

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ім. Ігоря Сікорського**

Навчально-науковий комплекс «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Кафедра Системного проектування
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ А.І.Петренко
(підпис) (ініціали, прізвище)

“ ___ ” _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

з напрямку підготовки

6.050101 Комп'ютерні науки
(код і назва)

на тему: Автоматизована система «Документообіг кафедри». База даних та інтерфейс користувача

Виконав (-ла): студент (-ка) IV курсу, групи ДА-32
(шифр групи)

Бугайов Олександр Сергійович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

доцент, к.т.н., Безносик О.Ю.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант

економічний

(назва розділу)

доцент, к.е.н., Рощина Н.В.

(посада, вчене звання, науковий ступінь, прізвище, ініціали)

_____ (підпис)

Рецензент

_____ (посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Нормоконтроль

старший викладач Бритов О.А.

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2017 року

**Національний технічний університет України
«Київський політехнічний інститут»
ім. Ігоря Сікорського**

Інститут (факультет) ННК «Інститут прикладного системного аналізу
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050101 Комп'ютерні науки
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ А.І.Петренко
(підпис) (ініціали, прізвище)

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту
Бугайову Олександр Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система «Документообіг кафедри». База даних та інтерфейс користувача

керівник роботи Безносик Олександр Юрійович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи _____ навчальний та робочий навчальний плани, форма КЗ, вимоги до їх оформлення, система управління базами даних - MySQL, технології використані при реалізації клієнтської частини - JSP, HTML, CSS, JavaScript, jQuery, JSON.

4. Зміст роботи 1. Аналіз документації, що ведеться кафедрою

2. Розробка бази даних для збереження інформації для забезпечення документації

3. Розробка інтерфейсу користувача системи

4. Проведення функціонально-вартісного аналізу даної розробки

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економіко-організаційний	Рощина Н.В., доцент		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	1.03.2017	
2	Збір інформації	15.03.2017	
3	Розробка варіантів реалізації та вибір варіанту розробки	28.03.2017	
4	Реалізація бази даних та наповнення тестовою інформацією	10.04.2017	
5	Перевірка правильності взаємодії бази даних з серверною частиною	18.04.2017	
6	Розробка інтерфейсу користувача	15.05.2017	
7	Оформлення дипломної роботи	28.05.2017	
8	Проходження нормоконтролю	8.06.2017	
9	Отримання допуску до захисту та подача роботи в ДЕК	15.06.2017	

Студент

(підпис)

Бугайов О.С.

(ініціали, прізвище)

Керівник роботи

(підпис)

Безносик О.Ю.

(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломної роботи.

АНОТАЦІЯ

до бакалаврської дипломної роботи Бугайова Олександра Сергійовича
на тему «Автоматизована система «Документообіг кафедри». База даних та
інтерфейс користувача»

У роботі розглядається побудова компонентів для автоматизованої системи документообігу кафедри.

Метою роботи є дослідження проблем побудови та реалізація компонентів автоматизованої системи: бази даних та інтерфейсу користувача для взаємодії з серверною частиною.

При виконанні роботи було досліджено можливі інструменти та шляхи вирішення даної проблеми.

Результатом роботи є побудовані компоненти системи, опис процесу розробки цих компонентів: реалізована база даних для збереження необхідної інформації, побудовано інтерфейс користувача для керування системою.

Розробка бази даних здійснювалася засобами системи управління базами даних MySQL, а розробка інтерфейсу – мовами HTML, CSS та технологіями JSP та jQuery.

Загальний обсяг роботи 78 сторінок, 29 рисунків, 7 таблиць, 9 використаних джерел та 1 додаток на 1 сторінці.

Ключові слова: документообіг, клієнтська частина, навчальний план, реляційна база даних, форма К-3, CSS, HTML, jQuery, jTable, MySQL.

АННОТАЦИЯ

к бакалаврской дипломной работе Бугаёва Александра Сергеевича
на тему: «Автоматизированная система «Документооборот кафедры». База
данных и интерфейс пользователя»

В работе рассматривается построение компонентов для автоматизированной системы документооборота кафедры.

Целью работы является исследование проблем построения и реализация компонентов автоматизированной системы: базы данных и пользовательского интерфейса для взаимодействия с серверной частью.

При выполнении работы было исследовано возможные инструменты и пути решения проблемы.

Результатом работы являются построенные компоненты системы, описание процесса разработки этих компонентов: реализованная база данных для сохранения необходимой информации, построен пользовательский интерфейс для управления системой.

Разработка базы данных осуществлялась средствами системы управления базами данных MySQL, а разработка интерфейса – языками HTML, CSS и технологиями JSP и jQuery.

Общий объем работы 78 страниц, 29 рисунков, 7 таблиц, 9 использованных источников, 1 приложение на 1 странице.

Ключевые слова: документооборот, клиентская часть, учебный план, реляционная база данных, форма К-3, CSS, HTML, jQuery, jTable, MySQL.

ANNOTATION

on Buhaiov Oleksandr bachelor's degree

thesis: "Automated system "Document management of department". Database and user interface"

This work covers the process of creating components of automated system of department's document management.

Discovering problems of designing and realizing components of automated system: database and user interface is the main purpose of the work.

There were discovered available instruments and ways of overcoming of problems during the completion of graduate work.

Results of the work are following: built components of the system and their build description, complete database and user interface.

Development of database was complete by means of RDBMS MySQL, and interface design by means of HTML and CSS languages and JSP, and jQuery technologies.

Total amount of work is 78 pages, 29 pictures, 7 tables, 9 sources, 1 addition on 1 page.

Key words: document management, client-side, curriculum, relational database, K-3 form, CSS, HTML, jQuery, jTable, MySQL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	10
ВСТУП	11
1 АНАЛІЗ НОРМАТИВНИХ ДОКУМЕНТІВ КАФЕДРИ	13
1.1 Вступ.....	13
1.2 Аналіз навчального плану	14
1.3 Аналіз робочого навчального плану	16
1.4 Аналіз форми К-3	18
1.5 Висновок щодо аналізу документів	19
2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ СИСТЕМИ	20
2.1 Вступ.....	20
2.2 Огляд та вибір типів БД, СУБД.....	21
2.2.1 Класифікації баз даних	21
2.2.2 Сучасні системи управління базами даних	24
2.3 Розробка архітектури БД системи.....	27
2.3.1 Аналіз необхідного для генерації документів обсягу даних	27
2.3.2 Абстрактна модель даних.....	30
2.4 Реалізація таблиць БД.....	32
2.4.1 Таблиця «curriculum».....	32
2.4.2 Таблиця «practice».....	34
2.4.3 Таблиця «state_certification».....	35
2.4.4 Таблиця «diploma_preparation»	35
2.4.5 Таблиці «type_of_work» та «department»	37

	8
2.4.6 Таблиця «academic_group»	38
2.4.7 Таблиці «form_of_education» та «qualification»	39
2.4.8 Таблиця «specialization»	40
2.4.9 Таблиця «academic_subject»	41
2.4.10 Таблиця «curriculum_subject»	42
2.4.11 Таблиця «dict_subjects»	43
2.4.12 Таблиця «subject_type»	45
2.4.13 Таблиці «section» та «cycle»	46
2.4.14 Таблиці «control» та «dict_control»	47
2.4.15 Таблиця «study_load_results»	48
2.5 Висновки щодо розробки БД системи	50
3 РОЗРОБКА ІНТЕРФЕЙСУ КОРИСТУВАЧА.....	51
3.1 Вступ.....	51
3.2 Аналіз можливих реалізацій	51
3.3 Реалізація інтерфейсу користувача	52
3.4 Висновки щодо реалізації інтерфейсу	55
4 ЕКОНОМІКО-ОРГАНІЗАЦІЙНИЙ РОЗДІЛ.....	56
4.1 Вступ.....	56
4.2 Постановка задачі техніко-економічного аналізу.....	57
4.2.1 Обґрунтування функцій програмного продукту.....	58
4.2.2 Варіанти реалізації основних функцій.....	59
4.3 Обґрунтування системи параметрів програмного продукту	61
4.3.1 Опис параметрів	61
4.3.2 Кількісна оцінка параметрів.....	62

	9
4.3.3 Аналіз експертного оцінювання параметрів	63
4.4 Аналіз рівня якості варіантів реалізації функцій.....	67
4.5 Економічний аналіз варіантів розробки програмного продукту	68
4.6 Вибір кращого варіанта ПП техніко-економічного рівня.....	73
4.7 Висновки	74
ВИСНОВКИ.....	75
ПЕРЕЛІК ПОСИЛАНЬ.....	77
ДОДАТОК А.....	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД	– база даних.
ВНЗ	– Вищий Навчальний Заклад.
ЕОМ	– електронна обчислювальна машина.
СУБД	– система управління базами даних.
МКР	– модульна контрольна робота.
ОКР	– освітньо-кваліфікаційний рівень.
НП	– навчальний план.
РНП	– робочий навчальний план.
КР	– курсова робота.
КП	– курсовий проект.
ДКР	– домашня контрольна робота.
СРС	– самостійна робота студентів.
ПП	– програмний продукт.
ECTS	– European Credit Transfer and Accumulation System (Європейська система переведення і накопичення кредитів).
XML	– eXtensible Markup Language (розширювана мова розмітки).
JSON	– JavaScript Object Notation (об'єктний запис JavaScript).
JSP	– Java Server Page technology.
UX	– User experience (користувацький досвід).

ВСТУП

Діяльність будь-якої установи регулюється та залежить від певного набору відповідних документів, які покликані структурувати та контролювати усі робочі процеси, що відбуваються. Для вищого навчального закладу, зокрема, для кафедри факультету/інституту важливою складовою документації – є нормативні документи, які регулюють проведення навчальної роботи та її контроль, а також використовуються при складанні статистичної звітності, яка впливає на подальший перебіг процесів як кафедри, так і вищого навчального закладу.

Така документація кафедри включає в себе наступні документи:

- навчальний план;
- робочий навчальний план;
- форма К-3;
- розподіл навантаження на викладачів;
- інші.

Процес створення такої документації – складний та трудомісткий та вимагає значних людських ресурсів від особи, що займається її веденням. Великі обсяги інформації, особливості її подання – це ті фактори, що підвищують можливість виникнення помилок та суперечностей. Це, та той факт, що у більшості документів використовується однотипна або одна й та ж сама інформація, приводить до думки про можливість створення програмного продукту, що дозволить автоматизувати більшу частину роботи з документами, що розглядаються. Застосування бази даних для збереження інформації допоможе структурувати її, запобігти суперечностям та помилкам, спричиненим людським фактором. Програмна частина забезпечить правильне та чітке дотримання норм наповнення документів, коректного прорахунку даних, що потребують додаткового обчислення. Інтерфейс користувача дозволить легко

редагувати інформацію, що знаходиться у БД без використання відповідних засобів СУБД, зробіть роботу з даними більш наочною та інтуїтивно зрозумілою.

Отже, темою даної дипломної роботи є розробка компонентів автоматизованої системи генерації навчальної документації – бази даних та інтерфейсу користувача.

1 АНАЛІЗ НОРМАТИВНИХ ДОКУМЕНТІВ КАФЕДРИ

1.1 Вступ

Діяльність кафедри – комплексний процес, що містить в собі безліч складових, які повинні бути задокументовані для збереження та створення відповідної звітності. Усі викладачі в повсякденній роботі стикаються з такими документами, а вчені секретарі також розробляють навчальні, робочі навчальні плани, звітність щодо розподілу навантаження на викладачів. Усі ці документи мають спільну інформаційну базу, тобто спираються у багатьох аспектах на однакові дані. Таким чином, при розробці інформаційної бази для системи, що розглядається, недоцільно використовувати окремі бази даних для кожного компонента/модуля, оскільки це б призвело до дублювання інформації у різних базах, спричинило помилки чи розбіжності в інформації, що недопустимо при створенні офіційних документів кафедри. Тому, натомість, краще створити єдину базу даних, яка буде використовуватися різноманітними компонентами системи, створеними як на цьому етапі, так і в майбутньому. Це дозволить значно спростити пошук та редагування інформації, а також скоротить час наповнення системи потрібними даними.

Для повноцінної роботи початкового функціоналу система повинна мати доступ до наступної інформації:

- перелік усіх необхідних даних по предметам, що викладаються в рамках навчальних планів;
- дані про діяльність кафедри (наявні напрямки підготовки, форми навчання та інше);
- дані про усі групи, що навчаються на кафедрі, а також групи, для яких викладачами кафедри ведуться предмети.

Також, система повинна бути побудована таким чином, щоб у майбутньому її можна було розширити додатковою інформацією та функціоналом.

1.2 Аналіз навчального плану

НП – один з основних документів ВНЗ, згідно з яким проводиться організація навчального процесу. Він містить у собі базову інформацію щодо навчання студентів відповідного освітнього ступеня, форми навчання, спеціальності та спеціалізації, таку як:

- графік навчального процесу;
- бюджет часу;
- практики, що проходяться студентами під час навчання;
- інформація про державну атестацію випускників;
- план навчального процесу, що містить в собі:
 - перелік навчальних дисциплін, що викладаються в рамках даного освітнього ступеня, спеціальності, спеціалізації;
 - розподіл за семестрами екзаменів, заліків, курсових робіт/проектів відповідно до кожної дисципліни;
 - кількість кредитів ECTS, що виділяються для дисциплін;
 - кількість аудиторних годин, та годин, виділених на самостійне вивчення дисципліни;
 - розподіл аудиторних годин за курсами та семестрами.

Отже, робочий план складається окремо для кожного ОКР, та форми навчання.

Відповідно до наведеного переліку інформації, що представлена НП, документ містить таку саму структуру, а саме – складається з шапки та п'яти інформаційних блоків:

- I. Графік навчального процесу.
- II. Зведені дані про бюджет часу.
- III. Практика.
- IV. Атестація випускників.
- V. План навчального процесу.

Графік начального процесу визначає роки навчання за відповідною програмою із зазначенням для кожного з них обсягів теоретичного навчання, екзаменаційних сесій, практик, дипломного проектування, складання державного екзамену, захисту дипломного проекту (роботи), а також канікул. Тривалість всіх цих заходів у тижнях зазначається у другій таблиці – зведених даних про бюджет часу.

У розділі інформації про практику надається назва практики, семестр проведення та тривалість у тижнях. Атестація випускників описується назвою навчальної дисципліни, формою державної атестації, та семестром її проведення.

План навчального процесу розділяє усі навчальні дисципліни, що викладаються на два цикли підготовки: загальної, та професійної. В свою чергу, цикл загальної підготовки містить в собі:

- навчальні дисципліни природничо-наукової підготовки;
- навчальні дисципліни базової підготовки;
- навчальні дисципліни базової підготовки (за вибором студентів);
- навчальні дисципліни соціально-гуманітарної підготовки (за вибором студентів);
- дослідницький (науковий) компонент (за вибором студентів).

