

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Інформаційна система обліку навчального навантаження на кафедрі»

за напрямом підготовки 6.050101 «Комп'ютерні науки»

Виконавець роботи: студента групи ІТ-52 Тищенка Дмитра Вадимовича

Кваліфікаційна робота бакалавра

захищена на засіданні ЕК

з оцінкою

_____ «__» _____ 2019 р.

Науковий керівник

(підпис)

к.т.н., доц., Ващенко С. М.
(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.
(науковий ступінь, вчене звання, прізвище та ініціали)

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

Студент _____
(підпис)

Суми-2019

РЕФЕРАТ

Тема бакалаврської роботи «Інформаційна система обліку навчального навантаження на кафедрі».

Пояснювальна записка містить вступ, 4 розділи, висновки, додатки та список літератури, включає 113 сторінок, 7 таблиць, 67 ілюстрацій, 21 джерело.

В першому розділі наведений огляд актуальності напрямку досліджень. Визначається аналіз сучасного стану застосування інформаційних технологій для вирішення задач в обраній предметній області, визначаються наявні переваги та недоліки систем аналогів, обґрунтовується актуальність роботи.

Другий розділ включає в себе постановку задачі. А саме формулювання мети проекту, задач та вибору засобів реалізації.

Третій розділ призначений для опису поетапного проектування інформаційної системи та розробки архітектури системи. В даному розділі наведений структурно-функціональний аналіз проекту, описані основні принципи роботи розробленої інформаційної системи.

У четвертому розділі описується розробка інформаційної системи, що є результатом практичного застосування запропонованих методологій дослідження. Детально наводяться реалізація фізичної моделі бази даних, реалізація інтерфейсу, реалізація модуля парсингу PDF-файлів і тестування продукту.

Результатом проведеної роботи є розроблена інформаційна система розподілу навчального навантаження на кафедрі.

Ключові слова: прикладне програмування, бази даних, навчальне навантаження, кафедра, викладач.

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Робота з форматом PDF	10
1.2 Процес обробки PDF-таблиць.....	12
1.3 Огляд існуючих програмних продуктів.....	13
2 ПОСТАНОВКА ЗАДАЧІ.....	17
2.1 Мета та задачі	17
2.2 Вибір засобів реалізації	18
2.3.1 Вибір системи керування базою даних.....	18
2.3.2 Вибір мови програмування	21
2.3.3 Вибір модуля парсингу.....	22
3 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	26
3.1 Структурно-функціональне моделювання процесу розподілу навчального навантаження на кафедрі.....	26
3.2 Проектування моделі бази даних	29
3.3 Моделювання варіантів використання	33
3.4 Моделювання діаграми послідовності.....	34
4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗПОДІЛУ НАВЧАЛЬНОГО НАВАНТАЖЕННЯ НА КАФЕДРІ	38
4.1 Реалізація фізичної моделі БД.....	38
4.2 Загальна реалізація інтерфейсу	41
4.3 Модуль парсингу PDF-файлів	43

	4
4.4 Тестування продукту	44
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
ДОДАТОК А.....	54
ДОДАТОК Б	55
ДОДАТОК В.....	61
ДОДАТОК Г	71
ДОДАТОК Г	71
ДОДАТОК Д.....	74
ДОДАТОК Е	103

ВСТУП

На сьогодні накопичено значні обсяги інформації, створеної людьми. Раніше для обробки інформації використовувалися людські ресурси і на це витрачалося багато часу. В наш час швидкість обробки інформації в багатьох сферах людської діяльності замала і потребує автоматизації цього процесу. Тому проблема автоматизації обробки інформації є однією із найактуальніших проблем в наш час.

На сьогоднішній час методист кафедри для виконання розподілу навчального навантаження повинен виконати наступні дії. Після отримання файлу з навчальними планами методист кафедри повинен занести всі предмети і навантаження до файлу Excel вручну. Після занесення фалів методист проводить розподіл навантаження між викладачами та формує відповідну звітну документацію. З урахуванням тенденцій нашого часу такий підхід є нераціональним і потребує автоматизації.

Мета дипломного проекту полягає у розробці інформаційної системи, яка б дозволила автоматизувати роботу методиста кафедри університету по розподілу навантаження між викладачами кафедри.

У відповідності до мети дипломного проекту були поставлені наступні задачі для дослідження:

1. Дослідження літератури і програмних додатків по темі проекту.
2. Вивчення процесу роботи методиста кафедри університету щодо розподілу навчального навантаження.
3. Розробка моделі інформаційної системи.
4. Створення бази даних для збереження потрібної інформації.

5. Розробка програмного забезпечення по автоматизації робіт щодо розподілу навантаження між викладачами з реалізацією модуля автоматизованого внесення навантаження в базу даних.

Практичне значення – використання розробленого програмного забезпечення дозволить спростити процес розподілу навчального навантаження на кафедрі навчального закладу та скоротити час, який витрачає на це методист.

Результати дипломної роботи в частині технології організації парсингу табличної інформації з PDF-файлу проваджено в діяльність ФОП Кондрашова Н. О., про що складено відповідний акт (додаток А).

Результати роботи доповідалися на конференції «ІМА -2019» . За матеріалами роботи опубліковано тези. [1]

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Необхідність автоматизації процесу розподілу навчального навантаження на кафедрі виникла в результаті витрати великої кількості часу на виконання цього завдання людськими ресурсами, а саме методистом кафедри.

Розглянемо процес виконання розподілу навчального навантаження. Спочатку на кафедрі приходять навчальні плани у вигляді PDF-файлу. Приклад навчальних планів представлений на рисунку 1.1.

Сумський державний університет
 Інститут (факультет) Кафедра комп'ютерних наук

ЗАТВЕРДЖУЮ:
 Перший проректор _____ В.Д.Керуша
 _____ року

Обсяг навчальної роботи на 2018/2019 навчальний рік

кафедри (секції) секція "Інформаційні технології проєктування"

№ к.п.	Назва дисципліни та вид навчальної роботи	Вісн.	Академічний курс	Кількість студентів	Кількість груп	Кількість пар	Кількість годин	Розподілено робочий навантаження по											Розподілено годин																				
								лекції	практика	семинарні заняття	лабораторні заняття	наг. зан. і спец. роб. згідно з програмою дисципліни	виступи студентів	лекція викладача	спільна робота з викладачем	самостійна робота з викладачем	самостійна робота студентів	ІНД	індивідуальні завдання	Питання	Питання, семінарні заняття	Роботничі заняття	наг. зан. або спец. роб. згідно з програмою дисципліни	Аудиторні заняття	Прочитання текстів	Підготовка навчальних завдань	Корекція і графічний рід. завд.	Викорст. І-нав. засоб. програм. засоб.	Автоматиз. програм. засоб.	Заняття з використанням комп'ютерних засобів навчання	Державні історичні пам'ятники	Державні історичні пам'ятники	Курсові роботи	Інші	ВСЬОГО	у т.ч. постійна частина			
Осінній семестр																																							
двипл. форми навчання																																							
1	ПП	Об'єктно-орієнтовані мови та проєктування	1	ІІ-кв	17	1	2	1	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	ПП	Програмування графічними пристроями	1	ІІ-кв	17	1	2	1	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	ПП	Проєктування веб-орієнтованих інформаційних систем	1	ІІ-кв	17	1	2	1	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	ПП	Проєктування веб-орієнтованих інформаційних систем - КР	1	ІІ-кв	17	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	ПП	Сервісори та підсистеми програмного забезпечення	1	ІІ-кв	17	1	2	1	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	ПП	Сервісори та підсистеми програмного забезпечення	1	ІІ-кв	17	1	2	1	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	ПП	Організація та обробка мережевої інформації	1	ІІ-кв	17	1	2	1	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	ПП	Організація та обробка мережевої інформації	1	ІІ-кв	17	1	2	1	10	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	ПП	Організація та обробка мережевої інформації - КР	1	ІІ-кв	17	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 1.1 –Планові обсяги навчальної роботи

Головні елементи навчального плану це назва предмету. Назви предметів можуть повторюватися через те, що один предмет може викладатися для різних навчальних груп студентів. Наступний елемент це аббревіатура групи. В одній комірці може бути декілька груп, що навчаються на одному курсі. Наступні елементи стосуються груп, а саме кількості студентів, груп, підгруп, потоків. Далі йде розподіл годин на лекції, практичні заняття, лабораторні роботи, індивідуальні заняття, атестаційні заходи і кількість кредитів на практику. Наступні елементи це наявність заліку, підсумкового семестрового контролю, екзамену, додаткового семестрового контролю, а також далі йде вид індивідуального завдання і їх кількість.

Наступним елементом йде розрахунок годин. Для кожного типу заняття є своя формула розрахунку кількості годин. Кожний тип заняття має виділену кількість годин під одне заняття. Для розрахунку кількості годин перемножується кількість годин виділене під одне заняття на кількість занять. Наприклад, для одного лабораторного заняття виділяється 2 години, якщо ми маємо 16 занять, то отримуємо 32 години навчального навантаження для викладача.

Після отримання файлу методист кафедри в ручному режимі проводить перенесення даних з PDF-файлу до файлу Excel. Приклад занесених даних до файлу представлений на рисунку 1.2.

The screenshot shows an Excel spreadsheet with a table of student course loads. The table is organized into columns for course names, credit hours, and various numerical values. The interface includes the Microsoft Excel ribbon and a taskbar at the bottom.

Рисунок 1.2 – Навчальне навантаження у файлі Excel

Інформація по групам студентів для зручності заноситься у окрему вкладку. Приклад занесення груп представлений на рисунку 1.3.

The screenshot shows an Excel spreadsheet with a table of student groups. The table is organized into columns for group names, course names, and numerical values. The interface includes the Microsoft Excel ribbon and a taskbar at the bottom.

Рисунок 1.3 – Занесення груп

системи.[2] Винайдений компанією Adobe, PDF тепер є відкритим стандартом, який підтримує Міжнародна організація зі стандартизації (ISO). PDF-файли можуть містити посилання та кнопки, поля форм, аудіо, відео та бізнес-логіку. Їх можна також підписати в електронному вигляді і легко переглядати за допомогою безкоштовного програмного забезпечення Acrobat Reader DC.

Формат PDF в першу чергу розроблявся і призначений для представлення поліграфічної інформації у електронному вигляді.[3] Значна кількість сучасного професійного друкарського обладнання має апаратну підтримку формату PDF, що дозволяє проводити друк документів в даному форматі без використання будь-якого програмного забезпечення.

Саме в тому, що формат PDF в першу чергу розроблявся для роботи з друкованою продукцією його і мінус, бо сам формат є повністю векторним.[4] Мінус вектору у тому, що в ньому не має поняття «Таблиці». У PDF-файлах комірки таблиць створюються за допомогою векторних примітивів, а саме ліній накреслених між опорними точками.[5] Всередині цих комірок додається текст і в результаті у PDF-файлі отримується таблиця.

Тому для обробки календарних планів у нашому випадку необхідно буде застосовувати процеси обробки і перетворення таких таблиць у необхідні нам об'єкти, або інші представлення інформації, без втрати структури таблиці, для подальшого занесення інформації до бази даних і роботи з нею.

1.2 Процес обробки PDF-таблиць

Для занесення даних до бази даних необхідним буде вирішити проблему правильного занесення даних до бази. А саме так як PDF-документ не має у своїй структурі поняття «таблиця» і є всього лише файлом з набором векторних об'єктів, а саме тексту, різних примітивів і тому подібне. Тому для занесення інформації до бази даних спочатку її потрібно буде перетворити у об'єкти, які будуть вміщувати в себе, наприклад, прізвище викладача, або кількість годин певного предмету. Саме такий процес обробки таблиць у PDF-файлі, а саме процес діставання потрібної нам інформації із комірок в таблиці у вигляді об'єктів, або перетворення називають парсингом, або синтаксичним аналізом.

Parsers (синтаксичний аналіз) є однією з найбільш зрозумілих галузей інформатики. Parsers вже широко використовуються в ряді дисциплін: у комп'ютерних науках (для побудови компіляторів, інтерфейсів баз даних, самоописаних баз даних, штучного інтелекту), в лінгвістиці (для аналізу тексту, аналізу корпусів, машинного перекладу, текстовий аналіз біблійних текстів), у підготовці та перетворенні документів, в наборі хімічних формул і в розпізнаванні хромосом, назвавши декілька; вони можуть бути використовуються (і, можливо, є) у значно більшій кількості дисциплін[6].

Parsing це процес структурування лінійного представлення відповідно до заданої граматики.[7] Це визначення зберігається абстрактним, щоб дозволити якомога ширше його тлумачити. "Лінійне представлення" може бути пропозицією, комп'ютерною програмою, схемою в'язання, послідовністю геологічних шарів, музичним твором, діями в ритуальній поведінці, короткою лінійною послідовністю, в якій попередні елементи певним чином обмежують наступний елемент. Для деяких з прикладів граматика добре відома, для деяких

вона є об'єктом дослідження, а для деяких наше поняття граматики тільки починає формуватися.[8]

1.3 Огляд існуючих програмних продуктів

Для огляду існуючих програмних продуктів ми повинні визначитися з вимогами до них. По-перше програмний засіб повинен виконувати парсинг файлу з навчальним навантаженням. Після виконання парсингу нам потрібна бути доступна можливість виконувати розподіл навантаження, а саме ставити у відповідність викладачу дисципліну і рахувати кіськість робочих годин викладача. Також потрібна функція друку навчального навантаження викладача.

На поточний час кожен вищий навчальний заклад вирішує самостійно проблему виконання розподілу навчального навантаження на кафедрах. Література за запитом щодо розподілу навчального навантаження на кафедрі відсутня, тому виконаємо пошук за процесами роботи, а саме парсинг файлів, розподіл доручень в колективі.

Дуже функціональним сервісом є Docparser. Суть цього сервісу полягає у наданні вам широкого спектру можливостей обробки і перетворення PDF файлу у потрібний вам формат.[9] Також цей сервіс дуже гарно працює з таблицями, що для нас є важливим фактором для даного проекту. Цей сервіс дозволяє обрати тип документу, що ви хочете парсити. Вибір типу документу зображений на рисунку 1.5.

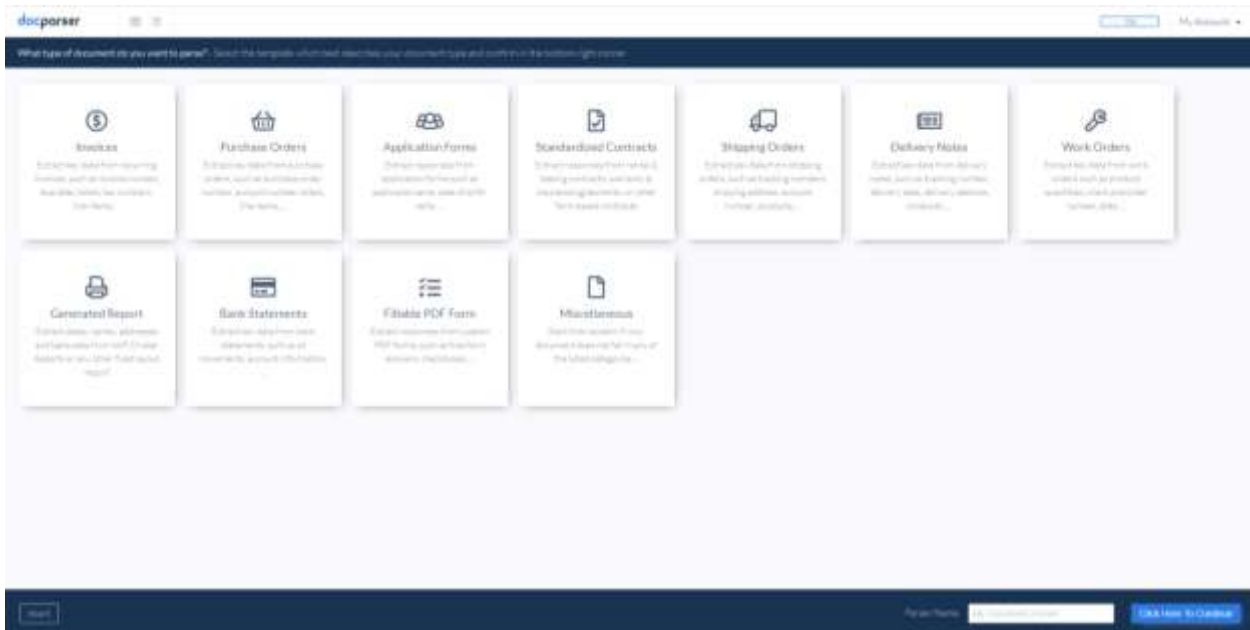


Рисунок 1.5 – Вибір типу документу для парсингу

Після вибору типу документу і подальшого додання самого документу ви можете обрати правила парсингу (Parsing Rule), або вже готові набори правил. Вікно створення правил парсингу зображено на рисунку 1.6.

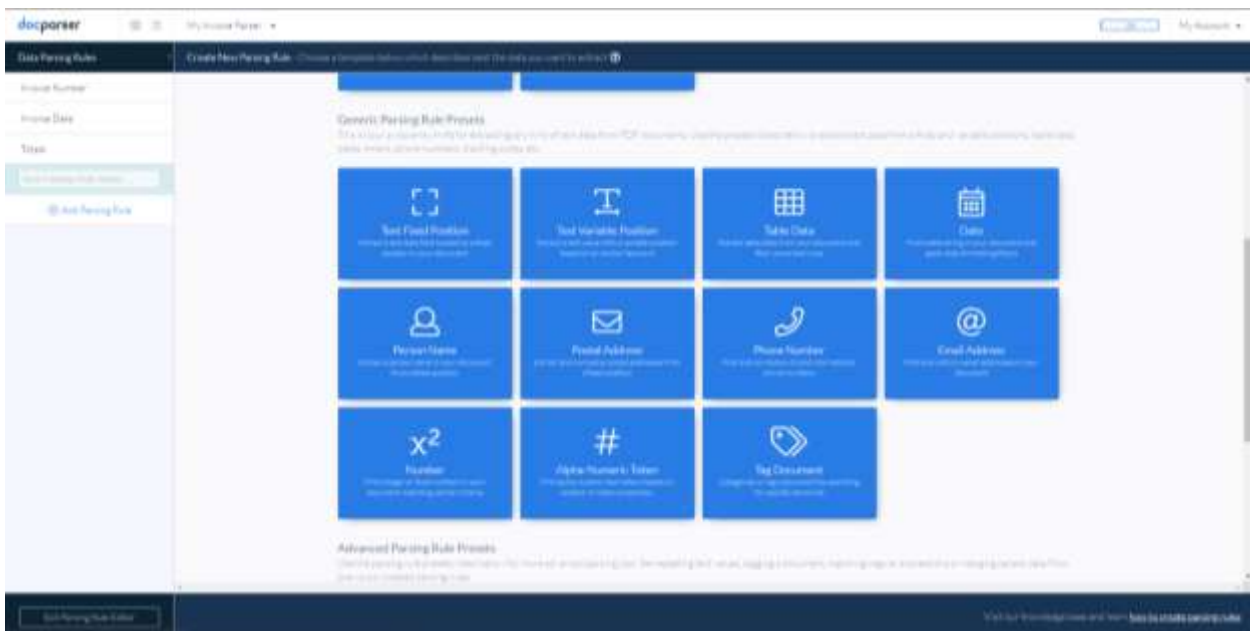


Рисунок 1.6 – Створення нового правила парсингу

Далі самий головний для нас момент це виведення даних після парсингу. Вивід даних в даному сервісі представлений тільки у вигляді виводу у файли форматів XLS, CSV, JSON, XML.

Наступний функціональний інструмент це Tabula. Цей інструмент дозволяє перевести дані із PDF формату у CSV, Excel таблиці, JSON через простий веб-інтерфейс.[10] Але так як і попередній сервіс Tabula не дозволяє з базами даних, а тим більш виконувати розподіл навантаження між викладачами, тобто ми не маємо можливості закріплювати викладача за певним предметом і групою.

Для виконання планування задач розроблено велику кількість програмного забезпечення. Одним з таких програмних продуктів є ЛідерТаск. В цій програмі легко вносити правки в тижневий розклад: рухати справи на інший час і день, міняти тривалість, оновлювати формулювання або просто скасовувати неактуальні[11]. Приклад погодинного розкладу тижня зображений на рисунку 1.7.

	понеділок 14	вівторок 15	среда 15	четверг 15	п'ятниця 16
09 ⁰⁰	Планёрка	Забронировать номер в гостинице	Расчитать бюджет на июнь	Планёрка	Обсудить систему бонусов
10 ⁰⁰	Сдать отчёт за неделю	Обновить дизайн страниц	Заказать два принтера и сканер	Проконтролировать ремонт офиса	Купить билеты на поезд
11 ⁰⁰	Забрать документы из банка	Страница «О нас»	Подготовить зал	Отправить документы заказным письмом	
12 ⁰⁰		Страница «Купить»			
13 ⁰⁰					

Рисунок 1.7 – Погодинний розклад тижня

Також в програмі ЛідерТаск представлений механізм доручень. Приклад доручень зображений на рисунку 1.8.

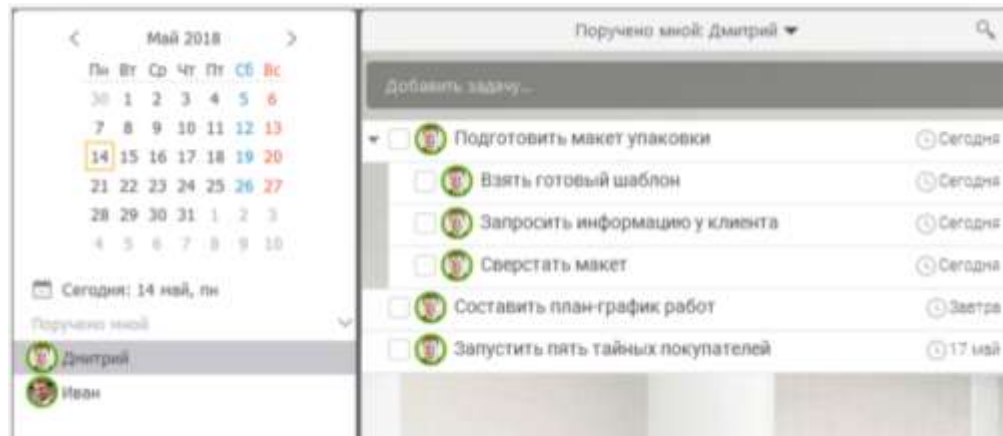


Рисунок 1.8 – Доручення

Після аналізу всіх представлених зараз на ринку рішень по вирішенню потрібної задачі видно, що ні один програмний продукт не може задовільнити всі потрібні вимоги.

Більшість представлених програмних продуктів дозволяють лише виконати процес парсингу PDF файлу без подальшої роботи з отриманими даними. А програмні продукти, що використовують для розподілу навантаження не дозволяють працювати з даними так, як потрібно в нашому випадку. Саме через специфічність поставленої задачі, а також через вище перелічені проблеми, було прийнято рішення про створення власного програмного засобу, що стане повністю утилітарним рішенням даної проблеми.

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Мета та задачі

Мета проекту: створити програмний продукт, який дозволить автоматизувати роботу методиста кафедри та розподілу кафедри, в тому числі розподіл навантаження між викладачами кафедри.

Створений програмний продукт орієнтований на його використання в закладах освіти, на кафедрах. Даний проект буде представлений безпосередньо Сумський державний університет (м. Суми).

Для роботи програмного продукту рекомендовано такі мінімальні системні вимоги до комп'ютерної техніки:

- операційна система Windows Xp, 7, 8, 10;
- одноядерний процесор з тактовою частотою 1 ГГц;
- оперативна пам'ять розміром 512 Мб.

Після запуску програмного продукту користувач буде повинен завантажити PDF-файл з навчальним планом. Потім обрати необхідні предмети і поставити їх у відповідність потрібного викладача. Після виконання всіх дій користувач матиме можливість роздрукувати навчальне навантаження конкретного викладача.

Схема роботи програмного додатку зображена на рисунку 2.1.



Рисунок 2.1 – Схема роботи

Повна та детальна інформація про вимоги до розробки програмного продукту представлена в технічному завданні (додаток Б).

Проведено планування виконання робіт та оцінка ризиків виконання проекту. Результати планування наведено у додатку В.

2.2 Вибір засобів реалізації

2.3.1 Вибір системи керування базою даних

На сьогоднішній час існує велика кількість систем керування базами даних. Розглянемо деякі з них і проведемо порівняльну характеристику. Найважливішим фактором підбору СКБД для порівняння була саме

безкоштовність. Із безкоштовних варіантів були підібрані 3 найбільш відомі і стабільні СКБД.

СКБД MongoDB призначена для додатків, які використовують як структуровані, так і неструктуровані дані. Ядро є дуже гнучким і працює при підключенні бази даних до додатків через драйвери MongoDB. Розповсюджується безкоштовно, хоча має і комерційну версію.[12]

Переваги даної СКБД:

- швидкість і простота у використанні;
- движок підтримує json і інші традиційні документи NoSQL;
- дані будь-якої структури можуть бути збережені / прочитані швидко і легко.

Недоліки:

- SQL не використовується в якості мови запитів;
- інструменти для перекладу SQL-запитів в MongoDB доступні, але їх слід розглядати саме як доповнення.

MySQL – це СКБД, що дозволяє вибирати різні двигуни для системи зберігання, які дозволяють змінювати функціонал інструменту і виконувати обробку даних, що зберігаються в різних типах таблиць. Гнучкість СКБД MySQL забезпечується підтримкою великої кількості типів таблиць.[13]

Переваги:

- розповсюджується безкоштовно;
- прекрасно документована;
- пропонує багато функцій, навіть у безкоштовній версії;
- підтримує набір призначених для користувача інтерфейсів;
- може працювати з іншими базами даних, включаючи DB2 і Oracle.

