

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ
ТА БЕЗПЕКИ**

Кафедра інформаційних технологій

**РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ
ФОРМУВАННЯ РОЗКЛАДУ ЗАНЯТЬ У НАВЧАЛЬНИХ ЗАКЛАДАХ**

Кваліфікаційна робота
здобувача вищої освіти
4 курсу заочної форми навчання
Катерини ГАЩУК

Науковий керівник:
доцент, кандидат технічних наук
Андрій ГОЛОВАТИЙ

Рецензент:

вчене звання, науковий ступінь

(Ім'я ПРІЗВИЩЕ рецензента)

Кваліфікаційна робота допущена до захисту
« ____ » _____ 2026 р., протокол № ____

Завідувач кафедри інформаційних технологій
_____ Олег ЗАЧЕК
(підпис)

Львів
2026

АНОТАЦІЯ

ГАЩУК К. Розроблення інформаційної системи автоматизації формування розкладу занять у навчальних закладах. – Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 126 «Інформаційні системи та технології». – Львівський державний університет внутрішніх справ, МВС України, Львів, 2026.

У роботі спроектовано та розроблено інформаційну систему автоматизації формування розкладу занять у навчальних закладах. Проведено аналіз існуючих інформаційних систем автоматизації формування розкладу занять у навчальних закладах. Спроектовано та розроблено інтерфейс, розроблено функції для обробки даних. Спроектовано та розроблено базу даних із підтримкою сутностей: групи, викладачі, предмети, аудиторії. Результатом є прототип системи з можливістю введення, редагування розкладу, перегляд розкладу студентами, різноманітна фільтрація та візуалізація даних.

Окрім цього, в роботі визначені перспективи її подальшого удосконалення.

Ключові слова: інформаційна система, розклад занять, веб-орієнтована система, архітектура, бази даних, php, MySQL.

ABSTRACT

Hashchuk K. Development of an information system for automating the formation of class schedules in educational institutions. – Manuscript.

Research for obtaining a bachelor's degree in specialty 126 «Information systems and technologies». – Lviv State University of Internal Affairs, MIA of Ukraine, Lviv, 2026.

The work designs and develops an information system for automating the formation of class schedules in educational institutions. An analysis of existing information systems for automating the formation of class schedules in educational

institutions is conducted. An interface is designed and developed, functions for data processing are developed. A database is designed and developed with support for entities: groups, teachers, subjects, classrooms. The result is a prototype of the system with the ability to enter, edit the schedule, view the schedule by students, various filtering and visualization of data.

In addition, the work identifies prospects for its further improvement.

Keywords: information system, class schedule, web-based system, architecture, databases, PHP, MySQL.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CSS	спеціальна мова, яка використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.
HTML	стандартизована мова розмітки документів у Всесвітній павутині.
PHP	серверна скриптова мова і потужний інструмент для створення динамічних та інтерактивних веб-сторінок.
API	Application Programming Interface (інтерфейс програмування застосунків)
SQL	Structured query language (мова структурованих запитів)
HTTP	HyperText Transfer Protocol (протокол передачі гіпертексту)
БД	База даних
ІС	Інформаційна система
ПЗ	Програмне забезпечення
СУБД	Система управління базами даних

ЗМІСТ

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ СКЛАДАННЯ РОЗКЛАДУ У НАВЧАЛЬНИХ ЗАКЛАДАХ.....	9
1.1. Аналіз предметної області.....	9
1.2. Аналіз існуючих інформаційних систем для складання розкладу.....	11
1.3. Системний аналіз.....	17
1.3.1. Побудова дерева проблем та дерева цілей.....	17
1.3.2. Аналіз вибору методів та засобів розв'язання задачі.....	23
1.3.3. Концептуальна модель інформаційної системи.....	24
1.4. Формування вимог до інформаційної системи.....	25
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ФОРМУВАННЯ РОЗКЛАДУ ЗАНЯТЬ.....	27
2.1. Структура інформаційної системи.....	27
2.1.1. Діаграми прецедентів.....	27
2.1.2. Діаграми діяльності.....	30
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ФОРМУВАННЯ РОЗКЛАДУ ЗАНЯТЬ.....	36
3.1. Проектування та створення БД для розроблюваної системи.....	36
3.2. Розроблення та характеристика користувацького інтерфейсу.....	37
3.3. Розробка програмного комплексу.....	38
3.3.1. Розробка інтерфейсу користувача.....	38
3.3.2. Розробка функцій обробки даних.....	44
3.4. Тестування розробленого програмного комплексу.....	46
3.5. Перспективи модернізації.....	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТКИ.....	57

ВСТУП

Оскільки ми живемо в час швидкого розвитку інформаційних технологій то їх впровадження в освітню сферу є неменучим, тому модернізація процесів управління у закладах вищої освіти зараз теж активно впроваджується. Можна сказати, що одним із напрямів такої модернізації є використання програмних засобів для автоматизації планування навчального процесу, зокрема створення розкладу занять. Формування розкладу занять є складним завданням, тому, що потрібно враховувати велику кількість факторів. А це – завантаженість викладачів, доступність навчальних приміщень, особливості проведення різних видів занять, а також необхідність запобігання часовим перетинам для академічних груп. Традиційний підхід до складання розкладу є трудомістким, займає багато часу та не забезпечує достатньої гнучкості коли виникає потреба у внесенні змін.

Проведений аналіз сучасних програмних продуктів для автоматизації розкладу серед яких FET, Asc Timetables, UniTime та Автоматизованої системи управління навчальним закладом показав, що існуючі рішення мають як переваги, так і обмеження. Одні системи мають відкриту архітектурою та широкі можливості налаштування, але вимагають спеціальних технічних знань для ефективного використання. Інші орієнтовані на простоту експлуатації, але поступаються за рівнем адаптивності та функціональної гнучкості. А також є комплексні платформи, які можуть обробляти великі обсяги даних та складні сценарії планування, та їх впровадження потребує багато ресурсів і є доцільним лише для великих університетів. У результаті дослідження було встановлено, що наявні програмні продукти не завжди можуть повною мірою задовольнити потреби різних освітніх установ, що обумовлює актуальність розроблення більш універсальної та гнучкої системи.

В таких умовах важливо створити автоматизований механізм формування розкладу, який забезпечить врахування всіх правил і обмежень, а

також підтримує можливість швидкого коригування результатів. Використання сучасних інформаційних технологій у цьому процесі дозволяє оптимізувати управління навчальним процесом, скоротити кількість конфліктних ситуацій, підвищити точність планування та мінімізувати вплив людського фактора. Крім того, інформаційної системи автоматизації формування розкладу занять у навчальних закладах дає змогу оперативно аналізувати різні варіанти розкладу, масштабування функціоналу та підтримки стабільної організації освітньої діяльності. Тому, впровадження таких рішень сприяє підвищенню ефективності функціонування закладу освіти та забезпечує якісне управління навчальним процесом.

Мета дослідження – розробити архітектуру та програмний прототип веб-орієнтованої автоматизованої системи формування розкладу занять для навчальних закладів, яка забезпечує базові механізми авторизації та адміністрування користувачів, збереження, обробку та відображення даних.

Для досягнення мети проекту необхідно виконати такі **завдання**:

- Провести аналіз предметної області;
- Провести аналіз існуючих інформаційних систем автоматизації формування розкладу занять у навчальних закладах;
 - Провести системний аналіз та сформулювати вимоги до інформаційної системи;
 - Побудувати комплекс UML-діаграм для моделювання архітектури та сценаріїв взаємодії користувачів із системою;
 - Спроектувати логічну та фізичну структуру бази даних;
 - Спроектувати зручний користувцький інтерфейс для роботи у системі;
 - Розробити програмне забезпечення відповідно до визначених вимог;
 - Провести перевірку працездатності системи провівши тестування.

Об'єктом дослідження є процеси які пов'язані із складання розкладу навчальних занять для закладів освіти.

Предметом дослідження виступають методи, алгоритми та програмні засоби, спрямовані на автоматизацію побудови розкладу занять для навчальних закладів із використанням Web-технологій.

У роботі використано комплекс **методів**: системний аналіз (для дослідження предметної області); об'єктно-орієнтоване проектування (для побудови архітектури); моделювання даних (для розробки бази даних) та тестування програмного забезпечення.

Практична значущість роботи полягає у розробленні прототипу інформаційної системи формування розкладу занять для навчальних закладів, який може бути впроваджений в закладі освіти. Запропонований програмний продукт дасть можливість працівникам, відповідальним за планування освітнього процесу, а також викладачам створювати та коригувати розклад відповідно до встановлених обмежень і правил.

Структура роботи. Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел, додатків. Обсяг основного тексту роботи складає 46 сторінок, 28 рисунків, 3 додатки і 23 бібліографічних джерела. Загальний обсяг роботи – 71 сторінка.

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ СКЛАДАННЯ РОЗКЛАДУ У НАВЧАЛЬНИХ ЗАКЛАДАХ

1.1. Аналіз предметної області

Організація навчального процесу у закладах вищої освіти є неможлива без якісно сформованого розкладу занять. Розклад занять є основним документом та регулюється положенням про організацію освітнього процесу [1]. Використання веб-орієнтованої інформаційної системи для автоматизованого складання розкладу дозволяє значно спростити цей процес та допомогти відповідальним за розклад особам або викладачам виконувати побудову розкладу занять. Призначенням такої системи є автоматичне формування розкладу відповідно до вимог та обмежень. Завдяки цьому відповідальні працівники отримують можливість уникнути тривалого аналізу великої кількості можливих варіантів та ручного пошуку конфліктних ситуацій. Найпоширенішою проблемою під час планування розкладу занять є виникнення часових перетинів. Коли один викладач має проводити заняття з двох або більше предметів одночасно, або коли одна група студентів повинна перебувати одночасно на двох або більше заняттях, або коли в одній аудиторії одночасно повинні проводитись два або більше занять. З іншого боку, студенти та викладачі матимуть зручний спосіб перевірити свій розклад для певної аудиторії чи розклад для викладача, із яким, припустимо, студенту необхідно зустрітись. Крім того викладачі і студенти будуть мати можливість переглядати розклад, знаходити інформацію про аудиторію зайнята вона чи ні, або переглянути графік роботи викладача.

Інформаційної системи для автоматизованого складання розкладу система має бути корисною через те, що вона дозволяє викладачам будувати розклад, не прикладаючи великої кількості зусиль для цього. Ця функція системи досягається наявністю алгоритму, що відповідно до заданих

користувачем правил перевіряє можливість побудови розкладу згідно них та, якщо це можливо, виконує таку побудову та представляє користувачу готовий розклад для подальшого внесення правок, редагування чи публікації розкладу.

Важливою перевагою такої системи також повинна стати зручність користування нею, простота процесу створення правил для розкладу та привабливого користувацького інтерфейсу із наявною великою кількістю зручних функцій. Сьогодні люди найчастіше віддають перевагу усім можливим онлайн сервісам, адже це дозволяє їм максимально ефективно використовувати свій, і без того обмежений, час. Саме тому важливо зробити систему інтуїтивно зрозумілою, а задання правил для розкладу, його побудову та подальше редагування легким та зрозумілим. Це досягається зручним та не перевантаженим зайвими деталями інтерфейсом та простотою побудови системи і її додаткових функцій.

Процес створення розкладу навчальних занять передбачає комікацію між різними підрозділами закладу освіти. Якщо є недостатня узгодженість між працівниками то це може призвести до виникнення помилок у плануванні, таких як накладання занять викладачів або конфліктів використання навчальних аудиторій. Такі ситуації погано впливають на організацію освітнього процесу та потребують додаткових ресурсів для їх усунення.

У зв'язку з цим актуальним питанням є розроблення інформаційної системи автоматизації формування розкладу занять у навчальних закладах, яка забезпечить врахування всіх вимог навчального процесу, та буде гнучкою, з можливістю масштабованості. Використання такої систем є підвищить ефективність управління освітньою діяльністю, дасть змогу швидко перебудувувати розклад коли виникнуть зміни. Також автоматизація формування розкладу занять знизить вплив людського фактора, що позитивно позначиться на стабільності та якості організації навчального процесу.

1.2. Аналіз існуючих інформаційних систем для складання розкладу

Проаналізувавши результати пошуку схожих систем та додатків у

пошуковій системі Google, стає зрозумілим те, що єдиної універсальної системи для побудови розкладу на момент пошуку не існує. Кожна із систем орієнтована на певні категорії користувачів і вирішує завдання планування навчального процесу з різною ефективністю.

На сучасному ринку є значна кількість програмних продуктів, призначених для автоматизації процесу складання розкладу занять. Вони відрізняються між собою архітектурними підходами, функціональними можливостями, методами реалізації алгоритмів планування, рівнем інтеграції з іншими інформаційними системами, вимогами до технічного забезпечення та ступенем адаптації до потреб конкретного закладу освіти. Найпоширенішими системи є FET, Asc Timetables та UniTime. Кожен із них орієнтований на власний сегмент користувачів. FET використовується невеликими освітніми установами або фахівцями, які мають достатній рівень технічної підготовки. Asc Timetables більш універсальний інструмент для середніх та великих навчальних закладів, де важливим є поєднання функціональності та зручності роботи. UniTime розроблялася для великих університетських структур, які потребують комплексної автоматизації процесів.

Під час вибору програмного забезпечення для формування розкладу треба враховувати не тільки кількість студентських груп, викладацького складу та навчальних приміщень. А й особливості організаційної структури закладу, рівень розвитку його інформаційної інфраструктури, фінансові можливості та наявність фахівців, здатних супроводжувати відповідне програмне забезпечення. Важливо є регулярно оновлювати систему, можливості адаптації системи до індивідуальних потреб установи, підтримка програмних розширень, API-інтерфейсів, механізмів резервного копіювання, друку, експорту інформації та синхронізації з іншими цифровими сервісами. Саме тому проведення аналізу існуючих програмних рішень є необхідним перед розробленням власної інформаційної системи.

Одним із найвідоміших відкритих програмних продуктів для автоматизованого створення розкладу є система **FET** (Free Timetabling Software) (рис. 1.1). Це безкоштовне програмне забезпечення має відкритий вихідний код, що надає широкі можливості для його модифікації та адаптації. Робота системи основана на використанні набору обмежень, які визначаються користувачем відповідно до особливостей організації освітнього процесу. До таких обмежень належать часові обмеження роботи викладачів, доступність аудиторій, специфіка навчальних дисциплін, допустимі інтервали між заняттями, особливості проведення лабораторних чи практичних робіт та інші фактори, що впливають на процес планування [2].

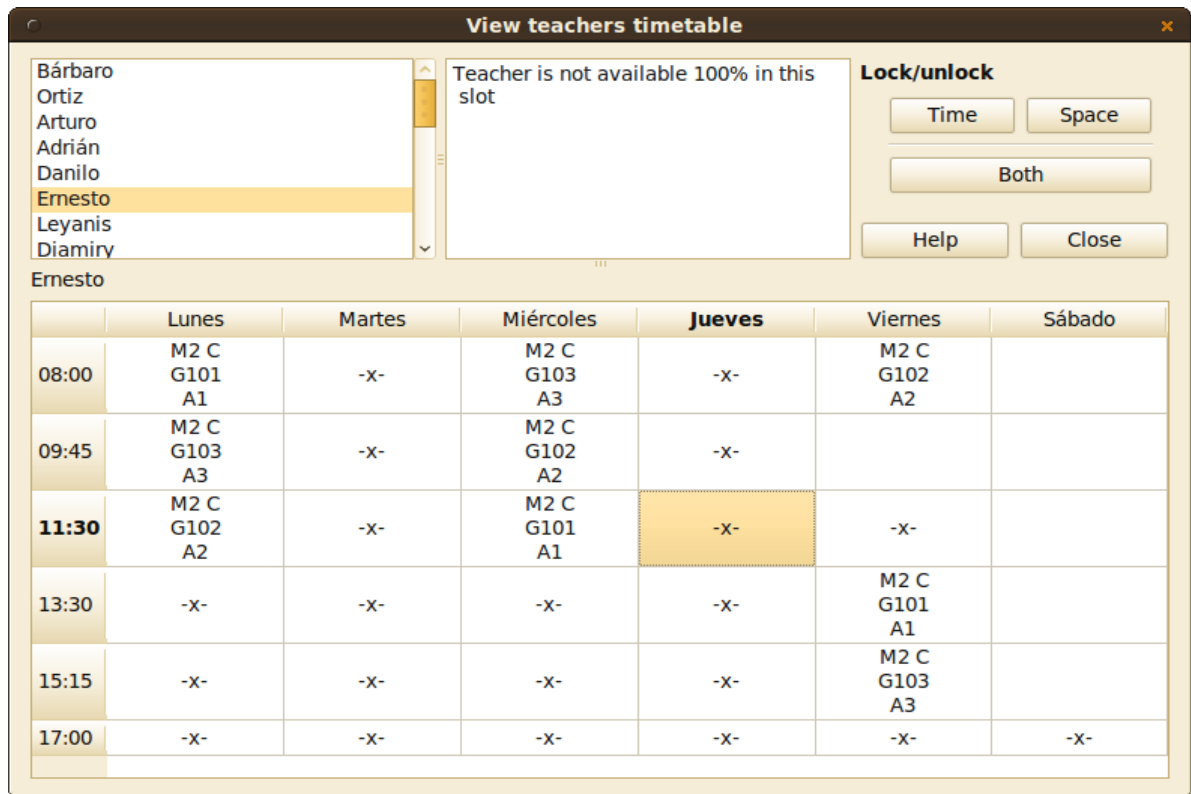


Рис. 1.1 Система FET (Free Timetabling Software)

Ця система здатна обробляти великі масиви даних і формувати розклад навіть для закладів освіти зі значною кількістю академічних груп, викладачів та аудиторного фонду. Її перевагою є можливість встановлення пріоритетності обмежень, поділяючи їх на жорсткі та бажані. Такий підхід дозволяє знаходити оптимальний баланс між обов'язковими вимогами та додатковими

побажаннями під час генерації розкладу. Функціональні можливості FET передбачають використання випадкових стартових параметрів для побудови альтернативних варіантів розкладу, підтримку підгруп студентів, роботу з дисциплінами, які викладаються кількома викладачами, а також експорт результатів у різні формати для подальшого використання. Крім того, користувачі можуть зберігати створені налаштування та використовувати їх повторно під час формування нових розкладів.

