

5.6 Автоматизація розкладу занять як необхідна умова оптимізації функціонування закладу освіти

Об'єктивний розвиток інформатизації суспільства неминуче сприяє впровадженню нових форм управління навчальним процесом у закладах освіти, зокрема у формуванні розкладу занять. Практика складання розкладу вручну значною мірою відійшла у минуле для більшості освітніх закладів. Очевидно, що традиційне ручне формування розкладу є трудомістким процесом, схильним до людських помилок і конфліктів. З огляду на це, автоматизація процесу складання розкладу є критично важливою для оптимізації функціонування закладів освіти. Автоматизація розкладу у освітніх закладах має безліч переваг, які спрямовані на оптимізацію навчального процесу, зменшення адміністративного навантаження та покращення досвіду всіх учасників освітнього процесу (рис.1).



Рисунок 1. Переваги автоматизації розкладу занять в закладах освіти

Процес формування розкладу навчального процесу у закладах освіти є багатофакторною оптимізаційною задачею [283]. Вона передбачає ефективне розміщення різних академічних сесій у визначений інтервал часу та з одночасним задоволенням великої кількості жорстких та м'яких обмежень. Ці обмеження охоплюють різноманітні напрями, зокрема, кваліфікацію викладачів, доступність аудиторій, вимоги навчальних програм, а також побажання учасників освітнього процесу та інші аспекти. Тому складання розкладу характеризується високою обчислювальною складністю.

Вибір адекватної математичної моделі та надійного алгоритму є важливим етапом для розробки автоматизованої системи складання розкладу. За допомогою математичної моделі здійснюється формалізація усіх суттєвих параметрів формування розкладу, а саме – це викладачі, студенти, аудиторії, дисципліни, їхні властивості та взаємозв'язки між ними. Також потрібно визначити цільову функцію та систему обмежень. Розробка алгоритму, який відповідає цій математичній моделі, є ключовим елементом, бо завдяки йому здійснюється пошук оптимального розв'язку.

Існує кілька технік або підходів для автоматизованого вирішення проблеми формування розкладу у закладах освіти. Розглянемо їх більш детально.

1. Метод імітації відпалу

Даний метод було запропоновано у сфері фізичних процесів кристалізації речовини для покращення однорідності металу. Відпал металу – це імітація пошуку оптимального стану кристалічної решітки з мінімальною енергією. Під час нагрівання металу атоми змінюють своє положення, а після повільного охолодження вибирають енергетично вигідніші позиції. Разом з температурою зменшується ймовірність переходу в гірший стан – низькоенергетичний, який є необхідною умовою для уникнення ув'язнення у локальному мінімумі. При цьому потрібно періодично підвищувати температуру. Таким чином знайти той стан кристалічної решітки, що робить метал міцнішим.

Метод імітації відпалу можливо застосувати для розв'язання задачі складання розкладу в закладах освіти [283-286]. При формуванні розкладу

функцією енергії є критеріальна функція, пов'язана із штрафами при конфліктній ситуації в розкладі, а низькоенергетичний стан – коректний розклад [283]. Блок-схема алгоритму імітації відпалу представлено на рис.2.