Недоліки:

- доведеться витратити багато часу і зусиль, щоб змусити MySQL виконувати нескладні завдання, хоча інші системи роблять це автоматично, наприклад: створювати інкрементні резервні копії;
- відсутня вбудована підтримка XML або OLAP;
- для безкоштовної версії доступна тільки платна підтримка.

Також до розгляду включимо СКБД MariaDB. Ядро бази даних дозволяє робити вибір з кількох систем зберігання, і це робить використання ресурсів більш оптимізованим, що підвищує продуктивність запитів і обробки. Є безкоштовною і має багато готових плагинів розширень.[14]

Переваги:

- система працює швидко;
- індикатори дадуть вам знати, як обробляється запит;
- розширювана архітектура і плагіни дозволяють налаштовувати інструмент відповідно до ваших потреб;
- шифрування є в мережі, сервері і на рівні додатку.

Недоліки:

- на даний момент стабільність нижче, ніж у MySQL, тому навіть на нових проектах можна рекомендувати встановлювати mysql;
- двигун досить новий, тому поки немає ніяких гарантій подальших оновлень.

Після проведення порівняння було виявлено, що MariaDB є менш стабільною ніж MySQL, тому її використання не є раціональним. MongoDB була не обрана через відсутність SQL-запитів в якості мови запитів. Тому було прийнято рішення використати у саме MySQL.

2.3.2 Вибір мови програмування

Програмний продукт буде розроблятися під операційну систему Windows. Для реалізації продукту під Windows найчастіше використовуються такі мови програмування: C#, C++ і Java. Давайте проведемо порівняльну характеристику цих мов програмування і визначимо, яка більше підходить для нашого продукту.

C# - дуже гнучка і потужна мова програмування з цікавою історією. Він був розроблений корпорацією Майкрософт з різних причин, деякі технічні, деякі політичні та інші маркетингові. C# має сильну схожість з мовами програмування C++ і Java, запозичивши (або покращивши) функції, що надаються цими мовами. Витоки як Java, так і C++ можна простежити до мови C, яка є дуже небезпечною і цікавою мовою, винайденою на початку 1970-х років.[15]

Мова C++ виникла на початку 1980-х років, коли співробітник фірми Bell Labs Бйорн Страуструп придумав ряд удосконалень до мови C під власні потреби. C++ є складеною мовою. Щоб програма запускалася, її вихідний текст повинен оброблятися компілятором, виробляючи об'єктні файли, які об'єднуються linker, даючи виконувану програму. Програма C++ зазвичай складається з багатьох файлів вихідного коду (зазвичай просто називаються вихідними файлами).[16]

Java - сильно типізована об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems (в подальшому придбаної компанією Oracle). Java компілюється і інтерпретується. Замість перекладу програм на машинну мову, компілятор Java генерує байтовий код.[17] Байт-код легко (і швидко) інтерпретувати, як і машинна мова, але він також портативний, як мова високого рівня. Таким чином, можна компілювати програму на одній машині, передавати байтовий код на іншу машину, а потім інтерпретувати

байтовий код на іншій машині. Ця здатність є перевагою Java на багатьох інших мовах високого рівня.

Після порівняння всіх цих мов програмування було прийнято рішення про використання мови C# для розробки програмного продукту. Мова C# багато запозичує з мов C++ і Java. Функціональність мови повністю задовольняє потребам для розробки програмного продукту, а саме роботі з парсингом файлів і базами даних.

2.3.3 Вибір модуля парсингу

Розглянемо засоби для парсингу написані на мові C#. Перший засіб це iTextSharp. iTextSharp це потужний PDF Toolkit для генерації PDF, програмування PDF, обробки та маніпулювання.[18] Приклад застосування цього інструменту представлений на рисунку 2.2.



Рисунок 2.2 – Використання iTextSharp

Наступний інструмент PDFBox. Цей модуль дозволяє створювати нові документи PDF, маніпулювати існуючими документами і можливістю отримувати контент з документів.[19] Приклад використання цього інструменту наведений на рисунку 2.3.

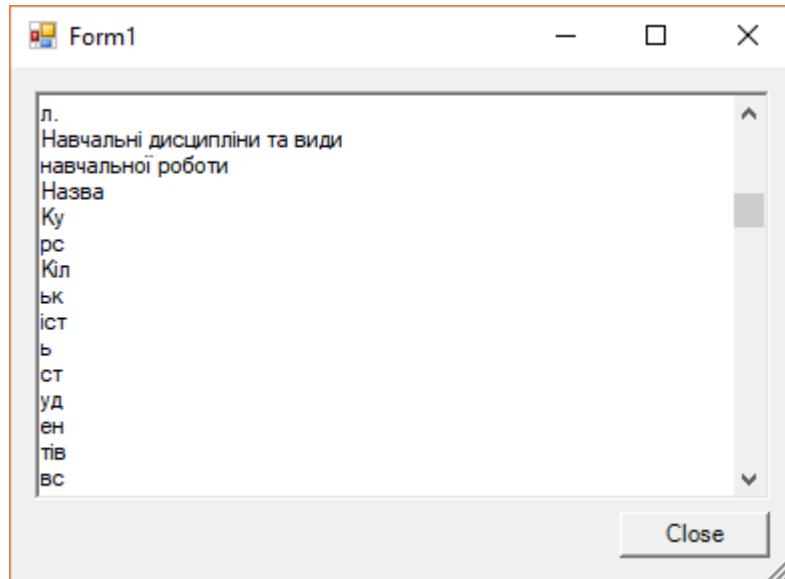


Рисунок 2.3 – Використання PDFBox

Після розгляду інструментів на мові програмування C# можна зробити висновок, що їх результати парсингу не задовільні для використання у нашому проєкті.

Тому для виконання парсингу PDF-файлу було вирішено розглянути програмні засоби реалізовані іншими мовами програмування. Огляд літературних джерел показав, що найпопулярнішими в цій області є додати і бібліотеки написані мовою Python. Python має більший функціонал для виконання парсингу. Ще одним фактором вибору стало те, що скрипти Python можна вільно виконувати з програмного коду C#.

Розглянемо модулі парсингу, написані на Python. Перший модуль це Tabula. Цей модуль є безкоштовним і виконує парсинг PDF-файлів у формати `xlsx`, `csv`, `json`. Недоліком цього модуля є те, що в ньому потрібно вказувати місцерозташування таблиць у файлі PDF.

Наступний інструмент це Camelot. На відміну від інших бібліотек та інструментів, які або дають хороший результат, або погано працюють (без

проміжних даних), Camelot дає вам можливість налаштувати автоматизований пошук таблиць у документі.[20]

Tabula і Camelot дають задовільні результати парсингу PDF-файлів. Недоліком Tabula для нас є обов'язкове вказування розташування таблиць у PDF-файлі. На відміну від Tabula Camelot має у своєму функціоналі автоматичний пошук таблиць, тому було прийнято використати модуль Camelot написаний на мові програмування Python.

3 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Структурно-функціональне моделювання процесу розподілу навчального навантаження на кафедрі

Першим етапом виконання роботи з розробки програмного забезпечення є побудова структурно-функціональної моделі основного процесу, який буде автоматизуватися – процесу розподілу навантаження методистом кафедри. Моделювання виконується згідно нотації IDEF0. Модель буде побудовано у форматі «to - be».

Розподіл навчального навантаження на кафедрі починається з розбору файлу PDF з навчальним навантаженням, який надсилають методу кафедри. Методист керуючись положенням про нормування робочого часу виконує розподіл навчального навантаження між викладачами.

Після розподілу навантаження ми отримуємо звіт про навантаження для кожного викладача і архів розподілу навантаження, а саме базу даних в якій буде зберігатися інформація з навантаженням кожного викладача. Контекстна діаграма структурно-функціональної моделі зображена на рисунку 3.1. Ця діаграма описує модель процесу розподілу навантаження в узагальненому вигляді.

Далі згідно методології проводилася декомпозиція контекстної діаграми та діаграм нижчих рівнів. Отримані діаграми структурно-функціональної модулі наведено у додатку Г.

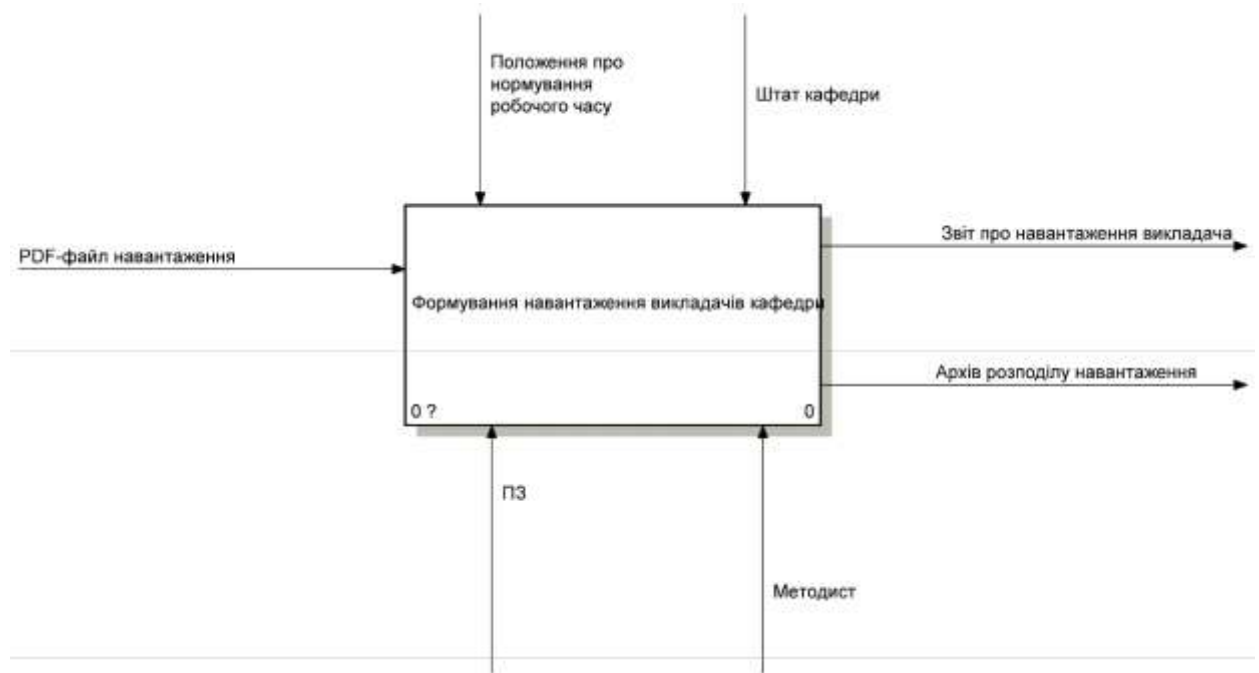


Рисунок 3.1 – Контекстна діаграма

«Формування навантаження викладачів кафедри» декомпозується на два процеси – заповнення планового навантаження і розподіл навантаження між викладачами. Діаграму декомпозиції першого рівня наведено у додатку Г.

Блок «Заповнення планового навантаження» на вході має PDF-файл навантаження. На виході ми отримуємо дані навантаження в БД. Управління процесом здійснюється згідно положення про нормування робочого часу, а також штату кафедри. Механізмами процесу виступають програмне забезпечення і методист кафедри.

Процес «Заповнення планового навантаження» виконується в чотири етапи, які в діаграмах будуть представлені відповідними на чотири процеси – саме: парсінг, перевірка коректності парсінгу, за потреби редагування отриманої інформації та її завантаження в базу даних.

Вхідною інформацією для «Парсінгу» є PDF-файл з навантаженням. На виході ми отримуємо розпарсені дані, які містять конкретні значення відповідних комірок таблиці PDF-файл. Механізмом процесу виступає програмне забезпечення, що буде розроблено.

«Перевірка коректності парсінгу» на вході має розпарсені дані, PDF-файл з навчальним навантаженням і відредаговані дані. Управління здійснюється згідно положення про нормування робочого часу і штатом кафедри. Механізмом є програмне забезпечення і методист кафедри. На виході отримуються коректні дані, або некоректні дані.

На вхід процесу «Редагування» подаються некоректні дані. Механізмом є програмне забезпечення і методист. На виході отримуються відредаговані дані.

«Завантаження в БД» на вході має коректні дані. Механізмом процесу виступає програмне забезпечення, а на виході отримуються дані навантаження у базі даних.

На вхід процесу «Розподіл навантаження» подаються дані навантаження в БД. На виході отримуються звіт про навантаження викладача і архів розподілу навантаження. Управління процесом здійснює штат кафедри, а механізмом є програмне забезпечення і методист кафедри.

Розподіл навантаження декомпозується на чотири процеси, а саме вибірку інформації з бази даних, закріплення викладача, збереження в базі даних, формування звітності для викладача.

«Вибірка інформації з БД» на вході має дані навантаження в БД. На виході отримуються обрані дані. Механізмом є програмне забезпечення.

На вхід процесу «Закріплення викладача» подаються обрані дані. На виході отримують закріплені викладачі. Управління здійснюється штатом кафедри. Механізмом є програмне забезпечення і методист кафедри.

«Збереження в БД» на вході має закріпленого викладача. На виході архів розподілу навантаження. Механізмом є програмне забезпечення.

«Формування звітності для викладача» на вході маж архів розподілу навантаження. На виході отримується звіт про навантаження викладача. Механізмом є методист кафедри.

Схема отриманої функціональної моделі зображена на рисунку 3.2.

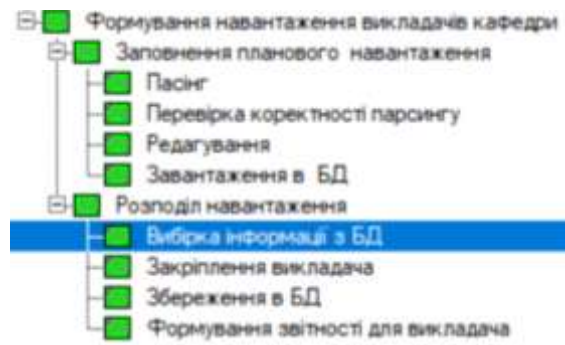


Рисунок 3.2 – Схема функціональної моделі

3.2 Проектування моделі бази даних

Як зазначалося раніше, всі інформація про навантаження та виконаний розподіл має зберігатися в базі даних. Для кращого розуміння предметної області створимо модель предметної області, а саме ER-діаграму. ER-діаграма зображена на рисунку 3.3.

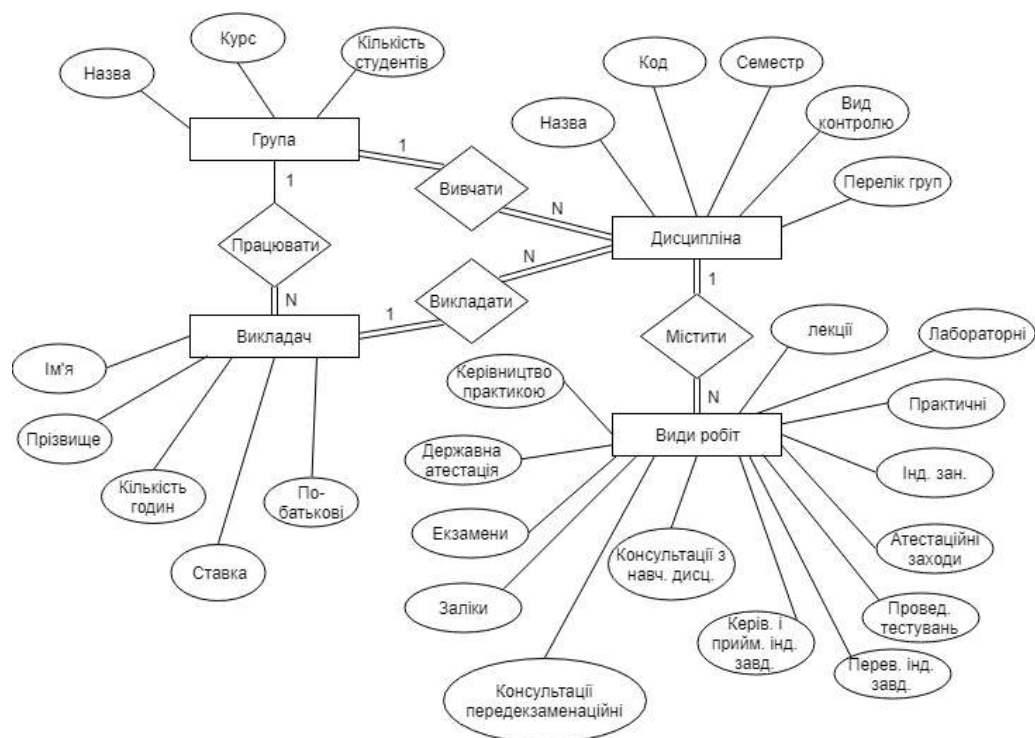


Рисунок 3.3 – ER-діаграма в нотатції Чена

Для створення бази даних програмного продукту були визначені наступні сутності: навчальні дисципліни та види навчальної роботи, а саме шифр циклу та назва дисципліни; курс навчання; аббревіатура груп; кількість студентів всього і в тому числі іноземних; кількість груп; кількість підгруп; кількість потоків. Далі йдуть дані, що передбачені робочим навчальним планом – це години, а саме лекції, практичні, семінарські заняття, лабораторні заняття, індивідуальні заняття і консультаційні роботи за розкладом, атестаційні заходи, практика (кредити). Також додаються заліки, підсумковий семестровий контроль, екзамени, додатковий семестровий контроль і індивідуальні заняття, а саме вид і кількість одиниць. Далі йде розрахунок годин, який включає в себе лекції, практичні семінарські заняття, лабораторні заняття, індивідуальні заняття або консультаційна робота за розкладом, атестаційні заходи, проведення тестування, перевірка індивідуальних завдань, керівництво і приймання індивідуальних завдань, консультації з навчальними дисциплінами протягом семестру, консультації передекзаменаційні, заліки, підсумковий семестровий контроль, екзамени, додатковий семестровий контроль, керівництво, консультування та рецензування кваліфікаційних проектів, державна атестація, 35, керівництво практикою, інше. Також є загальна кількість годин і погодинна оплата. Також додаються прізвище, ім'я та по батькові викладача.

Проведемо нормалізацію бази даних. База даних у першій нормальній формі знаходить на рисунку 3.4.

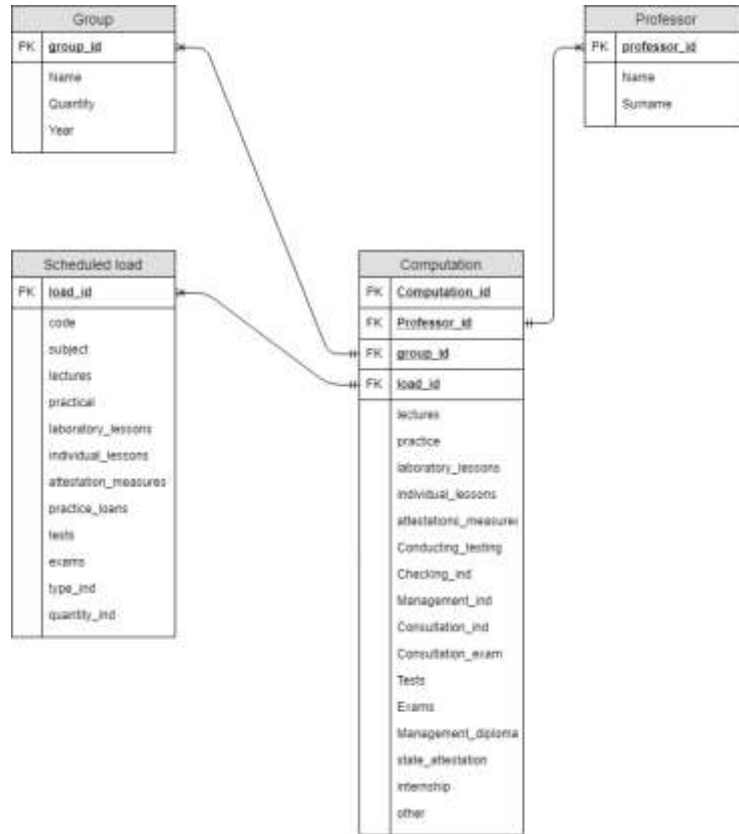


Рисунок 3.4 – Перша нормальна форма

Для нормалізації до другої і третьої нормальної форми необхідно додати додаткові таблиці для уникнення збитковості даних. Третя нормальна форма зображена на рисунку 3.5.

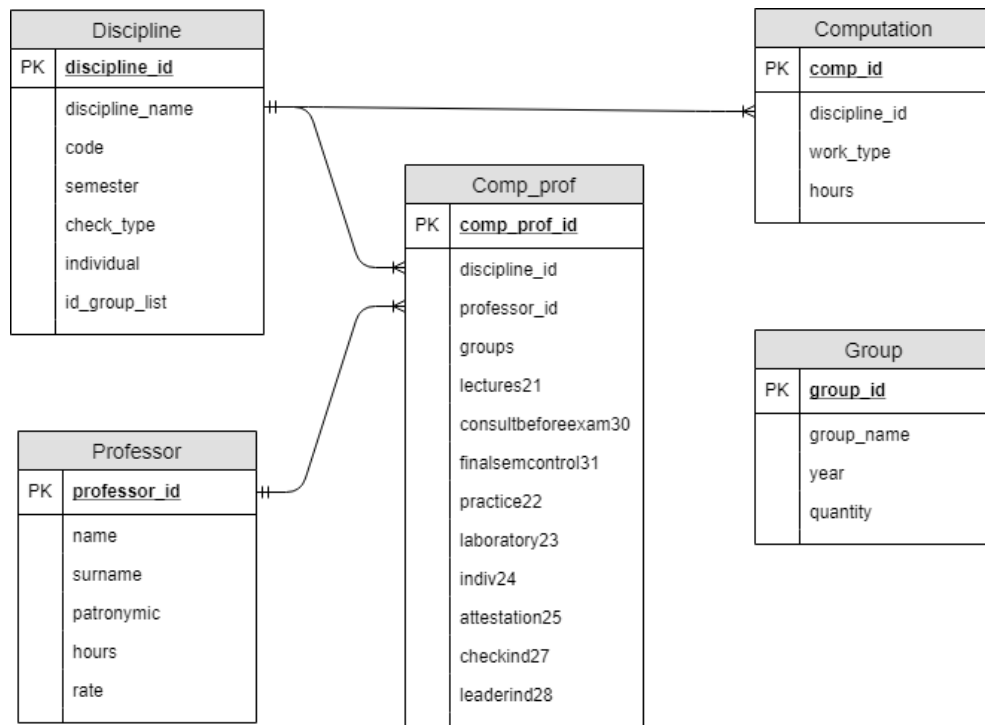


Рисунок 3.5 – Третя нормальна форма

У третій нормальній формі сутностями є group (група), professor (викладач), discipline (дисципліна), comp_prof (навантаження викладача), computation (розрахунок навантаження).

Сутність Group має такі атрибути: group_name (назва групи), quantity (кількість студентів у підгрупі), year (курс).

Сутність Professor має наступні атрибути: name (ім'я викладача), surname (прізвище викладача), patronymic (по-батькові викладача), hours (кількість робочих годин), rate (ставка).

Сутність Discipline у якості атрибутів має code (код дисципліни) і discipline_name (назву дисципліни), semester (семестр), check_type (вид контролю), individual (вид індивідуального завдання), id_group_list (перелік id груп).

Сутність Computation має атрибути hours (кількість годин навантаження), work_type (тип робіт (лекції, практики і т.п.)).

Сутність Comp_prof має наступні атрибути: group (id групи або декількох груп на даній дисципліні), lectures21 (кількість лекційних годин),

consultbeforeexam30 (кількість годин на передекзаменаційні консультації), finalemcontrol31 (кількість годин на проведення екзаменів), practice22 (кількість годин на практичні заняття), laboratory23 (кількість годин на лабораторні заняття), indiv24 (кількість годин на індивідуальні заняття), attestation25 (кількість годин на проведення атестаційних заходів), checkind27 (кількість годин на перевірку індивідуальних завдань), leaderind28 (кількість годин на керівництво і приймання індивідуальних завдань).

3.3 Моделювання варіантів використання

Для кращого розуміння інформаційної системи виконаємо моделювання варіантів використання. Схема варіантів використання зображена на рисунку 3.6.