Водночас система має певні недоліки. Інтерфейс програми є доволі простим і зорієнтованим переважно на технічно підготовлених користувачів, що може створювати труднощі для працівників навчальних підрозділів без відповідного досвіду. Також відсутня можливість повноцінної інтеграції з сучасними інформаційними системами університетів та коледжів. Зміна окремих елементів уже сформованого розкладу потребує повторного запуску процесу генерації, що ускладнює оперативне внесення коректив. Додатковим обмеженням є відсутність веб-інтерфейсу, через що колективна робота над розкладом або віддалений доступ можливі лише за допомогою сторонніх інструментів. Незважаючи на це, завдяки відкритості, гнучкості налаштування та підтримці активної спільноти розробників FET залишається популярним рішенням серед багатьох освітніх установ світу [2].

Серед комерційних програм особливої уваги заслуговує **Asc Timetables** (рис. 1.2), яка належить до найпоширеніших систем автоматизованого складання розкладу. Протягом багатьох років вона використовується у школах, коледжах та університетах різних країн. Має сучасний та зрозумілий інтерфейс, який забезпечує комфортну роботу навіть користувачам без спеціальної технічної підготовки.

Asc Timetables має велику кількість функціональних можливостей таких як: перетягування об'єктів, автоматичне та ручне коригування розкладу, візуальне виділення різних категорій даних за допомогою кольорів, налаштування правил планування та створення альтернативних варіантів розкладу. Програма також дозволяє формувати шаблони занять, визначати

пріоритетність окремих подій, вести журнал змін і працювати багатьма мовами. Підтримує велику кількість форматів експорту, що спрощує поширення готових розкладів серед викладачів та студентів [3].

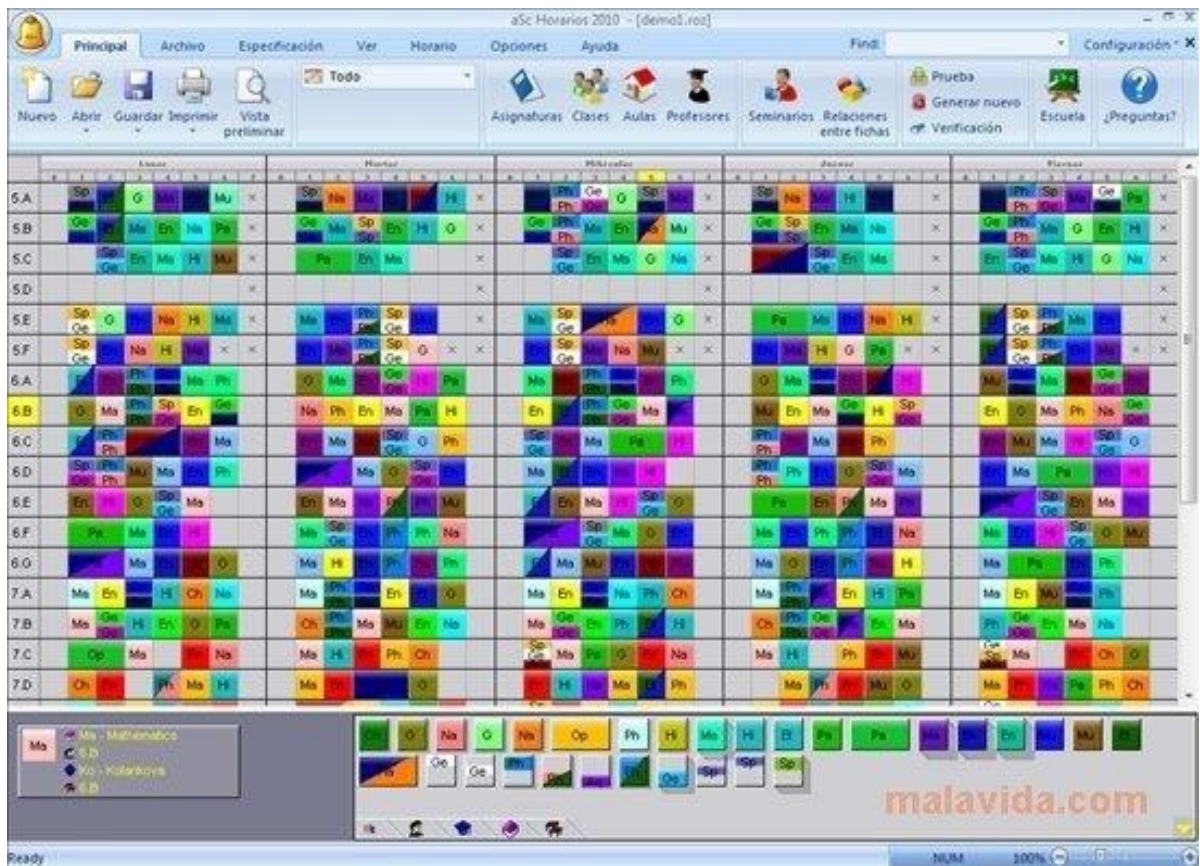


Рис. 1.2 Система Asc Timetables

Крім функцій планування, система інтегрується з іншими адміністративними підсистемами навчального закладу, включаючи електронні журнали, бази студентів та хмарні сервіси. Використання Asc Timetables має певні обмеження, оскільки, система має закритий вихідний код, що унеможлиблює модифікацію системи без залучення компанії-розробника. Безкоштовна версія містить обмежений набір функцій, а впровадження програмного комплексу у великих освітніх структурах часто потребує значного часу на налаштування та тестування [3].

Ще одним прикладом сучасної системи автоматизації є **UniTime** (рис. 1.3), яка створювалася насамперед для великих університетів та багатопрофільних освітніх комплексів. На відміну від багатьох аналогів,

UniTime охоплює не лише формування розкладу занять, а й управління аудиторним фондом, планування екзаменаційних сесій, координацію використання ресурсів та підтримку адміністративних процесів. Система була розроблена на базі багаторічних досліджень в Університеті Індіани та успішно використовується в реальних умовах навчального процесу [4].

The screenshot shows the UniTime web interface in a Mozilla Firefox browser. The URL is <https://www.smas.purdue.edu/Timetabling/selectPrimaryRole.do>. The page title is "Timetable". There are buttons for "Export PDF" and "Refresh".

The main content is a timetable grid. The first section is for "PHYS 112 (268)". The columns represent time slots from 7:30a to 5:00p. The rows represent days of the week. The grid contains various course sections with their respective room numbers and times. For example, on Monday, there are sections for ENGR 126R Lec 1 (8:30a-9:00a), OLS 274 Lec 1 (9:30a-10:00a), MA 154 Lec 2 (10:30a-11:00a), OLS 274 Lec 3 (11:30a-12:00p), PHYS 219 Lec 1 (12:30p-1:00p), OLS 274 Lec 2 (1:30p-2:00p), CGT 163 Lec 1 (2:30p-3:00p), ENGR 126A Lec 1 (3:30p-4:00p), and EPCS 101 Lec 1 (4:30p-5:00p).

The second section is for "PHYS 114 (273)". It also shows a grid of class times and rooms. For example, on Monday, there is a section for CGT 163 Lec 2 (7:30a-8:00a), PHYS 214 Lec 1 (8:30a-9:00a), ANTH 205 Lec 1 (9:30a-10:00a), PHYS 172H Lec 1 (10:30a-11:00a), MA 165 Lec 5 (11:30a-12:00p), PHYS 218 Lec 1 (12:30p-1:00p), PHYS 218 Lec 2 (1:30p-2:00p), AGECE 217 Lec 2 (2:30p-3:00p), AGECE 217 Lec 3 (3:30p-4:00p), and PSY 200 Lec 1 (4:30p-5:00p).

Рис. 1.3 Система UniTime

Архітектура UniTime побудована за модульним принципом і включає окремі компоненти для роботи з розкладом занять, екзаменів, приміщень та студентських сервісів. Система реалізована як веб-орієнтований програмний продукт із використанням сучасних технологій Java EE, Spring, Hibernate та реляційних систем керування базами даних. Передбачено підтримку різних категорій користувачів, автоматичне виявлення конфліктних ситуацій, розподіл навантаження між викладачами та академічними групами, резервування аудиторій та моделювання кількох альтернативних варіантів розкладу одночасно.

Перевагою UniTime є її здатність інтегруватися з іншими інформаційними ресурсами університету, формуючи єдине цифрове середовище управління освітнім процесом. Відкритий API забезпечує можливість розширення функціональності та розроблення власних програмних модулів. До недоліків цієї програми належать: складність розгортання серверної інфраструктури, налаштування баз даних та підготовки персоналу робить [4].

Окремо варто розглянути програмний комплекс «Автоматизована система управління навчальним закладом», який використовується у Львівському державному університеті внутрішніх справ. Це інтегроване інформаційне середовище об'єднує значну кількість взаємопов'язаних програмних модулів, призначених для підтримки різних аспектів діяльності університету. До складу комплексу входять підсистеми управління навчальним процесом, деканатами, приймальною комісією, кадровою службою та іншими структурними підрозділами. Водночас функціонування вебпорталу забезпечує доступ до розкладів, інформації про успішність студентів, навчальних планів, фінансових операцій та інших сервісів. Усі дані зберігаються в єдиній централізованій базі даних [5].

Аналіз наявних рішень дозволяє зробити висновок, що кожна із розглянутих систем має власні переваги та недоліки. FET характеризується відкритістю та широкими можливостями налаштування, проте поступається конкурентам у зручності використання. Asc Timetables відзначається простотою роботи та якісною візуалізацією, але є комерційним продуктом із закритим кодом. UniTime має високий рівень масштабованості та функціональних можливостей, але потребує значних ресурсів для впровадження. Автоматизована система управління навчальним закладом пропонує комплексний підхід до автоматизації діяльності університету, проте також характеризується складністю налаштування та супроводу.

Проведене дослідження демонструє, що сучасні програмні засоби для формування розкладів вирішують поставлені завдання різними способами,

проте жодне з рішень не може вважатися універсальним. Існує потреба у створенні більш адаптивних інформаційних систем, які поєднуюватимуть зручний користувацький інтерфейс, підтримку складних обмежень, інтелектуальні механізми прийняття рішень та можливість швидкого реагування на зміни в організації навчального процесу.

1.3. Системний аналіз

Системний аналіз є комплексним підходом до дослідження об'єктів, процесів або організаційних структур, який дає змогу визначити їхнє призначення, функції та взаємозв'язки між окремими елементами. З допомогою системного аналізу можна досягти поставлених цілей шляхом вивчення структури системи та особливостей її функціонування [6].

Із сторони практичного застосування системний аналіз можна розглядати як методологію розв'язання складних завдань декомпозиціонуванням системи на окремі складові для детального дослідження їхньої ролі та взаємодії. Такий підхід дозволяє встановити закономірності функціонування системи та визначити шляхи підвищення її ефективності. [10].

Отже, використання системного аналізу дає змогу на початковій стадії розроблення інформаційної системи виявити ключові структурні залежності, спрогнозувати можливі результати функціонування майбутньої системи, обрати технології, інструменти й методи реалізації програмного продукту.

1.3.1. Побудова дерева проблем та дерева цілей

Важливим інструментом системного аналізу є побудова дерева проблем. Воно призначене для полегшення процесу формування завдань, виконання яких необхідне для створення системи, та пошуку варіантів їх розв'язання.

Дерево проблем дає можливість встановити логічні зв'язки між першопричинами та наслідками проблеми, мінімізуючи вплив зовнішніх факторів під час аналізу.

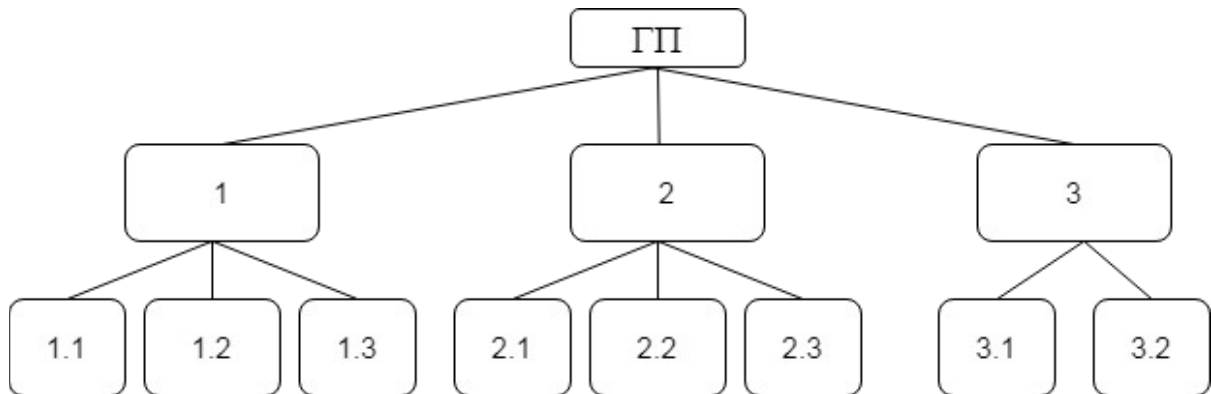


Рис. 1.4 Дерево проблем ІС

На рис. 1.4. представлено дерево проблем побудоване для інформаційної системи автоматизованого формування розкладу занять у закладах. Центральним елементом дерева є:

ГП – головна проблема, яка полягає у необхідності створення програмного засобу, здатного автоматизувати процес складання розкладу та забезпечити його зручний перегляд для всіх категорій користувачів.

Перша група проблем:

1.1 Призначення та основна мета системи:

Призначення системи – надати можливість викладачам виконувати побудову розкладу у автоматизованому режимі за допомогою задання правил та вимог до розкладу, при цьому система повинна надавати можливість неавторизованим користувачам (студентам) переглядати розклад та виконувати його фільтрацію без виконання авторизації.

1.2 Особливості функціонування середовища:

Розроблювана система працюватиме в умовах наявності аналогічних програм, що використовуються навчальними закладами. Це означає, що при створенні системи потрібно врахувати усі переваги та недоліки аналогів та правильно визначити функції системи таким чином, щоб розробити

функціонал, який забезпечить зручність використання та конкурентоспроможність програмного продукту.

Також у перспективі до системи буде надходити велика кількість інформації, що повинна ефективно зберігатись та швидко бути обробленою перед збереженням.

1.3 Шляхи досягнення поставлених цілей.

Зважаючи на конкуренцію між подібними системами, свою систему потрібно зробити такою, аби містила мінімальну кількість недоліків які матимуть інші системи та мати ряд переваг. Вона повинна бути зручною у використанні, із зрозумілим інтерфейсом та мати певні особливості, що вирізняють її з-поміж інших. Для того, щоб досягнути цих цілей необхідно приділити багато уваги дизайну та структурі front-end частини додатку.