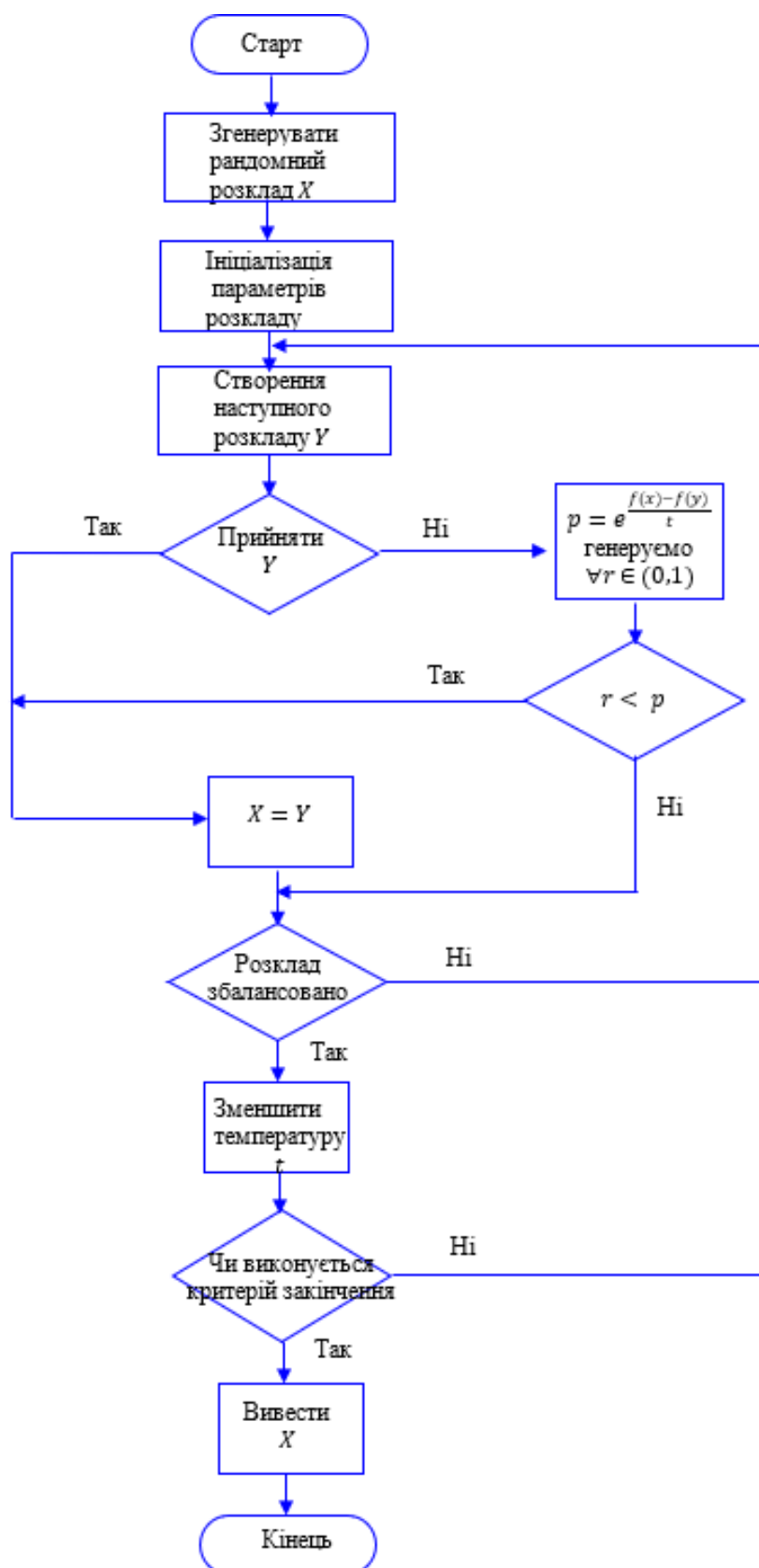


Рисунок 2. Блок-схема моделювання розкладу занять методом відпалу

2. Метод розфарбування графів

Метод розфарбування графів полягає у знаходженні мінімального числа кольорів для розфарбування вершин графа, де сусідні не можуть мати однаковий колір. У процесі формування розкладу в закладах освіти автори досліджень [287-289] пропонують будувати граф, у якого кожна вершина – це навчальне заняття, колір – це часовий слот, ребро – спільні студенти, тобто два заняття (або один викладач) в одній аудиторії, що призводить до конфліктної ситуації при складанні розкладу. Алгоритм має високу складність, тому на практиці найчастіше його застосовують разом з іншими методами.

3. Імітаційне моделювання

За допомогою даного методу здійснюється імітація роботи адміністратора, який створює розклад і намагається задовольнити задані обмеження найкращим чином [283,290]. На першому кроці створюється порожній розклад і задаються усі параметри, да другому – вибирається заняття (за різною логікою) із списку і включають його у розклад, тобто вибирається час і аудиторія, враховуючи усі обмеження. Процес продовжується доти, доки не буде сформовано повний розклад. При цьому, якщо виникають конфліктні ситуації, то дане заняття видаляють з розкладу і вносять у список неврахованих. Значний обсяг ітерацій у даному методі є непривабливим для застосування.

