

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

«Розробка та дослідження компонента автоматизованого створення розкладу інформаційної системи медичного закладу»
(тема)

Виконав: студент II курсу, групи ІТІм-20-1

Спеціальність 122-Компютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології проектування
(повна назва освітньої програми)

Гончаренко О.І.
(прізвище, ініціали)

Керівник Решетнік В.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри системотехніки _____
(підпис)

Гребеннік І.В.
(прізвище, ініціали)

2021р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Системотехніки
Рівень вищої освіти другий (магістерський)
Спеціальність 122-Компютерні науки
Тип програми освітньо-професійна
(код і повна назва)
Освітня програма Інформаційні технології проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри

Гребеннік І.В.

(підпис)

«___» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Гончаренко Олександрі Ігорівні

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка та дослідження компоненту автоматизованого створення розкладу інформаційної системи медичного закладу»

затверджена наказом по університету від «08» листопада 2021р. No 1663 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 17 грудня 2021 р.

3. Вихідні дані до роботи ОС Windows 7, Visual Studio Code, MongoDB, Express, Mongoose, Draw.io.

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Аналіз особливостей предметної області; 4.3 Огляд існуючих МІС; 4.3.1 Система Helse; 4.3.2 Система Medcard24; 4.3.3 Система «Поліклініка без черг»; 4.4 Аналіз сучасного стану досліджень з теми; 4.5 Постановка задачі дослідження; 4.6 Табличний метод створення розкладу; 4.7 Автоматична генерація розкладу методом задоволення обмежень; 4.8 Генетичний алгоритм; 4.9 Мурашиний алгоритм; 4.10 Метод гілок та меж; 4.11 Вибір алгоритму для проектування та розробки компоненту генерації розкладу; 4.12 Проектування функціоналу компоненту автоматичного створення розкладу; 4.13 Діаграми класів для компоненту, інтегрованого у МІС; 4.14 Діаграма варіантів використання для компоненту ІСМЗ; 4.15. Обґрунтування вибору мови програмування та СУБД; 4.16 Розробка структури компоненту МІС; 4.17 Програмна реалізація генетичного алгоритму автоматизованої генерації розкладу роботи лікаря; 4.18 Оцінка ефективності розробленого компоненту; 4.19 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 5.1 Діаграма структури ІСМЗ; 5.2 Схема генетичного алгоритму; 5.3 Контекстна діаграма функціональної моделі компоненту ІСМЗ; 5.4 Діаграма декомпозиції першого рівня; 5.5 Діаграма класів ІСМЗ; 5.6 Діаграма варіантів використання компоненту медичної ІС; 5.7 Структурна схема ІСМЗ; 5.8 Схема структури бази даних ІСМЗ; 5.9 Графік залежності кількості записів від часу після інтеграції у систему модуля автоматичної генерації розкладу; 5.10 Графік ефективності розробленого компоненту автоматичної генерації розкладу

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	1.09.2021	
2	Аналіз предметної області з медичних інформаційних систем	05.09.2021 - 15.10.2021	
3	Огляд існуючих підходів технологій та методів у ПО	16.10.2021 - 20.10.2021	
4	Дослідження алгоритмів розв'язання задачі автоматичної генерації розкладу автоматичної генерації розкладу	21.10.2021 - 02.11.2021	
5	Розробка вимог та проектування компоненту генерації розкладу	03.11.2021 - 06.11.2021	
6	Програмна реалізація компоненту ІСМЗ	07.11.2021 - 06.12.2021	
7	Оформлення пояснювальної записки	7.12.2021 - 16.12.2021	
8	Представлення на рецензування		
9	Представлення дипломної роботи в деканат		

Дата видачі завдання 01 вересня 2021 р.

Студент _____
(підпис)

Гончаренко О.І.

Керівник роботи _____
(підпис)

доц. каф. СТ Решетнік В.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до звіту містить: 70 сторінку, 10 рисунків, 12 формул, 25 джерел, 1 таблицю, 2 додатки.

БАЗА ДАНИХ, СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ, ІНФОРМАЦІЙНА СИСТЕМА МЕДИЧНОГО ЗАКЛАДУ, АВТОМАТИЧНА ГЕНЕРАЦІЯ РОЗКЛАДУ, АЛГОРИТМ

Об'єкт дослідження – процес створення розкладу працівників в ІС медичного закладу.

Предмет дослідження: технології та методи автоматичної генерації розкладу працівників медичного закладу.

Метою дослідження є знаходження оптимального алгоритму для автоматизації формування розкладу лікарів, програмна реалізація компоненту генерації розкладу та його інтеграція у ІСМЗ.

Методи дослідження: системний підхід, методи структурного аналізу і моделювання.

Результати досліджень, отримані у роботі можуть бути використані для модернізації ІСМЗ.

ABSTRACT

Master's Thesis: 70 pages, 10 figures, 12 formulas, 25 titles, 1 table, 2 appendices.

DATABASE, DATABASE MANAGEMENT SYSTEM, HOSPITAL
INFORMATION SYSTEM, AUTOMATIC SCHEDULE GENERATION,
ALGORITHM

The object of research is the process of creating a schedule of employees in the IS of a medical institution.

Subject of research: technologies and methods of automatic generation of employees' schedule.

The aim of the study is to find the optimal algorithm for automating the formation of doctors' schedules, software implementation of the schedule generation component and its integration into the hospital information system.

Research methods: systems approach, methods of structural analysis and modeling.

