрямку підготовки, первинний ключ;
- Name – назва напрямку підготовки;
- Course – позначення напрямку підготовки.

Інформація щодо успішності та відвідуваності міститься в таблиці Evaluations:

- Id - ідифікаційний номер запису, первинний ключ;
- Rate – поле містить оцінка за певний вид контролю;
- IsAttend – поле містить інформацію про відвідування занять;
- StudentId - ідифікаційний номер студента, що оцінюється;
- WorkId - ідифікаційний номер виду контролю, що оцінюється.

В таблиці Works зберігається інформація про види контролю успішності:

- Id – ідифікаційний номер типу роботи, первинний ключ;
- WorkType – тип контролю успішності;
- Title – назва контролю;
- Description – опис роботи, що виконується
- Coefficient – коефіцієнт значущості роботи;
- MaxRate – максимальний бал, який можна отримати за виконання роботи;
- CreationDate – дата створення роботи;
- DueDate – дата здачі роботи;
- StudyItemId - ідифікаційний номер навчальної діяльності (курсу дисципліни чи курсової роботи/проекту) в рамках якої виконується робота.
 - В ворк для построения выборки по курсам либо по курсовым по этапам.

Передбачається, що ця таблиця буде містити велику кількість записів, тому для полегшення та пришвидшення побудови виборки видів контролю по курсам дисциплін та етапів обліку курсових робіт чи проектів використовуємо індексацію.

В ASP таблиці AspNetUsers зберігаються загальні дані про користувачів системи:

- Id - ідифікаційний номер користувача, первинний ключ;
- FirstName – ім'я користувача системи;
- LastName – поле містить прізвище користувача системи;
- Patronymic – поле містить по-батькові користувача системи;
- IsTeacher – зберігає інформацію про те, чи є користувач викладачем;
- UserName – ім'я користувача в системі;
- Email – електронна пошта користувача.

Таблиця AspNetRoles містить інформацію про ролі для доступу до функціональностей системи:

- Id - ідифікаційний номер ролі, первинний ключ;

- Name – назва ролі.

Концептуальна схема бази даних з усіма зв'язками представлена в додатку А.

Entity Framework – технологія через, яку буде відбуватися взаємодія системи з базою даних, як правило формує запити до бази самостійно, але вони не завжди є найкращим варіантом. Тому, для пришвидшення роботи із даними скористаємося методами оптимізації запитів та використанні збережених процедур та функцій. Для деяких запитів побудуємо правильну фільтрацію даних та збережемо в процедуру. Так як, це збережена процедура, то вона компілюється заздалегідь та відповідно виконується швидше ніж простий запит на вибірку даних.

Висновки до другого розділу

У другому розділі описано моделі контролю та оцінювання навчальної діяльності студентів. Було прийнято рішення про побудову універсальної моделі обліку успішності студентів на осові накопичувальної та бально-рейтингової моделей контролю з врахуванням проміжних етапів навчального процесу – модулів.

В ході проведення дослідження розроблено алгоритми для проведення обліку на осовні побудованої моделі.

Для зберігання даних про навчальний процес та його учасників, а також можливість ефективного управління спроектовано базу даних. В ході проектування застосовані методи оптимізації, які викладені у розділі.

3 ПЕРЕВІРКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНИХ РІШЕНЬ

Для проведення дослідного експерименту з питання ефективності винайдених методів обліку, було прийнято рішення розробити свою інформаційну систему, що представляє собою електронний журнал обліку відвідування та успішності студентів із застосуванням бально-рейтингової системи.

3.1 Проектування системи обліку успішності

Для подальшого дослідження буде розроблено автоматичну систему обліку академічної успішності студентів – UniversityApp.

З аналізу робіт попередників було вирішено, що розроблена система буде мати вигляд веб-застосунок. Веб-застосунок — розподілений застосунок, в якому клієнтом виступає браузер, а сервером — веб-сервер[20]. Принцип роботи системи:

1. Система матиме три рівню доступу:
 - Студент – має право на перегляд власних успіхів та загального рейтингу.
 - Викладач – додає та налаштовує курси дисциплін та курсових робіт, обирає план курсу, виставляє та змінює оцінки, має можливість редагувати курс у продовж навчального семестру.
 - Адміністрація – до неї належать завідувачий кафедри, деканат, ректорат та інші адміністративні одиниці – мають право переглядати успішність усіх курсів, груп, потоків студентів, доступна функція формування загального рейтингу.
2. Основними функціональними вимогами, яким має відповідати системи є :
 - Можливість обирати активності, що передбачає навчальна дисципліна;
 - Налаштування кількості можливих балів за виконання певної роботи;
 - Можливість визначити коефіцієнти робіт;

- Виставлення оцінок за кожну діяльність викидачем;
- Формування таблицю студента в кінці кожного семестру (модулю);
- Можливість налаштувати план моніторингу та оцінювання виконання курсової роботи чи проекту:
- Контроль відвідування занять
- Формування загального рейтингу для групи чи потоку

Основними функціональними можливостями системи є :

- Можливість обирати активності, що передбачає навчальна дисципліна;
- Налаштування кількості можливих балів за виконання певної роботи;
- Можливість визначити коефіцієнти робіт;
- Виставлення оцінок за кожну діяльність викидачем;
- Формування таблицю студента в кінці кожного семестру (модулю);
- Можливість налаштувати план моніторингу та оцінювання виконання курсової роботи чи проекту:
- Контроль відвідування занять
- Формування загального рейтингу для групи чи потоку

Всі варіанти використання системи представлені у вигляді діаграми використання в додатку Б.

3. Архітектура

На високому рівні архітектура системи є клієнт-серверною.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними[21]. Вона передбачає такі основні компоненти набір серверів, набір клієнтів, мережу (забезпечує взаємодію між клієнтами та серверами).

Модель функціонування системи полягає в наступному: клієнт робить запит серверу, сервер отримує запит, виконує його і відсилає результат клієнту (Рис. 3.1).