Для того, щоб вирішити проблему із швидкою обробкою даних потрібно ефективно спроектувати базу даних та архітектуру серверної частини веб-аплікації.

Друга група проблем:

2.1 Функції системи:

Для того, щоб система була повноцінною та корисною необхідно реалізувати такий набір основних функцій:

- Вхід у систему;
- Додавання груп студентів;
- Додавання аудиторій;
- Додавання викладачів;
- Додавання предметів;
- Редагування груп студентів;
- Редагування викладачів;
- Редагування аудиторій;
- Редагування предметів;
- Редагування предметів, які може викладати викладач;
- Задання правил до побудови розкладу;

- Автоматизована побудова розкладу;
- Можливість відредагувати побудований системою розклад;
- Перегляд розкладу;
- Можливість відфільтрувати розклад за групою;
- Можливість відфільтрувати розклад за викладачем;
- Можливість відфільтрувати розклад за аудиторією;

2.2 Архітектура системи:

Правильний вибір архітектурного підходу системи є необхідним для швидкої роботи розроблюваної системи та для зручності розробника при внесенні будь-яких подальших змін. При використанні мови програмування PHP та Bootstrap - фреймворку найбільш оптимальною є побудова багаторівневої архітектури серверної частини додатку.

2.3 Принцип роботи системи:

Принцип роботи системи базується на клієнт-серверній взаємодії. Користувач працює через веб-інтерфейс, який надсилає запити до серверної частини. Сервер обробляє отриману інформацію, взаємодіє з базою даних та повертає результат клієнту для відображення.

Такий підхід забезпечує підтримку одночасної роботи великої кількості користувачів та ефективну обробку запитів.

Третя група проблем:

3.1 Організація процесу розроблення:

Для того, щоб процес створення системи проходив злагоджено, успішно та вчасно необхідно правильно організувати розробку. Варто розглянути можливість використання методологій розроблення програмного забезпечення. Потрібно також розділити роботу на чітко визначені етапи, встановити часові межі виконання кожного з них, створити детальний план робіт. Чітке планування та дотримання схеми робіт дає змогу значно зменшити ймовірність запізень у розробці та виникнення помилок на етапі реалізації.

3.2 Організація функціонування системи:

Двома основними підсистемами головної системи є клієнтська та серверна частина. Їх взаємодія повинна відбуватись наступним чином: обидві підсистеми розгортаються на різних серверах та проводять обмін повідомленнями та даними за допомогою HTTP-протоколу. Клієнтська частина надсилає запит із даними до серверної. Серверна у свою чергу обробляє отриманий запит та дані й повертає відповідь клієнтській із статус-кодом та потрібними даними. Клієнтська частина організовує дані та відображає їх у зручному для користувача вигляді.

Після побудови дерева проблем наступним етапом системного аналізу є формування дерева цілей. Чітке розуміння цілей системи дає можливість бачити її загальну картину та знати, які кроки (підцілі) потрібно виконати, щоб досягти головної цілі.

Дерево цілей системи відображає рівневий розподіл цілей на підцілі. В основі лежить головна ціль, що розгалужується на підцілі першого рівня. Далі в свою чергу кожна з підцілей першого рівня знову розгалужується на підцілі другого рівня. Процес розгалуження повторюється поки не буде досягнуто цілей.

Дерево цілей це ієрархічна структура, в якій головна мета послідовно деталізується на підцілі нижчих рівнів. Кожна підціль конкретизує окремий аспект досягнення основної мети та може, розподілятися на більш дрібні завдання. Такий підхід забезпечує логічну послідовність реалізації проєкту та дозволяє контролювати досягнення результатів на кожному етапі.

На рис. 1.5 наведено дерево цілей для розроблюваної інформаційної системи автоматизованого формування розкладу занять.

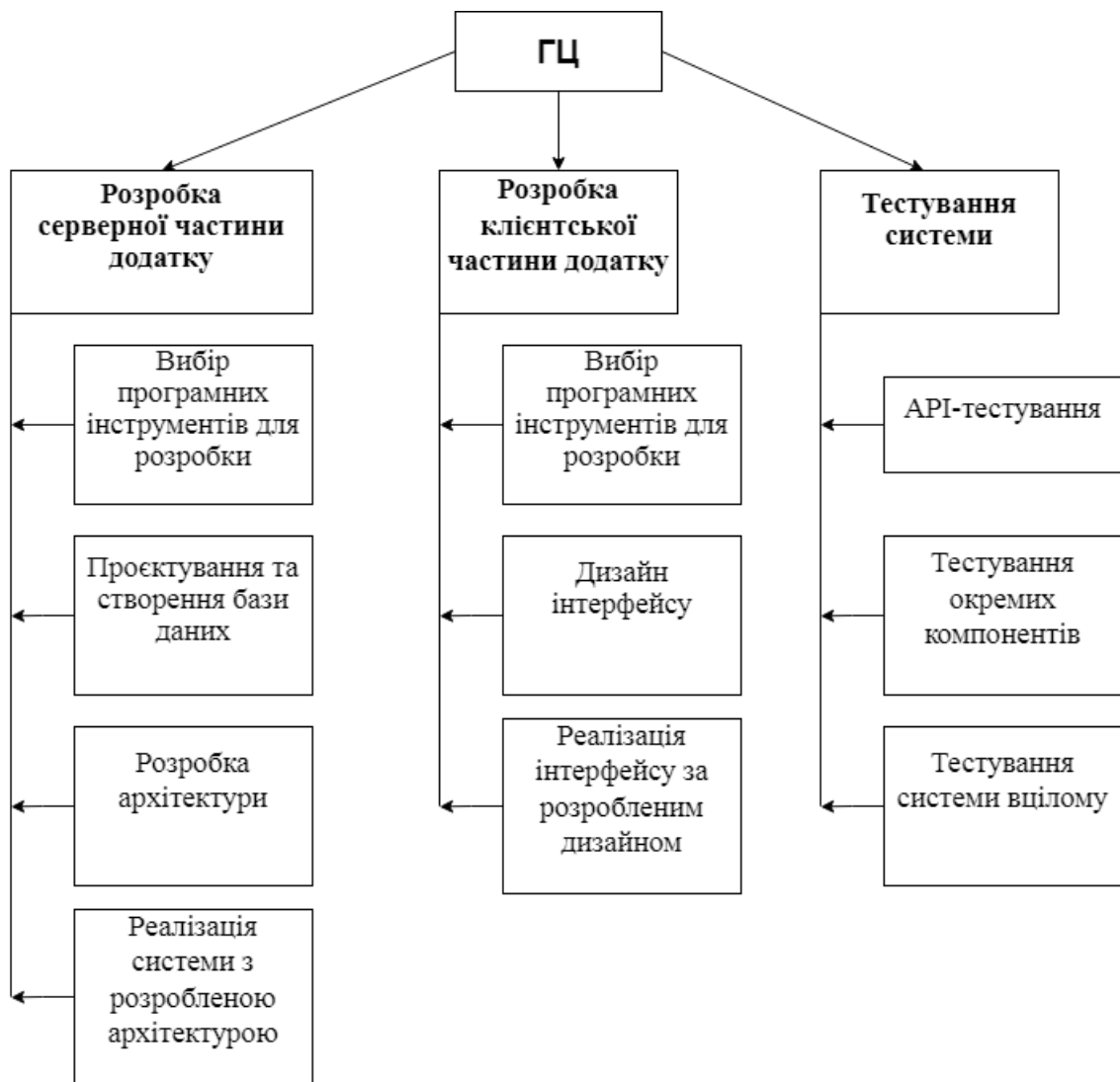


Рис. 1.5 Дерево цілей

Основною метою проекту є створення системи, що надає можливість викладачам виконати автоматизовану побудову розкладу із заданням правил та вимог до нього та надає можливість студентам провести перегляд розкладу із застосуванням різноманітних фільтрів.

Для досягнення цієї головної цілі, мети проекту потрібно спочатку виконати три підцілі.

Перша підціль – це розробка серверної частини веб-додатку. Цю ціль також, можна розбити на декілька підцілей: вибір технологій і засобів розробки, проектування структури бази даних, яка забезпечуватиме ефективно

зберігання інформації, а також розроблення архітектури програмної системи та механізмів оброблення даних.

Друга підціль – це розробка стильного, простого та зручного в користуванні графічного інтерфейсу. Для його реалізації потрібно визначити інструменти розробки, створити дизайн майбутньої системи та реалізувати інтерфейс. Особлива увага приділяється зручності використання, адаптивності та доступності системи для різних категорій користувачів.

Третя підціль – це проведення тестуванням програмного забезпечення. Контроль якості повинен здійснюватися на всіх етапах життєвого циклу системи. До комплексу перевірок належать тестування програмних інтерфейсів (API), перевірка окремих компонентів та модулів, а також комплексне тестування готового програмного продукту після завершення його розроблення. Це забезпечить стабільність роботи системи та мінімізує кількість програмних помилок.

1.3.2. Аналіз вибору методів та засобів розв'язання задачі

Найважливішим етапом проєктування інформаційної системи автоматизованого формування розкладу занять є вибір середовища розробки та програмних засобів, які використовуватимуться під час реалізації проєкту.

Для розробки back-end частини системи було обрано мову програмування PHP. Для створення веб-додатків чудово підходить Bootstrap найпопулярніший CSS-фреймворк для швидкого створення responsive (адаптивних) додатків.

Як систему управління базами даних обрано MySQL, а для її адміністрування використовується phpMyAdmin. Використання цього інструменту значно спрощує процес адміністрування, налаштування та супровіду бази даних [7].

Для реалізації серверної логіки використовується мова програмування PHP, яка широко застосовується для розроблення вебзастосунків. Побудова користувацького інтерфейсу здійснюється за допомогою HTML, CSS та

фреймворку Bootstrap, що дозволяє швидко створювати адаптивні вебсторінки для різних типів пристроїв.

Клієнтська та серверна частина розгортатимуться на різних серверах та обмінюватимуться повідомленнями та даними за допомогою HTTP-протоколу. Користувач взаємодіє із front-end частиною за допомогою браузера. Клієнтська частина надсилає запити із даними до серверної. Серверна у свою чергу обробляє отриманий запит та дані й повертає відповідь клієнтській із статус-кодом та потрібними даними. Клієнтська частина організовує дані та відображає їх у зручному для користувача вигляді.

Для реалізації проєкту обрано інтегроване середовище розробки Visual Studio Code. Він достатньо популярний через високу швидкодію, підтримку великої кількості мов програмування.

Отже, середовищем розробки буде Visual Studio Code, PHP 8.4, HTML5, CSS3, Bootstrap 5.1, MySQL та phpMyAdmin. Ці засоби забезпечать створення інформаційної системи, яка зможе вирішувати завдання автоматизованого формування розкладу навчальних занять.

в якості середовища розробки було вирішено використати середовище Visual Studio Code. HTML 5, CSS 3 та PHP 8.4 – це основні складові в розробці. PhpMyAdmin є веб-інтерфейсом для управління базами даних MySQL, фреймворк Bootstrap 5.1 із їх допомогою можна створити дизайн сторінки, підключити до бази даних та реалізувати функціонал для складання розкладу занять, що чудово підходить для розроблюваної системи.

1.3.3. Концептуальна модель інформаційної системи

Концептуальна модель визначає загальну структуру інформаційної системи відображає взаємодію між користувачами, взаємозв'язки між ними та функціональні елементи програмного комплексу (рис. 1.6)



Рис. 1.6 Концептуальна модель системи для побудови розкладу

На наведеній вище моделі бачимо, що у ролі користувачів системи постають викладачі – ті хто мають можливість зареєструватись та увійти до системи після чого провести побудову розкладу та його редакцію, адміністратори, які матимуть права як і у викладачів і можуть змінювати інформацію про навчальний заклад, надавати або забирати роль адміністратора у інших користувачів та студенти, які мають можливість перейти на сайт та провести перегляд розкладу, його фільтрацію за одним із доступних критеріїв. [6, 10].

1.4. Формування вимог до інформаційної системи

Важливим етапом проєктування програмного забезпечення є визначення функціональних та нефункціональних вимог до майбутньої системи. Саме на цьому етапі встановлюються характеристики, яким повинна відповідати інформаційна система.

Основні вимоги до інформаційної системи:

- розміщення на локальному сервері навчального закладу;
- використання СУБД MySQL 8.4;
- використання PHP 8.4;
- наявність простого та інтуїтивно зрозумілого інтерфейсу;

- коректна робота на різних типах пристроїв;
- обмеження доступу неавторизованих користувачів лише до перегляду розкладу;
- можливість редагування власного розкладу викладачами;
- реєстрація користувачів виключно головним адміністратором;
- керування групами, предметами та викладачами адміністраторами системи;
- управління структурою відділень лише головним адміністратором;
- врахування всіх можливих варіантів проведення занять під час формування розкладу;
- недопущення конфліктів між аудиторіями, викладачами та навчальними групами;
- правильне відображення розкладу після експорту до Microsoft Excel.

У результаті інформаційна система повинна забезпечувати стабільну роботу, відповідати потребам навчального закладу та мати можливість подальшого розширення функціоналу.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ФОРМУВАННЯ РОЗКЛАДУ ЗАНЯТЬ

2.1. Структура інформаційної системи

Для відображення архітектури розробленої інформаційної системи було використано інструменти UML-моделювання, зокрема діаграми прецедентів та діаграми діяльності. Такі засоби дозволяють представити функціональні можливості системи, взаємодію користувачів із нею та основні бізнес-процеси.

Діаграма прецедентів (Use Case Diagram) є одним із ключових елементів опису функціональних вимог до програмного забезпечення. Вона демонструє набір дій, які можуть виконувати користувачі, та визначає взаємозв'язки між ними. Основною перевагою такого підходу є можливість розглядати систему з позиції кінцевого користувача, акцентуючи увагу на очікуваному результаті роботи, а не на деталях реалізації окремих функцій. Використання діаграм прецедентів сприяє кращому розумінню вимог і забезпечує ефективну комунікацію між розробниками та майбутніми користувачами системи [8, 11, 23].

2.1.1. Діаграми прецедентів

Під час проєктування інформаційної системи було розроблено три діаграми прецедентів, кожна з яких відповідає окремій категорії користувачів.

На рис. 2.1 наведено діаграму прецедентів для ролі **Студент**. Вона відображає основні можливості користувача, серед яких перегляд розкладу занять та застосування різних критеріїв фільтрації. Для отримання необхідної інформації студент може обрати фільтр за навчальною групою, аудиторією або іншими параметрами.

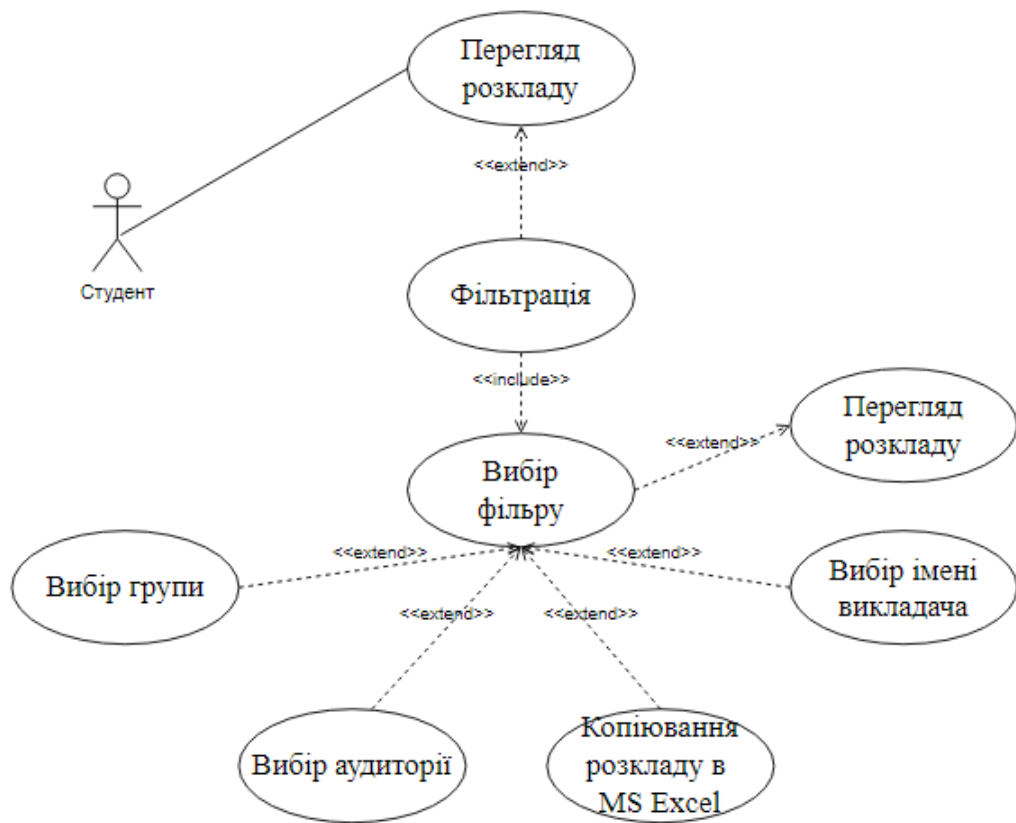


Рис. 2.1 Діаграма прецедентів для користувача Студент

На рис. 2.2 представлено побудовану діаграму прецедентів для ролі **Головний адміністратор**. Цей користувач має найширший набір повноважень. До його функцій належать перегляд і фільтрація розкладу, реєстрація та авторизація, керування інформацією про навчальний заклад, налаштування ролей користувачів і призначення адміністративних прав.