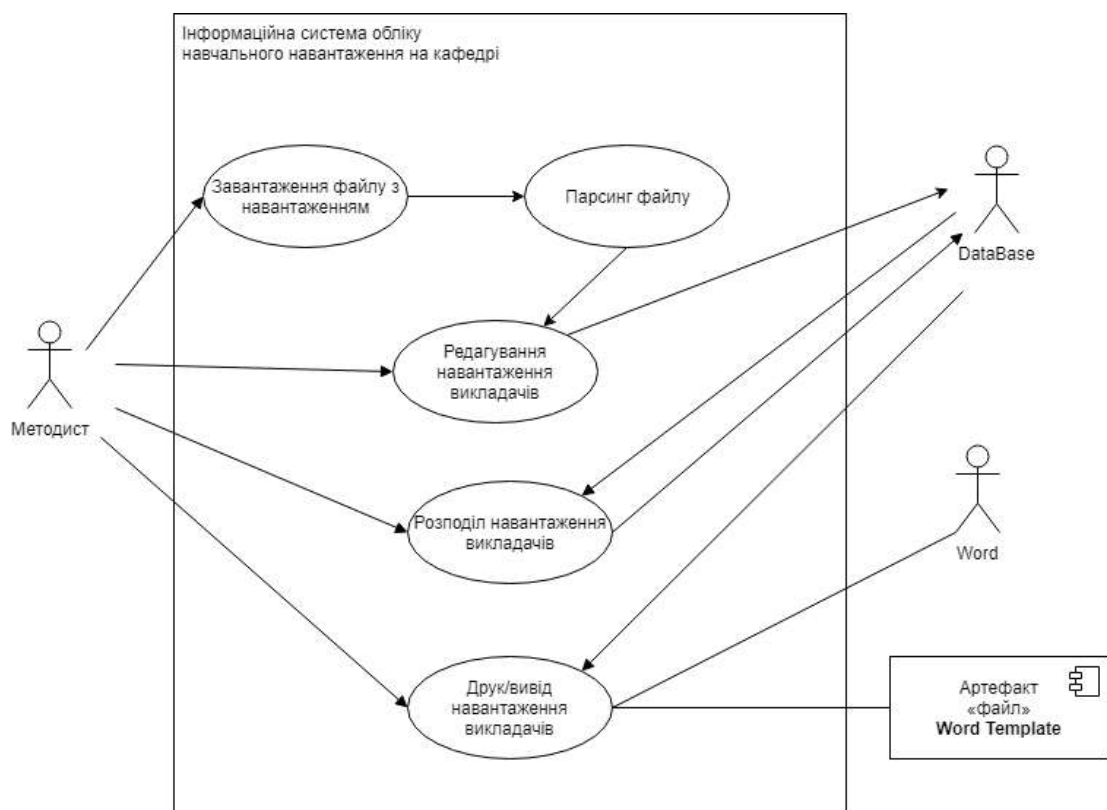


Рисунок 3.6 – Діаграма варіантів використання

Актором є методист кафедри, який взаємодіє з системою. Завантаження файлу з навантаженням – це процес занесення PDF-файлу з навантаженням методистом до системи. Парсинг файлу – це процес обробки файлу інформаційної системи з подальшим занесенням даних до бази даних. Розподіл навантаження викладачів – це процес виконання розподілу навантаження методистом. Після виконання розподілу дані завантажуються до бази даних. Редагування навантаження викладачів – це процес перевірки і редагування даних методистом в разі їх невідповідності, або неточності. Друк/вивід навантаження викладачів – це процес вибору методистом викладача і друк його навантаження.

3.4 Моделювання діаграми послідовності

Для кожного варіанту використання було створено діаграми послідовностей зображені на рисунку 3.7 – 3.11.

Діаграми наглядно демонструють динаміку взаємодії основних елементів системи між собою при виконанні тієї чи іншої дії.

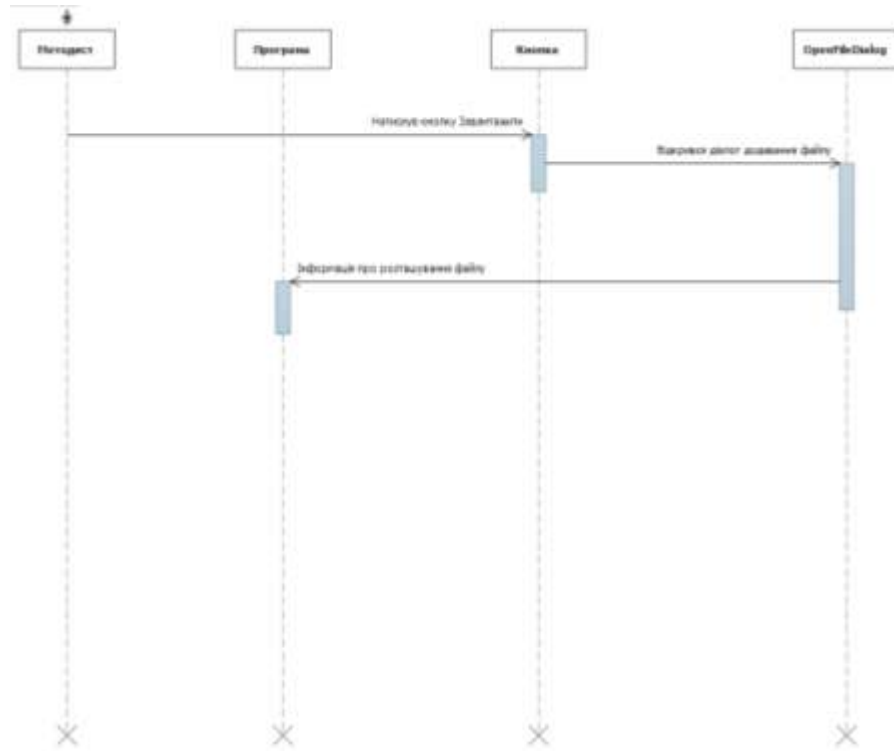


Рисунок 3.7 – Диаграмма последовательности «Загрузка файла с навантаження»

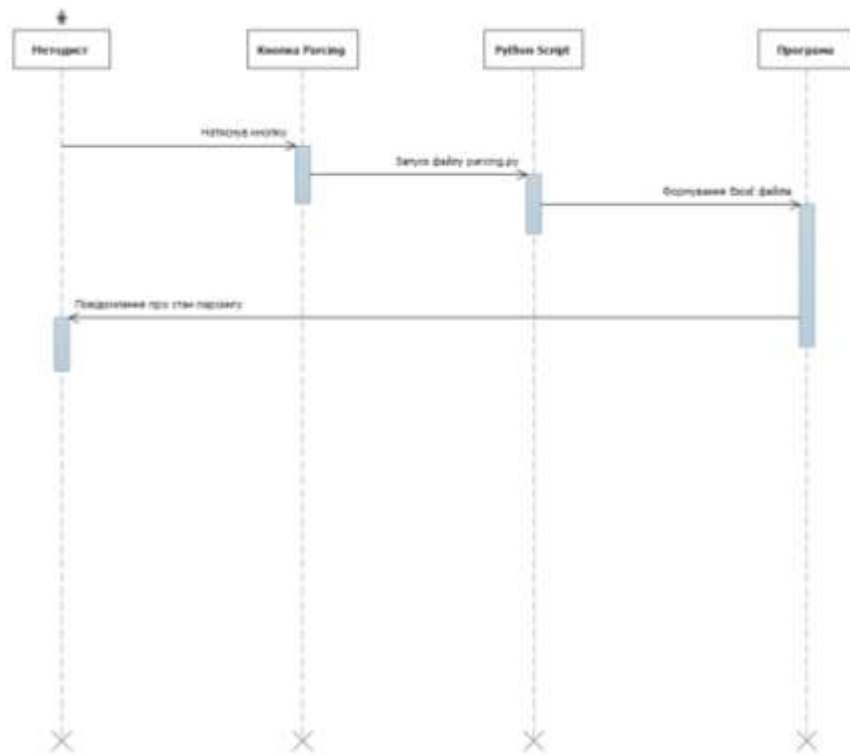


Рисунок 3.8 – Диаграмма последовательности «Парсинг файла»

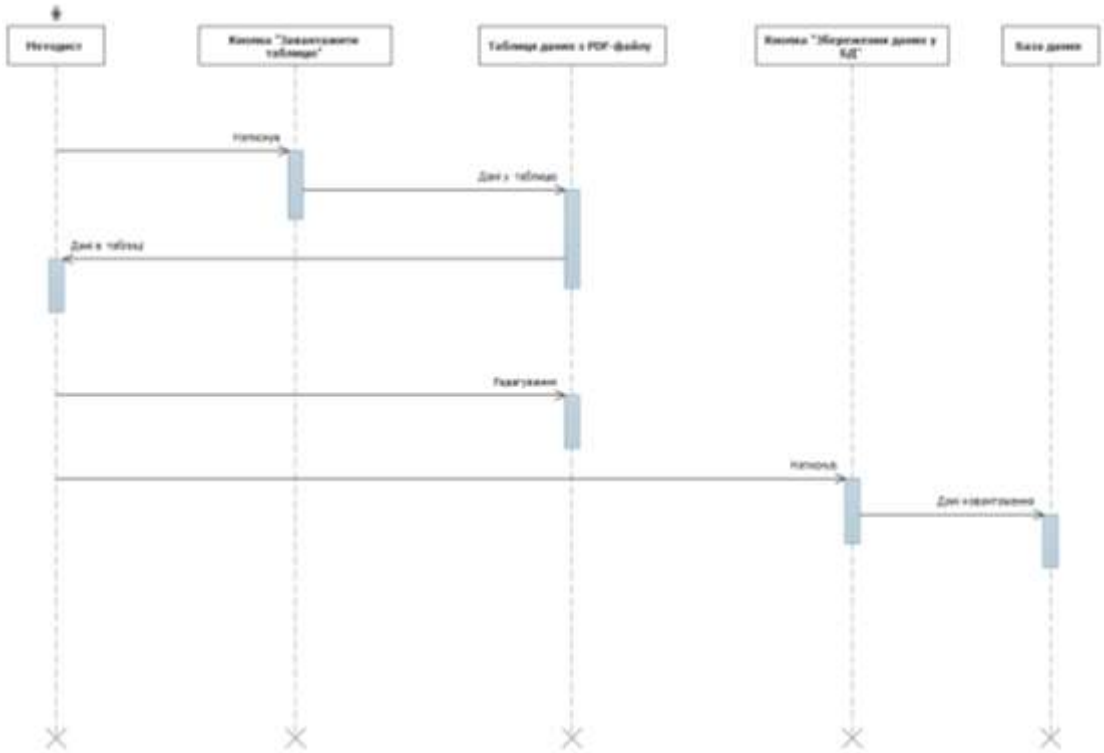


Рисунок 3.9 – Діаграма послідовності «Редагування навантаження викладачів»

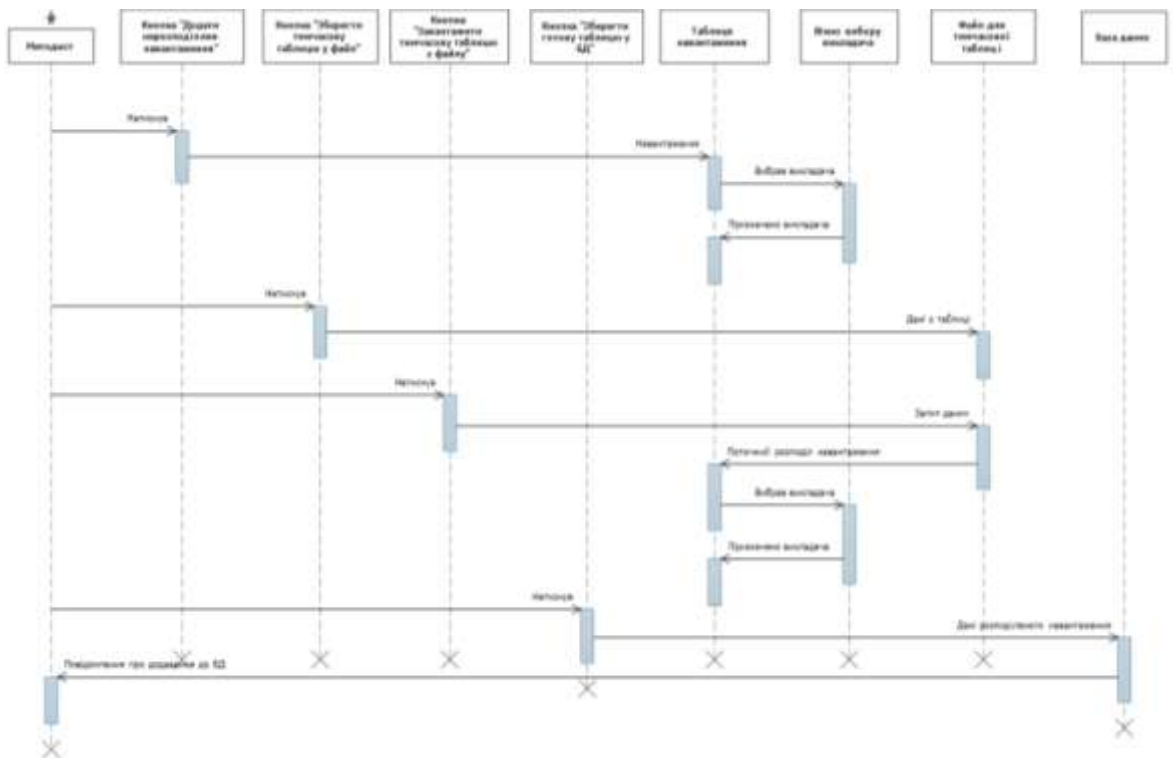


Рисунок 3.10 – Діаграма послідовності «Розподіл навантаження викладачів»

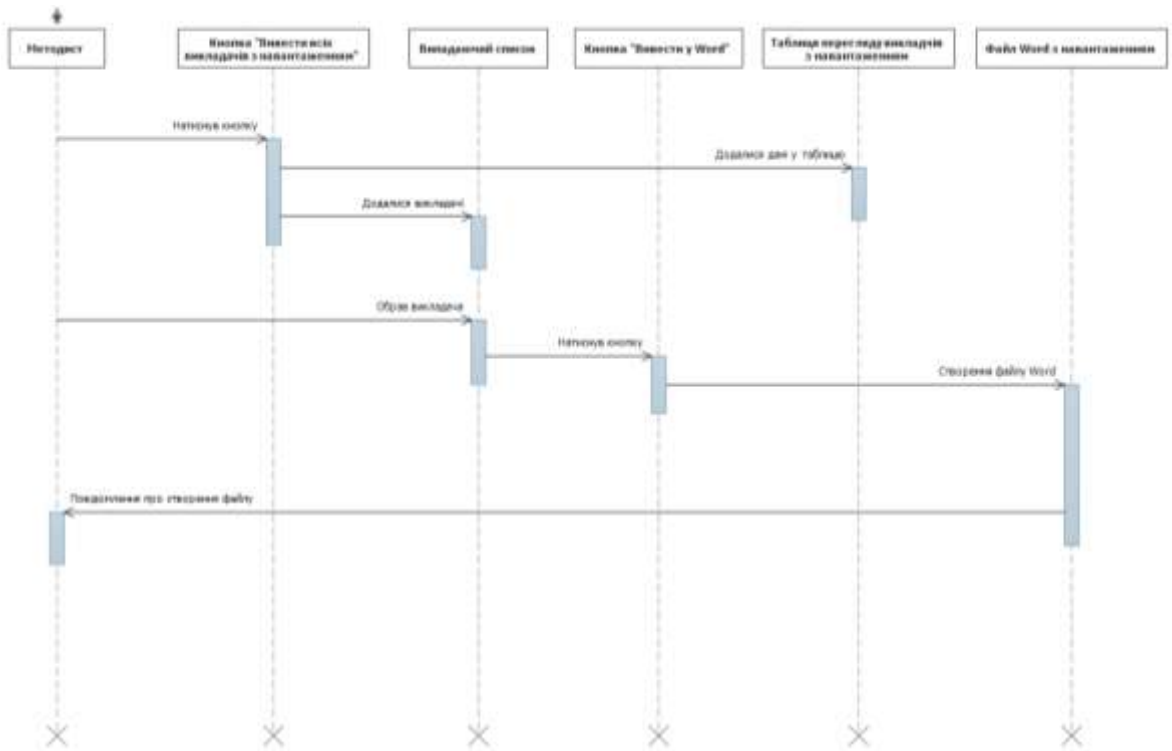


Рисунок 3.11 – Діаграма послідовності «Друк/вивід навантаження викладачів»

4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗПОДІЛУ НАВЧАЛЬНОГО НАВАНТАЖЕННЯ НА КАФЕДРІ

4.1 Реалізація фізичної моделі БД

Для застосування бази даних у нашому програмному продукті був вибраний локальний сервер Denwer. Засобом для адміністрування серверу був обраний PHPMyAdmin. Інтерфейс PHPMyAdmin зображений на рисунку 4.1.

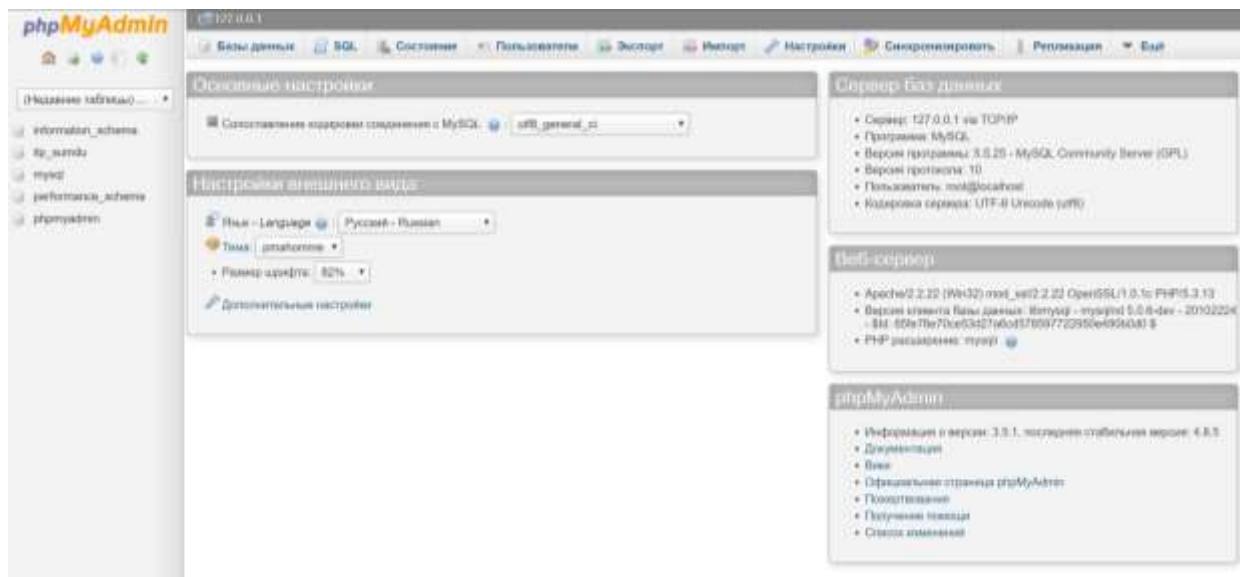


Рисунок 4.1 – Интерфейс PHPMyAdmin

Для роботи програмного засобу була створена база даних під назвою `itp_sumdu`. Після створення бази даних були додані таблиці: `computation`, `comp_prof`, `discipline`, `group`, `professor`. Розглянемо процес створення таблиці `Group`. Була створена нова таблиця з 4 стовпчиками. Перший стовпчик це

primary key нашої таблиці. Додаємо назву `group_id`, тип `INT`, обираємо індекс `primary`, а також обираємо `Auto increment`. Після цього заповнюємо інші поля – це `group_name` (назва групи) з типом `VARCHAR`, `year` (курс навчання) з типом `INT`, `quantity` (кількість студентів у групі) з типом `INT`. Скріншот процесу створення таблиці зображений на рисунку 4.2.

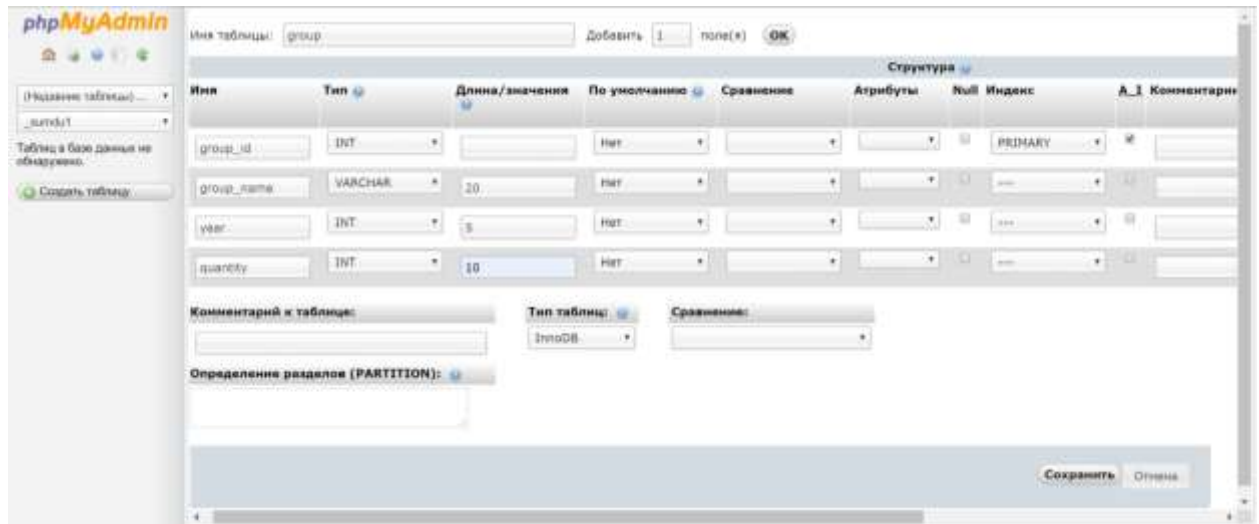


Рисунок 4.2 – Створення таблиці Group

Всі інші таблиці створюються таким же чином. Після створення всіх таблиць зв'язуємо між собою потрібні вторинні ключі. Процес зв'язування ключів між собою показаний на рисунку 4.3.

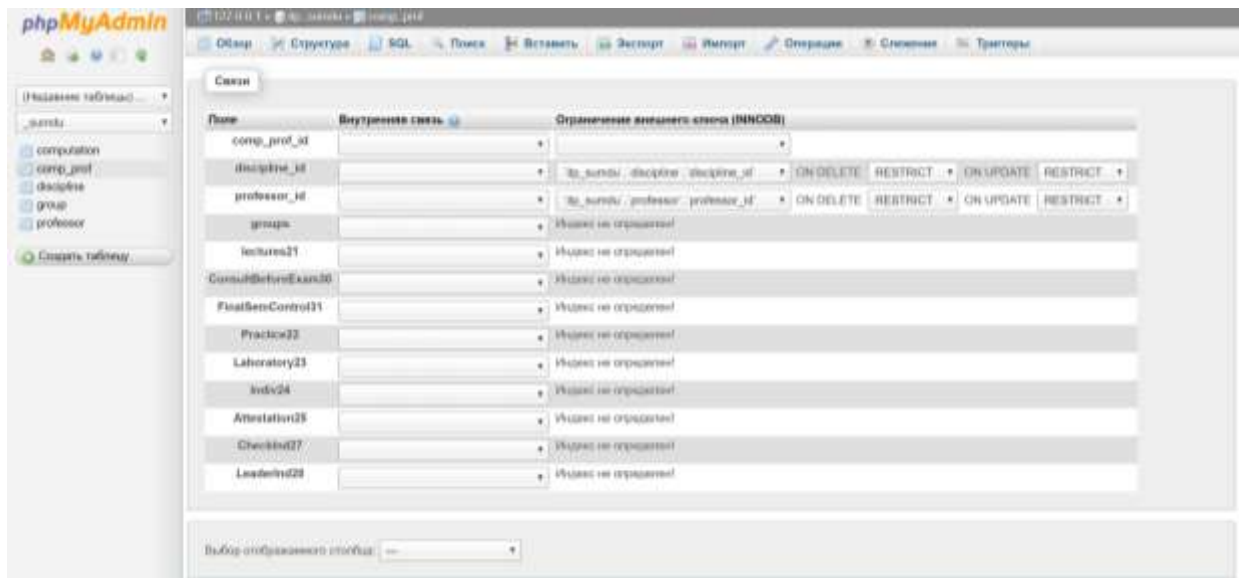


Рисунок 4.3 – Процесс пов'язання вторинних ключів

Після закінчення визначення всіх зв'язків між ключами ми отримуємо необхідну базу даних, що відповідає розробленим діаграмам. На рисунку 4.4 зображена структура бази даних.

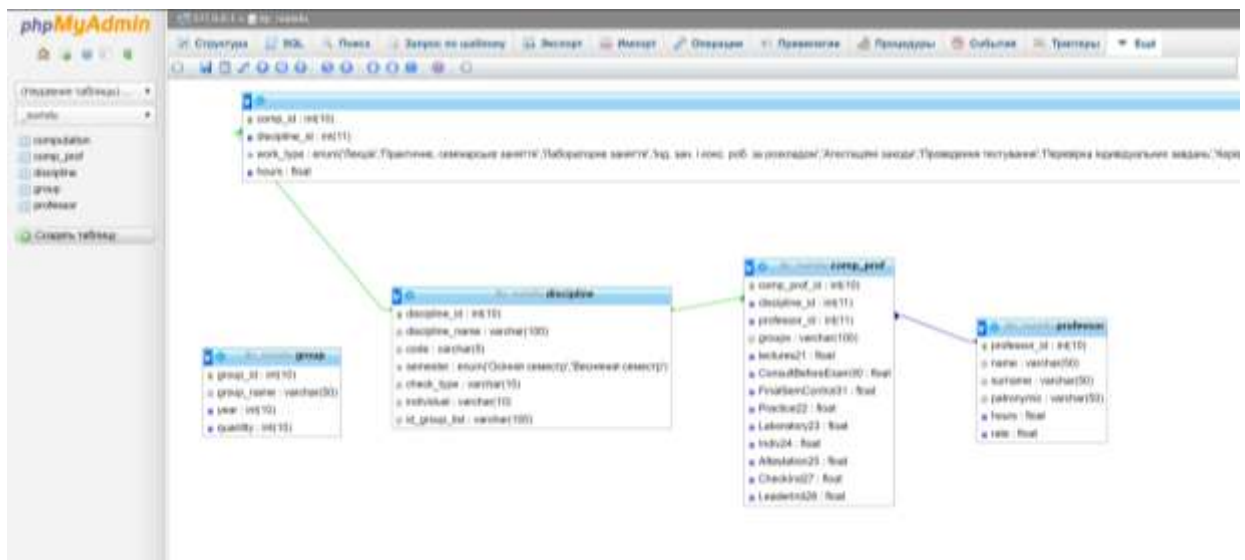


Рисунок 4.4 – Структура бази даних

4.2 Загальна реалізація інтерфейсу

Програмне забезпечення виконується як Windows – додаток. Буде передбачено одне головне вікно, яке міститиме весь необхідний функціонал.