The results of research obtained in this work can be used to upgrade the hospital information system.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ З МЕДИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ.....	8
1.1 Аналіз особливостей предметної області.....	8
1.2 Огляд існуючих МІС.....	23
1.2.1 Система Helsi.....	23
1.2.2 Система Medcard24.....	25
1.2.3 Система “Поліклініка без черг”.....	27
1.3 Аналіз сучасного стану досліджень з теми.....	29
1.4 Постановка задачі дослідження.....	30
2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ РОЗКЛАДУ.....	32
2.1 Табличний метод створення розкладу.....	32
2.2 Автоматична генерація розкладу методом задоволення обмежень.....	32
2.3 Генетичний алгоритм.....	34
2.4 Мурашиний алгоритм.....	39
2.5 Метод гілок та меж.....	44
2.6 Вибір алгоритму для проектування та розробки компоненту генерації розкладу.....	47
3 РОЗРОБКА ВИМОГ ТА ПРОЕКТУВАННЯ КОМПОНЕНТУ ГЕНЕРАЦІЇ РОЗКЛАДУ.....	48
3.1 Проектування функціоналу компоненту автоматичного створення розкладу.....	48
3.2 Діаграми класів для компоненту, інтегрованого у МІС.....	52
3.3 Діаграма варіантів використання для компоненту ІСМЗ.....	56
4 ПРОГРАМНА РЕАЛІЗАЦІЯ КОМПОНЕНТУ МІС.....	59
4.1. Обґрунтування вибору мови програмування та СУБД.....	59
4.2 Розробка структури компоненту МІС.....	59
4.3 Програмна реалізація генетичного алгоритму автоматизованої генерації розкладу роботи лікаря.....	64
4.4 Оцінка ефективності розробленого компоненту.....	65
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	69
ДОДАТОК А.....	71
ДОДАТОК Б.....	82
Відомість магістерської кваліфікаційної роботи.....	91

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ

ІС – інформаційна система.

МІС – медична інформаційна система.

ПО – предметна область.

СУБД – система управління базами даних.

БД – база даних.

ЕМК – електронна медична картка.

ЛІС – лабораторна інформаційна система.

АРМ – автоматизоване робоче місце.

СУБД – система управління базами даних.

API – Application Programming Interface.

ІСМЗ – інформаційної системи медичного закладу.

ВСТУП

У зв'язку з загальною тенденцією переводити інформаційні системи підприємств у цифрову форму, наявність зручної та ефективної інформаційної системи є необхідною мірою будь-якого підприємства для підтримання здорової конкуренції серед інших.

Актуальність роботи обумовлена загальним поширенням інформаційних систем медичних закладів(ІСМЗ) та зростанням попиту населення на медичні послуги.

Метою дослідження є знаходження оптимального алгоритму для автоматизації формування розкладу лікарів, програмна реалізація компоненту генерації розкладу та його інтеграція у ІСМЗ.

Об'єкт дослідження – процес створення розкладу працівників в ІС медичного закладу.

Предмет дослідження: технології та методи автоматичної генерації розкладу працівників медичного закладу.

Методи дослідження: системний підхід, методи структурного аналізу.

Наукова новизна роботи полягає у дослідженні алгоритмів для генерації розкладу лікаря та у пропонуванні методу генерації, нового для даної предметної області.

Результати досліджень, отримані у роботі можуть бути використані для модернізації ІСМЗ.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ З МЕДИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Аналіз особливостей предметної області

Сучасний стан медичних інформаційних систем (ІСМЗ) на цей момент є таким, що переважна більшість медичних закладів відійшли від фізичного зберігання даних у картотеках та архівах і використовують ІСМЗ для автоматизації основних бізнес-процесів, для поліпшення роботи працівників, покращення якості обслуговування, зменшення витрат часу та забезпечення зручної роботи за даними як лікарів так і пацієнтів.

Сучасне розуміння інформаційної системи припускає використання комп'ютера, як основного технічного засобу переробки інформації, ІСМЗ не є винятком, тому для її використання в облаштованих лікарнях медичні працівники мають свій персональний комп'ютер або автоматизоване робоче місце (АРМ), тому що, технічне втілення інформаційної системи не має сенсу, якщо не врахована роль людини, для якої призначена вироблена інформація і без якої неможливе її одержання.

Оскільки система призначена для передачі, збору, накопичення, опрацювання, зберігання, подання і використання інформації, необхідно визначити до якого типу технології відноситься аналізована ІСМЗ, щоб виділити основні особливості системи та розробленого компоненту.

Інформаційні технології значно різняться за видами процесів управління, рівнем управління, сферою роботи економічного об'єкту, типами об'єктів управління, ступенем автоматизації управління, характером та обсягом розв'язуваних задач та іншими ознаками.

Загальноприйнятої класифікації інформаційних технологій в цей час не існує, тому у багатьох випадках їх класифікують за різними ознаками.

Для визначення типу технологій компоненту автоматизованої генерації розкладу для цієї системи, розглянемо приклад класифікації інформаційних

технологій, наведений у таблиці нижче [2].

