

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ
ФІКСУВАННЯ ТА КОНТРОЛЮ ІНФОРМАЦІЇ ПРО НАВЧАННЯ
СТУДЕНТІВ З МОЖЛИВІСТЮ ІНТЕГРАЦІЇ ЗІ СТОРОННІМИ РЕСУРСАМИ
RESEARCH AND SOFTWARE IMPLEMENTATION OF THE SYSTEM FOR
RECORDING AND MONITORING THE INFORMATION ON STUDENT
LEARNING WITH THE OPTION OF INTEGRATING WITH THIRD-PARTY
RESOURCES

Виконав: студент(ка) 2 курсу, групи КНТ-129м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Шлома Р.О.

(прізвище та ініціали)

Керівник Каплієнко Т.І.

(прізвище та ініціали)

Рецензент Шитікова О.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Шломи Романа Олеговича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація системи фіксування та контролю інформації про навчання студентів з можливістю інтеграції зі сторонніми ресурсами. Research and Software Implementation of the System for Recording and Monitoring the Information on Student Learning with the Option of Integrating with Third-Party Resources

керівник проєкту (роботи) Каплієнко Тетяна Ігорівна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “03” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 16 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Аналіз вимог. 3. Розробка архітектури. 4. Основні рішення щодо реалізації компонентів системи. 5. Експлуатація, тестування, експериментальне дослідження. 6. Економіко-організаційна частина. 7. Охорона праці та безпека в надзвичайних ситуаціях. 8. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-6 Основна частина	Каплієнко Т.І., доцент		
Економіко-організаційна частина	Пожуєва Т.О., професор		
Охорона праці та цивільна безпека	Коробко О.В., ст. викладач		
Нормоконтролер	Камінська Ж.К., асистент		

7. Дата видачі завдання “ 1 ” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Аналіз вимог	4–5 тижні	Розділ 2
4	Розробка архітектури	6 тиждень	Розділ 3
5	Розробка програми.	7–8 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 5
7	Розробка економіко-організаційної частини	10 тиждень	Розділ 6
8	Розробка заходів з охорони праці та безпеки в надзвичайних ситуаціях.	11 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14–15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

(підпис)

Р.О. Шлома
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Т.І. Каплієнко
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до випускної кваліфікаційної роботи магістра:
130 с., 16 табл., 48 рис., 2 дод., 29 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, РУХ КОНТИНГЕНТУ, БАЗА ДАНИХ,
MICROSOFT ACCESS, MICROSOFT EXCEL, VBA.

Об'єкт дослідження – методи фіксування та контролю інформації про навчання студентів.

Предмет дослідження – комп'ютерні програми для фіксування та контролю інформації про навчання студентів у межах університету.

Мета роботи – дослідження та програмна реалізація програмної системи для фіксування та контролю інформації про процес навчання студентів в закладі вищої освіти.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, реляційна база даних, мова програмування VBA, персональний комп'ютер з офісним пакетом Microsoft Office 2019 під управлінням операційної системи Microsoft Windows 7 або 10.

Результати. Створено реляційну базу даних і програмно-інтерфейсну частину для роботи з нею під керівництвом системи управління базами даних Microsoft Access.

Висновки. Розроблено програму і реляційну базу даних для роботи із контингентом університету, які дозволяють автоматизувати, спростити і прискорити роботу із великим об'ємом даних.

Галузь використання – автоматизація та контроль інформації про навчання студентів у закладах вищої освіти.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 130 pages, 48 figures, 16 tables, 2 appendixes, 29 sources.

**INFORMATION SYSTEM, CONTINGENT MOVEMENT, DATABASE
MICROSOFT ACCESS, MICROSOFT EXCEL, VBA.**

The object of research - methods for recording and monitoring the information on student learning.

The subject of the research is computer programs for recording and monitoring the information on student learning within the university.

The purpose of the work - research and software implementation of a system for recording and monitoring the information on student learning in higher education institutions.

Materials, methods and technical means: structural and object-oriented programming, relational database, VBA programming language, personal computer with office suite Microsoft Office 2019 running Microsoft Windows 7 or 10.

Results. A relational database and software-interface part for working with it under the guidance of the Microsoft Access database management system have been created.

Conclusions. An application and a relational database for working with the university contingent have been developed, which allow automating, simplifying and speeding up work with a large amount of data.

Field of use - automation and control of information about students' learning in higher education institutions.

ЗМІСТ

	С.
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК	8
ВСТУП.....	9
1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ...	11
1.1 Аналіз предметної області	11
1.2 Формальна постановка задачі	14
1.3 Існуюча система	15
1.4 Наявні рішення поставленої задачі	18
1.5 Висновки першого розділу	21
2 АНАЛІЗ ВИМОГ	23
2.1 Вимоги користувача	23
2.2 Нефункціональні вимоги	24
2.3 Функціональні вимоги	24
2.4 Вимоги до даних.....	25
3 РОЗРОБКА АРХІТЕКТУРИ	27
3.1 Вибір середовища розробки.....	27
3.2 Архітектура системи та її місце в інформаційній мережі ЗВО	27
3.3 Концептуальна модель бази даних	29
4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ	31
4.1 Імпорт даних із попередньої системи.....	31
4.2 Схема бази даних.....	31
4.3 Розробка методів розрахунку вакансій та ліцензії	40
5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ, ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ	43
5.1 Порядок дій для підключення системи	43
5.2 Завантаження даних	43
5.3 Опис роботи системи	44
5.4 Тестування, введення в експлуатацію	59
6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА	61

6.1 Вступ	61
6.2 Планування розробки	61
6.3 Визначення витрат на розробку.....	62
6.4 Економічний ефект та економічна ефективність.....	68
7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	69
7.1 Аналіз потенційних небезпек	69
7.2 Заходи із забезпечення безпеки	70
7.3 Заходи з забезпечення виробничої санітарії та гігієни праці.....	74
7.4 Заходи безпеки у надзвичайних ситуаціях.....	75
ВИСНОВКИ	82
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	83
ДОДАТОК А Текст програми	86
ДОДАТОК Б Слайди презентації	121

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- API – Application Programming Interface;
CRM – Customer Relationship Management;
VBA – Visual Basic for Applications;
ЗВО – Заклад Вищої Освіти;
ІС – Інформаційна Система;
МС – Молодший Спеціаліст;
НВ – Навчальний відділ;
ПЗСО – Повна Загальна Середня Освіта;
СУБД – Система Управління Базами Даних.

ВСТУП

Необхідність у зберіганні і обробці даних завжди є і буде актуальною темою для великих установ. Для закладу вищої освіти (ЗВО) дане питання займає особливе місце, так як великі об'єми даних про студентів і освітні послуги потрібно не тільки зберігати, а й обробляти і мати можливість в будь-який момент звернутися до даних минулих років, які все ще є важливими і необхідними. На даний момент всі дані про навчання студентів знаходяться у файлах Excel, що створює проблему зручності, ефективності, швидкості використання і архівного зберігання. Також при завершенні кожного місяця на основі цих даних потрібно створювати різноманітні звіти, через що потрібно витрачати безліч часу на збір даних і подачу у потрібному вигляді. У ході формування звітності з огляду на людський фактор можуть виникати помилки, розбіжності, через які виникає гостра необхідність в пошуку розбіжностей та повторному створенню звіту.

Робота присвячена актуальній проблемі ефективної організації процесу моніторингу руху контингенту закладів вищої освіти, зокрема розробці схеми бази даних для інформаційної системи з метою автоматизації формування статистичних звітів та контролю дотримання ліцензійних обсягів та обсягів державного замовлення.

Мета розроблення полягає в створенні програмного забезпечення, яке б дозволило б працювати максимально ефективно, а саме: знизити імовірність можливих помилок, забезпечити швидкий обмін даними, цілісність і достовірність даних.

Система, яка буде розроблена, має дозволити імпортувати дані з існуючих систем ведення контингенту, зручно опрацьовувати дані про рух здобувачів вищої освіти, контролювати процес відпрацювання наказів по студентам в Єдиній державній базі даних з питань освіти (ЄДЕБО), швидко отримувати усі типи статистичних звітів.

На базі вище сказаного, постає завдання у дослідженні структури даних, якими університет оперує, і розробці системи, яка б дозволяла ефективно і зручно зберігати дані для подальшої обробки і фіксування, а також зниження можливих ризиків появи помилок і розбіжностей.

Об'єкт дослідження – методи фіксування та контролю інформації про навчання студентів.

Предмет дослідження – комп'ютерні програми для фіксування та контролю інформації про навчання студентів у межах університету.

Мета роботи – дослідження та програмна реалізація програмної системи для фіксування та контролю інформації про процес навчання студентів в закладі вищої освіти.

Для досягнення мети роботи необхідні виконати наступні задачі:

- провести аналіз предметної області, виявити основні процеси обліку руху контингенту студентів, можливі типи наказів щодо зміни статусів студента, виконати їх аналіз;
- розробити та оптимізувати інформаційну модель системи обліку руху контингенту, яка дозволить створити структуру даних та функції інформаційної системи (ІС) для автоматизації обліку;
- спроектувати та програмно реалізувати ІС для Національного університету «Запорізька політехніка», що вирішує завдання обліку руху контингенту здобувачів вищої освіти та інтегрується з ЄДЕБО, як системою по обліку контингенту, що запроваджено на державному рівні;
- проведення тестування та експериментального дослідження;
- створення відповідної програмної документації;
- аналіз економіки та організації виробництва щодо економічного обґрунтування доцільності розробки;
- аналіз стану охорони і умов праці об'єкта проектування.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Для сучасних ЗВО, особливо державного підпорядкування, першочерговим завданням є ведення та моніторинг контингенту студентів. Важливість процесу аналізу та контролю контингенту студентів полягає у необхідності щорічно на початку календарного року на основі аналізу плинності контингенту студентів та прийому здобувачів вищої освіти університет розробляти проектні показники плану прийому та випуску за державним замовленням відповідно до нормативних актів та внутрішніх розпоряджень [1]. Одночасно університет здійснює розрахунок прогнозованих показників кількості ставок науково-педагогічних працівників.

Також в будь-який період часу необхідно мати змогу отримати інформацію про кожного студента, форму навчання та вид фінансування, курс навчання, тощо. Для визначення цих даних необхідно контролювати усі накази, що стосуються процесу навчання даного суб'єкта, починаючи з зарахування. Також необхідно агрегувати дані для визначення групових показників за групами, спеціальностями, курсами та по ЗВО загалом.

Окрім стандартних звітів на певний період часу (наприклад, на 1 число кожного місяця), ЗВО має відповідати на запити на отримання публічної інформації, запити державних установ та відомств статистики. Зважаючи на останні тенденції, окрім показників щодо навчання за формами навчання та видами фінансування, з'являються запити на статистичні показники за статтю та віком студентів, результатами вступних іспитів, причому звітний період може складати до 10 років.

Центральним об'єктом даної предметної області є студент, для якого існує і працює ЗВО. Основними характеристиками студента є його повне ім'я, місце навчання, стать, день народження і інша потрібна персональна інформація. Окремим об'єктом можна виділити місце навчання студента узагальнене під шифром (назвою) групи.

Група несе в собі інформацію про форму навчання, тип фінансування, факультет, кафедру, спеціальність, освітню програму, спеціалізацію, курс, рік.

Навчання студентів в ЗВО може здійснюватися за такими формами:

- денною (очною);
- заочною (дистанційною);
- екстерном – особлива форма навчання для здобуття певного рівня вищої освіти шляхом самостійного вивчення навчальних дисциплін і складання в університеті заліків, екзаменів та проходження інших форм підсумкового контролю, передбачених навчальним планом.

Навчання – це оплачувана послуга, яку надає університет, тому між університетом і здобувачем освіти складається договір. Однак, якщо держава потребує фахівців у певній сфері, то вона ж і оплачує навчання студентів з найвищими балами, які забажали навчатися у потрібному напрямку. Із цього витікає два типи фінансування навчання студента:

- за кошти фізичних осіб (контракт);
- за кошти державного бюджету (бюджет).

Курс – це порядковий номер року навчання. Для здобуття ступеня бакалавра за стандартом потрібно навчатися 4 роки (4 курси, 8 семестрів), для ступеня магістра – 1,5 роки (2 курси, 3 семестри).

Факультет — це структурний підрозділ закладу вищої освіти, що об'єднує не менш як три кафедри та/або лабораторії, які в державних і комунальних закладах вищої освіти у сукупності забезпечують підготовку не менше 200 здобувачів вищої освіти денної форми навчання.

Кафедра — основне об'єднання науково-викладацького складу ЗВО по одній або декількох споріднених навчальних дисциплінах; є адміністративно-структурною складовою вишу. Це базовий структурний підрозділ вищого навчального закладу державної (комунальної) форми власності (його філій, інститутів, факультетів), що провадить освітню, методичну та/або наукову діяльність за певною спеціальністю (спеціалізацією) чи міжгалузевою групою спеціальностей, до складу якого входить не менше п'яти науково-педагогічних

працівників, для яких кафедра є основним місцем роботи, і не менш як три з них мають науковий ступінь або вчене (почесне) звання.

Спеціальність – це комплекс набутих людиною знань і практичних навичок, що дає їй можливість займатися певним родом занять у якійсь галузі діяльності.

Освітня програма – це деталізація фаху, набуття особливої здатності виконувати окремі завдання та обов'язки, які мають особливості, в межах спеціальності тощо. Це додаткові спеціальні знання та навички у будь-якій сфері.

Спеціалізація – це поняття подібне до освітньої програми, але із більш вужчим захватом спеціальних знань та навичок.

Кваліфікація — це наявність підготовки, професійних знань, навичок та досвіду, які дають можливість особі належним чином виконувати відповідну роботу. Для кожної освітньої програми визначена кваліфікація, яку буде отримувати студент під час навчання і отримає після завершення навчання.

Статистичні звіти про рух контингенту формуються в кінці кожного місяця, перевіряються, підписуються і відправляються до архіву. До основних звітів входять:

- «Рух контингенту» - у табличному форматі подається інформація про групи і фактичну кількість студентів у них;
- «Відомості про рух контингенту» - у табличному, строго формованому вигляді подається кількісна інформація про зарахованих, поновлених, відрахованих, переведених студентів, тощо;
- «Ліцензія» - у табличному форматі кількість місць ліцензованого обсягу і кількість зайнятих місць у вигляді активного контингенту;
- «Вакансії» - у табличному форматі подається інформація аналогічно ліцензії, але тільки місця державного замовлення;
- «Ставки» - у табличному форматі на базі активного контингенту і наукового складу університету розраховуються ставки.

Місця ліцензованого обсягу або ліцензія – це максимальна кількість місць для студентів за спеціальністю, курсом і формою навчання.

Місця державного замовлення або бюджет– це місця, на яких студенти навчаються за кошти держави.

За час свого навчання студент може змінювати власну персональну інформацію або місце навчання. У свій момент інформація надзвичайно важлива і для роботи із студентом, і для звітності. При будь-яких змінах відповідальний відділ університету створює наказ, у якому повідомляється, коли, з ким, які і чому сталися зміни. Після дані зміни фіксуються навчально-методичним відділом, а всі накази відправляються до архіву [1].

1.2 Формальна постановка задачі

Для надання коректної інформації необхідно мати інформаційну систему, що дозволить фіксувати інформацію про студента під час навчання у кожний конкретний проміжок часу, з можливістю отримання статистичних показників за попередні періоди та на будь-яку дату, зазначену у запиті на отримання інформації. Для організації цього процесу в 2012 році державою було запроваджено ІС - ЄДЕБО, яка є автоматизованою системою збирання, реєстрації, оброблення, зберігання та захисту відомостей та даних з питань освіти. Власником ЄДЕБО є держава, розпорядником – Міністерство освіти і науки України, а технічним адміністратором – державне підприємство «Інфоресурс» [2]. ЗВО у цій системі є користувачем без права модернізації чи доопрацювання програмного забезпечення, тому, на жаль, не має можливості отримати статистичні звіти за різними типами запитів з ЄДЕБО, хоча по кожному студенту окремо інформація зберігається в історичному порядку.

Зважаючи на вищесказане, ЗВО вимушені розробляти власну ІС для автоматизації роботи з контингентом студентів, підтримки освітнього процесу, генерації звітів та отримання статистичної інформації.

Метою даної роботи є розробка моделі зберігання даних про стан навчання студентів під час отримання ступеня вищої освіти, яка дозволить мінімізувати розмір інформації, що зберігається, і в той же час забезпечити можливість фіксувати інформацію по студентам під час навчання у кожний конкретний проміжок часу, з можливістю отримання статистичних даних за попередні періоди на будь-яку дату з групуванням по різним показникам (форма навчання, вид фінансування, курс, група, стать, вік, тощо).

1.3 Існуюча система

Вже існуючою системою для контролю контингенту є група файлів Excel, в яких у зручному форматі для людини зберігається та сама важлива інформація про навчання студентів.

На рисунку 1.1 зображено документ для роботи із актуальною, останньою інформацією про студентів, а інформація про всі накази зберігається в окремому документі у короткому узагальненому вигляді для подальшого використання у місячній звітності. Система розташування та зберігання даних майже інтуїтивно зрозуміла або швидко освоювана. У поле зору одразу потрапляє дублювання даних в плані шифру групи, освітньої програми, спеціальності і інформації про наказ, що призводить до надлишкового витрачання пам'яті. Контингент поділений сторінками за курсом і формою навчання і на кожній сторінці міститься в середньому 1900 записів для денної і 900 записів для заочної. Для роботи з таким масивом даних необхідна постійна сконцентрованість і уважність, так як невеличка помилка може призвести до розбіжності у статистиці і доведеться звіряти дані контингенту із іншими даними у пошуках тієї самої помилки.

Фактично програма Excel розрахована для роботи із статистикою і однотипною інформацією за допомогою потужного функціоналу, але з ростом даних у таблицях і формул, які постійно обробляють ці дані і повертають результат наростають і вимоги до потужності і часу обробки, що змушує

схилятися до повністю ручної роботи, або максимальної автоматизації (бо людський фактор не виключає імовірність помилок), або пошуку оптимального балансу.

№	Группа	Примечание	МЧ	По данным	Итого	Среднее	И	МЧ	МЧ	МЧ	Оценки по плану	Средняя оценка	Повышение к началу	Наказ	Доп.	РК	Примеч.
1	МО	М-119	Авдеев	Степ	Среднее	1					Тел. Маг	100 Промыслов (класс)					
2	МО	М-119	Васко	Дмитри	Среднее	1					Тел. Маг	100 Промыслов (класс)					
					2						2						
1	МО	М-119	Ванова	Лена	Воскресенский	1					Тел. Маг	100 Промыслов (класс)					
2	МО	М-119	Вардан	Мария	Воскресенский	1					Тел. Маг	100 Промыслов (класс)					
3	МО	М-119	Васильев	Александр	Степав	1					Тел. Маг	100 Промыслов (класс)					
4	МО	М-119	Вино	Семён	Воскресенский	1					Тел. Маг	100 Промыслов (класс)					
5	МО	М-119	Васильева	Светлана	Кавказ	1					Тел. Маг	100 Промыслов (класс)					
6	МО	М-119	Вит	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
7	МО	М-119	Васильева	Ольга	Воскресенский	1					Тел. Маг	100 Промыслов (класс)					
8	МО	М-119	Гурбан	Александр	Магистрат	1					Тел. Маг	100 Промыслов (класс)					
9	МО	М-119	Давид	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
10	МО	М-119	Вино	Светлана	Магистрат	1					Тел. Маг	100 Промыслов (класс)					
11	МО	М-119	Вино	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
12	МО	М-119	Вино	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
13	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
14	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
15	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
16	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
17	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
18	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
19	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
20	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
21	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
22	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
23	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
24	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
25	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
26	МО	М-119	Васильев	Александр	Среднее	1					Тел. Маг	100 Промыслов (класс)					
					24	1					25						
1	МО	М-120	Вино	Александр	Магистрат	1					Тел. Маг	100 Промыслов (класс)					
2	МО	М-120	Вино	Александр	Магистрат	1					Тел. Маг	100 Промыслов (класс)					
3	МО	М-120	Вино	Александр	Магистрат	1					Тел. Маг	100 Промыслов (класс)					
4	МО	М-120	Вино	Александр	Магистрат	1					Тел. Маг	100 Промы					

Рисунок 1.1 – Контингент університету

На рисунку 1.2 зображено документ для роботи із місця державного замовлення. Тут все подається просто і зрозуміло: білі стовпці – місця державного замовлення, червоні – зайняті місця студентами; зелені – вакантні місця. Числа подаються за курсами, ЗО\МС (загальна освіта\молодший спеціаліст) та освітньою програмою. Але навіть досвідченому працівнику доводиться відчувати незручності у пошуку і редагування потрібних даних, до того ж числа фактичного контингенту доводиться редагувати і вносити власноруч.

На рисунку 1.3 зображено документ для роботи із місцями ліцензованого обсягу, який не в такій мірі, але знову ж призводить до аналогічних проблем, що і попередній файл.

На рисунку 1.3 зображено документ для роботи із місцями ліцензованого обсягу, який не в такій мірі, але знову ж призводить до аналогічних проблем, що і попередній файл.

Контингент денної форми навчання на 01.12.19 р.																	
Зведена																	
Спеціальність		Освітня програма		Бакалаври				Магістри				Всього					
				Ліцензія	1 курс	Ліцензія	2 курс	Ліцензія	3 курс	Ліцензія	4 курс	Ліцензія	1 курс	Ліцензія	2 курс	Бакалаври	Магістри
017	Фізична культура і спорт	Фізичне виховання	100	71	100	45	100	43	100	65	44	39	45	35	224	74	
227	Фізична терапія, ерготерапія	Фізична реабілітація	70	54	70	38	70	36	70	55	35	35	35	30	183	65	
016	Спеціальна освіта	Корекційна освіта	25	19	25	23	30	18	15	6	3	3	3	66		6	
242	Туризм	Туризмознавство	170	108	170	114	170	116	170	92	60	27	60	54	430	81	
022	Дизайн	Дизайн	45	41	45	32	40	31	40	27	40	9	40	14	131	23	
035	Філологія	035 41 германські мови та літератури (період включно), перше відділення	85	32	85	27	85	38	85	48	35	20	35	14	145	34	
231	Соціальна робота	Соціальна робота	70	28	70	30	70	28	70	32	15	15	16	14	118	29	
053	Психологія	Психологія	60	10	60	19	60	28	60	31	30	9	30	13	88	22	
054	Соціологія	Соціологія	45	5	45	4	45	5	45		30		30	1	14	1	
052	Політологія	Політологія	40	19	40	26	40	17	40	25	40	6	40		87	6	
292	Міжнародні економічні відносини	Міжнародний бізнес	100	13	100	11	100	22	100	7	35	12	35	23	53	35	
061	Журналістика	Журналістика	75	7	75	10	75	24	75	26	20	10	20	8	67	18	
081	Право	Правознавство	100	52	100	36	100	27	100	74	40	22	40	31	189	53	
262	Правоохоронна діяльність	Правоохоронна діяльність		27						60	56	30		19	27	75	
051	Економіка	Управління персоналом та економіка праці		19		20		22		25		12		14	86	26	
		Міткова справа	150	2	150		150		150		30	3	30		2	3	
		Цифрова економіка та електронний бізнес		2											2		
075	Маркетинг	Маркетинг	30	10	30	15	30	14	30	16	40	4	40	8	57	12	
			ДН(Б)	ДН(К)	ДН(зе)	ЗН(Б)	ЗН(К)	ЗН(зе)									

Рисунок 1.3 – Місця ліцензованого обсягу і фактично зайняті місця

В загальному, існуючою системою на даний момент є група окремих файлів Excel, які не зв'язані між собою програмно, але фактично взаємозалежні. Програма Microsoft Excel теоретично дозволяє не тільки наповнити потрібним функціоналом дану системи, а й зв'язати її між собою, але це викликає величезне навантаження на обчислювальну машину. Також із даною системою може працювати лише одна особа в один проміжок часу, а робота з іншими системами, такими як ЄДЕБО, взагалі знаходиться за межами можливого.

На основі даного огляду і аналізу існуючої системи можна зробити висновок, що система може працювати із інформацією про навчання студентів у межах університету, але з великими обмеженнями і з сумнівними можливостями автоматизації роботи людини. Розглянемо існуючі інформаційні системи для вирішення цих проблем.

1.4 Наявні рішення поставленої задачі

1.4.1 Salesforce for Education

Побудований на основі провідних Customer Relationship Management (CRM) у світі, Education Cloud [3] поєднує студентів, персонал та випускників з інтегрованими рішеннями для сучасної освіти, адже успіх студентів починається з успіху університету.

Education Cloud пропонує рішення для ЗВО у таких сферах:

- прийому;
- контролю успіху студентів;
- просування навчання студентів;
- маркетинг та комунікації.

Education Cloud пропонує рішення, які допоможуть прийняти та записати інформацію про студентів, забезпечити персоналізовану підтримку студентів у масштабі ЗВО [3].

1.4.2 Oracle Higher Education Cloud

Oracle Higher Education Cloud спрощує навчання студентів з допомогою комплексного програмного забезпечення для фінансування, управління персоналом і студентами, а також перевіреної хмарної платформи.

Хмарна інфраструктура другого покоління Oracle представляє фундаментальну реконструкцію звичайної публічної хмари. Безпека забезпечується від ядра до краю в хмарній інфраструктурі Oracle, багаторівневому, машинному навчанні - заснована на безпеці дизайну, що допомагає університетам мінімізувати ризик, передбачаючи, виявляючи та реагуючи складні загрози. Завдяки самокерованому, самозахисному та самовідновлюваному характеру, автономна база даних Oracle мінімізує час простою, зменшує ручні оперативні зусилля та надає ІТ-співробітникам та менеджерам більше часу для інновацій [4]-[5].

1.4.3 Workday Student

Workday Student - це наскрізна інформаційна система життєвого циклу студентів та викладачів, яка розроблена для безперебійної роботи з фінансовим управлінням Workday, управлінням людським капіталом Workday, оплатою праці та управлінням грантами. Він об'єднує цілий кампус ЗВО під єдиним застосунком, який працює у хмарі та легко налаштовується [6].

1.4.4 Порівняння існуючих систем

Комерційні програмні застосунки переважно є платними і не відповідають поставленим вимогам, а саме не забезпечують:

- можливість взаємодії із системою ЄДЕБО;

- можливість доопрацювання системи;
- можливість зміни системи враховуючи особливості ЗВО.

Порівняння всіх переглянутих систем подано у таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця існуючих систем

Пункт\Система	Salesforce for Education	Oracle Higher Education Cloud	Workday Student	Розроблювана система
Вартість	Платна	Платна	Платна	Безкоштовна
Можливість взаємодії з іншими системами	-	-	-	+
Простота використання	+	+	-	+
Хмарна база даних	+	+	+	-
Необхідність постійного підключення до інтернету	+	+	+	-
Право власності	-	-	-	+
Можливість взаємодії з ЄДЕБО	-	-	-	+

Існує багато інших публікацій про автоматизацію управління закладами вищої освіти, про планування змін контингенту студентів, аналіз та прогноз кількості здобувачів вищої освіти та інші [7]-[12]. Майже кожний ЗВО

проводить дослідження та розробляє комплекси економіко-математичних моделей аналізу та прогнозування руху контингенту студентів ЗВО для надання звітності та вдосконалення інструментарію підтримки прийняття ефективних управлінських рішень. У той же час автоматизація зазвичай проводиться або у вигляді спеціалізованих рішень, або у спробах розробити універсальне рішення. В першому випадку, проблеми виникають через відсутність зв'язків між різними підрозділами ЗВО та інтеграції даних як в межах ЗВО, так і між загальнодержавними базами даних. В другому випадку, проблеми полягають у:

- відсутності можливості автоматизованого перенесення існуючих масивів інформації у нову систему;
- відсутності можливості самостійно додавати новий функціонал;
- неможливості врахувати особливості ЗВО;
- відсутності інтеграції з ЄДЕБО;
- відсутність інтеграції з іншими підрозділами ЗВО.

Через проблеми, що існують з універсальними рішеннями, постає завдання реалізації в рамках ЗВО Національний університет «Запорізька політехніка» ІС для автоматизації роботи з контингентом студентів, підтримки освітнього процесу, генерації звітів та отримання статистичної інформації, що буде враховувати досягнення у цій сфері в інших закладах освіти та дозволить інтегрувати дані з ЄДЕБО за допомогою системи Application Programming Interface (API).

1.5 Висновки першого розділу

У першому розділі було проведено аналіз предметної області, наведено основні сутності та терміни, опис існуючої системи та її недоліки, фактична постановка проблем до вирішення та описано існуючі аналоги, які не здатні у повній мірі задовольнити всі вимоги. Хоч ЄДЕБО і є найближчою системою

до бажаного, однак як вже було сказано, модифікувати за бажанням дану систему і отримати зведену інформацію не можливо.

Виконавши аналіз предметної області, можна зробити висновок про те, що вимагається від системи, що розробляється, а саме:

- можливість автоматизованого перенесення існуючих масивів інформації у нову систему;
- можливість самостійно додавати новий функціонал;
- можливість врахувати особливості ЗВО;
- інтеграція з ЄДЕБО;
- інтеграція з програмним забезпеченням інших підрозділів ЗВО.

2 АНАЛІЗ ВИМОГ

2.1 Вимоги користувача

Основні вимоги користувача зображено на рисунку 2.1 у вигляді use case діаграми.

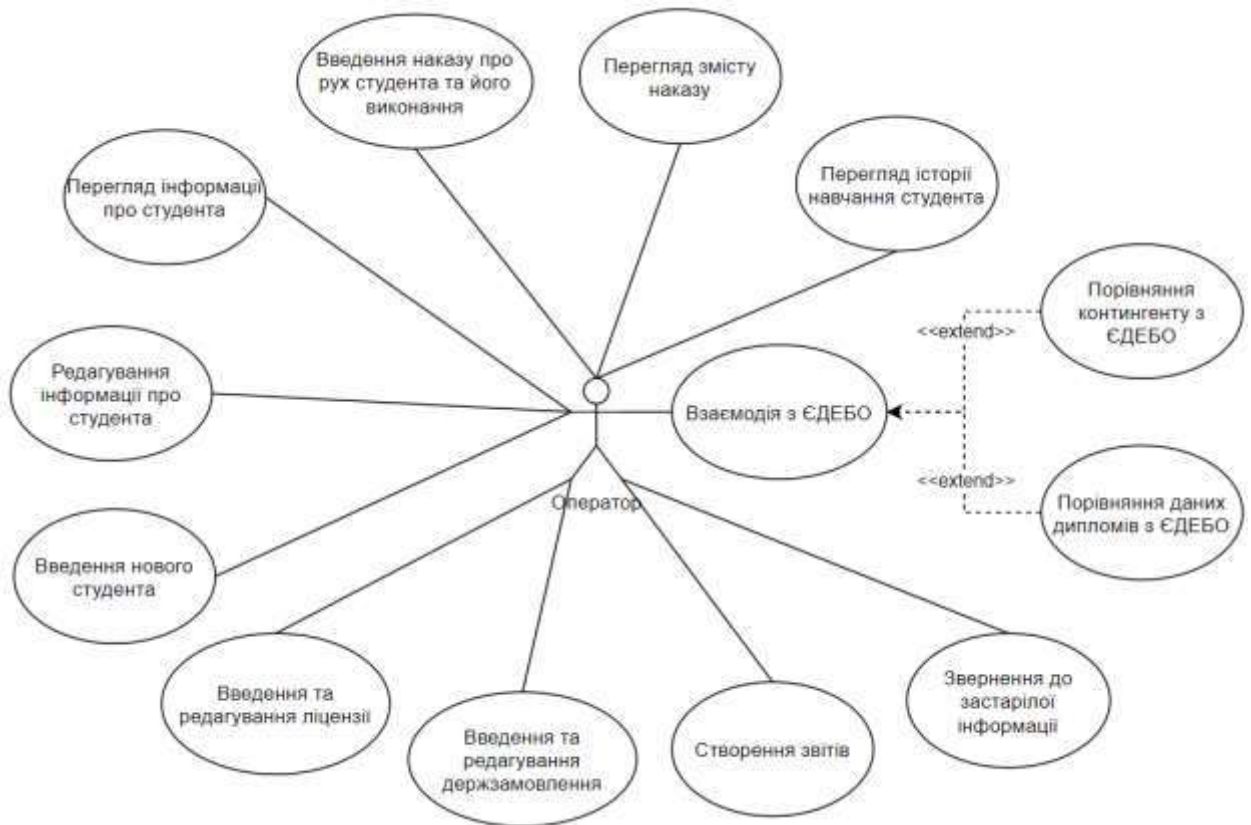


Рисунок 2.1 – Use case діаграма

До ключових вимог користувача, яким є співробітник навчального відділу, належать:

- внесення нових студентів;
- відпрацювання наказів про рух контингенту;
- перегляд інформації про рух та життя контингенту;
- генерація звітності на базі інформації про рух контингенту та історії навчання студентів;
- можливість порівняння даних про рух контингенту та замовлені дипломи з даними в ЄДЕБО.