Головний адміністратор може також створювати та редагувати розклад, виконувати його автоматизовану генерацію та задавати необхідні параметри для цього процесу: назви навчальних дисциплін, груп, викладачів, аудиторій, кількість занять протягом тижня та інші обмеження. Також передбачено можливість коригування окремих елементів розкладу шляхом зміни часу проведення занять, викладача або аудиторії.

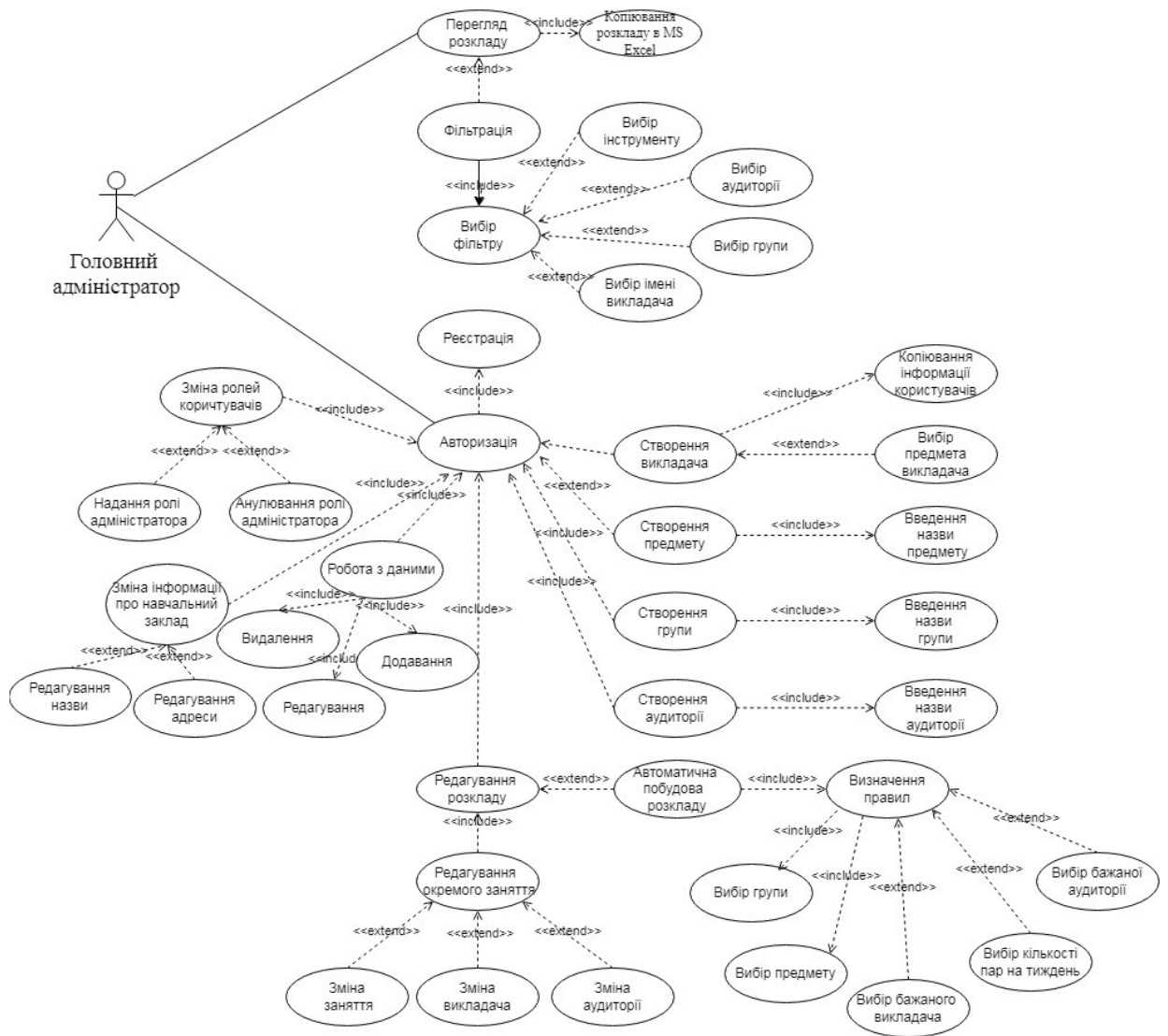


Рис. 2.2 Діаграма прецедентів для користувача Головного адміністратора

Ще одна діаграма прецедентів була створена для користувача **Викладач** яка наведена на рис. 2.3. Вона містить функції перегляду та фільтрації розкладу, експорту даних до формату MS Excel, реєстрації та авторизації в системі.

Також викладач має доступ до інструментів автоматизованого формування розкладу та його подальшого редагування. Для створення розкладу користувач задає перелік дисциплін, навчальних груп, викладачів, аудиторій і кількість занять на тиждень. Після формування розкладу доступне внесення змін до окремих занять, включаючи зміну часу, аудиторії або викладача.

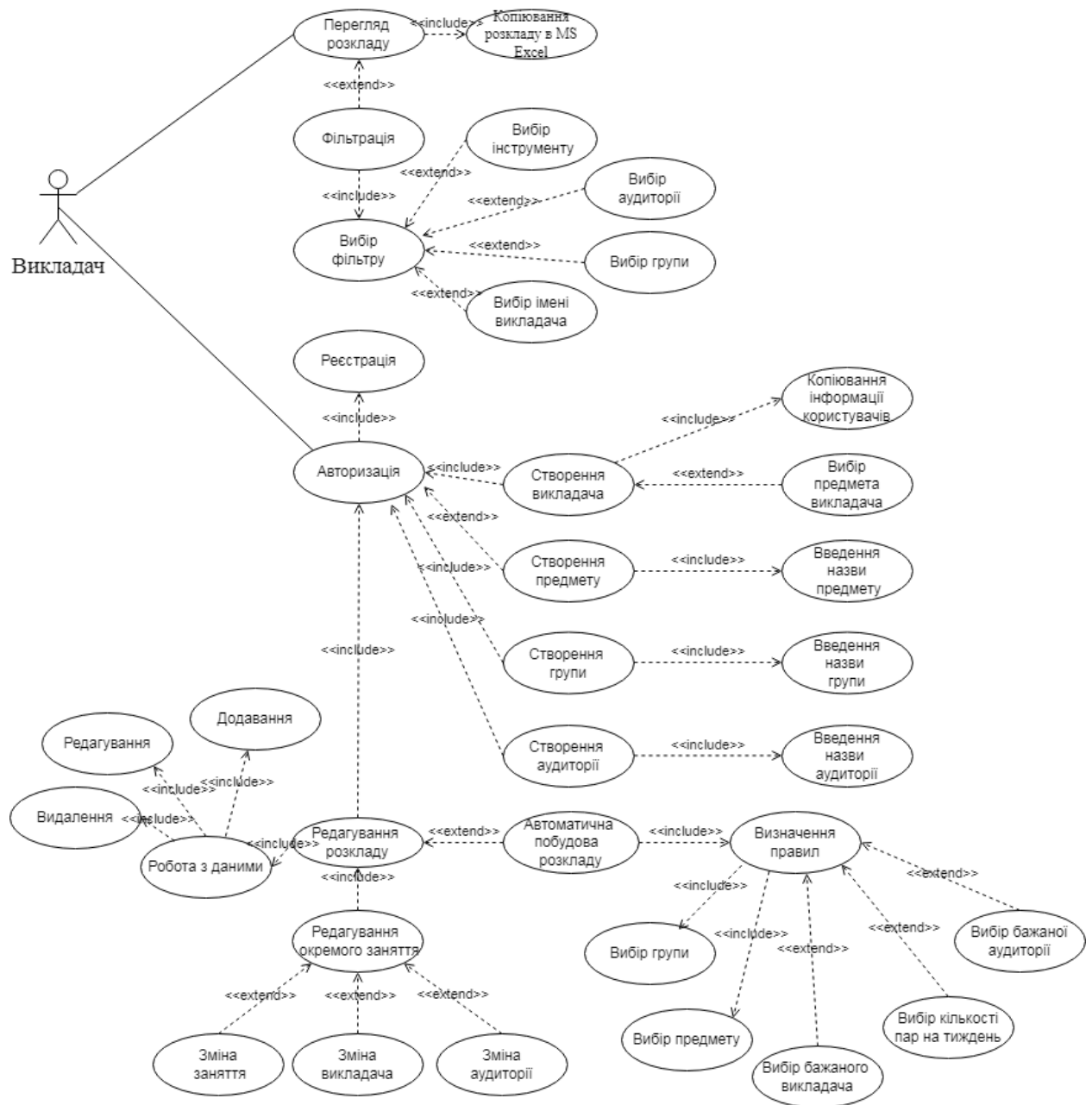


Рис. 2.3 Діаграма прецедентів для користувача Викладач

Функціональні можливості системи залежать від ролі користувача. Найбільший рівень доступу має головний адміністратор, тоді як студент володіє лише базовими можливостями перегляду інформації.

2.1.2. Діаграми діяльності

Для моделювання внутрішньої логіки роботи системи були використані діаграми діяльності. Вони дають змогу відобразити послідовність виконання операцій, альтернативні сценарії та переходи між окремими етапами процесів.

Діаграми діяльності широко застосовуються для візуалізації бізнес-процесів, алгоритмів та потоків керування. На відміну від діаграм взаємодії, які акцентують увагу на обміні повідомленнями між об'єктами. Діаграми діяльності показують порядок виконання дій і переходів між ними. Це дозволяє детальніше описати поведінку системи та логіку реалізації її функціональних можливостей [9].

На рис. 2.4 показано процес реєстрації нового користувача. Він включає введення реєстраційних даних, створення пароля та його підтвердження.

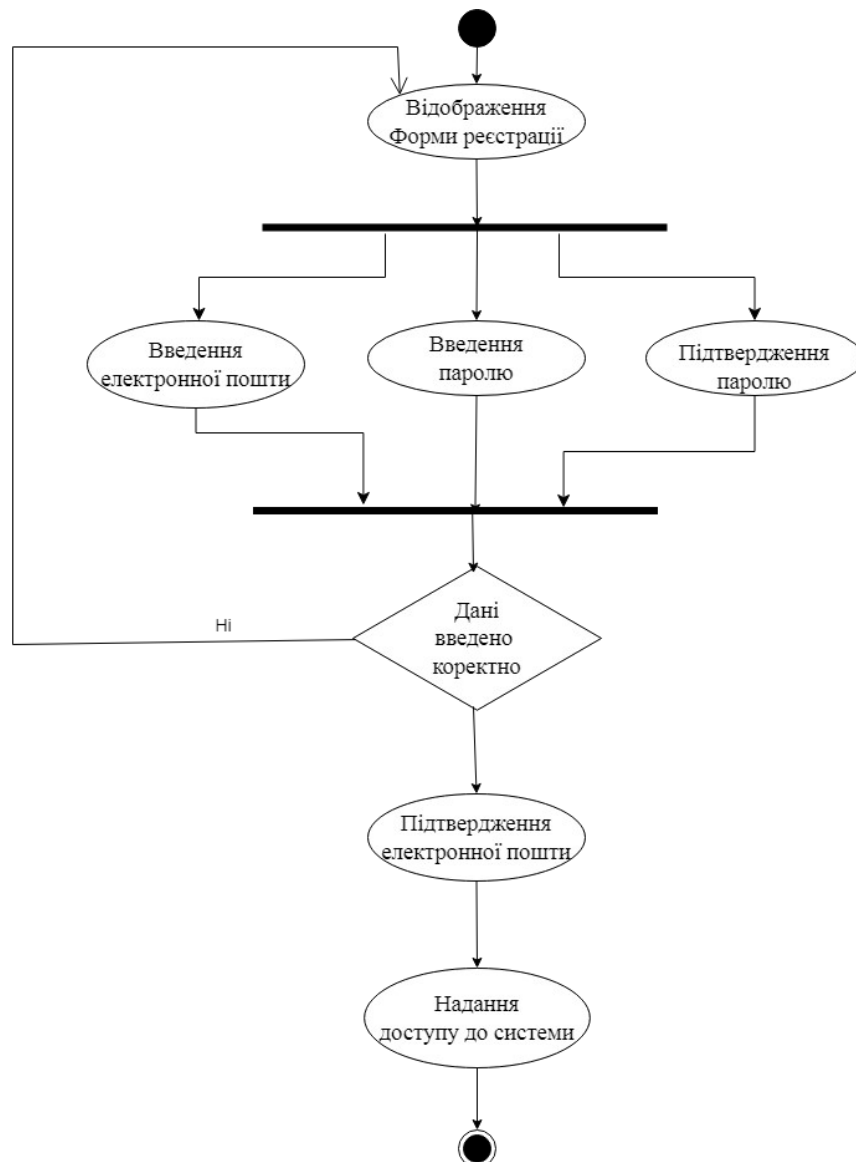


Рис. 2.4 Діаграма діяльності для реєстрації в системі

На рис. 2.5 наведено діаграму діяльності авторизації в системі. Для успішного входу користувач повинен ввести адресу електронної пошти та пароль.

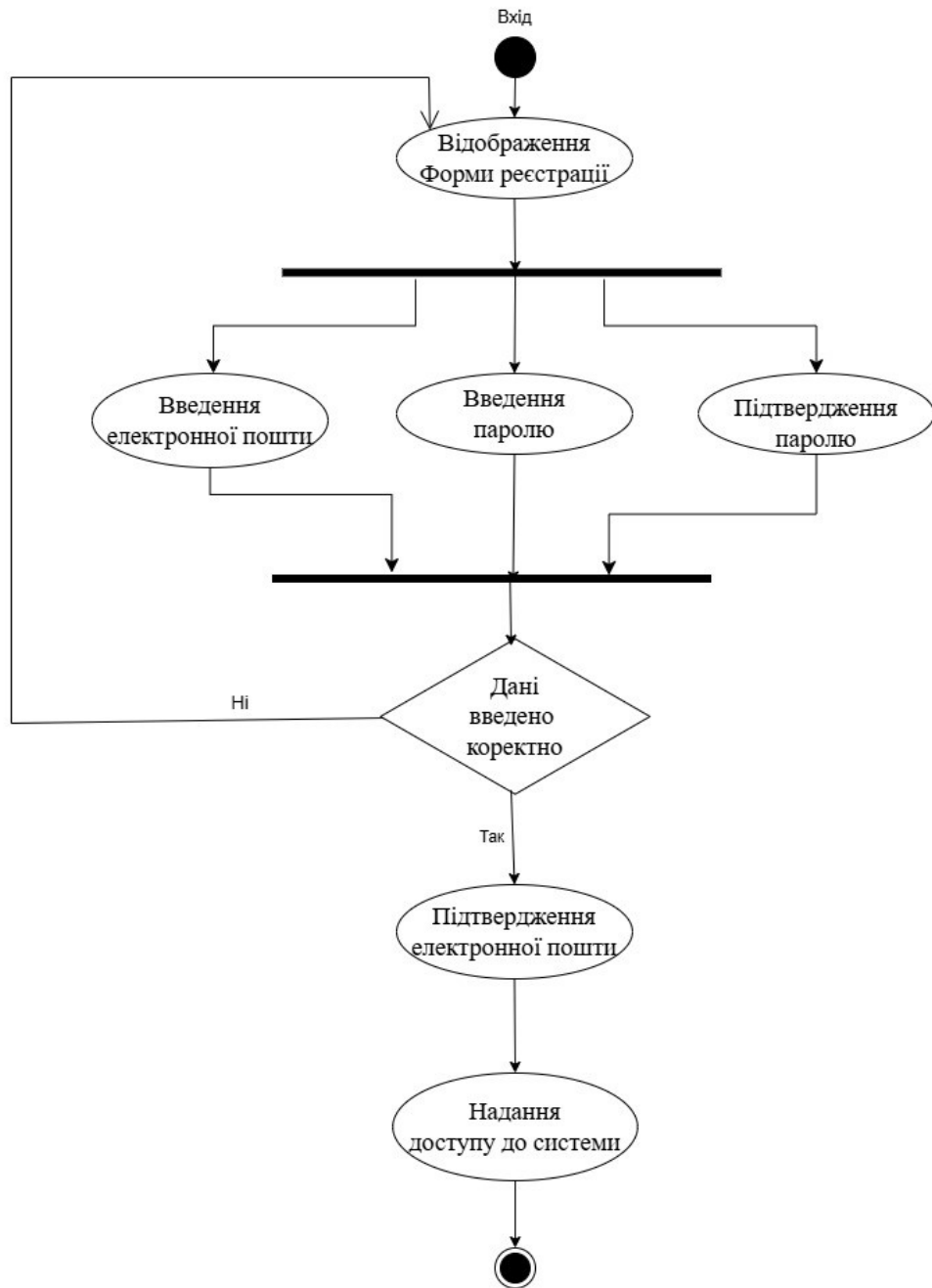


Рис. 2.5 Діаграма діяльності для входу

На рис. 2.6 зображено процес перегляду розкладу. Користувач обирає потрібний тип розкладу та застосовує потрібні параметри фільтрації для отримання актуальної інформації.