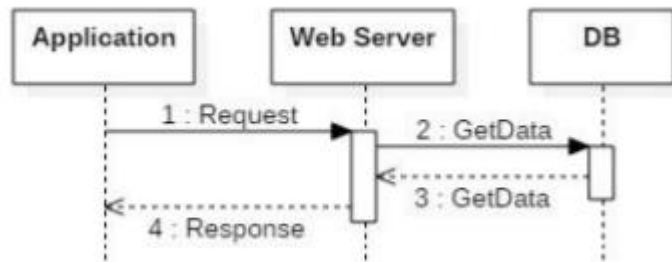


Рисунок 3.1 – Модель функціонування системи

В залежності від того, як між серверною і клієнтською частиною розподілені функції розрізняють[21]:

– модель тонкого клієнта, в рамках якої вся логіка застосунку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта.

Отже, клієнти даної системи тонкий, тому що він потрібен для представлення інформації, а відповідно сервер – товстий, так як велика кількість функцій зосереджена саме на клієнтській стороні.

Архітектура веб-застосування будується на основі архітектурного шаблону MVC. Концепція паттерна (шаблону) MVC (model - view - controller) передбачає поділ додатка на три компоненти (Рис 3.2)[21].

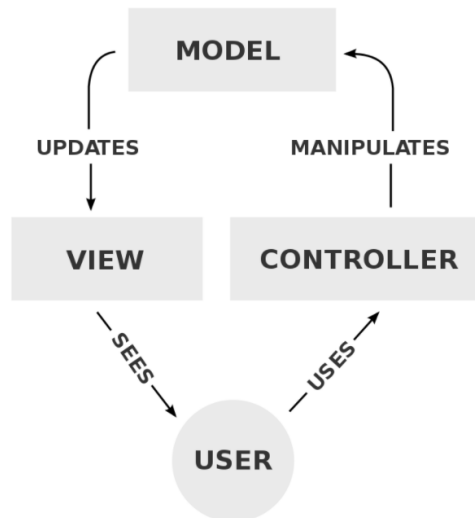


Рисунок 3.2 – Модель шаблону MVC

Контролер (controller) представляє клас, що забезпечує зв'язок між користувачем і системою, поданням і сховищем даних. Він отримує вводяться користувачем дані і обробляє їх. І в залежності від результатів обробки відправляє користувачеві певний висновок, наприклад, у вигляді подання. Один контролер може працювати з кількома моделями, і навпаки, одна модель може використовуватись у кількох контролерах[21]. У веб-застосунку зазвичай контролери - це набір однотипних класів, кожному розділу на сайті відповідає свій клас, і в ньому робляться методи (їх називають "дії", "action") для окремих сторінок.

Подання (view) - це власне візуальна частина або призначений для користувача інтерфейс програми. Як правило, html-сторінка, яку користувач бачить, зайшовши на сайт[21]. Може існувати кілька різних уявлень для виведення тих самих даних, наприклад, як таблиці, графіка чи xls-файла.

Модель (model) представляє клас, що описує логіку використовуваних даних. модель містить у собі всю логіку програми, вона зберігає та обробляє дані, при цьому не взаємодіючи з користувачем безпосередньо (звернутися до моделі можна тільки з коду, викликаючи її функції). Наприклад, збереження інформації в БД, перевірка правильності введених у форму даних - це завдання Моделі, але отримання цих даних

від користувача або виведення інформації на екран або обробка натискання на кнопку немає[21].

Завдяки цьому реалізується концепція поділ відповідальності, в зв'язку з чим легше побудувати роботу над окремими компонентами.

4. Технології програмування

Для реалізації та забезпечення надійного функціонування веб-застосунку, що розробляється в рамках дослідження обрано наступні технології.

Для клієнтської частини:

- HTML (англ. HyperText Markup Language — мова розмітки гіпертексту) — це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет[22]. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа[22].
- Каскадні таблиці стилів — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних[23]. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів[23].
- Bootstrap — це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографій, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків[22].
- JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість

на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки[24].

Для серверної частини:

- C # - сучасна об'єктно-орієнтована і типізована мову програмування. C# дозволяє розробникам створювати різні типи безпечних і надійних додатків, що виконуються в .NET[25].
- Платформа ASP.NET MVC являє собою фреймворк для створення сайтів і веб-додатків за допомогою реалізації паттерна MVC[25].
- ASP.NET Identity представляє вбудовану в ASP.NET систему автентифікації і авторизації. Дана система дозволяє користувачам створювати облікові записи, аутентифікуватись, управляти обліковими записами або використовувати для входу на сайт облікові записи зовнішніх провайдерів, таких як Facebook, Google, Microsoft, Twitter та інших[26].
- Entity Framework - це рішення для роботи з базами даних, яке використовується в програмуванні на мовах сімейства.NET. Воно дозволяє взаємодіяти з СУБД за допомогою сутностей (entity), а не таблиць. Також код з використанням EF пишеться набагато швидше[27].

5. Організація класів веб-застосунку

Організація класів настільного застосування повинна відповідати обраній архітектурі MVC [28]. Візуалізацію відношень між класами веб-застосунку можна побачити на діаграмі класів у додатку В.

З назв класів на діаграмі видно, що проект ділиться на декілька частин.

Model частина представлена бібліотекою класів Models та наступними класами:

- 1) Students – клас для формування інформації про студентів.
- 2) Teachers – клас для формування інформації про викладачів.
- 3) Groups – клас для формування інформації про групи студентів.

- 4) `StudyItems` – клас відповідальний за формування та збереження інформації про курс або курсову роботу (проект).
- 5) `Directions` – клас, який відповідає за інформацію про напрям підготовки спеціалістів.
- 6) `Evaluations` – клас, що формує дані про оцінювання та відвідуваність.
- 7) `Works` – клас відповідальний за дані про види контролю.
- 8) `ApplicationUser` – клас, що формує загальну для всіх користувачів інформацію.