Цикл професійної підготовки включає в себе:

- навчальні дисципліни професійної та практичної підготовки;
- навчальні дисципліни професійної та практичної підготовки (за вибором студентів).

Також у циклах професійної та загальної підготовки можуть бути внесені підрозділи для зазначення предметів, що стосуються підготовки PhD (на момент написання роботи вони не були остаточно затверджені).

Відповідно для кожної дисципліни надається перелік наступної інформації:

- шифр;
- назва;
- розподіл за семестрами: екзаменів, заліків, курсових робіт (проектів);

- кількість кредитів ECTS;
- розподіл годин між аудиторними (лекції, практичні, лабораторні) та СРС;
- розподіл аудиторних годин на тиждень за курсами та семестрами.

У кінці документу наводиться зведена інформація про загальну кількість годин, кількість екзаменів, заліків, курсових проектів та курсових робіт [1].

У окремій таблиці зазначаються позакредитні навчальні дисципліни, обсяги яких не входять до загальної програми підготовки – фізичне виховання та військова підготовка.

1.3 Аналіз робочого навчального плану

РНП – документ, що деталізує інформацію, зазначену у НП, в проекції на один навчальний рік для відповідного освітнього ступеня, форми навчання, спеціальності та спеціалізації. У робочому навчальному плані вказується, для якої групи він призначений.

У РНП конкретизується інформація за дисциплінами, що встановлені у навчальному процесі. Розподіл на цикли підготовки та підрозділи аналогічний до такого у НП. Натомість додається більш детальна інформація про предмет:

- кафедра, відповідальна за викладання;
- контрольні заходи та їх розподіл за семестрами – на відміну від НП, тут додаються наступні види контролю:
 - модульні (тематичні) контрольні роботи;
 - розрахунково-графічні, розрахункові та графічні роботи;
 - ДКР;
 - реферати.
- кількість годин аудиторних занять на тиждень за семестрами деталізується зазначенням розподілу годин на лекції, практичні, та лабораторні години.

Якщо на курсі проходить практика чи держава атестація, то таблиці з відповідною інформацією можуть бути включені в РНП. Проте в даному випадку

для практики вказується її вид, термін проведення, тривалість в тижнях, та семестр. Державна атестація визначається формою та терміном проведення [1].

Також, в робочому плані присутні таблиці розподілу годин по підготовці та захисту дипломного проекту (роботи) (рис. 1.1) та розподілу годин з комплексного державного екзамену (рис. 1.2).

РОЗПОДІЛ ГОДИН ПО ПІДГОТОВЦІ ТА ЗАХИСТУ ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

Вид роботи	Норма в годинах на 1 студента	Кафедра	Кількість студентів		Всього годин	
			Б	К	Б	К
Керівництво						
Консультування						
Рецензування						
ДЕК	0,5 x d					
Всього годин	24,5*		Всього годин			

Рис. 1.1 – Таблиця розподілу годин по підготовці та захисту дипломного проекту (роботи)

РОЗПОДІЛ ГОДИН З КОМПЛЕКСНОГО ДЕРЖАВНОГО ЕКЗАМЕНУ

Вид роботи	Норма в годинах	К-ть дисциплін	Кафедра	Кількість груп		Кількість студентів		Всього годин	
				Б	К	Б	К	Б	К
Консультування дисциплін, що внесені в екзамен	2 x Г								
усний екзамен	0,5хd на 1 студ.								
письмовий екзамен	4 x d x Г+0,5 на 1 студ.								
				Всього годин					

Рис. 1.2 – Таблиця розподілу годин з комплексного державного екзамену

1.4 Аналіз форми К-3

Форма № К-3-К(Б) являє собою документ для обсягу навчального навантаження кафедри відповідно до прийнятих навчальних планів.

Навчальне навантаження – розподіл годин, що витрачається викладачами даної кафедри на проведення наступних видів робіт:

- лекції;
- практичні заняття;
- лабораторні заняття;
- індивідуальні заняття;
- екзамени;
- заліки;
- контрольні (модульні, тематичні) роботи;
- курсові проекти;
- курсові роботи;
- РГР, РР, ГР;
- ДКР;
- реферати;
- консультації.

Усі вищезазначені складові навантаження обчислюються за чітко визначеними формулами, які використовують дані, зібрані на основі робочих навчальних планів, встановлених на поточний навчальний рік, а саме все те, що фігурує у назвах складових навантаження. Окрім цього вводяться такі показники, як кількість бюджетних та контрактних груп та підгруп, а також студентів у бюджетних та контрактних групах. Навантаження обраховується для наборів навчальних планів, згрупованих за формою навчання та джерелом фінансування, наприклад, «денна, держбюджет», «денна, контракт», «заочна, держбюджет», «заочна, контракт».

Документ також містить додаткові сторінки, але вони являють собою

зведені дані з описаних вище сторінок з порахованими показниками навчального навантаження.

1.5 Висновок щодо аналізу документів

Проаналізувавши такі документи, як НП, РНП, форму розрахунку навчального навантаження К-3, можна зробити висновок, що усі вони в багатьох випадках спираються на однакові дані.

Так, наприклад, навчальний та робочий навчальний плани містять однакову інформацію стосовно дисциплін, що викладаються, а в РНП спільна інформація доповнюється більш детальним описом, що зрозуміло враховуючи специфіку цього документу. Таким чином, при розробці моделі даних слід врахувати цей факт, щоб спроектувати модель таким чином, що одні й ті самі записи можливо було використовувати для побудови обох документів.

Далі, очевидним є прямий зв'язок між робочим навчальним планом та обчисленням навчального навантаження, відображеним у формі К-3. Наприклад, зазначення кафедри, що викладає предмет в РНП важливе для ідентифікації потрібних для розрахунку предметів у К-3, тобто тих, що викладаються сам на кафедрі, для якої складається дана форма. Також, усі конкретизовані дані стосовно контрольних заходів, заліків, екзаменів та інших використовуються для обрахунку навчального навантаження. Окрім цього, робочі плани містять інформацію про групи, що теж використовується у К-3, як в записі з предметом, так і при зазначенні кількості студентів в групах та підгрупах. Тобто, очевидним є необхідність створення такої моделі даних для груп, яка б об'єднувала в собі вищезазначену інформацію, тим самим забезпечивши багатофункціональність та захист від дублювання чи виникнення суперечностей інформації.

2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ СИСТЕМИ

2.1 Вступ

Однією з цілей даної дипломної роботи є розробка бази даних, яка буде забезпечувати стабільну та коректну роботу серверної частини шляхом надання актуальної інформації для генерації відповідних документів кафедри.

Як вже було зазначено, з того факту, що більшість документів використовують однорідні, або зовсім однакові дані, було зроблено висновок про доцільність створення єдиної бази даних, яка б забезпечила своїм наповненням всю необхідну інформацію для генерації повного спектру документів, що використовуються на кафедрі. Таке рішення допоможе запобігти появі помилок при наповненні БД, а також спростить задачу створення відповідного інтерфейсу користувача для редагування наявних даних.

Таким чином, на даному етапі постає задача вибору типу бази даних, який буде більш доцільним для поставленої задачі, а також відповідного інструментарію для її розробки, який будуть задовольняти наступним вимогам:

- простота у використанні;
- безкоштовність;
- наявність достатньої кількості матеріалів з підтримкою у відкритому доступі.

Наступним етапом є проектування та реалізація БД, яка буде використовуватися серверною частиною системи для зберігання даних. При цьому мають бути враховані усі спільні та відмінні риси документів, що будуть генеруватися, а також повинен бути закладений потенціал для подальшого розширення структури БД, при появі необхідності доповнення системи новим функціоналом.

2.2 Огляд та вибір типів БД, СУБД

2.2.1 Класифікації баз даних

База даних – сукупність даних, що організовані у відповідності до концептуальної структури, яка описує характеристики цих даних та відношення між ними, при чому така сукупність даних, яка підтримує одну або більше областей використання. У джерелах наводяться класифікації за декількома ознаками.

За ступенем розподіленості існують **централізовані** та **розподілені** БД. Централізовані – повністю підтримуються на одній ЕОМ. Розподіленими є ті, компоненти яких розміщуються в різних вузлах комп'ютерної мережі у відповідності до певного критерію. Очевидно, що розподілені БД є більш надійними, оскільки при відмові одного з вузлів у мережі інші залишаються дієздатними, а при умові, що кожний вузол буде мати резервні копії даних з інших вузлів, система буде мати високий рівень відмовостійкості. Проте, такі БД потребують значно більших ресурсів для підтримки роботи, а в контексті діяльності кафедри, потрібно враховувати обмеженість ресурсів, як людських, так і матеріальних. Також при взаємодії компонентів через мережу можна очікувати меншу швидкодію, порівняно з централізованою системою. Саме тому для вирішення даної задачі можна зупинитися на централізованому типі БД, що значно спростить підтримку та використання. Така БД зможе бути використана або на сервері кафедри, або безпосередньо встановлена на ЕОМ людини, відповідальної за укладання документів, що генеруються автоматизованою системою, яка розробляється. А надійність бази даних можна забезпечити регулярним створенням резервних копій – дампів.

За середовищем постійного збереження виокремлюють БД, що зберігаються у **вторинній**, тобто постійній пам'яті комп'ютера (як правило жорсткий диск), в **оперативній** пам'яті, та **третинній** – змінних носіях інформації. В даному випадку, оскільки для підтримки документації важливим

фактором є надійність збереження даних, оптимальним варіантом слід вважати традиційну модель зберігання – у вторинній пам'яті. Вона програє у швидкодії оперативній пам'яті, проте менш вибаглива до характеристик ЕОМ, на якій буде працювати БД.

Ще однією принциповою класифікацією баз даних є класифікація за моделлю даних. Модель даних визначає логічну структуру БД та визначає основні принципи відносно того, як дані можуть зберігатися, організовуватись, та яким чином з ними можна працювати [2].

У **ієрархічній** моделі БД дані подаються у вигляді деревовидної структури. Вони зберігаються як записи, що поєднуються між собою за допомогою посилань (рис. 2.1). Запис – це набір полів, кожне з яких містить одне значення. Тип сутності запису визначає, які поля запис має.

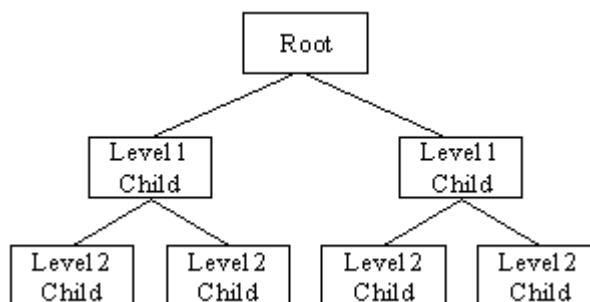


Рис. 2.1 – Ієрархічна модель даних

Запис у ієрархічній моделі БД відповідає рядку (кортежу) у реляційній моделі, а тип сутності відповідає таблиці.

Ієрархічна модель строго визначає, що кожний запис-нащадок, має тільки один запис-предок, в той час як один предок може мати декілька нащадків. Щоб отримати дані з ієрархічної моделі *усе дерево необхідно обійти починаючи від кореневого вузла*. Ця модель визнана першою моделлю БД та створена ІВМ ще у 1960-х, тож багато спеціалістів вважає її морально застарілою [3].

У **реляційній** моделі використовується підхід оперування даними використовуючи структуру та мову, сумісну з логікою предикатів першого

порядку, вперше описаною Едгаром Коддом, в якій усі дані представлені в термінах кортежів, згрупованих у відношення. БД, що організовані в рамках даної моделі, називаються реляційними базами даних (рис. 2.2).

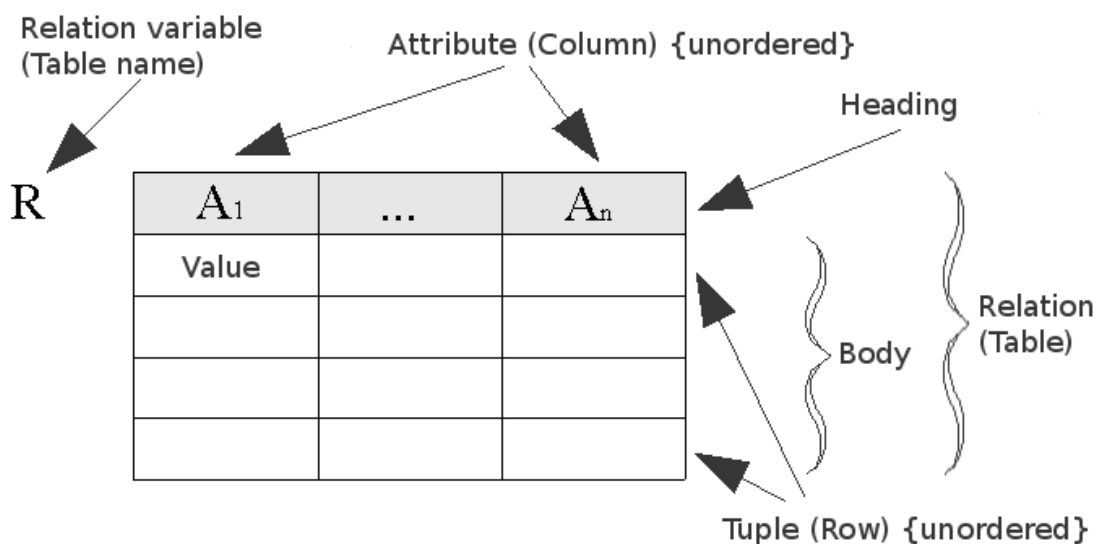


Рис. 2.2 – Реляційна модель

Метою реляційної моделі є надання декларативного методу визначення даних та запитів: користувачі чітко задають, яку інформацію містить БД, та яку вони хочуть отримати, залишаючи при цьому усі деталі реалізації на програмне забезпечення СУБД.

Більшість баз даних використовують мову SQL. Вона є стандартизованою, але різні реалізації в рамках певних СУБД можуть дещо відхилитися від цього стандарту з метою надання більших можливостей чи поліпшення швидкодії.

Слід зазначити, що дана модель організації даних у БД є наразі найбільш поширеною, та має безліч реалізацій, як з відкритим програмним кодом, так і комерційних, коштовних систем.