4. Метод програмування з обмеженнями

Формування розкладу в закладах освіти з використанням методу програмування з обмеженням здійснюється за допомогою спеціального програмного забезпечення, у якому декларативно описується задача із змінними: заняття X проводиться у час Y в аудиторії Z , та визначаються обмеження, які можуть вести до конфліктних ситуацій [291-293]. Система автоматично шукає рішення, які задовольняють усім обмеженням. При цьому скорочується процес пошуку не завдяки тестуванню кожного створеного розкладу, а система сама відсіює внаслідок відповідних прописаних правил. Метод програмування з обмеженнями часто застосовують при створенні розкладу для реальних закладів освіти.

5. Алгоритм мурашиної колонії

Даний метод застосовують у різних сферах для вирішення задач оптимізації, зокрема і у формуванні розкладу. Метод мурашиної колонії є ітераційним ймовірнісним алгоритмом, який повністю наслідує поведінку живих мурах у процесі пошуку найкоротшого шляху на зважених графах. Мурахи, шукаючи їжу, рандомно рухаються по стежках. При цьому залишають феромон на даному шляху, значення якого залежить від якості та розміру їжі. Проте він також із часом може випаровуватися. Найсильніший феромон вказує на найкоротший шлях. За допомогою даного методу створюється велике число розкладів, з кожною ітерацією розклади стають ефективнішими, оскільки мурахи "навчаються" від попередніх [294-297].

6. Генетичні алгоритми

В основі генетичних алгоритмів лежить теорія еволюції з такими характеристиками, як спадковість, мінливість, природний вибір. Більшість алгоритмів оптимізації працюють над удосконаленням одного розкладу, генетичні алгоритми фокусуються на популяції рішень, представлених у вигляді хромосом [298-302]. Генетичний алгоритм – ітераційний метод, блок-схема якого представлена на рис.3.

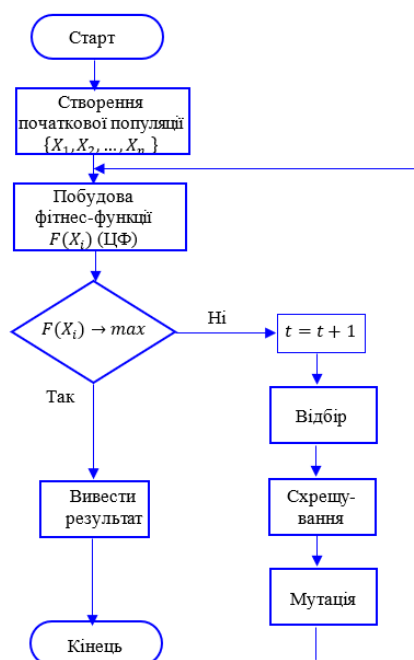


Рисунок 3. Блок-схема моделювання розкладу занять за допомогою генетичного алгоритму

Набір хромосом у формуванні розкладу – це навчальні заняття, які містять інформацію про час, викладачів, аудиторію. Для створення хромосом кодують інформацію за допомогою бінарного кодування. Використання цільової функції (у даному алгоритмі вона називається фітнес-функція) дозволяє перевірити, чи придатна дана хромосома для подальшого використання. У випадку негативного оцінювання хромосоми, її піддають процесам:

- схрещування (кожна хромосома є нащадком двох попередніх популяцій);
- мутації (зміна випадково відібраних генів).

Для формування розкладу схрещування застосовують рідко із-за громіздких обчислень. Після мутації за допомогою фітнес-функції знову оцінюється згенерована хромосома.

Генетичні алгоритми застосовують з іншими методами для більш ефективного створення розкладів.

Досліджені методи є потужним інструментом для формування розкладу в системах освіти. Проте кожний з них має свої переваги та недоліки, які представлені у таблиці 1.