Під час проектування програми скористаємося з підходом Code-First. Спершу створюємо класи моделі даних. Після того, як нам доведеться працювати з базою даних, ми використовуємо інструменти, які проектують структуру бази даних зі створеної моделі класів[28].

Після цього у разі потреби можна повернутися до цієї моделі в кодї та змінити її. Ці зміни потім можна буде відобразити в базі даних за допомогою тих самих інструментів.

Control частина представлена наступними класами:

`ServiceController` – клас за допомогою якого формується стипендіальний рейтинг.

- `GenerateRating()` – метод для генерації рейтингу.

`HomeController` – стартовий контролер, який приймає рішення про подальші можливості користувача залежно від рівня його доступу.

- `Index()` - основний метод, з якого починається робота системи;
- `Error()` – метод для обробки глобальних помилок.

`AccountController` – клас, який містить в собі необхідний функціонал для управління особистим кабінетом.

- RegisterUser() – метод, який виконує необхідні дії для реєстрації користувача;
- Login() – метод для входу в особистий кабінет користувача;
- EditUser() – метод для зміни даних про користувача системи;
- GetUserInfo() – дозволяє отримати інформацію про персональні дані користувача;
- DeleteUser() – метод, який видаляє користувача системи;
- ConfirmDeleteUser() – метод для підтвердження видалення даних про користувача системи;
- Manage() – метод для відображення інформації про користувачів.

GroupController – клас, який відповідає за роботу із інформацією про групи студентів.

- Create() – метод, що дозволяє створювати нові групи;
- Edit() – оновлює дані групи;
- Details() - метод, що дозволяє отримати повну інформацію про групу;
- Delete() – метод дозволяє видаляти групи
- ConfirmDelete() – метод, що відповідає за підтвердження видалення даних;
- AddStudentToGroup() – за допомогою цього методу створюється можливість додати студента до групи;
- DeleteStudentFromGroup() – метод, що дозволяє видалити студента з групи;
- SetCurator() – метод для призначення куратора групі студентів.

StudyItemController – клас для управління елементами навчального процесу.

Містить функціонал для роботи із курсами дисциплін або окремими проектами.

- Manage() - метод для отримання інформації про всі елементи навчальної діяльності викладача;
- Create() – метод, для створення елемента;
- Edit() – дозволяє змінювати дані про елемент навчального процесу;
- Details() – містить детальну інформацію про курс дисципліни або проект.

- Delete() – метод відповідає за видалення курсу чи проекту;
- ConfirmDelete() – метод для підтвердження видалення;
- AddGroup() – містить функціонал для того, щоб підключити групу для обліку даної діяльності;
- DeleteGroup() – виконує необхідні функції для видалення групи з обліку даного елемента навчального процесу;
- SetCoefficients() – метод, для встановлення коефіцієнтів для видів кон

WorkController – клас, який відповідає за управління видами контролю успішності.

- Create() – метод для додавання виду контролю до програми курсу;
- Edit() – метод, дозволяє редагувати та налаштовувати облік виду контролю;
- Details() – дозволяє отримати повню інформацію про вид контролю;
- Delete() – метод, що видаляє інформацію про вид контролю;
- ConfirmDelete() – метод для підтвердження видалення;
- SetEvaluations() – виконує необхідні функції для виставлення оцінок;
- GetEvaluations() – виконує функції для отримання оцінок.

DirectionController – клас, що дозволяє оперувати даними про напрямок підготовки студентів .

- Create() - метод для створення нового напрямку підготовки;
- Edit() – метод для зміни даних про напрямок підготовки;
- Details() – дозволяє отримати інформацію про спеціальність;
- Delete() – видаляє дані про напрямок підготовки;
- ConfirmDelete() – метод для підтвердження видалення;
- AddGroupToDirection() – метод, що відповідає за призначення групам напрямку підготовки;
- DeleteGroupFromDirection() – виконує функції для видалення групи з певного напрямку підготовки.

Settings - клас містить в собі налаштування пов'язані з встановленням коефіцієнтів.

ApplicationDbContext – клас для взаємодії з моделями та базою даних. В ньому ми зберігаємо набори моделей і вони містять об'єкти, що відносяться до записів в базі даних.

Для роботи з записами в базі даних на високому рівні ми використовуємо Entity Framework, яка пов'язує бази даних із концепціями об'єктно-орієнтованого програмування, створюючи «віртуальну об'єктну базу даних»[29].

Для мінімізації залежностей та підвищення гнучкості системи будемо використано механізм Dependency injection або впровадження залежностей, що дозволяє зробити взаємодіючі у додатку об'єкти слабопов'язаними[31].

6. Перевірка оптимізації бази даних

Як було сказано в попередньому розділі Entity Framework формує запити до бази даних самостійно. Але вони не оптимізовані. Наприклад, запит який було сформовано неефективно:

```
select * from Works where WorkType = 4 and DueDate > '2021-12-01' and StudyItemId = 10
```

Час виконання цього запиту складає приблизно дві секунди. Такий результат не є ефективним через те, що WorkType = 4 - поверне велику кількість записів і потім буде фільтрація далі знову у великому об'ємі даних. Тому необхідно оптимізувати запити до бази. Таким чином запит:

```
select * from Works where StudyItemId = 10 and WorkType = 4 and DueDate > '2021-12-01'
```

оптимізований запит, тому що StudyItemId = 10 - поверне мало записів і потім буде фільтрація далі знову в невеликій кількості записів. Час виконання такого запиту приблизно дорівнює 0,05 секунд.

Чим більше записів буде містити база даних тим більше буде різниця у часі витраченому на його виконання. Таким чином, налаштування фільтрації для більшості запиті, що використовує система.

8. Розробка інтерфейсу інформаційної системи

Розробка інтерфейсу інформаційної системи обліку успішності полягає в проведенні зовнішнього проектування системи, розробки взаємодії програми з користувачем: сценарій, екранні форми, набір підказок, тощо[31].