2.2 Нефункціональні вимоги

Персональний комп'ютер для роботи із контингентом повинен задовольняти такі вимоги:

- операційна система: Windows 10 (x64);
- оперативна пам'ять: 4ГБ;
- постійна пам'ять 10ГБ;
- наявність пакету Microsoft Office 2019.

Також система повинна задовольняти такі загальні вимоги:

- ергономічний та інтуїтивно зрозумілий інтерфейс;
- безпека та конфіденційність даних;
- надійність і відновлюваність системи;
- швидкодія.

2.3 Функціональні вимоги

Розроблювана система повинна надавати можливість виконувати такі дії:

- вносити нових студентів до контингенту;
- відпрацьовувати накази про рух контингенту та фіксувати дані історії навчання студента;
- завантажувати контингент із попередньої системи;
- завантажувати новий контингент після вступної кампанії із системи приймальної комісії;
- з'єднуватися з ІС ЄДЕБО та виконувати порівняння даних для виявлення помилок.

2.4 Вимоги до даних

Перш за все необхідно оцінити кількість даних, які необхідно зберігати та оброблювати. Прийmemo за n_x – кількість студентів, які навчаються у ЗВО у даний період часу (по кожному студенту необхідно зберігати його особисту інформацію, стать, дата народження, тощо), за n_y – кількість наказів, які фіксують та затверджують рух студента у межах університету, починаючи зарахуванням і закінчуючи відрахуванням з університету. Такими рухами також можуть бути переведення на іншу форму навчання, зміна фінансування, поновлення, відрахування і тому подібне. За час свого навчання у ЗВО студент може здійснювати безліч таких рухів, що вимагає зберігати деякий об'єм даних за кожним студентом у впорядковану вигляді (фіксується дата початку дії наказу, курс, форма та вид навчання, та інші дані в залежності від типу наказу). Мінімальне значення n_y для успішного студента, який отримує освітній ступінь бакалавра, дорівнює 5 (зарахування, 3 переведення на курс та завершення навчання), для освітнього ступеня магістр $n_y = 3$. У випадку зміни форми навчання, виду фінансування, надання академвідпустки, відрахування та поновлення, значення n_y майже необмежене.

Відповідно, для зберігання інформації про поточний склад контингенту необхідно зберігання $X = \langle x, y \rangle$, де X – є сукупністю кортежей з особистими даними x та кортежей з даними щодо навчання y , де $y = \{y_0, \dots, y_{n_y}\}$.

ЗВО, в якому планується впровадження даної системи, є Національний університет «Запорізька політехніка». Середня кількість студентів бакалаврату у цьому ЗВО дорівнює 8000, студентів магістратури – 2000. Відповідно, кількість кортежей даних, що необхідно зберігати для оцінки поточного складу контингенту дорівнює приблизно 46000 (для отримання цього значення будемо використовувати припущення, що кількість студентів, які мають більшу кількість наказів, ніж 5 для бакалаврату та 3 для

магістратури, приблизно дорівнює кількості студентів, які не завершили навчання і, відповідно, мають меншу кількість наказів).

Для формування відповідей на запити статистичних даних, окрім поточного складу контингенту, необхідно зберігати дані про студентів минулих років (мінімум за останні 10 років). Контингент студентів бакалаврату оновлюється кожен рік на 25% (четвертий курс випускається та набирається новий перший курс), студентів магістратури – на 50%. Відповідно, кількість записів про студентів минулих років дорівнює $n_{old} = c_{years} \times (0.25 \times n_{xb} + 0.5 \times n_{xm})$, де c_{years} – кількість років, за якими збирається статистика, n_{xb} – кількість студентів бакалаврату, n_{xm} – кількість студентів магістратури.

Відповідно до цих розрахунків для ЗВО «Запорізька політехніка» отримуємо, що загальна кількість студентів бакалаврату (поточний контингент та контингент попередніх 10 років) дорівнює 28 000 та магістратури 12 000 ($n_x = 40000$). Кількість кортежей даних, які необхідно зберігати для оцінки стану поточного контингенту та контингенту попередніх років, буде дорівнювати 200 000. Така кількість записів потребує детального планування структури даних та методів зберігання та обробки цих даних.

На основі проведеного аналізу можна зробити висновок, що необхідно створити інформаційну модель та базу даних (БД) для зберігання усієї інформації щодо зміни статусів навчання кожного студента з можливістю отримати комплексну інформацію по контингенту на будь-яку дату у розрізі різних угруповань (по ЗВО в цілому, за спеціальністю, курсом, формою навчання чи видом фінансування, тощо).

3 РОЗРОБКА АРХІТЕКТУРИ

3.1 Вибір середовища розробки

Для вирішення задачі зберігання, обробки, перегляду та аналізу великого об'єму даних про життя та рух контингенту була обрана реляційна система управління базами даних (СУБД) Microsoft Access 2019. Вона дозволяє створювати реляційні бази даних за допомогою простого та інтуїтивно зрозумілого інтерфейсу, який дозволяє навчатися і паралельно створювати власну БД. Окрім самої бази даних, система дозволяє створювати власні форми, запити та звіти на базі даних, а вбудоване середовище розробки на базі мови VBA (Visual Basic for Applications) та макроси дозволяють створити своє програмне забезпечення з нуля задовольняючи всі вимоги.

Окрім вбудованих можливостей створення звітів, система надає можливості імпорту/експорту даних у інші застосунки офісного пакету Microsoft, а за допомогою VBA можна написати алгоритм створення звітів за необхідним шаблоном, обмежуючись лише своєю уявою та можливостями програмних застосунків офісного пакету.

3.2 Архітектура системи та її місце в інформаційній мережі ЗВО

Архітектура роботи та взаємодії розроблюваної системи з іншими необхідними системами зображена на рисунку 3.1 у вигляді діаграми компонентів.

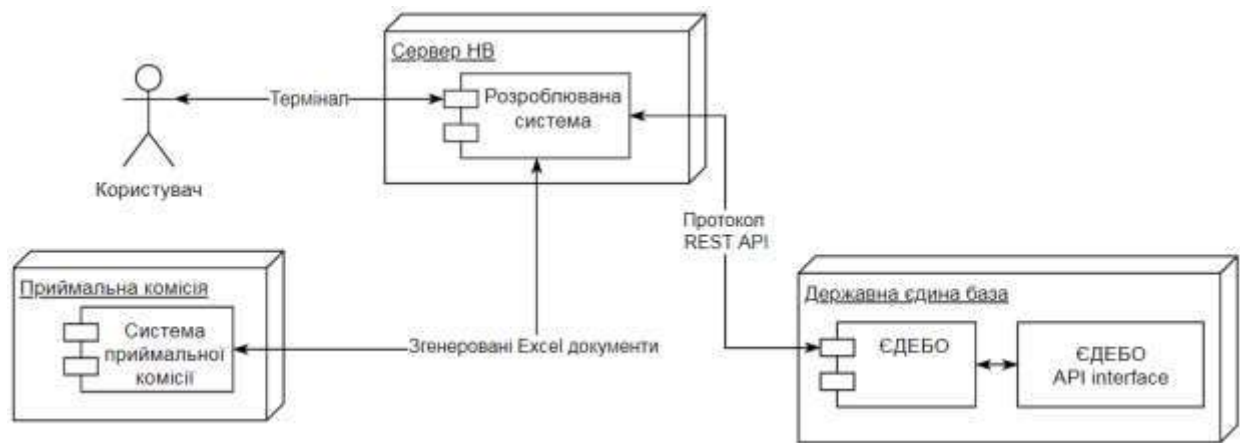


Рисунок 3.1 – Діаграма компонентів

Користувач через термінал входить до системи контролю контингенту, яка знаходиться на сервері навчального відділу (НВ) і працює з контингентом всіма доступними і потрібними способами, використовуючи графічний інтерфейс.

Для роботи з ЄДЕБО система університету зв'язується протоколом REST API із державною базою і через доступний API інтерфейс вивантажує потрібні дані до бази університету. Після разового завантаження даних необхідність в ЄДЕБО відпадає до наступного підключення, а завантажені дані можна обробляти та порівнювати із даними бази.

Для роботи з базою приймальної комісії було прийнято рішення завантажувати абітурієнтів, які вступили до університету і готові до початку навчання, за допомогою імпорту файлу Excel, так як система приймальної комісії знаходиться під управлінням застарілої СУБД Microsoft Access 2003 року, яка некоректно працює з її молодшою і актуальнішою версією 2019 року. Однак, даний спосіб обміну даних не приносить великих незручностей і внесення нових студентів (більше 2 тисяч) виконується від 60 до 180 секунд. Оновлення бази даних приймальної комісії до рівня системи контролю контингентом дозволить обмінюватися даними у режимі реального часу без відчутної мінімальної затримки.

3.3 Концептуальна модель бази даних

Для розроблення інформаційної системи було вирішено:

- побудувати інформаційну модель руху студентського контингенту з прив'язкою до навчальних груп та структурних підрозділів університету;
- створити базу даних для збереження інформації, яка буде задовольняти вимогам моделі, та інтуїтивно зрозумілий інтерфейс для внесення, редагування та аналізу даних працівниками навчального відділу університету;
- створити інтерфейси та функціонал для інтеграції з ЄДЕБО та генерації стандартних звітів щодо руху контингенту студентів.

На першому етапі розробки потрібної системи було виділено основні сутності, а саме: студент, група, освітня програма (спеціалізація), спеціальність, кафедра, факультет, наказ, пункт наказу, зміст наказу.

Одиницею контингенту є студент, який обов'язково повинен мати прізвище, ім'я, по батькові, тип фінансування і стать як особисту і не повторну інформацію.

Так як у одній групі може знаходитися декілька студентів, сутність «Група» виділяється як окрема і зв'язується із сутністю «Студент» зв'язком один до багатьох. Основні атрибути групи це – назва, форма навчання, курс, освітня програма.

«Освітня програма» є беззаперечно самостійною сутністю, яка затверджується в межах університету і присвоюється кожній групі. Так як кожна програма може повторятися у декількох груп, які відрізняються формою навчання, курсом або порядковим номером (якщо кількість студентів для однієї групи перебільшує допустиму кількість), тому між цими сутностями встановлюється теж зв'язок один до багатьох.

Сутності «Факультет», «Кафедра», «Спеціальність» через структурну складність взаємодії було прийнято рішення об'єднати з найменшою

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Імпорт даних із попередньої системи

Так як поточна кількість даних про студентів на момент завантаження даних до розробленої ІС сягала понад 10000 записів, перенесення вручну б виконувалося достатньо довго, щоб дані втратили свою актуальність, а, беручи до уваги людський фактор, при перенесенні було б допущено безліч помилок, які призвели б до великих проблем достовірності даних.

Завдяки родинним зв'язкам Microsoft Excel та Microsoft Access та середовищу розробки на базі мови VBA був створений алгоритм зчитування даних з попередньої бази до нової, який виконує повне зчитування приблизно за 2 хвилини. Так як робота із передньою системою була переважно ручна, у ній містилися помилки даних, які були виявлені під час роботи алгоритму по перенесенню. Після внесення коректив ключових даних для зчитування, система без проблем могла зчитувати всі дані за короткий проміжок часу та без участі людини.

4.2 Схема бази даних

На основі аналізу предметної області та концептуальної моделі у СУБД Access 2019 була створена БД, схема якої представлена на рисунку 4.1. До схеми були включені основні таблиці, у яких зберігається інформація про студентів, факультети, кафедри, спеціальності і так далі. У БД також наявні буферні таблиці, наприклад, для ЄДЕБО, структура яких змінюється за вимогою зовнішніх або внутрішніх вимог, а також статичні таблиці-довідники, дані яких є строго фіксованими і містять довідникову інформацію.

Обов'язковим атрибутом кожної таблиці є унікальний ключ, але при створенні самої БД більшість сутностей перетерпіли розширення для рішення як основних задач, так і додаткових, які не були взяті до уваги на момент створення концептуальної моделі.

Таблиця «_Student» представляє собою сутність «Студент». В даній таблиці пізніше було прийнято рішення вести поля `protocol_code` та `protocol_date` для зберігання даних протоколу про захист дипломного проекту і автоматизованого створення звітності щодо випуску студентів. Для більш тісної взаємодії з ЄДЕБО були введені ключові поля із державної системи, а саме: `Id_Person`, `PersonCodeU`, `educationId`. Для автоматизації було введено поле `id_moodle`, у якому міститься код профілю moodle студента, для роботи із студентом у системі Moodle. Також було додано мітку іноземця (`isForeigner`),

яка дозволяє фіксувати іноземних студентів і у подальших випадках швидко отримувати дані про всі іноземців на навчанні. Повний перелік полів подано у таблиці 4.1.

Таблиця 4.1 – Поля таблиці «_Student»

Назва поля	Тип	Опис
id_stud	Лічильник, первинний ключ	Неповторний код студента
last_name	Текстовий	Прізвище
name_stud	Текстовий	Ім'я
middle_name	Текстовий	По батькові
sex	Числовий	Код статі
type_financing	Текстовий	Тип фінансування
bool	Логічний	Мітка
protocol_code	Числовий	Код протоколу випуску
protocol_date	Дата	Дата протоколу випуску
isForeigner	Логічний	Мітка іноземця
id_group	Числовий	Код групи
Id_Person	Числовий	Код персони в ЄДЕБО
PersonCodeU	Текстовий	Персональний код в ЄДЕБО
educationId	Числовий	Код здобувача освіти в ЄДЕБО
date_birth	Дата	Дата народження
id_moodle	Числовий	Код профіля moodle

Таблиця «_Group» представляє собою сутність «Група». До цієї таблиці були додані поля року вступу групи (year_group) і типу групи (type_group) для простішого і швидшого аналізу даних. Під типом групи розуміється її особливість, а саме: тип 1 – група для студентів бакалаврів після здобуття повної загальної середньої освіти (ПЗСО), тип 2 – група для студентів бакалаврів після здобуття освітньо-кваліфікаційного рівня молодшого

спеціаліста, тип 3 – група для студентів магістрів, тип 4 – група для іноземних студентів. Повний перелік полів подано у таблиці 4.2.

Таблиця 4.2 – Поля таблиці «_Group»

Назва поля	Тип	Опис
id_group	Лічильник, первинний ключ	Код групи
code_group	Текстовий	Шифр групи
course	Текстовий	Курс
formnavch	Числовий	Код форми навчання
id_specialization	Числовий	Код освітньої програми
year_group	Текстовий	Рік групи
type_group	Числовий	Тип групи

Таблиця «s_Specialization» представляє собою сутність «Спеціалізація». До цієї таблиці були додані поля name_specializationd_abb та num_specialization, які дозволили завантажити дані із попередньої системи роботи з контингентом. Поля for_pzso, for_sp, for_mag були введені для інформування, на кого саме розповсюджувалася освітня програма, а template_group – шаблон групи, за яким утворюється група для відповідної освітньої програми. Повний перелік полів подано у таблиці 4.3.

Таблиця 4.3 – Поля таблиці «s_Specialization»

Назва поля	Тип	Опис
id_specialization	Числовий, первинний ключ	Код освітньої програми
name_specialization	Тестовий	Назва освітньої програми
name_specializationd_abb	Тестовий	Скорочена назва освітньої програми

Продовження таблиці 4.3

Назва поля	Тип	Опис
num_specialization	Тестовий	Внутрішній університетський шифр освітньої програми
id_spec	Тестовий	Шифр спеціальності
id_fac	Числовий	Код факультету
id_kaf	Числовий	Код кафедри
for_pzso	Логічний	Мітка належності освітньої програми для ПЗСО
for_sp	Логічний	Мітка належності освітньої програми для молодших спеціалістів
for_mag	Логічний	Мітка належності освітньої програми для магістрів
bool_activity	Логічний	Мітка активності освітньої програми
budget_2019	Тестовий	Зашифрований бюджет 2019 року
budget_2020	Тестовий	Зашифрований бюджет 2020 року
template_group	Тестовий	Шаблон групи

Таблиця «s_Facultet» представляє собою сутність «Факультет». Повний перелік полів подано у таблиці 4.4.

Таблиця 4.4 – Поля таблиці «s_Facultet»

Назва поля	Тип	Опис
id_fac	Числовий, первинний ключ	Код факультету
name_fac	Текстовий	Назва факультету
sign_fac	Текстовий	Абревіатура факультету
sign_fac_w_f	Текстовий	Абревіатура факультету без слова «факультет»
id_inst	Числовий	Код інституту
out	Числовий	Порядковий номер

Таблиця «s_Kafedra» представляє собою сутність «Кафедра». Повний перелік полів подано у таблиці 4.5.

Таблиця 4.5 – Поля таблиці «s_Kafedra»

Назва поля	Тип	Опис
id_kaf	Числовий, первинний ключ	Код кафедри
name_kaf	Текстовий	Назва кафедри
name_kaf_abb	Текстовий	Абревіатура кафедри

Таблиця «s_Specialty» представляє собою сутність «Спеціальність». Повний перелік полів подано у таблиці 4.6.

Таблиця 4.6 – Поля таблиці «s_Specialty»

Назва поля	Тип	Опис
id_spec	Текстовий, первинний ключ	Шифр спеціальності
name_spec	Текстовий	Назва спеціальності
id_galuz	Текстовий	Код галузі
lic	Текстовий	Зашифрована ліцензія

Таблиця «s_Qualification» представляє собою сутність «Кваліфікація». Для реалізації зв'язку багато до багатьох між кваліфікацією і освітньою програмою була введена додаткова таблиця «join_specialization_qual», якій міститься унікальний ключ з кожної сторони для створення унікальної пари. Повний перелік полів подано у таблиці 4.7.

Таблиця 4.7 – Поля таблиці «s_Qualification»

Назва поля	Тип	Опис
id_qual	Лічильник, первинний ключ	Код кваліфікації
code_qual	Текстовий	Шифр кваліфікації
name_qual_uk	Текстовий	Назва кваліфікації українською мовою
name_qual_en	Текстовий	Назва кваліфікації англійською мовою
type_qual	Числовий	Тип кваліфікації (1 – на прийом, 2 – на випуск)
for_whom	Текстовий	Для якого освітнього ступеня
bool	Логічний	Мітка

Таблиця «_Order» представляє собою сутність «Наказ». Повний перелік полів подано у таблиці 4.8.

Таблиця 4.8 – Поля таблиці «_Order»

Назва поля	Тип	Опис
id_order	Лічильник, первинний ключ	Код наказу
num_order	Текстовий	Шифр наказу
year_order	Текстовий	Рік наказу
dateIn	Дата	Дата наказу

Таблиця «_PointOrder» представляє собою сутність «Пункт наказу». Повний перелік полів подано у таблиці 4.9.

Таблиця 4.9 – Поля таблиці «_PointOrder»

Назва поля	Тип	Опис
id_point	Лічильник, первинний ключ	Код пункту наказу
id_order	Числовий	Код наказу
point	Текстовий	Текст пункту
detail	Текстовий	Текст деталі
another	Текстовий	Текст іншої інформації

Таблиця «_StudentOrder» представляє собою сутність «Студент в наказі». Повний перелік полів подано у таблиці 4.10.

Таблиця 4.10 – Поля таблиці «_StudentOrder»

Назва поля	Тип	Опис
id	Лічильник, первинний ключ	Код кортежу
id_point	Числовий	Код пункту
id_stud	Числовий	Код студента
last_name	Текстовий	Прізвище
name_stud	Текстовий	Ім'я
middle_name	Текстовий	По батькові
type_financing	Текстовий	Тип фінансування
code_group	Текстовий	Шифр групи
id_specialization	Числовий	Код освітньої програми
course	Текстовий	Курс
formnavch	Числовий	Код форми навчання
bool	Логічний	Мітка
id_group	Числовий	Код групи

У ході створення схеми бази даних були виведені ще таблиці для збереження заяв студентів на місця ліцензованого обсягу і держзамовлення «_StudentRequest», які бажають поновитися або перевестися, і таблиця

«a_ChangeName» для збереження змін частин імені. Повний перелік полів даних таблиць подано у таблицях 4.11 і 4.12 відповідно.

Таблиця 4.11 – Поля таблиці «_StudentRequest»

Назва поля	Тип	Опис
id_request	Лічильник, первинний ключ	Код заяви
id_stud	Числовий	Код студента
id_specialization	Числовий	Код освітньої програми
formnavch	Числовий	Код форми навчання
type_financing	Текстовий	Тип фінансування
osv_stup	Текстовий	Освітній ступінь
date_in	Дата	Дата подання заяви

Таблиця 4.12 – Поля таблиці «a_ChangeName»

Назва поля	Тип	Опис
id_stud	Числовий	Код студента
old_name	Текстовий	Стара частина імені
new_name	Текстовий	Нова частина імені
type_name	Текстовий	Частина імені, яка змінюється
num_order	Текстовий	Номер наказу
dateIn	Дата	Дата наказу

Для повноцінної роботи системи були ще введені буферні таблиці (проміжні таблиці зберігання даних, наприклад, ЄДЕБО), таблиці-довідники (перелік стандартних незмінних значень, наприклад, форми навчання і її коди). Таблиці-довідники дозволяють виключити дублювання і надлишкове використання пам'яті. Наприклад, на рисунку 4.2 подано заповнену таблицю-довідник форми навчання. Завдяки цьому у поле форми навчання таблиці «Група» вказується код форми навчання, який займає 1 байт пам'яті, тоді як кожний символ займає вже 1 байт, а мінімальна пам'ять, потрібна для

зберігання текстової назви форми навчання мінімум 6 байт, що в 6 раз перевищує необхідність у обсязі пам'яті.

id_formnavch	formnavch	code	char_formnavch
0	0	0	
1	Денна	d	Д
2	Заочна	z	З

Рисунок 4.2 – Таблиця-довідник форми навчання

4.3 Розробка методів розрахунку вакансій та ліцензії

Ефективність представлення даних саме у такій формі була підтверджена швидкістю та безпомилковістю під час створення звітів про рух контингенту, статистики щодо зайнятих ліцензійних та бюджетних місць. Проте з'явилося питання щодо врахування вакантних ліцензійних місць та місць державного замовлення, що є вкрай важливим моментом для державного ЗВО.

За правилами Міністерства освіти ліцензований обсяг надається загалом на спеціальність, яка може включати декілька освітніх програм. Державне замовлення також надається на спеціальність в цілому, проте потім, в рамках ЗВО, розподіляється і в майбутньому відноситься до освітніх програм.

Для зберігання інформації щодо ліцензійного обсягу спочатку було вирішено додати новий атрибут до сутності «s_Speciality». Проте є мінімум дві форми навчання (денна, заочна), тобто тепер потрібно додати два атрибути; та є ще 6 курсів, що вимагає додати у 6 раз більше атрибутів. Звичайно ліцензію можна було б виділити як окрему сутність, але вона не має такої ваги, лише мінімум 12 однотипних атрибутів. Рішенням цієї проблеми було обрано зберігати ліцензію одним атрибутом сутності, але у зашифрованому вигляді, тобто у вигляді текстового рядку із блоками, які складаються за формулою: перший символ – форма навчання (d – денна, z - заочна), другий символ – курс

(а точніше його порядковий код), а після курсу вказується число, що дорівнює кількості ліцензійних місць. Такий спосіб дозволяє зберігати ліцензію у більш компактному вигляді і не боятися структурних змін, які можуть статися із ліцензованим обсягом. Приклад шифрування зображено на рисунку 4.3 на декількох спеціальностях, а на рисунку 4.4 зображено на основі спеціальності «016 Спеціальна освіта» вже саму форму для роботи із ліцензованим обсягом у розшифрованому, зручному для людини вигляді.

Код	Назва спеціальності	Ліцензія
016	Спеціальна освіта	d135 z125 d225 z235 d325 z335 d53 z53 d430 z430 d63 z63
017	Фізична культура і спорт	d1100 z150 d2100 z250 d3100 z350 d545 z530 d4100 z450 d662 z631
022	Дизайн	d153 z17 d245 z215 d345 z315 d540 z520 d440 z420 d640 z620

Рисунок 4.3 – Приклад зберігання зашифрованого ліцензійного обсягу

	1 курс	2 курс	3 курс	4 курс	1м курс	2м курс
Ліцензія ДН	35	25	25	30	3	3
Факт ДН	32+0	16+0	22+0	18+0	3+0	3+0
Ліцензія ЗН	25	35	35	30	3	3
Факт ЗН	12+0	2+0	6+0	27+0	3+0	3+0

Зберегти

Рисунок 4.4 – Приклад розшифрованого ліцензійного обсягу для спеціальності «016 Спеціальна освіта»

Випадок із зберіганням обсягів державного замовлення (держзамовлення) аналогічний, але трохи складніший. Держзамовлення, крім форми навчання і курсу, поділяється ще за освітнім ступенем групи (для ПЗСО, молодших спеціалістів (МС) та магістрів) і роком, в якому державне замовлення було виділено. Для вирішення цього завдання було прийнято рішення ускладнити формулу шифрування. Перший і другий символи

аналогічно ліцензії відповідають формі навчання і курсу відповідно, третій символ – освітній ступень (ПЗСО – Р, МС - У, М - М), 4 і 5 останні цифри року, 6 службовий символ дефіс і після дефісу подається держзамовлення (на рисунку 4.5).

d1P20-1 d1Y20-7 d2P19-3 d2Y19-4 d3P18-5 d3Y18-6 d3Y19-7 d4P16-8 d4Y16-9 d4Y17-10 d4Y18-11 d5M19-12 d6M19-13

Рисунок 4.5 – Приклад зберігання у зашифрованому вигляді обсягу державного замовлення

Приклад розшифрованих, у зручному для користувача вигляді, місць державного замовлення подано на рисунку 4.6.

	1 курс		2 курс		3 курс			4 курс				1м курс	2м курс
Державне замовлення	ЗО 20	МС 20	ЗО 19	МС 19	ЗО 18	МС 18	МС 19	ЗО 17	МС 17	МС 18	МС 19	Mar 20	Mar 19
Факт	0	0	0	0	0	0	0	0	0	0	0	0	0
Валанс	-4	14	-6	0	7	0	14	7	1	-15	0	-4	18
	-4	-14	-6	0	-7	0	-14	-7	-1	-15	0	-4	-18

Рисунок 4.6 – Розшифровані місця державного замовлення

Безпомилковість під час роботи з даними щодо ліцензійних місць та місць державного бюджету дуже важлива, тому що від цієї інформації залежить можливість поновлення / переведення / зарахування студентів та виконання фінансової звітності щодо діяльності ЗВО в цілому.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ, ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

5.1 Порядок дій для підключення системи

Система складається із двох частин: програмної (файл «Soft DMED (client).accdb») і табличної (файл «Data DMED.accdb»). Також до системи додаються файли шаблони типу Excel для створення різних звітів.

У програмній частині знаходиться графічний інтерфейс для роботи із БД, буферні таблиці та таблиці-довідники.

У табличній частині знаходиться схема БД і всі основні дані.

Для коректної роботи система повинна бути розміщена на серверну або звичайну робочу станцію із встановленою системою Windows 7 або 10 і офісним макетом Microsoft Office 2019 року або новішим. Таблична частина із файлами шаблонами розміщується на постійному місці, а у програмній частині вказується місце знаходження даних і шаблонів. Після чого, після відкриття файлу «Soft DMED (client).accdb» можна працювати із системою.

5.2 Завантаження даних

Після остаточної перевірки та завершення тестування, було проведено завантаження реальних даних із існуючої системи та введення системи в експлуатацію.

На рисунку 5.1 зображено форму, за допомогою якої виконується читання даних про контингент із попередньої системи, та достатню для розуміння процесу зчитування даних інструкцію.

Після розрахункового часу у 2 – 3 хвилини всі дані були внесені і доступні у новій системі.

Головна форма (Октябрь)

Головна | Студенти і групи | **Форми ФКССК** | Звіти | Витягування | ЄДЕБО | Читання

Рух контингенту | E:\Projects\Project DMED\Інформація\контингент_2019.xls

	Назва сторінки	Курс	Форма навчання
☒	1к	1	Денна
☒	2к	2	Денна
☒	3к	3	Денна
☒	4к	4	Денна
☒	1к(магістр)	1м	Денна
☒	2к(магістр)	2м	Денна
☒	ЗН 1	1	Заочна
☒	ЗН 2	2	Заочна
☒	ЗН 3	3	Заочна
☒	ЗН 4	4	Заочна
☒	6к(магістри)	4	Денна

Читати | Сторінка: отр.бакалавр

Р1: -
Р2: -

1. Обрати файл для завантаження
2. Видалити мітки навпроти кожної сторінки, яку не потрібно читати
3. Встановити курс і форму навчання для кожної сторінки
4. Почати зчитування
5. Очікувати завершення зчитування

Рисунок 5.1 – Форма для втягування даних із попередньої системи в розроблену з інструкцією

5.3 Опис роботи системи

Якщо всі умови для запуску ІС дотримано, то після відкриття файлу «Soft DMED (client).accdb» відкриється СУБД Access і головна форма графічного інтерфейсу (рис. 5.2).

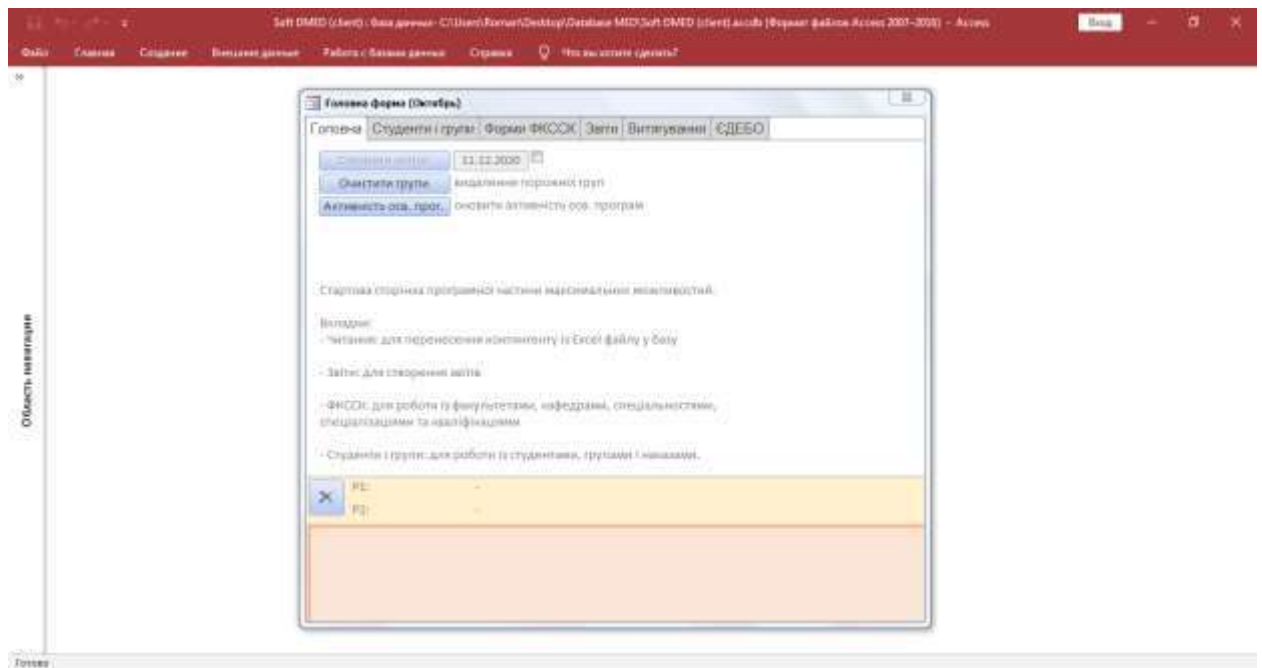


Рисунок 5.2 – Головна форма, початок роботи

Для кожної сфери роботи існує своя вкладка, а саме:

- вкладка «Головна» – стартова сторінка, на якій знаходяться особливі службові кнопки та коротка інструкція по змісту всіх вкладок;
- вкладка «Студенти і групи» – сторінка для роботи із студентами, наказами, протоколами (рис. 5.3);
- вкладка «Форми ФКССК» - сторінка для роботи із факультетами, спеціальностями, освітніми програмами, спеціалізаціями, кваліфікаціями, місця ліцензованого обсягу, місцями державного замовлення (рис. 5.4);
- вкладка «Звіти» - сторінка для генерації звітів (рис. 5.5);
- вкладка «Витягування» - сторінка призначена для різноманітних вивантажування даних у рамках встановленого алгоритму (рис. 5.6);
- вкладка «ЄДЕБО» - сторінка для роботи з ЄДЕБО (рис. 5.7).

The screenshot shows a web application window titled "Головна форма (листопад)". The "Студенти і групи" (Students and Groups) tab is selected. The interface includes a sidebar with buttons: "Контингент", "Накази", and "Протоколи". The main area contains a "Новий студент" (New Student) button, a "Студент" dropdown menu, and a "Редигувати" (Edit) button. Below these are buttons for "Moodle", "Moodle 2", and "Історія імен" (Name History). A "Студенти" (Students) section has two radio buttons: "Навчаються" (Studying) and "Відраховані" (Expelled). At the bottom, there is a yellow status bar with "P1: -" and "P2: -", and a large orange rectangular area below it.