Таблиця 1.
Порівняння методів формування розкладу в закладах освіти

№ з/п	Методи формування розкладу	Переваги	Недоліки
1	Метод імітації відпалу	1) Уникнення локальних мінімумів. 2) Універсальність. 3) Простота реалізації. 4) Ефективність для складних задач.	1) Витрати часу. 2) Залежність від параметрів. 3) Відсутність гарантії абсолютного мінімуму. 4) Ймовірнісний характер результатів.
2	Метод розфарбування графів	1) Візуалізація конфліктів, що дозволяє їх мінімізувати. 2) Простота у використанні і реалізації. 3) Зменшення ресурсів.	1) Складність обчислень (NP-повнота). 2) Статичний характер. 3) Слабка деталізація обмежень. 4) Недостатня увага на часових слотах.

Продовження таблиці 1

№ з/п	Методи формування розкладу	Переваги	Недоліки
3	Метод імітаційного моделювання	1) Більш реалістичний та гнучкий. 2) Можливість тестування різних сценаріїв. 3) Можливість виявлення переваг ресурсів	1) Трудомісткість і складність розробки. 2) Відсутність гарантії оптимальності. 3) Залежність від параметрів. 4) Ймовірнісний характер результатів.
4	Метод програмування з обмеженнями	1) Декларативність. 2) Ефективна обробка різноманітних обмежень. 3) Врахування м'яких обмежень. 4) Ефективний пошук рішень та раннє виявлення суперечностей. 5) Підтримка інкрементальних змін.	1) Проблеми масштабованості для об'ємних завдань. 2) Вимагає спеціалізованого програмного забезпечення. 3) Відсутність адаптивності.
5	Алгоритм мурашиної колонії	1) Здатність створення якісного розкладу. 2) Адаптованість до різних типів обмежень та критеріїв оптимізації. 3) Гнучкість до динамічних змін. 4) Стійкість до невеликих неточностей.	1) Обчислювальна і складність та час виконання. 2) Залежність від правильного вибору параметрів. 3) Відсутність гарантії глобального оптимуму. 4) Потребує місця для зберігання інформації.
6	Генетичний алгоритм	1) Ефективність у пошуку рішень для складних задач. 2) Можливість уникнути локальні оптимуми. 3) Гнучкість та адаптивність. 4) Можливість розпаралелювання.	1) Обчислювальна і складність та час виконання. 2) Складність налаштування параметрів. 3) Відсутність гарантії глобального оптимуму. 4) Складність розробки функції придатності та кодування рішення хромосомами.

З метою автоматизації процесу формування розкладу занять був розроблений вебзастосунок SchedGo, що виступає як централізована платформа для управління академічним розкладом. Як мову програмування для backend частини обрано JavaScript. Для створення серверної логіки використано платформу Node.js. Це сучасний інструмент для створення ефективних backend-рішень, який базується на JavaScript, що забезпечує єдність мови в усьому стеку. Завдяки асинхронній моделі обробки запитів Node.js забезпечує високу продуктивність, що критично важливо при роботі з великою кількістю одночасних підключень. На клієнтській стороні також використовується JavaScript. Для front-end реалізації застосовано бібліотеку React. Головною перевагою є використання віртуального DOM, що дозволяє ефективно оновлювати лише змінені елементи сторінки. Це значно підвищує швидкодію й полегшує створення інтерактивних інтерфейсів. Для збереження даних, таких як розклад, викладачі, академічні групи, тощо, використано базу даних MongoDB. Вона поєднує гнучкість нереляційних баз з потужністю традиційних СУБД, дозволяє зберігати документи у JSON-подібному форматі, що ідеально підходить для роботи з Node.js. Система підтримує індекси за будь-якими полями, що підвищує ефективність запитів, а також дозволяє зберігати великі об'єкти (наприклад, файли).

Для кращого розуміння загального функціоналу на рис. 4 зображено діаграму варіантів використання, на рис. 5 представлено діаграму класів бази даних для визначення логіки зв'язків між сутностями.