1) Взаємодія системи і користувача

Взаємодія системи і користувача відбувається через веб-застосунок. При першому відвідуванні системи користувач бачить форму входу до системи. Після введення необхідної інформації користувач потрапляє до особистого кабінету. Функції акаунта різняться залежно від виду користувача.

1.1) Види користувачів

Для даної системи існує три види користувачів: викладач, студент і адміністратор.

Студент має право лише на перегляд інформації, що стосується його власної успішності та перегляд рейтингу.

Викладач оперує створенням, налаштуванням та редагуванням інформації щодо курсів дисциплін та проектів. Має можливість виставляти оцінки та контролювати відвідування занять.

Адміністратор має право на перегляд всієї інформації про успішність студентів та оперує можливістю введення до системи нових користувачів.

1.2) Сторінки

Головна сторінка:

Так як функціональні можливості різних користувачів системи значно відрізняються між собою, було прийнято рішення про розробку інтерфейсу з єдиною

структурою, але різними функціями. В верхній частині буде розташовано меню з основними функціями. З лівого боку знаходитиметься додаткове меню. Основна частина екрану відводиться безпосередньо під журнал успішності або іншу інформацію про облік навчальної діяльності (Рис. 3.3).

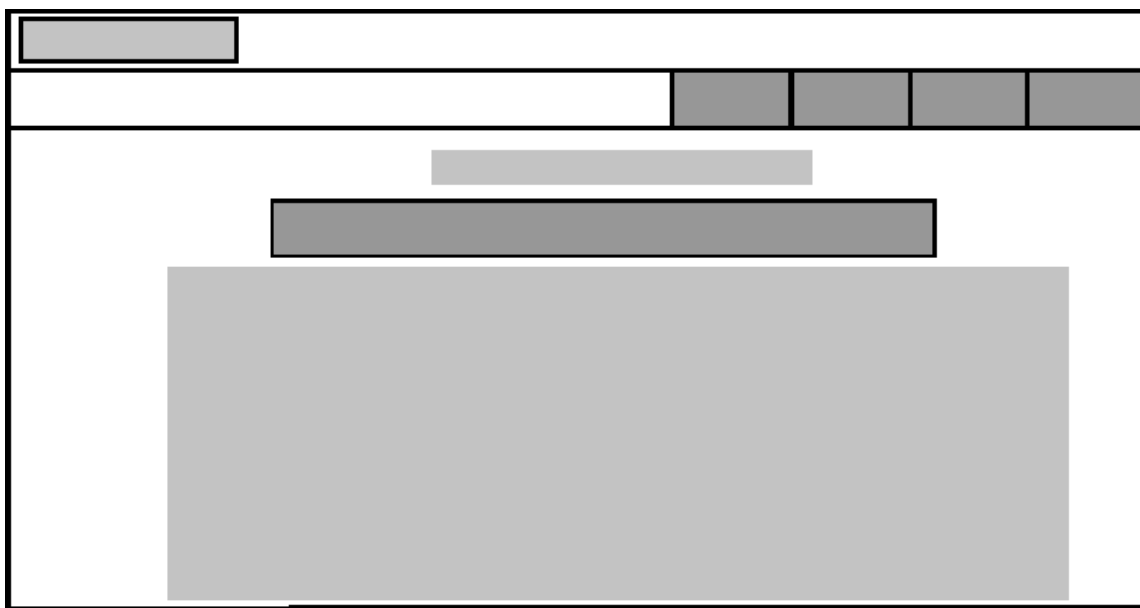


Рисунок 3.3 - Структурний вигляд головної сторінки застосунку

Вхід: сторінка з якої здійснює вхід в особистий кабінет користувача.

Особистий кабінет: сторінка містить інформацію про користувача системи.

2) Склад меню

Для кожного типу користувача буде розроблено власне головне меню з посиланням на відповідні розділи системи.

Меню студента буде складатися із таких пунктів (Рис 3.4):

- Оцінки – розділ містить усі дані про власну успішність студента;
- Рейтинг – містить рейтинг успішності потоку на якому навчається студент;

- Відвідування – містить облік відвідування занять студентом;



Рисунок 3.4 – Вигляд головного меню для користувача «Студент»

Меню для викладача містить такі пункти (Рис. 3.5):

- Курси – розділ курси дисциплін, які викладає користувач, містить випадаючий список дисциплін
- Проекти – містить посилання на проекти, містить випадаючий список проектів;
- Матеріали – містить посилання на розділ із методичними матеріалами;



Рисунок 3.5 – Вигляд головного меню для користувача «Викладач»

В розділі «Курсу» у верхній частині екрану буде розташовано, ще одне додаткове меню. В ньому будуть знаходитися вкладки відповідні видам контролю, з яких складається даний курс дисциплін.

Меню для адміністрації містить такі пункти (Рис. 3.6):

- Відвідування – розділ, що містить звіт про облік відвідування за потоком;
- Рейтинг – розділ, що містить інформацію, щодо успішності студентів за потоком;



Рисунок 3.6 – Вигляд головного меню для користувача «Адміністратор»

Однією з вимог ситеми є інтуєтивно-зрозумілий інтерфейс, яким легко користватися. Виходячи з цього дизайн проекту буде виконано у мінімалістичному стилі, для того щоб не обтяжувати зовнішній вигляд зайвою інформацією.

Останньою важливою дією є вибір кольорів. Для дизайну обрано спокійні кольори синіх відтінків та текст світлого кольору для контрасту. Вигляд розробленого дизайн-макету приведено на рисунку 3.7.