Таблиця 1.1 – Класифікація інформаційних технологій

Класифікаційна ознака	Вид інформаційних технологій
За способом реалізації в ІС	Традиційні
	Нові інформаційні технології
За ступенем охоплення завдань управління	Електронна обробка даних
	Автоматизація функцій управління
	Підтримка прийняття рішень
	Електронний офіс
	Експертна підтримка
За класом реалізованих технологічних операцій	Робота з текстовим редактором
	Робота з табличним процесором
	Робота з СУБД
	Робота з графічними об'єктами
	Мультимедійні системи
	Гіпертекстові системи
За типом інтерфейсу користувача	Пакетні
	Діалогові
	Мережеві
За способом побудови мережі	Локальні
	Багаторівневі
	Розподілені
За предметними областями	Бухгалтерський облік
	Банківська діяльність
	Податкова діяльність
	Страхова діяльність
	Медична діяльність
	Інші

Компонент медичної системи згідно класифікації є новими інформаційними технологіями для електронної обробки даних та роботою з СУБД, діалоговим типом інтерфейсу та багаторівневою побудовою мережі. Особливість нових інформаційних технологій – зручний інтерфейс, використання персональних

комп'ютерів. Компонент автоматичного створення розкладу у перспективі вбудована у архітектуру інформаційної системи медичного закладу, що використовує БД та зберігає інформацію про користувачів, персонал лікарні, результати медичних досліджень та розклад лікарів. Діалоговий тип інтерфейсу дозволяє користувачеві взаємодіяти з даними та впливати на процес їх обробки.

ІСМЗ – комплексний програмний продукт, головним призначенням якого є автоматизація всіх основних процесів, пов'язаних із роботою медичних закладів загальної та вузької спеціалізації. Автоматизовані ІСМЗ дозволяють швидко та ефективно налагодити електронний документообіг, роботу з пацієнтами, вести оперативний облік роботи адміністративного персоналу, контролювати всі організаційні та фінансові питання [4].

Тепер коли визначений тип технологій цієї ІСМЗ, розглянута класифікація саме медичних інформаційних систем та визначено місце системи у цій класифікації.

Класифікація медичних інформаційних систем на ієрархічному принципі відповідає багаторівневій структурі охорони здоров'я [5]. Розрізняють:

а) ІСМЗ базового рівня, основна мета яких – комп'ютерна підтримка роботи лікарів різних спеціальностей. Вони дозволяють підвищити якість профілактичної та діагностичної роботи, в умовах масового обслуговування при дефіциті часу кваліфікованих фахівців. За завданнями виділяють:

– інформаційно-довідкові системи (призначені для пошуку та видачі медичної інформації на запит користувача);

– консультативно-діагностичні системи (для діагностики патологічних станів, включаючи прогноз та вироблення рекомендацій щодо способів лікування, при захворювання різного профілю);

– прикладно-комп'ютерні системи (для інформаційної підтримки та/або автоматичного діагностичного та лікувального процесу, що здійснюються при безпосередньому контакті з організмом хворого);

– автоматизовані робочі місця спеціалістів.

б) ІСМЗ рівня лікувально-профілактичних закладів. Виділені в основні групи:

- інформаційними системами консультативних центрів (призначеними для забезпечення інформаційної підтримки лікарів при консультуванні, діагностиці та прийнятті рішень при невідкладних станах);

- банками інформації медичних служб (містять зведені дані про якісних та кваліфікованих працівників закладів, населення, основні статистичні відомості);

- персоніфікованими реєстрами;

- скринінговими системами (для проведення долікарського профілактичного огляду населення);

- інформаційними системами лікувально-профілактичного закладу (засновані на об'єднанні всіх інформаційних потоків у єдину систему та забезпеченні автоматизації різних видів діяльності закладу);

- інформаційними системами та медичні вузів.

в) ІСМЗ територіального рівня, такі як:

- ІС територіального органу охорони здоров'я;

- ІС для вирішення медико-технологічних завдань, що забезпечують інформаційної підтримки діяльності медичних працівників медичних служб;

- комп'ютерні телекомунікаційні медичні мережі, які забезпечують створення єдиного інформаційного простору лише на рівні регіону.

г) державний рівень, призначені для інформаційної підтримки державного рівня системи охорони здоров'я.

Система, що розглядається є ІС лікувально-профілактичного закладу, а саме компонент генерації розкладу є підсистемою інформаційно-довідкового типу.

ІСМЗ лікувально-профілактичного закладу призначена для автоматизації його бізнес-процесів та заміни новою ІС старої традиційної інформаційної системи з фізичним зберіганням даних. Основними цілями впровадження ІСМЗ є:

- підвищення якості та доступності медичної допомоги населенню;

- зниження витрат на надання медичної допомоги при збереженні рівня

результата;

- підвищення ефективності роботи медичної організації;
- залучення громадян у турботу про власне здоров'я;
- забезпечення обґрунтованості та оперативності прийняття управлінських рішень;
- підтримка прийняття лікарських рішень;
- покращення та оптимізація роботи реєстратури медичного закладу;
- спрощення процесу взаємодії з іншими структурними підрозділами медичного закладу;
- впровадження інформаційних технологій як невід'ємної частини функціонування сучасної медичного закладу, активне залучення до їх використання медичного персоналу.

Тепер, коли виділені основні цілі системи, виділемо способи їх реалізації, що можна віднести до основних вимог до існуючої медичної інформаційної системи. Перелічені цілі досягаються реалізацією таких частин ІСМЗ:

- автоматизації медичної та адміністративної діяльності під час здійснення лікувального процесу на об'єктах автоматизації;
- ведення медичної документації в електронному вигляді (ведення електронної медичної картки);
- забезпечення персоніфікованого обліку надання медичних послуг;
- зіставлення переліку рекомендованих лікувальними стандартами заходів та послідовності їх проведення із зафіксованими в ІСМЗ лікувально-діагностичними призначеннями та їх виконанням;
- забезпечення інформаційної взаємодії організацій системи охорони здоров'я, учасників лікувально-діагностичного процесу.