Для реалізації функції занесення PDF-файлу у програму і налаштувань програми було використано меню. Інструментом є MenuStrip. У це меню було додано дві кнопки – це «Файл» і «Налаштування».

Елемент «Файл» включає в себе кнопку «Завантажити». Після натискання кнопки «Завантажити» відкривається діалогове вікно вибору файлу, де користувач обирає PDF-файл з навантаженням і після цього він зберігається у програмі.

Елемент «Налаштування» має дві кнопки – це «Коеф. ставки» і «Сервер». Кнопка «Коеф. ставки» відкриває діалогове вікно де користувач може ввести коефіцієнт ставки, який пізніше буде використовуватися для розрахунку ставки викладача. Кнопка «Сервер» відкриває діалогове вікно де користувач вводить дані сервера, до якого буде підключатися програма.

У програмному додатку використовується технологія закладок. Інструментом є елемент tabControl. У процесі розробки програми було створено 5 закладок: «Робота з PDF-файлом», «Додавання викладача», «Редагування груп», «Розподіл навантаження», «Вивід навантаження у Word».

Перша закладка це «Робота з PDF-файлом». Це головна закладка програми, вона є відкритою при завантаженні програми. У робочій області цієї закладки були додані 3 кнопки (button) і одна таблиця для перегляду даних (dataGridView). Процес налаштування вкладки зображений на рисунку 4.5.

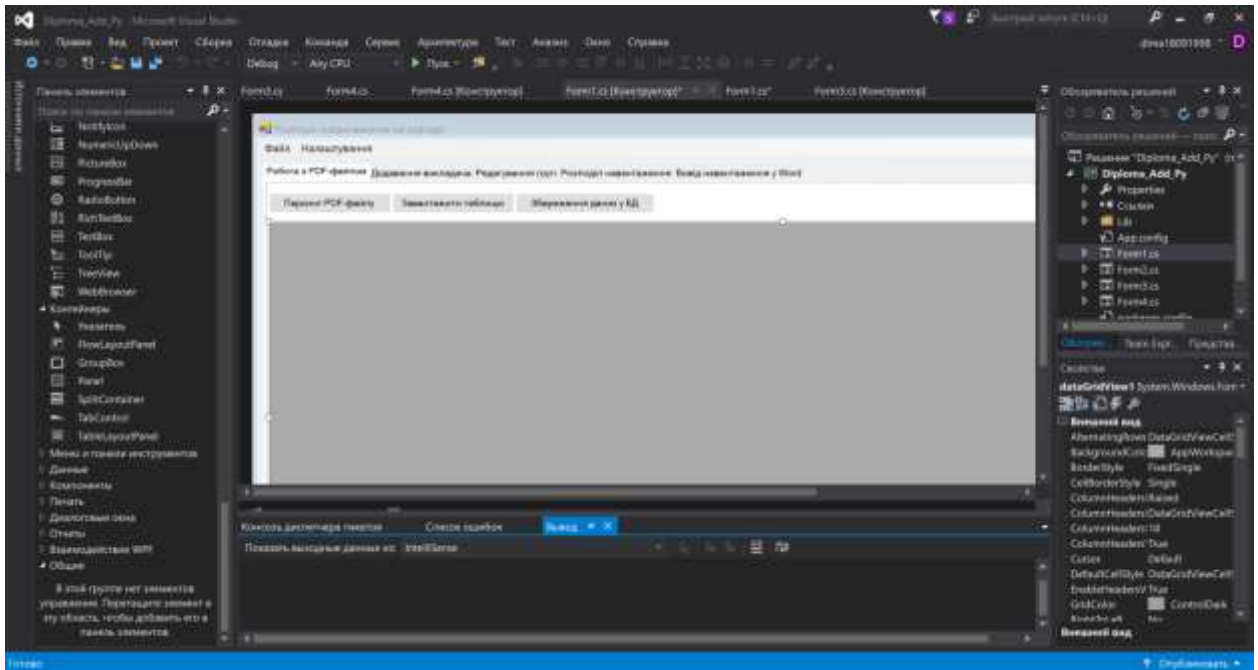


Рисунок 4.5 – Налаштування вкладки «Робота з PDF-файлом»

Кнопка «Робота з PDF-файлом» - це «Парсинг PDF-файлу». При натисканні на цю кнопку запускається модуль парсингу і виконується парсинг. Після завершення парсингу користувачу виводиться повідомлення про його вдалість або не вдалість.

Кнопка «Завантажити таблицю» виконує вивід відпаршеної інформації у таблицю, де користувач може переглянути результати парсингу і в разі їх некоректності відредагувати. Так як скрипт, який виконує парсинг PDF-файлу, парсить окремо кожну сторінку файлу PDF і виводить декілька файлів таблиць Excel, тому потрібно їх об'єднати між собою. Для об'єднання таблиць використовується бібліотека Spire.Xls.

Наступна вкладка це «Додавання викладача». В ній користувач може ввести прізвище, ім'я і по-батькові викладача і занести ці дані у базу даних.

Вкладка «Редагування груп» дозволяє користувачеві обрати групу із бази даних і додати до неї кількість студентів.

Вкладка «Розподіл навантаження» включає в себе 4 кнопки і таблицю для роботи з даними навантаження. У цій вкладці користувач виконує розподіл навантаження між викладачами. Натискаючи на кнопку «Додати нерозподілене

навантаження» користувач заповнює таблицю даними з БД. Далі натискаючи на комірку у стовпці «Викладач» відкривається діалогове вікно, де користувач обирає одного з викладачів, що занесені до БД. Також у цій вкладці є можливість збереження тимчасових результатів розподілу, коли у відповідності до всіх предметів не поставлені викладачі. Зберігання готової таблиці з розподіленим навантаженням відбувається при натисканні на кнопку «Зберегти готову таблицю у БД».

Вкладка «Вивід навантаження у Word». На цій вкладці користувач натискаючи на кнопку «Вивести всіх викладачів з навантаженням» у таблицю виводяться всі викладачі, їх сумарна кількість годин і ставка. Також є можливість обрати викладача і вивести його навантаження у файл Word.

4.3 Модуль парсингу PDF-файлів

Після натискання на кнопку «Парсинг PDF-файлу» програма виконує запуск скрипту, який написаний на мові Python. Цей скрипт використовує додатковий безкоштовний модуль Camelot. Розглянемо головні фрагменти коду.

Для початку ми імпортуємо модуль Camelot у наш файл для використання його методів: `import Camelot`.

Далі ми створюємо змінну під назвою `tables` якій присвоюємо метод Camelot `read_pdf()`. У цей метод ми передаємо назву файлу навантаження, який методист додає до програми і другим параметром ми вказуємо, що виконуємо парсинг усіх сторінок файлу: `tables=camelot.read_pdf('ITP.pdf', pages='all').[21]`

Наступним етапом є створення циклу виводу `for` для експорту обробленої інформації з PDF-файлу:



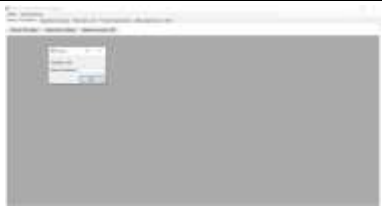
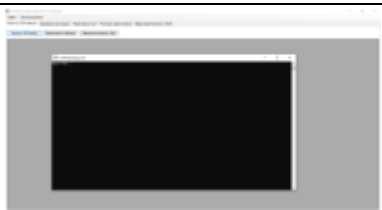


```
for index in range(len(tables)):
```


Таблиця 4.1 – Параметри комп'ютера, на якому проводилось тестування

Назва параметру	Значення	Мінімальне значення, задане в інструкції системного адміністратора
Процесор	Intel Core i5-6200U @2.30 ГГц	Одноядерний процесор з тактовою частотою 1 ГГц
ОЗУ	16 Гб	512 Мб
Графічний адаптер	Intel HD Graphics 520	-
Роздільна здатність екрану	3200x1800	-
Операційна система	Windows 10 Pro	Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Xp
Microsoft .NET Framework 4.0	+	+

- 1) Інсталятор передбачений.
- 2) Під час запуску програми помилок не виявлено. Зчитані вхідні дані відображуються коректно.
- 3) Тестування виконувалось у відповідності до керівництва користувача програми. Інструкція надає розуміння виконання усіх функцій програми.
- 4) Внутрішня довідка в програмі відсутня.
- 5) Реалізований в програмі інтерфейс інтуїтивно зрозумілий.
- 6) В ході тестування було перевірено функціонал програми. Результати наведені в табл. 4.2.


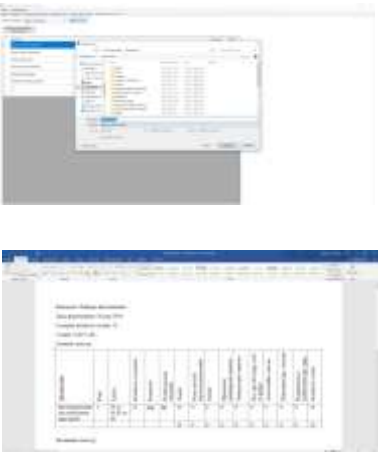
Таблиця 4.2 – Перелік виконаних тестів

Тест (номер та коротка назва)		Дії під час проходження тесту	Очікуваний результат	Результат (пройдений/не пройдений)
1.	Запуск програми	Запуск програми		Пройдений
2.	Виконання налаштування серверу	Натискання «Налаштування» -> «Сервер»		Пройдений
3.	Виконання налаштування коефіцієнта ставки	Натискання «Налаштування» -> «Коеф. ставки»		Пройдений
4.	Парсинг PDF-файлу	Натискання «Парсинг PDF-файлу»		Пройдений
5.	Завантажити таблицю	Натискання «Завантажити таблицю»		Пройдений
6.	Збереження даних у БД	Натискання «Збереження даних у БД»		Пройдений

Продовження таблиці 4.2

Тест (номер та коротка назва)		Дії під час проходження тесту	Очікуваний результат	Результат (пройдений/не пройдений)
7.	Додавання викладача	Натискання «Додати дані до БД»		Пройдений
8.	Редагування груп	Натискання «Зберегти у БД»		Пройдений
9.	Додавання нерозподіленого навантаження	Натискання «Додати нерозподілене навантаження»		Пройдений
10	Збереження тимчасової таблиці	Натискання «Зберегти тимчасову таблицю у файл»		Пройдений
11	Завантаження тимчасової таблиці	Натискання «Завантажити тимчасову таблицю»		Пройдений

Продовження таблиці 4.2

Тест (номер та коротка назва)		Дії під час проходження тесту	Очікуваний результат	Результат (пройдений/не пройдений)
12.	Додавання викладача до предмету	Натискання на комірку викладача у таблиці		Пройдений
13.	Збереження розподіленого навантаження у БД	Натискання «Зберегти готову таблицю у БД»		Пройдений
14.	Перегляд навантаження викладачів	Натискання «Вивести всіх викладачів з навантаженням»		Пройдений
15.	Вивід навантаження викладача у Word	Натискання «Вивести у Word»		Пройдений

При тестуванні даного програмного продукту не було виявлено помилок під час запуску, тобто вхідні дані були завантажені та відображались коректно. Інтерфейс простий для розуміння, тобто інтуїтивно зрозумілий, тому складностей при роботі з ним не виявлено. Даний програмний продукт буде корисним у вищих навчальних закладах для співробітників, які мають справу з розподілом навантаження викладачів.

ВИСНОВКИ

В результаті виконання дипломної роботи розроблено інформаційну систему обліку навчального навантаження на кафедрі.

У аналізі предметної області був розглянутий процес обліку навчального навантаження на поточний час. Виконаний огляд існуючих продуктів на ринку показав, що для автоматизації процесу навчального навантаження на кафедрі потрібно створити новий програмний продукт, через відсутність готових рішень у даній тематиці, які б задовольняли поставленим задачам.

Після прийняття рішення по розробці програмного продукту у постановці задачі було детальніше розглянуто мету та задачі програмного продукту, створено схему роботи та складено технічне завдання на виконання робіт. Розглянуто метод парсингу файлів. Проведено порівняльну характеристику СКБД і мов програмування. По результатам характеристик було прийнято рішення використати для створення програмного додатку мову програмування C#, а для роботи з базами даних використати MySQL. Для виконання парсингу PDF-файлів було обрано використати модуль написаний на мові Python.

Для забезпечення досягнення поставленої в роботі мети виконано планування робіт, визначено календарний графік та проведено аналіз можливих ризиків. Процес роботи програми описаний за допомогою функціональних діаграм нотації IDEF0, а також моделі варіантів використання і діаграм послідовностей.

По створеним моделям було виконано реалізацію програмного продукту. Модуль парсинг було реалізовано за допомогою мови програмування Python, а реалізація діалогових вікон і всього іншого функціонала програми була виконана за допомогою C#. Результатом реалізації стало програмне

забезпечення, що відповідає всім функціональним вимогам,ю які висувалися в технічному завданні.

Після створення програмного продукту було проведено його тестування і створено інструкцію користувача.

Створений програмний продукт дозволяє автоматизувати процес розподілу навчального навантаження на кафедрі, який виконує методист і заощадити його час на розподіл навантаження між викладачами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тищенко Д.В. Автоматизація обліку навчального навантаження на кафедрі. / Тищенко Д.В., Ващенко С.М. // Інформатика, математика, автоматика : матеріали та програма науково-технічної конференції, м. Суми, 2019 р. – Суми : СумДУ, 2019.
2. Adobe. What is PDF? [Електронний ресурс] / Adobe – Режим доступу до ресурсу: <https://acrobat.adobe.com/us/en/acrobat/about-adobe-pdf.html>.
3. Rouse M. Portable Document Format (PDF) [Електронний ресурс] / Margaret Rouse. – 2010. – Режим доступу до ресурсу: <https://whatis.techtarget.com/definition/Portable-Document-Format-PDF>.
4. Document managment - Portable document format - Part 1: PDF 1.7, 2008. – (Adobe System Incorporated).
5. What is a PDF file? [Електронний ресурс] – Режим доступу до ресурсу: <https://edu.gcfglobal.org/en/basic-computer-skills/what-is-a-pdf-file/1/>.
6. Syntax Analysis: Top Down, Bottom Up Parsing Types [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://www.guru99.com/syntax-analysis-parsing-types.html>.
7. Parse [Електронний ресурс]. – 2010. – Режим доступу до ресурсу: <https://www.dictionary.com/browse/parse>.
8. Aho A. The Theory of Parsing, Translation, and Compiling / A. Aho, J. Ullman.. – 542 с.
9. Document Capture And Data Extraction Made Easy [Електронний ресурс] – Режим доступу до ресурсу: <https://docparser.com/features#feature-wrapper>.
10. Погодинний розклад тижня [Електронний ресурс] – Режим доступу до ресурсу: <https://www.leadertask.ru/features>.

11. Aristarán M. Tabula [Електронний ресурс] / M. Aristarán, M. Tigas, J. Merrill – Режим доступу до ресурсу: <https://tabula.technology/>.
12. Драч В.Є., Родіонов А.В., Чухраєва А.І. Вибір системи управління базами даних для інформаційної системи промислового підприємства // Електромагнітні хвилі і електронні системи. – 2018.
13. Miles R. C# Programming Yellow Book / Rob Miles., 2015. С. 14.
14. STROUSTRUP B. A Tour of C++ / BJARNE STROUSTRUP., 2014. – (Pearson Education, Inc.).
15. Downey A. B. Think Java How to Think Like a Computer Scientist / Allen Downey., 2012.
16. Advantages of MongoDB | Disadvantages of MongoDB [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://data-flair.training/blogs/advantages-of-mongodb/>.
17. Why should you migrate from MySQL to MariaDB? [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: <https://mariadb.com/resources/blog/why-should-you-migrate-from-mysql-to-mariadb/>.
18. Harness the power of PDF [Електронний ресурс] – Режим доступу до ресурсу: <https://itextpdf.com/en>.
19. PDFBox [Електронний ресурс] – Режим доступу до ресурсу: <http://www.pdfbox.org/>.
20. Why Camelot? [Електронний ресурс] – Режим доступу до ресурсу: <https://camelot-py.readthedocs.io/en/master/>.
21. Mehta V. Camelot Documentation [Електронний ресурс] / Vinayak Mehta. – 2019. – Режим доступу до ресурсу: <https://buildmedia.readthedocs.org/media/pdf/camelot-py/master/camelot-py.pdf>.

ДОДАТОК А

ДОДАТОК Б**ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи «Облік навчального навантаження на
кафедрі»****Суми 2019**

1 Призначення й мета створення інформаційної системи

1.1 Призначення інформаційної системи

Інформаційна система повинна представляти систему обліку навчального навантаження секції Інформаційних технологій проектування Сумського державного університету

1.2 Мета створення інформаційної системи –

Автоматизація процесу обліку навчального навантаження секції Інформаційних технологій проектування

1.3 Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

1. Викладачі.
2. Співробітники.
3. Аспіранти.

2 Вимоги до інформаційної системи

2.1 Вимоги до інформаційної системи в цілому

2.1.1 Вимоги до структури й функціонування інформаційної системи

Інформаційна система повинна бути реалізована у вигляді програми для операційної системи Windows. Програмний додаток повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2 Вимоги до персоналу

Для підтримки і експлуатації програмного додатку від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером.

2.1.3 Вимоги до збереженні інформації

У системі керування програмою повинен бути передбачений механізм використання бази даних. Занесення даних до бази повинно виконуватися в автоматичному режимі.

2.2 Вимоги до функцій, виконуваних програмою

2.2.1 Основні вимоги

2.2.1.1 Структура програми

Програмний додаток повинен складатися з наступних розділів:

- Робота з PDF-файлом – парсинг PDF-файлу, перевірка і редагування даних після парсингу, занесення відредагованих даних до бази даних.
- Додавання викладача – додання ППП викладача до БД.
- Редагування груп – додання кількості студентів у групі по кожній групі.
- Розподіл навантаження – розподіл навантаження між викладачами кафедри з можливістю збереження поточного результату розподілу і занесення фінального результату до БД.
- Вивід навантаження у Word – вивід навантаження всіх викладачів у таблицю, виведення навантаження конкретного обраного викладача у Word-файл.

2.2.1.2 Навігація

Користувацький інтерфейс програмного додатку повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів. Навігаційні елементи повинні забезпечувати однозначне розуміння користувачем їх змісту: посилання на вікна програми повинні мати заголовки.

Система повинна забезпечувати навігацію по всіх доступних користувачеві ресурсам і відображати відповідну інформацію. Для навігації повинна використовуватися система вкладок. Вкладки повинні являти собою набір кнопок з прив'язаними до них робочими областями.

2.2.2 Вимоги до функціональних можливостей

2.2.2.1 Функціональні можливості розділів

- Автоматизоване завантаження даних з PDF-файлу навчального навантаження кафедри з можливістю їх редагування.
- Занесення інформації щодо поточного навантаження у базу даних системи.
- Додавання викладача.
- Розподіл навчальних доручень між викладачами.
- Облік навантаження по конкретному викладачу.

- Формування звіту по навантаженню конкретного викладача (перегляд та друк).
- Перегляд зведеної інформації по навантаженню викладачів кафедри.

2.2.2.2 Загальні вимоги

Розташування елементів у головному вікні програми схематично показано на рис. Б.1.

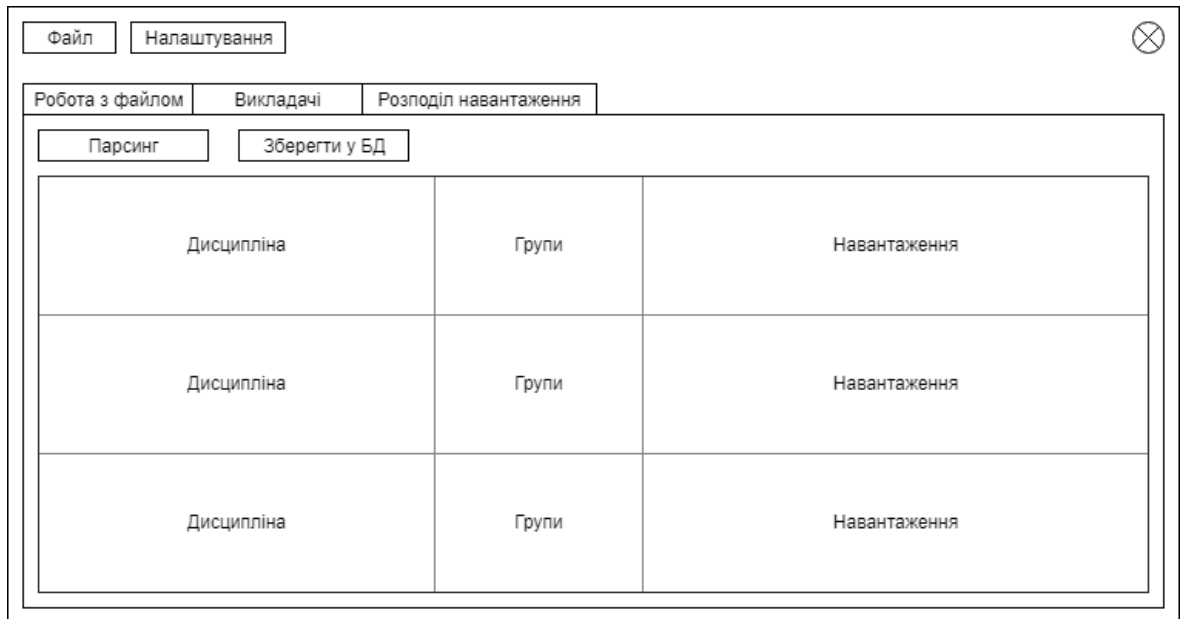


Рисунок Б.1 – Типова сторінка

2.2.2.3 Типові навігаційні й інформаційні елементи

- Меню.
- Робоча область.
- Вкладки.

2.2.2.4 Меню

Меню програми повинно містити кнопки для завантаження файлу та налаштувань програми.

2.2.2.5 Робоча область

Робоча область повинна містити необхідні елементи, такі як кнопки, поля вводу і виводу даних і т.п. для кожного розділу програми.

2.2.2.6 Вкладки

Вкладки при натисканні повинні відкривати необхідно робочу область програми.

2.3 Вимоги до видів забезпечення

2.3.1 Вимоги до інформаційного забезпечення

Реалізація програми відбувається з використанням:

- NET.Framework 4.0;
- IDE MS Visual Studio 2010;
- MySQL 5.6.
- Python 3.5
- Ghostscript 9.27 for Windows

2.3.2 Вимоги до лінгвістичного забезпечення

Програма повинна бути виконана українською мовою. Також повинна бути підтримка англійської мови для можливих назв дисциплін або груп студентів.

2.3.3 Вимоги до програмного забезпечення

Програмне забезпечення повинне задовольняти наступним вимогам:

- операційна система Windows 7, 8, 10;
- NET.Framework 4.0.

2.3.4 Вимоги до апаратного забезпечення

Апаратне забезпечення повинне задовольняти наступним вимогам:

- одноядерний процесор з тактовою частотою 1 ГГц і краще;
- оперативна пам'ять розміром 512 Мб і більше;
- не менш 500 МБ вільного місця на диску.

3 Склад і зміст робіт зі створення програми

Докладний опис етапів роботи зі створення програмного додатку наведено в табл. Б.1.

Таблиця Б.1 – Етапи створення програмного додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Розробка каркасу: Проектування розмітки та вигляду вікон програми	4 день
2	Додавання фалів: Додавання алгоритму занесення файлу до програмного додатку	5 день
3	Парсинг файлу: Розроблення та додавання алгоритму парсингу доданого PDF-файлу	23 днів
4	Робота з базою даних: Розробка алгоритму занесення, зміни і пошуку даних у базі даних	51 днів
5	Друк навантаження викладача: Розробка алгоритму пошуку і друку навчального навантаження конкретного викладача	9 дні
6	Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту	92

4 Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного програмного додатку вимогам сьогодення ТЗ і можливість його ефективної роботи, в організації Замовника повинні бути встановлені всі програми і драйвери перелічені у п. 2.2.3. Після цього встановлюється програмний додаток, який буде придатний до роботи.

ДОДАТОК В

Планування робіт

Деталізація мети проекту методом SMART. Мета проекту: створити програмний продукт, який дозволить автоматизувати роботу методиста кафедри та розподілу кафедри, в тому числі розподіл навантаження між викладачами кафедри. Мета є досяжною, підґрунтям створення програмного продукту є навички отримані на курсі з дисципліни ТСПП і IDE Microsoft Visual Studio 2010. Проект буде виконано вчасно, що підтверджується календарним планом проекту. Результати деталізації методом SMART розміщені у табл. В.1.