Рисунок 5.3 – Головна форма. Вкладка «Студенти і групи»

The screenshot shows the same web application window, but with the "Форми ФКССК" (Forms FKSSK) tab selected. The sidebar buttons remain the same. The main area displays four buttons with corresponding descriptions: "ФКССК" (Факультети, Кафедри, Спеціальності, Освітні програми), "Ліцензія" (Робота із ліцензією), "Вакансія" (Робота із держзамовленням), and "Кваліфікації" (Робота із кваліфікаціями). The bottom status bar and orange area are identical to the previous screenshot.

Рисунок 5.4 – Головна форма. Вкладка «Форми ФКССК»

На рисунку 5.5 зображено вкладку «Звіти» на головній формі, на рисунку 5.6 – вкладка «Витягування» на головній формі, на рисунку 5.7 – вкладка «ЄДЕБО» на головній формі.

The screenshot shows the 'Звіти' (Reports) tab of the 'Головна форма (листопад)' window. The window has a title bar with a close button and a tab bar with the following tabs: 'Головна', 'Студенти і групи', 'Форми ФКСК', 'Звіти' (active), 'Витягування', and 'ЄДЕБО'. The main content area contains several input fields and buttons. On the left, there are date pickers for 'Початок' (01.11.2020) and 'Кінець' (01.12.2020), a month dropdown set to 'Листопад', and buttons for 'Рух контингенту', 'Ліцензія', 'Вакансії', 'Відомості про рух контингенту', and 'Ставки'. On the right, there is a checkbox for 'Іноземці' which is checked. At the bottom, there is a yellow bar with a close button and labels 'P1:' and 'P2:'.

Рисунок 5.5 – Головна форма. Вкладка «Звіти»

The screenshot shows the 'Витягування' (Extraction) tab of the 'Головна форма (листопад)' window. The window has a title bar with a close button and a tab bar with the following tabs: 'Головна', 'Студенти і групи', 'Форми ФКСК', 'Звіти', 'Витягування' (active), and 'ЄДЕБО'. The main content area contains several input fields and buttons. On the left, there is a 'Що це?' button, an 'Індекс рядка' input field, and buttons for 'Перенести дані' and 'Очистити дані файлу'. In the center, there is a 'Що це?' button and a 'Кафедра/Група/Кількість студентів' input field. On the right, there is a 'Списки для Moodle' section with radio buttons for 'Накази' (selected), 'Списки', and 'За статусом', and a 'Відбір з:' input field. Below this is a 'Зібрати списки' button. At the bottom, there is a yellow bar with a close button and labels 'P1:' and 'P2:'.

Рисунок 5.6 – Головна форма. Вкладка «Витягування»

Рисунок 5.7 – Головна форма. Вкладка «ЄДЕБО»

5.3.1 Вкладка «Студенти і групи»

На даній вкладці є три основні кнопки: «Контингент», «Накази» і «Протоколи».

При натисканні на кнопку «Контингент» відкривається форма для перегляду контингенту (рис. 5.8). За допомогою фільтрів можна відсортовувати і шукати потрібних студентів, а при подвійному натисканні на будь-якого студента відкривається його більш детальна інформація і історія навчання у окремій формі (рис. 5.9). Через цю ж форму можна внести зміни у прізвище, ім'я чи по батькові, подати заяву на місце ліцензованого обсягу чи державного замовлення, переглянути профіль студента у moodle.

Фільтри

Факультет:

Спеціальність:

Освітня програма:

Форма навчання

☒ Денна
☐ Заочна

Курс:

Активність

☒ Навчаються
☐ Відраховані
☐ Заяви
☐ Академічна відпустка
☐ Іноземні студенти
☐ Прийом (плановий)

	Прізвище	Ім'я	По батькові	Б/К	Стать	Група
☒	Абакуменко	Єва	Олегівна	К	Жіноча	МТЕ-127
☐	Абакумова	Оксана	Василівна	Б	Жіноча	РТ-419сп
☐	Абальмас	Антон	Олегович	К	Чоловіча	КНТ-129
☐	Абдулла Ходжі	Беназір		К	Чоловіча	УФКС-217ін
☐	Абдуллаєв	Мамед	Самандар огли	К	Чоловіча	КНТ-117
☐	Абдуллаєв	Раміль	Латіфович	Б	Чоловіча	Т-319м
☐	Абдурахімова	Карина	Гаффоржонівна	Б	Жіноча	Т-310м
☐	Абібулаєва	Катерина	Артурівна	К	Жіноча	КНТ-210
☐	Абілов	Зейнал		К	Чоловіча	УФКС-119
☐	Абілов	Нахід	Заур Огли	К	Чоловіча	Ю-319м
☐	Аблякімов	Сеттар	Наріманович	Б	Чоловіча	КНТ-118
☐	Аболимов	Арсен	Олегович	Б	Чоловіча	ІФ-510сп
☐	Абрамов	Валентин	Андрійович	К	Чоловіча	УФКС-126(ак)
☐	Абрамов	Владислав	Юрійович	К	Чоловіча	Ю-119сп
☐	Абрамов	Максим	Камович	Б	Чоловіча	ФЕУ-110
☐	Абрамчев	Ілля	Кирилович	Б	Чоловіча	М-819
☐	Абрамян	Тимур	Анатолійович	К	Чоловіча	МТЕ-219сп
☐	Абросімова	Валерія	Сергіївна	Б	Жіноча	Е-119м

Кількість: 7506 К: 3234 Ж: 2497 МС К: 218 Ін К: 53
 Б: 4272 Ч: 5009 МС Б: 1823 Ін Б: 2

Рисунок 5.8 – Форма «Контингент»

Студент

Код студента: 479824

Прізвище: Абакуменко

Ім'я: Єва

По батькові: Олександрівна

Іноземець: ☐

Стать: Жіноча

Тип фінансування: К

Історія студента

Наказ	Дата	Пункт	Причина	Ниж
212-С	06.07.2020	Переведення групи	інші випадки	на наступний курс
000-5	06.10.2019	Перенесено із бази	10.06.2020	

Шифр групи: МТЕ-127

Форма навчання: Денна

Курс: 4

Спеціальність: 242

Туркизм

Освітня програма: туризмознавство

[Moodle](#) [Заява](#) [Зміна ПІБ](#) [Вся історія однієї людини](#)

Рисунок 5.9 – Форма даних студента

При натисканні на кнопку «Накази» відкривається форма для введення, відпрацювання та перегляду наказів про рух студентів (рис 5.10). Для зручності опису алгоритму роботи, ця форма була поділена на зони: червону (а), зелену (б), блакитну (в), оранжеву (г), фіолетову (д).

У червоній області у розділі «Наказ» у полі «Шифр» потрібно обрати бажаний наказ або у розділі «Новий наказ» ввести у поля «Шифр» та «Дата» відповідно шифр і дату наказу і натиснути кнопку «Зберегти». Після цього знову ж таки обрати у розділі «Наказ» новостворений наказ.

У зеленій області після вибору наказу виводяться відповідні пункти. Для відкриття змісту пункту потрібно двічі натиснути на потрібний рядок. Після цього дані пункту виводяться у блакитну область, а дані про студентів, включених у цей пункт у оранжеву область. Для створення нового пункту потрібно натиснути на кнопку «Новий пункт» і заповнити поля пункту.

Для внесення пункту наказу потрібно послідовно виконати один із можливих сценарії із введенням нових наказу та пункту або вже наявних у БД. Після у фіолетовій області потрібно обрати студентів, яких потрібно включити до наказу і натиснути на кнопку «Зберегти». У випадку помилкового включення студента до наказу для видалення потрібно обрати наказ та пункт і в оранжеві області відмітити студентів, яких потрібно виключити. Після натиснути на кнопку «Видалити» і підтвердити видалення.

При натисканні на кнопку «Протоколи» відкривається форма для роботи з протоколами (рис. 5.11). Форма дозволяє відфільтровувати студентів за наданими фільтрами і призначена для введення протоколів на випуск. Також форма дозволяє отримати списки студентів у файлі Excel після введення дати початку і закінчення відбору та запуску алгоритму виведення кнопкою «Вивести». Після завершення виведення протоколів у папці із стартовим файлом системи буде знаходитися документ Excel.

Протоколи

Фільтри

Факультет

Спеціальність

Освітня програма

Форма навчання
☒ Денна
☐ Заочна

Старт

Фініш

Курс

Вивести

Прізвище	Ім'я	По батькові	Група	Протокол	Дата
Абакуменко	Єва	Олегівна	МТЕ-127	0	
Абакумова	Катерина	Ігорівна	МТЕз-210м	0	
Абакумова	Оксана	Василівна	РТ-419сп	0	
Абальмас	Антон	Олегович	КНТ-129	0	
Аббасова	Ульвія Аділ кизи		СНз-110ін	0	
Абдулла Ходжі	Беназір		УФКС-217ін	0	
Абдуллаєв	Магаррам		Ю 116	1	01.06.2020
Абдуллаєв	Мамед	Самандар огли	КНТ-117	0	
Абдуллаєв	Раміль	Латіфович	Т-319м	0	
Абдурахімова	Карина	Гаффоржонівна	Т-310м	0	
Абдурахімова	Карина	Гаффоржонівна	Тз 827сп	2	02.06.2020
Абібулаєва	Катерина	Артурівна	КНТ-210	0	
Абілов	Зейнал		УФКС-119	0	
Абілов	Нахід	Заур Огли	Ю-319м	0	
Аблякімов	Сеттар	Наріманович	КНТ-118	0	
Аболимов	Арсен	Олегович	ІФ-510сп	0	
Абрамов	Валентин	Андрійович	УФКС-126(ак)	0	
Абрамов	Владислав	Юрійович	Ю-119сп	0	
Абрамов	Максим	Камович	ФЕУ-110	0	
Абрамов	Микола	Олександрович	УФКС 116	1	02.06.2020
Абрамова	Валерія	Віталіївна	МТЕ 146	6	06.06.2020
Абрамова	Юлія	Олександрівна	МТЕ 116	1	01.06.2020
Абрамчев	Ілля	Кирилович	М-819	0	
Абрамян	Тимур	Анатолійович	МТЕ-219сп	0	
Абраров	Олександр	Сергійович	Юз-310м	0	
Абросімова	Валерія	Сергіївна	Е-119м	0	

Рисунок 5.11 – Форма «Протоколи»

На даній вкладці знаходяться ще менш значимі, але потрібні елементи, які дозволяють ввести нового студента, обрати існуючого та відкрити форму з інформацією про студента, форму його профіля moodle. Кнопка «Історія імен»

дозволяє відкрити форму для перегляду та пошуку всіх змін прізвищ, імен та по батькові, збережених у базі (рис. 5.12).

Історія імен								
Код	Прізвище	Ім'я	По батькові	Старе	Нове ім'я	Частина імені	Наказ	Дата
482012	Гнедаш	Єлизавета	Юріївна	Терьохіна	Гнедаш	Прізвище	481-С	23.11.2020
487899	Копоть	Олена	Ігорівна	Левенкова	Копоть	Прізвище	481-С	23.11.2020
482364	Гнатієнко	Лілія	Василівна	Мануйлова	Гнатієнко	Прізвище	470-С	09.11.2020
482394	Колесник	Катерина	Сергіївна	Клименко	Колесник	Прізвище	470-С	09.11.2020
479996	Проніна	Карина	Юріївна	Боровик	Проніна	Прізвище	470-С	09.11.2020
478445	Беспавлова	Ольга	Олександрівна	Галан	Беспавлова	Прізвище	470-С	09.11.2020
477008	Садова	Наталія	Едуардівна	Рудченко	Садова	Прізвище	450-С	23.10.2020
484350	Удод	Таїсія	Сергіївна	Сичова	Удод	Прізвище	445-С	19.10.2020
485871	Чернявський	Станіслав	Сергійович	Горбик	Чернявський	Прізвище	445-С	19.10.2020
481833	Строй	Артем	Андрійович	Запара	Строй	Прізвище	438-С	13.10.2020
479640	Полявка	Мар'яна	Сергіївна	Варфоломеєва	Полявка	Прізвище	431-С	13.10.2020
478676	Алімова	Анна	Анатоліївна	Скопик	Алімова	Прізвище	431-С	13.10.2020
482266	Романюк	Вікторія	Віталіївна	Мартиненко	Романюк	Прізвище	431-С	13.10.2020
482203	Бондаренко	Ксенія	Олегівна	Кошарна	Бондаренко	Прізвище	431-С	13.10.2020
482745	Стовбур	Анна	Станіславівна	Нужненко	Стовбур	Прізвище	438-С	13.10.2020
483327	Демченко	Надія	Павлівна	Мороз	Демченко	Прізвище	410-С	01.10.2020
483340	Гранкіна	Марія	Геннадіївна	Вакальчук	Гранкіна	Прізвище	410-С	01.10.2020
486524	Баранова	Анастасія	Володимирівна	Швайка	Баранова	Прізвище	410-С	01.10.2020
482488	Губіна	Яна	Леонідівна	Хінчик	Губін	Прізвище	360-С	23.09.2020
484239	Ходаніч	Єлизавета	Степанівна	Тонієвич	Ходаніч	Прізвище	359-С	23.09.2020
482542	Шаповал	Міла	Петрівна	Людмила	Міла	Ім'я	360-С	23.09.2020
476816	Таран	Анастасія	Сергіївна	Писанко	Таран	Прізвище	355-С	21.09.2020
479583	Куватова	Олена	Василівна	Єгорова	Куватова	Прізвище	346-С	16.09.2020
484196	Квасова	Аліна	Андріївна	Костюченко	Квасова	Прізвище	337-С	14.09.2020
484397	Барвінок	Катерина	Сергіївна	Кістрінь	Барвінок	Прізвище	329-С	09.09.2020
479983	Мінза	Ольга	Сергіївна	Павлюк	Мінза	Прізвище	329-С	09.09.2020
479786	Закревська	Ангеліна	Вікторівна	Клименко	Закревська	Прізвище	329-С	09.09.2020
483498	Корнієць	Крістіна	Сергіївна	Дивуляк	Корнієць	Прізвище	329-С	09.09.2020
484394	Барвінок	Ілля	Олександрович	Гамалєєв	Барвінок	Прізвище	329-С	09.09.2020
479787	Медвідь	Катерина	Олексіївна	Клюєва	Медвідь	Прізвище	329-С	09.09.2020
487459	Бабанська	Олександра	Сергіївна	Ципко	Бабанська	Прізвище	319-С	07.09.2020

Рисунок 5.12 – Форма історії імен

5.2.2 Вкладка «Форми ФКССК»

На даній вкладці надаються кнопки для відкриття форм:

– «ФКСС» – форма для зв'язування факультетів, кафедр, спеціальностей освітніх програм (рис. 5.13);

- «Ліцензія» – форма для роботи із місцями ліцензованого обсягу (5.14);
- «Вакансія» – форма для роботи із місця державного замовлення;
- «Кваліфікації» – форма для роботи із кваліфікаціями.

При подвійному натисканні на заповнений кортеж у таблиці факультетів/кафедр/спеціальностей/освітніх програм інші таблиці відфільтровуються за обраними параметрами, а дані обраного кортежу зносяться у праву частину. Для редагування будь-якого елементу його потрібно знести у праву частину ввести зміни і натиснути на кнопку збереження (із зображенням галочки). Для внесення нового елементу потрібно натиснути на кнопку нового запису, заповнити всі поля і натиснути на кнопку збереження (із зображенням галочки). Для видалення потрібно знести елемент у праву частину і натиснути на кнопку видалення (із зображенням корзини). Для утворення нової зв'язаної компоненти факультет\кафедра\спеціальність\освітня програма потрібно знести всі чотири елементи у праву частину і натиснути «Зв'язати».

Рисунок 5.13 – Форма «ФКСС»

Форма «Ліцензія» при виборі коду спеціальності дозволяє переглядати та редагувати місця ліцензованого обсягу. На формі подається ліцензія за формою навчання та курсом на перехресті відповідних параметрів у рядках «Ліцензія ДН» та «Ліцензія ЗН» (для денної та заочної форм навчання відповідно), а в рядках факту подається фактична кількість студентів, які займають певну кількість місць ліцензованого обсягу, та студенти, які зайняли місце ліцензованого обсягу, але ще не вступили на навчання (після символу «+»).

	1 курс	2 курс	3 курс	4 курс	1м курс	2м курс
Ліцензія ДН	35	25	25	30	3	3
Факт ДН	32+0	16+0	22+0	18+0	3+0	3+0
Ліцензія ЗН	25	35	35	30	3	3
Факт ЗН	12+0	2+0	6+0	27+0	3+0	3+0

Зберегти

Рисунок 5.14 – Форма «Ліцензія»

Форма «Вакансія» (рис. 5.15) дозволяє працювати з місцями державного замовлення аналогічно як і з місця ліцензованого обсягу. Форма надає можливість відфільтровувати держзамовлення у верхній частині форми, виконувати пошук вакантних місць, спираючись на вказані параметри і виконувати перерозподіл місць держзамовлення. Так як бюджетні місця потребують особливо жорсткої уваги, будь-які зміни при перерозподілу місць зберігаються в історії. У кожному кортежу кожної освітньої програми наявна кнопка «Заява» для відображення форми із списком заяви, поданих на відповідну програму. На формі білі поля це місця державного замовлення,

червоні поля – студенти, які по факту навчаються на відповідній освітній програмі, формі навчання та курсі, зелені поля – вакантні місця (різниця держзамовлення та фактичної кількості студентів).

Вакансії

Фільтри

Факультет:

Спеціальність:

Освітня програма:

Вакансії

Форма навчання:

Курс:

Освітній ступінь:

Фільтрувати

Перерозподіл вакантних місць

Звідки:

Куди:

Курс:

Форма навчання:

Освітній ступінь:

Рів:

Кількість:

Перерозподілити

Історія

172 Освітня програма телемедичні та біомедичні системи Заяви

Денна Зачісна

	1 курс		2 курс		3 курс			4 курс				1м курс	2м курс
	ЗО 20	МС 20	ЗО 19	МС 19	ЗО 18	МС 18	МС 19	ЗО 17	МС 17	МС 18	МС 19	Mar 20	Mar 19
Державне замовлення	0	0	0	0	0	0	0	0	0	0	0	0	0
Факт	0	0	0	0	0	0	0	0	0	0	0	1	9
Вакансії	0	0	0	0	0	0	0	0	0	0	0	-1	-9

131 Освітня програма обладнання та технології ливарного виробництва Заяви

Денна Зачісна

	1 курс		2 курс		3 курс			4 курс				1м курс	2м курс
	ЗО 20	МС 20	ЗО 19	МС 19	ЗО 18	МС 18	МС 19	ЗО 17	МС 17	МС 18	МС 19	Mar 20	Mar 19
Державне замовлення	0	0	0	0	0	0	0	0	0	0	0	0	0
Факт	4	14	6	0	7	0	14	7	1	15	0	4	18
Вакансії	-4	-14	-6	0	-7	0	-14	-7	-1	-15	0	-4	-18

Всього:

	Денна												
Денна	0	0	0	0	0	0	0	0	0	0	0	0	0
	340	547	226	0	294	1	687	337	25	554	10	579	673
	-340	-547	-226	0	-294	-1	-687	-337	-25	-554	-10	-579	-673
Зачісна	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	21	30	0	34	0	45	52	1	40	0	21	55
	-10	-21	-30	0	-34	0	-45	-52	-1	-40	0	-21	-55

Рисунок 5.15 – Форма «Вакансія»

Форма кваліфікацій (рис. 5.16) дозволяє зчитувати, вносити, редагувати, зв'язувати кваліфікації з освітніми програмами. Для зчитування кваліфікацій з файлу у правому верхньому куті є кнопка «Читати», після натискання якої з'являється діалогове вікно для вказівки потрібного файлу для зчитування. Зчитування виконується менш ніж за хвилину. Для зв'язування або зняття прив'язки потрібно обрати освітню програму у полі і кваліфікації, позначивши голочкою потрібні у відповідному рядку. Після цього, натиснути потрібну кнопку для відповідної операції. Для введення нової кваліфікації

потрібно натиснути на кнопку «Нова кваліфікація», заповнити потрібні поля і зберегти. Після цього можна виконувати прив'язку до освітньої програми.

Кваліфікації

Освітня програма

Прив'язати

Відв'язати

Нова кваліфікація

Тип

☒Прийом

☐Випуск

Зчитування всіх кваліфікацій

Читати

Бач/Мат	Код	Прийом/Випуск (укр)	Прийом/Випуск (анг)	
☐	Маг	2419.2	Професіонал у сфері маркетингу, ефективності господарської діяльності	Marketing, Performance, Streamlining, Intellectual Property and Innovation Professional
☐	Маг	2447.2	Професіонал з управління проектами та програмами	Professional in Project and Program Management
☐	Маг	2413.2	Професіонал у сфері біржової діяльності	Stock Exchange Professional
☐	Маг	2419.2	Фахівець з публічних закупівель	Public Procurement Specialist
☐	Маг	2419.2	Фахівець-аналітик з дослідження товарного ринку	Commodity Market Analyst
☐	Маг	242	Професіонал в галузі правознавства, прокурорського нагляду, правосуддя	Professional in Law, Public Prosecution, Justice and Law Enforcement
☐	Маг	2149.2	Професіонал із організації інформаційної безпеки	Information Security Professional
☐	Маг	2141	Професіонал в галузі архітектури та планування міст	Architecture and Urban Planning Professional
☐	Маг	2141.2	Архітектор та планувальник міст	Architect and Urban Planner
☐	Маг	2141.2	Архітектор	Architect
☐	Маг	2141.2	Архітектор з реставрації пам'яток архітектури та містобудування	Monuments Restoration and Urban Development Architect
☐	Маг	2446	Професіонал в галузі соціального захисту населення	Social Welfare Professional
☐	Маг	2340	Викладач із соціальної педагогіки	Social Pedagogy Teacher
☐	Маг	2419	Професіонал у сфері державної служби, маркетингу, ефективності росту	Public Service, Marketing, Performance, Streamlining, Intellectual Property and Innovation Professional
☐	Маг	2419.3	Професіонал державної служби	Public Service Professional
☐	Маг	2482	Професіонал в галузі готельної та ресторанної справи	Hotel and Restaurant Services Professional
☐	Маг	2481	Професіонал в галузі туризму	Tourism Professional
☐	Маг	2419.2	Експерт із зовнішньоекономічних питань	International Economic Policy Expert
☐	Бач	3228	Логопед	Speech therapist
☐	Бач	3330	Асистент вчителя з корекційної освіти	Assistant to remedial teacher
☐	Бач	3340	Вихователь	Educator
☐	Бач	3475	Тренер-викладач з виду спорту	Coach (sport)
☐	Бач	3475	Суддя з виду спорту	Sports referee
☐	Бач	3471	Дизайнер-виконавець промислових виробів та об'єктів	Industrial designer and fabricator
☐	Бач	3471	Дизайнер-виконавець інтер'єру	Interior designer and fabricator
☐	Бач	3471	Дизайнер-виконавець графічних робіт	Graphic designer and fabricator
☐	Бач	3330	Фахівець в галузі спеціалізованої (особливої) освіти	Special education specialist

Рисунок 5.16 – Форма «Кваліфікації»

5.3.2 Вкладка «Звіти»

Для обробки інформації в БД та зручного її перегляду було розроблено велику кількість звітів у середовищі розробки Microsoft Office 2019, а також розроблено багато функціоналу за допомогою мови програмування VBA для заповнення стандартних звітів у вигляді Excel – таблиць.

На вкладці «Звіти» вказаний поточний місяць, з яким пов'язана програмна частина. Однак при зміні місяцю до програмної частини приєднується таблична частина за вказаний місяць, якщо така існує у репозитарії табличних частин. Робота із звітами зведена до натискання на одну

відповідну кнопку для кожного типу звіту. Після натискання кнопки потрібно очікувати від 30 до 360 секунд в залежності від кількості даних та загальної завантаженості робочої станції.

Для генерації звітів підготовлені відповідні файли-шаблони, які включають у собі частину стандартної інформації кожного звіту (заголовки, макет) і програму встановлення стилів. Реалізація генерації звітів виконується за подібними алгоритмамию.

Крок 1. Обирається потрібний місяць і натискається кнопку відповідного звіту.

Крок 2. На основі копії шаблону створюється звіт, який буде заповнюватися, і відкривається для редагування на першій сторінці (переважно існують звіти окремо для денної і заочної форм навчання);

Крок 3. Виконується витягування потрібних даних і перенесення відповідно до вимог звіту. Пересування по файлу Excel по рядках виконується за допомогою ітератора, а по стовпцях за допомогою масиву із перебором імен потрібних стовпців;

Крок 4. Запускається функція файлі Excel встановлення стилів, якщо така є;

Крок 5. Змінюється сторінка і алгоритм повторюється до останньої сторінки. Після завершення генерації звіту файл зберігається і закривається.

5.3.3 Вкладка «Витягування»

Дана вкладка стала певною надбудовою для формування окремих файлів за спеціальним запитом на базі даних контингенту.

5.3.4 Вкладка «ЄДЕБО»

Взаємодія з ЄДЕБО найчастіше виконується за двома причинами: для порівняння контингенту та для порівняння дипломників.

Для порівняння контингенту потрібно виконати таку послідовність дій:

- зчитати контингент з ЄДЕБО, натиснувши на кнопку «Зчитати» у групі «Контингент»;
- виконати зв'язування контингенту бази із ЄДЕБО, натиснувши кнопку «Зв'язати»;
- виконати перевірку знайдених невідповідностей даних. Невідповідність може виникнути як через розбіжності у виконанні наказів, так через різницю навіть однієї букви, наприклад, у прізвищі;
- переглянути всі розбіжності та розв'язати всі проблеми.

Для порівняння даних про випуск виконується аналогічний алгоритм, але у розділі «Дипломники». Обравши освітній ступінь можна швидко згенерувати накази на випуск, журнали видачі дипломів, статистичні звіти тощо.

5.4 Тестування, введення в експлуатацію

При створенні програмної частини системи паралельно проводилися тестування системи. Реальне використання системи почалося у травні поточного року, паралельно використовувалась попередня система для порівняння результатів роботи і пошуку помилок. Основні вимоги, поставлені під час проведення аналізу вимог до системи виконувалися, але через короткий час, з'явилися нові вимоги або модифікація існуючих. Після внесення всіх необхідних правок, система була реалізована у повній мірі та в робочому стані без проблем оновлювалась і доопрацьовувалася за бажанням користувача.

Процес створення звітів, який виконувався раніше вручну і займав чимало часу, тепер майже повністю автоматичний і займає у середньому 2 хвилини 30 секунд: генерація звіту «Рух контингенту» потребує 41 секунди, звіту «Ліцензія» – 33 секунди, «Вакансії» – 38 секунди, «Відомості про рух контингенту» - від 10 до 60 секунд (залежно від кількості наказів за звітний період та апаратних можливостей персонального комп'ютера), «Ставки» - 15 секунд.

При цьому розроблена ІКТ не потребує великої кількості апаратних ресурсів, під час генерації звітів використовується 24 000 – 28 000 Kbyte оперативної пам'яті, загалом ІКТ займає 8 632 Kbyte (дані займають 3 828 Kbyte, програмний інтерфейс – 4 804 Kbyte).

Створений графічний інтерфейс дозволив працювати із потрібними даними у зручному для користувача способі, а саме: відсутність прямого контакту до таблиць з даними, відсутність необхідності працювати із схемою БД та ключовими поля таблиць. Однак у той же час структура БД дозволяє на інтуїтивно зрозуміло рівні працювати з масивами даних, виконувати пошук, відбір і витягувати з ІС дані у потрібно вигляді декількома кліками.

Створена методологія роботи із вакансіями державного замовлення і ліцензійним обсягом у реальних умовах доказала свою дієвість і практичність використання.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

6.1 Вступ

Надається економічне обґрунтування програмної реалізації системи управління та контролю контингентом, оцінка затрат на її створення, розрахунок економічної ефективності програмного продукту.

Система, що розробляється, дозволить фіксувати, обробляти та переглядати інформацію про місце навчання студентів та його історію у межах університету та створювати різноманітні звіти.

На даний момент часу аналогів програмного забезпечення, які б задовольняли всі вимоги, не існує.

Система розроблялася виключно як безкоштовна для університету.

6.2 Планування розробки

Весь процес розробки розділяється на етапи. Для кожного етапу вказується трудомісткість, кількість виконавців та тривалість робіт. В розробці беруть участь програміст терміном 3 місяці та консультант 0,75 місяця. Розробка починається 1 вересня і повинна бути виконана до 1 грудня 2020 року включно. Етапи розробки програмного продукту наведені у таблиці 6.1.

Таблиця 6.1 – Етапи розробки

Назва етапу	Трудомісткість		Виконавці	Тривалість, днів
	людино-дні	% до підсумку		
1. Аналіз предметної області	10	12,5	Програміст Консультант	5
2. Визначення вимог та створення структури системи	8	10	Програміст Консультант	4
3. Розробка структурної схеми	8	10	Програміст	8
4. Концептуальне моделювання	12	15	Програміст	12
5. Розробка схеми БД	8	10	Програміст	8

Продовження таблиці 6.1

Назва етапу	Трудомісткість		Виконавці	Тривалість, днів
	людино-дні	% до підсумку		
6. Розробка програмного інтерфейсу та написання коду	24	30	Програміст	24
7. Внесення початкових даних, тестування та налагоджування програмного продукту	4	5	Програміст Консультант	2
8. Складання програмної документації	6	7,5	Програміст Консультант	3
Разом	80	100		66

За даними таблиці 6.1 складено стрічковий графік планування розробки продукту, зображений на рисунку 6.1 (синій колір – програміст, оранжевий – консультант, червоний – неробочі дні).

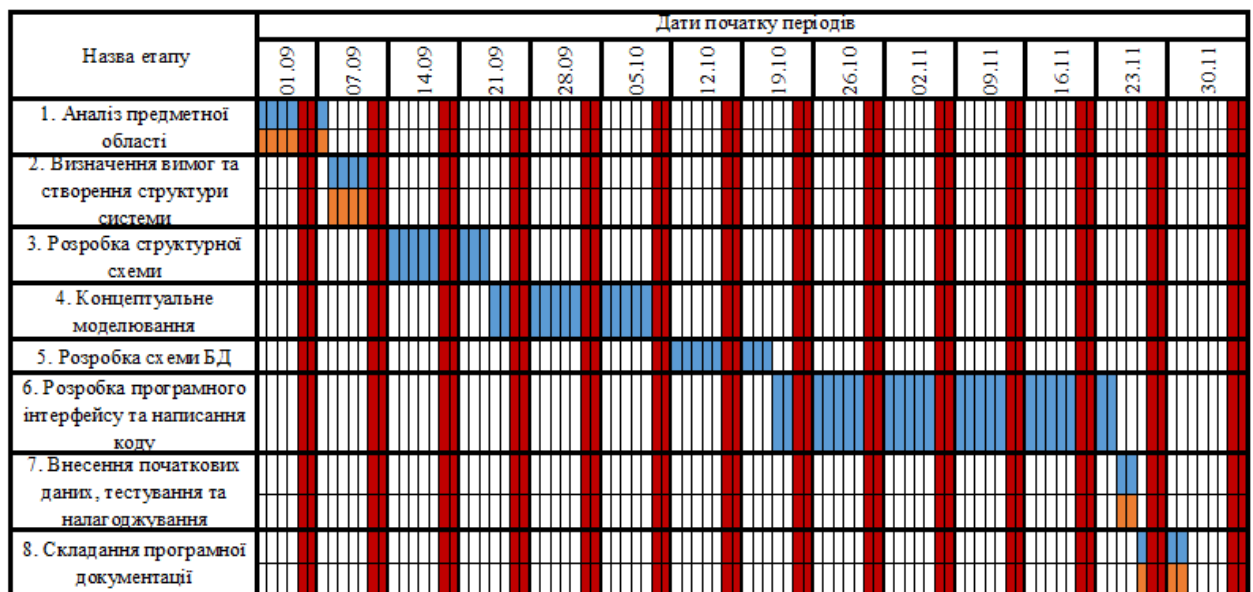


Рисунок 6.1 – Стрічковий графік

6.3 Визначення витрат на розробку

Для визначення витрат на розробку системи контролю контингентом складено калькуляцію кошторисної вартості робіт, яка містить наступні статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок;
- матеріали та комплектуючі вироби;
- витрати на спеціальне обладнання;
- накладні витрати.

Для розрахунку заробітної плати маємо на увазі, що у вересні, жовтні та листопаді 2020 року кількість робочих днів загалом дорівнює 66 дням.

Тривалість робіт визначається за формулою:

$$T_{\text{ц}} = \frac{Q}{R}, \quad (6.1)$$

де $T_{\text{ц}}$ – кількість днів;

Q – трудомісткість;

R – кількість робітників.

6.3.1 Розрахунок основної заробітної плати

Проводиться розрахунок основної заробітної плати, результати якого наведено у таблиці 6.2.