Для забезпечення реалізації цілей ІСМЗ має зберігати персональні дані пацієнтів та лікарів, що занесені у систему на сайті: їх адреси, прізвища, ініціали, телефони, час роботи персоналу медичного закладу та інше. Ці дані система медичної закладу використовує для роботи, отже вони зберігатимуться у БД та оброблятися, то ж БД повинна дати змогу не тільки зберігати але й змінювати та

видаляти дані. Наявність БД дозволяє використовувати стандартні процедури обробки файлів ІСМЗ, як і кожен її автономний блок та складається з обов'язкових програмних модулів:

- модуль збирання інформації;
- база даних;
- модуль обробки та аналізу даних;
- модуль керування документообігом;
- модуль керування медичними апаратними засобами.

Для кращого розуміння ПО крім можливостей бази даних інформаційної системи розглянуто базовий функціонал який характерний для більшості систем такого типу. Виділені та більш детально розглянуті функції, інформація про які використана для подальшого проектування компоненту автоматизованої генерації розкладу лікарів та його інтеграція у ІСМЗ:

- інформаційної підтримки процесу надання медичної допомоги на рівні медичної організації, ведення електронної медичної картки пацієнта, медико-технологічних процесів у межах медичної організації;

- інформаційної підтримки процесу управління медичної організації, управління адміністративно-господарською діяльністю медичної організації, формування та передачу даних про витрати за надану медичну допомогу та лікарське забезпечення;

- інформаційної підтримки процесів взаємодії з пацієнтами, надання можливості запису та самозапису пацієнта на прийом до лікаря, інформаційного наповнення особистого кабінету пацієнта, видачі пацієнту електронних копій медичних документів.

Бізнес-процес – це послідовність дій, виконаних для перетворення інформаційних і матеріальних потоків з метою збільшення їх цінності для клієнта або підвищення вартості товару, або надання послуг, що задовольняють потребам клієнта. Перетворення інформаційних та матеріальних потоків тягне за собою отримання нової інформації та продуктів, які й набувають більшої цінності для клієнта [6].

Для того щоб краще зрозуміти функціонал ІСМЗ розглянемо структуру медичної системи широкого профілю з автоматизацією широкого спектру підсистем на рисунку 1.1.

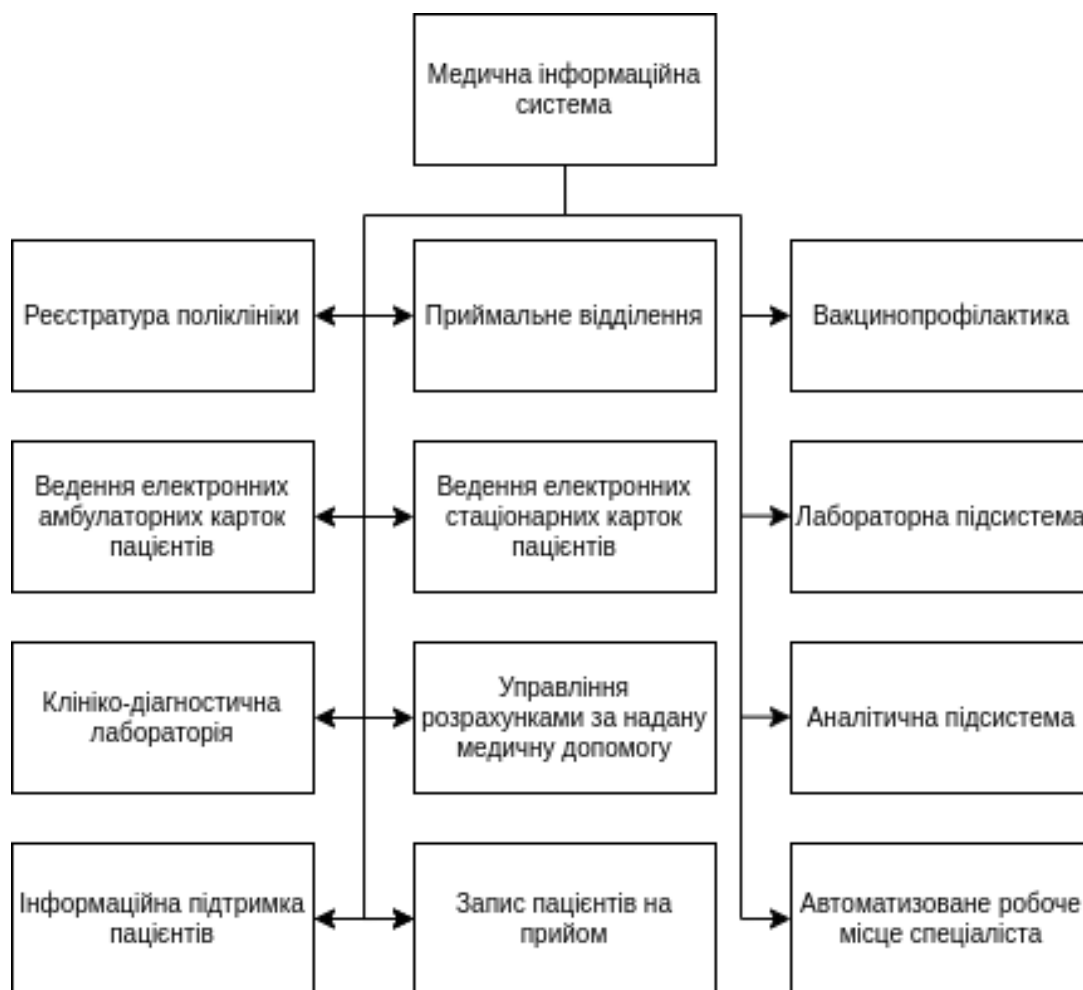


Рисунок 1.1 – Діаграма структури ІСМЗ

Як правило, кожна медична інформаційна система базового рівня складається з блоків, які відповідають за автоматизацію різних складових. Базова ІСМЗ має такі підсистеми:

- підсистема «Реєстратура поліклініки»;
- підсистема «Приймальне відділення»;
- підсистема «Ведення електронних амбулаторних карток пацієнтів»;
- підсистема «Ведення електронних стаціонарних карток пацієнтів»;
- підсистема «Клініко-діагностична лабораторія»;
- підсистема «Управління розрахунками за надану медичну допомогу»;
- підсистема «Інформаційна підтримка пацієнтів»;

- підсистема "Запис пацієнтів на прийом";
 - підсистема «Вакцинопрофілактика»;
 - лабораторна підсистема;
 - аналітична підсистема;
- автоматизоване робоче місце спеціаліста.

Детальний опис підсистем ІСМЗ та їх функцій:

Можливості підсистеми «Реєстратура медичного закладу»

Функція реєстрації пацієнтів:

- запис пацієнтів на прийоми лікаря за допомогою електронних талонів;
- реєстрація персональних даних пацієнтів;
- пошук громадянина за ідентифікатором (номер полісу обов'язкового медичного страхування та інші);
- можливість створення електронної карти пацієнта(ЕМК);
- автоматизоване ведення реєстру пільговиків та інвалідів, закріплених за медичним закладом;
- можливість внесення полісів, договорів на обслуговування;
- облік, перереєстрації громадян, що обслуговуються, аналіз руху прикріпленого контингенту;
- можливість внесення позначок про прикріплення пацієнтів;
- зчитування відомостей про страхування зі штрих-коду друкованого поліса єдиного зразка з універсальної електронної карти.

Функція «Друк документів»:

- талон для пацієнта (нагадування про час і місце прийому);
- ЕМК;
- статистичний талон амбулаторного пацієнта, медична картка амбулаторного пацієнта;
- інформована згода на обробку персональних даних;
- можливість гнучкого налаштування друкованої форми будь-яких документів, у тому числі можливість створення медичним закладом власних варіантів друкованих форм будь-яких документів.

Така підсистема дозволяє скорочує рутинну роботу персоналу, допомагає в оперативному управлінні, що дозволяє отримувати всі види статистичних даних, необхідних для фінансового та економічного аналізу, а також організувати спільну роботу всіх служб і тим самим скоротити як тимчасові, так і фінансові втрати, пов'язані з помилками персоналу чи пацієнтів.

Підсистема «Приймальне відділення»

Можливості функції «Реєстрація пацієнтів»:

- реєстрація медичних даних, які обслуговують пацієнти;
- пошук громадянина за ідентифікатором (номер полісу обов'язкового медичного страхування та інші);
- внесення інформації з документів, що засвідчують особистість;
- створення ЕМК;
- внесення полісів, договорів на обслуговування;
- одержання інформованої згоди на обробку персональних даних;
- інтеграція з централізованою базою даних застрахованих (реєстром застрахованих);
- використання універсальної електронної картки як універсальної ідентифікатора пацієнта та користувача;
- зчитування відомостей про страхування зі штрих-коду друкованого поліса єдиного зразка з електронного полісу, універсальної електронної карти;
- облік відмов від госпіталізації;
- ведення черги планових госпіталізацій пацієнтів.

«Документація лікування»:

- огляд лікаря приймального відділення;
- рахунок медичних послуг, наданих під час перебування пацієнта у приймальному відділенні;
- згода пацієнта на медичне втручання.

У приймальному відділенні стаціонару також встановлюється програма «Реєстратор стаціонару», що дозволяє реєструвати всіх пацієнтів які на лікуванні. Все це дає можливість створити єдиний реєстр пацієнтів, які отримують медичну

допомогу в клініці, незалежно від того, у яких підрозділах проходило їх лікування та діагностика. На наступному етапі налагоджується комп'ютерний облік усіх наданих пацієнту послуг - у стаціонарі, поліклініці, лабораторії, що дозволяє сформувати медичну статистику. В результаті цього кожен пацієнт отримує електронну історію хвороби, що містить усі відомості про звернення, випадки госпіталізації, надані послуги та встановлені діагнози. В результаті можна буде отримувати весь комплекс медико-статистичної інформації як стандартизованої, так і довільним параметрам.

Підсистема "Ведення електронних амбулаторних карток пацієнтів"

Можливості функції «Ведення медичної документації»:

- ведення документації лікарських оглядів;
- реєстрація діагнозів пацієнта;
- реєстрація лікарських призначень пацієнту (консультацій, лабораторних, інструментальних, рентгенологічних досліджень, амбулаторних операцій, процедур, медикаментозних призначень) та їх результатів;
- облік випадків звернень пацієнта, включаючи реєстрацію фактів відкриття, закриття випадку та результату звернення, наданих послуг;
- облік та реєстрація показників стану здоров'я пацієнта результатів профілактичних оглядів;
- підтримка облікових форм для медичних закладів;
- реєстрація вакцинації, імунізації та їх результатів;
- автоматичний контроль та підказки рекомендованих доз та сумісності призначених лікарських препаратів.