Таблиця В.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити програмний додаток для автоматизації обліку навчального навантаження кафедри
Measurable (вимірювана)	Результатом роботи проекту є оцінка замовника.
Achievable (досяжна)	Реалізації системи здійснюється за допомогою середовища розробки MS Visual Studio 2010, з використанням мови програмування C#
Relevant (реалістична)	У наявності є всі необхідні технічні та програмні засоби. Розробники достатньо кваліфіковані для виконання поставлених задач.
Time-framed (обмежена у	Ціль має часове обмеження. Робота повинна бути виконана у терміни, що були оговорені замовником проекту. Проект повинен бути виконаний згідно з

часі)	календарним планом.
-------	---------------------

Планування змісту структури робіт. Основним інструментом для планування змісту структури робіт служить WBS діаграма – графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов’язані з продуктом проекту. Побудуємо структуру WBS, у якій детально опишемо роботи, які потрібно виконати на кожному етапі створення проекту. Виконаємо декомпозицію робіт для даного проекту. Діаграма WBS зображена на рис. В.1.

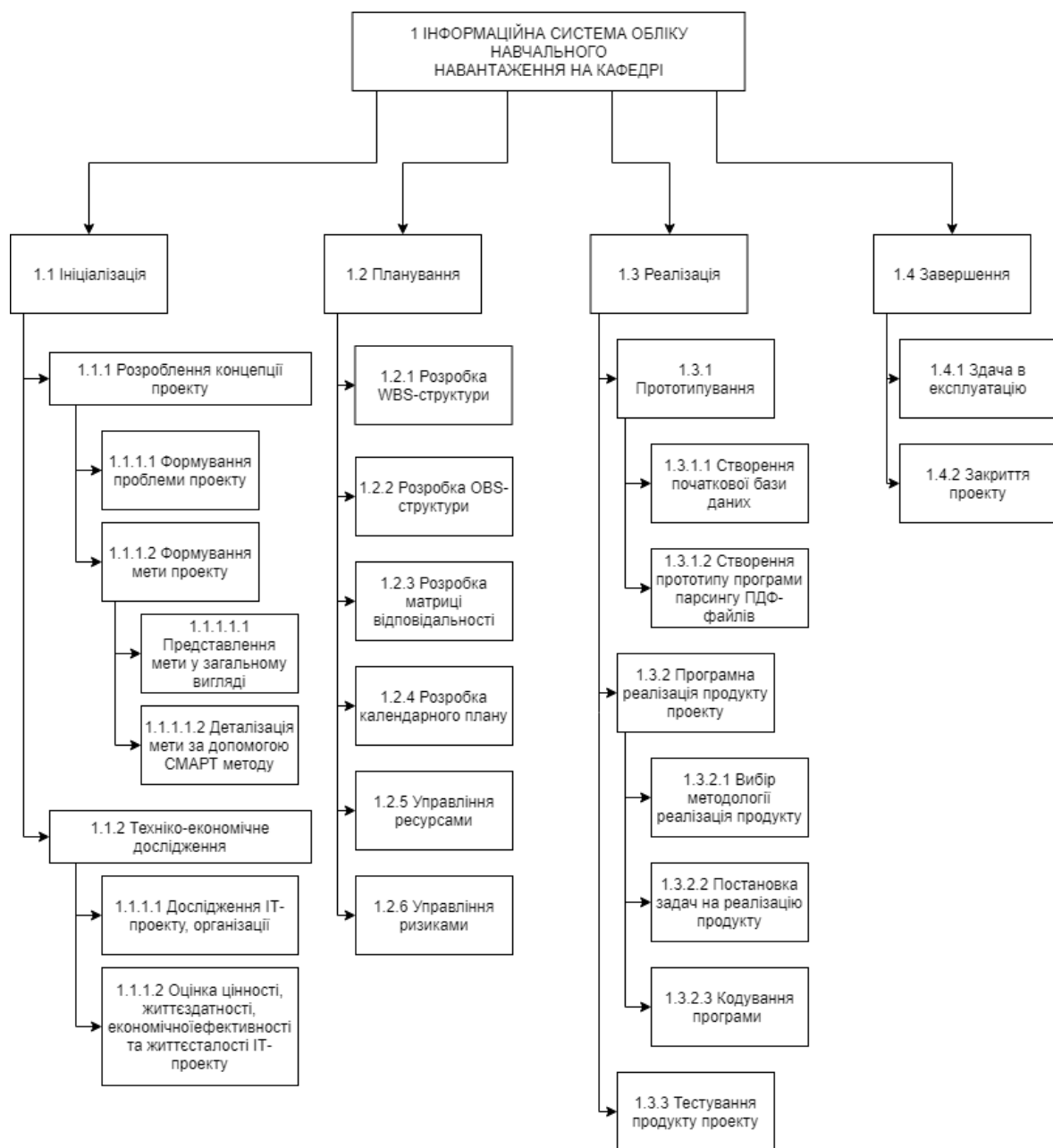


Рисунок В.1 – WBS. Структура робіт проекту

Планування структури організації, для впровадження готового проекту (OBS). Після побудови WBS розробимо організаційну структуру виконавців OBS. Організаційна структура проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Діаграма OBS зображена на рис. В.2. Список виконавців, що функціонують в проекті знаходиться в табл. В.2.

Таблиця В.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Виконавець	Тищенко Д. В.	Виконує розробку основного функціоналу проекту та інтерфейс користувача.
Керівник проекту	Ващенко С. М.	Формує завдання на розробку проекту та виконує введення проекту в експлуатацію

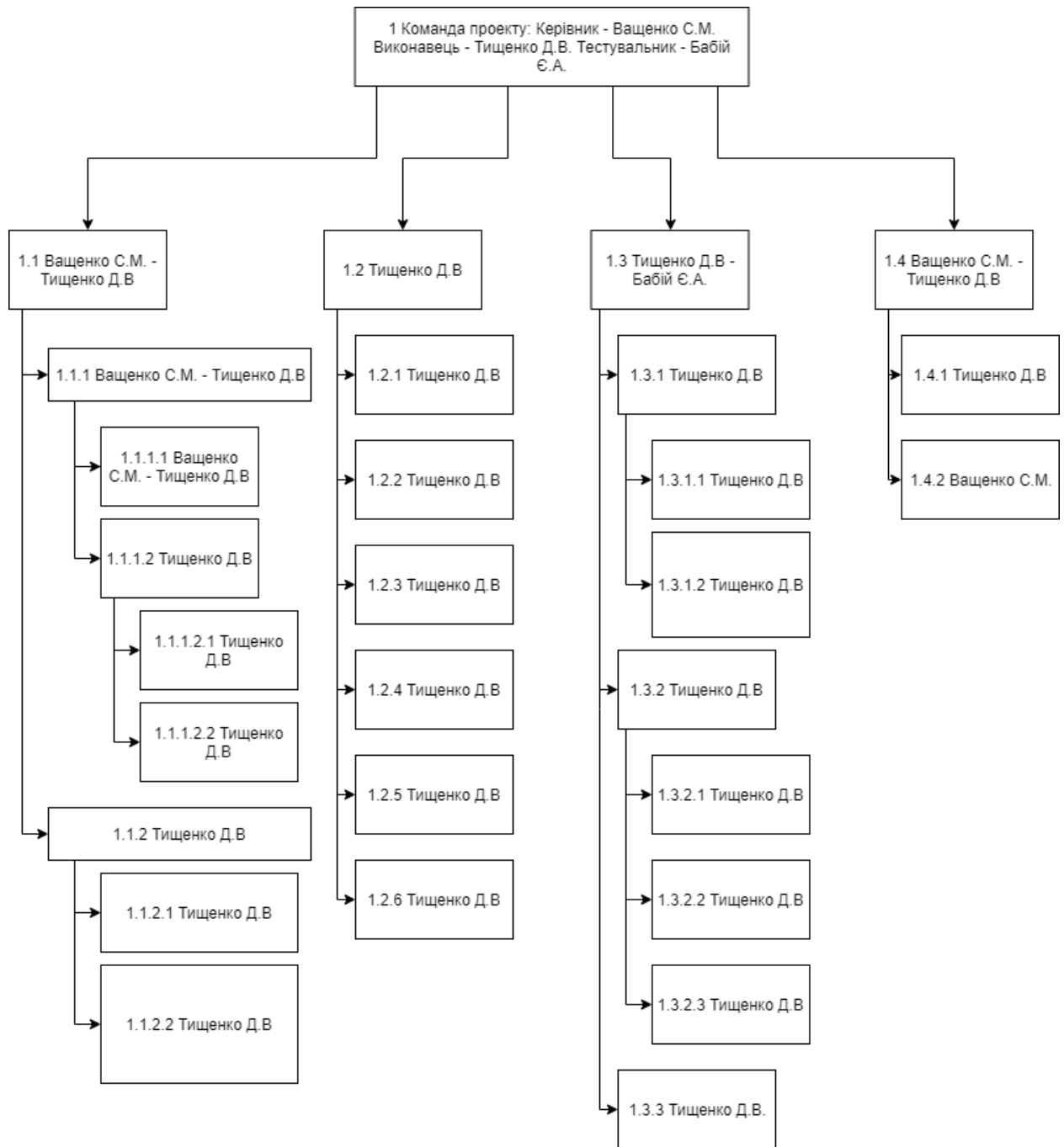


Рисунок В.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі, а також, проекту в цілому з урахуванням вихідних та святкових днів, будують календарний графік робіт, що називається Діаграмою Ганта. Діаграма Ганта зображена на рис. В.3-В.4.

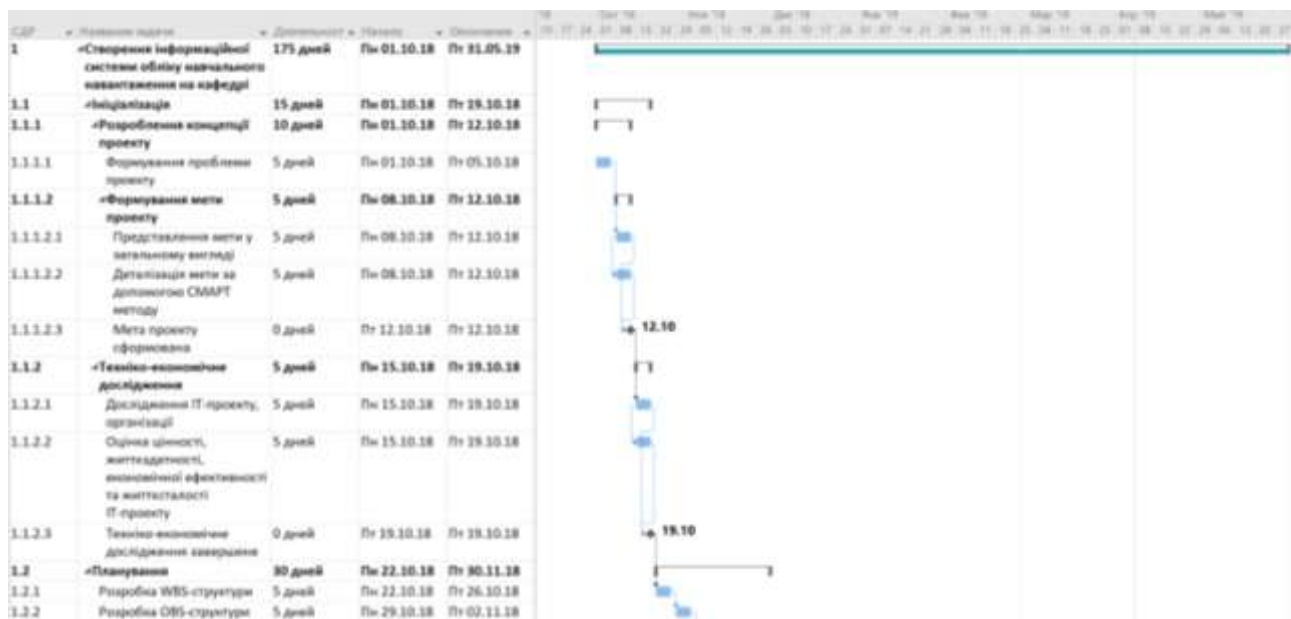


Рисунок В.3 – Діаграма Ганта



Рисунок В.4 – Продовження діаграми Ганта

Аналіз ризиків. Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така

оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків. У табл. В.5 знаходиться класифікація ризиків за показниками ймовірності виникнення ризику та величині втрат.

Далі виконаємо планування реагування на ризики — це розробка методів і технологій зниження негативного впливу ризиків на проект. Визначимо ефективність розробки реагування на проект, визначимо чи будуть наслідки впливу ризику на проект позитивними або негативним. Оцінюємо ризики за показниками, що знаходяться в табл. В.3. На основі оцінки будемо матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рис. В.7.

Таблиця В.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Ймовірність виникнення	3	RS_2	RS_3,	RS_5, RS_9	
	2	RS_1	RS_4	RS_7	
	1	RS_12	RS_8	RS_10, RS_15	
			1	2	3
			Вплив ризику		

Рисунок В.7 – Матриця ймовірності виникнення ризиків та впливу ризику

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. В.4. Оцінка ймовірності виникнення, впливу і рангу ризику наведена у таблиці В.5.

Таблиця В.4 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 2$	1,8,11,12,13
2	Виправдані	$3 \leq R \leq 4$	2,4,6,10,15
3	Недопустимі	$6 \leq R \leq 9$	3,5,7,9,14

Таблиця В.5 – Оцінка ймовірності виникнення, впливу і рангу ризику.

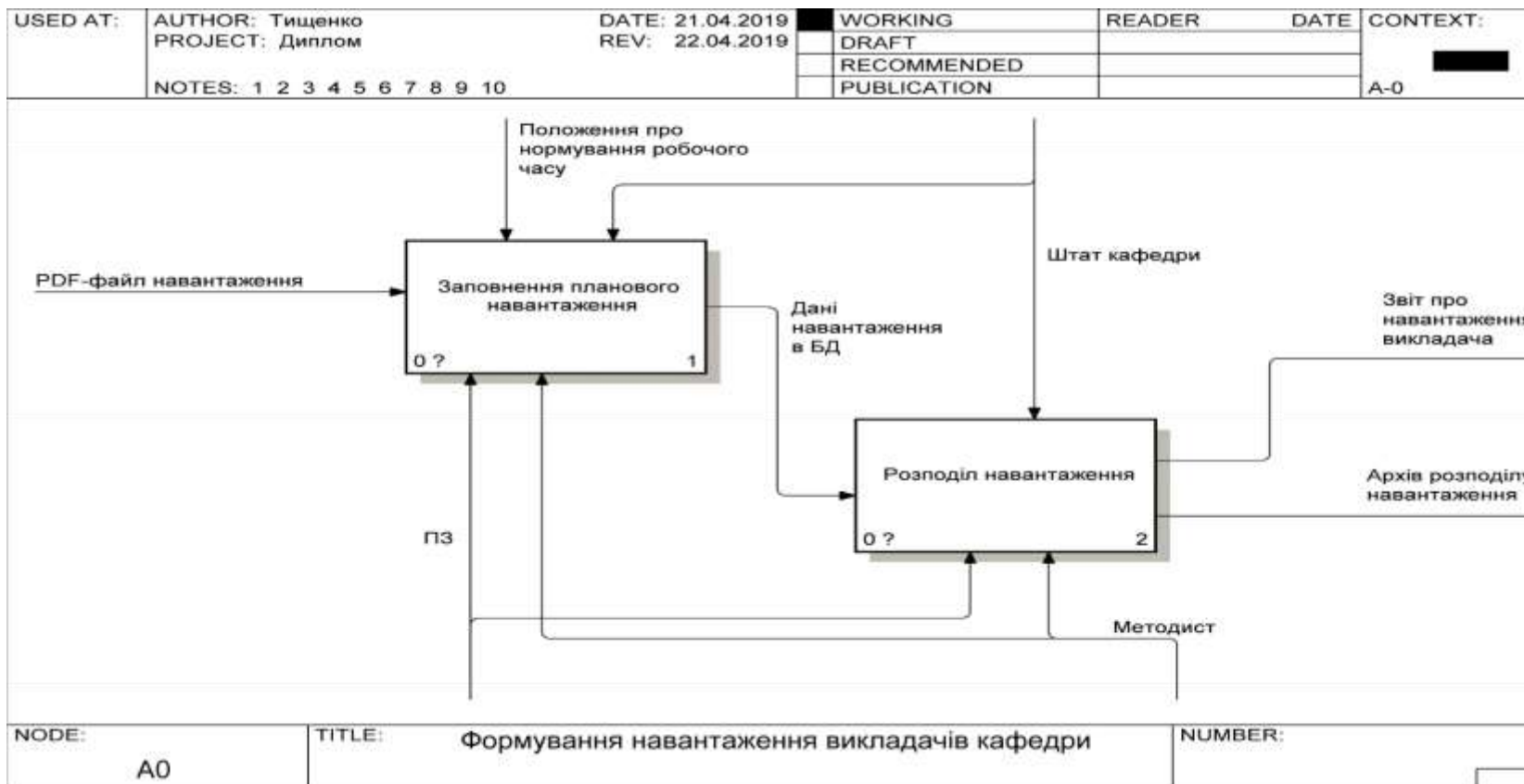
ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
S_1	Відкритий	Непорозуміння між розробником та замовником	Низька	Середній	2	<ol style="list-style-type: none"> Дотримуватися ділового етикету спілкування. Створити комфортні умови для співпраці 	Попередження	При виявленні проблеми непорозуміння потрібно з'ясувати, що саме стало причиною. Після знаходження причини її потрібно усунути
S_2	Відкритий	Поява альтернативного продукту	Низька	Високий	3	Провести аналіз ринку на наявність альтернативних продуктів	Прийняття	
S_3	Відкритий	Нечітке завдання на розробку	Середня	Високий	6	<ol style="list-style-type: none"> Ясно і однозначно обговорити із замовником усі види вимог і задач програмного продукту. 	Попередження	Уважно та чітко окреслити те, що було виконано невірно та зробити правки.

Продовження таблиці В.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_4	Відкритий	Низька кваліфікація розробників проекту	Середня	Середній	4	<p>1. Підвищити кваліфікацію персоналу.</p> <p>2. Переглянути онлайн-ресурси для підвищення рівня знань.</p>	Пом'якшення	<p>Врахувати час на підготовку працівників.</p> <p>Видати літературу, переглянути онлайн-уроки.</p>
RS_5	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	<p>Провести аналіз актуальності найважливіших процесів та робіт, розрахувати час необхідний для їхнього виконання.</p> <p>Створити календарний план з урахуванням резервного строку виправлення помилок і чітко його дотримуватися</p>	Пом'якшення	<p>Змінити порядок пріоритетів робіт. Знайти способи оптимізації роботи із вже існуючою розстановкою.</p> <p>Обговорити варіанти внесення правок до термінів реалізації із замовником.</p>

Продовження таблиці В.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_6	Відкритий	Не вірна оцінка масштабів проекту	Низька	Середній	2	Провести детальний аналіз проекту. Визначити основні етапу проекту, розподілити час на їх виконання. Проаналізувати масштаби проекту на основі додаткових джерел.	Пом'якшення	Переоцінка масштабів проекту. Перебудова стратегії реалізації проекту.
RS_7	Відкритий	Помилки розробки	Висока	Високий	9	На етапі розробки тісно співпрацювати із замовником та на певних етапах демонструвати поточні результати.	Пом'якшення	Здійснювати проміжний контроль результатів в ході виконання проекту.
RS_8	Відкритий	Збої в роботі програмного забезпечення	Низька	Високий	3	Залучити спеціаліста для усунення збоїв.	Попередження	Замінити програмне забезпечення.



ДОДАТОК Г

Рисунок Г.1 – Діаграма декомпозиції 1 рівня

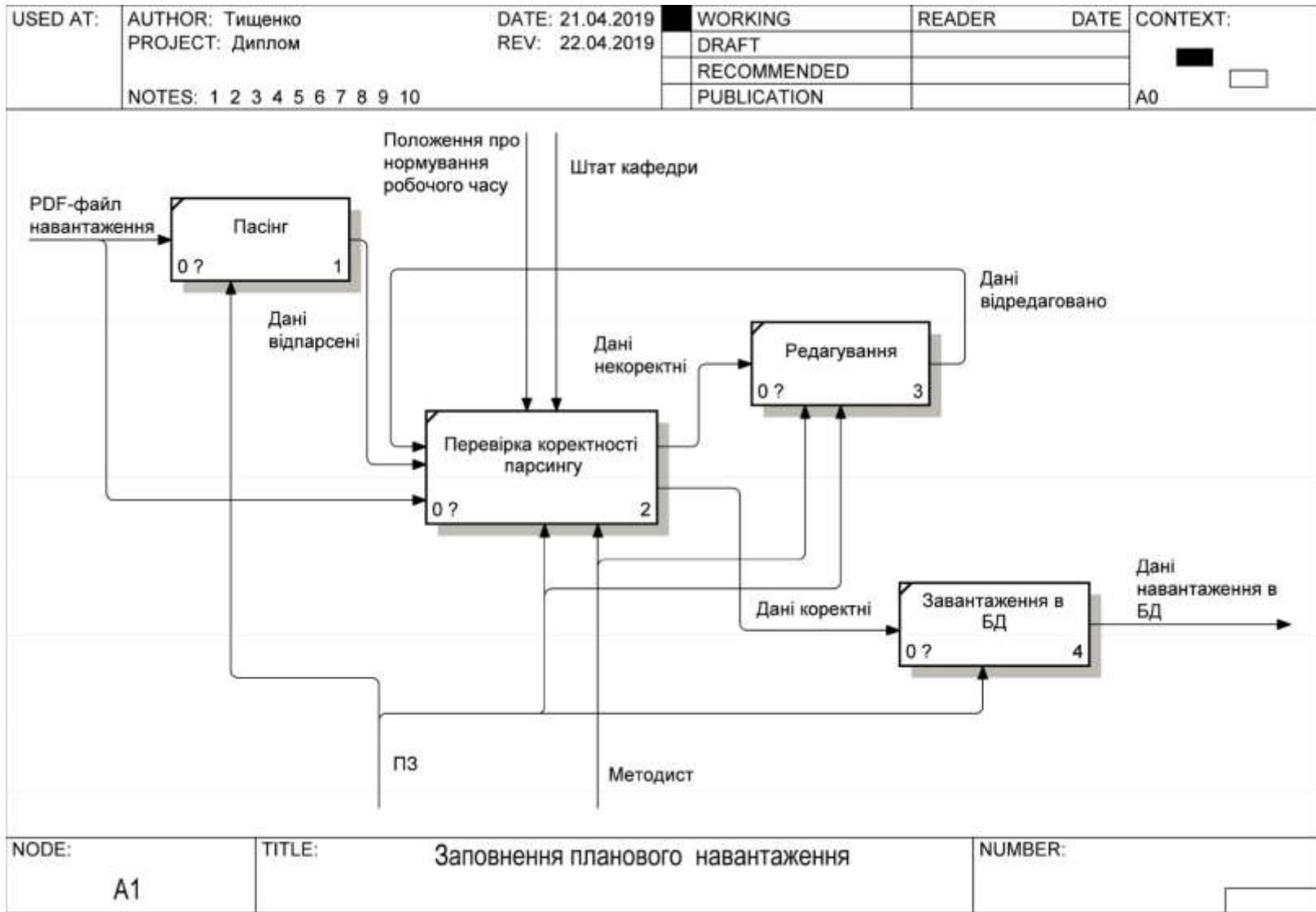


Рисунок Г.2 – Діаграма декомпозиції 2 рівня

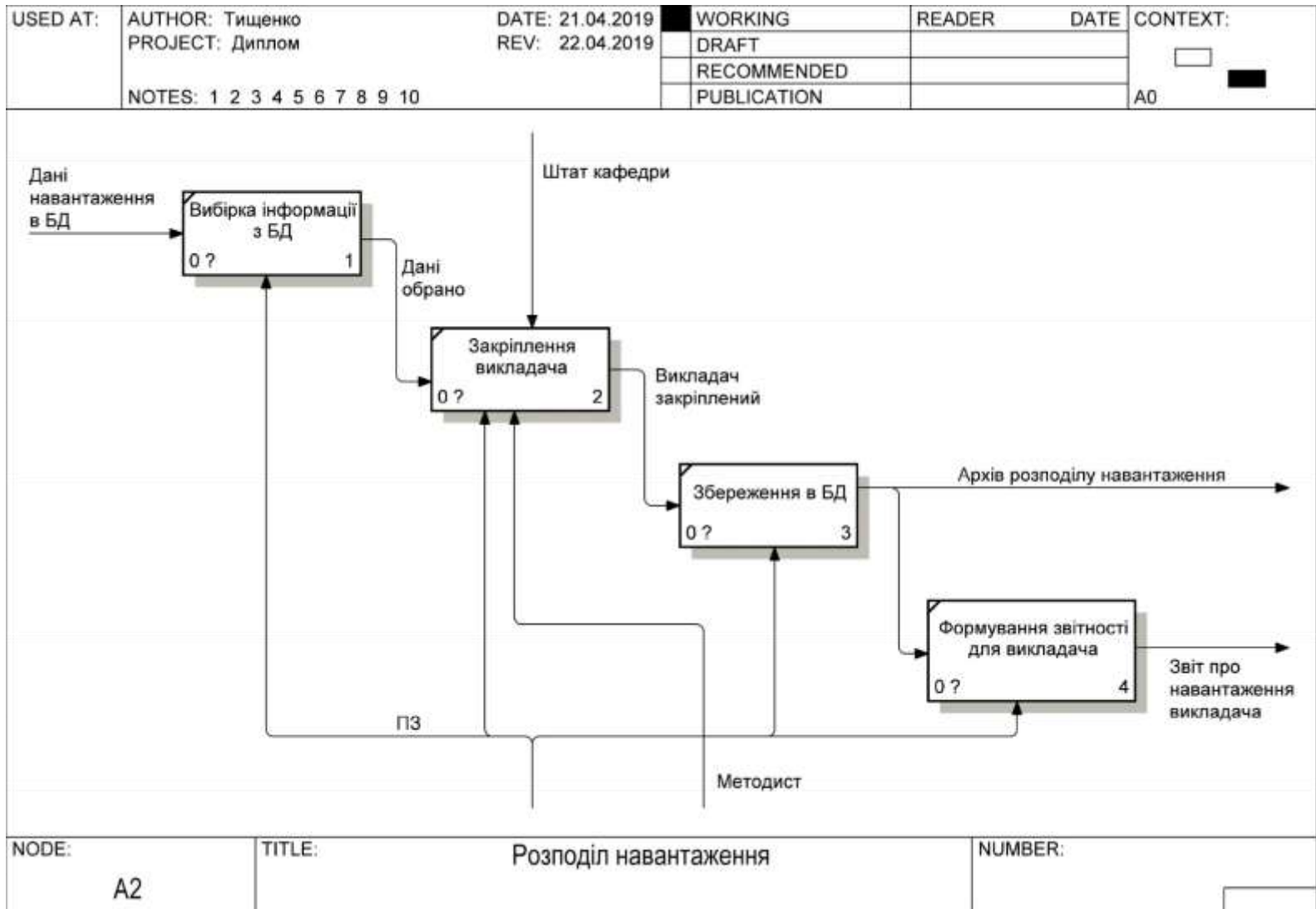


Рисунок Г.3 – Діаграма декомпозиції 3 рівня

ДОДАТОК Д

Лістинг програми

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Microsoft.Scripting.Hosting;
using System.IO;
using System.Diagnostics;
using Spire.Xls;
using Spire.DataExport;
using MySql.Data.MySqlClient;
using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.Word;
using System.Reflection;
namespace Diploma_Add_Py
{
    public partial class Form1 : Form
    {
        public Form2 form;
        public Form3 form3;

        public Form1()
        {
            InitializeComponent();
            WindowState = FormWindowState.Maximized;
            tabControl1.Dock = DockStyle.Fill;
            this.dataGridView3.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
            this.dataGridView3.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.DisplayedCells;
```