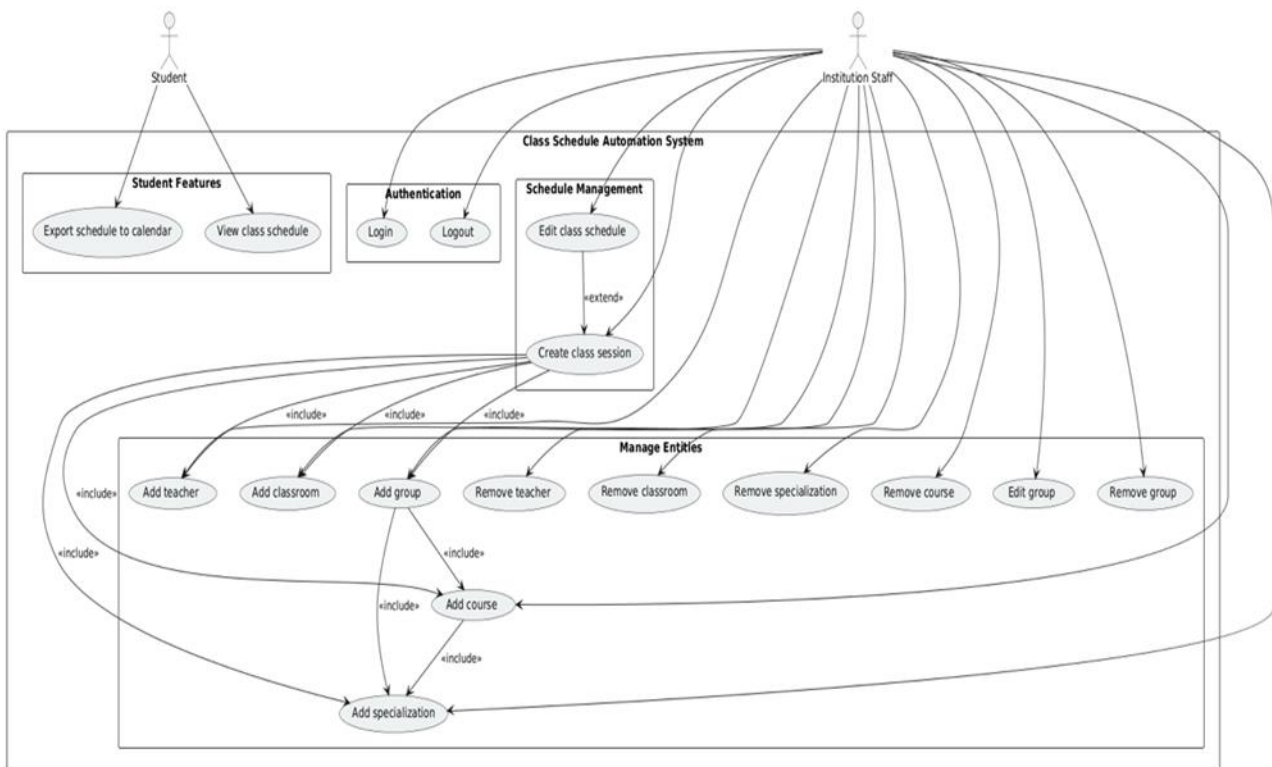


Рисунок 4. Діаграма варіантів використання

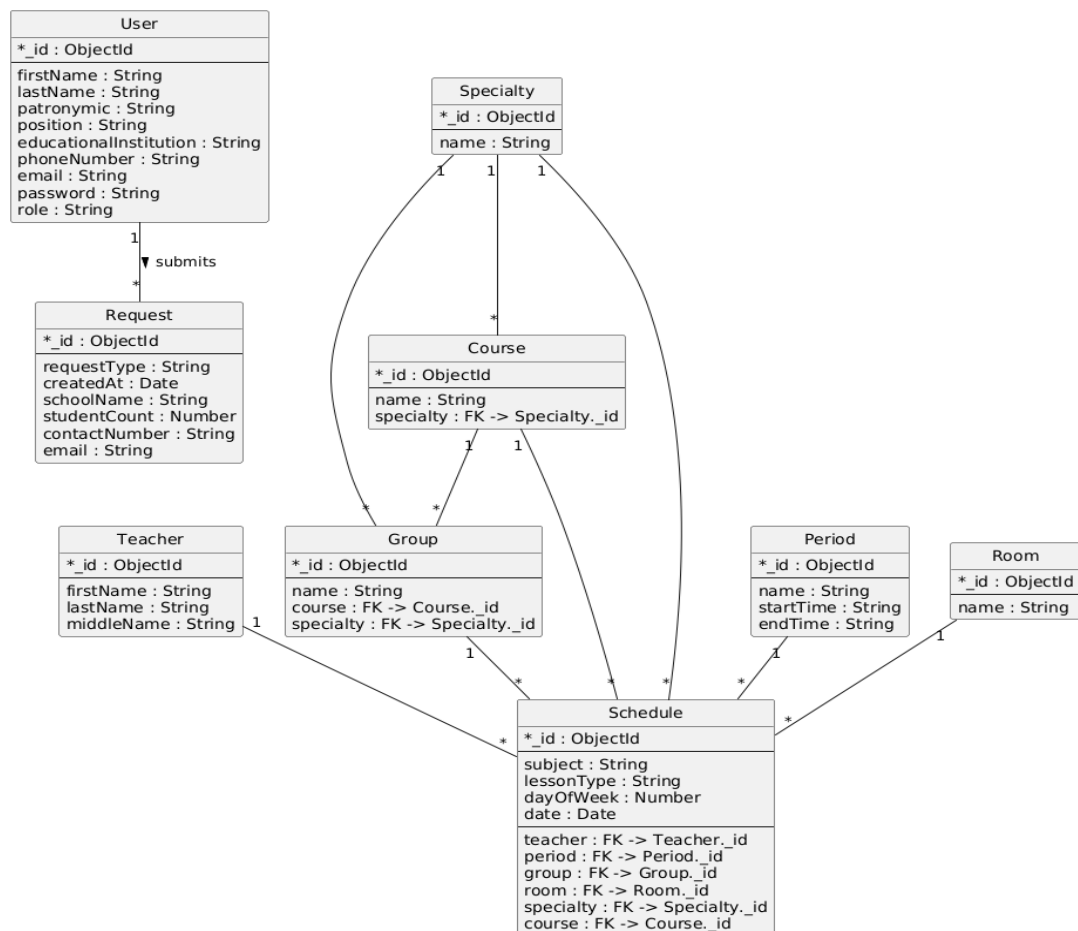


Рисунок 5. Діаграма структур баз даних

Система базується на ключових сутностях:

1. Користувач (User): фізична особа (адміністратор, викладач, студент) з персональними даними (ім'я, прізвище, логін, email, пароль, роль) та прив'язкою до навчального закладу (schoolId).
2. Запит (Request): фіксує звернення користувачів щодо додавання нового навчального закладу, містить тип, статус, дату та дані відправника.
3. Спеціальність (Specialty): назва освітнього напрямку, що може містити кілька курсів.
4. Курс (Course): рік навчання (1-4), підпорядкований певній спеціальності.
5. Група (Group): назва навчальної групи, що об'єднує студентів певного курсу та спеціальності.
6. Викладач (Teacher): прізвище, ім'я та по-батькові викладача.
7. Період (Period): часові рамки проведення занять (пар).
8. Кабінет (Room): номери аудиторій, у яких проводяться заняття.
9. Центральною частиною бази є сутність розклад (Schedule). Вона об'єднує в собі всю інформацію про окрему пару: назву предмета, тип заняття (лекція, практика тощо), день тижня, дату, а також зв'язки з викладачем, періодом, аудиторією, групою, курсом і спеціальністю. Кожне поле цієї моделі є зовнішнім ключем до відповідної сутності, що дозволяє не дублювати дані, а лише посилатися на них.

Таким чином, база даних побудована за принципами нормалізації, що забезпечує уніфікацію та зменшення надлишковості інформації. Завдяки такій структурі система здатна ефективно керувати складними навчальними розкладами та легко масштабуватись при збільшенні обсягу даних.

На рис. 6 представлено інтерфейс, з яким безпосередньо взаємодіє користувач, який переглядає розклад за спеціальністю, курсом та групою.