У **об'єктно-орієнтованій** моделі інформація представляється у вигляді об'єктів, так само, як і в об'єктно-орієнтованих мовах програмування. Це дозволяє значно спростити налагодження взаємодії між БД та ОО-мовою програмування, оскільки не потрібно використовувати ORM-фреймворки, чи

самостійно писати об'єкти для доступу к даним БД.

Для використання в рамках даної роботи було прийнято рішення обрати реляційну базу даних, оскільки даний формат має велику розповсюдженість, відповідно наявна велика кількість джерел інформації стосовно використання таких БД, а також наявна велика кількість СУБД під цей тип, тобто є можливість обрати оптимальний варіант для використання в системі, що розробляється.

2.2.2 Сучасні системи управління базами даних

На сьогоднішній день існує досить багато систем управління базами даних. Було розглянуто деякі з них.

PostgreSQL. Безкоштовна об'єктно-реляційна система управління базами даних. Вона є однією з найбільш просунутих СУБД, що в першу чергу орієнтується на повну відповідність стандартам та можливості розширення, тобто повністю намагається відповідати SQL-стандартам ANSI/ISO. З-поміж інших цю СУБД також виокремлює той факт, що вона має об'єктно-орієнтований функціонал та підтримує відповідні концепти.

Система PostgreSQL заснована на ядрі, створеному безліччю розробників. У подібних випадках розумно зосередитися на оснащенні системи новими можливостями, але не займатися оптимальним їх втіленням, оскільки у випадку виникнення необхідності завжди можна буде повернутися до оптимізації відповідних ділянок коду.

Також варто зазначити що ця система не достатньо оптимізована для вирішення повсякденних не дуже важких задач. Її використання більше передбачається у випадках, коли важливими виявляються підвищена надійність та підтримки об'єктних підходів до БД. У випадку, коли більш важливим є виконання простих операцій зчитування-запису, PostgreSQL показує не найкращі результати [4].

Oracle Database або **Oracle RDBMS.** Об'єктно-реляційна система управління базами даних компанії Oracle Corporation. Ця СУБД забезпечує

ефективне, надійне і безпечне управління даними таких критично важливих для бізнесу додатків, як онлайнові середовища, виконує масштабну обробку транзакцій (OLTP), сховища даних з високою інтенсивністю потоку запитів, а також ресурсоємні інтернет-додатки. Редакція Oracle Database Enterprise Edition надає інструментальні засоби і функції, що забезпечують відповідність вимогам сучасних корпоративних додатків в області доступності та масштабованості. Ця редакція містить всі компоненти Oracle Database, а також допускає розширення за допомогою придбання додаткових модулів та програм [5].

Система Oracle Database дозволяє звертатися до даних з будь-якого додатку, розробленого із застосуванням технологій Microsoft. NET, Visual Studio та веб-додатків. Основною умовою є лише наявність справних бібліотек, що дають змогу підключатися до серверу бази даних Oracle. Це комерційна СУБД, але є безкоштовна версія, яку можна без проблем скачати прямо з офіційного сайту компанії. Втім, варто зазначити, що безкоштовна версія має зменшений функціонал.

Microsoft SQL Server. Система Microsoft SQL Server відштовхується від концепції платформи даних Майкрософт: вона спрощує управління будь-якими даними в будь-якому місці і в будь-який момент часу. Вона дозволяє зберігати в базах даних інформацію, отриману з структурованих, напівструктурованих і неструктурованих джерел, таких як зображення та музика. У SQL Server є великий набір інтегрованих служб, які розширюють можливості використання даних: можна складати запити, виконувати пошук, проводити синхронізацію, робити звіти, аналізувати дані. Всі дані зберігаються на основних серверах, що входять до складу центру обробки даних. До них здійснюється доступ з настільних комп'ютерів і мобільних пристроїв. Таким чином, можна повністю контролювати дані незалежно від того, де вони збережені.

Система MS SQL Server дозволяє звертатися до даних з будь-якого додатку, розробленого із застосуванням технологій Microsoft. NET та Visual Studio, а також в межах сервісно-орієнтованої архітектури і бізнес-процесів –

через Microsoft BizTalk Server. SQL Server дозволяє створити надійну, продуктивну, інтелектуальну платформу, що відповідає всім вимогам по роботі з даними [6]. Ця система є комерційною, тобто потребує значних витрат на встановлення та налагодження роботи.

Система **MySQL**. Безкоштовна система управління базами даних. MySQL є власністю компанії Oracle Corporation, що отримала її разом з поглиненою Sun Microsystems, що здійснює розробку і підтримку програми. Розповсюджується під GNU General Public License або під власною комерційною ліцензією. Крім цього розробники створюють функціональність за замовленням ліцензійних користувачів.

MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, LAMP і в портативні збірки серверів Denver, XAMPP. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або віддалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми [7].

Як вже було зазначено раніше, СУБД, що обиралася, повинна задовольняти вимогам щодо простоти використання, безкоштовності та наявності достатньої кількості довідкового матеріалу. Тому, вибір було зроблено на користь MySQL. По-перше, дана СУБД є безкоштовним open-source продуктом, тобто не викличе жодного фінансового навантаження при використанні. По-друге, супутній продукт MySQL Workbench дозволяє просто взаємодіяти зі встановленою БД, надаючи можливість через зручний інтерфейс користувача виконувати будь-які маніпуляції з даними. І нарешті, ця СУБД є однією з найбільш розповсюджених, займаючи друге місце за популярністю у світі, поступаючись лише корпоративній СУБД від Oracle [8]. Саме тому в мережі наявна велика кількість матеріалів, як довідкових, включно з офіційним сайтом, так і навчальних, де розглядається широкий спектр проблем та шляхи їх вирішення.

Отже, для більш чіткого обґрунтування вибору СУБД, що буде

використовуватись, наводиться перелік переваг MySQL [9]:

- Відкритий вихідний код та безкоштовність.
- Продумана та швидкодіюча.
- Займає небагато дискового простору.
- Може бути легко встановлена на багатьох операційних системах, таких як Windows, Unix-like і інших.
- Враховуючи її розповсюдження в Мережі можна знайти безліч матеріалів, в яких розглядається вирішення певних питань та проблем, що виникають при роботі з нею.
- Добре підходить для невеликих застосунків.

2.3 Розробка архітектури БД системи

2.3.1 Аналіз необхідного для генерації документів обсягу даних

Метою даного пункту є визначення оптимального набору даних, якого буде достатньо для обслуговування серверної частини на етапі генерації будь-якого документа. Це важливо не лише з точки зору спрощення виконання поставленого завдання, але й тому що з меншим обсягом даних зменшується і час, витрачений на їх заповнення чи редагування. Також, як буде показано згодом, від деякої інформації можна абстрагуватися, оскільки вона не впливає жодним чином на процес генерування документації.

РНП

Розгляд документів варто розпочати з робочого навчального плану, оскільки саме цей документ має спільні дані як з НП, так і з формою К-3.

З першого погляду шапка документа містить багато визначальної інформації, як спеціальність, спеціалізація, освітній ступінь, форма навчання та інші. Але вся ця інформація не потребує виокремлення при визначенні РНП, оскільки у документі зазначається **група**, яка навчається за даним планом, тобто усю перераховану інформацію можна отримувати, посилаючись на зазначену

групу.

Ключовою інформацією РНП є перелік **навчальних дисциплін**. Тобто потрібно мати таке представлення сутності предмета, яке буде включати в собі усі необхідні для заповнення полів РНП дані. Частина з них може бути використана при заповненні НП, а усі дані з РНП, як вже зазначалося, переносяться до форми К-3 з урахуванням інформації про групи, яким викладається даний предмет. Також важливою частиною інформації про предмети є перелік **контрольних заходів**, що впроваджуються протягом семестру. Дані щодо контрольних заходів також використовуються на етапі створення форми К-3. Зв'язок між предметом та групою можна буде встановити через відповідний робочий навчальний план.

Наступною інформацією, що містить РНП є **практики, державна атестація, розподіл годин по підготовці та захисту дипломного проекту (роботи) а також розподіл годин з комплексного державного екзамену**. Для робочих планів, що відповідають четвертому року (курсу) навчання за освітнім ступенем «бакалавр», ці блоки заповнюються відповідною інформацією, таким чином вони також виокремлюються у сутності, що мають бути враховані при проектуванні структури БД.

Також варто пам'ятати, що кожний робочий план спирається на певний навчальний, тому варто передбачити певне посилення робочого навчального плану на відповідний навчальний.

Усі інші дані, якими наповнені структурні елементи документу доцільно залишити у шаблоні, що буде наповнюватися системою, а в кожний шаблон треба буде помістити відмітку про те, даними для якого курсу, спеціалізації, форми навчання потрібно заповнити сторінку.

НП

Навчальний план містить схожу інформацію, що стосується предметів, порівняно з РНП. Як було встановлено за допомогою аналізу оформлення навчальних планів – верхня частина (графік навчального процесу, зведені дані

про бюджет часу, практику та атестацію випускників) не змінюється зовсім, або змінюється не значно протягом тривалого проміжку часу. Тому виникає ідея залишити всі ці дані у шаблоні документу, який при потребі можна буде легко виправити у відповідності до нових стандартів, які можуть бути встановлені у майбутньому.

Інформація щодо освітнього ступеня, спеціальності, спеціалізації, форми навчання та інше також є надлишковою, оскільки робочі навчальні плани містять у собі цю інформацію, яка визначається групою, для якої план створено, а враховуючи те, що робочий навчальний план буде містити посилання на навчальний план, на який він спирається, немає потреби наділяти НП такими даними. Натомість, шаблон просто буде позначений міткою, яка буде визначати, даними з якого саме навчального плану слід наповнити документ.

Форма К-3

Даний документ, як вже зазначалося раніше, будується на основі інформації зібраної з усіх робочих навчальних планів, які використовуються у поточному році. Для його генерації не потрібні ніякі інші дані, окрім тих, що зібрані у всіх робочих планах на поточний навчальний рік. Крім цього потрібно враховувати нормативні вказівки щодо визначення типів груп та студентів за джерелом фінансування

При генерації цієї форми отримуються дані щодо **навчального навантаження** по кожній формі навчання та джерелу фінансування. Отримані чотири сторінки документу містять числові значення обрахованих показників навчального навантаження по кожному з тринадцяти параметрів (лекції, практичні заняття, лабораторні заняття, індивідуальні заняття, екзамени, заліки, контрольні та модульні роботи, курсові проекти та роботи, розрахунково-графічні роботи, ДКР, реферати та консультації). Ці дані потрібно зберігати, оскільки у подальшому вони будуть використанні у системі модулем розподілу та обчислення викладацького навантаження.

При цьому, при розробці архітектури БД можуть з'явитися додаткові

сутності для більш ефективного та лаконічного вирішення наявних завдань.

2.3.2 Абстрактна модель даних

Отже, на попередньому етапі були описані обсяги даних, що потрібні для створення відповідних документів. Для кращого усвідомлення задачі створення архітектури БД наводяться основні інформаційні сутності, що мають бути в ній присутні. На даному етапі виділені наступні сутності: навчальний план, робочий навчальний план, предмет, контрольний захід, кафедра, група, навчальне навантаження, практика, державна атестація, підготовка дипломного проекту (роботи), а також розподіл годин з комплексного державного екзамену. Для більшої наочності вони всі зображені на рис. 2.3 з позначенням видів взаємодії між ними.

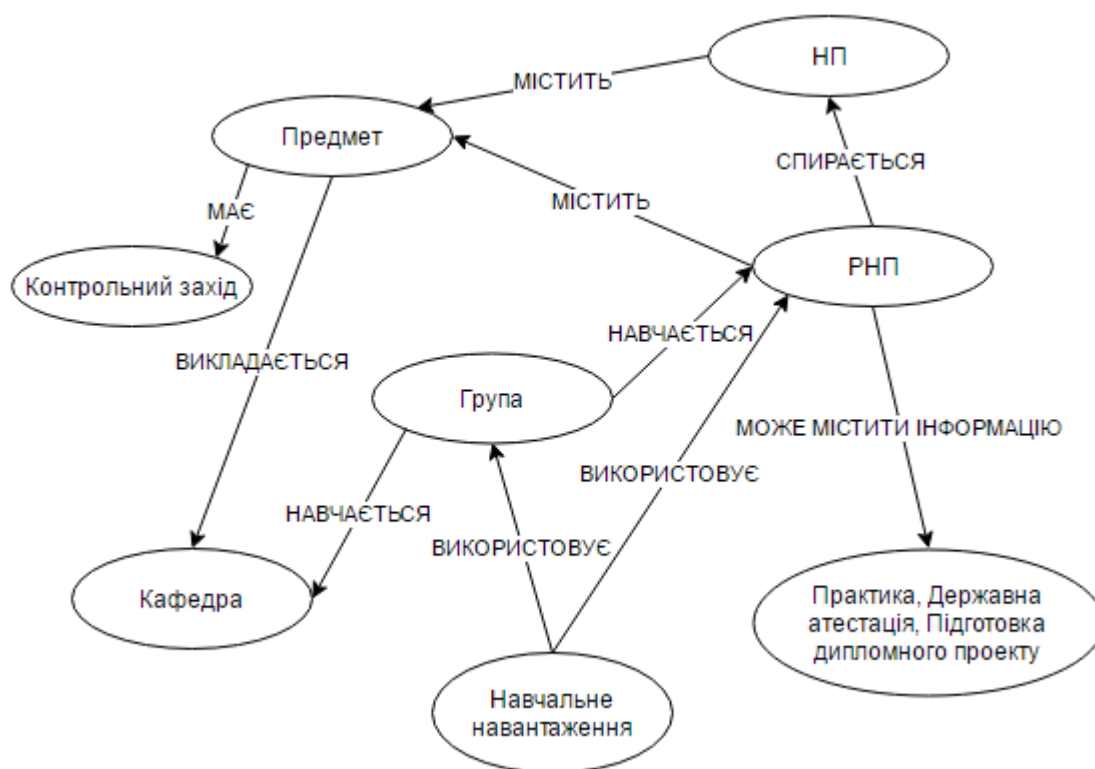


Рис. 2.3 – Абстрактна модель даних

На даній діаграмі (див. рис 2.3) зображено базове абстрактне представлення даних, що мають бути враховані в системі. З її допомогою можна

розкрити базові взаємозв'язки між даними.

Так, наприклад, видно, що робочий та навчальний плани містять в собі інформацію про предмети, які викладаються в їх рамках. Деталі тут не вказуються, оскільки це – абстракція, і вона покликана пояснити принципові моменти.

Робочий навчальний план спирається на відповідний навчальний, а також, як вже було зазначено раніше, якщо РНП відповідає четвертому року (курсу) навчання за освітнім ступенем «бакалавр», або другому курсу навчання за освітнім ступенем «магістр», то він також містить інформацію про практику, атестацію, підготовку диплому, ДЕ.