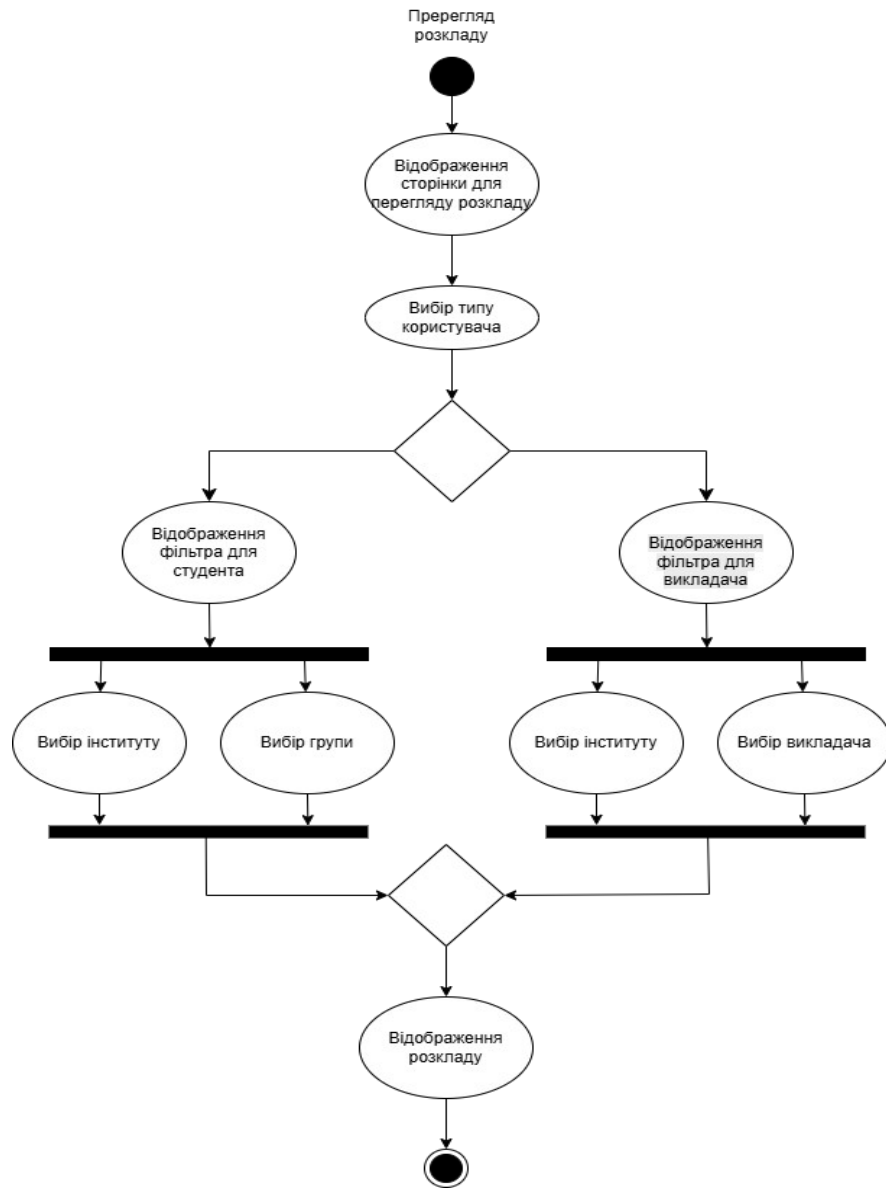


Рис. 2.6 Діаграма діяльності для перегляду розкладу

Діаграма діяльності автоматизованого формування розкладу наведена на рис. 2.7. Процес передбачає вибір навчальних груп, вибір дисциплін, викладачів та аудиторій, після чого система автоматично буде розклад відповідно до заданих параметрів.

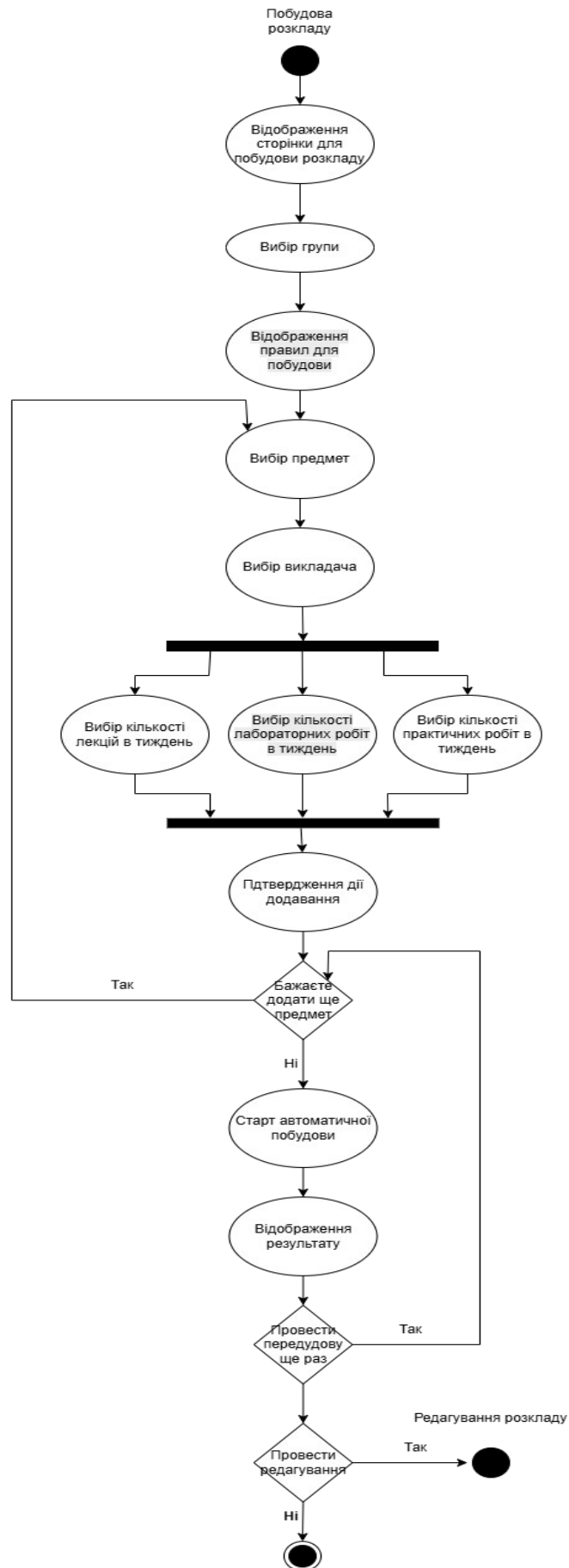


Рис. 2.7 Діаграма діяльності для побудови розкладу.

Після завершення автоматичного створення розкладу користувач може внести необхідні корективи. Процес редагування передбачає вибір конкретного заняття та зміну його параметрів, зокрема часу проведення, викладача або аудиторії. Відповідна діаграма діяльності представлена на рис. 2.8.

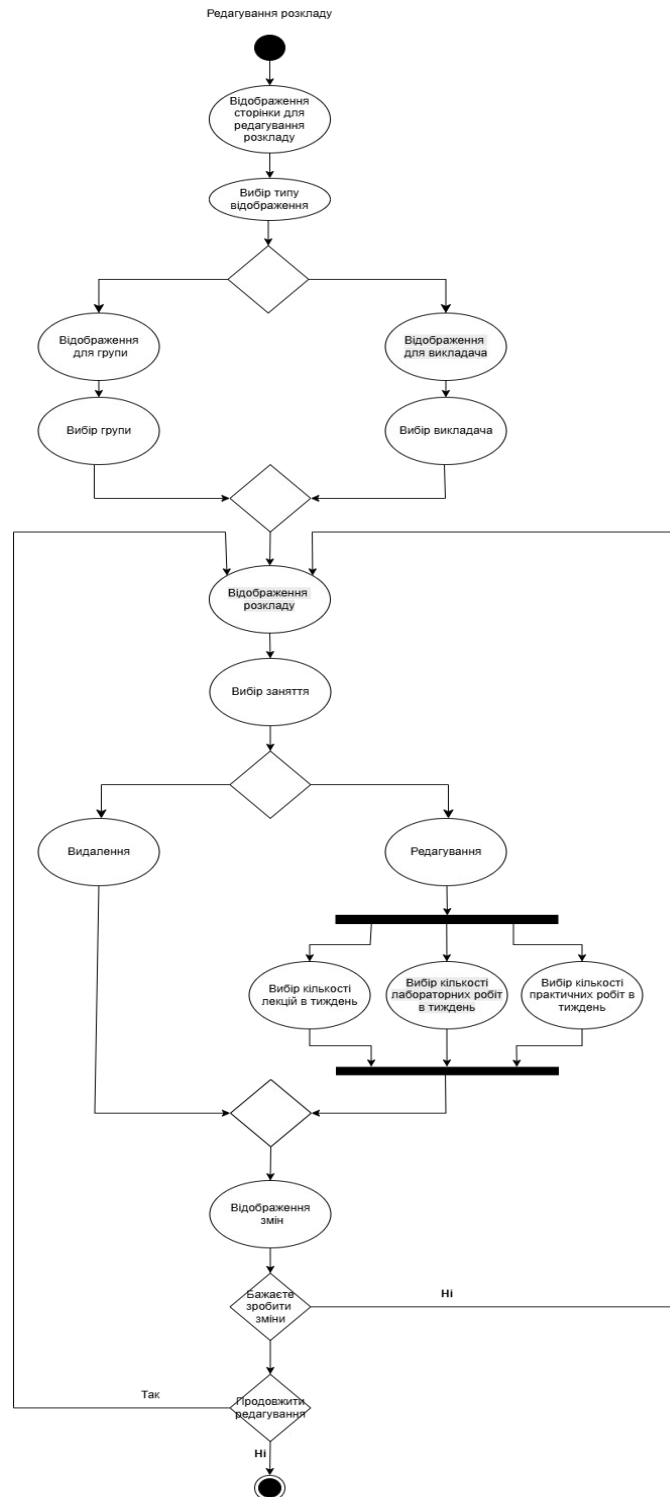


Рис. 2.8 Діаграма діяльності редагування розкладу

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ФОРМУВАННЯ РОЗКЛАДУ ЗАНЯТЬ

3.1. Проектування та створення БД для розроблюваної системи

Для забезпечення зберігання та обробки інформації в розробленій системі було створено реляційну базу даних за допомогою середовища PhpMyAdmin, який є веб-інтерфейсом для адміністрування СУБД MySQL. Використання цього інструменту дозволило швидко виконати проектування структури бази даних завдяки його зручному графічному інтерфейсу та широким можливостям керування даними.

Структура створеної бази даних наведена на рис. 3.1. Вона складається з дев'яти взаємопов'язаних таблиць, призначених для зберігання інформації, необхідної для формування та відображення навчального розкладу [12].

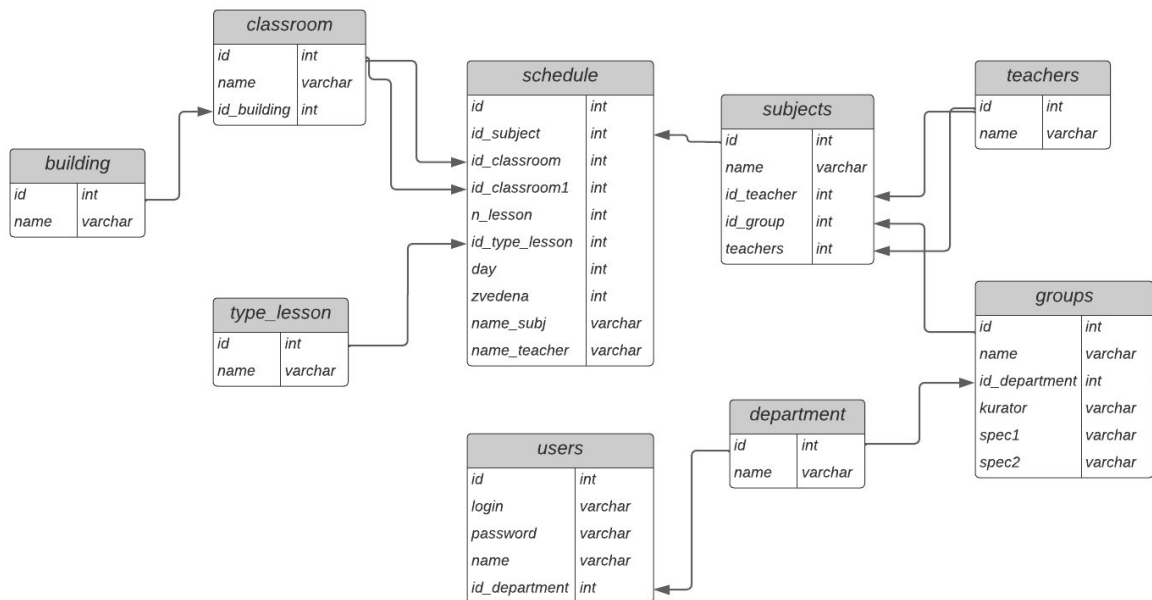


Рис. 3.1 Структура бази даних

В базі даних є таблиця **schedule**, яка містить записи про всі заняття. Для кожного заняття зберігаються відомості про день проведення, номер пари, тип

заняття, навчальну дисципліну, аудиторію та викладача. Також є спеціальне поле для позначення спільних занять, які проводяться одночасно для декількох академічних груп.

Таблиця **type_lesson** використовується для зберігання переліку можливих типів занять. Інформація про навчальні корпуси міститься в таблиці **building**, тоді як усі аудиторії зберігаються в таблиці **classroom**.

Дані про викладацький склад навчального закладу розміщуються в таблиці **teachers**, де для кожного викладача вказується його прізвище, ім'я по батькові.

Таблиця **groups** містить інформацію про академічні групи, їхніх кураторів, спеціалізації та належність до відповідного структурного підрозділу. Для опису навчальних дисциплін використовується таблиця **subjects**, яка містить назву предмета та зв'язки з викладачами й групами.

У таблиці **department** зберігаються відомості про структурні підрозділи навчального закладу. Дані зареєстрованих користувачів системи зберігаються в таблиці **users**, де також реалізовано механізм розмежування прав доступу залежно від належності користувача до певного відділення.

3.2. Розроблення та характеристика користувацького інтерфейсу

Після запуску веб-застосунку користувач потрапляє на головну сторінку, де відображається розклад занять. Для пошуку необхідної інформації передбачено систему фільтрів, яка дозволяє вибирати дані за викладачем, аудиторією або академічною групою.

Під час створення дизайну інформаційної системи було обрано синьо-білу кольорову гаму, яка відповідає стилю навчального закладу. Таке поєднання кольорів забезпечує гарну читабельність інформації та зручність роботи користувачів із системою.

Основним елементом інтерфейсу є таблиця розкладу, що використовується для відображення навчальних занять. Для навігації використано кнопки, меню та випадаючі списки.

Під час проектування форм введення та редагування інформації було використано такі елементи інтерфейсу:

- кнопки керування;
- текстові поля для введення даних;
- списки вибору;
- прапорці для встановлення додаткових параметрів.