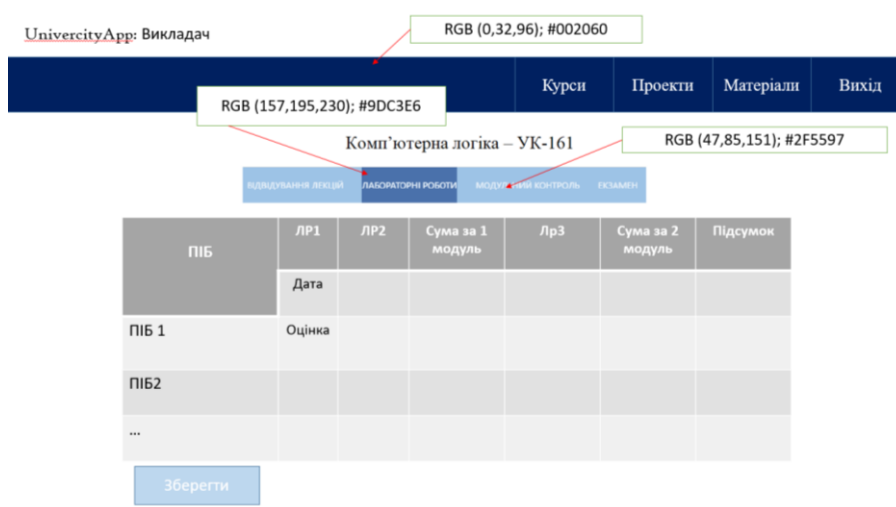


Рисунок 3.7 – Дизайн-макет сторінки для користувача «Викладач»

Задля перевірки роботи алгоритмів обліку, які були описані в другому розділі даної роботи проведемо експеримент. У системі буде створено курс «Комп'ютерна логіка». Для обліку успішності даного курсу обрано такі види контролю, як:

- Відвідування
- Лабораторні роботи
- Модульний контроль
- РГР
- Типом підсумкового обліку обрано залік

Налаштування для кожного виду контролю задано такі:

1) Облік відвідуваності ведеться, але не впливає на підсумкову оцінку. Бали не виставляються.

2) Кількість лабораторних робіт дорівнює 7 (Рис. 3.8). В першому модулі відбудуться три лабораторні роботи, в другому чотири відповідно. Максимальна кількість балів, що можна отримати за одну роботу дорівнює п'яти. Максимальна кількість балів, яку можна отримати за один модуль – 15. Ведеться контроль значущості.

Оберіть вид контролю

Лабораторні роботи

Кількість робіт

7

Кількість балів за роботу

5

Кількість балів за модуль

15

Кількість робіт в модулі №1

3

Зберегти

Рисунок 3.8 – Вибір та налаштування виду контролю

3) Розрахунково-графічна робота поділяється на два етапи, які виконуються в першому та другому модулі відповідно. Кожна частина роботи оцінюється в десять балів.

4) Модульний контроль оцінюється по 25 балів кожний.

Для перевірки можливості обліку було зареєстровано в системі групу студентів, яка складається з 11 чоловік. Цю групу підключено до створеного вище курсу.

В ході експерименту проведемо контроль обліку знань студентів. Виставляємо бали за виконання завдань. На рисунку 3.9 приведено вигляд обліку лабораторних робіт.

Комп'ютерна логіка

Група : АК-162

Відвідування Лабораторні роботи Модульні роботи РГР

№	ПІБ	лаб.1	лаб.2	лаб.3	мод.1	лаб.4	лаб.5	лаб.6	лаб.7	мод.2	Підсумок
1	Андріяшенко М.О.	5	5	4	14	3	5	4	4	16	30
2	Баєв В.Ф.	5	3	4	12	5	3	4	4	16	28
3	Бойко В.О.	4	4	4	12	4	5	4	4	17	29
4	Букреєв А.В.	5	5	4	14	3	5	4	5	17	31
5	Гімон О.С.	5	4	4	13	4	4	5	5	18	31
6	Добровольський Д.В.	4	4	5	13	5	5	5	4	19	32
7	Закордонець О.Г.	5	5	4	14	4	4	4	4	16	30
8	Калафицький А.Д.	5	5	5	15	5	5	4	4	18	33
9	Канюс В.О.	5	4	5	14	5	5	5	5	20	34
10	Колесніченко Д.В.	5	5	5	15	4	4	4	5	17	32
11	Лесюк В.М.	4	4	4	12	4	4	4	4	16	30

Зберегти

Рисунок 3.9 – Облік лабораторних робіт

Необхідно також перевірити, як надається інформація студенту протягом навчального семестру. При правильній роботі системи в особистому кабінеті учня формується таблиць успішності. Для окремого студента цієї групи облік його власної успішності виглядає, як показано на рисунку 3.10.

Комп'ютерна логіка

№	Вид роботи	Оцінка	Термін здачі	Статус
1	Лабораторна робота №1	5	09.09	Виконано
2	Лабораторна робота №2	5	25.09	Виконано
3	Лабораторна робота №3	4	10.10	Виконано
4	Лабораторна робота №4		20.10	Не оцінено
5	Лабораторна робота №5		02.11	Не оцінено
6	Лабораторна робота №6		12.11	Не оцінено
7	Лабораторна робота №7		28.11	Не оцінено
8	РГР (1 частина)	10	07.10	Виконано
9	РГР (2 частина)		05.12	Не оцінено
10	Модульна робота №1	19	17.10	Виконано
11	Модульна робота №2		15.12	Не оцінено
12	Залік		16.12	Не оцінено

Рисунок 3.10 – Табель успішності студента

Після закінчення проведення контролю знань зробимо перевірку формування рейтингу. В разі успішності операції отримаємо таблицю з підрахованим рейтинговим балом потому УК-16. Так, як в створений умовах, потік включає в себе лише одну групу, рейтинг буде сформовано лише для неї.

Для проведення перевірки було створено ще декілька дисциплін із мінімальною кількістю видів контролю, проведено оцінювання студентів в рамках цих дисциплін. Вигляд рейтингу приведено на рисунку 3.11.