Кожен предмет має дані, про контрольні заходи, що здійснюються протягом семестру, або всього часу навчання. Також кожний предмет викладається певною кафедрою.

Група студентів навчається на певній кафедрі. Кожен рік ця група має певний робочий навчальний план, згідно з яким відбувається організація навчального процесу.

Навчальне навантаження, що обчислюється у формі К-3, для розрахунку повинно використовувати дані з РНП, які стосуються предметів, що викладаються на кафедрі, для якої визначається навантаження. Дані містять інформацію про розподіл аудиторних годин та контрольні заходи. Окрім цього в документі визначаються такі параметри як кількість бюджетних груп, підгруп, потоків. Для їх визначення необхідно мати доступ до інформації щодо груп, яким викладається даний предмет.

Отже в даному пункті розглянуто основні інформаційні сутності, які мають бути застосовані для зберігання інформації в системі. Слід зазначити, що при реалізації таблиць БД з'являться додаткові, більш деталізовані елементи.

2.4 Реалізація таблиць БД

Умовно таблиці, що складають всю БД для документообігу кафедри, можна поділити на два типи: таблиці-словники, та таблиці з інформацією, що може часто змінюватися, або доповнюватися.

Таблиці-словники найчастіше містять в собі статичну інформацію, яка не змінюється протягом довгого періоду часу. Найпростішим прикладом такої таблиці в даній базі даних будуть співвідношення «ключ – розшифрування». Вони використовуються для визначення певних даних, як наприклад, назва предмету, чи тип контролю.

Таблиці з даними, що змінюються загалом містять посилання на таблиці-словники та певну динамічну інформацію.

Загальну структуру розробленої БД з усіма зв'язками наведено у додатку А.

2.4.1 Таблиця «curriculum»

Дана таблиця відображає одну з основних сутностей в даній роботі – навчальний план. Таблиця побудована таким чином, щоб поєднувати в собі обидва типи планів – робочий та навчальний плани (рис. 2.4).

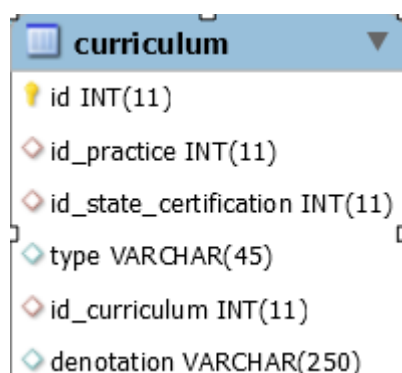


Рис. 2.4 Таблиця БД «curriculum»

Різниця між типом навчального плану встановлюється за допомогою поля «**type**», в якому вказується, який саме тип плану описується в даному записі

таблиці. Ці процеси відбуваються на серверній стороні, що не є предметом даного дослідження, тому детальний опис цього механізму опускається.

Загалом, таблиця містить наступні поля:

- **id** – INT(11) – дане поле є первинним ключем таблиці, що однозначно визначає НП/РНП;
- **id_practice** – INT(11) – вторинний ключ, тобто посилання на відповідний запис у таблиці «**practice**», яка визначає практику, що проводиться в рамках певного навчально плану. Відповідно, дане поле використовується (не має значення NULL), коли запис відповідає РНП.
- **id_state_certification** – INT(11) – вторинний ключ на запис у таблиці «**state_certification**», яка визначає державну атестацію, вказану у відповідному робочому плані. Отже, це поле, так само як і попереднє, використовується при записі стосовно РНП.
- **id_curriculum** – INT(11) – вторинний ключ на цю саму таблицю («**curriculum**»), який використовується, коли запис відповідає РНП. Позначає НП, на якому план заснований.
- **type** – VARCHAR(45) – символічне поле, що містить в усіх записах один з двох варіантів запису : «**curriculum**» або «**workplan**», які означають, відповідно, що даний запис стосується ПН або РНП.
- **denotation** – VARCHAR(45) – символічне поле, що містить назву навчального чи робочого плану, наприклад, «**БАК-КН-СП**» - навчальний план для бакалаврів комп'ютерних наук спеціалізації системного проектування. Дане поле введено для зручності, з метою відображення у графічному інтерфейсі.

Типи полів для таблиць БД обираються з міркувань досягнення балансу між оптимізацією швидкодії, зручності користування, та точності інформації.

2.4.2 Таблиця «practice»

Дана таблиця відповідає за збереження у БД частини робочого плану, що містить інформацію про проходження практики (рис. 2.5).

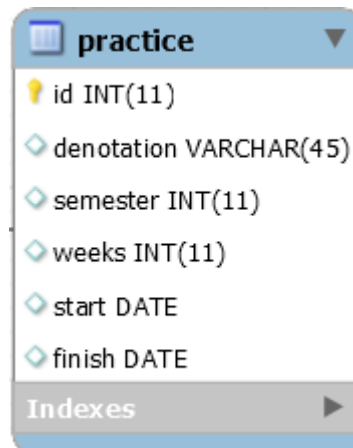


Рис. 2.5 – Таблиця БД «practice»

Таблиця містить наступні поля:

- **id** – INT(11) – первинний ключ запису.
- **denotation** – VARCHAR(45) – строкове значення, назва практики, наприклад «Виробнича», або «Переддипломна».
- **semester** – INT(11) – числове значення, що показує, в якому семестрі проходить практика, яку описує даний запис.
- **weeks** – INT(11) – числове значення, що описує кількість тижнів, відведених на проходження даної практики.
- **start** – DATE – значення у форматі дати СУБД MySQL, визначає початок практики.
- **finish** – DATE – значення у форматі дати, визначає кінцевий строк проходження практики.

2.4.3 Таблиця «state_certification»

Дана таблиця зберігає інформацію стосовно державної атестації випускників кафедри. Інформація про неї може бути використана як у навчальному, так і в робочому навчальному планах. Структура таблиці зображена на рисунку 2.6.

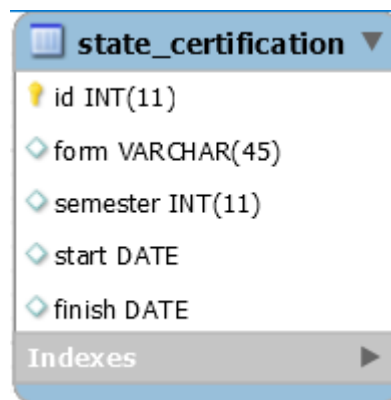


Рис. 2.6 – Таблиця БД «state_certification»

Таблиця містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ, що ідентифікує запис у таблиці.
- **form** – VARCHAR(45) – строкове значення, що містить назву форми, в якій проводиться державна атестація.
- **semester** – INT(11) – числове значення, що описує, в якому саме семестрі проводиться атестація випускників.
- **start** – DATE – дата початку.
- **date** – DATE – дата завершення.

2.4.4 Таблиця «diploma_preparation»

Дана таблиця описує частину РНП, в якій описується розподіл годин по підготовці та захисту дипломних проектів (робіт). Дана таблиця містить у собі перелік заходів (видів робіт), що проводяться для підготовки. Розроблена

таблиця показана на рис. 2.7.

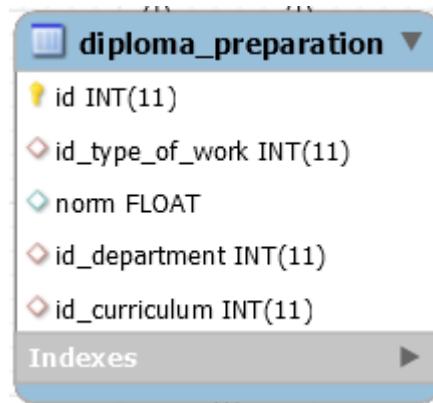


Рис. 2.7 – Таблиця БД «diploma_preparation»

Таблиця містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ, однозначно ідентифікує запис у таблиці.
- **id_type_of_work** – INT(11) – числове значення, вторинний ключ до таблиці-словника «**type_of_work**», що містить в собі записи наявних видів робіт по підготовці диплому, та буде описана нижче.
- **norm** – FLOAT – числове десяткове значення, що описує норму в годинах на одного студента по виконанню виду роботи, який описується даним записом таблиці.
- **id_department** – INT(11) – числове значення, вторинний ключ до таблиці-словника «**department**», в якій наведено перелік кафедр; описана нижче.
- **id_curriculum** – INT(11) – числове значення, вторинний ключ (посилання) до робочого плану, в якому описується розподіл годин з підготовки та захисту дипломного проекту.

Варто зазначити, що поле даної таблиці «**id_curriculum**» могло би бути відсутнім у даній таблиці. Натомість, посилання на відповідні види роботи з «Підготовки та захисту...» можна було б додати до таблиці «**curriculum**», але

цього не було зроблено з наступних міркувань. По-перше, відносно невелика кількість робочих планів містить записи про ці заходи. По-друге, архітектура була би менш гнучкою. Зараз записи зберігаються в окремій таблиці, де кожен запис має посилання на потрібний РНП, тобто при появі нового типу роботи можна легко замінити існуючий, чи додати новий. При зміні цього підходу на інший, що використовує посилання в таблиці «**curriculum**», кількість робіт була б фіксованою, тому при появі певних змін довелося би правити структуру таблиці, а тоді і серверну частину. Тому обраний підхід з посиланнями на РНП в записах видів робіт є обґрунтованим з точки зору гнучкості архітектури при необхідності внесення змін.

2.4.5 Таблиці «**type_of_work**» та «**department**»

Дані таблиці є типовим прикладом таблиць-словників, що містять пари значень «ключ – розшифрування». У випадку таблиці «**type_of_work**» у відповідність до первинного ключа ставиться назва типу роботи, а у випадку таблиці «**department**» – назва кафедри. Розроблені таблиці зображено на рисунку 2.8.

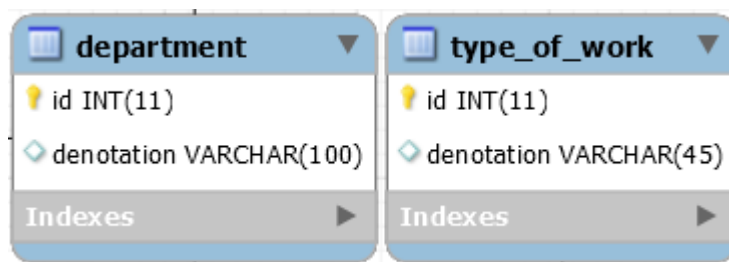


Рис. 2.8 – Таблиці БД «**type_of_work**» та «**department**»

Обидві таблиці містять поля:

- **id** – INT(11) – первинний ключ запису щодо виду роботи/кафедри.
- **denotation** – VARCHAR(100) – назва виду роботи/кафедри.

2.4.6 Таблиця «academic_group»

Дана таблиця описує ще одну принципову для даної системи сутність – академічну групу. Вона поєднує у собі всі необхідні дані для генерації робочого навчального плану, а також може бути використана для обчислення нормативних параметрів при генерації форми К-3 з метою визначення навчального навантаження. Розроблену таблицю зображено на рис. 2.9.

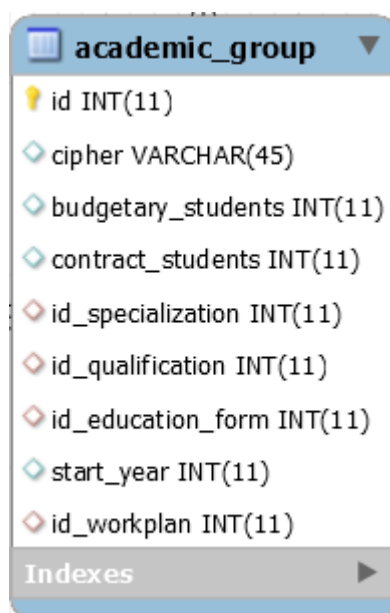


Рис. 2.9 – Таблиця БД «academic_group»

Таблиця містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **cipher** – VARCHAR(45) – строкове значення, містить шифр групи, наприклад, «ДА-32».
- **budgetary_students** – INT(11) – числове значення, показує, скільки в групі навчається студентів на бюджетному фінансуванні.
- **contract_students** – INT(11) – числове значення, показує, скільки в групі навчається студентів за контрактом.
- **id_specialization** – INT(11) – числове значення, вторинний ключ, посилається на таблицю «**specialization**», в якій зазначено перелік

спеціалізацій, що викладаються різноманітними кафедрами. Опис таблиці приводиться нижче.

- **id_qualification** – INT(11) – числове значення, вторинний ключ, посилається на таблицю «**qualification**», в якій наводиться перелік освітньо-кваліфікаційних рівнів, наприклад, «бакалавр», «магістр», «спеціаліст». Опис таблиці див. нижче.
- **id_education_form** – INT(11) – числове значення, вторинний ключ, який посилається на таблицю «**form_of_education**», що містить перелік форм навчання, що доступні на кафедрі, наприклад, «денна», «заочна».
- **start_year** – INT(11) – числове поле, позначає рік набору студентів, коли була сформована ця група.
- **id_workplan** – INT(11) – числове поле, вторинний ключ – посилення на таблицю «**curriculum**», тобто на робочий план, за яким група навчається у поточному навчальному році.

Таблиця має велике значення, оскільки використовується для визначення спеціалізації та форми навчання при генерації робочого плану, а також дані про склад та чисельність групи використовуються для обчислення навчального навантаження. Загалом, при веденні документообігу критично важливо мати основну інформацію про групи, які навчаються на даній кафедрі.

2.4.7 Таблиці «**form_of_education**» та «**qualification**»

Дані таблиці-словники використовуються в першу чергу для відокремлення в БД, конкретно в таблиці «**academic_group**», інформації про ОКР та форму навчання відповідної групи. Таким чином, інформація, що часто повторювалася б у таблиці «**academic_group**», замінюється вторинним ключем на запис в окремій таблиці.

Загальна структура таблиць наводиться на рисунку 2.10.

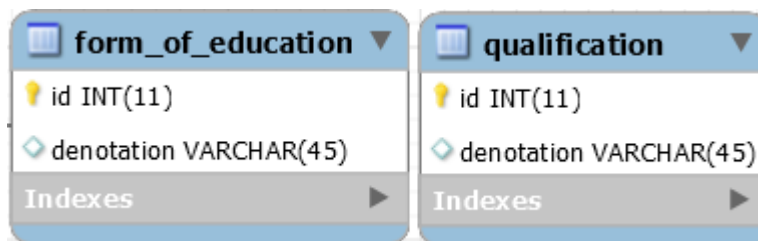


Рис. 2.10 Таблиці БД «form_of_education» та «qualification»

Таблиці містять однакові поля, зміст яких наступний:

- **id** – INT(11) – числове значення, первинний ключ.
- **denotation** – строкове значення, для таблиці «form_of_education» містить назву форми навчання («денна», «заочна»), а для таблиці «qualification» – назву ОКР («бакалавр», «магістр» і т.і.).