Застосування цих компонентів дозволило створити зрозумілий та зручний інтерфейс, який не потребує тривалого навчання користувачів і забезпечує швидкий доступ до основних функцій системи.

3. 3. Розробка програмного комплексу

Створення програмного комплексу здійснювалося поетапно: проектування користувацького інтерфейсу та реалізацію програмної логіки. Це дозволило забезпечити зручність взаємодії користувачів із системою та коректне виконання всіх функціональних можливостей.

3. 3. 1. Розробка інтерфейсу користувача

Першим етапом стала розробка головного елемента системи — таблиці для відображення розкладу занять. Вона була спроектована таким чином, щоб забезпечити зручне сприйняття інформації та швидкий доступ до необхідних даних (рис. 3.2) [12, 13].. Для покращення читабельності було використано оформлення з чітким розмежуванням комірок та заголовків.

Програмний код представлено у додатку А.

1			
2			
3			
4			

Рис. 3.2 Таблиця для виведення розкладу

Потім створено засоби навігації. До їх складу увійшли кнопки керування та випадаючі меню [14, 15], які забезпечують швидкий доступ до функцій системи. Частина пунктів меню доступна лише авторизованим користувачам відповідно до їхніх ролей та прав доступу (рис. 3.3).



Рис. 3.3 Елементи навігації

Нижче представлено фрагмент програмного коду зображених на рисунку 3.3. елементів навігації.

Фрагмент програмного коду містить теги: `div`, `button` та `ul` [14, 15].

```

<div class="dropdown col-sm-1">
  <button class="btn btn-outline-secondary dropdown-
toggle" type="button" id="dropdownMenuButton2" data-bs-
toggle="dropdown" aria-expanded="false">Профіль</button>
  <ul class="dropdown-menu" aria-
labelledby="dropdownMenuButton2">
    <?php if(empty($ SESSION['id'])) { ?>
      <li data-bs-toggle="modal" data-bs-
target="#staticBackdrop0"><a class="dropdown-
item">Увійти</a></li><?php } ?>
      <?php if($ SESSION['id department']=="1"){ ?>
      <li><a class="dropdown-item" href="index.php">
Переглянути розклад</a></li>
      <li><a class="dropdown-item" href="change.php">
Редагувати розклад</a></li>
      <li><a class="dropdown-item" href="users.php">
Користувачі</a></li>
      <li><a class="dropdown-item"
href="settings_subj.php?v=2">Налаштування</a></li>
      <li><a class="dropdown-item" href="exit.php">
Вихід</a></li>
      <?php } ?>
    </ul>
</div>

```

Реалізація форми для входу адміністраторів зображена на рис. 3.4. Форма для входу адміністраторів складається із двох полів в які можна ввести логін і пароль та кнопки увійти [16].

Рис. 3.4 Вхід для адміністратора

Процес створення або редагування розкладу занять відбувається наступним чином. У режимі редагування розкладу занять в кожній вільній клітинка знаходиться іконка [17] – кружечок з хрестиком всередині, як це

зображено на рис. 3.5. Після його вибору цієї іконки відкривається форма введення параметрів пари, де адміністратор зможе створити нову пару. У адміністратора буде можливість вибору типу пари: звичайна пара, поділ на верх чи низ, тобто чисельник чи знаменник, поділ по спеціальностям.



Рис. 3.5 Іконки для додавання пар

Залежно від вибраного типу заняття система автоматично формує необхідний набір полів введення. Адміністратор може обрати навчальну дисципліну, викладача, аудиторію та встановити додаткові параметри, заняття для декількох груп (рис. 3.6).

Для управління довідковою інформацією були створені окремі сторінки адміністрування. Вони призначені для роботи з користувачами, структурними підрозділами, академічними групами, дисциплінами та викладачами. Усі сторінки мають єдиний стиль оформлення та містять таблиці з можливістю додавання, редагування та видалення записів.

Код такого блоку наведено у додатку Б.

-

Відображається в розкладі:

14 (1к)
15 (1к)
17 (1к)
18 (1к)

*Для вибору декількох кабінетів зажміть Ctrl

Зведена пара

Рис. 3.6 Вигляд форми для додавання пари

Наступним було розроблено зручний інтерфейс сторінок для управління даними. Цей інтерфейс важливий для ефективного формування розкладу занять.

Першою розробили сторінку керування адміністраторами. На ній розміщено таблицю з інформацією про працівників. У нижній частині таблиці розташована кнопка «Додати нового адміністратора». У кожному рядку з даними користувача (крім головного адміністратора) є кнопка редагування.

Сторінка відділень закладу також побудована на основі таблиці (рис. 3.7), подібної до сторінки адміністраторів. Біля кожного відділення є кнопка редагування, а в кінці таблиці є кнопка додавання нового відділення [16, 17].

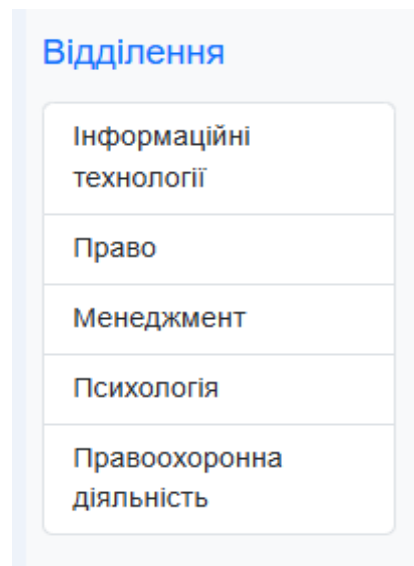


Рис. 3.7 Таблиця відділень закладу

Схожі таблиці використовуються на сторінках груп (рис. 3.8) та предметів. На цих сторінках передбачено бічне меню: на сторінці груп для вибору відділень, на сторінці предметів для вибору груп. Кожне відділення та кожна група мають власну таблицю з ідентичним набором функцій і дизайном. Програмний код відповідної сторінки наведено в додатку В.

Налаштування груп Профіль ▾
Львівського державного університету внутрішніх справ

Відділення

- Інформаційні технології
- Право
- Менеджмент
- Психологія
- Правоохоронна діяльність

Інформаційні технології

№	Назва групи	Куратор	Спеціальності	Дії
1	ІТ-11	Огірко О.		Редагувати
2	ІТ6-11	Мідик А.		Редагувати
3	ІТ-21	Дяков А.		Редагувати

Право

№	Назва групи	Куратор	Спеціальності	Дії
1	ІТ-4	Рудий Т.		Редагувати

[+ Додати нову групу](#)

Рис. 3.8 Зовнішній вигляд таблиці з групами

Після завершення розробки інтерфейсної частини було створено повний набір засобів для роботи з даними, що стало основою для подальшої реалізації функціональних можливостей системи.

3.3.2. Розробка функцій обробки даних

Найосновнішою функцією розроблюваної інформаційної системи автоматизації формування розкладу занять є взаємодія з функціями роботи з базою даних, зокрема різноманітні SQL-запити. Нижче наведено фрагмент коду з прикладами запитів на отримання списків викладачів та аудиторій [18, 19].

```
$sql="SELECT * FROM `teachers` WHERE `id`!='1' AND `id`!='2'
ORDER BY `name` ASC";
$res=mysqli_query($connect,$sql);
$array_teachers=[];
while ($result = mysqli_fetch_array($res)) {
    $array_teachers[]=$result;
}

$sql="SELECT * FROM `classroom`";
$res=mysqli_query($connect,$sql);
$array_cl=[];
while($result=mysqli_fetch_array($res)){
    $array_cl[] = $result;
}
```

Важливу роль відіграє функція автоматичного формування таблиці розкладу. Під час її виконання відбувається обробка отриманих даних та їх подальше виведення у вигляді структурованої таблиці. Для реалізації даного процесу використовуються циклічні конструкції та умовні оператори[9], що забезпечують правильне відображення інформації залежно від її типу та структури.

Спочатку будується шапка розкладу, яка містить назву групи та прізвище куратора. Якщо група поділяється за спеціальностями, їх назви відображаються одразу після куратора.

```

for ($g = 0; $g < count($array_groups); $g++) {
    $n=0;
    for ($ar = 0; $ar < count($array_1); $ar++) {
if($array_groups[$g]['id'] == $array_1[$ar]['group']){
    if($array_1[$ar]['id_type_lesson']=='1'){ echo "<td
colspan='4' style='height:
    23px; border-bottom: hidden; text-align: right; font-style:
    italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar
]['id_classroom1'], $array_cl)."</td>";
        $n=1;
    }
    if($array_1[$ar]['id_type_lesson']=='2'){ echo "<td
colspan='2' style='height:
    23px; border-right: hidden; border-bottom: hidden; text-
align: left; font-style: italic;'>
".clas($array_1[$ar]['id_classroom'],$array_1[$ar]['id_classroo
ml'
    ], $array_cl)."</td><td colspan='2' style='height: 23px;
border-bottom: hidden; text-align: right; font-style:
italic;'>".$array_1[$ar]['name_teacher']."</td>";
        $n=1;
    }
}
}

```

Ще у системі реалізовано низку допоміжних функцій для взаємодії з базою даних:

- додавання нових записів;
- оновлення (збереження змін) даних;
- видалення записів.

Ці функції доступні виключно адміністраторам на відповідних сторінках для управління предметами, групами, викладачами, відділеннями та користувачами [20]. Фрагмент однієї з таких функцій наведено нижче:

```

if (isset($ POST['new'])) {
    if($ POST['teach']){
        if (count($_POST['teach'])==2){
            $sql = "INSERT INTO `subjects`(`name`,
`id_teacher`, `id_group`, `teachers`) VALUES
('".$_POST['name']."', '".$_POST['teach'][0]."',
".$_POST['group']."', '".$_POST['teach'][1]."'");";
        } else {
            $sql = "INSERT INTO `subjects`(`name`,
`id_teacher`, `id_group`, `teachers`) VALUES
('".$_POST['name']."', '".$_POST['teach'][0]."',
".$_POST['group']."', '1')";";
        }
    } else {
        $sql = "INSERT INTO `subjects`(`name`, `id_teacher`,
`id_group`, `teachers`) VALUES ('".$_POST['name']."', '1',
".$_POST['group']."', '1')";";
    }
    $res=mysqli_query($connect,$sql);
    ?>
    <script>
        document.location.href =
"settings_subj.php?v=<?php echo $_GET['v']; ?>"
    </script>
    <?php
}

```

Після завершення розробки всього необхідного функціоналу програмного комплексу можна переходити до етапу тестування [21, 22].

3.4. Тестування розробленого програмного комплексу

Для перевірки належної роботи системи було проведено ручне тестування всіх компонентів системи. У ході перевірки виявлені окремі недоліки були усунуті, після чого здійснено повторний контроль коректності роботи системи.

Першим етапом стала перевірка інтерфейсу головної сторінки. Результати тестування підтвердили правильність відображення навігаційних елементів та блоку розкладу занять (рис. 3.9).

Розклад занять
Львівського державного університету внутрішніх справ

Оберіть групу: ІТ-11
Оберіть відділення: Всі відділення

Показати

Пара	ІТ-11 Огірко О.	ІТ6-11 Мідик А.	ІТ-21 Дяков А.	ІТ6-21 Мусійовська М.	ІТ-31 Галайко Н.	ІТ6-31 Зачек О.	ІТ-41 Магеровська Т.
1	Математика	Перерва	Історія		Фізика		Математика
2	Програмування	Математика		Аналітика		Біологія	
3	Бази даних	Фізика	Математика		Історія		Програмування
4	Англійська		Програмування	Історія		Математика	
5	Історія	Англійська			Фізика		Перерва

Копіювати розклад у MS Excel

Рис. 3.9 Головна сторінка системи

Наступним кроком було тестування механізму автентифікації користувачів. Для входу в систему необхідно перейти до розділу профілю. Після успішної авторизації користувач повертається на головну сторінку, однак у меню профілю стають доступними додаткові функції, призначені лише для зареєстрованих користувачів (рис. 3.10).

Розклад занять
Львівського державного університету внутрішніх справ

Оберіть групу: ІТ-11
Оберіть відділення: Всі відділення

Профіль

- Увійти
- Переглянути розклад
- Редагувати розклад
- Налаштування
- Вихід

Пара	ІТ-11 Огірко О.	ІТ6-11 Мідик А.	ІТ-21 Дяков А.	ІТ6-21 Мусійовська М.	ІТ-31 Галайко Н.	ІТ6-31 Зачек О.	ІТ-41 Магеровська Т.
1	Математика	Перерва	Історія		Фізика		Математика
2	Програмування	Математика		Аналітика		Біологія	
3	Бази даних	Фізика	Математика		Історія		Програмування
4	Англійська		Програмування	Історія		Математика	
5	Історія	Англійська			Фізика		Перерва

Копіювати розклад у MS Excel

Рис. 3.10 Інтерфейс авторизованого користувача

Користувач із правами адміністратора отримує доступ до розділу налаштувань, де здійснюється керування довідковими даними, необхідними для формування навчального розкладу.

Для початку роботи із системою необхідно заповнити базу даних інформацією про структурні підрозділи навчального закладу. Додавання нового відділення виконується через вкладку «Відділення» натисканням кнопки додавання. У відкритій формі (рис. 3.11) користувач вводить назву нового відділення та зберігає зміни.

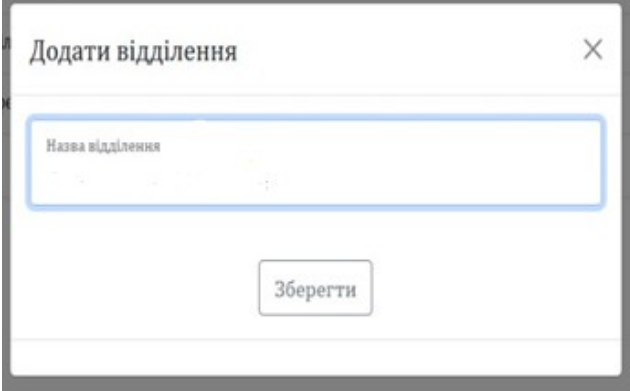
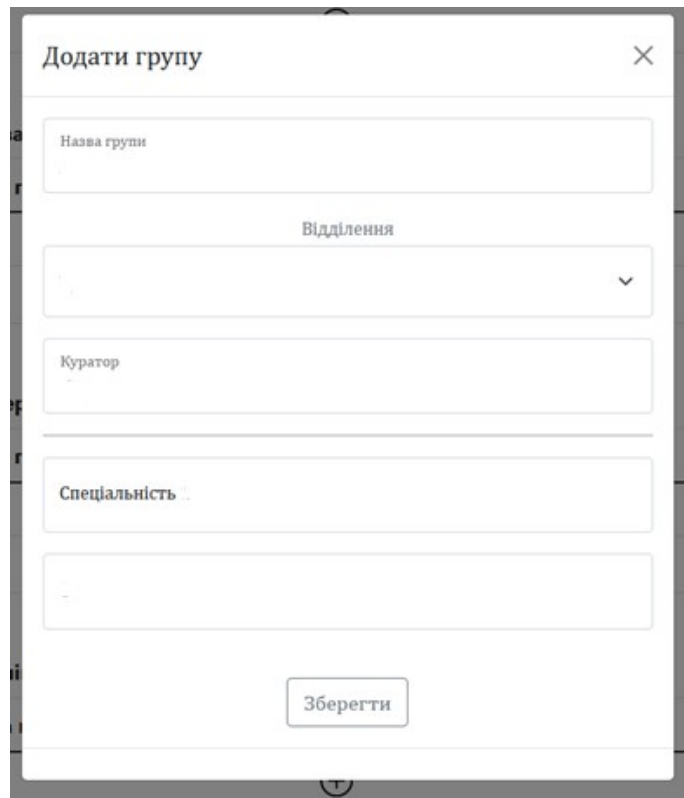
The image shows a screenshot of a web application window titled "Додати відділення" (Add department). The window has a close button (X) in the top right corner. Below the title bar is a text input field with the placeholder text "Назва відділення" (Department name). At the bottom of the form is a button labeled "Зберегти" (Save).

Рис. 3.11 Додавання відділення

Окрім створення нових записів, було протестовано можливість зміни та видалення відділень. Під час внесення п'яти тестових записів жодних помилок або некоректної поведінки системи не зафіксовано.

Після цього перевірено роботу модуля управління навчальними групами. Для створення групи користувач переходить до відповідного розділу та відкриває форму додавання (рис. 3.12). У формі вказується назва групи, обирається відділення, за необхідності призначається куратор та додаються спеціальності, для збереження результату натиснути кнопку Зберегти.