Таблиці 6.2 – Основна заробітна плата

Посада виконання	Чисельність, чол.	Місячний оклад, грн.	Кількість місяців	Сума ЗП, грн.
Програміст	1	8800	3	26400
Консультант	1	9000	0,75	6750
Разом	2			33150

6.3.2 Розрахунок додаткової заробітної плати

Додаткова заробітна плата приймається рівною 10% від основної заробітної плати працівника і розраховується за формулою:

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} * 0.1, \quad (6.2)$$

Після підстановки значення основної заробітної плати у формулу 6.2 отримаємо:

$$ЗП_{\text{дод}} = 33150 * 0,1 = 3315 \text{грн.}$$

6.3.3 Відрахування на єдиний соціальний внесок

Відрахування на єдиний соціальний внесок становлять 22% і беруться від основної та додаткової заробітної плати. Вони розраховуються за формулою 6.3:

$$ВІД = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) * 0,22, \quad (6.3)$$

де $ЗП_{\text{осн}}$ - основна заробітна плата (грн.);

$ЗП_{\text{дод}}$ – додаткова заробітна плата (грн.).

Після підстановки значень основної та додаткової заробітної плати в дану формулу отримаємо:

$$ВІД = (33150 + 3315) * 0,22 = 8022,30 \text{грн.}$$

Додаткові соціальні відрахування становлять 8022,30грн.

6.3.4 Визначення витрат на матеріали

Витрати на матеріали розраховуються за формулою:

$$M = \sum_{i=1}^n (\text{Ц}_i N_i (1 + K_{\text{т.з.}}) - \text{Ц}_{i0} N_{i0}), \quad (6.4)$$

де M – витрати на матеріали, покупні напівфабрикати та комплектуючі (грн.);

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Ц_i – ціна i -го найменування чи комплектуючого (грн.);

N_i – потреба в i -му матеріалі, напівфабрикаті, комплектуючому;

Ц_{i0} – ціна зворотних відходів i -го найменування матеріалу (грн.);

N_{i0} – кількість зворотних відходів i -го найменування;

N – кількість найменувань матеріалів, напівфабрикатів та комплектуючих.

Додатковими матеріалами, необхідними для розробки, є папір для оформлення документації, який коштує 80 грн.

6.3.5 Витрати на обладнання

До цього розділу входять витрати на придбання, транспортування, монтаж, та налагодження нестандартного обладнання. Також необхідно розрахувати витрати на оплату машинного часу ЕОМ (амортизаційні витрати на обслуговування та вартість спожитої електроенергії).

Амортизаційні відрахування на обслуговування ЕОМ розраховуються за формулою:

$$A = \Phi_B \frac{H_A}{100}, \quad (6.5)$$

де Φ_B – балансова вартість обчислювальної техніки (грн.);

N_A - норма амортизаційних відрахувань на повне відновлення обчислювальної техніки (%).

Після підстановки значень балансової вартості обчислювальної техніки у розмірі 10830 грн. в дану формулу, отримуємо:

$$A = 10830 * 0,25 = 2707,50 \text{ грн.}$$

Амортизаційні відрахування становлять 2707,50 грн.

Для розрахування вартості експлуатації обладнання використовується формула:

$$C_E = N_H \Phi_{E\Phi} K_{3ч} K_{3П} C_E, \quad (6.6)$$

де N_H - номінальна потужність ЕОМ, кВт;

$\Phi_{E\Phi}$ - річний ефективний фонд часу роботи ЕОМ;

$K_{3ч}$ - середній коефіцієнт завантаження за часом;

$K_{3П}$ - коефіцієнт завантаження за потужністю;

C_E - ціна одного кВт*год, електроенергії, грн./(кВт*год).

Річний фонд часу роботи ЕОМ становить 1800 годин. Середні коефіцієнти завантаження за часом та за потужністю становлять 0,8 та 0,6:

$$C_E = 0,4 * 1800 * 0,8 * 0,6 * 2,11 = 729,22 \text{ грн.}$$

Заробітна плата обслуговуючого персоналу розраховується за формулою:

$$ЗП_{\text{ОБСЛ}} = \Phi ЗП_{\text{Р}} (1 + K_{\text{ВІД}}) \frac{t_{\text{ОБСЛ}}}{\Phi_{E\Phi \text{ ОБСЛ}}}, \quad (6.7)$$

де $\Phi ЗП_r$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робіт (грн.);

$K_{ВІД}$ – коефіцієнт відрахувань на соціальне страхування і до інших фондів (%);

$t_{ОБСЛ}$ – час протягом року, необхідний на технічне обслуговування ЕОМ (год/рік);

$\Phi_{ЕФ ОБСЛ}$ – річний ефективний фонд часу обслуговування персоналу.

Місячна заробітна плата обслуговуючого персоналу становить 4173 грн., а річний фонд заробітної плати становить 50076 грн. Річний ефективний фонд робочого часу обслуговуючого ПК робітника становить 1738,75 ч/год. На обслуговування одного ПК витрачається по 1 годині на місяць, а на рік становить 12 годин. З чого виходить, що:

$$ЗП_{ОБСЛ} = 50076 * (1 + 0,22) * \frac{12}{1738,75} = 421,63 \text{ грн.}$$

Стаття «Поточний ремонт обладнання» приймається рівною 3% від балансової вартості обладнання складає 324,90 грн.

Стаття «Інші витрати» приймається рівною п'яти відсоткам від суми всіх попередніх статей витрат на утримання та експлуатацію устаткування. Сума всіх попередніх статей дорівнює грн., 5% від суми становлять грн.

Весь кошторис витрат подано у таблиці 6.4.

Таблиця 6.4 – Кошторис витрат на утримання та експлуатацію обладнання

Найменування статей витрат	Сума, грн
Амортизація обладнання	2707,50
Експлуатація обладнання (окрім витрат на поточний ремонт)	729,22
Заробітна плата основна та додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	421,63
Поточний ремонт обладнання	324,90
Інші витрати	209,16
Разом	4392,41

6.4 Економічний ефект та економічна ефективність

Економічний ефект заключається в тому, що розроблений програмний продукт буде повністю належати університету на безкоштовній основі (не рахуючи витрати, описані у даному розділі). Використовуючи свої права можна використовувати розроблену систему на власний розсуд і змінювати за новими вимогами університету, чого у комерційних застосунках часто не дозволяють роботи або не в повній мірі.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

Розроблювана система, яка дозволить працювати з інформацією про контингент, фіксувати її та обробляти, буде знаходитися у невеличкій офісній кімнаті. Розміри приміщення 6.0 x 4.0 x 3.0 м.

Площа приміщення становить $6.0 \times 4.0 = 24\text{м}^2$, об'єм приміщення $6.0 \times 4.0 \times 3.0 = 72\text{ м}^3$

Небезпечні та шкідливі виробничі фактори за природою виникнення поділяються на наступні групи:

- фізичні (підвищення або знижена температура повітря, надмірна запиленість або загазованість повітря та інше);
- хімічні (вміст у робочій зоні шкідливих речовин та інше);
- психофізіологічні (нервово-емоційні перевантаження, розумове напруження та інше);
- біологічні (підвищений вміст у повітрі мікроорганізмів та інше).

У даному розділі розглянуто наступні питання:

- мікроклімат та склад повітря робочої зони;
- виробниче освітлення;
- шум;
- виробничі випромінювання;
- безпека щодо організації робочих місць;
- безпечність технологічного обладнання та процесу;
- електробезпека;
- пожежна безпека.

7.2 Заходи із забезпечення безпеки

Приміщення офісу, в яких перебувають співробітники навчального відділу, відносяться до приміщень без підвищеної небезпеки ураження електричним струмом.

Обладнання, що використовується в цих приміщеннях є споживачем електроенергії, що живиться від змінного струму 220 В від мережі з заземленою нейтраллю, та відноситься до електроустановок до 1000В закритого виконання. За способом захисту людини від ураження електричним струмом відповідає згідно з [13] І (стаціонарні комп'ютери,) та ІІ (освітлювальні прилади, кондиціонери, опалювальні пристрої, ноутбуки, сканери) класу захисту.

Згідно з «Правилам улаштування електроустановок» (далі «ПУЕ») виконані такі групи заходів з електробезпеки:

- конструктивні заходи забезпечують захист від випадкового дотику до струмопровідних частин за допомогою їх ізоляції та захисних оболонок.
- схемно-конструктивні заходи. Призначені для забезпечення захисту від ураження електричним струмом при дотику до металевих оболонок, які можуть опинитися під напругою в результаті аварій.
- організаційні заходи. Експлуатація електроустановок і електроустаткування проводиться відповідно до [14] та [15].

Нервово напруження впливає на серцево-судинну систему, збільшуючи артеріальний тиск і частоту пульсу, а також на терморегуляцію організму та емоційні стани працівника. Особливу роль у запобіганні втомі працівників відіграють професійний відбір, організація робочого місця, правильне робоче положення, ритм роботи, раціоналізація трудового процесу, використання емоційних стимулів, впровадження раціональних режимів праці і відпочинку тощо. Боротьба зі втомою, в першу чергу, зводиться до покращення санітарно-гігієнічних умов виробничого середовища (ліквідація забруднення повітря, шуму, вібрації, нормалізація мікроклімату, раціональне освітлення тощо).

Крім того, для профілактики втоми працівників застосовуються специфічні методи, до яких можна віднести засоби відновлення функціонального стану зорового та опорно-рухового апарату, зменшення гіподинамії, підсилення мозкового кровообігу, оптимізацію розумової діяльності.

Загальні ергономічні вимоги встановлено [16]. Організація робочого місця передбачає: правильне розміщення робочого місця у виробничому приміщенні; вибір ергономічно обґрунтованого робочого положення, виробничих меблів з урахуванням антропометричних характеристик людини; раціональне компонування обладнання на робочих місцях; врахування характеру та особливостей трудової діяльності.

Визначимо скільки комп'ютеризованих робочих місць, оснащених відеодисплейними терміналами (ВДТ) можна встановити у даному приміщенні і як їх розташувати відповідно до встановлених норм та правил з охорони праці. Розміри приміщення: довжина $a = 6,0$ м, ширина $b = 4,0$ м, висота $h = 3,0$ м.

Перш за все необхідно проаналізувати чи підходить дане приміщення для того, щоб розмістити в ньому комп'ютеризовані робочі місця.

Відповідно до [17] є неприпустимим розташування приміщень, призначених для роботи з ВДТ у підвалах та цокольних поверхах. Також забороняється розташування вибухонебезпечних приміщень категорії А і Б ([18]) та виробництв з мокрими технологічними процесами поряд з приміщенням, де розташовуються ЕОМ (ПЕОМ), а також над такими приміщеннями, або під ними. Окрім того, виробничі приміщення для роботи з ВДТ не повинні межувати з приміщеннями, у яких рівень шуму і вібрації перевищує допустимі значення.

Вибране приміщення відповідає вищезазначеним вимогам, тому переходимо до наступного етапу, а саме, до визначення кількості комп'ютеризованих робочих місць, що можна розмістити в даному приміщенні. Оскільки площа приміщення становить $S_{\text{пр}} = 24 \text{ м}^2$, а площа, на якій розташовується одне робоче місце з ВДТ, відповідно до вимог [19]

повинна становити не менше $6,0 \text{ м}^2$, то в даному приміщенні можна розмістити щонайбільше чотири комп'ютеризованих робочих місць.

Перевіримо, чи відповідає це число нормативу щодо мінімального об'єму приміщення на одне робоче місце з ВДТ, відповідно до вимог [19] ($VP.M.Min = 20 \text{ м}^3$). Об'єм приміщення становить 72 м^3 , а об'єм, що припадає на одне комп'ютеризоване робоче місце - $VP.M. = 18 \text{ м}^3$. Таким чином норматив щодо об'єму приміщення на одне робоче місце з ВДТ не виконується. Тому розміщаємо в даному приміщенні щонайбільше три комп'ютеризованих робочих місць.

Планування розміщення комп'ютеризованих робочих місць у приміщенні проводимо із врахуванням наступних вимог:

- робочі місця з ВДТ розміщуються на відстані не менше 1 м від стіни зі світловими прорізами;
- відстань між бічними поверхнями ВДТ має бути не менше за 1,2 м;
- відстань між тильною поверхнею одного ВДТ та екраном іншого не повинна бути меншою за 2,5 м;
- прохід між рядами робочих місць має бути не меншим 1 м.

Необхідно також врахувати розміри меблів на комп'ютеризованих робочих місцях, зокрема робочого столу. Відповідно до [20] рекомендовані розміри столу для робочого місця з ВДТ становлять: висота – 725 мм, ширина – 600-1400 мм, глибина – 800-1000 мм. Приймаємо, що робочий стіл має такі розміри: ширина – 1200 мм, глибина – 800 мм.

Найкраще розмістити комп'ютеризовані робочі місця рядами вздовж стіни з вікнами. Це дасть змогу унеможливити дзеркальне відбиття на екрані ВДТ джерел природного світла (вікон) та потрапляння останніх у поле зору операторів, що погіршує умови їх зорової роботи. На рисунку 7.1 наведено план приміщення з комп'ютеризованими робочими місцями.

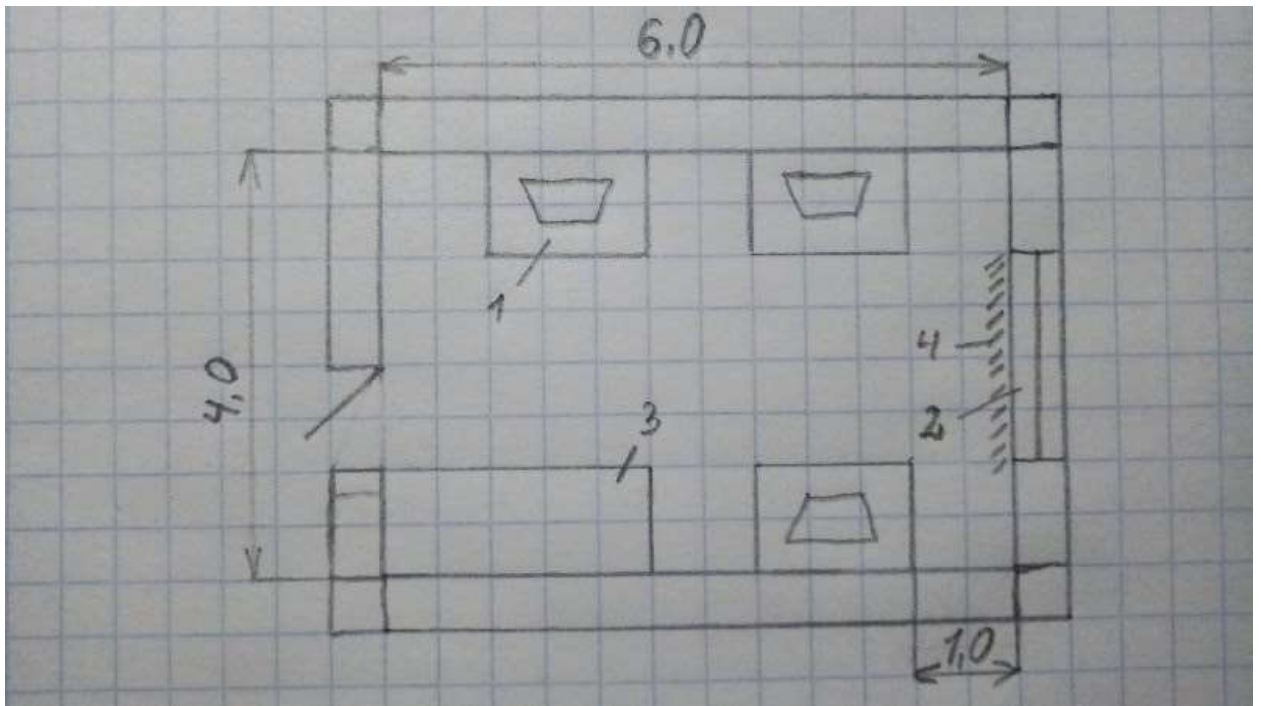


Рисунок 7.1 – План приміщення:

1 – робоче місце; 2 – вікно; 3 – шафи для зберігання документації та інших документів; 4 – сонцезахисні жалюзі.

7.3 Заходи з забезпечення виробничої санітарії та гігієни праці

Згідно [21] в офісних приміщеннях використовується природне та штучне освітлення. Природне освітлення здійснено через світлові прорізи, які забезпечують коефіцієнт природної освітленості (КПО) не нижче 1,5%. Для захисту від прямих сонячних променів, які створюють прямі та відбиті відблиски на поверхні екранів і клавіатури, використовуються сонцезахисні пристрої, на вікнах встановлені жалюзі або штори. Штучне освітлення в приміщенні здійснено системою загального рівномірного освітлення. Значення освітленості на поверхні робочого столу в зоні розміщення документів становить 300-500 лк. Як джерела штучного освітлення в приміщенні застосовуються люмінесцентні лампи типу ЛБ.

Показники мікроклімату в офісних приміщеннях відповідають встановленим санітарно-гігієнічним вимогам [22], [23] і [24].

Роботи в приміщеннях обслуговування туристів, належать до категорії 1б - легка робота, тому встановлені наступні оптимальні значення параметрів мікроклімату:

- у холодний період року: температура 22-24°C; відносна вологість: 40- 60%; швидкість переміщення повітря: 0,1 м/с;
- у теплий період року: температура 23-25°C; відносна вологість: 40-60%; швидкість переміщення повітря: 0,2 м/с.

Оптимальні рівні позитивних (n+) і негативних (n-) іонів у повітрі приміщення з ПК нормовані згідно [24] і становлять: $n^+ = 1500-30000$ (шт. на 1см^3); $n^- = 3000-5000$ (шт. на 1см^3). Підтримка оптимального рівня легких позитивних і негативних аероіонів у повітрі на робочих місцях забезпечено за допомогою біполярних коронних аероіонізаторів.

Дотримання вимог цих документів досягається оснащенням приміщень і транспортних засобів пристроями кондиціювання і вентиляції, дезодорації повітря, опалювання згідно вимог [25].

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях у приміщення нормуються згідно [19] та [26].

Зниження рівня шуму в приміщенні здійснено за допомогою:

- використання більш сучасного обладнання;
- розташування принтерів та різноманітного устаткування колективного користування на значній відстані від більшості робочих місць працівників;
- переведення жорсткого диска в режим сну (Standby), якщо комп'ютер не працює протягом визначеного часу;
- використання блоків живлення ПК з вентиляторами на гумових підвісках.

7.4 Заходи безпеки у надзвичайних ситуаціях

7.4.1 Заходи з пожежної безпеки

Закон України «Про пожежну безпеку» визначає загальні правові, економічні та соціальні основи забезпечення пожежної безпеки на території України, регулює відносини державних органів, юридичних і фізичних осіб у цій галузі незалежно від виду їх діяльності та форм власності.

Пожежна безпека – стан об'єкта, при якому з регламентованою ймовірністю виключається можливість виникнення та розвиток пожежі і впливу на людей її небезпечних факторів, а також забезпечується захист матеріальних цінностей.

Для забезпечення пожежної безпеки в установах проводять пожежну профілактику, яка включає в себе комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожежі, обмеження її поширення, а також на створення умов для успішного гасіння пожежі.

Для ліквідації пожежі у початковій стадії її розвитку силами персоналу об'єктів застосовуються первинні засоби пожежогасіння. До них відносяться: вогнегасники, пожежний інвентар (покривала з негорючого теплоізоляційного полотна, ящики з піском, пожежні відра, совкові лопати, ломи, сокири тощо), системи автоматичного пожежогасіння.

Первинні засоби пожежогасіння, в залежності від категорії приміщень, можуть розташовуватись як окремо, так і в складі пожежних щитів.

Залежно від агрегатного стану й особливостей горіння різних горючих речовин і матеріалів пожежі за [27] поділяються на відповідні класи.

В офісному приміщенні знаходиться дерев'яна мебель, електронна апаратура, бумажні носії інформації.

Клас пожежі у офісному приміщенні (згідно із [27]) – пожежі твердих речовин, переважно органічного походження, горіння яких супроводжується тлінням (деревина, пластмаси, папір) – визначається як клас А. Категорія приміщення (згідно із [28]) – визначається як категорії Д.

Визначення типу та розрахунок кількості первинних засобів пожежогасіння (згідно із [27]) – для адміністративного приміщення площею 24 м² слід застосовувати два порошкових вогнегасниками типу ВП–5 ([28]). Крім цього адміністративні приміщення повинні бути обладнані автоматичними пожежними сповіщувачами, що реагують на підвищення температури, дим, полум'я. Наприклад, сповіщувачі моделей ДТЛ, ІТМ.

7.4.2 Заходи з цивільного захисту

Інформування, оповіщення і дії робітників та службовців промислового об'єкту при загрозі виникнення надзвичайної ситуації.

Головним і невід'ємним елементом всієї системи захисту населення та територій від надзвичайних ситуацій техногенного та природного характеру є інформація та оповіщення.

Зміст інформації має становити відомості про надзвичайні ситуації, що прогнозуються або вже виникли, з визначенням їхньої класифікації, меж поширення і наслідків, а також заходи реагування на них.

Оперативну і достовірну інформацію про стан захисту населення та територій від надзвичайних ситуацій техногенного та природного характеру, методи та способи їх захисту, заходи безпеки зобов'язані надавати населенню через засоби масової інформації, центральні та місцеві органи виконавчої влади та виконавчі органи рад.

Оповіщення про загрозу виникнення надзвичайних ситуацій і постійне інформування про них населення забезпечується шляхом:

- завчасного створення, підтримання в постійній готовності загальнодержавної і територіальних автоматизованих систем централізованого оповіщення населення;
- організаційно-технічного з'єднання територіальних систем централізованого оповіщення і систем оповіщення на об'єктах господарювання;
- завчасного створення та організаційно-технічного з'єднання із системами спостереження і контролю постійно діючих локальних систем оповіщення та інформування населення в зонах можливого катастрофічного затоплення, районах розміщення радіаційних і хімічних підприємств та інших об'єктів підвищеної небезпеки;
- централізованого використання загальнодержавних і галузевих Л^АЦІ зв'язку, радіопровідного, телевізійного оповіщення, радіотрансляційних мереж та інших технічних засобів передавання інформації.

Оповіщення населення про загрозу і виникнення надзвичайної ситуації у мирний, в особливий період та постійне інформування його про наявну обстановку — одне з важливих завдань цивільного захисту України. Для цього створюється система, організаційно-технічне об'єднання засобів для передачі сигналів і розпоряджень органів управління цивільного захисту.

Система оповіщення та інформативного забезпечення створюється завчасно в усіх ланках пунктів управління.

Основу системи оповіщення утворюють автоматизована система централізованого оповіщення мережі зв'язку та радіомовлення, а також спеціальні засоби.

Автоматизована система оповіщення створюється завчасно на базі загальнодержавної мережі зв'язку та радіомовлення і поділяється на державну та територіальну. Вона може забезпечити оповіщення населення, поєднавши місцеву телефонну мережу для подачі сигналу «Увага всім!» та повну інформацію за допомогою засобів радіомовлення й телебачення.

Оповіщення підпорядкованих штабів, органів управління цивільного захисту і населення організовується та забезпечується старшими органами управління.

Додатково в Україні створюється локальна державна система оповіщення про загрозу катастрофічного затоплення в разі руйнування гідроспоруд на р. Дніпро.

Відповідальним за організацію оповіщення про загрозу і виникнення надзвичайної ситуації та постійне інформування населення про становище є органи управління цивільного захисту відповідної адміністративно-територіальної одиниці.

Сигнали передаються каналами зв'язку, радіотрансляційними мережами і засобами телебачення. Одночасно з інформацією про надзвичайну ситуацію передаються вказівки про порядок дій формувань цивільного захисту і населення. Сигнали, подані вищим органом управління цивільного захисту, мають дублюватися всіма підпорядкованими підсистемами. Дублювати сигнали на об'єктах та в населених пунктах можна за допомогою місцевого радіотрансляційного вузла, гудків підприємств, сирен транспорту, ударами в рейку, дзвони.

Для своєчасного попередження населення введені сигнали попередження населення у мирний і воєнний час.

Сигнал «Увага всім!» повідомляє населення про надзвичайну обстановку в мирний час і на випадок загрози нападу противника у воєнний час. Сигнал подається органами цивільного захисту за допомогою сирени і виробничих гудків. Тривалі гудки означають попереджувальний сигнал.

Почувши їх, необхідно включити радіо, телевізор і прослухати текст інформації про дії населення після одержання сигналу. Якщо немає радіо, телевізора або вони не працюють, слід з'ясувати значення і зміст інформації у сусідів або інших людей, які знають про неї.

Після одержання інформації необхідно виконати всі вказівки тексту інформації сигналу.

Сигнали і варіанти оповіщення населення в мирний час такі:

«Аварія на атомній електростанції». Повідомляються місце, час, масштаби аварії, інформація про радіаційну обстановку та дії населення. Якщо є загроза забруднення радіоактивними речовинами, необхідно провести герметизацію житлових, виробничих і складських приміщень. Провести заходи захисту від радіоактивних речовин сільськогосподарських тварин, кормів, урожаю, продуктів харчування та води. Прийняти йодні препарати. Надалі діяти відповідно до вказівок штабу органів цивільного захисту.

«Аварія на хімічно небезпечному об'єкті». Повідомляються місце, час, масштаби аварії, інформація про можливе хімічне зараження території, напрямки та швидкість можливого руху зараженого повітря, райони, яким загрожує небезпека. Дається інформація про поведінку населення. Залежно від обставин: залишатися на місці, у закритих житлових приміщеннях, на робочих місцях чи залишати їх і застосувати засоби індивідуального захисту, вирушити на місця збору для евакуації або в захисні споруди. Надалі діяти відповідно до вказівок штабу органів управління цивільного захисту.

«Землетрус». Подається повідомлення про загрозу землетрусу або його початок. Населення попереджається про необхідність відключити газ, воду, електроенергію, погасити вогонь у печах, повідомити сусідів про одержану інформацію, взяти необхідний одяг, документи, продукти

харчування, вийти на вулицю і розміститися на відкритій місцевості на безпечній відстані від будинків, споруд, ліній електропередачі.

«Затоплення». Повідомляється район, в якому очікується затоплення в результаті підйому рівня води на річці чи аварії дамби.

Населення, яке проживає в даному районі, повинне взяти необхідні речі, документи, продукти харчування, воду, виключити електроенергію, відключити газ і зібратись у вказаному місці для евакуації. Повідомити сусідів про стихійне лихо і надалі слухати інформацію штабу органів управління цивільного захисту.

«Штормове попередження». Подається інформація для населення про посилення вітру. Населенню необхідно зачинити вікна, двері. Закрити в приміщеннях сільськогосподарських тварин. Повідомити сусідів. Населенню, по можливості, перейти в підвали, потреби.

Сигнали оповіщення населення у воєнний час такі.

Сигнал «Повітряна тривога» подається для всього населення. Попереджається про небезпеку ураження противником даного району. По радіо передається текст: «Увага! Увага! Повітряна тривога! Повітряна тривога!» Одночасно сигнал дублюється сиренами, гудками підприємств і транспорту. Тривалість сигналу 2-3 хв.

При цьому сигналі об'єкти припиняють роботу, транспорт зупиняється, і все населення укривається в захисних спорудах. Робітники і службовці припиняють роботу відповідно до інструкції і вказівок адміністрації. Там, де неможливо через технологічний процес або через вимоги безпеки зупинити виробництво, залишаються чергові, для яких мають бути захист споруди.

Сигнал може застати у будь-якому місці й будь-який час. В усіх випадках необхідно діяти швидко, але спокійно, впевнено, без паніки.

Суворо дотримуватися правил поведінки, вказівок органів цивільного захисту.

Сигнал «Відбій повітряної тривоги». Органами цивільного захисту через радіотрансляційну мережу передається текст: «Увага! Увага!

Громадяни! Відбій повітряної тривоги!». За цим сигналом населення залишає захисні споруди і повертається на свої робочі місця і в житла.

Сигнал «Радіаційна небезпека» подається в населених пунктах і в районах, в напрямку яких рухається радіоактивна хмара, що утворилася від вибуху ядерного боєприпаси.

Почувши цей сигнал, необхідно з індивідуальної аптечки АІ-2 прийняти 6 таблеток радіозахисного препарату № 1 із гнізда 4, надіти респіратор, протипилову пов'язку, ватно-марлеву маску або протигаз, взяти запас продуктів, документи, медикаменти, предмети першої потреби і направитися у сховище або ПРУ.

Сигнал «Хімічна тривога» подається у разі загрози або безпосереднього виявлення хімічного або бактеріологічного нападу (зараження). При цьому сигналі необхідно прийняти з індивідуальної аптечки АІ-2 одну таблетку препарату при отруєнні фосфорорганічними речовинами з гнізда 2 або 5 таблеток протибактеріального препарату №1 із гнізда 5, швидко надіти протигаз, а за необхідності - і засоби захисту шкіри, якщо можливо, та укритися в захисних спорудах. Якщо таких поблизу немає, то від ураження аерозолями отруйних речовин і бактеріальних засобів можна сховатися в житлових чи виробничих приміщеннях.

При застосуванні противником біологічної зброї населенню буде подана інформація про наступні дії.

Успіх захисту населення залежатиме від дисциплінованості, своєчасної і правильної поведінки, суворого дотримання рекомендацій і вимог органів цивільного захисту [29].

ВИСНОВКИ

У ході виконання випускної кваліфікаційної роботи магістра був виконаний аналіз предметної області, існуючої системи та наявних рішень. На базі даного аналізу було поставлено завдання і визначено напрямки та призначення дослідження.

На базі проведеного аналізу та отриманих вимог до системи було зроблено вибір середовища розробки, розроблено архітектуру та концептуальну модель системи.

Після розробки концепції системи було розроблено фізичну схему БД, графічний інтерфейс та програмний функціонал для роботи із інформацією про контингент студентів, виконано завантаження даних із попередньої системи та проведено повне тестування системи щодо реалізації всіх поставлених вимог у другому розділі.

Згідно з вимогами до дипломної роботи магістра були розроблені розділи економіко-організаційний і охорони праці та безпеки в надзвичайних ситуаціях.

Наукова новизна полягає у тому, що під час дослідження та програмної реалізації системи фіксування та контролю інформації про навчання студентів були розроблені методи зберігання та обробки місць державного замовлення та місць ліцензованого обсягу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Закон України від 01.07.2014 р. № 1556-VII «Про вищу освіту». <https://zakon.rada.gov.ua/laws/show/1556-18>
2. Про ЄДЕБО [Електрон. ресурс] - Режим доступу: <https://info.edbo.gov.ua/about/>, last accessed 2020/01/29
3. Salesforce for Education Reviews & Product Details [Electronic resource] — Access mode: — <https://www.g2.com/products/salesforce-for-education/reviews>
4. Oracle Higher Education Cloud Reviews & Product Details [Electronic resource] — Access mode: <https://www.g2.com/products/oracle-higher-education-cloud/reviews>
5. New Realities for Higher Education [Electronic resource] — Access mode: <https://www.oracle.com/a/ocom/docs/oct-pubsec-higher-ed-ebook.pdf>
6. Workday Student Reviews & Product Details [Electronic resource] — Access mode: <https://www.g2.com/products/workday-student/reviews>
7. Програмний комплекс «Автоматизована система управління навчальним закладом» Національний юридичний університет імені Ярослава Мудрого. <http://acs.nlu.edu.ua>
8. Програмний комплекс «Автоматизована система управління навчальним закладом». <http://mkr.org.ua/aboutprojects/index/2/>
9. Иванченко, Д.А. Оптимизация построения информационной системы управления вузом: концептуальные подходы // Университетское управление: практика и анализ. 2011. № 2. С. 40–48.
10. Керносов М. А. Гнучка множинна модель навчального плану в підсистемі планування і контролю навчального процесу ВУЗа / М. А. Керносов, А. В. Міхнова, Д. А. Імшенецький // Радіоелектроніка, інформатика, управління. – 2013. – №1. – С. 66–71.

11. Петросян Л. Є. Автоматизація процесу обліку руху контингенту студентів в вишах та філіалах // Современная техника и технологии. 2012. № 4.
12. Вербa, О. В., Зайцева, Т. А., Марченко, О. О.: Автоматизована система обліку руху контингенту здобувачів вищої освіти Дніпровського національного університету імені Олеся Гончара. ISSN 2312-119X. Актуальні проблеми автоматизації та інформаційних технологій. Том 22. 2018, С. 38 – 48.
13. «ССБТ. Изделия электротехнические. Общие требования безопасности» : ГОСТ 12.2.007.0-75. – [Действующий от 1978-01-01]. – М.: Государственный комитет СРСР, 1975. – 12 с.
14. «Правила безпечної експлуатації електроустановок» (далі «ПБЕЕ») : НПАОП 40.1-1.01-97. – [Чинний від 1998-03-01]. – К.: Державний стандарт, 1994, - 237 с.
15. «Правила безпечної експлуатації електроустановок споживачів» (далі «ПБЕЕС») : НПАОП 40.1-1.21-98. – [Чинний від 1998-09-01]. – К.: Державний стандарт України, 1998, - 394
16. «Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 1. Загальні положення» : ДСТУ ISO 9241-1:2003. – [Чинний від 2003-07-21]. – К.: Держстандарт України, 2003, - 326 с.
17. «Правила охорони праці під час експлуатації електронно-обчислювальних машин» : НПАОП 0.00-1.28-10. - [Чинний від 2010-03-26]. – К.: Державний стандарт України, 2010, - 157 с.
18. «Нормы определения категорий помещений, зданий и наружных установок по взрывопожарной и пожарной безопасности» : НАПБ Б.03.002-2007. – [Действующий от 2007-12-03]. – Государственный стандарт Украины, 2007, - 129 с.
19. «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» : ДСанПіН 3.3.2.007-98. – [Чинний від 1998-12-10]. – Державний стандарт України, 1998, - 211 с.