Потік:

№	ПІБ	Комп'ютерна логіка	ООП	Екологія	Цифрова електроніка	ТОЕ	Рейтинг
1	Гімон О.С.	95	90	90	91	86	90.4
2	Баєв В.Ф.	90	95	90	90	81	89.2
3	Колесніченко Д.В.	84	94	89	79	90	87.2
4	Букреєв А.В.	90	67	94	80	95	85.2
5	Андріяшенко М.О.	95	90	75	77	87	84.8
6	Добровольський Д.В.	85	87	87	99	65	84.6
7	Лесюк В.М.	74	81	90	96	79	84
8	Калафицький А.Д.	82	85	65	78	87	79.4
9	Канюс В.О.	85	60	78	78	85	77.2
10	Бойко В.О.	75	89	70	80	69	76.6
11	Закордонець О.Г.	78	67	75	85	72	75.4

Рисунок 3.11 – Рейтинг успішності

3.2 Дослідження ефективності системи методом Т. Саати

Зробимо порівняльне дослідження алгоритмів програмних продуктів та їх застосування для вирішення задачі ведення електронного журналу успішності. За допомогою процесу аналітичної ієрархічної процедури Т. Сааті [32] отримаємо кількісні значення критеріїв ефективності.

Оберемо критерії для порівняльного аналізу алгоритмів систем автоматизованого обліку успішності:

1. Облік практичної частини курсу – А1;
2. Облік успішності теоретичної частини курсу – А2;
3. Облік відвідування студентів – А3;
4. Облік успішності окремих проектів – А4;
5. Облік заборгованості з дисциплін – А5;

6. Формування рейтингу успішності – А6;
7. Візуалізація журналу успішності- А7;
8. Формування звітів журналу успішності –А8.

Використовуємо процес аналітичної ієрархії процедуру Т. Сааті для знаходження ваги критеріїв [33]. Визначимо ваги за правилами заповнення матриці парних порівнянь методом Т. Сааті, продемонстровані в таблиці 3.1.

Таблиця 3.1 - Значущість коефіцієнтів матриці парних порівнянь

X_{ij}	Значущість
1	і-ий і j-ий критерії однаково важливий
3	і-ий критерій незначно переважний j-го
5	і-ий критерій переважний j-го
7	і-ий критерій значно переважний j-го
9	і-ий критерій явно переважний j-го

Матриця парних порівнянь, середні геометричні та ваги критеріїв представлені у таблиці 3.2.

Таблиця 3.2 - Матриця парних порівнянь, середні геометричні та ваги критеріїв

	A1	A2	A3	A4	A5	A6	A7	A8	Ср.геом.	Вага (a_i)
A1	1	1	7	1	5	1	9	3	2,35	0,21
A2	1	1	7	1	5	1	9	3	2,35	0,21
A3	0,14	0,14	1	0,14	0,33	0,2	3	0,2	0,32	0,03
A4	1	1	7	1	5	1	9	3	2,35	0,21
A5	0,2	0,2	3	0,2	1	0,2	7	0,33	0,60	0,05
A6	1	1	5	1	5	1	9	3	2,26	0,20

Продовження таблиці 3.2

	A1	A2	A3	A4	A5	A6	A7	A8	Ср.геом.	Вага (a _i)
A7	0,11	0,11	0,33	0,11	0,14	0,11	1	0,14	0,18	0,01
A8	0,33	0,33	5	0,33	3	0,33	7	1	1,03	0,09
Сума									11,04	1

Діаграма вагових коефіцієнтів для критеріїв представлена рисунку 3.12.



Рисунок 3.12 – Діаграма вагових коефіцієнтів критеріїв

Знайдемо інтегральний показник ефективності для алгоритмів програмного забезпечення. Для порівняння ефективності алгоритмів взято системи обліку успішності. Огляд яких висвітлено в першому розділі.

- ПЗ-1: Комп'ютерна система «Деканат» ;
- ПЗ-2: ІАС Університет «Електронний журнал»;

- ПЗ-3: Підсистема обліку академічної успішності;
- ПЗ-4: Система Електронний журнал успішності хабаровської державна академія економіки і права;
- ПЗ-5: Система обліку академічної успішності “UnivercityApp”;

Зробимо вибір категоріальної шкали від 0 до 7 для оцінки ефективності розглянутих вище критеріїв (де 0 – рівень недостатньої ефективності, 7 – рівень граничної якості) для алгоритмів обліку вкладених у дані програмні продукти.

Визначимо кількісні значення алгоритмів (таблиця 3.3). Для кожного програмного продукту обчислимо інтегральний показник ефективності. Інтегральний показник знака якості для j-го програмного засобу розраховується за формулою (3.1).

$$Q_j = \sum a_i * X_{ij} \quad (3.1)$$

Таблиця 3.3 – Інтегральні показники ефективності

Критерій	Вага (a _i)	ПЗ-1	ПЗ-2	ПЗ-3	ПЗ-4	ПЗ-5
Облік практичної частини курсу	0,21	3	4	6	5	7
Облік теоретичної частини курсу	0,21	5	7	6	7	6
Облік відвідування студентів	0,03	7	0	7	7	6
Облік успішності окремих проектів	0,21	0	2	0	3	7
Облік заборгованості з дисциплін	0,05	7	3	6	6	5
Формування рейтингу успішності	0,20	0	0	0	7	7
Візуалізація журналу успішності	0,01	6	6	7	7	6
Формування звітів журналу успішності	0,09	5	7	7	5	7
Інтегральний показник Q _j		2,75	3,57	3,73	5,58	6,72

Виконаємо побудову радіальної діаграми інтегрального показника ефективності для наочності отриманих результатів (рис. 3.13).