The image shows a mobile application interface for adding a new group. The form is titled "Додати групу" (Add group) and includes the following fields:

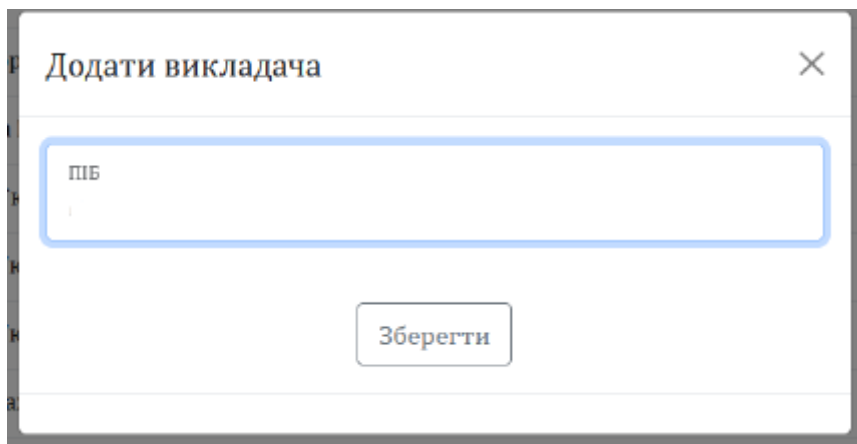
- Назва групи (Group name)
- Відділення (Department) - a dropdown menu
- Куратор (Curator)
- Спеціальність (Specialty)
- An empty text input field
- Зберегти (Save) button

Рис. 3.12 Створення нової групи

Функції редагування та видалення груп теж були перевірені та показали стабільну роботу. Зокрема, під час зміни відділення група автоматично переносилася до відповідного розділу без втрати пов'язаних даних. Відсутність інформації про куратора або спеціальності не впливала на коректність функціонування модуля.

Під час тестування було введено всі групи навчального закладу, помилок в роботі не виникало.

Для додавання викладачів використовується окрема форма (рис. 3.13), де потрібно вказати прізвище, ім'я та по батькові викладача. Після підтвердження введені дані зберігаються в базі даних.



Додати викладача

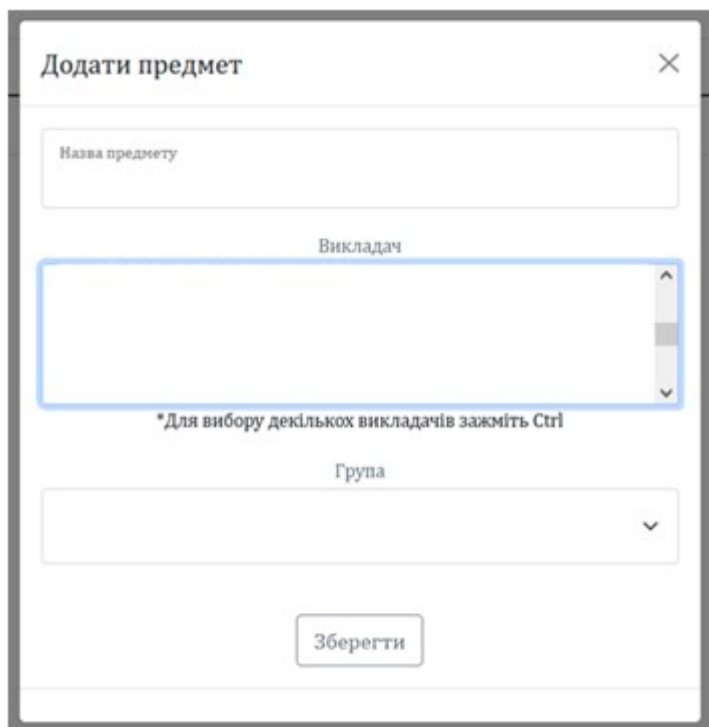
пб

Зберегти

Рис. 3.13 Додавання викладача

Аналогічно працює форма редагування викладачів, яка додатково містить інструменти для збереження змін або видалення запису. При перевірці не виявлено жодних проблем.

Для внесення нових дисциплін використовується форма, наведена на рис. 3.14. Вона відкривається після вибору відповідної групи та активації команди додавання.



Додати предмет

Назва предмету

Викладач

*Для вибору декількох викладачів зажмите Ctrl

Група

Зберегти

Рис. 3.14 Створення нового предмета

Під час створення предмета потрібно вказати його назву, обрати одного

або декількох викладачів зі списку та визначити навчальну групу. Після підтвердження введених даних запис зберігається в системі. Тестування цього модуля підтвердило його коректну роботу.

Під час тестування було введено предмети, помилок в роботі не виявлено.

Крім функціонального тестування виконано перевірку системи за такими критеріями на безпеку, коректність роботи та на швидкодію

Можемо оцінити роботу системи на відмінно. Це обгрунтовується тим, що безпека системи працює безвідмовно – кожен із запитів повертає статус-код 403 «Заборонено» у випадку, коли користувач намагається виконати запит до системи без автентифікації.

Щодо швидкодії, то кожен із запитів виконується в межах двох секунд, що є досить високим показником для такої немалої системи. Перевірка проводилась двома залогованими користувачами.

Якщо говорити про коректність даних, які віддаються системою клієнту, то це є одним із найвищих показників якості системи. Тому що це і є основною функцією системи. Також цей показник є вагомим, тому що, дані, отримані від сервісу результат роботи практично всіх форм системи. Тобто це свідчить про те, що база даних працює коректно, підключення та роботи з базою даних також працює коректно, форми, які проводять створення об'єктів передачі даних також працюють коректно, і передача даних клієнту також виконується коректно.

На основі проведених тестувань можна зробити такі висновки:

- помилок у відображенні інформації не виявлено;
- усі нові записи успішно зберігаються в базі даних;
- зміна існуючих даних виконується без втрати інформації та виникнення помилок;
- інтерфейс коректно відображається на різних пристроях і типах екранів.

3.5. Перспективи модернізації

Розроблена інформаційна система автоматизації формування розкладу занять виконує поставлені завдання та має потенціал для вдосконалення та розширення можливостей.. Додаткові функції дозволять автоматизувати процеси та інтегруватися з іншими програмами.

Серед перспективних напрямків:

- автоматичний розрахунок навантаження викладачів з онлайн-доступом для кожного педагога;
- автоматична перевірка розкладу на відповідність навчальному плану групи;
- додавання зручних фільтрів на головній сторінці для швидкого перегляду розкладу окремої групи чи викладача.

Незважаючи на успішну реалізацію основного функціоналу, дана версія системи може розглядатися як прототип. Подальшого вдосконалення потребує дизайн графічного інтерфейсу, що дозволить підвищити зручність взаємодії користувачів із системою та покращити її візуальне сприйняття.

Усі зазначені напрями розвитку доцільно врахувати під час наступних етапів модернізації програмного комплексу.

ВИСНОВКИ

У ході виконання кваліфікаційної бакалаврської роботи було створено веб-орієнтовану інформаційну систему, призначену для автоматизації процесу формування навчальних розкладів у закладах освіти.

Основною метою розробки було спрощення складного процесу складання розкладу шляхом автоматизації обробки великої кількості взаємопов'язаних даних та усунення потенційних конфліктів і помилок. Для досягнення поставленої мети в системі реалізовано механізми управління користувачами, ведення довідників навчальних груп, викладачів, дисциплін та аудиторій, налаштування параметрів формування розкладу, його автоматизованого генерування, редагування та перегляду.

Під час виконання роботи було отримано такі результати:

- досліджено предметну область та особливості організації навчального процесу;
- проведено аналіз існуючих програмних рішень для автоматизації складання розкладів;
- виконано системний аналіз і сформовано вимоги до майбутньої системи;
- розроблено UML-моделі для опису структури та логіки функціонування програмного продукту;
- спроектовано та реалізовано базу даних для зберігання інформації про групи, викладачів, дисципліни та аудиторії;
- створено користувацький та адміністративний інтерфейси;
- реалізовано веб-застосунок відповідно до сформованих вимог;
- проведено тестування функціональних можливостей системи.

Розроблене програмне забезпечення орієнтоване насамперед на працівників навчальних закладів, відповідальних за планування навчального процесу. Незважаючи на відносно вузьке коло користувачів, система здатна

суттєво підвищити ефективність їхньої роботи та зменшити втрати часу на складання розкладу.

Також програмний продукт має значні перспективи розвитку. Розширення функціональних можливостей, підвищення рівня безпеки та вдосконалення інтерфейсу сприятимуть зростанню його практичної цінності та рівня задоволеності користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Положення про організацію освітнього процесу у Львівському державному університеті внутрішніх від 28.08.2025 № 399. URL: <https://www.lvduvs.edu.ua/uk/lvsuia/legal-framework/provisions-orders.html>
2. FET – Free Timetabling Software. URL: <https://www.timetabling.com>
3. Asc Timetables. URL: <https://www.asctimetables.com>
4. UniTime – Comprehensive Academic Scheduling Solution. URL: <https://unitime.org>
5. МКР «Автоматизована система управління навчальним закладом». URL: <https://mkr.org.ua/>.
6. Ситник Г. П. Основи теорії систем та системного аналізу : навчальний посібник. Київ : КНУ імені Тараса Шевченка, 2024. 304 с.
7. phpMyAdmin. URL: <https://www.phpmyadmin.net/>
8. MySQL Tutorial - Learn MySQL Fast, Easy and Fun. MySQL Tutorial. URL: <https://www.mysqltutorial.org/>.
9. Васильєв О. Програмування мовою PHP : навчальний посібник. Ліра-К, 2022. 368 с.
10. Жуков І.М. Інформаційні технології в освіті: підручник. Львів: Видавництво ЛНУ, 2021. 356 с.
11. Попов В.В. Алгоритми та структури даних. Київ: Наукова думка, 2018. 276 с.
12. MySQL Tutorial - Learn MySQL Fast, Easy and Fun. MySQL Tutorial. URL: <https://www.mysqltutorial.org/>.
13. Довідник по CSS властивостям. Український веб-довідник. URL: <https://css.in.ua/css/properties>.
14. Columns. Bootstrap The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/docs/5.3/layout/columns/>.
15. Роббинс Дж. HTML5. Кишеньковий довідник / trans. by А. Сергеев

5th ed. Науковий світ, 2022. 190 p.

16. Forms. Bootstrap The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/docs/5.3/forms/overview/>.
17. Kumar R. Complete HTML Practical Course: HTML. Independently Published, 2020. 180 p.
18. Grippa V. M., Kuzmichev S. Learning MySQL. O'Reilly Media, Incorporated, 2021. 692 p.
19. MySQL Tutorial - Learn MySQL Fast, Easy and Fun. MySQL Tutorial. URL: <https://www.mysqltutorial.org/>.
20. Butler T. PHP and mysql: novice to ninja. SitePoint Pty, Limited, 2021. 680 p.
21. PHP: PHP Manual - Manual. PHP: Hypertext Preprocessor. URL: <https://www.php.net/manual/en/>.
22. PHP Tutorial. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com/php/>.
23. Шаховська Н. Б., Литвин В. В. Проектування інформаційних систем: навч. посібник. Львів: "Маг-нолія-2006". 2022. 380 с.

ДОДАТКИ

Додаток А

Програмний код таблиці розкладу

```

<table class="table-bordered" style="font-family: Cambria;
font-size: 14pt; width: <?php echo
69+count($array_groups)*180; ?>px;">
  <tr>
    <td colspan="2" rowspan="3"></td>
    <?php for ($i = 0; $i < count($array_groups); $i++) { ?>
      <td colspan="4" style="font-size: 22pt; background-color:
#C0C0C0; text-align: center; font-weight: bold;
width: 180px;"><?php echo $array_groups[$i]['name']
?></td>
      <?php } ?>
    </tr>
    <tr>
      <?php for ($i = 0; $i < count($array_groups); $i++) { ?>
        <td colspan="4" style="font-style: italic; text-align:
center; width: 180px;" ><?php echo $array_groups[$i]
['kurator']
?></td>
        <?php } ?>
      </tr>
    <tr>
      <?php for ($i = 0; $i < count($array_groups); $i++)
      { if (!empty($result['spec1'])) { ?>
        <td colspan="4" style="font-style: italic; text-align:
center; width: 180px;" ><?php echo $array_groups[$i]['spec1']
?></td>
        <?php } else { ?>
          <td colspan="2" class=" td-2" style="font-style:
italic; text-align: center; width: 90px; height:
22px;"><?php echo
$array_groups[$i]['spec1'] ?></td>
          <td colspan="2" class=" td-2" style="font-style:
italic; text-align: center; width: 90px; height:
22px;"><?php echo
$array_groups[$i]['spec2'] ?></td>
          <?php }
        } ?>
      </tr>
    <?php
    for ($i = 0; $i < count($array_day); $i++)
    { for ($j = 1; $j < 8; $j++) { if(!
empty($_POST['teacher']))){
      $sql="SELECT `groups`.`id` AS
`group`, `groups`.`id_department`, `schedule`.* FROM
`schedule` INNER JOIN `subjects` ON
`schedule`.`id_subject`=`subjects`.`id` INNER JOIN `groups` ON
`subjects`.`id_group`=`groups`.`id` WHERE
`schedule`.`day`='". $i.'" AND `schedule`.`n_lesson`='". $j.'"

```

```

AND ( ";
    $sql=$sql."`schedule`.`id_subject`='". $array_s_id[0]."' ";
    for($idd=1;$idd<count($array_s_id);$idd++){
        $sql=$sql."OR `schedule`.`id_subject`='".
            $array_s_id[$idd]."'
    ";
    }
    $sql=$sql.") ORDER BY `groups`.`id_department` ASC;";
    } else {
        $sql="SELECT `groups`.`id` AS `group`,
`groups`.`id_department`, `schedule`.* FROM `schedule` INNER
JOIN
`subjects` ON `schedule`.`id_subject`=`subjects`.`id` INNER JOIN
`groups` ON `subjects`.`id_group`=`groups`.`id` WHERE
`schedule`.`day`='". $i."' AND `schedule`.`n_lesson`='". $j."'
ORDER BY `groups`.`id_department` ASC;";
    }
    $res=mysqli_query($connect,$sql);
    $array_1=[];
    while ($result = mysqli_fetch_array($res)) {
        $array_1[]=$result;
    }?>
    <tr>
        <?php if($j==1){ ?><td rowspan="28" style="font-size:
22pt; font-weight: bold; width: 44px; text-align: center;"><?
php echo
$array_day[$i][0]; ?></td><?php } ?>
        <td rowspan="4" style="font-weight: bold; width: 25px;
text-align: center;"><?php echo $j; ?></td>
        <?php
        for ($g = 0; $g < count($array_groups); $g++) {
            $n=0;
            for ($ar = 0; $ar < count($array_1); $ar++)
            { if($array_groups[$g]['id']==$array_1[$ar]['group'])
            { if($array_1[$ar]['id_type_lesson']=='1'){
                echo "<td colspan='4' style='height: 23px; border-bottom:
hidden; text-align: right; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]
['id_cl assroom1'], $array_cl)."</td>";
                $n=1;
            }
            if($array_1[$ar]['id_type_lesson']=='2'){
                echo "<td colspan='2' style='height: 23px; border-right:
hidden; border-bottom: hidden; text-align: left; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]
['id_cl assroom1'], $array_cl)."</td><td colspan='2'
style='height: 23px; border-bottom: hidden; text-align: right;
font-style: italic;'>". $array_1[$ar]['name_teacher']."</td>";
                $n=1;
            }
            if($array_1[$ar]['id_type_lesson']=='3'){echo "";$n=1;}
            if($array_1[$ar]['id_type_lesson']=='4'){
                echo "<td colspan='2' style='height: 23px; border-bottom:

```

```

hidden; text-align: right; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]
['id_cl assroom1'], $array_cl)."</td>";
    $n=1;
    }
    if($array_1[$ar]['id_type_lesson']=='5'){
    echo "<td colspan='2' style='height: 23px; border-bottom:
hidden; text-align: right; font-style:
italic;'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]
['id_cl assroom1'], $array_cl)."</td>";
    $n=1;
    }
    if($array_1[$ar]['id_type_lesson']=='6'){
    echo "<td style='height: 23px; border-right: hidden;
border- bottom: hidden; text-align: left; font-style: italic;
'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]
['id_classroom 1'], $array_cl)."</td><td style='height: 23px;
border-bottom: hidden; text-align: right; font-style:
italic;'>".$array_1[$ar]['name_teacher']."</td>";
    $n=1;
    }
    if($array_1[$ar]['id_type_lesson']=='7'){echo ""; $n=1;}
    if($array_1[$ar]['id_type_lesson']=='8'){
    echo "<td style='height: 23px; border-right: hidden;
border- bottom: hidden; text-align: left; font-style: italic;
'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]
['id_classroom 1'], $array_cl)."</td><td style='height: 23px;
border-bottom: hidden; text-align: right; font-style:
italic;'>".$array_1[$ar]['name_teacher']."</td>";
    $n=1;
    }
    if($array_1[$ar]['id_type_lesson']=='9'){echo ""; $n=1;}
    }
    }
    if($n==0){
    ?>
    <td colspan="4" rowspan="4" style="height: 94px; text-
align: center; font-weight: bold; vertical-align: middle;"></td>
    <?php } } ?>
    </tr>
    <tr>
    <?php
    for ($g = 0; $g < count($array_groups); $g++) {
    for ($ar = 0; $ar < count($array_1); $ar++) {
    if($array_groups[$g]['id']==$array_1[$ar]['group'])
    { if($array_1[$ar]['id_type_lesson']=='1'){
    echo "<td colspan='4' rowspan='2' style='height: 48px;
border-bottom: hidden; text-align: center; font-weight: bold;
vertical-align: middle;'>".$array_1[$ar]['name_subj']."</td>";
    }
    if($array_1[$ar]['id_type_lesson']=='2'){
    echo "<td colspan='4' style='height: 24px; text-align:
center; font-weight: bold; vertical-align: middle;'>".