```

this.dataGridView3.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.DisplayedCells;
}
private void button1_Click(object sender, EventArgs e)//Парсинг файлу
{
    try
    {
        Process PYproces = new Process();
        PYproces.StartInfo.FileName = @"Parcing_module.py";
        PYproces.Start();
        PYproces.WaitForExit();
        MessageBox.Show("Парсинг завершено! Перейдіть до редагування");
    }
    catch (Exception ex) { MessageBox.Show(ex.ToString()); }
}

private void button2_Click(object sender, EventArgs e)//Формування таблиці для редагування
{
    try
    {
        int i = 0;
        for (int j = 0; j < 50; j++)
        {
            string fileName = String.Format(@"ITP{0}.xlsx", j);
            if (File.Exists(fileName) == true) { i += 1; }
        }
        Spire.Xls.Workbook MerBook = new Spire.Xls.Workbook();
        MerBook.LoadFromFile(@"ITP0.xlsx");
        Spire.Xls.Worksheet MerSheet = MerBook.Worksheets[0];
        int index = 1;
        MessageBox.Show("Початок формування таблиці Excel!");
        do
        {
            Spire.Xls.Workbook SouBook1 = new Spire.Xls.Workbook();
            string fileName = String.Format(@"ITP{0}.xlsx", index);
            SouBook1.LoadFromFile(fileName);

            int a = SouBook1.Worksheets[0].LastRow;

```

```

int b = SouBook1.Worksheets[0].LastColumn;

SouBook1.Worksheets[0].Range[1, 1, a, b].Copy(MerSheet.Range[MerSheet.LastRow + 1, 1, a +
MerSheet.LastRow, b - 1]);

MerBook.SaveToFile(@"result_1.xlsx", ExcelVersion.Version2010);

index++;
} while (index < i);
MessageBox.Show("Таблиця Excel з навантаженням сформована!\nПерейдіть до редагування");

//Add data grid view
//Create a new workbook
Spire.Xls.Workbook workbook = new Spire.Xls.Workbook();

//Load a file and imports its data
workbook.LoadFromFile(@"result_1.xlsx");

//Initialize worksheet
Spire.Xls.Worksheet sheet = workbook.Worksheets[0];
int i1 = workbook.Worksheets[0].LastRow;
int i2 = workbook.Worksheets[0].LastColumn - 1;
// get the data source that the grid is displaying data for
this.dataGridView1.DataSource = sheet.ExportDataTable(sheet.Range[1, 1, i1, i2], false);
}
catch (Exception ex) { MessageBox.Show(ex.ToString()); }
}

private void button3_Click(object sender, EventArgs e)//Додавання даних у БД
{
string conn_s;

using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" +
"Server.ini"))
{
conn_s = sr.ReadToEnd();
}

try
{

```

```

MySQLConnection conn = new MySQLConnection(conn_s);
conn.Open();

```

```

string query = "";
string semester = "Осінній семестр";
int rows = 0;
do
{
    if (dataGridView1.Rows[rows].Cells[0].Value.ToString() == "Осінній семестр")
    {
        semester = "Осінній семестр";

        rows++;
    }
    else if (dataGridView1.Rows[rows].Cells[0].Value.ToString() == "Весняний семестр")
    {
        semester = "Весняний семестр";

        rows++;
    }
    else
    {
        string zeroCell = dataGridView1.Rows[rows].Cells[0].Value.ToString(); //№ з./п.
        string firstCell = dataGridView1.Rows[rows].Cells[1].Value.ToString(); //Шифр циклу
        int res;
        bool isInt = Int32.TryParse(zeroCell, out res);
        bool isDiscipline = firstCell == "ПП" && firstCell == "ПН" &&
            firstCell == "ГС" && firstCell == "ПА" &&
            firstCell == "П";

        if ((isInt == true) || (isDiscipline == true))
        {
            string year = dataGridView1.Rows[rows].Cells[3].Value.ToString();
            string groups = dataGridView1.Rows[rows].Cells[4].Value.ToString();
            string[] delimiterChars = { ", " };
            string[] group = groups.Split(delimiterChars, System.StringSplitOptions.RemoveEmptyEntries);

            //Цикл додавання груп студентів

```

```

foreach (var g in group)
{
    string st = g.Replace("\n", "");
    query = "SELECT group_name FROM `group`";
    MySqlCommand command = new MySqlCommand(query, conn);
    MySqlDataReader reader = command.ExecuteReader();
    int check = 0;
    while (reader.Read())
    {
        if (reader[0].ToString() == st) { check = 1; }
    }
    reader.Close();

    if (check == 0)
    {
        query = String.Format("INSERT INTO `group`(group_name, year) VALUES ({0}\", {1})", st,
year);

        command = new MySqlCommand(query, conn);
        command.ExecuteNonQuery();
    }
}

//Додаємо дисципліну
string discipline_name = dataGridView1.Rows[rows].Cells[2].Value.ToString();

string groups1 = dataGridView1.Rows[rows].Cells[4].Value.ToString();
string[] delimiterChars1 = { ", " };
string[] group1 = groups1.Split(delimiterChars1, System.StringSplitOptions.RemoveEmptyEntries);

query = "SELECT group_id, group_name FROM `group`";
MySqlCommand comm = new MySqlCommand(query, conn);
string group_id_list = "";
MySqlDataReader read = comm.ExecuteReader();

while (read.Read())
{
    foreach (var g in group1)

```

```

    {
        string st = g.Replace("\n", "");
        if (read[1].ToString() == st)
        {
            group_id_list = group_id_list + String.Format("{0}", read[0].ToString());
            //MessageBox.Show(read[0].ToString() + " " + read[1].ToString());
        }
    }

}

read.Close();

//Оставить сумму или сделать проверку?
string control_d = dataGridView1.Rows[rows].Cells[16].Value.ToString() +
dataGridView1.Rows[rows].Cells[17].Value.ToString();
string ind_d = dataGridView1.Rows[rows].Cells[18].Value.ToString();

query = String.Format("SELECT discipline_name FROM discipline WHERE semester = \"{0}\"",
semester);

MySQLCommand cmd = new MySQLCommand(query, conn);
MySQLDataReader reader1 = cmd.ExecuteReader();
int check1 = 0;
while (reader1.Read())
{
    if (reader1[0].ToString() == discipline_name) { check1 = 1; }
}
reader1.Close(); // закрываем reader
int discipline_id = 0;
if (check1 == 0)
{
    query = String.Format("INSERT INTO `discipline` (discipline_name, code, semester, check_type,
individual, id_group_list) +
    \"VALUES ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}')\", discipline_name, firstCell, semester,
control_d, ind_d, group_id_list);
    cmd = new MySQLCommand(query, conn);
    cmd.ExecuteNonQuery();

    query = String.Format("SELECT discipline_id FROM `discipline` WHERE
discipline_name='{0}'\"", discipline_name);

```

```

cmd = new MySqlCommand(query, conn);

string discipline_id_st = cmd.ExecuteScalar().ToString();

discipline_id = Int32.Parse(discipline_id_st); //Запам'ятовуємо id доданої дисципліни

//Додаємо навантаження

string[] work_type = { "Лекція", "Практичне, семінарське заняття", "Лабораторне заняття",
"Інд. зан. і конс. роб. за розкладом", "Атестаційні заходи", "Проведення тестування", "Перевірка
індивідуальних завдань", "Керівництво і приймання інд. завд.", "Консул. з навч. дисц. протягом. сем.",
"Консультації передекзаменаційні", "Заліки, підсумковий сем. контроль", "Екзамени, додатковий сем.
контроль", "Керів., конс. та реценз. кваліф. проектів", "Державна атестація", "Керів. аспір., докт., здоб.,
стажування викл.", "Керівництво практикою", "Інше", "Погодинна оплата" };

//Лекція,'Практичне, семінарське заняття','Лабораторне заняття','Інд. зан. і конс. роб. за
розкладом','Атестаційні заходи','Проведення тестування','Перевірка індивідуальних завдань','Керівництво і
приймання інд. завд.','Консул. з навч. дисц. протягом. сем.','Консультації передекзаменаційні','Заліки,
підсумковий сем. контроль','Екзамени, додатковий сем. контроль','Керів., конс. та реценз. кваліф.
проектів','Державна атестація','Керів. аспір., докт., здоб., стажування викл.','Керівництво
практикою','Інше','Погодинна оплата'

string hours = "";

for (int i = 20; i < 37; i++)
{
    var value = dataGridView1.Rows[rows].Cells[i].Value;

    hours = (value != null || value!="" ? value.ToString() : "0");

    if (hours == "") { hours = "0"; }

    hours = hours.Replace(",", ".");

    query = String.Format("INSERT INTO computation(discipline_id, work_type, hours) VALUES
({0}, \"{1}\", {2})", discipline_id, work_type[i - 20], hours);

    cmd = new MySqlCommand(query, conn);

    cmd.ExecuteNonQuery();

}

hours = dataGridView1.Rows[rows].Cells[38].Value.ToString();

if (hours == "") { hours = "0"; }

hours = hours.Replace(",", ".");

query = String.Format("INSERT INTO computation(discipline_id, work_type, hours) VALUES
({0}, \"Погодинна оплата\", {1})", discipline_id, hours);

cmd = new MySqlCommand(query, conn);

cmd.ExecuteNonQuery();

}

}

rows++;

}

} while (rows < dataGridView1.Rows.Count - 1);

```



```

MessageBox.Show("Дані завантажені у БД!");
conn.Close();
}
catch (Exception ex){ MessageBox.Show(ex.ToString()); }
}
private void button4_Click(object sender, EventArgs e)//Додавання викладача
{
    string conn_s;
    using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() + "\\ +
"Server.ini"))
    {
        conn_s = sr.ReadToEnd();
    }
    try
    {
        MySqlConnection conn = new MySqlConnection(conn_s);
        conn.Open();
        string surname = textBox1.Text;
        string name = textBox2.Text;
        string patr = textBox3.Text;

        string query = "SELECT name, surname, patronymic FROM professor";
        MySqlCommand command = new MySqlCommand(query, conn);
        MySqlDataReader reader = command.ExecuteReader();
        int check = 0;
        while (reader.Read())
        {
            if (reader[0].ToString() == name ||
                reader[1].ToString() == surname ||
                reader[2].ToString() == patr) { check = 1; }
        }
        reader.Close();
        if (check == 0)
        {
            query = String.Format("INSERT INTO professor(name, surname, patronymic) VALUES ({0}\", \"{1}\",
\"{2}\")", name, surname, patr);
            command = new MySqlCommand(query, conn);
            command.ExecuteNonQuery();
        }
    }
}

```

```

    MessageBox.Show("Викладач: " + surname + " " + name + " " + patr + " доданий до БД!");
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
}
else { MessageBox.Show("Виклач " + surname + " " + name + " " + patr + " вже доданий до БД!"); }
conn.Close();
}
catch (Exception ex){ MessageBox.Show(ex.ToString()); }
}
private void button5_Click(object sender, EventArgs e)//Завантаження груп для редагування
{
    string conn_s;
    using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" +
"Server.ini"))
    {
        conn_s = sr.ReadToEnd();
    }
    try
    {
        MySqlConnection conn = new MySqlConnection(conn_s);
        conn.Open();
        string query = "SELECT count(group_name) FROM `group`";
        MySqlCommand com = new MySqlCommand(query, conn);
        int groups_count;
        bool check = Int32.TryParse(com.ExecuteScalar().ToString(), out groups_count);
        if (groups_count != 0)
        {
            string[,] groups = new string[2, groups_count];

            query = "SELECT group_id, group_name FROM `group`";
            com = new MySqlCommand(query, conn);
            MySqlDataReader read = com.ExecuteReader();
            int k = 0;
            while (read.Read())
            {
                groups[0, k] = read[0].ToString();
            }
        }
    }
}

```

```

        groups[1, k] = read[1].ToString();
        k++;
    }
    read.Close();

    for (int i = 0; i < k; i++)
    {
        comboBox1.Items.Add(groups[1, i]);
    }
    MessageBox.Show("Групи додані у випадючий список");
}
else { MessageBox.Show("У БД немає груп!"); }
conn.Close();
}
catch (Exception ex){ MessageBox.Show(ex.ToString()); }
}
private void button6_Click(object sender, EventArgs e)//Додавання групи до БД
{
    string group = comboBox1.Text;
    if (group != "")
    {
        if (textBox4.Text != "")
        {
            string conn_s;
            using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() +
"\\\" + "Server.ini"))
            {
                conn_s = sr.ReadToEnd();
            }
            try
            {
                MySqlConnection conn = new MySqlConnection(conn_s);
                conn.Open();
                string query = String.Format("SELECT group_id FROM `group` WHERE group_name=\"{0}\"",
group);
                MySqlCommand com = new MySqlCommand(query, conn);
                string group_id = com.ExecuteScalar().ToString();

```

```

        query = String.Format("UPDATE `group` SET quantity=\"{0}\" WHERE group_id={1}",
textBox4.Text, group_id);

        com = new MySqlCommand(query, conn);
        com.ExecuteNonQuery();
        conn.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.ToString()); }
    MessageBox.Show("Дані додані до БД!");
    textBox4.Clear();
}
else { MessageBox.Show("Введіть кількість студентів"); }
}
else { MessageBox.Show("Оберіть групу у випадяючому списку"); }
}
private void button7_Click(object sender, EventArgs e)//Додавання даних у таблицю для розподілу
{
    dataGridView2.Rows.Clear();
    string conn_s;
    using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() + "\\ +
"Server.ini"))
    {
        conn_s = sr.ReadToEnd();
    }
    try
    { MySqlConnection conn = new MySqlConnection(conn_s);
        conn.Open();
        string query = "SELECT COUNT( discipline_name ) FROM discipline";
        MySqlCommand com = new MySqlCommand(query, conn);
        int count;
        bool n = Int32.TryParse(com.ExecuteScalar().ToString(), out count);
        if (n == true)
        { string[,] discipline_names = new string[count, 3];

            //query = "SELECT d.semester, d.discipline_name, d.id_group_list, c.work_type, c.hours FROM
discipline d, computation c WHERE c.discipline_id = d.discipline_id AND(c.work_type = \"Лекція\" OR
c.work_type = \"Консультації передекзаменаційні\" OR c.work_type = \"Заліки, підсумковий сем. контроль\"
OR c.work_type = \"Практичне, семінарське заняття\" OR c.work_type = \"Лабораторне заняття\" OR
c.work_type = \"Інд. зан. і конс. роб. за розкладом\" OR c.work_type = \"Атестаційні заходи\" OR c.work_type
= \"Перевірка індивідуальних завдань\" OR c.work_type = \"Керівництво і приймання інд. завд.\");

            query = "SELECT semester, discipline_name, id_group_list FROM discipline";
            com = new MySqlCommand(query, conn);

```

```

MySQLDataReader read = com.ExecuteReader();
int i = 0;
while (read.Read())
{
    discipline_names[i, 0] = read[0].ToString();
    discipline_names[i, 1] = read[1].ToString();
    discipline_names[i, 2] = read[2].ToString();
    i++;
}
read.Close();
for (int j = 0; j < count; j++) {
    query = String.Format("SELECT c.hours FROM discipline d, computation c " +
        "WHERE c.discipline_id = d.discipline_id " +
        "AND c.work_type = \"Лекція\" " +
        "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
    com = new MySqlCommand(query, conn);
    var value = com.ExecuteScalar().ToString();
    string c21 = ((value != null || value != "") ? value.ToString() : "");
    query = String.Format("SELECT c.hours FROM discipline d, computation c " +
        "WHERE c.discipline_id = d.discipline_id " +
        "AND c.work_type = \"Консультації передекзаменаційні\" " +
        "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
    com = new MySqlCommand(query, conn);
    value = com.ExecuteScalar().ToString();
    string c30 = ((value != null || value != "") ? value.ToString() : "");
    query = String.Format("SELECT c.hours FROM discipline d, computation c " +
        "WHERE c.discipline_id = d.discipline_id " +
        "AND c.work_type = \"Заліки, підсумковий сем. контроль\" " +
        "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
    com = new MySqlCommand(query, conn);
    value = com.ExecuteScalar().ToString();
    string c32 = ((value != null || value != "") ? value.ToString() : "");
    string[] group = discipline_names[j, 2].Split(new char[] {','});
    string groups="";
    int groups_count=0;
    foreach (string s in group)
    {
        if (s != "")
        {
            query = String.Format("SELECT group_name FROM `group` WHERE group_id={0}", s);
            com = new MySqlCommand(query, conn);

```

```

value = com.ExecuteScalar().ToString();
groups += ((value != null || value != "") ? value.ToString() : "") + ",";
groups_count++;} }
if (discipline_names[j, 1].Contains("- KP") == false)
    dataGridView2.Rows.Add(discipline_names[j, 0], discipline_names[j, 1], groups, c21, c30, c32);
foreach (string s in group)
{
    if (s != "") {
        query = String.Format("SELECT group_name FROM `group` WHERE group_id={0}", s);
        com = new MySqlCommand(query, conn);
        value = com.ExecuteScalar().ToString();
        groups += ((value != null || value != "") ? value.ToString() : "") + ",";
        string group1 = ((value != null || value != "") ? value.ToString() : "");
        query = String.Format("SELECT c.hours FROM discipline d, computation c " +
            "WHERE c.discipline_id = d.discipline_id " +
            "AND c.work_type = \"Практичне, семінарське заняття\" " +
            "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
        com = new MySqlCommand(query, conn);
        value = com.ExecuteScalar().ToString();
        string c22 = ((value != null || value != "") ? value.ToString() : "");
        float res = float.Parse(c22);
        if (res != 0 || groups_count != 0)
        {
            res = res / (float)groups_count;
            c22 = res.ToString("0.##");
        }
        query = String.Format("SELECT c.hours FROM discipline d, computation c " +
            "WHERE c.discipline_id = d.discipline_id " +
            "AND c.work_type = \"Лабораторне заняття\" " +
            "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
        com = new MySqlCommand(query, conn);
        value = com.ExecuteScalar().ToString();
        string c23 = ((value != null || value != "") ? value.ToString() : "");
        res = float.Parse(c23);
        if (res != 0 || groups_count != 0)
        {
            res = res / (float)groups_count;

```

```

c23 = res.ToString("0.##");
}
query = String.Format("SELECT c.hours FROM discipline d, computation c " +
    "WHERE c.discipline_id = d.discipline_id " +
    "AND c.work_type = \"Інд. зан. і конс. роб. за розкладом\" " +
    "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
com = new MySqlCommand(query, conn);
value = com.ExecuteScalar().ToString();
string c24 = ((value != null || value != "") ? value.ToString() : "");
res = float.Parse(c24);
if (res != 0 || groups_count != 0)
{
    res = res / (float)groups_count;
    c24 = res.ToString("0.##");
}
query = String.Format("SELECT c.hours FROM discipline d, computation c " +
    "WHERE c.discipline_id = d.discipline_id " +
    "AND c.work_type = \"Атестаційні заходи\" " +
    "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
com = new MySqlCommand(query, conn);
value = com.ExecuteScalar().ToString();
string c25 = ((value != null || value != "") ? value.ToString() : "");
res = float.Parse(c25);
if (res != 0 || groups_count != 0)
{
    res = res / (float)groups_count;
    c25 = res.ToString("0.##");
}
query = String.Format("SELECT c.hours FROM discipline d, computation c " +
    "WHERE c.discipline_id = d.discipline_id " +
    "AND c.work_type = \"Перевірка індивідуальних завдань\" " +
    "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
com = new MySqlCommand(query, conn);
value = com.ExecuteScalar().ToString();
string c27 = ((value != null || value != "") ? value.ToString() : "");
res = float.Parse(c27);
if (res != 0 || groups_count != 0)

```

```

    { res = res / (float)groups_count;
      c27 = res.ToString("0.##");}
    query = String.Format("SELECT c.hours FROM discipline d, computation c " +
      "WHERE c.discipline_id = d.discipline_id " +
      "AND c.work_type = \"Керівництво і приймання інд. завд.\" " +
      "AND d.discipline_name = \"{0}\"", discipline_names[j, 1]);
    com = new MySqlCommand(query, conn);
    value = com.ExecuteScalar().ToString();
    string c28 = ((value != null || value != "") ? value.ToString() : "");
    res = float.Parse(c28);
    if (res != 0 || groups_count != 0)
    { res = res / (float)groups_count;
      c28 = res.ToString("0.##");}
    dataGridView2.Rows.Add(discipline_names[j, 0], discipline_names[j, 1], group1,null,null,null,
c22, c23, c24, c25, c27, c28);}}}
    conn.Close(); }catch (Exception ex){ MessageBox.Show(ex.ToString()); } }
private void dataGridView2_CellClick(object sender, DataGridViewCellEventArgs e)//Считывание клика на
колонку с преподами
{if (dataGridView2.CurrentCell.ColumnIndex.Equals(12) && e.RowIndex != -1)
  {if (dataGridView2.CurrentCell != null && dataGridView2.CurrentCell.Value == null)
    { //Инициализируем DataTransfer.data
      DataTransfer.data = new object[] { "" };
      int CursorX = Cursor.Position.X;
      int CursorY = Cursor.Position.Y;
      Form2 form2 = new Form2(CursorX - 277, CursorY - 175);
      form2.ShowDialog();
      form2.Location = new System.Drawing.Point(CursorX, CursorY);
      dataGridView2.CurrentCell.Value = DataTransfer.data[0].ToString();
      form2.Close();}}}
private void button9_Click(object sender, EventArgs e)//Зберегти тимчасову таблицю у файл
{ Microsoft.Office.Interop.Excel.Application ExcelApp = new Microsoft.Office.Interop.Excel.Application();
  Microsoft.Office.Interop.Excel.Workbook ExcelWorkBook;
  Microsoft.Office.Interop.Excel.Worksheet ExcelWorkSheet;
  ExcelWorkBook = ExcelApp.Workbooks.Add(System.Reflection.Missing.Value);
  ExcelWorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);
  for (int i = 1; i < dataGridView2.Columns.Count + 1; i++)
  { ExcelWorkSheet.Cells[1, i] = dataGridView2.Columns[i - 1].HeaderText;}
  for (int i = 2; i < dataGridView2.Rows.Count+1; i++)

```



```

    { for (int j = 0; j < dataGridView2.Columns.Count; j++)
      { var value = dataGridView2.Rows[i - 2].Cells[j].Value;
        string s = (value == null ? "" : value.ToString());
        if (s != "")
          { ExcelWorkSheet.Cells[i, j+1] = s; } } }
ExcelWorkBook.SaveAs(System.IO.Directory.GetCurrentDirectory().ToString()+"\\temp_table.xls");
ExcelApp.AlertBeforeOverwriting = false;
ExcelApp.Visible = false;
ExcelApp.Quit();
System.Runtime.InteropServices.Marshal.FinalReleaseComObject(ExcelApp);
GC.Collect();
MessageBox.Show("Таблиця успішно збережена!", "Збереження таблиці", MessageBoxButtons.OK,
MessageBoxIcon.Information);}

private void button10_Click(object sender, EventArgs e)//Вивід тимчасової таблиці
{ Microsoft.Office.Interop.Excel.Application ObjExcel = new Microsoft.Office.Interop.Excel.Application();
  Microsoft.Office.Interop.Excel.Workbook ObjWorkBook =
ObjExcel.Workbooks.Open(System.IO.Directory.GetCurrentDirectory().ToString() + "\\temp_table.xls");
  Microsoft.Office.Interop.Excel.Worksheet ObjWorkSheet = ObjExcel.ActiveSheet as
Microsoft.Office.Interop.Excel.Worksheet;
  Microsoft.Office.Interop.Excel.Range rg = null;
  Int32 row = 2;
  dataGridView2.Rows.Clear();
  List<String> arr = new List<string>();
  while (ObjWorkSheet.get_Range("a" + row, "a" + row).Value != null)
  { rg = ObjWorkSheet.get_Range("a" + row, "u" + row);
    foreach (Microsoft.Office.Interop.Excel.Range item in rg)
    { try
      { arr.Add(item.Value.ToString().Trim()); }
      catch { arr.Add(""); } }
    if (arr[12] == "") { arr[12] = null; }
    dataGridView2.Rows.Add(arr[0], arr[1], arr[2], arr[3], arr[4], arr[5], arr[6], arr[7], arr[8], arr[9], arr[10],
arr[11], arr[12]);
    arr.Clear();
    row++; }
  MessageBox.Show("Таблиця успішно додана!", "Відкриття таблиці", MessageBoxButtons.OK,
MessageBoxIcon.Information);
  ObjWorkBook.Close(false, "", null);
  ObjExcel.Quit();
  System.Runtime.InteropServices.Marshal.FinalReleaseComObject(ObjExcel);

```