2.4.8 Таблиця «specialization»

Таблиця «**specialization**» містить перелік спеціалізацій, за якими відбувається підготовка студентів на кафедрі. Дана таблиця за своєю структурою дуже схожа на попередньо описані «form_of_education» та «qualification», але має одну суттєву відмінність, що полягає у наявності вторинного ключа з посиланням на таблицю «**department**», тобто запису відповідає кафедра, на якій ведеться підготовка за даною спеціалізацією. Розроблена таблиця показана на рис. 2.11.

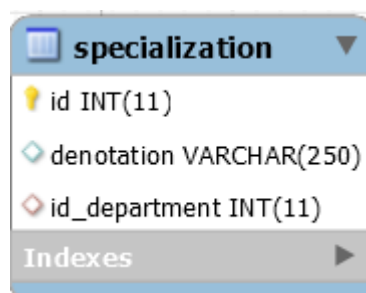


Рис. 2.11 – Таблиця БД «specialization»

Таблиця містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ запису.
- **denotation** – VARCHAR(250) – строкове значення, повна назва спеціалізації.
- **id_department** – INT(11) – числове значення, вторинний ключ на таблицю «**department**». Означає, що дана спеціалізація викладається відповідною кафедрою.

2.4.9 Таблиця «academic_subject»

Дана таблиця відповідає за збереження ще одної принципової для системи сутності – предмету. Таблиця містить в собі динамічну інформацію стосовно дисципліни, що викладається, а саме: розподіл аудиторних години, що назначені цьому предмету, семестр, в якому викладається предмет, а також посилання на відповідний запис у таблиці «**dict_subjects**», що буде описана згодом. Тут варто зазначити, що в таблиці «**dict_subjects**» зберігається статична інформація про предмети (див. наступний пункт). Структуру таблиці зображено на рис. 2.12.

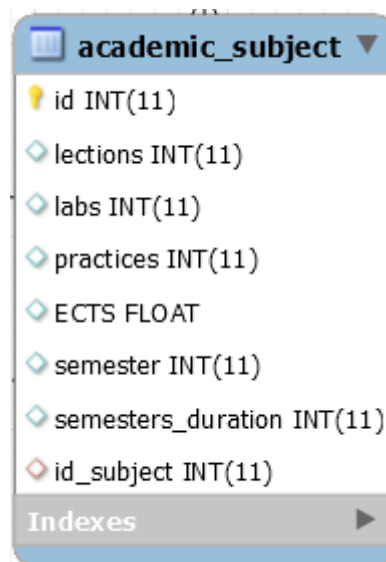


Рис. 2.12 – Таблиця БД «academic_subject»

Таблиця має наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.

- **lections** – INT(11) – числове значення, позначає кількість годин зазначених у навчальному плані чи РНП на лекції.
- **labs** – INT(11) – числове значення, що позначає кількість відведених на лабораторні роботи годин.
- **practices** – INT(11) – числове значення, що показує кількість годин, відведених на практики.
- **ECTS** – INT(11) – числове значення, позначає кількість кредитів за системою.
- **semester** – INT(11) – числове значення, що вказує, в якому семестрі викладається даний предмет.
- **semesters_duration** – INT(11) – числове значення, вказує тривалість предмета у семестрах.
- **id_subject** – INT(11) – вторинний ключ, посилається на таблицю «dict_subject», тобто на запис із відповідною статичною інформацією по предмету.

2.4.10 Таблиця «curriculum_subject»

Ця таблиця призначена для організації зв'язку «Many-to-many» між таблицями «academic_subject» та «curriculum». Суть полягає у тому, що навчальний план/робочий навчальний план містить у собі багато предметів, в той самий час один і той самий предмет може міститись більш ніж в одному навчальному плані. Таким чином, для того, щоб зв'язати відповідні записи в таблицях «academic_subject» та «curriculum» вводиться допоміжна таблиця, що встановлює цей зв'язок. Також в даному випадку у таблицю зв'язку додано поле, що вказує який шифр (порядковий номер) має предмет у відповідному навчальному плані. Справа в тому, що НП та РНП мають різні порядкові позначення для списку предметів. В НП застосовуються шифри, а в РНП – порядковий номер. Отже, просто зазначаючи відповідний номер для предмету в НП/РНП вказується і його розміщення серед інших дисциплін.

Структуру розробленої таблиці зображено на рисунку 2.13.

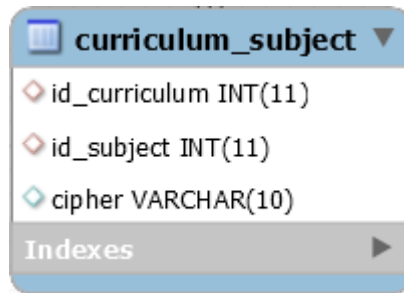


Рис. 2.13 – Таблиця БД «curriculum_subject»

Вона містить наступні поля:

- **id_curriculum** – INT(11) – числове поле, вторинний ключ до таблиці з записами щодо НП/РНП.
- **id_subject** – INT(11) – числове поле, вторинний ключ до таблиці з записами стосовно предметів.
- **cipher** – VARCHAR(10) – строкове поле, що містить порядковий номер чи шифр предмету у РНП чи НП відповідно.

2.4.11 Таблиця «dict_subjects»

Таблиця «dict_subjects» відповідає за збереження статичної частини інформації про предмети, таку як назва, положення у секціях документа (РНП чи НП) тип цього предмету а також кафедру, що викладає даний предмет. Таким чином, на один запис «dict_subjects» може посилатись декілька записів з таблиці «academic_subject», таким чином не треба повторювати одну і ту саму інформацію, якщо, наприклад, на різних спеціалізаціях один і той самий предмет викладається з різним розподілом аудиторних годин, проте вся інша інформація (назва, кафедра, що викладає) зберігається. Таблиця «academic_subject» просто матиме два записи для відповідних розподілів аудиторних годин, а в таблиці «dict_subjects» зберігатиметься спільна інформація.

Ще однією особливістю таблиці є наявність запису, в якому вказується

інший запис з цієї самої таблиці. Це зроблено з метою організації можливості створення, так званого «батьківського», предмета. Ця можливість може стати в нагоді у двох випадках.

По-перше, якщо предмет викладається протягом декількох семестрів, як, наприклад, деякі математичні дисципліни, то у відповідних робочих планах він буде розбитий на відповідну кількість семестрів, в яких буде викладатися. В той самий час, у навчальному плані робиться лише один запис про таку дисципліну з зазначенням того, що вона викладається у декількох семестрах. Отже, маючи можливість створити батьківський предмет, немає потреби у створенні окремого запису, який би був «сумою» предметів. Натомість, розглядаючи «батьківський» предмет, можна легко знайти його нащадків та порахувати відповідні сумарні показники при генерації навчального плану.

По-друге, деякі предмети мають курсові роботи (проекти), що зазначаються в робочих планах в якості окремої дисципліни, а в навчальному, навпаки, міститься лише запис про сам предмет, з тією відмінністю, що він має більшу кількість годин, тобто години за виконання курсової

Розроблена структура таблиці представлена на рис. 2.14.

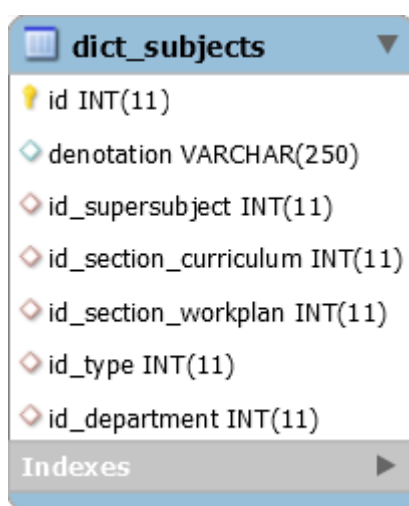


Рис. 2.14 – Таблиця БД «dict_subject»

Таблиця містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **denotation** – VARCHAR(250) – строкове поле, що містить повну назву дисципліни.
- **id_supersubject** – INT(11) – числове значення, вторинний ключ до запису в цій самій таблиці, що позначає «батьківський» предмет.
- **id_section_curriculum** – INT(11) – числове значення, вторинний ключ до таблиці «**section**», що містить перелік секцій (категорій) предметів, що зустрічаються в НП чи РНП. Дане поле заповнюється, якщо відповідний предмет міститься у навчальному плані.
- **id_section_workplan** – INT(11) – так само, як і попереднє поле, але заповнюється у випадку, коли предмет міститься у РНП.
- **id_type** – INT(11) – числове значення, вторинний ключ до таблиці-словника, що містить перелік типів дисциплін.
- **id_department** – INT(11) – числове значення, вторинний ключ до таблиці «**department**», вказує на кафедру, яка викладає даний предмет.

2.4.12 Таблиця «**subject_type**»

Дана таблиця, на яку посилається «**dict_subject**», містить перелік типів предметів. Необхідність введення такої таблиці в тому, що в робочих навчальних планах окрім звичайних дисциплін, які вивчаються студентами, записуються на відповідних роках навчання практики та дипломне проектування. Тому потрібно відрізнити звичайний предмет, який буде враховуватись при розрахунку навчального навантаження від подібних заходів. Структура таблиці показана на рисунку 2.15.

Як і будь-яка інша таблиця словник, містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **denotation** – VARCHAR(45) – строкове значення, містить назву типу, наприклад, «Предмет», «Кафедра».

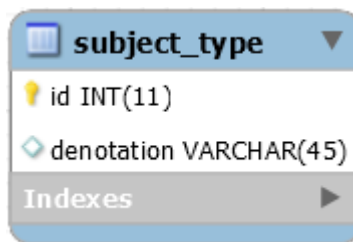


Рис. 2.15 – Таблиця БД «subject_type»

2.4.13 Таблиці «section» та «cycle»

Список предметів у РНП чи НП поділяється за циклами підготовки, кожен з яких додатково ділиться на підрозділи. В даній роботі вони позначаються як секції.

Таблиця «**section**» зберігає список цих підрозділів з посиланнями на відповідні записи в таблиці «**cycle**», що позначають відповідні цикли підготовки, або інші види розділів (для сумісності зі стрими шаблонами) документа.

Таблиця «**cycle**» являє собою ще одну таблицю-словник.

Розроблені таблиці зображені на рисунку 2.16.

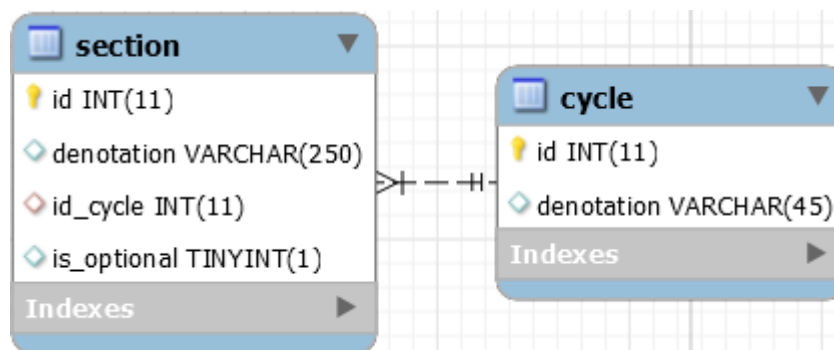


Рис. 2.16 – Таблиці БД «section» та «cycle»

Таблиця «**section**» містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **denotation** – VARCHAR(250) – строкове значення, назва секції.
- **id_cycle** – INT(11) – числове значення, вторинний ключ до запису у таблиці «**cycle**», на відповідний розділ (цикл) списку дисциплін.

- **is_optional** – TINYINT(1) – числове значення, використовується в якості логічного поля, вказує, чи є дана категорія предметів виборною студентами.

Таблиця «**cycle**» має наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **denotation** – VARCHAR(45) – строкове значення, вказує назву розділу (циклу) списку дисциплін.

2.4.14 Таблиці «**control**» та «**dict_control**»

Таблиці «**control**» та «**dict_control**» відображають важливу сутність, що стосується предмета, але не була включена до абстрактної моделі даних (див. рис. 2.3). Ця сутність напряду витікає з сутності предмета, оскільки являє собою його важливу складову – контрольний захід. Записи з переліком контрольних заходів використовуються у робочому плані, а також є обов’язковими при розрахунку навчального навантаження. Таблиця «**control**» напряду пов’язана з таблицею «**academic_subject**» посиланням на запис у цій таблиці, що відповідає предмету, за яким закріплено даний вид контролю. Окрім цього, контроль містить інформацію про семестр, в якому застосовується та посилання на словник типів контролю (екзамен, залік, диференційний залік, РГР, ДКР, контрольна робота і т.і.).

Структура розроблених таблиць зображена на рисунку 2.17.

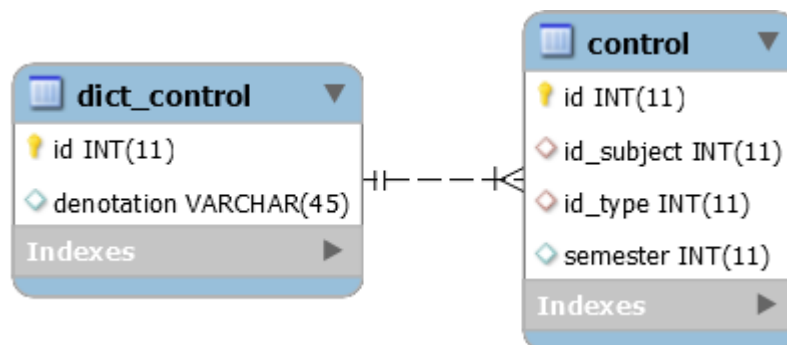


Рис. 2.17 – Таблиці БД «**dict_control**» та «**control**»

Таблиця «control» містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **id_subject** – INT(11) – числове значення, вторинний ключ до запису у таблиці БД щодо предмета, за яким закріплено даний контрольний захід.
- **id_type** – INT(11) – числове значення, вторинний ключ до запису у таблиці, що вказує на відповідний тип контрольного заходу.
- **semester** – INT(11) – числове значення, вказує, у якому семестрі проводиться відповідний контрольний захід.

Таблиця «dict_control» має наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **denotation** – VARCHAR(45) – строкове значення, вказує назву відповідного типу контрольного заходу.