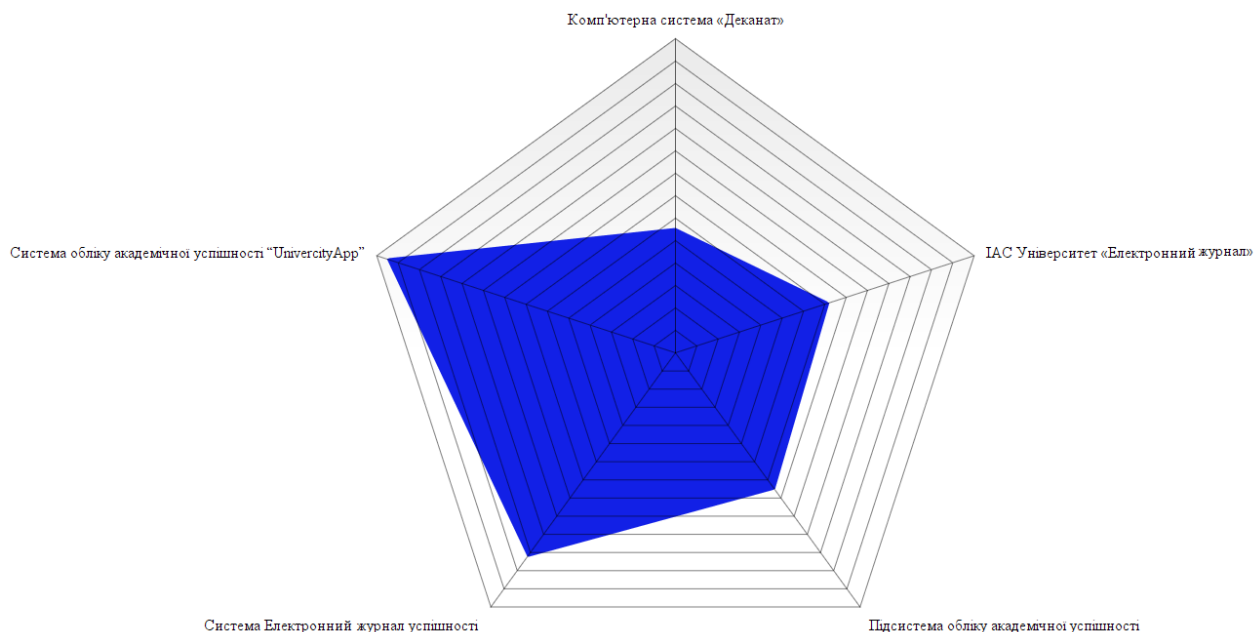


Рисунок 3.13 – Радіальна діаграма інтегральних показників

Як показав порівняльний аналіз, інтегральний показник ефективності винайдений на порівняння можливостей обліку найпливовіших критеріїв успішності найбільший в розробленій системі на основі методів, що вдосконалені у роботі.

Висновки до третього розділу

Для проведення порівняльного аналізу розділі проведено розробку автоматизованої системи обліку успішності. Визнчено функціональні та нефункціональні вимоги програмного продукту, спроектовано архітектуру системи. Проведено зовнішнє проектування веб-застосування, тобто розглянуто, який інтерфейс матиме система.

В ході виконання роботи було проведено тестування працездатності системи з метою перевірки готовності застосунку до використання. Перевірено та продемонстровано роботу основних функцій продукту та виявлено, що всі прецеденти виконуються правильно.

Для проведення дослідження ефективності розроблених методів обліку виконано порівняльний аналіз за допомогою аналітичною ієрархічною процедури Т. Сааті. В ході порівняння винайдено інтегральні показники, що підтверджують ефективність розроблених методів обліку.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра було досліджено методи підвищення обліку успішності студентів. Після проведеного аналітичного існуючих наукових досліджень та програмних розробок виділено критерії, які впливають на контроль успішності та запропоновано методи їх обліку.

На базі розглянутих моделей обліку побудовано універсальну математичну модель обліку успішності. Вона базується на бально-рейтинговій та накопичувальній системах обліку. В ході розробки моделі визначено найважливіші аспекти, які дозволяють всебічно враховувати та керувати результатами навчальної діяльності студентів. Для кожного з аспектів запропоновано шляхи їх обліку. На основі запропонованих методів спроектовано алгоритми обліку практичної частини курсу, алгоритм обліку курсових робіт та проектів, а також алгоритм розрахунку рейтингового показника для формування стипендіального рейтингу та поточних звітів про тривання навчально-процесу. Важливою особливістю розроблених алгоритмів є можливість гнучкого налаштування процесу обліку.

Для перевірки винайдених рішень було спроектовано та реалізовано проект системи автоматизованого обліку успішності у вигляді електронного журналу. Під час розробки системи:

- Розроблено архітектуру веб-застосування. При проектуванні системи було обрано загальну архітектуру «клієнт-сервер». Також було створено діаграми класів для всі проектів системи для наочної візуалізації відношень між класами
- Спроектовано базу даних для зручного зберігання інформації про всі аспекти системи обліку. Для пришвидшення доступу до даних було розглянуто методи оптимізації баз даних та застосовано для даного проекту.
- Визначено, яким чином будуть реалізовані методи обліку.

- Розроблені функції налаштування автоматичного обліку успішності навчально-виховного процесу.
- Розроблено автоматичне складання стипендіального рейтингу.
- Спроектовано адаптивний інтерфейс для системи.

Усі поставлені вимоги вдалося реалізувати завдяки ефективному використанню та надійній інтеграції технологій і програмних засобів розробки.

Кодування настільного застосування і веб-застосування було реалізовано на сучасних мовах програмування C#.