```

GC.Collect();}

private void завантажитиToolStripMenuItem_Click(object sender, EventArgs e)//Додавання файлу
навантаження
{ OpenFileDialog openFileDialog1 = new OpenFileDialog();
  openFileDialog1.Filter = "Pdf Files|*.pdf";
  if (openFileDialog1.ShowDialog() == DialogResult.OK)
  { if (File.Exists(System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" + "ITP.pdf"))
      File.Delete(System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" + "ITP.pdf");
    var filePath = openFileDialog1.FileName;
    File.Copy(filePath, System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" + "ITP.pdf");
    MessageBox.Show("Файл збережено!");}}

private void коефСтавкиToolStripMenuItem_Click(object sender, EventArgs e)//Форма коефіцієнта ставки
{ form3 = new Form3();
  form3.ShowDialog();
  form3.Close(); }

private void button8_Click(object sender, EventArgs e)//Збереження розподіленого навантаження у БД
{ string conn_s;
  using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" +
"Server.ini"))
  { conn_s = sr.ReadToEnd(); }
  try
  { MySqlConnection conn = new MySqlConnection(conn_s);
    conn.Open();
    MySqlCommand comm;
    string query;
    if (dataGridView2.RowCount > 1)
    { int RowsCount = dataGridView2.RowCount;
      for (int i = 0; i < RowsCount - 1; i++)
      { if (dataGridView2.Rows[i].Cells[12].Value != null)
          { string groups_id = "";
            string[] groups = dataGridView2.Rows[i].Cells[2].Value.ToString().Split(',');
            foreach (var elem in groups)
            { if (elem != "")
                { query = String.Format("SELECT group_id FROM `group` WHERE group_name=\"{0}\"",
elem);
                  comm = new MySqlCommand(query, conn);
                  var v = comm.ExecuteScalar();
                  groups_id += ((v != null || v.ToString() != "") ? v.ToString() + "," : "");}}

```

```

string[] professor = dataGridView2.Rows[i].Cells[12].Value.ToString().Split(' ');
string surname = professor[0];
string name = professor[1];
string patr = professor[2];

query = String.Format("SELECT professor_id FROM `professor` WHERE name=\"{0}\" AND
surname=\"{1}\" AND patronymic=\"{2}\"", name, surname, patr);

comm = new MySqlCommand(query, conn);
var value = comm.ExecuteScalar();
string professor_id = ((value != null) ? value.ToString() : "");

query = String.Format("SELECT discipline_id FROM `discipline` WHERE discipline_name=\"{0}\"
AND semester=\"{1}\"", dataGridView2.Rows[i].Cells[1].Value.ToString(),
dataGridView2.Rows[i].Cells[0].Value.ToString());

comm = new MySqlCommand(query, conn);
value = comm.ExecuteScalar();
string discipline_id = ((value != null) ? value.ToString() : "");
string[] comp = new string[9];
float sum_hours = 0;
for (int j = 3; j < 12; j++)
{
    value = dataGridView2.Rows[i].Cells[j].Value;
    comp[j - 3] = (value != null ? value.ToString() : "0");
    if (comp[j - 3] == "") { comp[j - 3] = "0"; }
    sum_hours += float.Parse(comp[j - 3]);
    comp[j - 3] = comp[j - 3].Replace(',', '.');
}

query = String.Format("SELECT comp_prof_id FROM comp_prof WHERE discipline_id={0} AND
professor_id={1} AND groups=\"{2}\"", discipline_id, professor_id, groups_id);

comm = new MySqlCommand(query, conn);
var check = comm.ExecuteScalar();
bool chek = (check != null ? chek = true : chek = false);
if (chek == false)
{
    if (discipline_id != "" || professor_id != "")
    {
        query = String.Format("INSERT INTO comp_prof(discipline_id, professor_id, " +
            "groups, lectures21, ConsultBeforeExam30, FinalSemControl31, " +
            "Practice22, Laboratory23, Indiv24, Attestation25, CheckInd27, " +
            "LeaderInd28) VALUES ({0},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10},{11})",
            discipline_id, professor_id, groups_id,
            comp[0], comp[1], comp[2], comp[3], comp[4], comp[5], comp[6],
            comp[7], comp[8]);

        comm = new MySqlCommand(query, conn);
        comm.ExecuteNonQuery();
    }
}

```

```

professor_id);
        query = String.Format("SELECT hours FROM professor WHERE professor_id={0}",
                                professor_id);

        comm = new MySqlCommand(query, conn);
        var a = comm.ExecuteScalar();
        float hours = (a != null ? float.Parse(a.ToString()) : 0);
        hours = hours + sum_hours;

        float r;
        string path = System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" + "ITP.txt";
        using (StreamReader sr = new StreamReader(path))
        { r = float.Parse(sr.ReadToEnd()); }

        float rate = hours / r;
        string hours_s = hours.ToString();
        hours_s = hours_s.Replace(',', '.');
        string rate_s = rate.ToString();
        rate_s = rate_s.Replace(',', '.');

        query = String.Format("UPDATE professor SET hours={0}, rate={1} WHERE
professor_id={2}",
                                hours_s, rate_s, professor_id);
        comm = new MySqlCommand(query, conn);
        comm.ExecuteNonQuery();
        MessageBox.Show("Дані додані до БД!"); }
        else { MessageBox.Show("Ви не вказали дисципліну або викладача у " + i + " рядку"); }}
        else { MessageBox.Show("Навантаження у рядку " + (i + 1) + " уже є в БД!"); }}
        else { MessageBox.Show("У рядку " + (i+1) + " не вказаний викладач"); }}
    else { MessageBox.Show("Додайте дані в таблицю"); }
    conn.Close(); }
    catch (Exception ex){ MessageBox.Show(ex.ToString()); } }

private void button12_Click(object sender, EventArgs e)//Перегляд викладачів з навантаженням
{ button11.Enabled = true;
    string conn_s;
    using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" +
"Server.ini"))
    { conn_s = sr.ReadToEnd(); }
    try
    { MySqlConnection conn = new MySqlConnection(conn_s);
        conn.Open();

        string query = "SELECT name, surname, patronymic, hours, rate FROM professor WHERE hours IS
NOT NULL";
        MySqlCommand comm = new MySqlCommand(query, conn);

```

```

MySQLDataReader read = comm.ExecuteReader();
while (read.Read()) {
    string professor = "", hours = "", rate = "";
    if (read[1].ToString() != null) {
        professor += read[1].ToString() + " ";
    }
    if (read[0].ToString() != null)
    { professor += read[0].ToString() + " "; }
    if (read[2].ToString() != null)
    { professor += read[2].ToString() + " "; }
    if (read[3].ToString() != null)
    { hours = read[3].ToString(); }
    if (read[4].ToString() != null)
    { rate = read[4].ToString(); }
    comboBox2.Items.Add(professor);
    dataGridView3.Rows.Add(professor, hours, rate);
} conn.Close();
} catch (Exception ex){ MessageBox.Show(ex.ToString()); } }
private void button11_Click(object sender, EventArgs e)//Вывод у Word
{ Microsoft.Office.Interop.Word.Application wordApp;
  Microsoft.Office.Interop.Word.Document wordDoc;
  wordApp = new Microsoft.Office.Interop.Word.Application();
  wordDoc = wordApp.Documents.Add(System.IO.Directory.GetCurrentDirectory().ToString() +
  "\\Res.docx");
  Object missing = Type.Missing;
  Microsoft.Office.Interop.Word.Range selText;
  selText = wordDoc.Range(wordDoc.Content.Start, wordDoc.Content.End);
  Microsoft.Office.Interop.Word.Find find = wordApp.Selection.Find;
  string conn_s;
  using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() + "\\ +
  "Server.ini"))
  { conn_s = sr.ReadToEnd(); }
  try
  { MySqlConnection conn = new MySqlConnection(conn_s);
    conn.Open();
    string[] s = comboBox2.Text.Split(' ');
    string name=s[1];
    string surname=s[0];
    string patronymic=s[2];

```

```

string hours = "";
string rate = "";
string professor_id = "";
string query = "SELECT professor_id, name, surname, patronymic, hours, rate FROM professor";
MySqlCommand comm = new MySqlCommand(query, conn);
MySqlDataReader read = comm.ExecuteReader();
while (read.Read()) {
    if (read[0] != null || read[1] != null || read[2] != null || read[3] != null || read[4] != null || read[5] != null) {
        if (read[1].ToString() == name || read[2].ToString() == surname || read[3].ToString() == patronymic)
            { professor_id = read[0].ToString();
              hours = read[4].ToString();
              rate = read[5].ToString();
            } } }
read.Close();
find.Text = "[professor]";
find.Replacement.Text = Convert.ToString(surname + " " + name + " " + patronymic);
Object wrap = Microsoft.Office.Interop.Word.WdFindWrap.wdFindContinue;
Object replace = Microsoft.Office.Interop.Word.WdReplace.wdReplaceAll;
find.Execute(FindText: Type.Missing,
MatchCase: false,
MatchWholeWord: false,
MatchWildcards: false,
MatchSoundsLike: missing,
MatchAllWordForms: false,
Forward: true,
Wrap: wrap,
Format: false,
ReplaceWith: missing, Replace: replace);
find.Text = "[date]";
find.Replacement.Text = Convert.ToString(DateTime.Now.ToString("dd MMMM yyyy"));
wrap = Microsoft.Office.Interop.Word.WdFindWrap.wdFindContinue;
replace = Microsoft.Office.Interop.Word.WdReplace.wdReplaceAll;
find.Execute(FindText: Type.Missing,
MatchCase: false,
MatchWholeWord: false,
MatchWildcards: false,
MatchSoundsLike: missing,

```

```
MatchAllWordForms: false,
Forward: true,
Wrap: wrap,
Format: false,
ReplaceWith: missing, Replace: replace);
find.Text = "[hours]";
find.Replacement.Text = Convert.ToString(hours);
wrap = Microsoft.Office.Interop.Word.WdFindWrap.wdFindContinue;
replace = Microsoft.Office.Interop.Word.WdReplace.wdReplaceAll;
find.Execute(FindText: Type.Missing,
MatchCase: false,
MatchWholeWord: false,
MatchWildcards: false,
MatchSoundsLike: missing,
MatchAllWordForms: false,
Forward: true,
Wrap: wrap,
Format: false,
ReplaceWith: missing, Replace: replace);
find.Text = "[rate]";
find.Replacement.Text = Convert.ToString(rate);
wrap = Microsoft.Office.Interop.Word.WdFindWrap.wdFindContinue;
replace = Microsoft.Office.Interop.Word.WdReplace.wdReplaceAll;
find.Execute(FindText: Type.Missing,
MatchCase: false,
MatchWholeWord: false,
MatchWildcards: false,
MatchSoundsLike: missing,
MatchAllWordForms: false,
Forward: true,
Wrap: wrap,
Format: false,
ReplaceWith: missing, Replace: replace);
query = "SELECT count(groups) FROM comp_prof";
comm = new MySqlCommand(query, conn);
var value = comm.ExecuteScalar();
int count_groups = (value != null ? Int32.Parse(value.ToString()) : 0);
```

```

query = "SELECT groups FROM comp_prof";
comm = new MySqlCommand(query, conn);
read = comm.ExecuteReader();
string[,] groups_id = new string[count_groups, 4];
int k = 0;
while (read.Read()) {if (read[0] != null)
    { groups_id[k, 0] = read[0].ToString();
      k++;}}
read.Close();
int kol=0;
for (int i = 0; i < count_groups; i++)
{ string[] gr = groups_id[i, 0].Split(',');
  kol = 0;
  foreach (var g in gr) {
    if (g != "") {
      query = String.Format("SELECT group_name FROM `group` WHERE group_id={0}", g);
      comm = new MySqlCommand(query, conn);
      var v = comm.ExecuteScalar();
      groups_id[i, 1] += (v!=null ? v.ToString() : "")+" ";
      query = String.Format("SELECT year FROM `group` WHERE group_id={0}", g);
      comm = new MySqlCommand(query, conn);
      v = comm.ExecuteScalar();
      groups_id[i, 2] = (v != null ? v.ToString() : "");
      query = String.Format("SELECT quantity FROM `group` WHERE group_id={0}", g);
      comm = new MySqlCommand(query, conn);
      v = comm.ExecuteScalar();
      kol = kol + (v != null ? Int32.Parse(v.ToString()) : 0);
      groups_id[i, 3] = kol.ToString(); } }}

  query = String.Format("SELECT d.discipline_name, d.check_type, d.individual, cp.groups, cp.lectures21,
cp.ConsultBeforeExam30, cp.FinalSemControl31, cp.Practice22, cp.Laboratory23, cp.Indiv24, cp.Attestation25,
cp.CheckInd27, cp.LeaderInd28 "+
      "FROM discipline d, comp_prof cp "+
      "WHERE cp.discipline_id = d.discipline_id AND cp.professor_id = {0} AND d.semester =
\"Осінній семестр\"","professor_id);
  comm = new MySqlCommand(query, conn);
  read = comm.ExecuteReader();
  int row = 0;
  float[] sum_cells = new float[9];

```



```

while (read.Read()) {
    if (read[0] != null) {
        wordDoc.Tables[1].Cell(2 + row, 1).Range.Text = read[0].ToString();
        if (read[1] != null || read[2] != null)
            { wordDoc.Tables[1].Cell(2 + row, 5).Range.Text = read[1].ToString();
              wordDoc.Tables[1].Cell(2 + row, 6).Range.Text = read[2].ToString();}
        if (read[3] != null) {
            for (int i = 0; i < count_groups; i++) {
                if (read[3].ToString().Equals(groups_id[i, 0])) {
                    wordDoc.Tables[1].Cell(2 + row, 2).Range.Text = groups_id[i, 2].ToString();
                    wordDoc.Tables[1].Cell(2 + row, 3).Range.Text = groups_id[i, 1].ToString().Trim(',');
                    wordDoc.Tables[1].Cell(2 + row, 4).Range.Text = groups_id[i, 3].ToString();} }
            float sum_c = 0;
            for (int i = 4; i <= 12; i++) {if (read[i] != null) { wordDoc.Tables[1].Cell(2 + row, i + 3).Range.Text =
read[i].ToString();
                sum_c += (read[i].ToString()!=" " ? float.Parse(read[i].ToString()) : 0);
                sum_cells[i - 4] += (read[i].ToString() != " " ? float.Parse(read[i].ToString()) : 0); } }
            wordDoc.Tables[1].Cell(2 + row, 16).Range.Text = sum_c.ToString();
        wordDoc.Tables[1].Rows.Add();
        row++;}
read.Close();
float sum_sum_cells = 0;
for (int i = 0; i < 9; i++) { wordDoc.Tables[1].Cell(2 + row, i + 7).Range.Text = sum_cells[i].ToString();
    sum_sum_cells += sum_cells[i]; }
wordDoc.Tables[1].Cell(2 + row, 16).Range.Text = sum_sum_cells.ToString();

    query = String.Format("SELECT d.discipline_name, d.check_type, d.individual, cp.groups, cp.lectures21,
cp.ConsultBeforeExam30, cp.FinalSemControl31, cp.Practice22, cp.Laboratory23, cp.Indiv24, cp.Attestation25,
cp.CheckInd27, cp.LeaderInd28 " +
        "FROM discipline d, comp_prof cp " +
        "WHERE cp.discipline_id = d.discipline_id AND cp.professor_id = {0} AND d.semester =
\"Весняний семестр\"", professor_id);
    comm = new MySqlCommand(query, conn);
    read = comm.ExecuteReader();
    row = 0;
    float[] sum_cells1 = new float[9];
    while (read.Read())
        { if (read[0] != null)
            { wordDoc.Tables[2].Cell(2 + row, 1).Range.Text = read[0].ToString();
              if (read[1] != null || read[2] != null)

```

```

{ wordDoc.Tables[2].Cell(2 + row, 5).Range.Text = read[1].ToString();
  wordDoc.Tables[2].Cell(2 + row, 6).Range.Text = read[2].ToString();}
if (read[3] != null)
{for (int i = 0; i < count_groups; i++)
  {if (read[3].ToString().Equals(groups_id[i, 0]))
    { wordDoc.Tables[2].Cell(2 + row, 2).Range.Text = groups_id[i, 2].ToString();
      wordDoc.Tables[2].Cell(2 + row, 3).Range.Text = groups_id[i, 1].ToString().Trim(',');
      wordDoc.Tables[2].Cell(2 + row, 4).Range.Text = groups_id[i, 3].ToString();} } }
float sum_c = 0;
for (int i = 4; i <= 12; i++)
{if (read[i] != null)
  { wordDoc.Tables[2].Cell(2 + row, i + 3).Range.Text = read[i].ToString();
    sum_c += (read[i].ToString() != "" ? float.Parse(read[i].ToString()) : 0);
    sum_cells1[i - 4] += (read[i].ToString() != "" ? float.Parse(read[i].ToString()) : 0); } }
wordDoc.Tables[2].Cell(2 + row, 16).Range.Text = sum_c.ToString();
wordDoc.Tables[2].Rows.Add();
row++;}
read.Close();
sum_sum_cells = 0;
for (int i = 0; i < 9; i++)
{ wordDoc.Tables[2].Cell(2 + row, i + 7).Range.Text = sum_cells1[i].ToString();
  sum_sum_cells += sum_cells1[i]; }
wordDoc.Tables[2].Cell(2 + row, 16).Range.Text = sum_sum_cells.ToString();
wordApp.Visible = false;
wordDoc.Save();
wordDoc.Close();
wordApp.Quit();
System.Runtime.InteropServices.Marshal.FinalReleaseComObject(wordDoc);
GC.Collect();
conn.Close();}
catch (Exception ex){ MessageBox.Show(ex.ToString()); }}
private void cepBepToolStripMenuItem_Click(object sender, EventArgs e)
{
  Form4 form4 = new Form4();
  form4.ShowDialog(); }}

```

Form2.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.IO;
namespace Diploma_Add_Py
{
    public partial class Form2 : Form
    {
        public Form2(int x, int y)
        {
            InitializeComponent();
            this.Location = new Point(x-277, y-175);

            string professor = comboBox1.Text;

            if (professor == "")
            {
                string conn_s;
                using (StreamReader sr = new StreamReader(System.IO.Directory.GetCurrentDirectory().ToString() +
                "\\\" + "Server.ini"))
                {
                    conn_s = sr.ReadToEnd();
                }
                MySqlConnection conn = new MySqlConnection(conn_s);
                conn.Open();

                string query = "SELECT count(name) FROM professor";
                MySqlCommand com = new MySqlCommand(query, conn);

                int count;
                bool a = Int32.TryParse(com.ExecuteScalar().ToString(), out count);
                if (count != 0)
                {
                    string[] prof = new string[count];
                    query = "SELECT name, surname, patronymic FROM professor";
                    com = new MySqlCommand(query, conn);
                    MySqlDataReader read = com.ExecuteReader();
                    int i = 0;
                    while (read.Read())
                    {
                        prof[i] = read[1].ToString() + " " + read[0].ToString() + " " + read[2].ToString();
                        i++;
                    }
                    read.Close();
                    for (i = 0; i < count; ++i) {
                        comboBox1.Items.Add(prof[i]);
                    }
                }
                else { MessageBox.Show("Викладачі не додані!\nДодайте викладчів і поверніться до розподілу
                навантаження"); }
            }
        }
        private void OK_Click(object sender, EventArgs e)
    }
}

```

```

    {
        if (comboBox1.Text != "")
        {
            //Изменяем dataTransfer
            DataTransfer.data[0] = comboBox1.Text;
            Dispose();
        }
        else { MessageBox.Show("Оберіть викладача!"); }
    }
}
}

```

Form3.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
namespace Diploma_Add_Py
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
            string path = System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" + "ITP.txt";
            using (StreamReader sr = new StreamReader(path))
            {label2.Text = "Коефіцієнт: " + sr.ReadToEnd();}
            private void OK_Click(object sender, EventArgs e)
            {using (StreamWriter sw = new StreamWriter(System.IO.Directory.GetCurrentDirectory().ToString() + "\\\" +
            "ITP.txt", false, System.Text.Encoding.Default))
            {
                sw.WriteLine(textBox1.Text);
                MessageBox.Show("Коефіцієнт: " + textBox1.Text + " доданий!");
                Close();}}}}

```

Form4.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
namespace Diploma_Add_Py
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            using (StreamWriter sw = new StreamWriter(System.IO.Directory.GetCurrentDirectory().ToString() + "\\Server.ini", false, System.Text.Encoding.Default))
            {
                if (textBox1.Text != "" && textBox1.Text != null)
                {
                    if (textBox2.Text != "" && textBox2.Text != null)
                    {
                        if (textBox3.Text != "" && textBox3.Text != null)
                        {
                            string connect = String.Format("server={0};user={1};database={2};password={3}",
                                textBox1.Text, textBox2.Text,
                                textBox3.Text, textBox4.Text);
                            sw.WriteLine(connect);
                            MessageBox.Show("Інформація додана!");
                            Close();
                        }
                    }
                }
            }
            else { MessageBox.Show("Введіть назву БД!"); }
        }
    }
}

```

```
    }  
    else { MessageBox.Show("Введіть ім'я користувача!"); }  
}  
else { MessageBox.Show("Введіть назву серверу!"); } } } }
```

ДОДАТОК Е

Інструкція користувача

Інсталяція програми

Перед інсталяцією основної програми необхідно інсталювати додаткові елементи. Спочатку інсталюйте Python 3.7.3. Виконайте файл python-3.7.3.exe. При запуску файлу обов'язково оберіть Add Python 3.7 to PATH, приклад зображений на рисунку Е.1. Після дотримуйтеся вказівок інсталлятора.



Рисунок Е.1 - Інсталяція Python 3.7.3

Після завершення інсталяції Python інсталюйте GhostScript. Для цього виконайте файл gs927w64.exe. Після відкриття інсталлятора дотримуйтеся його вказівок і стандартних налаштувань.

Після закінчення встановлення GhostScrip виконайте файл Setup.exe, щоб інсталювати інформаційну систему навчального навантаження. Після відкриття інсталятора дотримуйтесь його вказівок і стандартних налаштувань. Інсталювання «Облік навчального навантаження» зображений на рисунку Е.2.

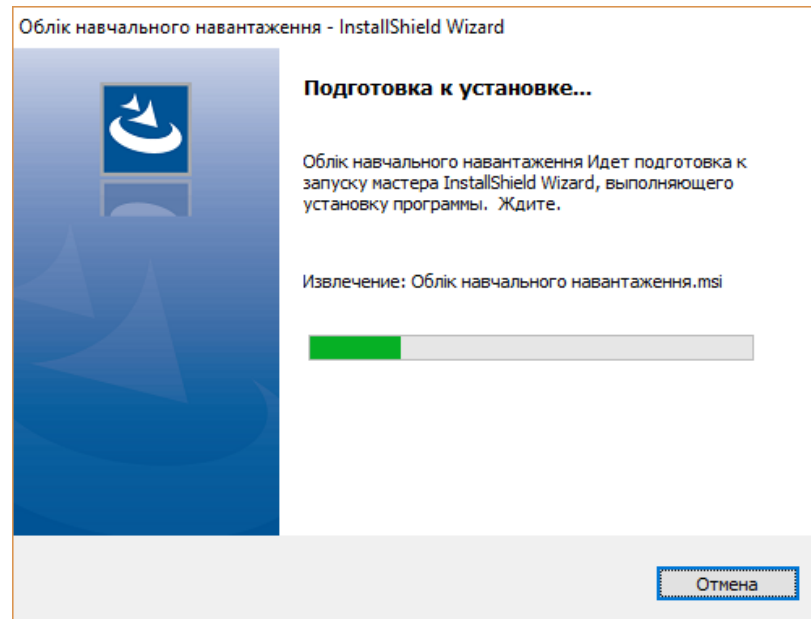


Рисунок Е.2 – Інсталювання «Облік навчального навантаження»

Після закінчення інсталяції програми знаходимо дерикторію програми, затискаючи shift натискаємо правою кнопкою миші на директорії. Обираємо «Открыть окно PowerShell здесь»(рисунок Е.3).