```



```

        if($array_1[$ar]['id_type_lesson']=='9'){
            echo "<td colspan='2' style='height: 24px; border-bottom:
hidden; text-align: center; font-weight: bold; vertical-align:
middle;'>".$array_1[$ar]['name_subj']."</td>";
        }
    }
}
}
?>
</tr>
<tr>
<?php
for ($g = 0; $g < count($array_groups); $g+
+) { for ($ar = 0; $ar < count($array_1);
$ar++) {
    if($array_groups[$g]['id']==$array_1[$ar]
['group']){ if($array_1[$ar]
['id_type_lesson']=='1'){
        echo "<td colspan='4' style='height: 23px; text-align:
right; font-style: italic;'>".$array_1[$ar]
['name_teacher']."</td>";
    }
    if($array_1[$ar]['id_type_lesson']=='2'){echo "";}
    if($array_1[$ar]['id_type_lesson']=='3'){
        echo "<td colspan='2' style='height: 23px; border-right:
hidden; text-align: left; font-style: italic;
'>".clas($array_1[$ar]['id_classroom'],$array_1[$ar]
['id_classroom 1'], $array_cl)."</td><td colspan='2'
style='height: 23px; text-align: right; font-style:
italic;'>".$array_1[$ar]['name_teacher']."</td>";
    }
    if($array_1[$ar]['id_type_lesson']=='4'){
        echo "<td colspan='2' style='height: 23px; text-align:
right; font-style: italic; '>".$array_1[$ar]
['name_teacher']."</td>";
    }
    if($array_1[$ar]['id_type_lesson']=='5'){
        echo "<td colspan='2' style='height: 23px; text-align:
right; font-style: italic; '>".$array_1[$ar]
['name_teacher']."</td>";
    }
    if($array_1[$ar]['id_type_lesson']=='6'){echo "";}
    if($array_1[$ar]['id_type_lesson']=='7'){
        echo "<td style='height: 23px; border-right: hidden; text-
align: left; font-style: italic; '>".clas($array_1[$ar]
['id_classroom'],$array_1[$ar]['id_classroom 1'],
$array_cl)."</td><td style='height: 23px; text-align: right;
font-style: italic;'>".$array_1[$ar]['name_teacher']."</td>";
    }
    if($array_1[$ar]['id_type_lesson']=='8'){echo "";}
    if($array_1[$ar]['id_type_lesson']=='9'){
        echo "<td style='height: 23px; border-right: hidden; text-
align: left; font-style: italic; '>".clas($array_1[$ar]

```

```
['id_classroom'],$array_1[$ar]['id_classroom 1'],
$array_cl)."</td><td style='height: 23px; text-align: right;
font-style: italic;'>".$array_1[$ar]['name_teacher']."</td>";
    }
  }
}
}
?>
</tr>
<?php } }
</table>
```

Додаток Б

Блок форми додавання нової пари

```

echo '<div class="row"><div class="col-sm-11"><select
name="subj'.'.$name.'" id="subj'.'.$name.'"
onchange="updateInputValue'.'.$name.'" class="form-select">';
    $sql123="SELECT * FROM `subjects` WHERE `name`=' ' AND
`id_group`='".$array_groups['id']."'";
    $res123=mysqli_query($connect,$sql123);
    $result123 = mysqli_fetch_array($res123);
    echo "<option value='".$result123['id']."'>-</option>";
    $sqlq="SELECT `schedule`.`id`, `subjects`.`id_teacher`
FROM
`schedule` INNER JOIN `subjects` ON
`schedule`.`id_subject`=`subjects`.`id` WHERE
`schedule`.`day`='".$i."' AND `schedule`.`n_lesson`='".$j."'
AND
`subjects`.`id_teacher`!=1' AND `subjects`.`name`!= ' ' AND
`schedule`.`zvedena`='0' UNION SELECT `schedule`.`id`,
`subjects`.`teachers` FROM `schedule` INNER JOIN `subjects` ON
`schedule`.`id_subject`=`subjects`.`id` WHERE
`schedule`.`day`='".$i."' AND `schedule`.`n_lesson`='".$j."'
AND
`subjects`.`teachers`!=1' AND `subjects`.`name`!= ' ' ";
    $resq=mysqli_query($connect,$sqlq);
    $teach="";
    while ($resultq = mysqli_fetch_array($resq)) {
        $teach=$teach." AND
`id_teacher`!= '".$resultq['id_teacher']."' AND
`teachers`!= '".$resultq['id_teacher']."' ";
    }
    $sql="SELECT * FROM `subjects` WHERE `name`!= ' ' AND
`id_group`='".$array_groups['id']."'".$teach." ORDER BY `name`
ASC";
    $res=mysqli_query($connect,$sql);
    while ($result = mysqli_fetch_array($res)) {
        echo "<option value='".$result['id']."'>".
$result['name']." (.pib($result['id_teacher'],
$result['teachers'],
$array_teachers_f).)". "</option>";
    }
    $sql2="SELECT `id_department` FROM `groups` WHERE
`id`='".$g."'";
    $res2=mysqli_query($connect,$sql2);
    $v = mysqli_fetch_array($res2);
    echo '</select></div><div class="col-sm-1"><a href=" php?
v='.$v[0].'#g'.'.$g.'" target="_blank"><svg xmlns=" svg"
width="19" height="19" fill="currentColor" class="bi bi-plus-
circle" viewBox="0 0 16 16"><path d="M8 15A7 7 0 1 1 8 1a7 7 0
0 1 0 14zm0
1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"/><path d="M8 4a.5.5 0 0 1
.5.5v3h3a.5.5 0 0 1 0 1h-3v3a.5.5 0 0 1-1 0v-3h-3a.5.5 0 0 1 0-
1h3v-3A.5.5 0 0 1 8 4z"/></svg></a></div></div><br>';

```

```
echo '<script>function updateInputValue'.$name.'() { var  
select = document.getElementById("subj'.$name.'"); var input =
```

```

document.getElementById("name_subj".$name.""); var input1 =
document.getElementById("name_teach".$name."");
switch(select.value){';
    $sql="SELECT * FROM `subjects` WHERE `name`!=' ' AND
`id_group`='". $array_groups['id']. "'

$res=mysqli_query($connect,$sql);
    while ($result = mysqli_fetch_array($res)) {
        echo " case '". $result['id']. "' :
input.value='". $result['name']. "' ;
input1.value='". pib($result['id_teacher'], $result['teachers'],
$array_teachers_f). "' ; break;";
    }
    echo 'default: input.value=" "; input1.value=" "; break; }
}</script>;
    echo "<label class='form-label'>Відображається в
розкладі:</label><input name='name_subj".$name.'"
id='name_subj".$name.'" value=' ' type='text'
class='form-control'><br><input name='name_teach".
$name.'" id='name_teach".$name.'" value=' '
type='text' class='form-control'><br>";
    echo '<select name="room".$name.'" multiple class="form-
select" style="height: 20%;">';
    for ($z = 0; $z < count($array_classroom); $z++)
    { if($array_classroom[$z]['id']=='0'){
        echo "<option selected value=' " . $array_classroom[$z]
['id']
. "'>" . $array_classroom[$z]['name'] . "</option>";
    } else {
        echo "<option value=' " . $array_classroom[$z]['id'] . "'>".
$array_classroom[$z]['name'] . " (".$array_classroom[$z]
['nameK'].")". "</option>";
    }
}
}
}
echo '</select><p>*Для вибору декількох кабінетів зажміть
Ctrl</p><input name="zvedena".$name.'" type="checkbox"
value="1">Зведена пара<br><br>';

```

Додаток В

Код сторінки для редагування груп

```

<div class="row">
  <div class="col-sm-1"></div>
  <div class="col-sm-2">
    <ul class="nav flex-column" style="margin-top: 15%;">
      <?php for ($i = 0; $i < count($array_department);$i++)
        { echo "<li class='nav-item'><a class='nav-link'
href='#v".$array_department[$i][0]."' style='color: #7e868d;
margin-bottom: 5%;'>".$array_department[$i][1]."</a></li>";
        } ?>
      </ul>
    </div>
    <div class="col-sm-6">
      <?php for ($i = 0; $i < count($array_department);$i++){ ?>
      <table class="table">
        <thead>
          <tr><th colspan="5" id="v<?php echo $array_department[$i]
[0];
?>"><?php echo $array_department[$i][1]; ?></th></tr>
          <tr>
            <th scope="col">№</th>
            <th scope="col">Назва групи</th>
            <th scope="col">Куратор</th>
            <th scope="col">Спеціальності</th>
            <th scope="col"></th>
          </tr>
        </thead>
        <tbody>
          <?php
          $sql="SELECT `groups`.*, `department`.`name` AS `d_name`
FROM
`groups` INNER JOIN `department` ON
`groups`.`id_department`=`department`.`id` WHERE
`groups`.`id_department`='".$array_department[$i][0]."'";
          $res=mysqli_query($connect,$sql);
          $n=1; while($result=mysqli_fetch_array($res)){
            <tr>
              <th scope="row"><?php echo $n; ?></th>
              <td><?php echo $result['name'] ?></td>
              <td><?php echo $result['kurator'] ?></td>
              <td><?php if (!empty($result['spec1'])) {
echo $result['spec1'] . " | " . $result['spec2'];
} ?></td>
              <td data-bs-toggle="modal" data-bs-
target="#staticBackdrop<?php echo $result['id'] ?>">
                <svg xmlns=" svg" width="16" height="16"
fill="currentColor" class="bi bi-pencil " viewBox="0 0 16 16">
                  <path d="M12.146 14.6a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1 0

```

```

.708l-10 10a.5.5 0 0 1-.168.11l-5 2a.5.5 0 0 1-.65-.65l2-5a.5.5
0
0 1 .11-.168l10-10zM11.207 2.5 13.5 4.793 14.793 3.5 12.5 1.207
11.207 2.5zm1.586 3L10.5 3.207 4 9.707V10h.5a.5.5 0 0 1
.5.5v.5h.5a.5.5 0 0 1 .5.5v.5h.293l6.5-6.5zm-9.761 5.175-
.106.106-
1.528 3.821 3.821-1.528.106-.106A.5.5 0 0 1 5 12.5V12h-.5a.5.5 0
0
1-.5-.5V11h-.5a.5.5 0 0 1-.468-.325z"/>
</svg>
</td>
</tr>
<div class="modal fade" id="staticBackdrop<?php echo
$result['id'] ?>" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="staticBackdropLabel">
Педагогати</h5>
<button type="button" class="btn-close text-center" data-
bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<form method="POST" action="settings_group.php">
<div class="form-floating mb-3">
<input type="hidden" name="id" value="<?php echo
$result['id'] ?>">
<input type="text" name="name" value="<?php echo
$result['name'] ?>" class="form-control" id="floatingInput"
placeholder="Назва групи">
<label for="floatingInput">Назва групи</label>
</div>
<div class="mb-3">
<label class="text-muted text-start" style="margin-bottom:
3px;">Відділення</label>
<select class="form-select" style="height: 55px;"
name="vid">
<?php
for ($j= 0; $j < count($array_department);$j++){ echo
"<option value='".$array_department[$j][0]."' ";
if($array_department[$j][0]==$result['id_department'])
{ echo "selected";
}
echo ">".$array_department[$j][1]."</option>";
}
?>
</select>
</div>

```

```

        <div class="form-floating mb-3">
            <input type="text" name="kurator" value="<?php echo
$result['kurator'] ?>" class="form-control" id="floatingInput"
placeholder="Куратор">
            <label for="floatingInput">Куратор</label>
        </div>
        <hr>
        <div class="form-floating mb-3">
            <input type="text" name="spec1" value="<?php echo
$result['spec1'] ?>" class="form-control" id="floatingInput"
placeholder="Спеціальність 1">
            <label for="floatingInput">Спеціальність 1</label>
        </div>
        <div class="form-floating mb-3">
            <input type="text" name="spec2" value="<?php echo
$result['spec2'] ?>" class="form-control" id="floatingInput"
placeholder="Спеціальність 2">
            <label for="floatingInput">Спеціальність 2</label>
        </div>
        <br>
        <div class="form-floating mb-3 text-center">
            <button type="submit" name="save" class="btn btn-outline-
secondary" style="margin-right: 40%;">Зберегти</button>
            <button type="submit" name="delete" class="btn btn-outline-
secondary">Видалити</button>
        </div>
    </form>
</div>
<div class="modal-footer">
</div>
</div>
</div>
</div>
<?php $n++;} ?>
<tr>
<td colspan="5" style="text-align: center;">
<button type="button" data-bs-toggle="modal" data-bs-
target="#staticBackdropNew<?php echo
$array_department[$i][0]; ?>">
    <svg xmlns="svg" width="25" height="25"
fill="currentColor" class="bi bi-plus-circle" viewBox="0 0 16
16">
        <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8
0a8 8 0 0 0 0 16z"/>
        <path d="M8 4a.5.5 0 0 1 .5.5v3h3a.5.5 0 0 1 0 1h-3v3a.5.5 0
0 1-1 0v-3h-3a.5.5 0 0 1 0-1h3v-3A.5.5 0 0 1 8 4z"/>
    </svg>
</button>
    <div class="modal fade" id="staticBackdropNew<?php echo
$array_department[$i][0]; ?>" data-bs-backdrop="static" data-
bs-keyboard="false" tabindex="-1" aria-

```

```

labelledby="staticBackdropLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">Додати
групу</h5>
        <button type="button" class="btn-close text-center" data-
bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form method="POST" action="settings_group.php">
          <div class="form-floating mb-3">
            <input type="text" name="name" class="form-control"
id="floatingInput" placeholder="Назва групи">
            <label for="floatingInput">Назва групи</label>
          </div>
          <div class="mb-3">
            <label class="text-muted text-start" style="margin-bottom:
3px;">Відділення</label>
            <select class="form-select" style="height: 55px;"
name="vid">
              <?php
for ($q = 0; $q < count($array_department);$q++){ echo
"<option value='".$array_department[$q][0]."' ";
if($array_department[$q][0]==$array_department[$i][0])
{ echo "selected";
}
echo ">".$array_department[$q][1]."</option>";
} ?>
            </select>
          </div>
          <div class="form-floating mb-3">
            <input type="text" name="kurator" class="form-control"
id="floatingInput" placeholder="Куратор">
            <label for="floatingInput">Куратор</label>
          </div>
          <hr><div class="form-floating mb-3">
            <input type="text" name="spec1" class="form-control"
id="floatingInput" placeholder="Спеціальність 1">
            <label for="floatingInput">Спеціальність 1</label>
          </div>
          <div class="form-floating mb-3">
            <input type="text" name="spec2" class="form-control"
id="floatingInput" placeholder="Спеціальність 2">
            <label for="floatingInput">Спеціальність 2</label>
          </div>
          <br><div class="form-floating text-center">
            <button type="submit" name="new" class="btn btn-outline-
secondary">Зберегти</button>
          </div>
        </form>
      </div>
    </div>
  </div>

```

```
</div>
<div class="modal-footer"></div>
</div>
</div>
</td>
</tr>
</tbody>
</table>
<br><?php } ?>
</div>
<div class="col-sm-1"></div>
</div>
```