2.4.15 Таблиця «study_load_results»

Таблиця «study_load_results» призначена для збереження в БД результатів розрахунку навчального навантаження на викладачів кафедри. Таблиця містить відповідні дані для поточного навчального року, а також у ній зазначається, для якої форми навчання та за яким фінансуванням розраховане дане навантаження.

Таблиця містить наступні поля:

- **id** – INT(11) – числове значення, первинний ключ.
- **id_education_form** – INT(11) – числове значення, вторинний ключ до запису у таблиці «», що відповідає формі навчання, для якої визначається дане навантаження.
- **financing** – INT(11) – числове значення, позначає тип фінансування (0 – бюджетне, 1 – контрактне)
- **id_academic_subject** – INT(11) – числове значення, вторинний ключ до відповідного запису предмету, по якому обраховані параметри,

що відображаються наступними полями.

- **lections** – FLOAT – десяткове значення, лекції у годинах.
- **practices** – FLOAT – десяткове значення, практичні заняття у годинах.
- **labs** – FLOAT – десяткове значення, лабораторні заняття у годинах.
- **individuals** – FLOAT – десяткове значення, індивідуальні заняття у годинах.
- **exams** – FLOAT – десяткове значення, час на проведення екзамену у годинах.
- **credits** – FLOAT – десяткове значення, час на проведення заліків у годинах.
- **contr_works** – FLOAT – десяткове значення, час на проведення контрольних, модульних робіт у годинах.
- **course_projs** – FLOAT – десяткове значення, час, відведений на прийом курсових проектів, у годинах.
- **course_works** – FLOAT – десяткове значення, час, відведений на прийом курсових робіт, у годинах.
- **rgr** – FLOAT – десяткове значення, час на прийом розрахунково-графічних, розрахункових чи графічних робіт у годинах.
- **dkr** – FLOAT – десяткове значення, час на перевірку домашніх контрольних робіт у годинах.
- **referats** – FLOAT – десяткове значення, час на перевірку рефератів у годинах.
- **consult** – FLOAT – десяткове значення, час, відведений на проведення консультацій у годинах.

Структура розробленої таблиці зображена на рис. 2.18.

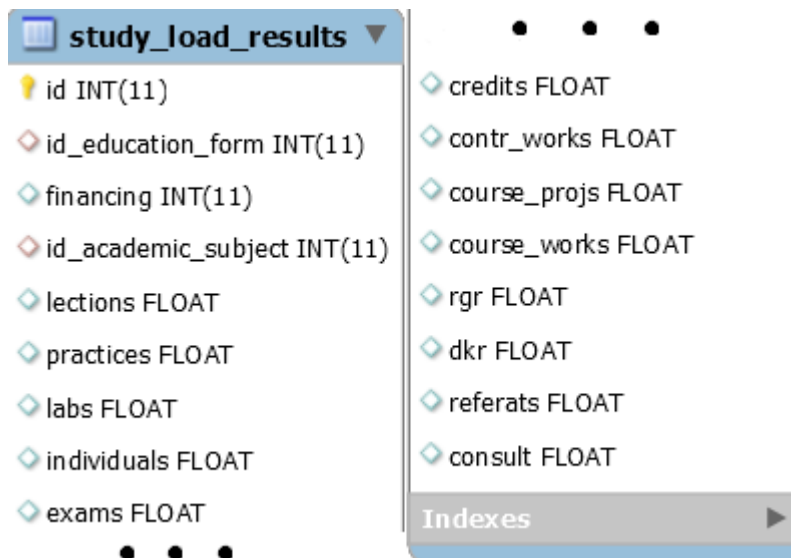


Рис. 2.18 – Таблица БД «study_load_results»

2.5 Висновки щодо розробки БД системи

У даному розділі було проведено теоретичний огляд методів та засобів побудови БД, а також проведено аналіз необхідного для збереження обсягу даних, з метою розробки оптимальної структури БД для системи, що розробляється. Було виділено основні інформаційні сутності, що сформували основу для абстрактної моделі даних, яка допомогла у визначенні подальших дій з розробки безпосередньо таблиць БД.

При розробці таблиць головною метою було охопити увесь обсяг потрібних для збереження даних та запобігти логічним суперечкам, чи надлишковому повторенню інформації.

Розроблена БД спроектована таким чином, щоб містити в собі інформацію тільки на **поточний навчальний рік**. Отже, при розробці серверної частини однією з задач буде передбачення алгоритму створення БД для нового навчального року а також перенесення даних з минулого.

Такий підхід використано з метою забезпечення цілісності даних, що стосуються організації навчального процесу за минулі роки при переході на новий та редагуванні даних в поточному році.

3 РОЗРОБКА ІНТЕРФЕЙСУ КОРИСТУВАЧА

3.1 Вступ

Метою розробки графічного інтерфейсу користувача є створення ефективного засобу взаємодії між людиною та системою.

Інтерфейс повинен підтримувати увесь наявний функціонал серверної частини, надавати доступ до перегляду усієї інформації, що зберігається в БД, а також можливість редагування та доповнення даної інформації. Окрім цього, зазначені властивості повинні бути реалізовані таким чином, щоб користування системою було зручним та зрозумілим.

3.2 Аналіз можливих реалізацій

При розробці графічного інтерфейсу можливі два варіанти реалізації. Оскільки система, що розробляється, буде побудована за принципами клієнт-серверної архітектури, клієнт представлятиме окрему підсистему, яка буде виконувати свої функції за рахунок взаємодії з прикладним програмним інтерфейсом (API), який надаватиме серверна частина.

В рамках даної роботи клієнтська частина буде взаємодіяти з серверною через веб-застосунок, реалізований на мові програмування Java. Клієнт взаємодіє з сервісом за допомогою HTTP(S) протоколу, використовуючи відповідні протокольні запити. Загальна схема представлена на рисунку 3.1.

Відповідно до цього можливі два варіанти побудови інтерфейсу: крос-платформна реалізація настільного застосунку, або веб-інтерфейс.

Перевагою повноцінної програми є більша гнучкість, яка доступна при написанні коду. Однак такий варіант буде потребувати набагато більше часу, а також висуватиме вимоги до апаратної платформи. Можна створити крос-платформну реалізацію, наприклад, за допомогою мови програмування Java, але

в такому разі для використання системи клієнту необхідно мати встановлене середовище виконання Java (JRE).

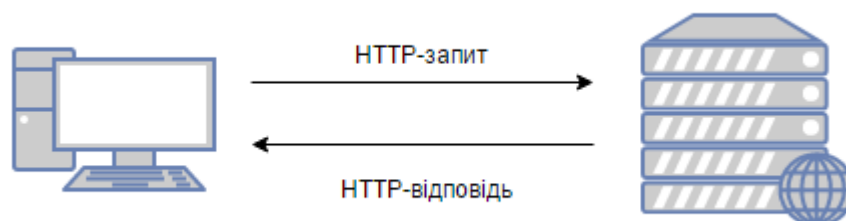


Рис. 3.1 – Загальний принцип мережевої клієнт-серверної взаємодії

На противагу, веб-інтерфейс дозволить користуватися системою без встановлення будь-якого додаткового ПЗ на свою машину. Тобто доступ до системи може здійснюватися за допомогою взаємодії між сервером кафедри та користувацькою машиною. Це, по-перше, зменшує вимоги до апаратних можливостей робочої станції, на якій буде здійснюватися доступ, по-друге в перспективі програму можна буде доробити для забезпечення повноцінного багато-користувацького режиму. Втім, слід зазначити, що на етапі налагодження програми на кафедрі уся система буде встановлена на одній ЕОМ.

Враховуючи переваги реалізації веб-інтерфейсу, цей варіант обрано в якості клієнтської частини.

3.3 Реалізація інтерфейсу користувача

Як вже було зазначено, для реалізації клієнтської частини було обрано веб-інтерфейс. Це дозволить пришвидшити та спростити процес розробки без втрати візуальної якості та зручності у використанні.

Для реалізації було використано найбільш розповсюджені технології веб-розробки: HTML, CSS, JavaScript. Також для спрощення роботи з табличною інформацією використано плагін jQuery – jTable. Даний плагін є проектом з

відкритим програмним кодом, отже його використання є повністю безкоштовним. На рисунку 3.2 зображено приклад інтерфейсу при редагуванні записів з БД.

Предмети							+ Add new record
Предмети	Назва	Кафедра	Загальний предмет	Розділ у навчальному плані	Розділ у робочому навчальному плані	Тип	
☰	Інтелектуальний аналіз даних	Системного проектування	-	Навчальні дисципліни базової підготовки	Цикл професійної та практичної підготовки	Предмет	✎ 🗑
Інтелектуальний аналіз даних - предмети							+ Add new record ✕
☰	Контрольні заходи	Семестр	Тривалість у семестрах	ECTS	Лекції	Лабораторні	Практичні
☰	6	1	5	36	36	0	✎ 🗑
Інтелектуальний аналіз даних - контрольні заходи							+ Add new record ✕
Тип контрольного заходу						Семестр	
Екзамен			6				✎ 🗑
☰	6	1	3.5	36	27	0	✎ 🗑
☰	Інформаційно-аналітичні системи прийняття рішень	Системного проектування	-	Навчальні дисципліни професійної та практичної підготовки	Навчальні дисципліни професійної та практичної підготовки	Предмет	✎ 🗑
☰	Автоматизоване проектування систем	Системного	-	Навчальні дисципліни професійної та	Навчальні дисципліни професійної та	Предмет	✎ 🗑

Рис. 3.2 – Приклад користувацького інтерфейсу.

Як видно з рисунку 3.2 користувачу надається можливість переглядати усі наявні в БД дані та редагувати їх. Перелік даних зосереджений в пункті меню «Дані», структура якого зображена на рис. 3.3.

Для більшої зручності користування дані було розбито за змістом на 3 категорії: «Предмети», «Факультет», «Документи».

Підпункт меню «Предмет» містить доступ до тих даних в БД, що безпосередньо стосуються навчальних дисциплін.

Підпункт меню «Факультет» надає доступ до перегляду та редагування усієї інформації, що стосується організації навчального процесу на факультеті (кафедрі).

Підпункт «Документи» відображає наявні в БД дані про організацію нормативної документації та інші аспекти.



Рис. 3.3 – Структура меню «Дані»

У пункті меню «Генерація» надається можливість згенерувати відповідні документи

Пункт меню «Робота з шаблонами» надає доступ до завантаження шаблонів документів, що відповідають НП, РНП та формі К-3 (рис. 3.4)

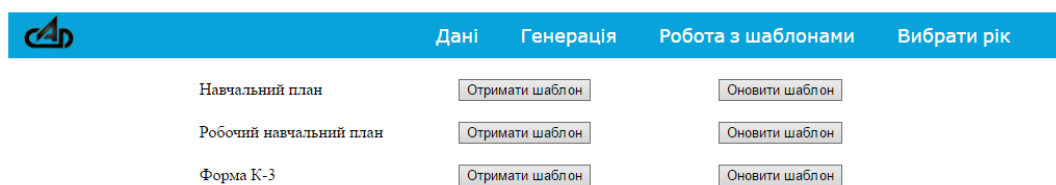


Рис. 3.4 – Робота з шаблонами через графічний інтерфейс

Пункт меню «Вибрати рік» дозволяє перемкнути поточний рік БД, та отримати доступ до відповідної інформації. Система пропонує відкрити існуючі роки, або створити базу даних на новий рік.

3.4 Висновки щодо реалізації інтерфейсу

В даному розділі було описано основні моменти реалізації інтерфейсу користувача, який дозволяє взаємодіяти з системою. Надано можливість переглядати та редагувати дані, що знаходяться в БД за поточний рік. При необхідності, можна переключитися на минулий чи новий рік для продовження роботи.

Інтерфейс реалізовано у вигляді веб-сторінок з використанням HTML/CSS, а також безкоштовного плагіну для роботи з AJAX-запитами та формуванням табличного представлення інформації – jTable.

Взаємодія клієнтської частини з серверною реалізується з використанням HTTP-запитів, в середину яких упаковується необхідна інформація у форматі JSON.

Інтерфейс надає можливість оновлювати шаблони документів, що знаходяться на сервері.

Отже, розроблений інтерфейс дозволяє використовувати можливості серверної частини по генерації документів, перегляду/зміні змісту БД, та виконаний у приємній кольоровій гамі та зі зручною, інтуїтивно зрозумілою структурою.

4 ЕКОНОМІКО-ОРГАНІЗАЦІЙНИЙ РОЗДІЛ

4.1 Вступ

Даний розділ призначений для проведення оцінки основних характеристик компонентів автоматизованої системи «Документообіг кафедри», які розробляються в рамках даної дипломної роботи. В цілому АС призначена для генерації документів, що використовуються на кафедрі для організації та аналізу навчального процесу. В даній роботі розглядається створення таких компонентів цієї системи, як база даних, та інтерфейс користувача. Системою управління базою даних було обрано MySQL, робота з якою здійснювалася за допомогою супутнього ПЗ MySQL Workbench. Розробка інтерфейсу проводилася у середовищі розробки NetBeans 8.1 за допомогою мови програмування JavaScript та мов розмітки HTML та CSS.

Компоненти системи, що розробляються повинні мати можливість використовуватися на персональних комп'ютерах під керуванням ОС Windows або найбільш розповсюджених дистрибутивів Linux, таких як Ubuntu та інші.

Нижче наведено аналіз різних варіантів реалізації компонентів системи з метою вибору оптимальної, з огляду, при цьому, як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прями, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів,

виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Алгоритм цього методу виглядає наступним чином:

- визначається послідовність функцій, необхідних для виробництва продукту (спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На цьому ж етапі оптимізується сама послідовність скороченням кроків, що не впливають на цінність і відповідно витрат);
- для кожної функції визначаються повні річні витрати й кількість робочих часів;
- для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат;
- після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

4.2 Постановка задачі техніко-економічного аналізу

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки компонентів автоматизованої системи для оптимізації організаційних питань навчального процесу. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляються для використання у системі впливають на загальні характеристики усієї системи, фактичний аналіз представлятиме собою аналіз функцій програмного продукту в цілому.

Отже для проведення цього аналізу необхідно обрати систему показників якості програмного продукту.

Можна виділити наступні показники якості, що можуть бути застосовані для автоматизованої системи документообігу кафедри:

- система повинна функціонувати на ЕОМ з встановленим сервером бази даних та програмним забезпеченням для виконання веб-застосунків, або користувачка ЕОМ з інтерфейсом користувача системи повинна мати доступ через мережеве з'єднання до серверу, на якому встановлені вищезазначені компоненти;
- система повинна генерувати документи наповнюючи надані шаблони коректним змістом, при умові введення у систему актуальних та правильних даних, а також вірно оформлених шаблонів документів;
- система повинна забезпечувати задовільну швидкодію;
- вартість компонентів, що використовуються при реалізації системи повинна задовольняти обмежені фінансові можливості кафедри;
- інтерфейс користувача повинен бути простим та зрозумілим у використанні.