Інтеграція з зовнішніми системами:

- отримання документів або записів електронних медичних карток пацієнта на запит з інтегрованої ЕМК;
- ведення електронного листа призначень;
- формування напрямків на отримання медичної допомоги в закладах охорони здоров'я, включаючи направлення на госпіталізацію, санаторно-курортне лікування, та реєстрація їх результатів;

- формування рецептів на отримання лікарських засобів;
- передача документів або записів електронних амбулаторних карт пацієнта, та відомості про напрямки та рецепти, екстрені повідомлення про захворювання в інтегровану ЕМК;

- ідентифікація лікаря та пацієнта на основі унікальних ідентифікаторів.

Підсистема «Ведення електронних стаціонарних карток пацієнтів»

Можливості функції «Ведення медичної документації»:

- ведення документації лікарських оглядів, включаючи первинний, епікризи, щоденникові записи;

- реєстрація діагнозів пацієнта;

- реєстрація лікарських призначень пацієнту (консультацій, лабораторних, інструментальних досліджень, процедур, та іншого) та їх результатів;

- формування листів призначень відповідно до лікарських призначень, вимірювання та реєстрація показників стану здоров'я пацієнта;

- ведення електронного температурного листа;

- підтримка облікових форм для закладів стаціонарного типу;

- автоматичний контроль та підказки рекомендованих доз та сумісності призначених препаратів.

Інтеграція та обмін даними:

- передача в ЕМК результатів досліджень;

- отримання з ЕМК напрямів на дослідження;

- передача медичних структурованих документів пацієнта, інтегровану ЕМК;

- ідентифікація лікаря та пацієнта на підставі універсальної електронної картки громадянина;

- отримання документів або записів електронних медичних карток пацієнта по запиту з інтегрованої електронної медичної карти.

Ця підсистема дозволяє лікарям у зручній формі зберігати та передавати один одному матеріали, пов'язані з діагностикою та лікуванням пацієнта.

Підсистема «Клініко-діагностична лабораторія»

Можливості функції «Ведення обліку обладнання»:

- ведення переліку лабораторного обладнання медичного закладу;
- ведення переліку показань до досліджень та порушень, які виявляються під час обстеження.

Можливості функції «Ведення медичної документації»:

- реєстрація зразків, які надходять до лабораторії;
- формування протоколу дослідження.
- система розподілу та маршрутизації зразків: Створення настроюваних робочих листів для ручних методик;
- врахування особливостей цитологічних, гістологічних, бактеріологічних та інших досліджень;
- забезпечення комплексу лабораторного контролю якості;
- історія пацієнта, який проходив обстеження у лабораторії, з кожного дослідження або тесту;
- технічне підтвердження результатів на підставі заданих критеріїв та проходження контролю за кожним тестом;
- ведення журналів лабораторії;
- формування статистичної звітності;
- формування спеціалізованих модульованих та настроюваних бактеріологічних, цитологічних та гістологічних звітів;
- прикріплення зображень електронної мікроскопії.

Можливості інтеграції та обмін даними:

- інтеграція з медичним обладнанням, з можливістю отримання даних обладнання на ПК користувача та внесення цих даних до протоколу обстеження;
- автоматичний обмін даними із підключеними аналізаторами;
- можливість швидкого додавання аналізатора до системи без переривання робочого процесу лабораторії;
- можливість підключення автоматичних преаналітичних систем;
- можливість підключення автоматичного постаналітичного обладнання;
- передача в ЕМК результатів досліджень;
- отримання з ЕМК направлень на дослідження.

Підсистема «Управління розрахунками за надану медичну допомогу»

Можливості функції налаштування підсистеми:

- облік видів фінансування, з якими працює медичний заклад (кошти громадян, бюджети різних рівнів);
- ведення номенклатури послуг, що надаються закладом;
- облік для кожної послуги номенклатури видів фінансування, в рамках яких може надаватися ця послуга;
- ведення картотеки преїскурантів цін на послуги, що надаються медичним закладом;
- налаштування імпорту цін на послуги із зовнішніх джерел;
- ведення переліку контрагентів та договорів на надання медичних послуг.

Можливості функції «Облік наданих послуг»:

- облік наданих послуг із внесенням інформації про форму оплати, статус оплати.

Можливості функції «Формування звітності»:

- формування реєстрів рахунків за надану медичну допомогу та їх передача до зовнішньої інформаційної системи;
- надання знижок (пільг) при наданні послуг та формуванні рахунків-реєстрів;
- отримання даних про оплату або про відмову в оплаті виставлених рахунків зовнішньої системи.

Облік високотехнологічної медичної допомоги

- формування звітності про надану високотехнологічну медичну допомогу та передача її у зовнішні інформаційні системи моніторингу.

Підсистема "Статистика"

Можливості функції «Формування звітності»:

- попередній перегляд сформованого звіту, друк звітів;
- експорт звітів до офісних програм;
- експорт звітів до інших форматів;
- підготовка довільних аналітичних звітів щодо діяльності організації.

Підсистема «Інформаційна підтримка пацієнтів»

«Підтримка інформаційних терміналів»:

- підтримка роботи з електронною реєстратурою через інформаційний термінал для пацієнтів;
 - друк номера під час запису пацієнта через інформаційний термінал.
- «Надання доступу пацієнту або його законним представникам до медичної документації»:
- особистий кабінет пацієнта;
 - надання електронних копій медичних документів на запит пацієнта;
 - надання доступу до довідкової інформації про захворювання, методи лікування та профілактики.