20. «Правила охорони праці під час експлуатації електронно-обчислювальних машин» : НПАОП 0.00-1.28-10. - [Чинний від 2010-03-26]. – К.: Державний стандарт України, 2010, - 157 с.
21. «Природне і штучне освітлення» : ДБН В.2.5-28-2006. – [Чинний від 2019-03-01]. – К.: Державний стандарт України, 2018, - 137 с.
22. «Санітарні норми мікроклімату виробничих приміщень» : ДСН 3.3.6-042-99. – [Чинний від 1999-12-01]. – К.: Постанова, 1999, - 97 с.
23. «ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны» : ГОСТ 12.1.005-88. – [Действующий от 1988-09-29]. – М.: Стандартиформ, 2008, - 437 с.
24. «Санітарно-гігієнічні норми допустимих рівнів іонізації повітря виробничих та громадських приміщень» : ГН 2152-80. – [Чинний від 2014-04-08]. – К.: Державний стандарт України, 2013, 215 с.
25. «Опалення, вентиляція та кондиціонування» : ДБН В.2.5-67:2013. – [Чинний від 2014-01-01]. – К.: Державний стандарт України, 2013, - 147 с.
26. «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» : ДСН 3.3.6.037-99. – [Чинний від 1999-12-01]. – К.: Постанова, 1999, 124 с.
27. «Пожежна безпека об'єктів будівництва» : ДБН В.1.1.7-2002. – [Чинний від 2003-05-01]. – К.: Державний стандарт України, 2003, - 72 с.
28. «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою» : НАПБ Б.03.002-2007. – [Чинний від 2007-12-03]. – К.: Державний стандарт України, 2007, - 27 с.
29. Безпека у надзвичайних ситуаціях. Навчання населення діям у надзвичайних ситуаціях. Основні положення : ДСТУ 5058:2008. – [Чинний від 2010-01-01]. – К.: Державний стандарт України, 2008, - 15 с.

ДОДАТОК А
Текст програми

A.1 Функція зчитування даних з попередньої бази

```

Private Sub ButtonReadKontingent_Click()
    'Видалення
    res = MsgBox("УВАГА! Всі дані поточного контингенту будуть видалені.", vbYesNo +
vbInformation)

    If (res = vbYes) Then
        res = MsgBox("Точно?", vbYesNo + vbInformation)
        If (res = vbYes) Then
            'Чистка заяв студентів
            CurrentDb.Execute "DELETE * " & _
                              "FROM _StudentRequest;"
            'Чистка змін ПІБ
            CurrentDb.Execute "DELETE * " & _
                              "FROM a_ChangeName;"
            'Чистка студентів у наказі
            CurrentDb.Execute "DELETE * " & _
                              "FROM _OrderStudent;"
            'Чистка пунктів
            CurrentDb.Execute "DELETE * " & _
                              "FROM _OrderPoint;"
            'Чистка наказів
            CurrentDb.Execute "DELETE * " & _
                              "FROM _Order;"
            'Чистка змін ПІБ
            CurrentDb.Execute "DELETE * " & _
                              "FROM a_ChangeName;"
            'Чистка заяв
            CurrentDb.Execute "DELETE * " & _
                              "FROM _StudentRequest;"
            'Чистка студентів
            CurrentDb.Execute "DELETE * " & _
                              "FROM _Student;"
            'Чистка груп
            CurrentDb.Execute "DELETE * " & _
                              "FROM _Group " & _
                              "WHERE (((code_group)<>""0 0""));"
            'Чистка студентів у буфері
            CurrentDb.Execute "DELETE * " & _
                              "FROM b_StudentBuffer;"
        ElseIf (res = vbNo) Then
            GoTo CancelLine
        End If
    ElseIf (res = vbNo) Then
        GoTo CancelLine
    End If

    Dim excelFile As String
    Dim indexRow As Integer
    Dim rst As DAO.Recordset, rstSheet As DAO.Recordset, rstSpecialization As
DAO.Recordset, rstConf As DAO.Recordset
    Dim num_order As String, point As String, detail As String
    Dim emptySheep As Byte

    emptySheep = 0
    If (Me!FieldPathFile <> "") Then
        excelFile = Me!FieldPathFile
        Set xlApp = CreateObject("Excel.Application")
        Set xlBook = xlApp.Workbooks.Open(excelFile)

        Set rstSheet = CurrentDb.OpenRecordset("SELECT b_ExcelSheet.name_sheet,
b_ExcelSheet.bool, b_ExcelSheet.course, b_ExcelSheet.formnavch, b_ExcelSheet.osvstup " & _
        "FROM b_ExcelSheet " & _
        "WHERE (((bool)="" & True & ""));")

        Do While Not rstSheet.EOF
            indexRow = 4

            Set xlSheet = xlBook.WorkSheets(CStr(rstSheet("name_sheet")))
            Me!name_sheet = rstSheet("name_sheet")

            Do While Not emptySheep > 3

                id_specialization = 0
                num_order = ""
                point = ""
                detail = ""
            
```

```

        Факультет
        'Якщо факультету немає, то сторінка закінчилася, але це може бути не кінець, а просто
інша група
        LineAgain:
                If (xlSheet.Range("B" + CStr(indexRow)) = "" Or
IsNull(xlSheet.Range("B" + CStr(indexRow)))) Then
                        indexRow = indexRow + 1
                        emptySheep = emptySheep + 1
                        If (emptySheep > 3) Then
                                emptySheep = 0
                                GoTo StopLine
                        End If
                        GoTo LineAgain
                Else
                        emptySheep = 0
                End If

                sign_fac = xlSheet.Range("B" + CStr(indexRow))

                'Група
                code_group = xlSheet.Range("C" + CStr(indexRow))

                'Прізвище
                last_name = xlSheet.Range("D" + CStr(indexRow))

                'Ім'я
                name_stud = xlSheet.Range("E" + CStr(indexRow))

                'По батькові
                middle_name = xlSheet.Range("F" + CStr(indexRow))

                b = xlSheet.Range("G" + CStr(indexRow))
                k = xlSheet.Range("H" + CStr(indexRow))
                type_financing = ""

                'Бюджет
                If (xlSheet.Range("G" + CStr(indexRow)) Like "1" Or
(xlSheet.Range("G" + CStr(indexRow)) Like "ін" And xlSheet.Range("J" + CStr(indexRow)) Like "1"))
Then
                        type_financing = "Б"
                        If (xlSheet.Range("G" + CStr(indexRow)) Like "ін" Or
xlSheet.Range("J" + CStr(indexRow)) Like "ін") Then
                                isForeigner = True
                        Else
                                isForeigner = False
                        End If
                ElseIf (xlSheet.Range("H" + CStr(indexRow)) Like "1" Or
(xlSheet.Range("H" + CStr(indexRow)) Like "ін" And xlSheet.Range("J" + CStr(indexRow)) Like "1"))
Then
                        type_financing = "К"
                        If (xlSheet.Range("H" + CStr(indexRow)) Like "ін" Or
xlSheet.Range("J" + CStr(indexRow)) Like "ін") Then
                                isForeigner = True
                        Else
                                isForeigner = False
                        End If
                End If

                'Якщо тип фінансування не визначено, студент був відрахований
(особливо магістри 2018 року)
                If (type_financing = "") Then
                        GoTo LineContinue
                End If

                'Освітня програма, визначається по скороченій назві
                If (xlSheet.Range("P" + CStr(indexRow)) <> "" And Not
IsNull(xlSheet.Range("P" + CStr(indexRow)))) Then
                        Set rst = CurrentDb.OpenRecordset("SELECT id_specialization " &
_
                                                "FROM s_Specialization " & _
                                                "WHERE
((name_specializationd_abb)=""" & Trim(xlSheet.Range("P" + CStr(indexRow))) & """);")
                        If (Not rst.EOF) Then
                                id_specialization = rst("id_specialization")
                        End If
                End If

                'Наказ, якщо не Відрахування і не МС в причині

```

```

If (xlSheet.Range("R" + CStr(indexRow)) = "B" Or xlSheet.Range("W" + CStr(indexRow)) = "MC") Then
    'Студент не вноситься в основну таблицю, але у таблицю буфер
    CurrentDb.Execute "INSERT INTO b_StudentBuffer (last_name, name_stud, middle_name, stud_msg, B, K) " & _
        "VALUES (" & last_name & ", " & name_stud & ", " & middle_name & ", " & stud_msg & ", " & b & ", " & k & ");"
    GoTo LineContinue
    'Накази зчитувати потрібно тільки академвідпустки
    'Тільки академвідпустка
    ElseIf (xlSheet.Range("R" + CStr(indexRow)) Like "*Ак*") Then
        num_order = xlSheet.Range("S" + CStr(indexRow))
        point = xlSheet.Range("R" + CStr(indexRow))
        detail = xlSheet.Range("W" + CStr(indexRow))
        year_order = xlSheet.Range("V" + CStr(indexRow))
        dateIn = CDate("01.01.2000")
        If (Len(xlSheet.Range("T" + CStr(indexRow))) = 6) Then
            dateIn = xlSheet.Range("T" + CStr(indexRow)) &
xlSheet.Range("V" + CStr(indexRow))
        End If
    ElseIf (xlSheet.Range("R" + CStr(indexRow)) = "") Then
        num_order = "000-S"
        point = "Перенесено із бази"
        detail = CStr(Date)
        year_order = CStr(Format(Date, "yyyy"))
        dateIn = Format(Date, "mm-dd-yyyy")
    End If

    'Внесення в базу
    'Група
    If (code_group Like "*-*") Then
        'Якщо у пункті академвідпустка, то створюється окрема група із
подавання частки (ак)

        If (point Like "*Ак*" Or point Like "*Ак*") Then
            code_group = code_group & "(ак)"
        End If

        'Якщо такої групи не знайдено, то вноситься як нова
        Set rst = CurrentDb.OpenRecordset("SELECT code_group " & _
            "FROM _Group " & _
            "WHERE (((code_group)="" &
code_group & ""));")

        If (rst.EOF) Then
            CurrentDb.Execute "INSERT INTO _Group (code_group, course,
formnavch, id_specialization, type_group) " & _
                "VALUES (" & code_group & ", " & rstSheet("course") & ", " & rstSheet("formnavch") & ", " & id_specialization & ", " &
typeGroup & ");"
        End If
    End If

    'Додавання студента
    With CurrentProject.Connection
        .Execute "INSERT INTO _Student (last_name, name_stud, middle_name,
sex, type_financing, code_group, isForeigner) " & _
            "VALUES (" & last_name & ", " & name_stud & ", " & middle_name & ", " & sex & ", " & type_financing & ", " & code_group & ", " &
isForeigner & ");"
        id_stud = .Execute("SELECT @@identity")(0)
    End With

    'Додавання наказу
    If (num_order <> "") Then
        'Пошук ідентичного наказу і якщо такого немає, вноситься
        Set rst = CurrentDb.OpenRecordset("SELECT id_order " & _
            "FROM _Order " & _
            "WHERE (((num_order)="" &
num_order & "")) " & _
                " AND ((year_order)="" &
year_order & "");")

        'Внесення наказу
        If (rst.EOF) Then
            With CurrentProject.Connection
                .Execute "INSERT INTO _Order (num_order, year_order, dateIn)
" & _
                    "VALUES (" & num_order & ", " & year_order & ",
"", " & dateIn & ");"
            id_order = .Execute("SELECT @@identity")(0)
        End With
    Else

```

```

        id_order = rst("id_order")
    End If
    'Внесення пункту
    Set rst = CurrentDb.OpenRecordset("SELECT id_point " & _
        "FROM _Order INNER JOIN
_OrderPoint ON [_Order].id_order = [_OrderPoint].id_order " & _
num_order & """) " & _
        "WHERE ((num_order)=""" &
        " AND ((point)=""" & point &
        " AND ((detail)=""" & detail &
        """)));"

    If (rst.EOF) Then
        With CurrentProject.Connection
            .Execute "INSERT INTO _OrderPoint (id_order, point, detail)
" & _
                "VALUES ('" & id_order & "', """ & point & """, """
& detail & """);"

            id_point = .Execute("SELECT @@identity")(0)
        End With
    Else
        id_point = rst("id_point")
    End If

    'Внесення студента до наказу
    CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud,
last_name, name_stud, middle_name, type_financing, code_group, id_specialization, course,
formnavch) " & _
        "VALUES ('" & id_point & "', '" & id_stud & "',
"" & last_name & "", "" & name_stud & "", "" & middle_name & "", "" & type_financing &
"", "" & code_group & "", '" & id_specialization & "', "" & rstSheet("course") & "", '" &
rstSheet("formnavch") & "");"

    End If

    LineContinue:
        indexRow = indexRow + 1
        Me!index = indexRow
    Loop

    StopLine:
        rstSheet.MoveNext
    Loop

    xlApp.Quit
End If

'Видати групам рік появи
Call SetTypeGroup
Call Form__FormMain.ButtonSetYearGroup_Click
CancelLine:
End Sub

```

A.2 Функція генерації звіту «Відомості про рух контингенту»

```

Private Sub ButtonInfoKontingent_Click()
    Set rstTypeFinancing = CurrentDb.OpenRecordset("SELECT type_financing " & _
        "FROM s_typeFinancing " & _
        "ORDER BY id_type_financing;")

    Set rstCourse = CurrentDb.OpenRecordset("SELECT course " & _
        "FROM s_course " & _
        "ORDER BY id_course;")

    Dim start_date As Date
    start_date = Time

    Dim excelFileTemplate As String, excelFileReport As String, pathProject As String,
Path As String
    Dim rstOrder As DAO.Recordset, rstGroup As DAO.Recordset, rstStud As DAO.Recordset,
rstPrevCont As DAO.Recordset
    Dim course As String, formnavch As Byte
    Dim row As Byte, column As Byte

    Dim first_day As Date, last_day As Date
    Dim detail As String
    Dim group_param_string As String

    If (Me!FieldFirstDay <> "" And Not IsNull(Me!FieldFirstDay) And Me!FieldLastDay <>
"" And Not IsNull(Me!FieldLastDay)) Then
        first_day = Me!FieldFirstDay
        last_day = Me!FieldLastDay
    Else

```

```

        MsgBox "Дату початку або кінця не вказано!"
        GoTo StopLine
    End If

    'Розміщення проекту
    pathProject = Application.CurrentProject.Path

    Path = pathProject + "\" & Format(Date, "dd.mm.yy")
    If Dir(Path, vbDirectory) = "" Then
        Mkdir Path
    End If

    excelFileTemplate = pathProject + "\\Template\\InfoMoveKontingent.xlsx"
    excelFileReport = Path + "\\Відомості про рух контингенту_" & Format(Now, "dd/mm/yyyy
hh/mm/ss") & ".xlsx"

    'Copy file template
    FileCopy excelFileTemplate, excelFileReport

    'Open file report for making changes
    Set xlApp = CreateObject("Excel.Application")
    Set xlBook = xlApp.Workbooks.Open(excelFileReport)
    xlApp.Visible = True

    Dim masSheets(3) As String
    masSheets(0) = "Денна форма навчання"
    masSheets(1) = "Заочна форма навчання"
    masSheets(2) = "(Ін)Денна форма навчання"
    masSheets(3) = "(Ін)Заочна форма навчання"

    Dim masColumn() As String
    Dim masKont() As String
    masColumn = Split("C D E F G H I J K L M N O P")
    indexCol = 0

    Set rstConfigSoft = CurrentDb.OpenRecordset("SELECT current_base " & _
                                                "FROM _configSoft;")
    current_base = rstConfigSoft("current_base")
    Set rstConfigBase = CurrentDb.OpenRecordset("SELECT month_base, first_day_base,
last_day_base " & _
                                                "FROM _configBase;")

    'Перемикання на попередню базу
    Call UpdateTableLinks(Application.CurrentProject.Path & "\\StorageBase\\Data DMED_" &
Format(DateAdd("m", -1, rstConfigBase("first_day_base")), "mm") & "_" &
rstConfigBase("first_day_base") & ".accdb")

    'Виставлення дат і завантаження кількості
    For Each sheet In masSheets
        Set xlSheet = xlBook.WorkSheets(sheet)
        If (sheet Like "(Ін)*") Then
            xlSheet.Range("B15") = xlSheet.Range("B15") + CStr(Me!FieldFirstDay)
            xlSheet.Range("B43") = xlSheet.Range("B43") + CStr(Me!FieldLastDay)
        Else
            xlSheet.Range("B14") = xlSheet.Range("B14") + CStr(Me!FieldFirstDay)
            xlSheet.Range("B42") = xlSheet.Range("B42") + CStr(Me!FieldLastDay)
        End If

        'Параметр для підрахунку студентів (форма навчання на основі назви сторінки)
        If (sheet Like "*Денна*") Then
            formnavch = 1
        ElseIf (sheet Like "*Заочна*") Then
            formnavch = 2
        Else
            MsgBox "Форму навчання у назві сторінки не знайдено!" & vbNewLine &
"Створення звіту не завершено.", vbCritical
            'Повернення до поточної бази
            Call UpdateTableLinks(current_base)
            GoTo StopLine
        End If

        'Параметр для підрахунку студентів (форма навчання на основі назви сторінки)
        If (Not sheet Like "(Ін)*") Then
            isForeigner = False
        ElseIf (sheet Like "(Ін)*") Then
            isForeigner = True
        Else
            MsgBox "Форму навчання у назві сторінки не знайдено!" & vbNewLine &
"Створення звіту не завершено.", vbCritical
            'Повернення до поточної бази
            Call UpdateTableLinks(current_base)

```

```

        GoTo StopLine
    End If

    xlSheet.Range("A7") = "за " & FieldMonth.column(1) & " " &
Format(Me!FieldFirstDay, "yyyy") & " року"

    indexCol = 0
    rstCourse.MoveFirst
    Do While Not rstCourse.EOF

        rstTypeFinancing.MoveFirst
        Do While Not rstTypeFinancing.EOF
            'Підрахунок студентів у попередній базі
            Set rstStud = CurrentDb.OpenRecordset("SELECT Count([_Student].id_stud)
As Count_stud " & _
                                                    "FROM _Group INNER JOIN _Student
ON [_Group].id_group = [_Student].id_group " & _
                                                    "WHERE ((([_Group].formnavch)=" &
formnavch & ") " & _
                                                    " AND ((([_Group].course)=" &
rstCourse("course") & "")) " & _
                                                    " AND ((([_Group].code_group) Like
""*-*)" " & _
                                                    " AND
(([_Student].type_financing)=" & rstTypeFinancing("type_financing") & "")) " & _
                                                    " AND ((([_Student].isForeigner)="
& isForeigner & "));")

            If (sheet Like "(И)*)") Then
                xlSheet.Range(masColumn(indexCol) & "16") = rstStud("Count_stud")
            Else
                xlSheet.Range(masColumn(indexCol) & "15") = rstStud("Count_stud")
            End If
            indexCol = indexCol + 1

            rstTypeFinancing.MoveNext
        Loop

        rstCourse.MoveNext
    Loop

    rstStud.Close

    'Повернення до поточної бази
    Call UpdateTableLinks(current_base)

    Set rstAllStudents = CurrentDb.OpenRecordset("SELECT [_OrderStudent].id_stud " & _
                                                    "FROM _Order INNER JOIN
(_OrderPoint INNER JOIN _OrderStudent ON [_OrderPoint].id_point = [_OrderStudent].id_point) ON
[_Order].id_order = [_OrderPoint].id_order " & _
                                                    "WHERE ((([_Order].dateIn)>=#" &
Format(first_day, "yyyy-mm-dd") & "#) " & _
                                                    " AND ((([_Order].dateIn)<=#" &
Format(last_day, "yyyy-mm-dd") & "#)) " & _
                                                    "GROUP BY
[_OrderStudent].id_stud;")

    Do While Not rstAllStudents.EOF

        Set rstOrdersStudent = CurrentDb.OpenRecordset("SELECT [_Student].isForeigner,
[_Order].dateIn, [_OrderPoint].id_point, [_OrderPoint].point, [_OrderPoint].detail,
[_OrderPoint].another, [_OrderStudent].last_name, [_OrderStudent].type_financing,
[_OrderStudent].course, [_OrderStudent].formnavch, [_OrderStudent].code_group " & _
                                                    "FROM _Order INNER JOIN
(_OrderPoint INNER JOIN (_OrderStudent INNER JOIN _Student ON [_OrderStudent].id_stud =
[_Student].id_stud) ON [_OrderPoint].id_point = [_OrderStudent].id_point) ON [_Order].id_order =
[_OrderPoint].id_order " & _
                                                    "WHERE
((([_OrderStudent].id_stud)=" & rstAllStudents("id_stud") & ")) " & _
                                                    " AND
(([_Order].dateIn)>=#" & Format(first_day, "yyyy-mm-dd") & "#) " & _
                                                    " AND
(([_Order].dateIn)<=#" & Format(last_day, "yyyy-mm-dd") & "#) " & _
                                                    "ORDER BY [_Order].dateIn
DESC;")

        '+
        Do While Not rstOrdersStudent.EOF

            r_id_point = rstOrdersStudent("id_point")
            r_point = rstOrdersStudent("point")
            If (r_point Like "Приєм*") Then

```

```

        Debug.Print r_point
    End If
    r_detail = rstOrdersStudent("detail")
    r_another = rstOrdersStudent("another")
    r_type_financing = rstOrdersStudent("type_financing")
    r_course = rstOrdersStudent("course")
    r_formnavch = rstOrdersStudent("formnavch")

    'Приставка до сторінки залежно від іноземного навчання
    If (rstOrdersStudent("isForeigner")) Then
        prefixPage = "(Ін)"
    Else
        prefixPage = ""
    End If

    'Перехід на потрібну сторінку за формою навчання і іноземцем
    If (r_formnavch = 1) Then
        str_sheet = prefixPage & "Денна форма навчання"
    ElseIf (r_formnavch = 2) Then
        str_sheet = prefixPage & "Заочна форма навчання"
    End If
    Set xlSheet = xlBook.WorkSheets(str_sheet)

    indexRow = 14

    'Якщо академвідпустка і студент був на бюджеті, то пункт перетворюється в
    Переведення а деталь у зміну типу фінансування
    isAK = False
    If (r_point = "Академвідпустка") Then
        r_point = "Переведення"
        r_detail = "зміна форми фінансування"
        isAK = True

    'Якщо Повернено з академвідпустки
    ElseIf (r_point = "Повернено з академвідпустки") Then
        r_point = "Переведення"
        r_detail = "зміна форми фінансування"
        isAK = True
    End If

    '.....'Для випадку переведення
    If (r_point = "Переведення" Or r_point = "Переведення групи") Then

        'Зчитується отриманий пункт переведення або переведення групи, який є
        пунктом <Переведено на інші форми навчання>
        p_point = "Переведено на інші форми навчання" 'Звідки студент був
        відрахований

        'Якщо переведення групи, то деталізація пункту буде (інші випадки)
        'Якщо переведення, то деталь із наказу
        If (r_point = "Переведення групи") Then
            p_detail = "інші випадки"
        Else
            p_detail = r_detail
        End If
        p_formnavch = r_formnavch
        p_type_financing = r_type_financing

        'Попередній наказ (звідки студент був переведений)
        rstOrdersStudent.MoveNext

        If (rstOrdersStudent.EOF) Then
            Set rstPrevOrder = CurrentDb.OpenRecordset ("SELECT
            [_OrderPoint].id_point, [_OrderPoint].point, [_OrderPoint].detail, [_OrderStudent].last_name,
            [_OrderStudent].type_financing, [_OrderStudent].course, [_OrderStudent].formnavch,
            [_OrderStudent].code_group " & _
            "FROM _Order INNER JOIN
            (_OrderPoint INNER JOIN _OrderStudent ON [_OrderPoint].id_point = [_OrderStudent].id_point) ON
            [_Order].id_order = [_OrderPoint].id_order " & _
            "WHERE
            ((([_OrderStudent].id_stud)= " & rstAllStudents("id_stud") & ") " & _
            " AND (([_Order].dateIn)<#"
            & Format(first_day, "yyyy-mm-dd") & "#)) " & _
            "ORDER BY [_Order].dateIn
            DESC;")

            prev_if_course = rstPrevOrder("course")
            prev_if_formnavch = rstPrevOrder("formnavch")
            prev_if_type_financing = rstPrevOrder("type_financing")
            rstOrdersStudent.MovePrevious
        ElseIf (Not rstOrdersStudent.EOF) Then

```

```

prev_if_course = rstOrdersStudent("course")
prev_if_formnavch = rstOrdersStudent("formnavch")
prev_if_type_financing = rstOrdersStudent("type_financing")
rstOrdersStudent.MovePrevious
End If

'Якщо академвідпустка і попередній тип фінансування був контракт,
виконується пропуск обробки пункта
If (isAK And prev_if_type_financing = "К") Then
    GoTo SkipLine
End If

'Якщо переведення групи, то курс стягується із попереднього наказу
'If (r_point = "Переведення групи") Then
    p_course = prev_if_course 'rstPrevOrder("course")
'Else
'    p_course = r_course
'End If

''''''''Пункт Переведення - Інші випадки - Іноземець отримав громадянство
If (p_detail = "інші випадки" And r_another = "отримання громадянства") Then
    'Іноземець виключається із сторінки іноземців і включається на сторінці
звичайних студентів

    'Змінюється префікс сторінки і індекс рядку
    previousPrefixPage = prefixPage
    prefixPage = "(Ін)"
End If

'Перехід на потрібну сторінку
If (prev_if_formnavch = 1) Then
    str_sheet = prefixPage & "Денна форма навчання"
ElseIf (prev_if_formnavch = 2) Then
    str_sheet = prefixPage & "Заочна форма навчання"
End If
Set xlSheet = xlBook.WorkSheets(str_sheet)

'rstOrdersStudent.MovePrevious

Do While indexRow <> 50
    'Пошук пункту
    If (xlSheet.Range("B" & CStr(indexRow)) <> "" And xlSheet.Range("B" &
CStr(indexRow)) Like "*:") Then
        point = Left(xlSheet.Range("B" & CStr(indexRow)),
Len(xlSheet.Range("B" & CStr(indexRow))) - 1)
    Else
        point = xlSheet.Range("B" & CStr(indexRow))
    End If

    If (point = p_point) Then

        'Пошук підпункту
        indexRow = indexRow + 1
        Do While indexRow <> 50 And xlSheet.Range("A" & CStr(indexRow)) = ""
            detail = xlSheet.Range("B" & CStr(indexRow))
            detail = Right(detail, Len(detail) - 3)
            If (detail = p_detail) Then
                'Підрахунок індексу колонки
                'за курсом
                If (p_course Like "*м") Then
                    indexCol = Left(p_course, 1) * 2 + 6
                Else
                    indexCol = (CInt(p_course) - 1) * 2
                End If
                'за типом фінансування
                If (prev_if_type_financing = "Б") Then
                    indexCol = indexCol + 0
                ElseIf (prev_if_type_financing = "К") Then
                    indexCol = indexCol + 1
                End If

                'Перенесення
                xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) =
xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) + 1
                'Закінчення
                GoTo OkLineSpecial1
            End If

            indexRow = indexRow + 1
        Loop
    End If

```

```

        indexRow = indexRow + 1
    Loop
OkLineSpecial1:

    'Поточний пункт (куди був перенесений студент) (Переведено з ...)
    indexRow = 14
    p_point = "Переведено з інших форм навчання"
    If (r_point = "Переведення групи") Then
        p_detail = "інші випадки"
    Else
        If (r_detail = "денної") Then
            p_detail = "заочної"
        ElseIf (r_detail = "заочної") Then
            p_detail = "денної"
        Else
            p_detail = r_detail
        End If
    End If

    p_formnavch = rstOrdersStudent("formnavch")
    p_type_financing = rstOrdersStudent("type_financing")
    p_course = rstOrdersStudent("course")

    'Пункт Переведення - Інші випадки - Іноземець отримав громадянство
    If (p_detail = "інші випадки" And r_another = "отримання громадянства") Then
        'Іноземець виключається із сторінки іноземців і включається на сторінці
звичайних студентів
        'Змінюється префікс сторінки на попередній
        prefixPage = previousPrefixPage
    End If

    'Перехід на потрібну сторінку
    If (r_formnavch = 1) Then
        str_sheet = prefixPage & "Денна форма навчання"
    ElseIf (r_formnavch = 2) Then
        str_sheet = prefixPage & "Заочна форма навчання"
    End If
    Set xlSheet = xlBook.WorkSheets(str_sheet)

    Do While indexRow <> 50
        'Пошук пункту
        If (xlSheet.Range("B" & CStr(indexRow)) <> "" And xlSheet.Range("B" &
CStr(indexRow)) Like "*:") Then
            point = Left(xlSheet.Range("B" & CStr(indexRow)), Len(xlSheet.Range("B"
& CStr(indexRow))) - 1)
        Else
            point = xlSheet.Range("B" & CStr(indexRow))
        End If

        If (point = p_point) Then
            'Пошук підпункту. Підпункт залишається поточним
            indexRow = indexRow + 1
            Do While indexRow <> 50 And xlSheet.Range("A" & CStr(indexRow)) = ""
                detail = xlSheet.Range("B" & CStr(indexRow))
                detail = Right(detail, Len(detail) - 3)
                If (detail = p_detail) Then
                    'Підрахунок індексу колонки
                    ''за курсом
                    If (p_course Like "*м") Then
                        indexCol = Left(p_course, 1) * 2 + 6
                    Else
                        indexCol = (CInt(p_course) - 1) * 2
                    End If
                    ''за типом фінансування
                    If (p_type_financing = "Б") Then
                        indexCol = indexCol + 0
                    ElseIf (p_type_financing = "К") Then
                        indexCol = indexCol + 1
                    End If
                    'Перенесення
                    xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) =
xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) + 1
                    'Закінчення
                    GoTo OkLineSpecial2
                End If

                indexRow = indexRow + 1
            Loop
        End If
    End If

```

```

        indexRow = indexRow + 1
    Loop
OkLineSpecial2:

    '.....'Поновлення
    ElseIf (r_point = "Поновлення") Then
        p_id_point = r_id_point
        p_point = r_point
        p_detail = r_detail
        p_type_financing = r_type_financing
        p_course = r_course
        p_formnavch = r_formnavch

        Do While indexRow <> 50
            'Пошук пункту
            If (xlSheet.Range("B" & CStr(indexRow)) <> "" And xlSheet.Range("B" &
CStr(indexRow)) Like "*:") Then
                point = Left(xlSheet.Range("B" & CStr(indexRow)),
Len(xlSheet.Range("B" & CStr(indexRow))) - 1)
            Else
                point = xlSheet.Range("B" & CStr(indexRow))
            End If

            If (point = p_point Or Left(point, 30) = Left(p_point, 30)) Then
                'Пошук підпункту
                save_num = xlSheet.Range("A" & CStr(indexRow))
                Do While indexRow <> 50 And (save_num = xlSheet.Range("A" &
CStr(indexRow)) Or xlSheet.Range("A" & CStr(indexRow)) = "")
                    detail = xlSheet.Range("B" & CStr(indexRow))
                    'Якщо це підпункт
                    If (detail Like "(*)") Then
                        detail = Right(detail, Len(detail) - 3)
                    End If

                    If (detail = p_detail Or Left(detail, 30) = Left(p_detail, 30))

Then
                        'Підрахунок індексу колонки
                        'За курсом
                        If (r_course Like "*м") Then
                            indexCol = Left(p_course, 1) * 2 + 6
                        Else
                            indexCol = (CInt(p_course) - 1) * 2
                        End If
                        'За типом фінансування
                        If (p_type_financing = "Б") Then
                            indexCol = indexCol + 0
                        ElseIf (p_type_financing = "К") Then
                            indexCol = indexCol + 1
                        End If
                        'Перенесення
                        xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) =
xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) + 1

                        'Тепер попередні накази читаються окремо
                        Set rstPrevOrder = CurrentDb.OpenRecordset("SELECT
[_OrderPoint].id_point, [_OrderPoint].point, [_OrderPoint].detail, [_OrderStudent].last_name,
[_OrderStudent].type_financing, [_OrderStudent].course, [_OrderStudent].formnavch,
[_OrderStudent].code_group " & _
                                "FROM _Order INNER JOIN
(_OrderPoint INNER JOIN _OrderStudent ON [_OrderPoint].id_point = [_OrderStudent].id_point) ON
[_Order].id_order = [_OrderPoint].id_order " & _
                                "WHERE
(([_OrderStudent].id_stud)= " & rstAllStudents("id_stud") & " ) " & _
                                " AND (([_Order].dateIn)<#"
& Format(first_day, "yyyy-mm-dd") & "#)) " & _
                                "ORDER BY [_Order].dateIn
DESC;")

                        If (rstPrevOrder.RecordCount <> 1 And Not rstPrevOrder.EOF)

Then
                                If (rstPrevOrder("point") = "Ак.в." Or
rstPrevOrder("point") = "Ак.в." Or rstPrevOrder("point") = "Академвідпустка") Then
                                    If (oneSheet Like "Ін.*") Then
                                        xlSheet.Range(masColumn(indexCol) & "45") =
xlSheet.Range(masColumn(indexCol) & "45") + 1
                                    Else
                                        xlSheet.Range(masColumn(indexCol) & "44") =
xlSheet.Range(masColumn(indexCol) & "44") + 1
                                    End If
                                End If
                            End If