Після реалізації системи на її базі проведено порівняльний аналіз із існуючими системами обліку. Під час аналізу визначено, що система дозволяє дуже ефективно враховувати всі критерії, які визначають успішність. Виведення підсумкової оцінки стало значно легше, пришвидшено формування стипендіального рейтингу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Електронний журнал обліку успішності слухачів як засіб раціоналізації інформаційних зв'язків об'єктів навчального процесу [Електронний ресурс] / Мізюк В.А. - Режим доступу: <http://dspace.idgu.edu.ua/xmlui/bitstream/handle/123456789/476>
2. Особливості розробки та використання підсистеми обліку академічної успішності студентів [Електронний ресурс] / О. П. Мулярчук, Ю. С. Антонов - Режим доступу: <https://jvestniksss.donnu.edu.ua/article/view/6182>
3. Розробка інформаційної системи обліку успішності студентів на основі хмарних технологій [Електронний ресурс] / Чуйко О.И. та Белозерова С.И - Режим доступу: <http://naukovedenie.ru/index.php?p=vol7-5>
4. Система обліку відвідуваності й успішності студентів – як засіб моніторингу рівня знань у студентів [Електронний ресурс] / Васильєв В.М - Режим доступу: <https://journals.indexcopernicus.com/search/article?articleId=1545125>
5. Пакет програм "Деканат"[Електронний ресурс] – Режим доступу: <http://www.politek-soft.kiev.ua/ru/index.php?do=products&product=deanery>
6. Положення про електронний журнал успішності полтавського державного медичного університету [Електронний ресурс] – Режим доступу: <https://www.pdmu.edu.ua/storage/department-npr>
7. Автоматизація оцінювання знань студентів у кредитно-модульній системі болонського процесу [Електронний ресурс] / І. М. Кульчицький, Т.В. Магеровська, В.В. Сенік - Режим доступу: <https://journal.ldubgd.edu.ua/index.php/Visnuk/ submissionId=843>
8. Види обліку успішності учнів [Електронний ресурс] - Режим доступу: <http://um.co.ua/7/7-5/7-56316.html>
9. Кузьмінський А. І. Педагогіка вищої школи : навч. посібник / А. І. Кузьмінський. – К. : Знання, 2005. – 486 с

10. Орлов А.І. Розмаїття об'єктів нечислової природи // Науковий журнал КубДАУ. 2014. №102. [Електронний ресурс]. – Режим доступу: Орлов А. І. Розмаїття об'єктів нечислової природи // Науковий журнал КубДАУ. 2014. №102. URL: <http://cyberleninka.ru/article/n/mnogoobrazie-obektov-nechislovooy-prirody>
11. Бондаренко М.Ф., Семенець В.В., Білоус Н.В., Куцевич І.В., Білоус І.О. Технологія оцінювання тестів залежно від типу та рівня складності тестових завдань на основі інтегрованої моделі // International Book Series "Information Science and Computing". Sofia: Human Aspects of Artificial Intelligence. 2009. №12. P. 55-62.
12. Богдан Н. В. Балально-рейтингова система як метод оцінки якості освіти у вузі // Вісник ЮУрГУ. Серія: Освіта. Педагогічні науки 2010. №3 (179). З. 42-45.
13. Лежніна Л. В., Шишковський В. І. Бальна система оцінювання як фактор підвищення мотивації студентів до навчальної діяльності // Вісник ТДПУ. 2010. №10. С. 91-94.
14. Малишева Т. В. Практика впровадження бально-рейтингової системи оцінювання результатів навчання студентів вузу // Актуальні питання сучасної педагогіки: матеріали IV Міжнар. наук. конф. 2013. С. 170-173.
15. Помазанов А.В., Білоусова А.І., Васильєва А.О., Остроух А.В. Методика оптимізації баз даних //У світі наукових відкриттів Грінченка [Електронний ресурс] – Режим доступу: <https://naukarus.com/metodika-optimizatsii-baz-dannyh>
16. Індекс (баз даних) [Електронний ресурс] – Режим доступу: <https://znaimo.com.ua/>
17. Метаніт: розділ Баз даних [Електронний ресурс] – Режим доступу: <https://metanit.com/sql/mysql/>
18. Методи оптимізації запитів до бази даних [Електронний ресурс] – Режим доступу: <https://www.osp.ru/dbms/1998/03/13031583>

19. Програмування сервера бази даних за допомогою збережених процедур і тригерів [Електронний ресурс] – Режим доступу: <http://um.co.ua/13/13-1/13-107534.html>
20. Wiki Київський національний університет ім.Б. Грінченка [Електронний ресурс] – Режим доступу: <http://wiki.kubg.edu.ua>
21. Грамотна структура веб-застосунку [Електронний ресурс]. – Режим доступу:<http://prodex.ua/blog/gramotnaya-struktura-internet-magazina/>
22. Архітектура клієнт-сервер. Сховища даних.Реляційні бази даних як основа сховищ даних [Електронний ресурс]. – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2021>
23. Архітектура MVC [Електронний ресурс]. – Режим доступу: <https://github.com/codedokode/pasta/blob/master/arch/mvc.md>
24. Метаніт: розділ веб-розробки [Електронний ресурс] – Режим доступу: <https://metanit.com/web/>
25. Прохоренок Н. HTML, JavaScript, PHP и MySQL. Джентельменский набір Web-мастера. – СПб.: БХВ-Петербург, 2009. – 880 с.
26. Кузнецов М. Практика створення Web-сайтів./ Кузнецов М., Симдянов И. – 2-е изд. перераб. и доп. – СПб.: БХВ-Петербург, 2008.– 1264 с.
27. DevDotNet: Що нового в .NET 5.0 [Електронний ресурс] – Режим доступу: <https://devdotnet.org/post/chto-novogo-v-net5/>
28. Microsoft SQL Server: Flexberry PLATFORM Documentation[Електронний ресурс] – Режим доступу: https://flexberry.github.io/ru/gbt_mssql.html
29. Метаніт: розділ мови програмування C# та платформі .NET [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp//>
30. Гленфорд Дж. Майерс. The Art of Software Testing 3rd Edition / Гленфорд Дж. Майерс. – Wiley&Sons, Inc., Hoboken, New Jersey. – 151 с.
31. Остервальдер О. Value Proposition Design: How to Create Products and Services Customers Want / Олександр Остервальдер. – Москва: Альпина, 2015. – 604 с

32. Богушенков А.С., Рибанов А.А. Розробка та дослідження алгоритмів автоматизованої системи обліку та пошуку інформації з пакетів труб на основі технології QR-коду // Молодий учений. 2015. №4 (84). З. 47-52.
33. Рибанов А.А. Визначення вагових коефіцієнтів складності тем навчального курсу на основі алгоритму Сааті // Педагогічні виміри. 2014. №4. С. 21-28.