Підсистема "Запис пацієнтів на прийом"

«Ведення розкладу лікарів»:

- автоматизоване ведення розкладу роботи лікарів та медичних сестер закладу;
- вбудована система обліку функції лікарської посади, навантаження;
- підтримка різних довідників видів прийому (первинний прийом, повторний прийом, консультація тощо);
- облік фактично прийнятих пацієнтів (які з'явилися і не з'явилися);
- вбудована система лімітів, обмежень доступу і т.д. для гнучкого налаштування календаря під індивідуальні особливості роботи кабінету;
- гнучкі функції індивідуального налаштування календаря для кожного користувача, включаючи налаштування відібрання номерків, бачить полів у таблиці календаря і т.д.

Функція друку:

- можливість автоматичного пошуку вільних талонів за групою лікарів;
- можливість групового запису відразу на кілька талонів (наприклад, при призначенні масажів або інших процедур, що вимагають неодноразового візиту до закладу);
- можливість автоматизованого перенесення записів до лікаря (наприклад, у

разі хвороби лікаря;

- можливість обмеження номерів, виходячи з рівня доступу користувача;
- можливість копіювання розкладу наступного тижня;
- друк розкладу лікаря/дільниці на день із записами пацієнтів;
- список пацієнтів на ділянку на день (виключивши вільний час).

Автоматизоване робоче місце спеціаліста

При створенні інформаційних медичних систем дуже важливо забезпечити санкціонування доступу до інформації, що зберігається в інформаційну систему. Система має працювати за таким принципом: якщо у користувача немає прав на отримання певної інформації, то ІСМЗ має вести себе так, начебто цієї інформації про пацієнта немає в базі даних.

ПК користувачі цієї ІСМЗ утворюють автоматизовані робочі місця (АРМ). АРМ - це робочий стіл та електронні документи спеціаліста. Робочий стіл має єдиний уніфікований інтерфейс – це організовані користувачем безліч об'єктів, до яких користувач має доступ. Робочий стіл забезпечує організацію уніфікованого робітника місця, яке має здатність конфігурації під конкретного користувача.

АРМ надає користувачам такі можливості:

- робота в уніфікованому інтерфейсі з будь-якими типами медичних електронних документів;
- доступ до документів та управління правами доступу до них;
- можливість розсилання медичних електронних документів на столи користувачів;
- можливість організації документів на своєму робочому столі;
- введення та зберігання даних лабораторних та діагностичних досліджень.

Забезпечення безпеки даних

Запобігти викрадення даних допоможуть стандартизація протоколів передачі даних у мережі, використання шифрування інформації, збереження зашифрованих пакетів даних у хмарних сервісах. Шифрування медичних даних, коли вони передаються окремими пакетами зашифрованими через незалежні лінії зв'язку, вирішує проблему безпеки. Фінальне об'єднання пакетів, внаслідок якого

інформація також залишається у зашифрованому вигляді, відбувається на сервері ІСМЗ, що робить безглуздом перехоплення окремих частин під час передачі. На етапі отримання даних у медичному закладі їх цілісність та незмінний характер перевіряють за допомогою коду аутентифікації повідомлення. Ключ для розшифровки даних є тільки у лікаря або, у разі персоналізованої медицини, на пристрої подачі ліків, що блокує доступ до інформації іншим медичним працівникам або зловмисникам.

Такий підхід до безпеки даних призведе до зростання довіри населення до пристроїв моніторингу показників здоров'я, що носять і вживлюються, розвиток галузі дистанційного медичного моніторингу, стандартизація лікувальних послуг, забезпечує створення безпечної розподіленої інфраструктури мережі передачі медичних даних, поширення технологій персоналізованої медицини.

Система розрахована на обслуговування великої кількості користувачів одночасно, з мінімальною затримкою, БД є високонавантаженою.

Користувачами системи будуть працівників медичного закладу та клієнти що є пацієнтами цього закладу та відвідують його для проведення медичних досліджень та лікування.

1.2 Огляд існуючих ІСМЗ

1.2.1 Система Helsi

Helsi – це зручна та надійна медична електронна система для пацієнтів, лікарів, державних та приватних медичних закладів, що зберігає дані про медичні заклади, їх персонал, автоматизує процеси реєстрації та документації медичних досліджень, спрощує їх пошук та обробку. Всі дані зберігаються у дата-центрі, який отримав сертифікат комплексної системи захисту інформації від Державної служби спеціального зв'язку і захисту інформації України [7].

Наведена система має такий функціонал для пацієнтів:

- можливість знайти та обрати лікаря;
- швидкий запис на прийом on-line;

- доступ до електронної медичної картки (ЕМК);
- результати аналізів та діагностики в кабінеті пацієнта;
- доступ до призначень лікаря та плану лікування.

Для лікарів:

- зручне ведення історії хвороби пацієнтів та емк;
- оперативне отримання результатів діагностики та аналізів;
- легке використання клінічних протоколів;
- зрозумілий кабінет для ведення прийому пацієнтів.

Для медичних закладів:

- комплексна автоматизація роботи медичного закладу;
- можливість налаштування helsinki під потреби медзакладу;
- моніторинг та управлінська статистика для керівників;
- формування поточної звітності та статистики;
- автоматизація реєстратури та роботи лікаря;
- управління розкладом лікаря;
- ведення емк;
- облік медичних препаратів та ведення оплат;
- формування звітів та статистики;
- конструктор бланків і форм;
- інтуїтивно зрозумілий веб-інтерфейс;
- гнучке налаштування прав доступу;
- фіксація дій користувачів;
- доступ пацієнтів до своєї емк;
- забезпечення надійного шифрування та безпека даних;
- портал з інформацією про лікарів, годинами їх прийому та адресами лікарень;
- підтримка для лікарів та пацієнтів від контакт-центру.