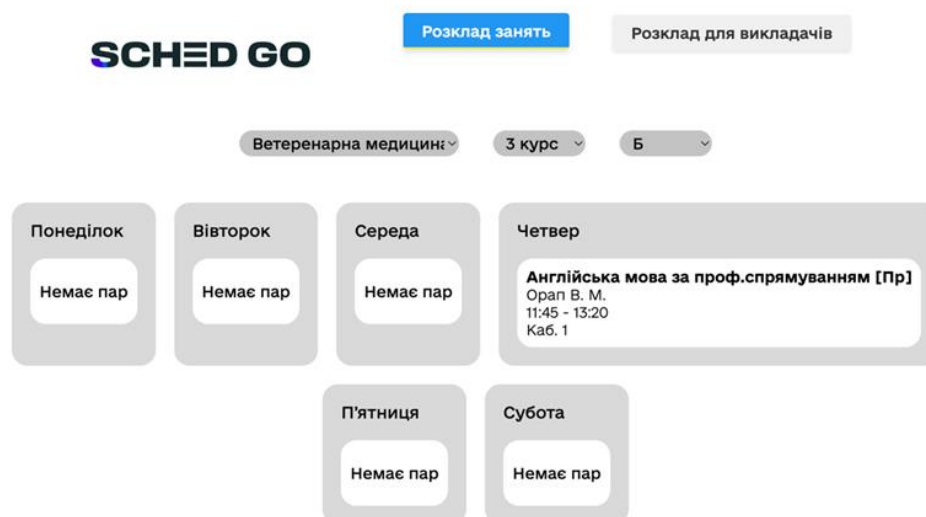


Рисунок 6. Користувацька частина відображення розкладу

Для швидкого перегляду розкладу занять та отримання повідомлень про зміну розкладу було розроблено Telegram-бот за допомогою мови програмування Python. Взаємодія з ботом відбувається через зручне меню, що дає змогу легко обирати розклад на сьогодні або на тиждень. Завдяки цьому бот допомагає ефективно організувати навчальний процес і бути завжди в курсі актуальної інформації.

Ще один важливий сценарій — створення нового заняття. Цей процес ініціюється адміністратором, який через вебінтерфейс обирає спеціальність, курс і групу. Далі зазначається дата проведення заняття, його назва, а також обирається час із фіксованого списку, що підтягується через API. Вид заняття (лекція, лабораторна тощо) обирається зі статичних значень, закладених у код. Останнім кроком є вибір викладача та аудиторії, дані про які також надходять через API. Після завершення всіх етапів інформація зберігається в базі даних, і нове заняття автоматично стає доступним для студентів у їхньому розкладі.

На рис. 7 представлена діаграма, що ілюструє логіку обміну інформацією між ключовими компонентами системи SchedGo — вебсайтом, Telegram-ботом, API, базою даних і користувачами — під час основних сценаріїв роботи: перегляд розкладу, взаємодія з ботом та створення нових занять.