```

```

        End If
    End If

    'Закінчення
    GoTo OkLine2
End If

    indexRow = indexRow + 1
Loop
End If

    indexRow = indexRow + 1
Loop
OkLine2:

    '.....'Для інших випадків
    ElseIf (r_point <> "Академвідпустка" And r_point <> "Повернено з
академвідпустки") Then

        p_id_point = r_id_point
        p_point = r_point
        p_detail = r_detail
        p_type_financing = r_type_financing
        p_course = r_course
        p_formnavch = r_formnavch

        Do While indexRow <> 50
            'Пошук пункту
            If (xlSheet.Range("B" & CStr(indexRow)) <> "" And xlSheet.Range("B" &
CStr(indexRow)) Like "**:") Then
                point = Left(xlSheet.Range("B" & CStr(indexRow)),
Len(xlSheet.Range("B" & CStr(indexRow))) - 1)
            Else
                point = xlSheet.Range("B" & CStr(indexRow))
            End If

            If (point = p_point Or Left(point, 30) = Left(p_point, 30)) Then
                'Пошук підпункту
                save_num = xlSheet.Range("A" & CStr(indexRow))
                Do While indexRow <> 50 And (save_num = xlSheet.Range("A" &
CStr(indexRow)) Or xlSheet.Range("A" & CStr(indexRow)) = "")
                    detail = xlSheet.Range("B" & CStr(indexRow))
                    'Якщо це підпункт
                    If (detail Like "**")* And Not detail Like "Приєм*") Then
                        detail = Right(detail, Len(detail) - 3)
                    End If

                    'Якщо Приєм*
                    If (detail Like "Приєм*") Then
                        indexRow = indexRow + 1
                    End If

                    If (detail = p_detail Or Left(detail, 30) = Left(p_detail, 30))

Then
                        'Підрахунок індексу колонки
                        'за курсом
                        If (r_course Like "*м") Then
                            indexCol = Left(p_course, 1) * 2 + 6
                        Else
                            indexCol = (CInt(p_course) - 1) * 2
                        End If
                        'за типом фінансування
                        If (p_type_financing = "Б") Then
                            indexCol = indexCol + 0
                        ElseIf (p_type_financing = "К") Then
                            indexCol = indexCol + 1
                        End If
                        'Перенесення
                        xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) =
xlSheet.Range(masColumn(indexCol) & CStr(indexRow)) + 1
                        'Закінчення
                        GoTo OkLine
                    End If

                    indexRow = indexRow + 1
                Loop
            End If

            indexRow = indexRow + 1
        Loop
    End If

```

```

OkLine:
SkipLine:
    End If

    '+
    rstOrdersStudent.MoveNext
    Loop

    rstAllStudents.MoveNext
    Loop

    'Save and close file report
    xlBook.Save
    xlApp.Quit

    MsgBox "Генерація звіту <Відомості про рух контингенту> завершено." & vbNewLine &
    CStr(start_date) & "-" & CStr(Time), vbInformation, "Генерація звіту"
    StopLine:
End Sub

```

A.3 Функція для генерації звіту «Ліцензія»

```

Private Sub ButtonLicense_Click()
    'Із іноземцями або без іноземців
    Dim withForeigner As Boolean
    If (Me!CheckBoxLicenseWithForeigner) Then
        prefix = "з іноземцями"
        unit_sql_with_foreiner = ""
    Else
        prefix = "без іноземців"
        unit_sql_with_foreiner = " AND ([_Student].isForeigner)=False) "
    End If

    Dim start_date As Date
    start_date = Time

    Dim excelFileTemplate As String, excelFileReport As String, pathProject As String,
    Path As String
    Dim rstStud As DAO.Recordset
    Dim course As String, formnavch As Byte
    Dim indexRow As Byte, indexColumn As Byte, indexLicRow As Byte
    Dim masColumn() As String
    Dim first_day As Date, last_day As Date
    Dim detail As String
    masColumn = Split("C D G H I J K L M N O P")
    indexColumn = 0

    If (Me!FieldFirstDay <> "" And Not IsNull(Me!FieldFirstDay) And Me!FieldLastDay <>
    "" And Not IsNull(Me!FieldLastDay)) Then
        first_day = Me!FieldFirstDay
        last_day = Me!FieldLastDay
    Else
        MsgBox "Дата початку або кінця не вказано! Генерацію зупинено."
    End If

    'Визначення розміщення проекту
    pathProject = Application.CurrentProject.Path

    Path = pathProject + "\" & Format(Date, "dd.mm.yy")
    If Dir(Path, vbDirectory) = "" Then
        MkDir Path
    End If

    excelFileTemplate = pathProject + "\Template\License.xlsm"
    excelFileReport = Path + "\Ліцензія_" & prefix & " " & Format(Now, "dd/mm/yyyy
    hh/mm/ss") & ".xlsm"
    Message "Створення звіту " & excelFileReport

    'Copy file template
    FileCopy excelFileTemplate, excelFileReport

    'Open file report for making changes
    Set xlApp = CreateObject("Excel.Application")
    Set xlBook = xlApp.Workbooks.Open(excelFileReport)
    xlApp.Visible = True

    Dim masSheets(5) As String
    masSheets(0) = "ДН(зв)"

```

```

masSheets(1) = "ЗН(ЗВ)"
masSheets(2) = "ДН(О)"
masSheets(3) = "ДН(К)"
masSheets(4) = "ЗН(О)"
masSheets(5) = "ЗН(К)"

Set rstCourse = CurrentDb.OpenRecordset("SELECT course " & _
                                         "FROM s_course " & _
                                         "ORDER BY id_course;")

Set rstSS = CurrentDb.OpenRecordset("SELECT s_Specialty.id_spec,
s_Specialty.name_spec, s_Specialization.id_specialization, s_Specialization.name_specialization "
& _
                                         "FROM s_Specialty INNER JOIN s_Specialization ON
s_Specialty.id_spec = s_Specialization.id_spec " & _
                                         "WHERE (((s_Specialty.id_spec)<>"" & "000" &
"")) " & _
                                         "      AND
((s_Specialization.bool_activity)=True)) " & _
                                         "ORDER BY s_Specialty.id_spec;")

'Основний цикл перебору сторінок
For Each oneSheet In masSheets
    Set xlSheet = xlBook.WorkSheets(oneSheet)
    xlSheet.Range("A1") = xlSheet.Range("A1") & " " & last_day & " p.  "

    indexRow = 5
    If (oneSheet Like "*Ден*" Or oneSheet Like "*ДН*") Then
        formnavch = 1
        indexLicRow = 0
    ElseIf (oneSheet Like "*Зно*" Or oneSheet Like "*ЗН*") Then
        formnavch = 2
        indexLicRow = 1
    Else
        MsgBox "Форма навчання не визначена!"
        xlBook.Save
        xlApp.Quit
        GoTo StopLine
    End If

    If (oneSheet Like "*(О)*") Then
        sql_type_financing = "      AND (([_Student].type_financing)=""Б"" )
    ElseIf (oneSheet Like "*(К)*") Then
        sql_type_financing = "      AND (([_Student].type_financing)=""К"" )
    ElseIf (oneSheet Like "*(ЗВ)*") Then
        sql_type_financing = "      AND ((([_Student].type_financing)=""Б"" ) & _
                                "      OR (([_Student].type_financing)=""К"")) " & _
    Else
        MsgBox "Тип фінансування не визначено!"
        xlBook.Save
        xlApp.Quit
        GoTo StopLine
    End If

    rstSS.MoveFirst
    Do While Not rstSS.EOF
        xlSheet.Range("A" & CStr(indexRow)) = rstSS("id_spec") & " " &
rstSS("name_spec")
        xlSheet.Range("B" & CStr(indexRow)) = rstSS("name_specialization")

        mas = ParsingLicense(CStr(rstSS("id_spec")))

        rstCourse.MoveFirst
        indexColumn = 0
        Do While Not rstCourse.EOF
            If (rstCourse("course") Like "*М") Then
                indexLicCol = CInt(Left(rstCourse("course"), 1)) + 3
            Else
                indexLicCol = CInt(rstCourse("course")) - 1
            End If

            xlSheet.Range(masColumn(indexColumn) & CStr(indexRow)) =
mas(indexLicRow, indexLicCol)

            Set rstStud = CurrentDb.OpenRecordset("SELECT Count([_Student].id_stud)
AS [Count_stud] " & _
                                         "FROM (s_Specialization INNER JOIN
_Group ON s_Specialization.id_specialization = [_Group].id_specialization) INNER JOIN _Student ON
[_Group].id_group = [_Student].id_group " & _
                                         "WHERE ((([_Group].course)="" &
rstCourse("course") & "") " & _

```

```

formnavch & ") " & _
Like "***-****") " & _
((s_Specialization.id_specialization)=" & rstSS("id_specialization") & ") " & _
                                                                    " AND ([_Group].formnavch)=" &
                                                                    " AND ([_Group].code_group)
                                                                    " AND
                                                                    unit_sql_with_foreiner & _
                                                                    sql_type_financing & ");")

    If (Not rstStud.EOF) Then
        xlSheet.Range(masColumn(indexColumn + 1) & CStr(indexRow)) =
rstStud("Count_stud")
    End If

    If (rstCourse("course") = "1") Then
        '1 крпс 30
        Set rstStud = CurrentDb.OpenRecordset("SELECT
Count([_Student].id_stud) AS [Count_stud] " & _
                                                                    "FROM (s_Specialization INNER
JOIN _Group ON s_Specialization.id_specialization = [_Group].id_specialization) INNER JOIN
_Student ON [_Group].id_group = [_Student].id_group " & _
                                                                    "WHERE ((([_Group].course)="")
& rstCourse("course") & "****") " & _
                                                                    " AND
                                                                    " AND
                                                                    " AND
                                                                    " AND
                                                                    " AND
                                                                    unit_sql_with_foreiner & _
                                                                    sql_type_financing & ");")
        xlSheet.Range("E" & CStr(indexRow)) = rstStud("Count_stud")

        '1 крпс MC
        Set rstStud = CurrentDb.OpenRecordset("SELECT
Count([_Student].id_stud) AS [Count_stud] " & _
                                                                    "FROM (s_Specialization INNER
JOIN _Group ON s_Specialization.id_specialization = [_Group].id_specialization) INNER JOIN
_Student ON [_Group].id_group = [_Student].id_group " & _
                                                                    "WHERE ((([_Group].course)="")
& rstCourse("course") & "****") " & _
                                                                    " AND
                                                                    " AND
                                                                    " AND
                                                                    " AND
                                                                    " AND
                                                                    unit_sql_with_foreiner & _
                                                                    sql_type_financing & ");")
        xlSheet.Range("F" & CStr(indexRow)) = rstStud("Count_stud")
    End If

    indexColumn = indexColumn + 2
    rstCourse.MoveNext
    Loop
    indexRow = indexRow + 1
    rstSS.MoveNext
    Loop
Next

For Each oneSheet In masSheets
    'Відмалювання файлу
    Set xlSheet = xlBook.WorkSheets(oneSheet)
    xlApp.Run "Painting", xlSheet.name
Next

'Save and close file report
xlBook.Save
xlApp.Quit

StopLine:
    MsgBox "Генерація звіту <Ліцензія> завершена." & vbNewLine & CStr(start_date) & "-"
& CStr(Time), vbInformation, "Генерація звіту"
    Message "Генерація звіту <Ліцензія> завершена." & vbNewLine & CStr(start_date) & "-"
& CStr(Time)
End Sub

```

А.4 Функція для генерації звіту «Ставки»

```

Private Sub ButtonReportBids_Click()
    Dim start_date As Date
    start_date = Time

    Dim excelFileTemplate As String, excelFileReport As String, pathProject As String,
Path As String
    Dim rstStud As DAO.Recordset
    Dim course As String, formnavch As Byte
    Dim indexRow As Byte, indexColumn As Byte, indexLicRow As Byte
    Dim masColumn() As String
    Dim first_day As Date, last_day As Date
    Dim detail As String
    masColumn = Split("C D G H I J K L M N O P")
    indexColumn = 0

    If (Me!FieldFirstDay <> "" And Not IsNull(Me!FieldFirstDay) And Me!FieldLastDay <>
"" And Not IsNull(Me!FieldLastDay)) Then
        first_day = Me!FieldFirstDay
        last_day = Me!FieldLastDay
    Else
        MsgBox "Дата початку або кінця не вказано! Генерацію зупинено.", vbOKOnly +
vbCritical
    End If

    'Визначення розміщення проекту
    pathProject = Application.CurrentProject.Path

    Path = pathProject + "\" & Format(Date, "dd.mm.yy")
    If Dir(Path, vbDirectory) = "" Then
        MkDir Path
    End If

    excelFileTemplate = pathProject + "\Template\Bids.xlsx"
    excelFileReport = Path + "\Ставки_" & Format(Now, "dd/mm/yyyy hh/mm/ss") & ".xlsx"

    'Copy file template
    FileCopy excelFileTemplate, excelFileReport

    'Open file report for making changes
    Set xlApp = CreateObject("Excel.Application")
    xlApp.Visible = False
    Set xlBook = xlApp.Workbooks.Open(excelFileReport)

    Set rstTypeFinancing = CurrentDb.OpenRecordset("SELECT type_financing,
type_financing_full " & _
"FROM s_typeFinancing;")
    Set rstFormnavch = CurrentDb.OpenRecordset("SELECT id_formnavch " & _
"FROM s_formnavch;")

    'Дата для сторінки Зведена
    Set xlSheet = xlBook.WorkSheets("Зведена")
    xlSheet.Range("A2") = xlSheet.Range("A2") & last_day & " p."

    Do While Not rstTypeFinancing.EOF
        Set xlSheet = xlBook.WorkSheets(CStr(rstTypeFinancing("type_financing_full")))
        xlSheet.Range("A2") = xlSheet.Range("A2") & last_day & " p."
        'Бюджет
        indexRow = 7
        Do While Not xlSheet.Range("A" & CStr(indexRow)) Like "Всьо*"
            id_spec = xlSheet.Range("A" & CStr(indexRow))
            'Бакалаври
            rstFormnavch.MoveFirst
            Do While Not rstFormnavch.EOF
                Set rstStud = CurrentDb.OpenRecordset("SELECT Count([_Student].id_stud)
As Count_stud " & _
"FROM (s_Specialization INNER JOIN
_Group ON s_Specialization.id_specialization = [_Group].id_specialization) INNER JOIN _Student ON
[_Group].id_group = [_Student].id_group " & _
"WHERE
((s_Specialization.id_spec)=""" & id_spec & """) " & _
" AND (([_Group].formnavch)=""" &
rstFormnavch("id_formnavch") & """) " & _
" AND (([_Group].course) <=""4""")
" & _
" AND (([_Group].course) Not
Like ""*М""") " & _

```

```

Like "*"~*") " & _
" AND ([_Group].code_group)
" AND
(([_Student].type_financing)=" " & rstTypeFinancing("type_financing") & " "));")
'Відштовхується від статичного стовпця, який знаходиться перед даним. На
основі цього стовпця знаходиться потрібний стовпець способом додавання коду форми навчання.
xlSheet.Range(Chr(Asc("F") + rstFormnavch("id_formnavch")) &
CStr(indexRow)) = rstStud("Count_stud")

rstFormnavch.MoveNext
Loop

'Mарістри
rstFormnavch.MoveFirst
Do While Not rstFormnavch.EOF
Set rstStud = CurrentDb.OpenRecordset("SELECT Count([_Student].id_stud)
As Count_stud " & _
"FROM (s_Specialization INNER JOIN
[_Group] ON s_Specialization.id_specialization = [_Group].id_specialization) INNER JOIN _Student ON
[_Group].id_group = [_Student].id_group " & _
"WHERE
" AND ([_Group].formnavch)=" &
" AND ([_Group].course) Like
" AND ([_Group].code_group)
" AND
(([_Student].type_financing)=" " & rstTypeFinancing("type_financing") & " "));")
'Відштовхується від статичного стовпця, який знаходиться перед даним. На
основі цього стовпця знаходиться потрібний стовпець способом додавання коду форми навчання.
xlSheet.Range(Chr(Asc("R") + rstFormnavch("id_formnavch")) &
CStr(indexRow)) = rstStud("Count_stud")

rstFormnavch.MoveNext
Loop

indexRow = indexRow + 1
Loop

rstTypeFinancing.MoveNext
Loop
'Save and close file report
xlBook.Save
xlApp.Quit

StopLine:
MsgBox "Звіт <Ставки> завершений." & vbNewLine & CStr(start_date) & "-" &
CStr(Time), vbInformation, "Генерація звіту"
End Sub

```

A.5 Функція для генерації звіту «Рух контингенту»

```

Private Sub ButtonReportMoveKontingent_Click()
Dim start_date As Date
start_date = Time

Dim first_day As Date
Dim last_day As Date
Dim month_report As String

If (Me!FieldFirstDay <> "" And Not IsNull(Me!FieldFirstDay) _
And Me!FieldLastDay <> "" And Not IsNull(Me!FieldLastDay) _
And Me!FieldMonth <> "" And Not IsNull(FieldMonth)) Then

first_day = Me!FieldFirstDay
last_day = Me!FieldLastDay
month_report = Me!FieldMonth.column(1)
Else
MsgBox "Дату початку, кінця або місяць не вказано!" & vbNewLine & "Генерацію
зупинено!"

GoTo StopLine
End If

Dim excelFileTemplate As String, excelFileReport As String, pathProject As String,
Path As String

```

```

Dim rst As DAO.Recordset, rstSpecialization As DAO.Recordset, rstGroup As
DAO.Recordset
Dim rstTemp As DAO.Recordset, rstStudents As DAO.Recordset
Dim rstFormnavch As DAO.Recordset, rstCourse As DAO.Recordset
Dim indexRow As Integer, sumGroupOnSpecialization As Integer, sumStudB As Integer,
sumStudK As Integer

Dim sum_group_spec As Integer
Dim sum_stud_B_spec As Integer, sum_stud_K_spec As Integer
Dim sum_stud_B_inoz_spec As Integer, sum_stud_K_inoz_spec As Integer

Dim sum_group_fac As Integer
Dim sum_stud_B_fac As Integer, sum_stud_K_fac As Integer
Dim sum_stud_B_inoz_fac As Integer, sum_stud_K_inoz_fac As Integer

Dim indexCol As Byte
Dim masCharColumn() As String

masCol = Split("D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD AE AF AG
AH AI AJ AK AL AM AN AO AP AQ AR AS AT AU AV AW AX AY AZ BA BB BC BD BE BF BG BH BI BJ BK BL BM
BN BO BP BQ BR BR BS BT BU BV BW BX BY BZ")

'Визначення розміщення проекту
pathProject = Application.CurrentProject.Path

Path = pathProject + "\" & Format(Date, "dd.mm.yy")
If Dir(Path, vbDirectory) = "" Then
    MkDir Path
End If

excelFileTemplate = pathProject + "\Template\MoveContingent.xlsm"
excelFileReport = Path + "\Рух контингенту_" & Format(Now, "dd/mm/yyyy hh/mm/ss") &
".xlsm"
Message "Створення звіту " & excelFileReport

'Copy file template
FileCopy excelFileTemplate, excelFileReport

'Open file report for making changes
Set xlApp = CreateObject("Excel.Application")
Set xlBook = xlApp.Workbooks.Open(excelFileReport)

Set rstFormnavch = CurrentDb.OpenRecordset("SELECT id_formnavch, formnavch " & _
"FROM s_formnavch " & _
"WHERE ((id_formnavch)<>0));")
Set rstCourse = CurrentDb.OpenRecordset("SELECT id_course, course " & _
"FROM s_course " & _
"ORDER BY id_course;")
Set rstTypeFin = CurrentDb.OpenRecordset("SELECT type_financing " & _
"FROM s_typeFinancing " & _
"ORDER BY type_financing;")
Set rstFSS = CurrentDb.OpenRecordset("SELECT s_Facultet.sign_fac,
s_Specialty.id_spec, s_Specialty.name_spec, s_Specialization.id_specialization,
s_Specialization.name_specialization " & _
"FROM s_Facultet INNER JOIN (s_Specialty INNER
JOIN s_Specialization ON s_Specialty.id_spec = s_Specialization.id_spec) ON s_Facultet.id_fac =
s_Specialization.id_fac " & _
"WHERE (((s_Facultet.id_fac)<>" & "0" & ") " & _
" AND ((s_Specialization.bool_activity)=True)) "
& _
"ORDER BY s_Facultet.out, s_Specialty.id_spec;")

Do While Not rstFormnavch.EOF
    Set xlSheet = xlBook.WorkSheets(CStr(rstFormnavch("formnavch")))

    xlSheet.Range("C1") = xlSheet.Range("C1") & month_report & "( на " & last_day &
" p. )"

    indexRow = 5

    rstFSS.MoveFirst
    saveSignFac = rstFSS("sign_fac")
    Do While Not rstFSS.EOF
        'Суми
        If (saveSignFac <> rstFSS("sign_fac")) Then
            xlSheet.Range("B" & CStr(indexRow)) = "Студентів на " & saveSignFac

            saveSignFac = rstFSS("sign_fac")
            indexRow = indexRow + 1
        End If
    
```

```

indexCol = 0

xlSheet.Range("A" & CStr(indexRow)) = rstFSS("sign_fac")
xlSheet.Range("B" & CStr(indexRow)) = rstFSS("id_spec") & " " &
rstFSS("name_spec")
xlSheet.Range("C" & CStr(indexRow)) = rstFSS("name_specialization")

rstCourse.MoveFirst
Do While Not rstCourse.EOF

    Set rstGroup = CurrentDb.OpenRecordset("SELECT
[_Group].id_specialization, [_Group].course, [_Group].formnavch, [_Group].id_specialization,
[_Group].id_group, [_Group].code_group " & _
                                                "FROM _Group " & _
                                                "WHERE
(([_Group].id_specialization)=" & rstFSS("id_specialization") & ") " & _
                                                " AND ([_Group].course)=" & _
rstCourse("course") & """) " & _
                                                " AND ([_Group].formnavch)=" & _
rstFormnavch("id_formnavch") & ") " & _
                                                " AND ([_Group].code_group)
Like "*"~*");")

    If (Not rstGroup.EOF) Then
        rstGroup.MoveLast
        size_rstGroup = rstGroup.RecordCount
        it = 1
        rstGroup.MoveFirst
    End If
    Do While Not rstGroup.EOF
        Set rstStud = CurrentDb.OpenRecordset("SELECT
Count([_Student].id_stud) As Count_stud, isForeigner, type_financing " & _
                                                "FROM _Student " & _
                                                "WHERE ((id_group)=" & _
rstGroup("id_group") & ") " & _
                                                "GROUP BY isForeigner,
type_financing;")
                                                "'AND
((type_financing)=" & rstTypeFin("type_financing") & "")) "
        If (Not rstStud.EOF) Then
            sum_group_spec = sum_group_spec + 1
            str_groups = str_groups & rstGroup("code_group")
        End If

        Do While Not rstStud.EOF
            If (rstStud("isForeigner") = True) Then
                If (rstStud("type_financing") = "B") Then
                    If (rstStud("Count_stud") <> 0) Then
                        str_sums_inoz_B = str_sums_inoz_B &
rstStud("Count_stud")
                        sum_stud_B_inoz_spec = sum_stud_B_inoz_spec +
rstStud("Count_stud")
                    Else
                        str_sums_inoz_B = str_sums_inoz_B
                    End If
                ElseIf (rstStud("type_financing") = "K") Then
                    If (rstStud("Count_stud") <> 0) Then
                        str_sums_inoz_K = str_sums_inoz_K &
rstStud("Count_stud")
                        sum_stud_K_inoz_spec = sum_stud_K_inoz_spec +
rstStud("Count_stud")
                    Else
                        str_sums_inoz_K = str_sums_inoz_K
                    End If
                End If
            ElseIf (rstStud("isForeigner") = False) Then
                If (rstStud("type_financing") = "B") Then
                    If (rstStud("Count_stud") <> 0) Then
                        str_sums_B = str_sums_B & rstStud("Count_stud")
                        sum_stud_B_spec = sum_stud_B_spec +
rstStud("Count_stud")
                    Else
                        str_sums_B = str_sums_B
                    End If
                ElseIf (rstStud("type_financing") = "K") Then
                    If (rstStud("Count_stud") <> 0) Then
                        str_sums_K = str_sums_K & rstStud("Count_stud")

```

```

sum_stud_K_spec = sum_stud_K_spec +
rstStud("Count_stud")
Else
    str_sums_K = str_sums_K
End If

End If
End If

rstStud.MoveNext
Loop

rstGroup.MoveNext
If (it < size_rstGroup) Then
    str_groups = str_groups & Chr(10)
    str_sums_B = str_sums_B & Chr(10)
    str_sums_K = str_sums_K & Chr(10)
    str_sums_inoz_B = str_sums_inoz_B & Chr(10)
    str_sums_inoz_K = str_sums_inoz_K & Chr(10)
    it = it + 1
End If
Loop

'ИНОЗЕМЦИ
xlSheet.Range(masCol(indexCol) & CStr(indexRow)) = str_groups
xlSheet.Range(masCol(indexCol + 1) & CStr(indexRow)) = str_sums_B
xlSheet.Range(masCol(indexCol + 2) & CStr(indexRow)) = str_sums_K
'ИНОЗЕМЦИ

xlSheet.Range(masCol(indexCol + 3) & CStr(indexRow)) = str_sums_inoz_B
xlSheet.Range(masCol(indexCol + 4) & CStr(indexRow)) = str_sums_inoz_K

If (sum_group_spec <> 0) Then
    xlSheet.Range(masCol(indexCol) & CStr(indexRow + 1)) =
sum_group_spec & " "
End If
If (sum_stud_B_spec <> 0) Then
    xlSheet.Range(masCol(indexCol + 1) & CStr(indexRow + 1)) =
sum_stud_B_spec & " "
End If
If (sum_stud_K_spec <> 0) Then
    xlSheet.Range(masCol(indexCol + 2) & CStr(indexRow + 1)) =
sum_stud_K_spec & " "
End If
'ИНОЗЕМЦИ
If (sum_stud_B_inoz_spec <> 0) Then
    xlSheet.Range(masCol(indexCol + 3) & CStr(indexRow + 1)) =
sum_stud_B_inoz_spec & " "
End If
If (sum_stud_K_inoz_spec <> 0) Then
    xlSheet.Range(masCol(indexCol + 4) & CStr(indexRow + 1)) =
sum_stud_K_inoz_spec & " "
End If

sum_group_fac = sum_group_fac + sum_group_spec
sum_stud_B_fac = sum_stud_B_fac + sum_stud_B_spec
sum_stud_K_fac = sum_stud_K_fac + sum_stud_K_spec
sum_stud_B_inoz_fac = sum_stud_B_inoz_fac + sum_stud_B_inoz_spec
sum_stud_K_inoz_fac = sum_stud_K_inoz_fac + sum_stud_K_inoz_spec

str_groups = ""
str_sums_B = ""
str_sums_K = ""
str_sums_inoz_B = ""
str_sums_inoz_K = ""

sum_group_spec = 0
sum_stud_B_spec = 0
sum_stud_K_spec = 0
sum_stud_B_inoz_spec = 0
sum_stud_K_inoz_spec = 0

indexCol = indexCol + 5
rstCourse.MoveNext
Loop

indexRow = indexRow + 2
rstFSS.MoveNext
Loop

```

```

xlSheet.Range("B" & CStr(indexRow)) = "Студентів на " & saveSignFac
indexRow = indexRow + 1

xlSheet.Range("B" & CStr(indexRow)) = "Усього по університету"

xlApp.Run "DrawTable", xlSheet.name

rstFormnavch.MoveNext
Loop

'Save and close file report
xlBook.Save
xlApp.Quit

MsgBox "Генерація звіту <Рух контингенту> завершено." & vbNewLine & CStr(start_date)
& "-" & CStr(Time), vbInformation, "Генерація звіту"
Message "Генерація звіту <Рух контингенту> завершено." & vbNewLine &
CStr(start_date) & "-" & CStr(Time)
StopLine:
End Sub

```

A.6 Функція для генерації звіту «Вакансії»

```

Private Sub ButtonVacancy_Click()
    Dim start_date As Date
    start_date = Time

    Dim excelFileTemplate As String, excelFileReport As String, pathProject As String,
Path As String
    Dim rst As DAO.Recordset, rstSS As DAO.Recordset
    Dim rstStudents As DAO.Recordset
    Dim rstFormnavch As DAO.Recordset
    Dim rstCourse As DAO.Recordset
    Dim indexRow As Integer

    Dim masIndexColDZ() As Byte
    Dim masIndexColFact() As Byte
    Dim masIndexColVac() As Byte

    Dim dbs As DAO.Database
    Dim objField As DAO.Field

    Dim masCharColumn() As String
    Dim indexMasCharColumn As Byte

    masCharColumn = Split("C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD
AE AF AG AH AI AJ AK AL AM AN AO AP AQ AR AS AT AU AV AW AX AY AZ BA BB BC BD BE BF BG BH BI BJ
BK BL BM BN BO BP BQ BR BR BS BT BU BV BW BX BY BZ CA CB CC CD CE CF CG CH CI CJ CK CL CM CN CO
CP CQ CR CS CT CU CV CW CX CY CZ DA DB DC DD DE DF DG DH DI DJ DK DL DM DN DO DP DQ DR DS DT DU
DV DW DX DY DZ")
    indexMasCharColumn = 0

    'Визначення розміщення проекту
    pathProject = Application.CurrentProject.Path

    Path = pathProject + "\" & Format(Date, "dd.mm.yy")
    If Dir(Path, vbDirectory) = "" Then
        Mkdir Path
    End If

    excelFileTemplate = pathProject + "\Template\Vacancy.xlsm"
    excelFileReport = Path + "\Вакансії_" & Format(Now, "dd/mm/yyyy hh/mm/ss") & ".xlsm"
    Message "Створення звіту " & excelFileReport

    'Copy file template
    FileCopy excelFileTemplate, excelFileReport

    'Open file report for making changes
    Set xlApp = CreateObject("Excel.Application")
    Set xlBook = xlApp.Workbooks.Open(excelFileReport)
    Set rstFormnavch = CurrentDb.OpenRecordset("SELECT id_formnavch, formnavch, code " &
-
"FROM s_formnavch " & _
"WHERE (((id_formnavch)<>0));")
    Set rstCourse = CurrentDb.OpenRecordset("SELECT id_course, course " & _
"FROM s_course " & _
"ORDER BY id_course;")

```

```

Set rstSS = CurrentDb.OpenRecordset("SELECT s_Specialty.id_spec,
s_Specialty.name_spec, s_Specialization.id_specialization, s_Specialization.name_specialization "
& _
"FROM s_Specialty INNER JOIN s_Specialization ON
s_Specialty.id_spec = s_Specialization.id_spec " & _
"WHERE ((s_Specialization.bool_activity)=True)
" & _
" And ((s_Specialty.id_spec)<>"000") " & _
"ORDER BY s_Specialty.id_spec;")

'Відкриття форми бюджету для перенесення заголовків
DoCmd.OpenForm "_FormBudget"
xlApp.Visible = True

Do While Not rstFormnavch.EOF
Set xlSheet = xlBook.WorkSheets(CStr(rstFormnavch("formnavch")))
string_fields = ""
Set dbs = CurrentDb
For Each objField In dbs.TableDefs("b_SplitBudget").Fields
If (Not objField.name Like "id*" And objField.name Like "*" &
rstFormnavch("code") & "*") Then
string_fields = string_fields & objField.name & ", "
End If
Next
string_fields = Left(string_fields, Len(string_fields) - 2)

.....'Шапка'.....
.....
xlSheet.Range("A4") = "Спеціальність"
xlSheet.Range("B4") = "Освітня програма"

saveCourse1 = "0к"
saveCourse2 = "0к"
saveCourse3 = "0к"
indexMasCharColumn = 0

Set dbs = CurrentDb
For Each objField In dbs.TableDefs("b_SplitBudget").Fields
objName = ""
If (objField.name Like "d*" Or objField.name Like "z*") Then
objName = objField.name

ElseIf (objField.name Like "k*" Or objField.name Like "v*" Or objField.name
Like "l*") Then
objName = Right(objField.name, Len(objField.name) - 1)
Else
GoTo SkipLine
End If

If (objName <> "") Then
Set rstSum = CurrentDb.OpenRecordset("SELECT Sum(" & objName & ") As
Sum_" & objName & ", Sum(k" & objName & ") As Sum_k" & objName & " " & _
"FROM b_SplitBudget;")

If (rstSum("Sum_" & objName) = 0 And rstSum("Sum_k" & objName) = 0) Then
GoTo SkipLine
End If
End If

'He id_specialization
If (Not objField.name Like "id*") Then
'Витягування курсу за id для бюджету або для іншого (позиція id курсу
відрізняється)

If (objField.name Like rstFormnavch("code") & "*") Then
Set rst = CurrentDb.OpenRecordset("SELECT course " & _
"FROM s_course " & _
"WHERE (((id_course)=" &
CInt(Mid(objField.name, 2, 1)) & "));")
ElseIf (objField.name Like "*" & rstFormnavch("code") & "*") Then
Set rst = CurrentDb.OpenRecordset("SELECT course " & _
"FROM s_course " & _
"WHERE (((id_course)=" &
CInt(Mid(objField.name, 3, 1)) & "));")
End If

'Заголовок із форми
labelCaption = [Forms]![_FormBudget].Controls("lb" &
Right(objField.name, 4)).Caption