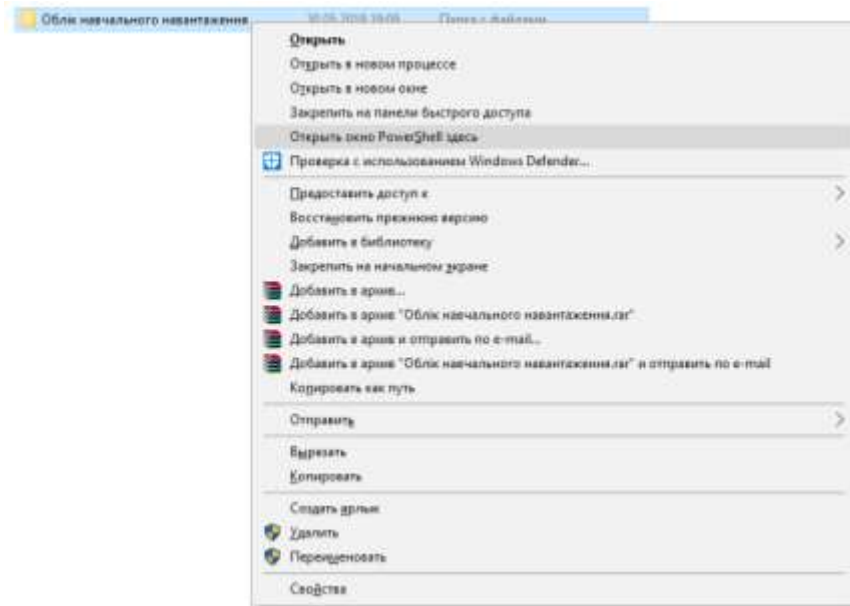


Рисунок Е.3 – Відкриття вікна PowerShell

Приклад вікна, що відкриється наведений в рисунку Е.4.

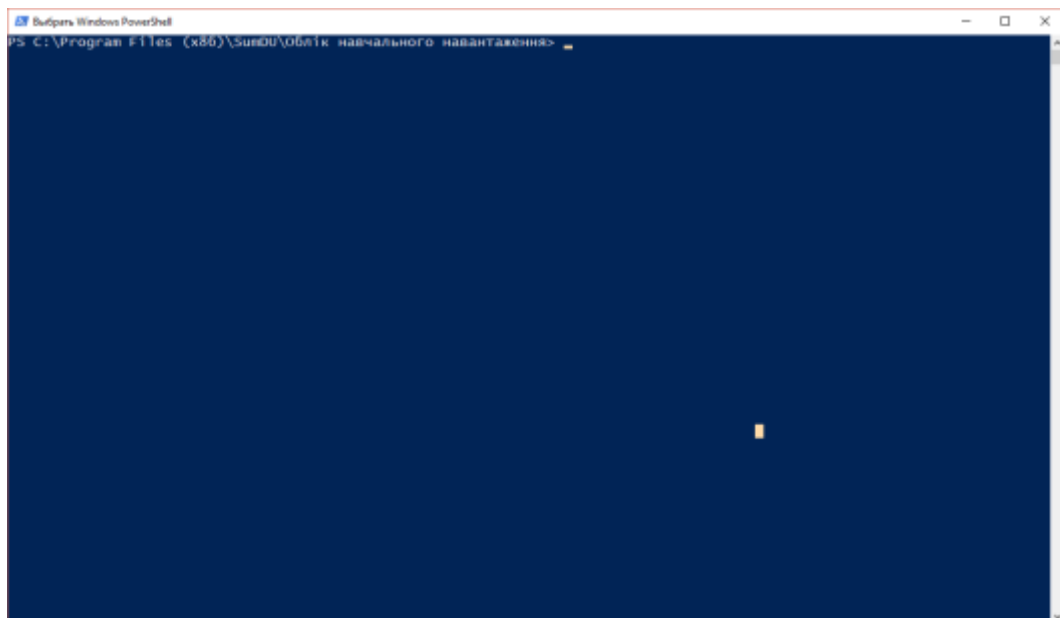
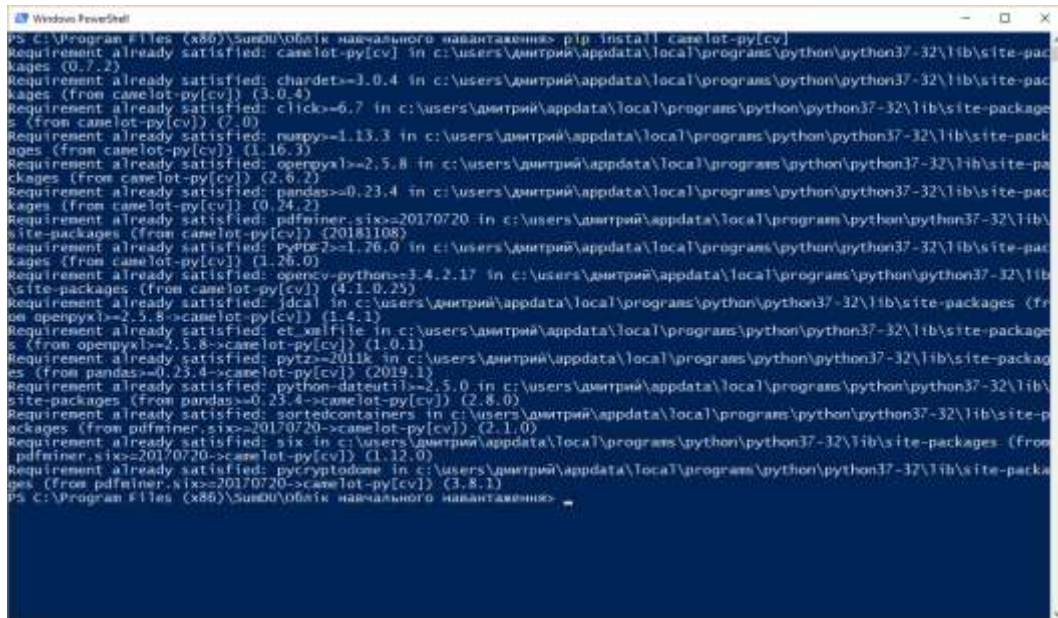


Рисунок Е.4 – Відкрите вікно PowerShell

Далі вводимо команду «`pip install camelot-py[cv]`» і натискаємо Enter (Рисунок Е.5)



```

PS C:\Program Files (x86)\SunODU\Облік навчального навантаження> pip install camelot-py[cv]
Requirement already satisfied: camelot-py[cv] in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (0.7.2)
Requirement already satisfied: chardet==3.0.4 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from camelot-py[cv]) (3.0.4)
Requirement already satisfied: click==6.7 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from camelot-py[cv]) (7.0)
Requirement already satisfied: numpy==1.13.3 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from camelot-py[cv]) (1.16.3)
Requirement already satisfied: openpyxl==2.5.8 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pa
ckages (from camelot-py[cv]) (2.6.2)
Requirement already satisfied: pandas==0.23.4 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from camelot-py[cv]) (0.24.2)
Requirement already satisfied: pdfminer-six==20170720 in c:\users\дмитрий\appdata\local\program\python\python37-32\lib\
site-packages (from camelot-py[cv]) (20181108)
Requirement already satisfied: PyPDF2==1.26.0 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from camelot-py[cv]) (1.26.0)
Requirement already satisfied: opencv-python==3.4.2.17 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib
\site-packages (from camelot-py[cv]) (4.1.0.25)
Requirement already satisfied: jdcal in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-packages (fr
om openpyxl==2.5.8->camelot-py[cv]) (1.4.1)
Requirement already satisfied: et_xmlfile in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from openpyxl==2.5.8->camelot-py[cv]) (1.0.1)
Requirement already satisfied: pytz==2018 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from pandas==0.23.4->camelot-py[cv]) (2019.1)
Requirement already satisfied: python-dateutil==2.5.0 in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\
site-packages (from pandas==0.23.4->camelot-py[cv]) (2.8.0)
Requirement already satisfied: sortedcontainers in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-p
ackages (from pdfminer-six==20170720->camelot-py[cv]) (2.1.0)
Requirement already satisfied: six in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-packages (from
pdfminer-six==20170720->camelot-py[cv]) (1.12.0)
Requirement already satisfied: pycryptodome in c:\users\дмитрий\appdata\local\programs\python\python37-32\lib\site-pac
kages (from pdfminer-six==20170720->camelot-py[cv]) (3.8.1)
PS C:\Program Files (x86)\SunODU\Облік навчального навантаження>

```

Рисунок Е.5 – Інсталювання Camelot

Інсталяція програми завершена, можете переходити до роботи.

Налаштування програми

Перед початком використання програми необхідно встановити коефіцієнт ставки, для розрахунку ставки викладачів, і дані серверу, на якому знаходиться база даних.

Для встановлення коефіцієнту ставки перейдіть у меню «Налаштування» і натисніть кнопку «Коеф. ставки». Після відкриття вікна ви можете побачити поточний коефіцієнт і ввести новий. Після введення коефіцієнту натисніть кнопку «Ок». Приклад вікна зображений на рисунку Е.6.

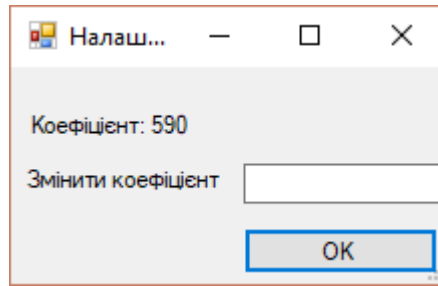


Рисунок Е.6 – Вікно налаштувань коефіцієнту ставки

Для додавання даних серверу перейдіть у меню «Налаштування» і натисніть кнопку «Сервер». Введіть назву серверу, ім'я користувача, назву бази даних, пароль і натисніть кнопку «Додати». Вікно налаштувань серверу зображено на рисунку Е.7.

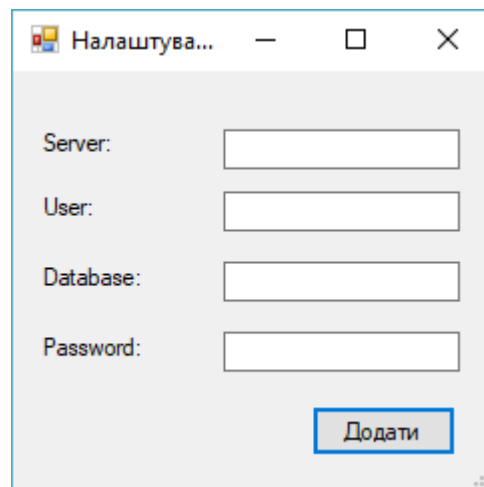


Рисунок Е.7 – Вікно налаштувань серверу

Процес роботи з програмою

Для додавання PDF-файлу навантаження після відкриття головного вікна програми перейдіть у меню файл і натисніть кнопку «Файл». Далі оберіть файл і натисніть кнопку «Открыть». Додавання файлу зображено на рисунку Е.8.

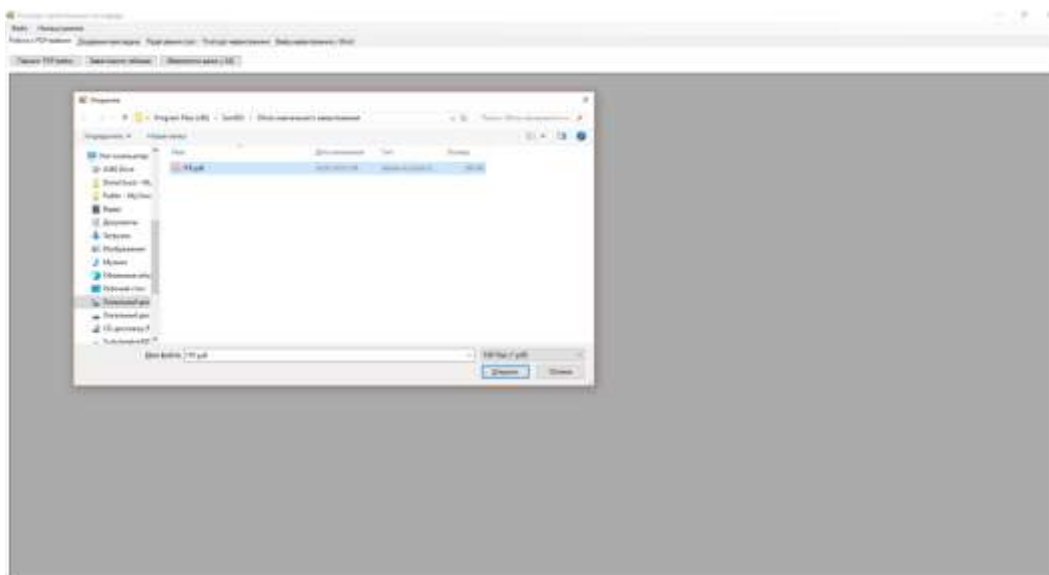


Рисунок Е.8 – Додавання PDF-файлу у програму

Після занесення файлу навантаження у програму натисніть кнопку «Парсинг PDF-файлу» для виконання обробки заданого файлу. Після натискання кнопки відкриється консольне вікно де ви побачите поточний стан парсингу, приклад зображений на рисунку Е.9.

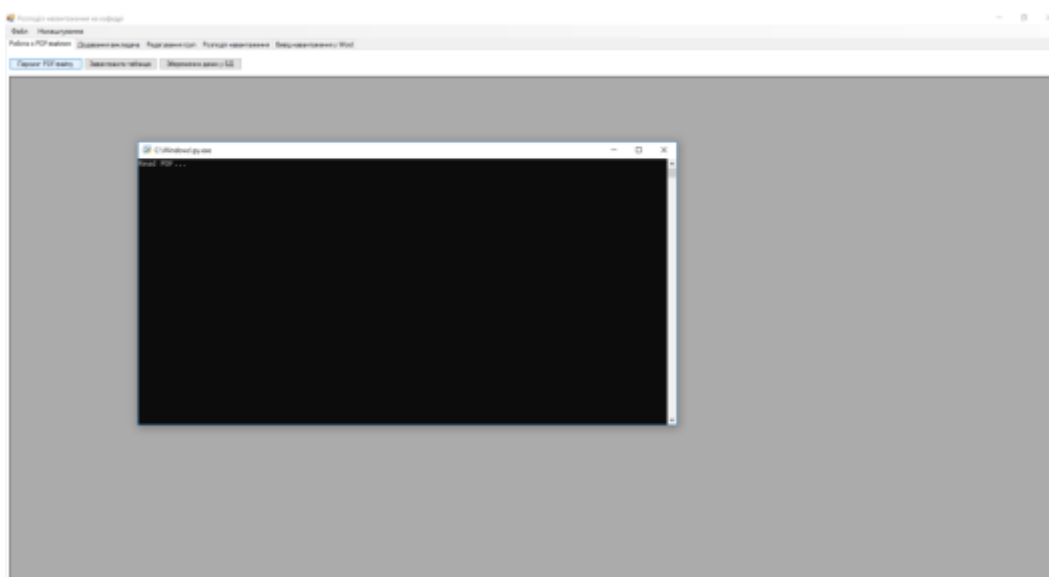


Рисунок Е.9 – Процес парсингу файлу

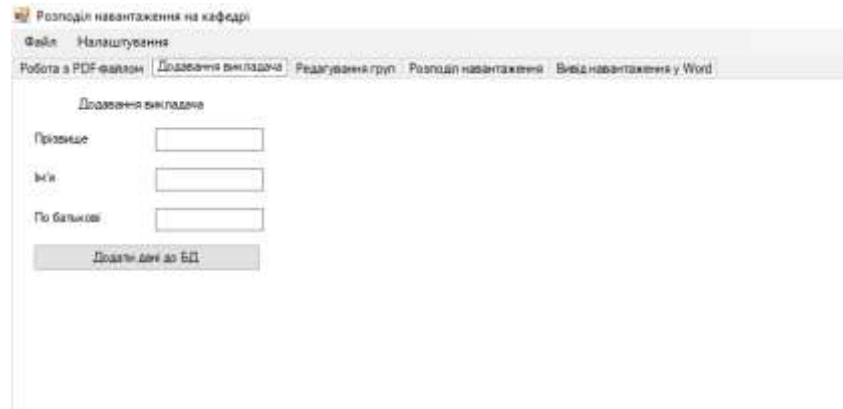


Рисунок Е.11 – Додавання викладача

У вкладці «Редагування груп» натисніть кнопку завантаження групи. Після додавання груп у випадаючий список оберіть групу у випадаючому списку і введіть кількість студентів. Натисніть кнопку «Зберегти у БД» для занесення кількості студентів у базу даних. Виконайте дані дії для кожної групи. Вкладка редагування груп зображена на рисунку Е.12.

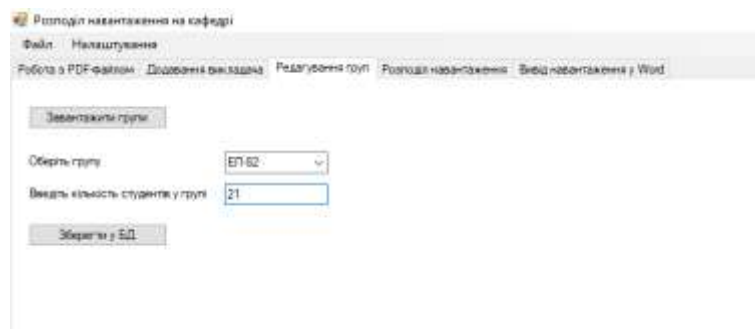


Рисунок Е.12 – Вкладка редагування груп

Перейдіть у вкладку «Розподіл навантаження» і натисніть кнопку «Додати нерозподілене навантаження». Для додавання викладача до дисципліни натисніть на пусту комірку у стовпчику «Викладач», у вікні, що відкрилося, оберіть викладача і натисніть «Додати викладача». Вкладка «Розподіл навантаження» зображена на рисунку Е.13.

Ідентифікатор	Ім'я викладача	Дата	Час	Кількість навчальних годин	Середній бал	Відсоток проходження	Підписаний викладач	Активність студентів	Активність викладача	Середній бал викладача	Середній бал студентів	Відсоток проходження викладача
001	Петренко	2023-01-15	08	3	3	100%		0	0	0	0	0
002	Петренко	2023-01-15	09	3	3	100%		0	0	0	0	0
003	Петренко	2023-01-15	10	3	3	100%		0	0	0	0	0
004	Петренко	2023-01-15	11	3	3	100%		0	0	0	0	0
005	Петренко	2023-01-15	12	3	3	100%		0	0	0	0	0
006	Петренко	2023-01-15	13	3	3	100%		0	0	0	0	0
007	Петренко	2023-01-15	14	3	3	100%		0	0	0	0	0
008	Петренко	2023-01-15	15	3	3	100%		0	0	0	0	0
009	Петренко	2023-01-15	16	3	3	100%		0	0	0	0	0
010	Петренко	2023-01-15	17	3	3	100%		0	0	0	0	0
011	Петренко	2023-01-15	18	3	3	100%		0	0	0	0	0
012	Петренко	2023-01-15	19	3	3	100%		0	0	0	0	0
013	Петренко	2023-01-15	20	3	3	100%		0	0	0	0	0
014	Петренко	2023-01-15	21	3	3	100%		0	0	0	0	0
015	Петренко	2023-01-15	22	3	3	100%		0	0	0	0	0
016	Петренко	2023-01-15	23	3	3	100%		0	0	0	0	0
017	Петренко	2023-01-15	24	3	3	100%		0	0	0	0	0
018	Петренко	2023-01-15	25	3	3	100%		0	0	0	0	0
019	Петренко	2023-01-15	26	3	3	100%		0	0	0	0	0
020	Петренко	2023-01-15	27	3	3	100%		0	0	0	0	0
021	Петренко	2023-01-15	28	3	3	100%		0	0	0	0	0
022	Петренко	2023-01-15	29	3	3	100%		0	0	0	0	0
023	Петренко	2023-01-15	30	3	3	100%		0	0	0	0	0
024	Петренко	2023-01-15	31	3	3	100%		0	0	0	0	0
025	Петренко	2023-01-15	32	3	3	100%		0	0	0	0	0
026	Петренко	2023-01-15	33	3	3	100%		0	0	0	0	0
027	Петренко	2023-01-15	34	3	3	100%		0	0	0	0	0
028	Петренко	2023-01-15	35	3	3	100%		0	0	0	0	0
029	Петренко	2023-01-15	36	3	3	100%		0	0	0	0	0
030	Петренко	2023-01-15	37	3	3	100%		0	0	0	0	0
031	Петренко	2023-01-15	38	3	3	100%		0	0	0	0	0
032	Петренко	2023-01-15	39	3	3	100%		0	0	0	0	0
033	Петренко	2023-01-15	40	3	3	100%		0	0	0	0	0
034	Петренко	2023-01-15	41	3	3	100%		0	0	0	0	0
035	Петренко	2023-01-15	42	3	3	100%		0	0	0	0	0
036	Петренко	2023-01-15	43	3	3	100%		0	0	0	0	0
037	Петренко	2023-01-15	44	3	3	100%		0	0	0	0	0
038	Петренко	2023-01-15	45	3	3	100%		0	0	0	0	0
039	Петренко	2023-01-15	46	3	3	100%		0	0	0	0	0
040	Петренко	2023-01-15	47	3	3	100%		0	0	0	0	0
041	Петренко	2023-01-15	48	3	3	100%		0	0	0	0	0
042	Петренко	2023-01-15	49	3	3	100%		0	0	0	0	0
043	Петренко	2023-01-15	50	3	3	100%		0	0	0	0	0
044	Петренко	2023-01-15	51	3	3	100%		0	0	0	0	0
045	Петренко	2023-01-15	52	3	3	100%		0	0	0	0	0
046	Петренко	2023-01-15	53	3	3	100%		0	0	0	0	0
047	Петренко	2023-01-15	54	3	3	100%		0	0	0	0	0
048	Петренко	2023-01-15	55	3	3	100%		0	0	0	0	0
049	Петренко	2023-01-15	56	3	3	100%		0	0	0	0	0
050	Петренко	2023-01-15	57	3	3	100%		0	0	0	0	0
051	Петренко	2023-01-15	58	3	3	100%		0	0	0	0	0
052	Петренко	2023-01-15	59	3	3	100%		0	0	0	0	0
053	Петренко	2023-01-15	60	3	3	100%		0	0	0	0	0
054	Петренко	2023-01-15	61	3	3	100%		0	0	0	0	0
055	Петренко	2023-01-15	62	3	3	100%		0	0	0	0	0
056	Петренко	2023-01-15	63	3	3	100%		0	0	0	0	0
057	Петренко	2023-01-15	64	3	3	100%		0	0	0	0	0
058	Петренко	2023-01-15	65	3	3	100%		0	0	0	0	0
059	Петренко	2023-01-15	66	3	3	100%		0	0	0	0	0
060	Петренко	2023-01-15	67	3	3	100%		0	0	0	0	0
061	Петренко	2023-01-15	68	3	3	100%		0	0	0	0	0
062	Петренко	2023-01-15	69	3	3	100%		0	0	0	0	0
063	Петренко	2023-01-15	70	3	3	100%		0	0	0	0	0
064	Петренко	2023-01-15	71	3	3	100%		0	0	0	0	0
065	Петренко	2023-01-15	72	3	3	100%		0	0	0	0	0
066	Петренко	2023-01-15	73	3	3	100%		0	0	0	0	0
067	Петренко	2023-01-15	74	3	3	100%		0	0	0	0	0
068	Петренко	2023-01-15	75	3	3	100%		0	0	0	0	0
069	Петренко	2023-01-15	76	3	3	100%		0	0	0	0	0
070	Петренко	2023-01-15	77	3	3	100%		0	0	0	0	0
071	Петренко	2023-01-15	78	3	3	100%		0	0	0	0	0
072	Петренко	2023-01-15	79	3	3	100%		0	0	0	0	0
073	Петренко	2023-01-15	80	3	3	100%		0	0	0	0	0
074	Петренко	2023-01-15	81	3	3	100%		0	0	0	0	0
075	Петренко	2023-01-15	82	3	3	100%		0	0	0	0	0
076	Петренко	2023-01-15	83	3	3	100%		0	0	0	0	0
077	Петренко	2023-01-15	84	3	3	100%		0	0	0	0	0
078	Петренко	2023-01-15	85	3	3	100%		0	0	0	0	0
079	Петренко	2023-01-15	86	3	3	100%		0	0	0	0	0
080	Петренко	2023-01-15	87	3	3	100%		0	0	0	0	0
081	Петренко	2023-01-15	88	3	3	100%		0	0	0	0	0
082	Петренко	2023-01-15	89	3	3	100%		0	0	0	0	0
083	Петренко	2023-01-15	90	3	3	100%		0	0	0	0	0
084	Петренко	2023-01-15	91	3	3	100%		0	0	0	0	0
085	Петренко	2023-01-15	92	3	3	100%		0	0	0	0	0
086	Петренко	2023-01-15	93	3	3	100%		0	0	0	0	0
087	Петренко	2023-01-15	94	3	3	100%		0	0	0	0	0
088	Петренко	2023-01-15	95	3	3	100%		0	0	0	0	0
089	Петренко	2023-01-15	96	3	3	100%		0	0	0	0	0
090	Петренко	2023-01-15	97	3	3	100%		0	0	0	0	0
091	Петренко	2023-01-15	98	3	3	100%		0	0	0	0	0
092	Петренко	2023-01-15	99	3	3	100%		0	0	0	0	0
093	Петренко	2023-01-15	100	3	3	100%		0	0	0	0	0

Рисунок Е.13 – Вкладка «Розподіл навантаження»

Для збереження ще не готової таблиці натисніть кнопку «Зберегти тимчасову таблицю у файл». Для завантаження збереженої таблиці натисніть кнопку «Завантажити тимчасову таблицю». Після завершення виконання розподілу навчального навантаження між викладачами натисніть кнопку «Зберегти готову таблицю у БД».

Для перегляду навантаження викладачів і виводу навантаження конкретного викладача перейдіть у вкладку «Вивід навантаження у Word». Після переходу натисніть кнопку «Вивести всіх викладачів з навантаженням». Для виводу навантаження конкретного викладача у Word у випадяючому списку оберіть потрібного викладача і натисніть кнопку «Вивести у Word». Приклад вкладки «Виведення навантаження у Word» зображений на рисунку Е.14, а приклад готового файлу Word з навантаженням зображений на рисунку Е.15.