4.2.1 Обґрунтування функцій програмного продукту

Головною функцією є F_0 – процес розробки програмного продукту загалом, який складається з окремої розробки кожного компоненту та налагодження їхньої спільної взаємодії. Виходячи з мети розробки конкретних компонентів даної системи в рамках цієї роботи, зазначених вище, можна виділити наступні основні функції ПП:

F_1 – вибір БД для системи;

F_2 – вибір засобів для реалізації інтерфейсу користувача;

F_3 – вибір засобів для взаємодії інтерфейсу та серверної частини.

Кожна з цих функцій має різні варіанти реалізації. Для F_1 можна вибрати один з двох варіантів програмного забезпечення системи керування БД: Oracle чи MySQL, кожен з яких має свої недоліки та переваги. Для реалізації графічного інтерфейсу користувача також можливий вибір технологій розробки. Те саме стосується і технологій для комунікації між клієнтом та сервером (табл. 4.1).

Таблиця 4.1 – Варіанти реалізації основних функцій

Функція	Варіант реалізації
F_1	MySQL
	Oracle
F_2	JavaScript, HTML/CSS
	JSP, Java
F_3	JSON
	XML

4.2.2 Варіанти реалізації основних функцій

Нижче наведено варіанти реалізацій основних функцій у вигляді морфологічної карти системи (рис. 4.1). Спираючись на цю карту побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 4.2).

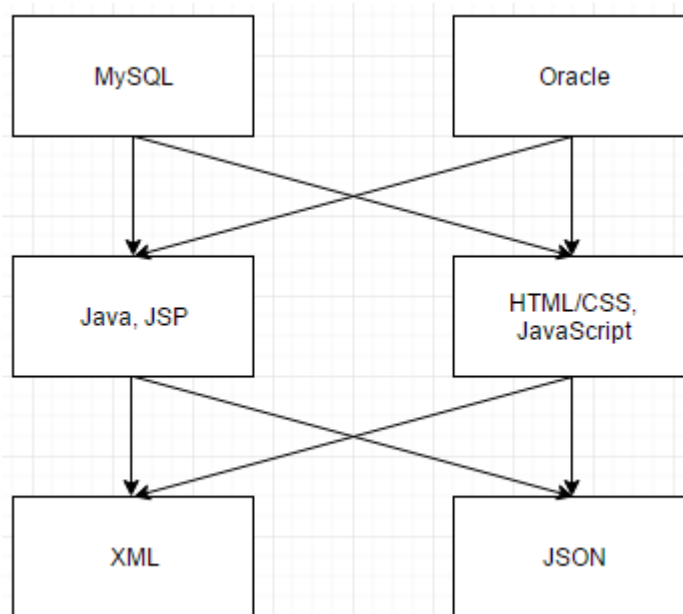


Рис. 4.1 – Морфологічна карта

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів ПП.

Таблиця 4.2 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
<i>F1</i>	MySQL (a)	Розповсюджена, безкоштовна	Менший обсяг можливостей
	Oracle (б)	Професійна, ширший функціонал	Платна
<i>F2</i>	JS, HTML/CSS (a)	Швидкодія, краще візуальне сприйняття	Більша кількість файлів із кодом
	Java, JSP (б)	Простіша реалізація взаємодії з серверною частиною	Більші об'єми трафіку передаються на клієнтську частину
<i>F3</i>	XML (a)	Краще на стороні сервера	Важче
	JSON (б)	Простіше, швидше	Більш нова технологія.

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція *F1*:

Зважаючи на фінансові обмеження, що накладаються на реалізацію продукту для кафедри, важливо обрати варіант, що буде більш економічно вигідний. Оскільки варіант (a) – open-source продукт, тобто є безкоштовним, а варіант (б) – комерційний – він має бути відкинтий.

Функція *F2*:

Оскільки швидкодія є важливою властивістю для програмних продуктів такого типу, а також навантаження на трафік краще зменшити – слід відкинути варіант (б).

Функція *F3*:

Оскільки обидві технології є широко вживаними та мають велику підтримку різноманітними мовами програмування, тож можна вважати обидва варіанти гідними розгляду.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

1. F1a – F2a – F3a

2. F1a – F2a – F3б

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів програмного продукту

4.3.1 Опис параметрів

Спираючись на дані про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

З метою охарактеризувати даний програмний продукт, використовується наступна система параметрів, яка враховує основні аспекти роботи ПП, з огляду на перелік компонентів, що розробляються в рамках цієї роботи:

- X1 – швидкодія виконання запитів у БД;
- X2 – середній об'єм даних, що передаються від клієнтської частини до серверної, в одному запиті;
- X3 – середній об'єм даних, що передається від серверної частини до клієнтської в одному запиті;
- X4 – потенційний об'єм коду для реалізації серверної частини.

Для кращого розуміння змісту зазначених параметрів наводиться розкриття їх суті.

X1: відображає час, що необхідний для отримання системою бази даних запиту від серверної частини, виконання, зазначених в цьому запиті інструкцій,

формування та повернення відповіді компонентам серверної частини.

X2: Оскільки взаємодія між клієнтською та серверною частиною відбувається за допомогою засобів мережевого спілкування, то дані, що будуть передаватися від клієнта на сервер, спочатку записуватимуться у пакети відповідної мережевої технології. Отже, даний параметр відображає, середній розмір цього пакету, який впливатиме на швидкість взаємодії користувача та системи, тобто на UX.

X3: те саме, що і X2, тільки в зворотному напрямку.

X4: відображає об'єм роботи та зусиль, вкладених у розробку компонентів інтерфейсу.

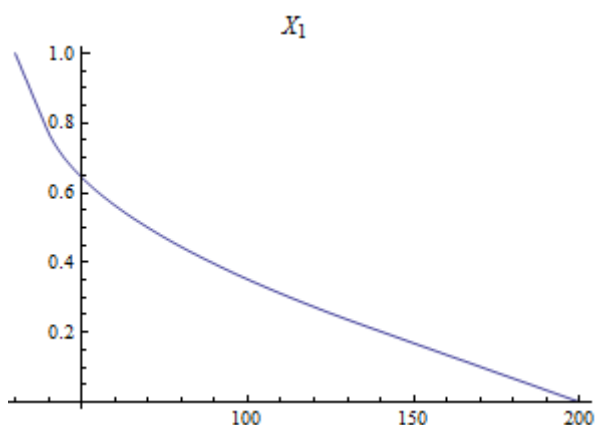
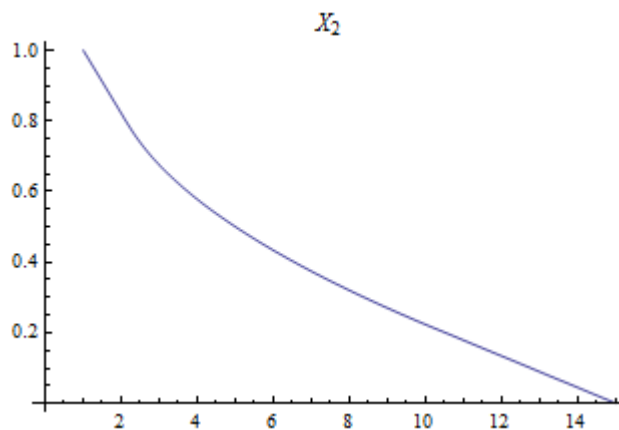
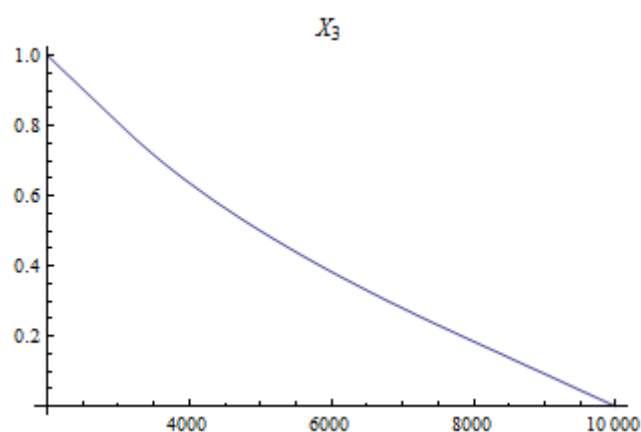
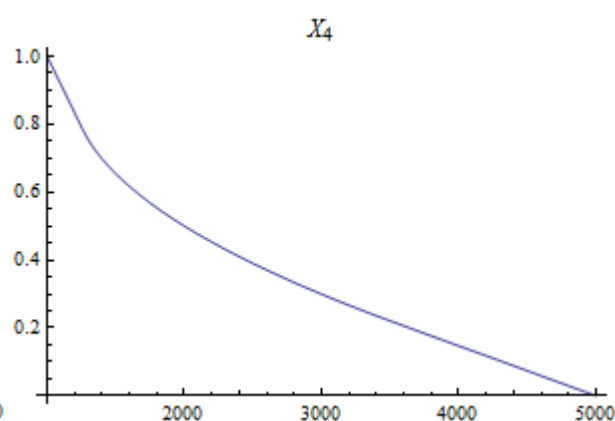
4.3.2 Кількісна оцінка параметрів

Загалом, гірші, середні, і кращі значення параметрів обираються на основі вимог замовника та специфіки процесу розробки та використання системи. В даному випадку, замовником виступає сам виконавець. У таблиці 4.2 показано обрану диференціацію параметрів з огляду на поставлене завдання.

Таблиця 4.3 – Основні параметри ПП

Умовні позначення	Одиниці виміру	Значення параметра		
		гірші	середні	кращі
X1	мс	200	70	30
X2	кБайт	15	5	1
X3	кБайт	10000	5000	2000
X4	строчок коду	5000	2000	1000

За даними з таблиці 4.3 будуються графічні характеристики параметрів, що розглядаються. Вони зображені на рисунках 4.2 – 4.5.

Рис. 4.2 – Параметр X_1 Рис. 4.3 – Параметр X_2 Рис. 4.4 – Параметр X_3 Рис. 4.5 – Параметр X_4

4.3.3 Аналіз експертного оцінювання параметрів

Після визначення параметрів проводиться детальне обговорення й аналіз експертами ступеня важливості кожного параметра для поставленої цілі – в даному випадку, розробки компонентів автоматизованої системи документообігу кафедри, яка повинна правильно генерувати визначені документи та форми, при умові введення коректних даних.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;

- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.4.

Таблиця 4.4 – Результати ранжування параметрів

Позначення параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
		1	2	3	4	5	6	7			
X1	мс	1	2	2	3	1	1	2	12	-5.5	30.25
X2	кБайт	2	2	2	3	3	1	1	14	-3.5	12.25
X3	кБайт	1	2	1	2	2	1	1	10	-7.5	56.25
X4	кількість строк коду	6	4	5	2	4	7	6	34	16.5	272.25
Разом		10	10	10	10	10	10	10	70	0	372

Для перевірки степені достовірності експертних оцінок, визначаються наступні параметри:

- а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \quad (1)$$

де N – число експертів, n – кількість параметрів;

- б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17.5, \quad (2)$$

с) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \quad (3)$$

Сума відхилень по всіх параметрах повинна дорівнювати 0;

д) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 372. \quad (4)$$

Коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 372}{7^2(4^3-4)} = 1,51 > W_k = 0,67 \quad (5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює **0,67**.

Скориставшись результатами ранжирування, проводиться попарне порівняння всіх параметрів, результати якого відображено у таблиці 4.5.

Таблиця 4.5 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	=	=	=	>	=	<	=	1
X1 і X3	=	=	<	<	>	=	<	<	0,5
X1 і X4	>	>	>	<	>	>	>	>	1,5
X2 і X3	<	=	<	<	<	=	=	<	0,5
X2 і X4	>	>	>	=	>	>	>	>	1,5
X3 і X4	>	>	>	>	>	>	>	>	1,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається за формулою:

$$a_{ij} = \begin{cases} 1.5, & \text{при } X_i > X_j \\ 1, & \text{при } X_i = X_j \\ 0.5, & \text{при } X_i < X_j \end{cases} \quad (6)$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра робиться розрахунок вагомості K_{bi} за наступними формулами:

$$K_{bi} = \frac{b_i}{\sum_{i=1}^n b_i}, \text{ де } b_i = \sum_{j=1}^N a_{ij}. \quad (7)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{bi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \text{ де } b'_i = \sum_{j=1}^N a_{ij} b_j. \quad (8)$$

Як видно з таблиці 4.6, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.6 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	b_i	K_{bi}	b_i^1	K_{bi}^1	b_i^2	K_{bi}^2
X1	1,0	1,0	0,5	1,5	4,0	0,25	14,5	0,243	53,63	0,243
X2	1,0	1,0	0,5	1,5	4,0	0,25	14,5	0,243	53,63	0,243
X3	1,5	1,5	1,0	1,5	5,5	0,344	21,5	0,360	78,87	0,358
X4	0,5	0,5	0,5	1,0	2,5	0,156	9,25	0,154	34,5	0,156
Всього:					16	1	59,75	1	220,63	1

4.4 Аналіз рівня якості варіантів реалізації функцій

На даному етапі визначається рівень якості кожного варіанту виконання функцій окремо.

Абсолютні значення параметрів X_1 (швидкодія виконання запитів у базі даних) та X_4 (потенційний об'єм коду для реалізації серверної частини), що відповідають функціям F_1 та F_2 , відповідають технічним вимогам щодо реалізації та функціонування даного програмного продукту.

Абсолютні значення параметра X_2 (середній об'єм даних, що передаються від клієнтської частини до серверної, в одному запиті) та X_3 (середній об'єм даних, що передаються від серверної частини до клієнтської, в одному запиті) у випадку реалізації А – обрання технології XML для формування даних, що будуть передаватися по мережі, виходять не найгіршими, тому варті того, щоб бути розглянутими.

Абсолютні значення параметра X_2 та X_3 у випадку реалізації Б – обрання технології JSON, теж знаходяться в допустимих межах, тому цей варіант реалізації не відкидається при розгляді.

Коефіцієнт технічного рівня для кожного варіанта реалізації програмного продукту розраховується за наступною формулою:

$$K_K(j) = \sum_{i=1}^n K_{ei,j} B_{i,j}, \quad (9)$$

де n – кількість параметрів; K_{ei} – коефіцієнт вагомості i -го параметра; B_i – оцінка i -го параметра в балах. Результати розрахунків наведено нижче у таблиці 4.7. Коефіцієнт вагомості параметра було визначено вище (див. табл. 4.7), а бальна оцінка параметра отримується згідно з наведеними графічними характеристикам параметрів (див. рисунки 4.2 – 4.5). Обчисливши суму рівня якості по кожній функції реалізації програмного продукту, отримується загальний рівень якості виконаної роботи.