```

```

'Для бюджету
If (objField.name Like rstFormnavch("code") & "**") Then
    'Для "Всього"
    If (saveCourse1 <> Mid(objField.name, 2, 1)) Then
        saveCourse1 = Mid(objField.name, 2, 1)
        'Курс
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "4") =
rst("course") & " курс"
        'Всього ДЗ
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "5") = "ДЗ" &
vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
        'Колонка "Всього"
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") =
"Всього" & vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
        indexMasCharColumn = indexMasCharColumn + 1
    End If
    'Заголовок поле
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "2") = "1b" &
Right(objField.name, 4)
    'Заголовок текст
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "3") =
labelCaption
    'Курс
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "4") =
rst("course") & " курс"
    'ДЗ
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "5") = "ДЗ" &
vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
    'Колонка
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") =
objField.name
    'Сума
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "1") =
rstSum("Sum_" & objName)

    indexMasCharColumn = indexMasCharColumn + 1

'Для контингенту
ElseIf (objField.name Like "k" & rstFormnavch("code") & "**") Then
    'Для "Всього"
    If (saveCourse2 <> Mid(objField.name, 3, 1)) Then
        saveCourse2 = Mid(objField.name, 3, 1)

        'Курс
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "4") =
rst("course") & " курс"
        'Всього факт
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "5") = "Факт"
& vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
        'Колонка "Всього"
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") =
"Всього" & vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
        indexMasCharColumn = indexMasCharColumn + 1
    End If
    'Заголовок поле
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "2") = "1k" &
Right(objField.name, 4)
    'Заголовок текст
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "3") =
labelCaption
    'Курс
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "4") =
rst("course") & " курс"
    'ДЗ
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "5") = "Факт" &
vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
    'Колонка
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") =
objField.name
    'Сума
    xlSheet.Range(masCharColumn(indexMasCharColumn) & "1") =
rstSum("Sum_k" & objName)

    indexMasCharColumn = indexMasCharColumn + 1

'Для вакансій
ElseIf (objField.name Like "v" & rstFormnavch("code") & "**") Then
    'Для "Всього"

```

```

        If (saveCourse3 <> Mid(objField.name, 3, 1)) Then
            saveCourse3 = Mid(objField.name, 3, 1)

            'Курс
            xlSheet.Range(masCharColumn(indexMasCharColumn) & "4") =
rst("course") & " курс"

            'Всього вакансія
            xlSheet.Range(masCharColumn(indexMasCharColumn) & "5") =
"Вакансії" & vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
            'Колонка "Всього"
            xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") =
"Всього" & vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
            indexMasCharColumn = indexMasCharColumn + 1
        End If
        'Заголовок поле
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "2") = "lv" &
Right(objField.name, 4)

        'Заголовок текст
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "3") =
labelCaption

        'Курс
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "4") =
rst("course") & " курс"

        'ДЗ
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "5") = "Вакансії"
& vbNewLine & Mid(currentYear, 1, 2) & Mid(labelCaption, Len(labelCaption) - 1, 2)
        'Колонка
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") =
objField.name

        'Сума
        xlSheet.Range(masCharColumn(indexMasCharColumn) & "1") =
rstSum("Sum_" & objName) - rstSum("Sum_k" & objName)

        indexMasCharColumn = indexMasCharColumn + 1
    End If
End If
Skipline:
Next

.....'Шапка'.....
.....

        rstSS.MoveFirst
        indexRow = 7
        Do While Not rstSS.EOF
            indexMasCharColumn = 0
            xlSheet.Range("A" & CStr(indexRow)) = rstSS("id_spec") & " " &
rstSS("name_spec")
            xlSheet.Range("B" & CStr(indexRow)) = rstSS("name_specialization")

            Set rstBudget = CurrentDb.OpenRecordset("SELECT " & string_fields & " " & _
"FROM b_SplitBudget " & _
"WHERE ((id_specialization)=" &
rstSS("id_specialization") & "));")
            Do While (Not rstBudget.EOF And
xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") <> "")
                If (Not xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") Like
"Всього*") Then
                    xlSheet.Range(masCharColumn(indexMasCharColumn) & CStr(indexRow)) =
rstBudget(xlSheet.Range(masCharColumn(indexMasCharColumn) & "6"))
                End If
                indexMasCharColumn = indexMasCharColumn + 1
            Loop

            indexRow = indexRow + 1
            rstSS.MoveNext
        Loop

        Debug.Print indexRow

.....'Шапка'.....
.....

        xlSheet.Range("A" & CStr(indexRow)) = "Всього:"
        xlSheet.Range("A" & CStr(indexRow) & ":B" & CStr(indexRow)).Merge
        indexMasCharColumn = 0
        Do While (xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") <> "")

```

```

        If (Not xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") Like
"Всього*") Then
            xlSheet.Range(masCharColumn(indexMasCharColumn) & "6") =
xlSheet.Range(masCharColumn(indexMasCharColumn) & "3")
            xlSheet.Range(masCharColumn(indexMasCharColumn) & "2") = ""
            xlSheet.Range(masCharColumn(indexMasCharColumn) & "3") = ""
        End If
        'Перенесення суми
        xlSheet.Range(masCharColumn(indexMasCharColumn) & CStr(indexRow)) =
xlSheet.Range(masCharColumn(indexMasCharColumn) & "1")

        indexMasCharColumn = indexMasCharColumn + 1
    Loop

.....'Шапка'.....
.....

        xlApp.Run "Painting", rstFormnavch("formnavch"), indexRow

        rstFormnavch.MoveNext
    Loop

    'Save and close file report
    xlBook.Save
    xlApp.Quit

    MsgBox "Генерація звіту <Вакансії> завершено." & vbNewLine & CStr(start_date) & "-"
& CStr(Time), vbInformation, "Генерація звіту"
    Message "Генерація звіту <Вакансії> завершено." & vbNewLine & CStr(start_date) & "-"
& CStr(Time)

    DoCmd.Close acForm, "_FormBudget"
End Sub

```

А.7 Функції виконання наказів

```

Private Sub ButtonSave_Click()
    Dim rst As DAO.Recordset, rstTrueStud As DAO.Recordset, rstOrder As DAO.Recordset
    Dim id_order As Integer
    Dim msg, Style, Title, Help, Ctxt, Response, MyString
    Dim detail_str As String

    'Перевірка наявності всіх необхідних даних
    If (Me!FieldOrder = "" Or IsNull(Me!FieldOrder)) Then
        MsgBox "Шифр наказу не вказано!", vbCritical
        GoTo StopLine
    ElseIf (Me!FieldPoint = "" Or IsNull(Me!FieldPoint)) Then
        MsgBox "Пункт не вказано!", vbCritical
        GoTo StopLine
    ElseIf (Me!FieldDetail = "" Or IsNull(Me!FieldDetail)) Then
        MsgBox "Деталь не вказано!", vbCritical
        GoTo StopLine
    ElseIf (Me!FieldNewGroup.Visible And (Me!FieldNewGroup = "" Or
IsNull(Me!FieldNewGroup))) Then
        MsgBox "Групу не вказано!", vbCritical
        GoTo StopLine
    ElseIf (Me!FieldNewTypeFin.Visible And (Me!FieldNewTypeFin = "" Or
IsNull(Me!FieldNewTypeFin))) Then
        MsgBox "Тип фінансування не вказано!", vbCritical
        GoTo StopLine
    ElseIf (Me!FieldNewCourse.Visible And (Me!FieldNewCourse = "" Or
IsNull(Me!FieldNewCourse))) Then
        MsgBox "Курс не вказано!", vbCritical
        GoTo StopLine
    End If

    'Підтвердження наказу
    msg = "Зберегти пункт наказу " & Me!FieldOrder.column(1) & " ?" & vbNewLine & _
        "Пункт: " & IIf(Not IsNull(Me!FieldPoint.column(1)), Me!FieldPoint.column(1),
Me!FieldPoint) & vbNewLine & _
        "Причина: " & Me!FieldDetail & vbNewLine
    Set rstTrueStud = CurrentDb.OpenRecordset("SELECT [_Student].id_stud,
[_Student].last_name, [_Student].name_stud, [_Student].middle_name, [_Group].code_group,
[_Student].type_financing, [_Group].course, [_Group].formnavch, [_Group].id_specialization " & _
        "FROM _Group INNER JOIN _Student ON
[_Group].id_group = [_Student].id_group " & _
        "WHERE ((([_Student].bool)= " & True &
");");")

```

```

Do While Not rstTrueStud.EOF
    string_stud = string_stud & rstTrueStud("last_name") & " " &
rstTrueStud("name_stud") & " " & rstTrueStud("middle_name") & " | " &
rstTrueStud("type_financing") & " | " & rstTrueStud("code_group") & vbNewLine
    rstTrueStud.MoveNext
Loop
If (rstTrueStud.RecordCount <> 0) Then
    rstTrueStud.MoveFirst
    msg = msg & "До наказу будуть включені студенти: " & vbNewLine & _
        string_stud
Else
    msg = msg & "Студенти не обрані."
End If

msg = msg & vbNewLine

If (Me!FieldPoint = "Відрахування" Or Me!FieldPoint.column(1) = "Відрахування") Then
    msg = msg & vbNewLine & "Студенти втраять свої групи" 'et facti sunt manes

    ElseIf (Me!FieldPoint = "Переведено з інших ЗВО" Or Me!FieldPoint.column(1) =
"Переведено з інших ЗВО") Then
        msg = msg & vbNewLine & "Студенти отримають групу <" &
Me!FieldNewGroup.column(1) & "> і тип фінансування <" & Me!FieldNewTypeFin & ">"

    ElseIf (Me!FieldPoint = "Переведення" Or Me!FieldPoint.column(1) = "Переведення")
Then
        msg = msg & vbNewLine & "Студенти будуть переведені до групи <" &
Me!FieldNewGroup.column(1) & "> і отримають тип фінансування <" & Me!FieldNewTypeFin & ">"

    ElseIf (Me!FieldPoint = "Поновлення" Or Me!FieldPoint.column(1) = "Поновлення") Then
        msg = msg & vbNewLine & "Студенти отримають групу <" &
Me!FieldNewGroup.column(1) & "> і тип фінансування <" & Me!FieldNewTypeFin & ">"

    ElseIf (Me!FieldPoint = "Поновлення з числа відрахованих з інших ЗВО" Or
Me!FieldPoint.column(1) = "Поновлення з числа відрахованих з інших ЗВО") Then
        msg = msg & vbNewLine & "Студенти отримають групу <" &
Me!FieldNewGroup.column(1) & "> і тип фінансування <" & Me!FieldNewTypeFin & ">"

    ElseIf (Me!FieldPoint = "Академвідпустка" Or Me!FieldPoint.column(1) =
"Академвідпустка") Then
        msg = msg & vbNewLine & "Студенти отримають групу із приставкою <(ак)>" ' і тип
фінансування <" & Me!FieldNewTypeFin & ">"

    ElseIf (Me!FieldPoint = "Переведення групи" Or Me!FieldPoint.column(1) =
"Переведення групи") Then
        msg = msg & vbNewLine & "Студенти будуть переведені до групи <" &
Me!FieldNewGroup.column(1) & "> на " & Me!FieldNewCourse.column(1) & " курс"

    ElseIf (Me!FieldPoint = "Повернено з академвідпустки" Or Me!FieldPoint.column(1) =
"Повернено з академвідпустки") Then
        msg = msg & vbNewLine & "Студенти будуть повернені з академвідпустки до групи <"
& Me!FieldNewGroup.column(1) & "> і отримають тип фінансування <" & Me!FieldNewTypeFin & ">"

    ElseIf (Me!FieldPoint = "Розподіл студентів по групах" Or Me!FieldPoint.column(1) =
"Розподіл студентів по групах") Then
        msg = msg & vbNewLine & "Студенти будуть зараховані до групи <" &
Me!FieldNewGroup.column(1) & ">"
    End If

    string_stud = ""

    Style = vbYesNo + vbQuestion + vbDefaultButton2
    Title = "Наказ: " & Me!FieldOrder.column(1)
    Response = MsgBox(msg, Style, Title)
    If Response = vbYes Then

        'Збереження нового "Інші"
        'Перевірка на існування
        If (Me!FieldPoint <> "" And Not IsNull(Me!FieldPoint.column(0))) Then
            Set rst = CurrentDb.OpenRecordset("SELECT id " & _
                "FROM s_pointInfoAnother " & _
                "WHERE ((id_point)=" &
Me!FieldPoint.column(0) & ") " & _
                " AND ((another)=""" & Me!FieldAnother &
"""));"

            If (rst.EOF And Me!FieldAnother <> "" And Not IsNull(Me!FieldAnother)) Then
                CurrentDb.Execute "INSERT INTO s_pointInfoAnother (id_point, another) " & _
                    "VALUES (" & Me!FieldPoint.column(0) & ", " & "" &
Me!FieldAnother & """);"
            End If

```

```

End If
id_order = Me!FieldOrder.column(0)

'Оновлення запису, якщо ід пункту є у полі
If (Me!FieldIdPoint <> "" And Not IsNull(Me!FieldIdPoint)) Then
    'Виконання відповідної дії до наказу
    respond = DoAction

    'Зняття міток із студентів
    CurrentDb.Execute "UPDATE [_Student] " & _
        "SET [_Student].bool = " & False & " " & _
        "WHERE ((([_Student].bool)=" & True & "));"

'Введення нового пункту
ElseIf (Me!FieldIdPoint = "" Or IsNull(Me!FieldIdPoint)) Then

    With CurrentProject.Connection
        .Execute "INSERT INTO _OrderPoint (id_order, point, detail, another) " & _
            "VALUES ("" & id_order & "", "" & Me!FieldPoint.column(1) & "", "" & Me!FieldDetail & "", "" & Me!FieldAnother & "");"
        id_point = .Execute("SELECT @@identity") (0)
    End With

    Me!FieldIdPoint = id_point

    'Виконання відповідної дії до наказу
    respond = DoAction

    'Зняття міток із студентів
    CurrentDb.Execute "UPDATE _Student " & _
        "SET [_Student].bool = " & False & " " & _
        "WHERE ((([_Student].bool)=" & True & "));"

End If

Me!FieldIdPoint = ""
Me!FieldPoint = ""
Me!FieldDetail = ""
Me!FieldAnother = ""

Call ChangeOrder

Me![_FormOrderPoint].Requery
Me![_Student].Requery
Me![_OrderStudent].Requery
StopLine:

'Якщо збереження відхилено
Else

End If
End Sub

Private Function DoAction() As Boolean
    Dim rst As DAO.Recordset
    Dim msg, Style, Title, Help, Ctxt, Response, MyString

    Set rstTrueStud = CurrentDb.OpenRecordset("SELECT Id_Person, [_Student].id_stud,
[_Student].last_name, [_Student].name_stud, [_Student].middle_name, [_Group].code_group,
[_Student].type_financing, [_Group].course, [_Group].formnavch, [_Group].id_specialization,
[_Group].year_group, [_Group].type_group, [_Group].id_group " & _
        "FROM _Group INNER JOIN _Student ON
[_Group].id_group = [_Student].id_group " & _
        "WHERE ((([_Student].bool)=" & True &
    ");");

    '.....'Відрахування (втрати групи)
    'Збереження студента із його минулою групою і типом фінансування
    If (Me!FieldPoint = "Відрахування" Or Me!FieldPoint.column(1) = "Відрахування") Then
        'Внесення студентів у наказ
        Do While Not rstTrueStud.EOF
            'Перевірка відсутності протоколу у студента
            Set rstVerProt = CurrentDb.OpenRecordset("SELECT protocol_code,
protocol_date " & _
                "FROM _Student " & _
                "WHERE ((id_stud)=" & _
                    " AND ((protocol_code)<>0) " & _
                    " AND Not
IsNull((protocol_code));")
            If (Not rstVerProt.EOF) Then

```

```

        rez = MsgBox("УВАГА! Відрахування студента із протоколом." & vbNewLine &
"Студент(-ка): " & rstTrueStud("last_name") & " " & rstTrueStud("name_stud") & " " &
rstTrueStud("middle_name") & vbNewLine & "Протокол: " & rstVerProt("protocol_code") & " " &
rstVerProt("protocol_date") & vbNewLine & "Продовжити?", vbCritical + vbYesNo, "Відрахування
студента із введеним протоколом")
        If (rez = vbYes) Then
            'Код продовжується
        ElseIf (rez = vbNo) Then
            'Пропуск студента
            GoTo SkipLine
        End If
    End If

    CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud, last_name,
name_stud, middle_name, code_group, type_financing, id_specialization, course, formnavch,
id_group) " & _
        "VALUES ('" & Me!FieldIdPoint & "', '" &
rstTrueStud("id_stud") & "', '" & rstTrueStud("last_name") & "'", '" & rstTrueStud("name_stud")
& "'", '" & rstTrueStud("middle_name") & "', '" & rstTrueStud("code_group") & "', '" &
rstTrueStud("type_financing") & "', '" & rstTrueStud("id_specialization") & "', '" &
rstTrueStud("course") & "'", '" & rstTrueStud("formnavch") & "', '" & rstTrueStud("id_group") &
"');"

    NotAktiveGroup = Replace(rstTrueStud("code_group"), "-", " ")
    Set rstGroup = CurrentDb.OpenRecordset("SELECT id_group " & _
        "FROM _Group " & _
        "WHERE (((code_group)='" &
NotAktiveGroup & "')) " & _
        " And ((course)='" &
rstTrueStud("course") & "'));")
    'Створення неактивної групи, якщо такої немає
    If (rstGroup.EOF) Then
        'Set rstGroup = CurrentDb.OpenRecordset("SELECT code_group, course,
formnavch, id_specialization, year_group, type_group " & _
        "FROM [_Group] " & _
        "WHERE (((code_group)='" &
rstTrueStud("code_group") & "'));")
        With CurrentProject.Connection
            .Execute "INSERT INTO _Group (code_group, course, formnavch,
id_specialization, year_group, type_group) " & _
                "VALUES ('" & NotAktiveGroup & "'", '" &
rstTrueStud("course") & "'", '" & rstTrueStud("formnavch") & "'", '" &
rstTrueStud("id_specialization") & "', '" & rstTrueStud("year_group") & "'", '" &
rstTrueStud("type_group") & "');"
            id_group = .Execute("SELECT @@identity")(0)
        End With
    Else
        id_group = rstGroup("id_group")
    End If

    'Встановлення примарної групи студенту
    CurrentDb.Execute "UPDATE [_Student] " & _
        "SET id_group = '" & id_group & "' " &
        "WHERE (((id_stud) = '" & rstTrueStud("id_stud") & "')";

    'Пропуск для тих, у кого є протокол
SkipLine:
    rstTrueStud.MoveNext
    Loop
    Me.Requery
    'Переведено з інших ЗВО
    'Збереження студента у наказі вже з новою групою
    'Збереження виконується або вибраних у списку або формі
    ElseIf (Me!FieldPoint = "Переведено з інших ЗВО" Or Me!FieldPoint.column(1) =
"Переведено з інших ЗВО") Then
        'Пошук потрібної групи
        If ((Me!FieldNewGroup <> "" And Not IsNull(Me!FieldNewGroup)) Or
(Me!FieldNewTypeFin <> "" And IsNull(Me!FieldNewTypeFin))) Then
            Set rstGroup = CurrentDb.OpenRecordset("SELECT code_group,
id_specialization, formnavch, course " & _
                "FROM _Group " & _
                "WHERE (((id_group)='" &
Me!FieldNewGroup & "');"
            'Внесення студентів у наказ
            Do While Not rstTrueStud.EOF
                If (VerifyStudentRequest(rstTrueStud("id_stud"), Me!FieldNewGroup)) Then
                    CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud,
last_name, name_stud, middle_name, code_group, type_financing, id_specialization, course,
formnavch, id_group) " & _

```

```

VALUES ('" & Me!FieldIdPoint & "', '" &
rstTrueStud("id_stud") & "', "" & rstTrueStud("last_name") & "", "" & rstTrueStud("name_stud")
& "", '" & rstTrueStud("middle_name") & "', '" & rstGroup("code_group") & "', '" &
Me!FieldNewTypeFin & "', '" & rstGroup("id_specialization") & "', "" & rstGroup("course") & "",
" & rstGroup("formnavch") & "', '" & Me!FieldNewGroup & "');
'Оновлення даних про студентів
CurrentDb.Execute "UPDATE _Student " & _
"SET id_group = "" & Me!FieldNewGroup & "",
type_financing = "" & Me!FieldNewTypeFin & "" " & _
"WHERE ((id_stud)=" & rstTrueStud("id_stud") &
");"

Else
    MsgBox rstTrueStud("last_name") & " " & rstTrueStud("name_stud") & "
" & rstTrueStud("middle_name") & " не включено в наказ."
End If

rstTrueStud.MoveNext
Loop

Call NewStudent("Переведено з інших ЗВО")
Else
    MsgBox "Групу або тип фінансування не обрано!"
End If

'Переведення
'Переведення з інших форм навчання
'Переведення на інші форми навчання
'Студенти прив'язуються до двох пунктів
ElseIf (Me!FieldPoint = "Переведення" Or Me!FieldPoint.column(1) = "Переведення")
Then
    If ((Me!FieldNewGroup <> "" And Not IsNull(Me!FieldNewGroup)) Or
(Me!FieldNewTypeFin <> "" And IsNull(Me!FieldNewTypeFin))) Then
        Set rstGroup = CurrentDb.OpenRecordset("SELECT code_group,
id_specialization, formnavch, course " & _
"FROM _Group " & _
"WHERE ((id_group)=" &
Me!FieldNewGroup & "));")
        'Збереження інформації про студентів
Do While Not rstTrueStud.EOF
    If (VerifyStudentRequest(rstTrueStud("id_stud"), Me!FieldNewGroup)) Then
        CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud,
last_name, name_stud, middle_name, code_group, type_financing, id_specialization, course,
formnavch, id_group) " & _
"VALUES ('" & Me!FieldIdPoint & "', '" &
rstTrueStud("id_stud") & "', "" & rstTrueStud("last_name") & "", "" & rstTrueStud("name_stud")
& "", '" & rstTrueStud("middle_name") & "', '" & rstGroup("code_group") & "', '" &
Me!FieldNewTypeFin & "', '" & rstGroup("id_specialization") & "', "" & rstGroup("course") & "",
" & rstGroup("formnavch") & "', '" & Me!FieldNewGroup & "');
CurrentDb.Execute "UPDATE _Student " & _
"SET id_group = '" & Me!FieldNewGroup & "',
type_financing = "" & Me!FieldNewTypeFin & "" " & _
"WHERE ((id_stud)=" & rstTrueStud("id_stud") &
");"

        'Якщо це отримання громадянства іноземцем, то знімається ще мітка
іноземця

If (Me!FieldAnother = "отримання громадянства") Then
    CurrentDb.Execute "UPDATE _Student " & _
"SET isForeigner = False " & _
"WHERE ((id_stud)=" & rstTrueStud("id_stud")
& "));"

    End If
Else
    MsgBox rstTrueStud("last_name") & " " & rstTrueStud("name_stud") & "
" & rstTrueStud("middle_name") & " не включено в наказ."
End If

rstTrueStud.MoveNext
Loop

DoAction = True
Else
    MsgBox "Група, або тип фінансування не обрано!", vbCritical, "Переведення"
DoAction = False
End If

'Поновлення
ElseIf (Me!FieldPoint = "Поновлення" Or Me!FieldPoint.column(1) = "Поновлення") Then
    If ((Me!FieldNewGroup <> "" And Not IsNull(Me!FieldNewGroup)) Or
(Me!FieldNewTypeFin <> "" And IsNull(Me!FieldNewTypeFin))) Then
        'Пошук групи

```

```

        Set rstGroup = CurrentDb.OpenRecordset("SELECT code_group,
id_specialization, formnavch, course " & _
                                                "FROM _Group " & _
                                                "WHERE (((id_group)=" &
Me!FieldNewGroup & "));")
        'Збереження інформації про студентів
        Do While Not rstTrueStud.EOF
            If (VerifyStudentRequest(rstTrueStud("id_stud"), Me!FieldNewGroup)) Then
                CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud,
last_name, name_stud, middle_name, code_group, type_financing, id_specialization, course,
formnavch, id_group) " & _
                                                "VALUES ('" & Me!FieldIdPoint & "', '" &
rstTrueStud("id_stud") & "', """" & rstTrueStud("last_name") & """, """" & rstTrueStud("name_stud")
& """, '" & rstTrueStud("middle_name") & "', '" & rstGroup("code_group") & "', '" &
Me!FieldNewTypeFin & "', '" & rstGroup("id_specialization") & "', """" & rstGroup("course") & """,
" & rstGroup("formnavch") & "', '" & Me!FieldNewGroup & "');"
                CurrentDb.Execute "UPDATE _Student " & _
                                                "SET id_group = '" & Me!FieldNewGroup & "',
type_financing = """" & Me!FieldNewTypeFin & """" " & _
                                                "WHERE (((id_stud)=" & rstTrueStud("id_stud") &
"););"
            Else
                MsgBox rstTrueStud("last_name") & " " & rstTrueStud("name_stud") & "
" & rstTrueStud("middle_name") & " не включено в наказ."
                End If

                rstTrueStud.MoveNext
        Loop

        DoAction = True
    Else
        MsgBox "Групу, або тип фінансування не обрано!"
        DoAction = False
    End If

    Call NewStudent("Поновлення")

    'Поновлення з числа відрахованих з інших ЗВО
    ElseIf (Me!FieldPoint = "Поновлено з числа відрахованих з інших ЗВО" Or
Me!FieldPoint.column(1) = "Поновлено з числа відрахованих з інших ЗВО") Then
        'Пошук потрібної групи
        If ((Me!FieldNewGroup <> "" And Not IsNull(Me!FieldNewGroup)) Or
(Me!FieldNewTypeFin <> "" And IsNull(Me!FieldNewTypeFin))) Then
            Set rstGroup = CurrentDb.OpenRecordset("SELECT code_group,
id_specialization, formnavch, course " & _
                                                "FROM _Group " & _
                                                "WHERE (((id_group)=" &
Me!FieldNewGroup & "));")
            'Внесення студентів у наказ
            Do While Not rstTrueStud.EOF
                If (VerifyStudentRequest(rstTrueStud("id_stud"), Me!FieldNewGroup)) Then
                    CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud,
last_name, name_stud, middle_name, code_group, type_financing, id_specialization, course,
formnavch, id_group) " & _
                                                "VALUES ('" & Me!FieldIdPoint & "', '" &
rstTrueStud("id_stud") & "', """" & rstTrueStud("last_name") & """, """" & rstTrueStud("name_stud")
& """, '" & rstTrueStud("middle_name") & "', '" & rstGroup("code_group") & "', '" &
Me!FieldNewTypeFin & "', '" & rstGroup("id_specialization") & "', """" & rstGroup("course") & """,
" & rstGroup("formnavch") & "', '" & Me!FieldNewGroup & "');"
                    'Оновлення даних про студентів
                    CurrentDb.Execute "UPDATE _Student " & _
                                                "SET id_group = '" & Me!FieldNewGroup & "',
type_financing = """" & Me!FieldNewTypeFin & """" " & _
                                                "WHERE (((id_stud)=" & rstTrueStud("id_stud") &
"););"
                Else
                    MsgBox rstTrueStud("last_name") & " " & rstTrueStud("name_stud") & "
" & rstTrueStud("middle_name") & " не включено в наказ."
                    End If

                    rstTrueStud.MoveNext
            Loop

            Call NewStudent("Поновлення з числа відрахованих з інших ЗВО")
        Else
            MsgBox "Групу або тип фінансування не обрано!"
        End If
    End If

```

```

'Академвідпустка (втрата бюджетного місця) 11.11.2020 втрата бюджетного місця
знімається
'Створення нової групи, такої ж самої як поточна, але із додавання частки "(ак)" у
кінець
ElseIf (Me!FieldPoint = "Академвідпустка" Or Me!FieldPoint.column(1) =
"Академвідпустка") Then
    Do While Not rstTrueStud.EOF
        If (InStr(rstTrueStud("code_group"), "(ак)")) Then
            'Приставка є
            codeGroup = rstTrueStud("code_group")
        Else
            'Приставки немає
            codeGroup = rstTrueStud("code_group") & "(ак)"
        End If

        'Внесення студента до наказу
        CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud, last_name,
name_stud, middle_name, code_group, type_financing, id_specialization, course, formnavch,
id_group) " & _
            "VALUES ('" & Me!FieldIdPoint & "', '" &
rstTrueStud("id_stud") & "', '" & rstTrueStud("last_name") & "'", '" & rstTrueStud("name_stud")
& "'", '" & rstTrueStud("middle_name") & "'", '" & rstTrueStud("code_group") & "'", '" &
rstTrueStud("type_financing") & "'", '" & rstTrueStud("id_specialization") & "', '" &
rstTrueStud("course") & "'", '" & rstTrueStud("formnavch") & "', '" & rstTrueStud("id_group") &
"');"

        Set rstGroup = CurrentDb.OpenRecordset("SELECT id_group " & _
            "FROM _Group " & _
            "WHERE (((code_group)='" & codeGroup &
            " And ((course)='" &
rstTrueStud("course") & "')));")
        'Збереження нової групи для академвідпустки, якщо такої немає
        If (rstGroup.EOF) Then
            With CurrentProject.Connection
                .Execute "INSERT INTO _Group (code_group, id_specialization, course,
formnavch, year_group, type_group) " & _
                    "VALUES ('" & codeGroup & "'", '" &
rstTrueStud("id_specialization") & "', '" & rstTrueStud("course") & "'", '" &
rstTrueStud("formnavch") & "', '" & rstTrueStud("year_group") & "'", '" &
rstTrueStud("type_group") & "');"
                id_group = .Execute("SELECT @@identity")(0)
            End With
        Else
            id_group = rstGroup("id_group")
        End If
        'Оновлення даних про студента. Бюджетна форма навчання більше не втращається
        'CurrentDb.Execute "UPDATE _Student " & _
            "SET id_group = '" & id_group & "', type_financing = "K"
" & _
            "WHERE (((id_stud)='" & rstTrueStud("id_stud") & "'));"
        CurrentDb.Execute "UPDATE _Student " & _
            "SET id_group = '" & id_group & "' " & _
            "WHERE (((id_stud)='" & rstTrueStud("id_stud") & "'));"
        rstTrueStud.MoveNext
    Loop

    'Переведення групи
    ElseIf (Me!FieldPoint = "Переведення групи" Or Me!FieldPoint.column(1) =
"Переведення групи") Then
        Do While Not rstTrueStud.EOF
            Set rstGroup = CurrentDb.OpenRecordset("SELECT id_group " & _
                "FROM _Group " & _
                "WHERE (((code_group)='" &
                " And ((course)='" &
Me!FieldNewGroup.column(1) & "')));")
            Me!FieldNewCourse.column(1) & "));")
            'Збереження нової групи із новим курсом, якщо такої немає
            If (rstGroup.EOF) Then
                With CurrentProject.Connection
                    .Execute "INSERT INTO _Group (code_group, id_specialization, course,
formnavch, year_group, type_group) " & _
                        "VALUES ('" & Me!FieldNewGroup.column(1) & "'", '" &
rstTrueStud("id_specialization") & "', '" & Me!FieldNewCourse.column(1) & "'", '" &
rstTrueStud("formnavch") & "', '" & rstTrueStud("year_group") & "'", '" &
rstTrueStud("type_group") & "');"
                    id_group = .Execute("SELECT @@identity")(0)
                End With
            Else
                id_group = rstGroup("id_group")
            End If
        End With
    End If
End If