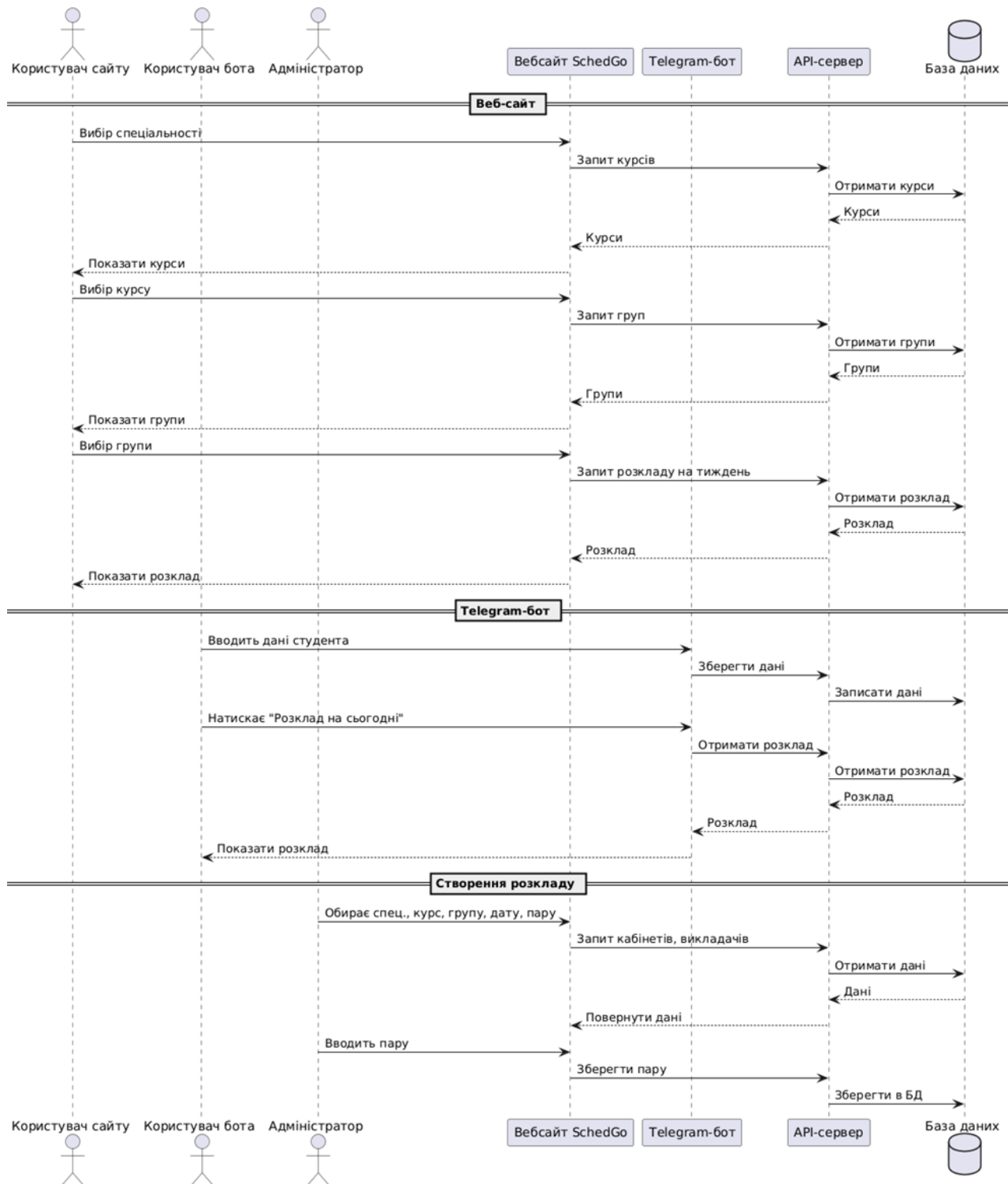


Рисунок 7. Діаграма потоків

На першому етапі взаємодії, користувач заходить на сайт <https://schedgo.online/schedule>, де йому пропонується обрати спеціальність. Система у відповідь на цей вибір надсилає запит до API, щоб отримати перелік курсів, прив'язаних до зазначеної спеціальності. Після отримання списку курсів користувач обирає потрібний, і тоді відбувається наступний запит — вже на

отримання груп, що відповідають зазначеним спеціальності та курсу. Завершивши вибір групи, користувач запускає фінальний запит, який повертає розклад на поточний тиждень. Усі ці дані проходять через API, що виконує відповідні запити до бази даних, обробляє їх і повертає результат вебінтерфейсу.

Інший шлях взаємодії — через Telegram-бот. Під час першого контакту з ботом користувач надсилає свої основні дані: спеціальність, курс і групу. Ця інформація зберігається в системі, що дозволяє надалі формувати запити автоматично. При кожному зверненні до бота — наприклад, натисканні кнопки «Розклад на сьогодні» — бот передає запит до API, використовуючи збережену інформацію. API, у свою чергу, звертається до бази даних і повертає відповідь з розкладом на поточний день або тиждень, залежно від дії.

Загалом, діаграма відображає чітку взаємодію між фронтом (вебсайт і бот), API та базою даних. Вона демонструє прозору логіку обробки запитів, автоматизацію дій користувача і ефективний обмін інформацією в рамках навчального процесу.

Особлива увага приділена безпеці: реалізовано реєстрацію, шифрування даних, захищену взаємодію з API [303].

Таким чином, у даному дослідженні були проаналізовані існуючі методи складання автоматизованого розкладу і представлено розробку вебзастосунка SchedGo та Telegram-бота, які виступають як централізовані платформи для управління академічним розкладом.

Вважаємо, що автоматизація розкладу у закладах освіти є важливим кроком до удосконалення навчального процесу, підвищення його ефективності та зручності для всіх учасників. Це інвестиція в якість освіти та оптимізацію адміністративних процесів.