Таблиця 4.7 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
$F1(X_1)$	А	50	0,5	0,243	0,1215
$F2(X_4)$	А	2500	0,35	0,156	0,0546
$F3(X_2, X_3)$	А	12 (X_2)	0,2	0,243	0,0486
		7000 (X_3)	0,3	0,358	0,1074
	Б	4 (X_2)	0,58	0,243	0,1409
		4000 (X_3)	0,65	0,358	0,2327

За отриманими даними можна обчислити рівень якості для кожного з варіантів за формулою:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \dots + K_{\text{ТУ}}[F_{zk}]. \quad (10)$$

За формулою (10):

$$K_{K1} = 0,1215 + 0,0546 + 0,0486 + 0,1074 = 0,3321$$

$$K_{K2} = 0,1215 + 0,0546 + 0,1409 + 0,2327 = 0,5497$$

З отриманих результатів легко переконатися, що кращим є другий варіант, для якого коефіцієнт технічного рівня якості має більше значення.

4.5 Економічний аналіз варіантів розробки програмного продукту

З метою визначення вартості розробки ПП спочатку проводиться розрахунок трудомісткості.

Будь-який варіант розробки компонентів, що створюються в рамках даної

роботи включає в себе наступні етапи:

1. Розробка та реалізація БД системи;
2. Розробка та реалізація інтерфейсу користувача.

Розробка бази даних за ступенем новизни відноситься до групи Б – розробка типових проектних рішень, оригінальних задач і систем, які не мають аналогів. До того ж ступеня новизни можна віднести і другу частину – розробку користувацької частини. Завдання 1 використовує алгоритми другої групи складності, а алгоритми, що використовуються при виконанні завдання 2 належать до групи 3.

Для реалізації завдань 1 та 2 використовується довідкова інформація та мови програмування.

Розрахунок норм часу на виконання завдань проектування та реалізації БД та створення графічного інтерфейсу користувача проводиться з використанням формули:

$$T_O = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (11)$$

де T_P – трудомісткість розробки програмного продукту; K_{Π} – поправочний коефіцієнт; $K_{СК}$ – коефіцієнт на складність вхідної інформації; K_M – коефіцієнт рівня мови програмування; $K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм; $K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеню новизни Б та групи складності алгоритму 2, трудомісткість дорівнює: $T_P = 27$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання при його ступені новизни Б: $K_{\Pi} = 1,08$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації $K_{СК} = 1,16$. Оскільки при розробці першого завдання не використовуються стандартні модулі, це відображається за

допомогою коефіцієнта $K_{CT} = 1$.

Отже, за формулою (11) загальна трудомісткість виконання першого завдання становить:

$$T_1 = 27 * 1,08 * 1 * 1,16 = 34 \text{ людино-днів.}$$

Для другого завдання при розробці першого варіанту реалізації (А) при використанні алгоритму третього ступеня складності, та ступеню новизни Б, тобто $T_P = 19$ людино-днів, $K_{II} = 1$, $K_{СК} = 1$, $K_{CT} = 1$:

$$T_{2A} = 19 \text{ людино-днів.}$$

Для другого завдання при розробці другого варіанту реалізації (Б) при використанні алгоритму тієї ж складності та ступеню новизни, змінюється поправочний коефіцієнт $K_{CT} = 0,8$:

$$T_{2B} = 19 * 0,8 = 15,2 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_A = (34 + 19) * 8 = 424 \text{ людино-годин;}$$

$$T_B = (34 + 15,2) * 8 = 393,6 \text{ людино-годин;}$$

Найвищу трудомісткість має варіант А.

В розробці приймає участь один програміст з окладом 14000 грн. Для визначення розміру заробітної плати за годину можна скористатися формулою:

$$C_{ч} = \frac{M}{T_m \cdot t} \text{ грн.,} \quad (12)$$

де M – місячний оклад працівників; T_m – кількість робочих днів; t – кількість робочих годин в день.

$$\text{Отже, } C_{ч} = 14000 / (20 * 8) = 87,6 \text{ грн.}$$

Заробітна плата розраховується за формулою:

$$C_{зп} = C_{ч} \cdot T_i \cdot K_d, \quad (13)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста; T_i – трудомісткість відповідного завдання; $K_{\text{д}}$ – норматив, який враховує додаткову заробітну плату.

Заробітна плата розробника за варіантами становитиме:

$$A: C_{\text{ЗП}} = 87,6 \cdot 424 \cdot 1,2 = 44570,88 \text{ грн.}$$

$$B: C_{\text{ЗП}} = 87,6 \cdot 393,6 \cdot 1,2 = 41375,232 \text{ грн.}$$

Відрахування на єдиний соціальний внесок в залежності від групи професійного ризику (II клас) становить 22%:

$$A: C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0,22 = 44570,88 \cdot 0,22 = 9805,59 \text{ грн.}$$

$$B: C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0,22 = 41375,232 \cdot 0,22 = 9102,55 \text{ грн.}$$

Наступним етапом є визначення витрат на оплату однієї машино-години.

Так як одна ЕОМ обслуговує одного програміста з окладом 14000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримується:

$$C_{\text{Г}} = 12 \cdot M \cdot K_3 = 12 \cdot 14000 \cdot 0,2 = 33600 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{\text{ЗП}} = C_{\text{Г}} \cdot (1 + K_3) = 33600 \cdot (1 + 0,2) = 40320 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0,22 = 40320 \cdot 0,22 = 8870,4 \text{ грн.}$$

Амортизаційні відрахування обчислюються за формулою:

$$C_{\text{А}} = K_{\text{ТМ}} \cdot K_{\text{А}} \cdot C_{\text{ПР}} \quad (14)$$

де $K_{\text{ТМ}}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача; $K_{\text{А}}$ – річна норма амортизації; $C_{\text{ПР}}$ – договірна ціна приладу.

При амортизації 25 % та вартості ЕОМ 20000 грн:

$$C_{\text{А}} = 1,15 \cdot 0,25 \cdot 20000 = 5750 \text{ грн.}$$

Витрати на ремонт та профілактику розраховується як:

$$C_{\text{Р}} = K_{\text{ТМ}} \cdot C_{\text{ПР}} \cdot K_{\text{Р}} = 1,15 \cdot 20000 \cdot 0,05 = 1150 \text{ грн.,}$$

де $K_{\text{Р}}$ – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховується за формулою:

$$T_{\text{ЕФ}} = (D_{\text{К}} - D_{\text{В}} - D_{\text{С}} - D_{\text{Р}}) \cdot t_{\text{З}} \cdot K_{\text{В}}, \quad (15)$$

де $D_{\text{К}}$ – календарна кількість днів у році; $D_{\text{В}}$, $D_{\text{С}}$ – відповідно кількість вихідних та святкових днів; $D_{\text{Р}}$ – кількість днів планових ремонтів устаткування; t – кількість робочих годин в день; $K_{\text{В}}$ – коефіцієнт використання приладу у часі протягом зміни.

Отже, $T_{\text{ЕФ}} = (365 - 116 - 16) \cdot 8 \cdot 0.9 = 1677.6$ годин.

Витрати на сплату електроенергії розраховуються за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_{\text{С}} \cdot C_{\text{ЕН}}, \quad (16)$$

де $N_{\text{С}}$ – середньо-споживча потужність приладу; $C_{\text{ЕН}}$ – тариф за 1 кВт-годин електроенергії.

Отже, $C_{\text{ЕЛ}} = 1677,6 \cdot 0,1 \cdot 1,93819 = 325,150$ грн.

Накладні витрати розраховується за формулою:

$C_{\text{Н}} = C_{\text{ПР}} \cdot 0.67 = 20000 \cdot 0,67 = 13400$ грн.

Тоді, річні експлуатаційні витрати будуть становити:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}}. \quad (17)$$

Отже, $C_{\text{ЕКС}} = 40320 + 8870,4 + 5750 + 1150 + 325,15 + 13400 = 69815.55$ грн.

Тож собівартість однієї ЕОМ на годину буде дорівнювати:

$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 69815.55 / 1677.6 = 41.61$ грн/год.

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T. \quad (18)$$

Отже, для наявних двох можливих вирішень даної задачі:

$$A: C_M = 41,61 * 424 = 17642,64 \text{ грн.}$$

$$B: C_M = 41,61 * 393,6 = 16377,7 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{ЗП} \cdot 0,67. \quad (19)$$

$$A: C_H = 44570,88 * 0,67 = 29862,49 \text{ грн.}$$

$$B: C_H = 41375,232 * 0,67 = 27721,41 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{ЗП} + C_{Вид} + C_M + C_H. \quad (20)$$

$$A: C_{ПП} = 44570,88 + 9805,59 + 17642,64 + 29862,49 = 101881,6 \text{ грн.}$$

$$B: C_{ПП} = 41375,232 + 9102,55 + 16377,7 + 27721,41 = 94576,9 \text{ грн.}$$

4.6 Вибір кращого варіанта ПП техніко-економічного рівня

Коефіцієнт техніко-економічного рівня розраховується за формулою:

$$K_{TEPj} = K_{Kj} / C_{Фj}. \quad (21)$$

Відповідно, для наявних варіантів розробки:

$$K_{TEP1} = 0,3321 / 101881,6 = 0.00000325966 = 3,25966 * 10^{-6};$$

$$K_{TEP2} = 0,5497 / 94576,9 = 0.0000058122 = 5,8122 * 10^{-6}.$$

Отже, як можна побачити, найбільш ефективним варіантом є другий варіант реалізації, оскільки він має вищий коефіцієнт техніко-економічного рівня.

4.7 Висновки

В даному розділі проведено повний функціонально-вартісний аналіз ПП, який було розроблено в рамках дипломного проекту. Процес аналізу можна умовно розділити на дві частини.

В першій з них проведено дослідження ПП з технічної точки зору: було визначено основні функції ПП та сформовано множину варіантів їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості було обчислено коефіцієнт технічного рівня, який і дав змогу визначити оптимальну з технічної точки зору альтернативу реалізації функцій ПП.

Другу частину ФВА присвячено вибору із альтернативних варіантів реалізації найбільш економічно обґрунтованого. Порівняння запропонованих варіантів реалізації в рамках даної частини виконувалось за коефіцієнтом ефективності, для обчислення якого були обчислені такі допоміжні параметри, як трудомісткість, витрати на заробітну плату, накладні витрати.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання автоматизованої системи організації документообігу кафедри оптимальним є другий варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{TEP}} = 5,8122 * 10^{-6}$.

Цей варіант реалізації програмного продукту має такі параметри:

- СУБД: MySQL;
- технології розробки інтерфейсу користувача: JavaScript, HTML/CSS;
- JSON для взаємодії клієнтської та серверної частини.

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, непоганий функціонал і швидкодію.

ВИСНОВКИ

У даній роботі проводилася розробка компонентів автоматизованої системи «Документообіг кафедри» – бази даних та інтерфейсу користувача.

Для виконання поставленого завдання на першому етапі роботи було проведено аналіз основних нормативних документів, що впливають на організацію навчального процесу. Серед цих документів: навчальний план, робочий навчальний план, форма розрахунку навчального навантаження К-З. При аналізі було досліджено основні особливості даних документів, їх структуру, дані, що містять ці документи, та в якому виді вони подаються.

На основі отриманих результатів першої частини роботи, а саме проаналізованих обсягів даних, які містяться у відповідних документах, у другій частині було сформовано абстрактну модель, яка розкривала характер взаємовідношень між інформацією.

Проведено дослідження наявних моделей баз даних, та обрано найбільш зручну для реалізації поставленого завдання – реляційну модель. Спираючись на це, було проведено аналіз доступних для використання в розробці систем управління базами даних. Обрана СУБД MySQL задовольняє своїм функціоналом, допоміжним програмним забезпеченням, а також доступністю довідкових матеріалів потреби, що виникають при вирішенні даної задачі. Також дана СУБД є безкоштовною, тобто не накладає фінансового навантаження на кафедру при її використанні.

Наступним етапом була реалізація на основі абстрактної моделі даних відповідних таблиць БД, які будуть обслуговувати серверну частину, забезпечуючи її необхідною інформацією для генерації документів. Створені таблиці охоплюють усю необхідну на даному етапі розробки АС дані, при цьому уникаючи надлишкових повторень даних, та суперечностей.

Останнім етапом розробки системи була побудова інтерфейсу

користувача. Створений інтерфейс відповідає клієнтській частині клієнт-серверної архітектури, яка взаємодіє з серверною частиною за допомогою HTTP-запитів. При розробці використані сучасні інструменти веб-розробки, що дозволило створити простий та зрозумілий інтерфейс користувача, який дозволяє проводити маніпуляції з даними, наявними у БД, а також керувати процесом генерації документів.

Останнім етапом виконання дипломної роботи було проведення функціонально-вартісного аналізу компонентів, що розроблялися. За допомогою цього аналізу продемонстровано економічну ефективність розроблених компонентів у порівнянні з іншими можливими шляхами вирішення проблеми.

Отже, у даній роботі було розроблено компоненти автоматизованої системи документообігу кафедри, яка являє собою ґрунтовну основу для подальшого розширення функціональності та оптимізації здобутих результатів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Рекомендації щодо розроблення навчальних планів [ТЕКСТ] / Уклад. В. П. Головенкін. – К. : Нац. техн. ун-т України «Київ. політех. ін-т», 2012. – 23 с. – 250 прим.
2. Database [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Database>.
3. Database [Електронний ресурс] // Oracle FAQ. – 2008. – Режим доступу до ресурсу: <http://www.orafaq.com/wiki/Database>.
4. PostgreSQL Wiki [Електронний ресурс] // postgresql.org – Режим доступу до ресурсу: https://wiki.postgresql.org/wiki/Main_Page.
5. Introduction to the Oracle Database [Електронний ресурс] // Oracle. – 2017. – Режим доступу до ресурсу: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm.
6. Microsoft SQL Server [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_SQL_Server.
7. MySQL [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/MySQL>.
8. DB-Engines Ranking [Електронний ресурс] // DB-engines. – 2017. – Режим доступу до ресурсу: <https://db-engines.com/en/ranking>.
9. Branson T. 8 Major Advantages of Using MySQL [Електронний ресурс] / Tony Branson // Datamation – Режим доступу до ресурсу: <http://www.datamation.com/storage/8-major-advantages-of-using-mysql.html>.

ДОДАТОК А

Схема БД