```

```

End If

'Дані нової групи студента
Set rstGroup = CurrentDb.OpenRecordset("SELECT id_group, id_specialization,
formnavch, course, code_group " & _
"FROM _Group " & _
"WHERE ((id_group)=" & id_group &
");");

'Внесення студента до наказу
CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud, last_name,
name_stud, middle_name, code_group, type_financing, id_specialization, course, formnavch,
id_group) " & _
"VALUES ('" & Me!FieldIdPoint & "'", '" &
rstTrueStud("id_stud") & "'", "" & rstTrueStud("last_name") & "", "" & rstTrueStud("name_stud")
& "", "" & rstTrueStud("middle_name") & "", "" & Me!FieldNewGroup.column(1) & "", "" &
rstTrueStud("type_financing") & "", '" & rstGroup("id_specialization") & "'", "" &
rstGroup("course") & "", " & rstGroup("formnavch") & "'", '" & rstGroup("id_group") & "');"

'Оновлення даних про студента
CurrentDb.Execute "UPDATE _Student " & _
"SET id_group = '" & id_group & "'" & _
"WHERE ((id_stud)=" & rstTrueStud("id_stud") & "));";

rstTrueStud.MoveNext
Loop

'Розподіл студентів по групам
ElseIf (Me!FieldPoint = "Розподіл студентів по групам" Or Me!FieldPoint.column(1) =
"Розподіл студентів по групам") Then
Do While Not rstTrueStud.EOF
'Дані нової групи студента
Set rstGroup = CurrentDb.OpenRecordset("SELECT id_group,
s_Specialization.id_specialization, formnavch, course, code_group, num_specialization " & _
"FROM _Group INNER JOIN
s_Specialization ON [_Group].id_specialization = s_Specialization.id_specialization " & _
"WHERE ((id_group)=" &
Me!FieldNewGroup & "));");
Set rstEntrant = CurrentDb.OpenRecordset("SELECT formnavch, faculcod,
specnum, specnazv, hodat_curs, osv_stup " & _
"FROM b_Entrant " & _
"WHERE ((id_stud)=" &
rstTrueStud("id_stud") & "));");
boolGo = True
If (rstGroup("formnavch") <> rstEntrant("formnavch") Or _
IIf(osv_stup = "Бакалавр", rstGroup("course") <>
rstEntrant("hodat_curs"), False) Or _
rstGroup("num_specialization") <> rstEntrant("specnum")) Then
boolGo = False

Debug.Print IIf(osv_stup = "Бакалавр", rstGroup("course") <>
rstEntrant("hodat_curs"), True)

rez = MsgBox("Дані обраної групи не співпадають із даними з приймальною
комісією для абітупієнта: " & vbNewLine & _
rstTrueStud("Id_Person") & " | " & rstTrueStud("last_name") & "
" & rstTrueStud("name_stud") & " " & rstTrueStud("middle_name") & "." & vbNewLine & _
">>> Обрана група <<<" & vbNewLine & "Шифр освітньої
програми: " & rstGroup("num_specialization") & vbNewLine & "Форма навчання: " &
rstGroup("formnavch") & vbNewLine & "Курс: " & rstGroup("course") & vbNewLine & _
">>>Приймальна комісія<<<" & vbNewLine & "Шифр освітньої
програми: " & rstEntrant("specnum") & vbNewLine & "Форма навчання: " & rstEntrant("formnavch") &
vbNewLine & "Курс: " & rstEntrant("hodat_curs") & vbNewLine & "Продовжити (Да), Пропустити
(Нєт)", vbCritical + vbYesNo, "Розбіжності прийому і зарахування")
If (rez = vbYes) Then
boolGo = True
ElseIf (rez = vbNo) Then
boolGo = False
End If
End If

If (boolGo) Then
'Внесення студента до наказу
CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud,
last_name, name_stud, middle_name, code_group, type_financing, id_specialization, course,
formnavch, id_group) " & _
"VALUES ('" & Me!FieldIdPoint & "'", '" &
rstTrueStud("id_stud") & "'", "" & rstTrueStud("last_name") & "", "" & rstTrueStud("name_stud")
& "", "" & rstTrueStud("middle_name") & "", "" & rstGroup("code_group") & "", "" &
rstTrueStud("type_financing") & "", '" & rstGroup("id_specialization") & "'", "" &
rstGroup("course") & "", " & rstGroup("formnavch") & "'", '" & rstGroup("id_group") & "');"

```

```

        'Оновлення даних про студента
        CurrentDb.Execute "UPDATE _Student " & _
            "SET id_group = '" & Me!FieldNewGroup & "', bool =
False " & _
            "WHERE (((id_stud)=" & rstTrueStud("id_stud") & "));"

        'Оновлення групи у профілі moodle
        CurrentDb.Execute "UPDATE _StudentInMoodle " & _
            "SET code_group = '" & Me!FieldNewGroup & "' " & _
            "WHERE (((id_stud)=" & rstTrueStud("id_stud") & "));"

    End If

    rstTrueStud.MoveNext
    Loop

    'Повернено з академвідпустки
    ElseIf (Me!FieldPoint = "Повернено з академвідпустки" Or Me!FieldPoint.column(1) =
"Повернено з академвідпустки") Then
        Do While Not rstTrueStud.EOF
            'Дані нової групи студента
            Set rstGroup = CurrentDb.OpenRecordset("SELECT id_group, id_specialization,
formnavch, course, code_group " & _
                "FROM _Group " & _
                "WHERE (((id_group)=" &
Me!FieldNewGroup & "));")
            'Внесення студента до наказу
            CurrentDb.Execute "INSERT INTO _OrderStudent (id_point, id_stud, last_name,
name_stud, middle_name, code_group, type_financing, id_specialization, course, formnavch,
id_group) " & _
                "VALUES ('" & Me!FieldIdPoint & "', '" &
rstTrueStud("id_stud") & "', "" & rstTrueStud("last_name") & "", "" & rstTrueStud("name_stud")
& "", "" & rstTrueStud("middle_name") & "", "" & rstGroup("code_group") & "", "" &
Me!FieldNewTypeFin & "", '" & rstGroup("id_specialization") & "', "" & rstGroup("course") &
"", " & rstGroup("formnavch") & "', '" & rstGroup("id_group") & "');"

            'Оновлення даних про студента
            CurrentDb.Execute "UPDATE _Student " & _
                "SET id_group = '" & Me!FieldNewGroup & "', type_financing
= "" & Me!FieldNewTypeFin & "" " & _
                "WHERE (((id_stud)=" & rstTrueStud("id_stud") & "));"

            rstTrueStud.MoveNext
        Loop

    Else
        MsgBox "Пункт не визначено! Дане повідомлення виводиться у випадку, якщо один із
пунктів був змінений, або введений пункт нового типу для, якого не була ще прописана активна
дія.", vbInformation, "Активну дія для " & Me!FieldPoint & " " & Me!FieldPoint.column(1) & " не
знайдено."
    End If
End Function

```

A.8 Функція розбиття ліцензії

```

Public Function ParsingLicense(ByRef id_spec As String) As Integer()
    'Спосіб шифрування ліцензії
    'Formula: formnavch,course,license
    'Приклад: dl22

    Dim rst As DAO.Recordset
    Dim mas_lic(1, 11) As Integer
    Dim mas_str_lic() As String

    For i = 0 To 1
        For j = 0 To 5
            mas_lic(i, j) = 0
        Next
    Next

    Set rst = CurrentDb.OpenRecordset("SELECT lic " & _
        "FROM s_Specialty " & _
        "WHERE (((id_spec)=" & id_spec & "));")

    If (rst("lic") <> "" And Not IsNull(rst("lic"))) Then
        mas_str_lic = Split(rst("lic"))
        For Each str_lic In mas_str_lic
            If (str_lic Like "d*") Then
                course = CInt(Mid(str_lic, 2, 1))
            End If
        Next
    End If
End Function

```

```

        mas_lic(0, course - 1) = Right(str_lic, Len(str_lic) - 2)

        ElseIf (str_lic Like "z*") Then
            course = CInt(Mid(str_lic, 2, 1))

            mas_lic(1, course - 1) = Right(str_lic, Len(str_lic) - 2)

        ElseIf (str_lic = "") Then

        Else
            Debug.Print "? Невідомий символ на місці форми навчання при розбитті -"
& str_lic & "-"
        End If
    Next
End If

'Hove
Set rstRequest = CurrentDb.OpenRecordset("SELECT formnavch, type_financing, course "
& _
        "FROM s_Specialty INNER JOIN
([_StudentRequest] INNER JOIN s_Specialization ON [_StudentRequest].id_specialization =
s_Specialization.id_specialization) ON s_Specialty.id_spec = s_Specialization.id_spec " & _
        "WHERE ((s_Specialty.id_spec)="" &
id_spec & ""));")
    Do While Not rstRequest.EOF
        mas_lic(rstRequest("formnavch") - 1, 6 + rstRequest("course") - 1) =
mas_lic(rstRequest("formnavch") - 1, 6 + rstRequest("course") - 1) + 1

        rstRequest.MoveNext
    Loop

    '
        0 1 2 3 4 5 6 7 8 9 10 11
'0 - Денна , (1, 2, 3, 4, 1м ,2м, 1заявки, 2заявки, 3заявки, 4заявки, 5заявки,
бзаявки)
'1 - Заочна, (1, 2, 3, 4, 1м ,2м, 1заявки, 2заявки, 3заявки, 4заявки, 5заявки,
бзаявки)
    ParsingLicense = mas_lic
End Function

```

A.9 Функція розбиття держзамовлення

```

Public Function ParsingBudget(ByVal idSpecialization As Integer) As Integer()
    'Спосіб шифрування
    'Formula
    'Форма навчання + курс + [Р Y M] + рік(останні два числа) + "-" + бюджет
    'd1P19-22
    'z1P18-5

    Dim rstBudget As DAO.Recordset
    Dim masBudget(12, 4) As Integer

    For i = 0 To 11
        For j = 0 To 3
            masBudget(i, j) = 0
        Next
    Next

    '0 1к ПЗСО МС МС МС
    '1 2к ПЗСО МС МС МС
    '2 3к ПЗСО МС МС МС
    '3 4к ПЗСО МС МС МС
    '4 5к Мар
    '5 6к Мар
    If (idSpecialization = 34) Then
        Debug.Print
    End If

    Set rstBudget = CurrentDb.OpenRecordset("SELECT budget_" & currentYear & " " & _
        "FROM s_Specialization " & _
        "WHERE (((id_specialization)= " & idSpecialization
& "));")

    If Not rstBudget.EOF And Not IsNull(rstBudget("budget_" & currentYear)) Then
        masStringBudget = Split(rstBudget("budget_" & currentYear))

        For Each stringBudget In masStringBudget

            index = 0

```

```

course = CInt(Mid(stringBudget, 2, 1))
osv_stup = Mid(stringBudget, 3, 1)
year_b = Mid(currentYear, 1, 2) & Mid(stringBudget, 4, 2)

'Зміщення за формою навчання
If (stringBudget Like "d*") Then
    offset = 0
ElseIf (stringBudget Like "z*") Then
    offset = 6
End If

i = course - 1 + offset

If (osv_stup = "P") Then
    index = index + 0
ElseIf (osv_stup = "Y") Then
    If (CInt(year_b) = currentYear - course + 1) Then
        index = index + 1
    ElseIf (CInt(year_b) = currentYear - course + 2) Then
        index = index + 2
    ElseIf (CInt(year_b) = currentYear - course + 3) Then
        index = index + 3
    End If
ElseIf (osv_stup = "M") Then
    index = index + 0
End If

' 01 23 456 78910 11 12
'1[PY] 2[PY] 3[PYY] 4[PYYY] 5[M] 6[M]
masBudget(i, index) = CInt(Right(stringBudget, Len(stringBudget) - 6))
Next
End If

ParsingBudget = masBudget
End Function

```

ДОДАТОК Б
Слайди презентації

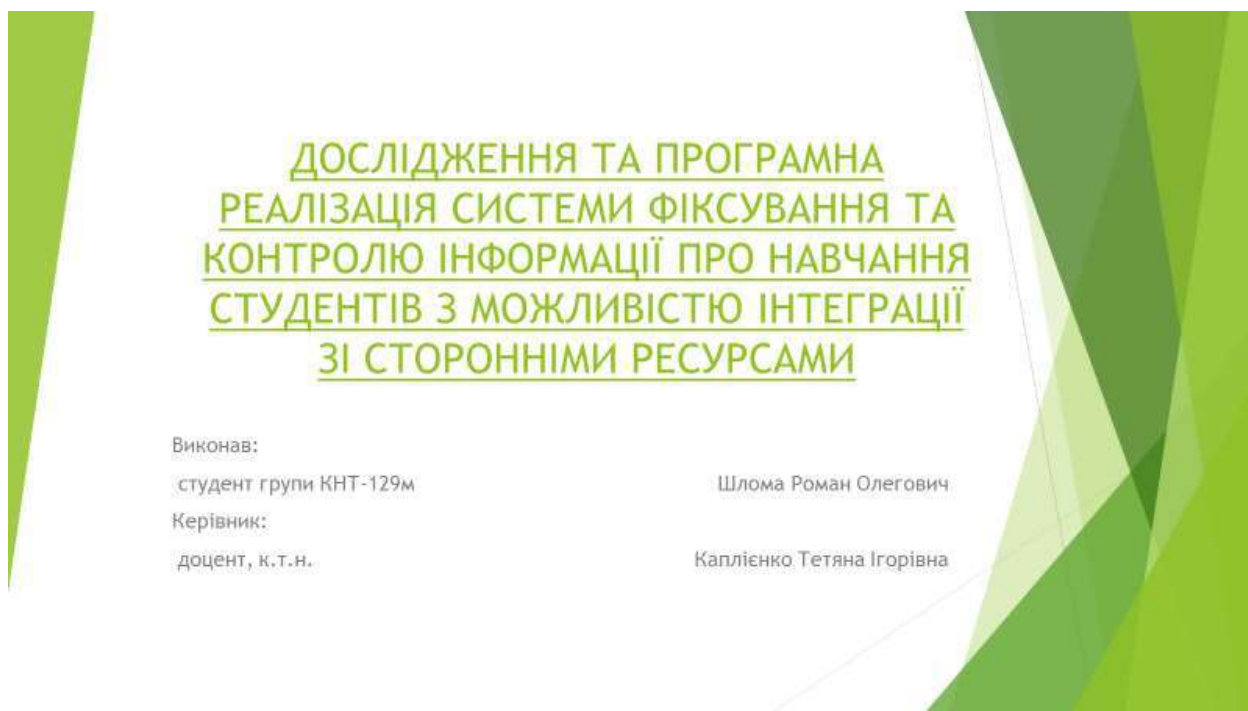


Рисунок Б.1 – Слайд 1. Титулка

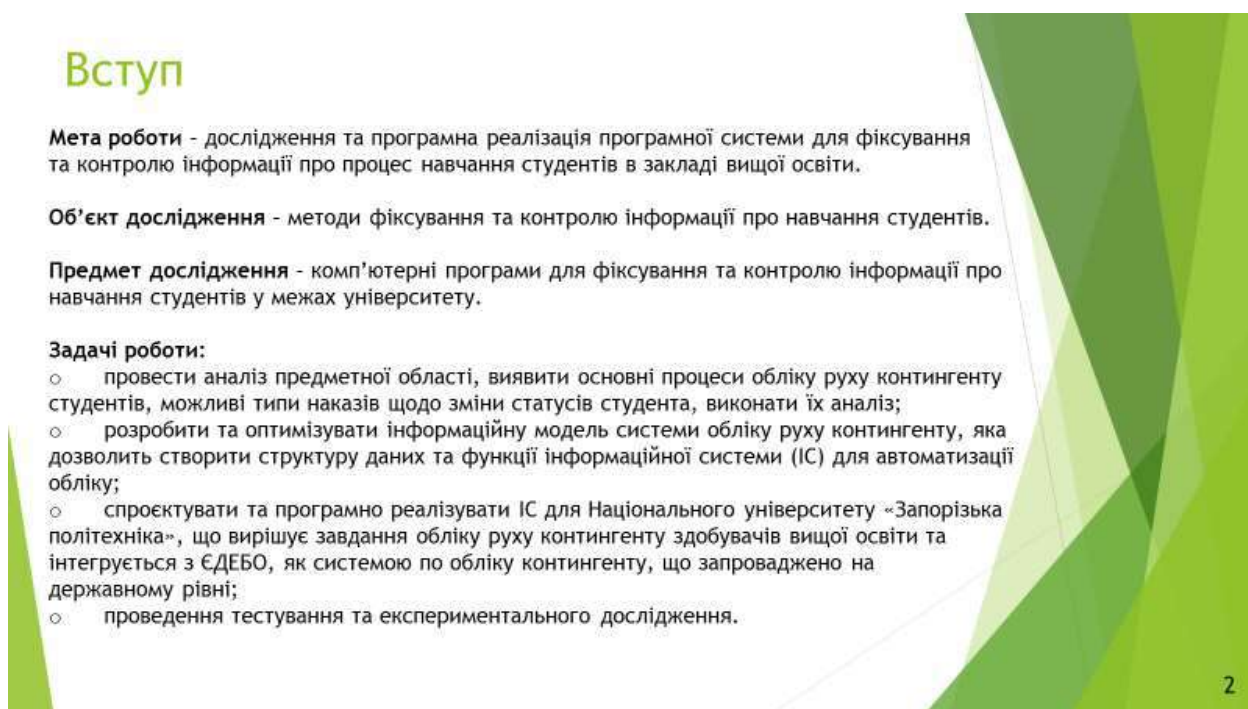


Рисунок Б.2 – Слайд 2. Вступ

Аналіз предметної області. Основні поняття



3

Рисунок Б.3 – Слайд 3. Аналіз предметної області

Існуюча система

№	Група	Прізвище	Ім'я	Побачення	Вік	Стать	Факультет	Кафедра	Освітня програма	Спеціальність	Посвідчення / Ліцензія	Дата	Рік	Примітки
1	М-119	Коростов	Сергій	Сергійович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
2	М-119	Коростов	Дмитро	Дмитрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
3	М-119	Ванова	Галина	Володимирівна	1	Жінка	Тех. Маши.	131	Прикладна механіка					
4	М-119	Ворова	Марина	Володимирівна	1	Жінка	Тех. Маши.	131	Прикладна механіка					
5	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
6	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
7	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
8	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
9	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
10	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
11	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
12	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
13	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
14	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
15	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
16	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
17	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
18	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
19	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
20	М-119	Ворова	Владислав	Владиславович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					
21	М-119	Ворова	Олександр	Олександрович	1	Чоловік	Тех. Маши.	131	Прикладна механіка					

4

Рисунок Б.4 – Слайд 4. Існуюча система

Огляд комерційних аналогів

Пункт\Система	Salesforce for Education	Oracle Higher Education Cloud	Workday Student	Розроблювана система
Вартість	Платна	Платна	Платна	Безкоштовна
Можливість взаємодії з іншими системами	-	-	-	+
Простота використання	+	+	-	+
Хмарна база даних	+	+	+	-
Необхідність постійного підключення до інтернету	+	+	+	-
Право власності	-	-	-	+
Можливість взаємодії з ЄДЕБО	-	-	-	+

5

Рисунок Б.5 – Слайд 5. Огляд комерційних аналогів

Аналіз вимог



Нефункціональні вимоги:

- ергономічний та інтуїтивно зрозумілий інтерфейс;
- безпека та конфіденційність даних;
- надійність і відновлюваність системи;
- швидкодія.

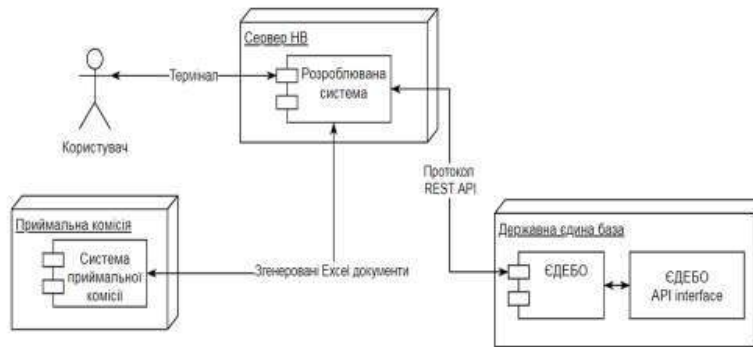
Системні вимоги:

- операційна система: Windows 10 (x64);
- оперативна пам'ять: 4ГБ;
- постійна пам'ять 10ГБ;
- наявність пакету Microsoft Office 2019.

6

Рисунок Б.6 – Слайд 6. Аналіз вимог

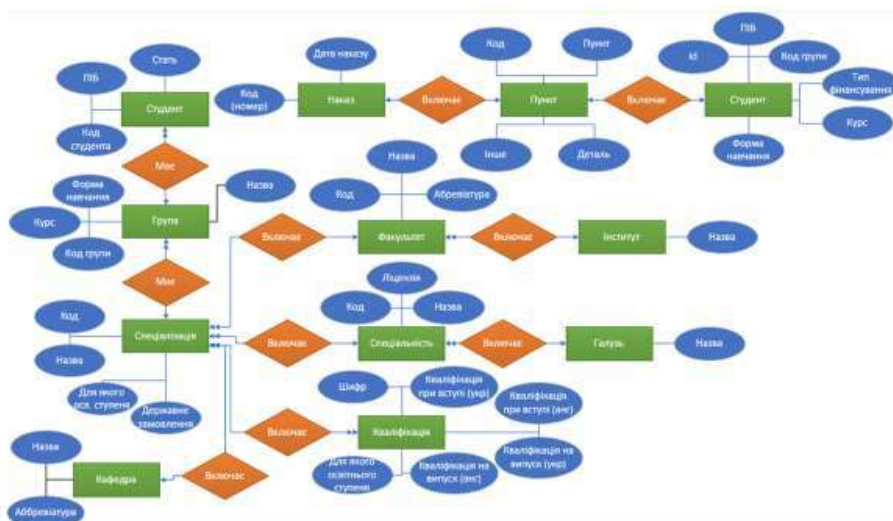
Розроблена система в інформаційній мережі університету



7

Рисунок Б.7 – Слайд 7. Розроблена система в інформаційній мережі університету

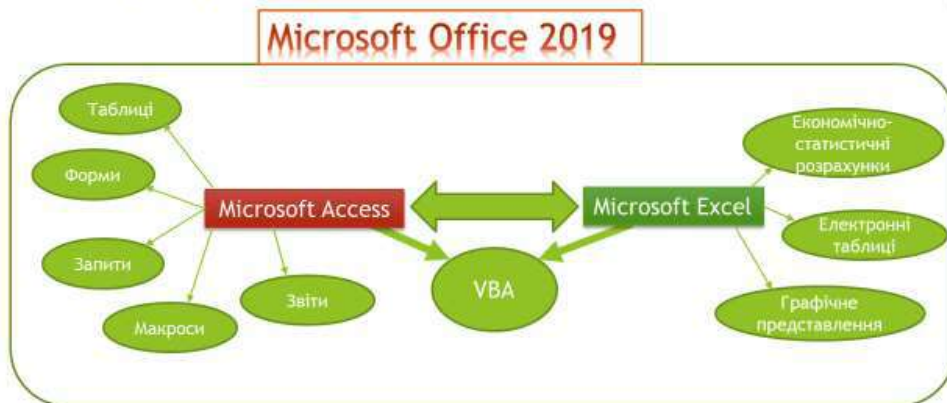
Концептуальна модель бази даних



8

Рисунок Б.8 – Слайд 8. Концептуальна модель бази даних

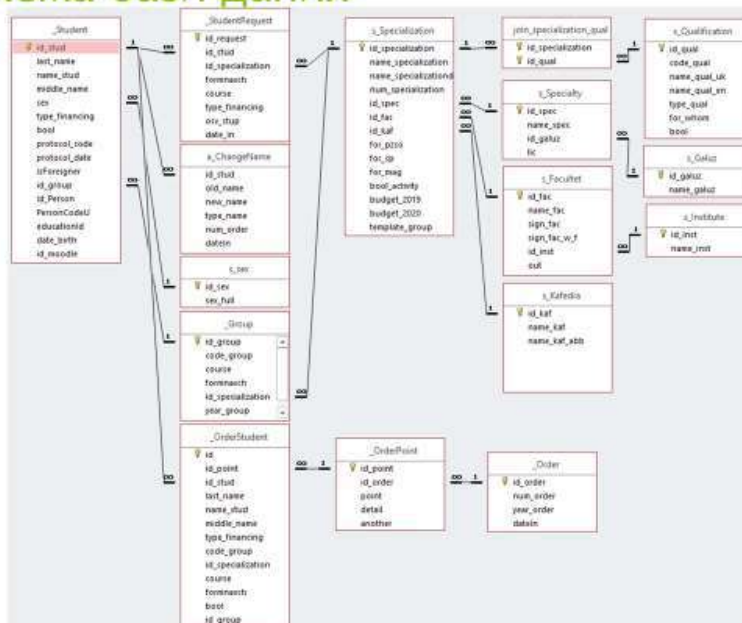
Середовище розробки та мова програмування



9

Рисунок Б.9 – Слайд 9. Середовище розробки та мова програмування

Схема бази даних



10

Рисунок Б.10 – Слайд 10. Схема бази даних

Місця ліцензованого обсягу

У зашифрованому вигляді

d135 z125 d225 z235 d325 z335 d53 z53 d430 z430 d63 z63

У дешифрованому вигляді

	1 курс	2 курс	3 курс	4 курс	1м курс	2м курс
Ліцензія ДН	35	25	25	30	3	3
Факт ДН	12+0	18+0	22+0	18+0	3+0	3+0
Ліцензія ЗН	25	35	35	30	3	3
Факт ЗН	12+0	2+0	18+0	27+0	3+0	3+0

Зберегти

11

Рисунок Б.11 – Слайд 11. Місця ліцензованого обсягу

Місця державного замовлення

У зашифрованому вигляді

d1P20-70 d1Y20-20 d2P19-10 d2Y19-20 d3P18-15 d3Y18-5 d3Y17-2

У дешифрованому вигляді

	1 курс		2 курс		3 курс			4 курс				1м курс	2м курс
	ЗО 20	МС 20	ЗО 19	МС 19	ЗО 18	МС 18	МС 19	ЗО 17	МС 17	МС 18	МС 19	Mar 20	Mar 19
Державне замовлення	70	20	10	20	2	5	0	0	0	0	0	0	0
Факт	6	3	10	0	6	0	0	4	0	0	0	2	0
Вакансії	64	17	0	20	-4	5	0	-4	0	0	0	-2	0

12

Рисунок Б.12 – Слайд 12. Місця державного замовлення

Перенесення даних з попередньої системи

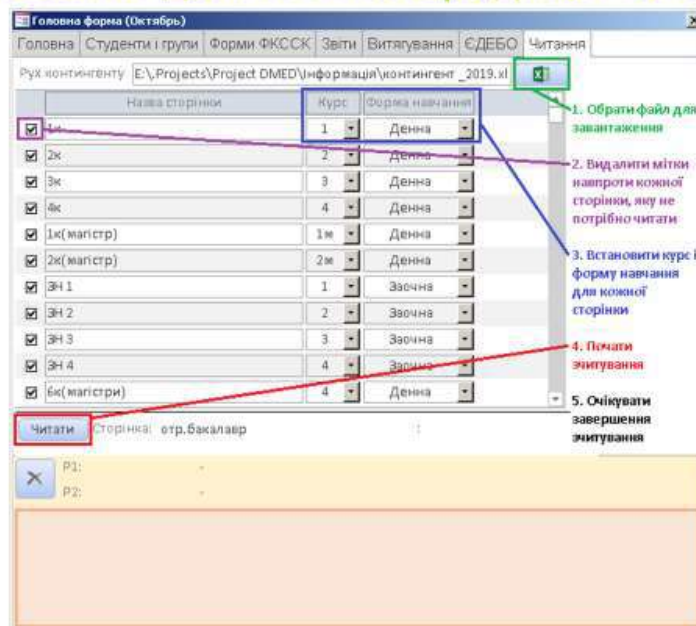


Рисунок Б.13 – Слайд 13. Перенесення даних з попередньої системи

Графічний інтерфейс. Контингент

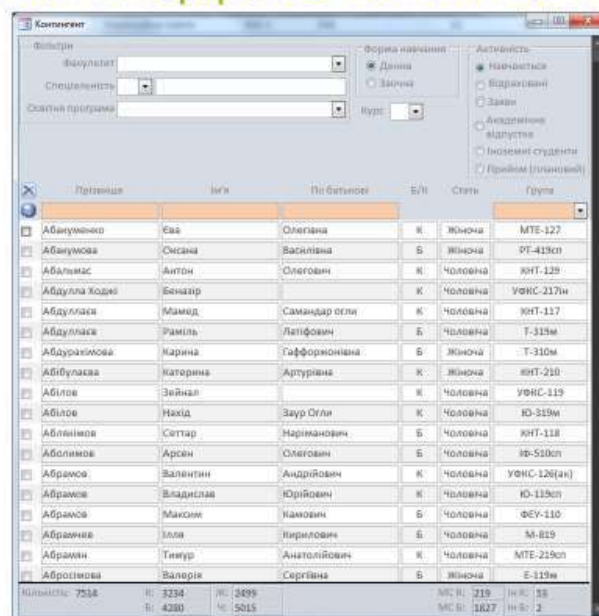
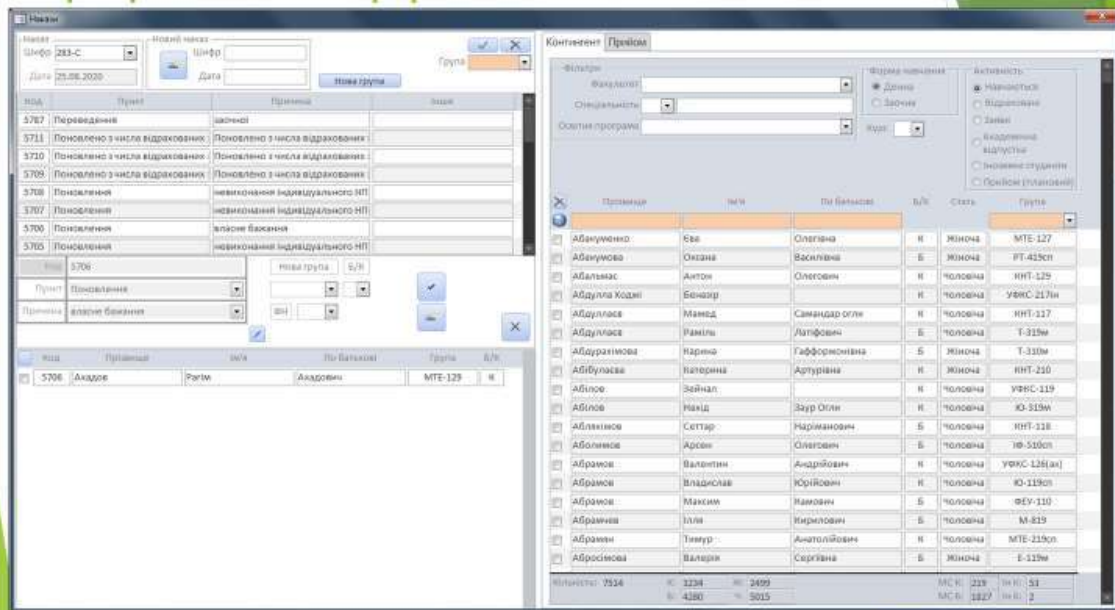


Рисунок Б.14 – Слайд 14. Графічний інтерфейс. Контингент

Графічний інтерфейс. Накази



15

Рисунок Б.15 – Слайд 15. Графічний інтерфейс. Накази

Тестування

1. Завантажено весь об'єм даних за ~3 хвилини із попередньої системи у нову;
2. Виконано альфа-тестування всього функціоналу розробником;
3. Виконано бета-тестування кінцевим користувачем;
4. Доопрацювання та виправлення помилок.

Експлуатація

- кінцеве завантаження даних;
- реальна експлуатація із травня місяця до поточного часу;
- розширення функціоналу.

Швидкодія

- «Рух контингенту» - 41 сек;
- «Ліцензія» - 33 сек;
- «Вакансії» - 38 сек;
- «Відомості про рух контингенту» - 10-60 сек;
- «Ставки» - 15 сек.

16

Рисунок Б.16 – Слайд 16. Графічний інтерфейс. Тестування, експлуатація та швидкодія

Висновки

- ✓ виконано аналіз предметної області та вимог;
- ✓ розроблено архітектуру та концептуальну модель системи;
- ✓ розроблено базу даних, графічний інтерфейс та програмний функціонал для роботи із інформацією про контингент, автоматизованого і швидкого створення різноманітної звітності;
- ✓ завантажено дані із попередньої системи;
- ✓ виконано повне тестування розробником і кінцевим користувачем системи та перевірку на виконання поставлених вимог.

Наукова новизна полягає у тому, що під час дослідження та програмної реалізації системи фіксування та контролю інформації про навчання студентів були розроблені методи розрахунку та зберігання місць державного замовлення та місць ліцензованого обсягу.

17

Рисунок Б.17 – Слайд 17. Висновки

Публікації

На момент захисту за темою даної кваліфікаційної дипломної роботи на науково-практичну конференцію «Тиждень науки», яка дистанційно проводилася у Національному університеті «Запорізька політехніка», були подані тези доповіді.

Також матеріали цієї роботи подано на розгляд до участі в конференції 2021 IEEE European Technology and Engineering Management Summit.

Дякую за увагу

Дякую за увагу

18

Рисунок Б.18 – Слайд 18. Публікації