Розглянутий інтерфейс функції запису до лікаря на сайті Helsinki, на рисунку 1.3 видно дні з назвою та датою, що поділені на робочі та вихідні та кнопки що відображають час прийому лікаря та дозволяють записатися до нього на прийом

на цей самий час. Зайняті та вихідні дні є недоступними для запису та та відрізняються від робочих днів кольором. Так доступний час відображено синім кольором, а недоступний – сірим.

Запис на прийом

Умови прийому

Уточнюйте в реєстратурі ⓘ

<	Сьогодні, 11 груд	Завтра, 12 груд	Пн, 13 Груд	Вт, 14 Груд	Ср, 15 Груд	Чт, 16 Груд	Пт, 17 Груд
	—	—	14:20	09:20	13:00	08:20	13:00
	—	—	14:40	09:40	13:20	08:40	13:20
	—	—	15:00	10:00	13:40	09:00	13:40
	—	—	15:20	10:20	14:00	09:20	14:00
	—	—	15:40	10:40	14:20	09:40	14:20
	—	—	16:00	11:00	14:40	10:20	14:40
	—	—	16:20	11:20	15:00	10:40	15:00
	—	—	16:40	11:40	15:20	11:00	15:20
	—	—	17:00- 18:00 Немає вільних місць	12:00	15:40	11:20	15:40
	—	—		12:20	16:00	12:00	16:00
	—	—		12:40	16:20	12:20	16:20

Рисунок 1.3 – Приклад розкладу лікаря з додатку Helsi(вид для пацієнта)

1.2.2 Система Medcard24

Medcard24 – це сервіс автоматизації робочих процесів медичного закладу, схожий до попередньо описаного, але має менше функцій та забезпечує автоматизацію тільки частину бізнес процесів медичного закладу [8].

Система має такі модулі для пацієнтів:

- інформаційний модуль з даними про медичні заклади та лікарів;
- особистий кабінет пацієнта;

- медична картка пацієнта;
- запис на прийом до лікаря до конкретного лікаря, конкретного медичного закладу, який обирає пацієнт.

Недоліком є ускладнений процес підтвердження персональних даних для отримання доступу до інформації про медичний заклад та до історії лікування в особистому кабінеті. Пацієнту необхідно підтвердити персональні дані за номером телефону та документом, що посвідчує особу, який необхідно надати до реєстратури медичного закладу.

На рисунку 1.4 наведено інтерфейс розкладу медичного працівника у ІСМЗ Medcard24.

Запис до лікаря ✕

① Дані пацієнта
② Розклад лікаря
③ Вибір лікаря
④ Запис на прийом

Пацієнт: Сниткой Кирилл Владимирович

Чол. 26 років

Організація: Районна центральна лікарня м. Б...

Спеціальність: Лікар загальної практики - сімейн...

Дата: 23.11.2017

Підрозділ: Амбулаторія №2

Лікар: Марченко М.В.

Послуга: Оберіть зі списку

10:00	10:00	10:15	10:30	10:45
11:00	11:00	11:15	11:30	11:45
12:00	12:00	12:15	12:30	12:45
13:00	13:00	13:15	13:30	13:45
14:00	14:00	14:15	14:30	14:45
15:00	15:00	15:15	15:30	15:45
16:00	16:00	16:15	16:30	16:45
17:00	17:00	17:15	17:30	17:45

ВІДМІНИТИ
ДАЛІ

Рисунок 1.4 – Приклад розкладу лікаря з додатку Medcard24

Система Medcard24 має такі елементи інтерфейсу розкладу медичного працівника з можливістю запису до нього:

- етапи запису до лікаря;
- інформація про організацію;
- інформація про підрозділ;
- спеціальності лікаря;
- інформація про лікаря: прізвище та ініціали;
- меню де можна обрати дату на яку буде створений запис;
- випадаючий список з можливістю обрати послугу яку буде надавати лікар;
- таблиця з кнопками, що відображають час прийому лікаря: доступний час прийому позначено зеленим кольором, виділений синім – це виділений користувачем час, а червоний – це зайнятий час, коли немає можливості записатися на прийом;
- кнопки відміни та підтвердження запису.

1.2.3 Система «Поліклініка без черг»

Поліклініка без черг – інформаційна система управління потоком пацієнтів та автоматизації бізнес-процесів в лікувальних закладах. Систему відрізняє від інших зручний інтерфейс та легкість сприйняття лікарями та пацієнтами. Система суттєво автоматизує модуль реєстратури, поліпшує роботу персоналі лікарні, підвищує лояльність пацієнтів до використання ІСМЗ для запису на прийом, підтримки контакту з лікарем, здійснює моніторинг і керує потоком пацієнтів та завантаженістю медичних працівників. Інтегрується з системами і модулями для роботи з електронною медичною картою пацієнта, електронним рецептом і обліковими системами [9].

Для ознайомлення розглянуто інтерфейс запису до лікаря з демонстрацією розкладу медичного працівника у системі «Поліклініка без черг», мобільна розгортка, на рисунку 1.5 наведено:

- персональна інформація медичного працівника лікарні: прізвище, ім'я та по батькові;

- номер кабінету лікаря;
- фотографія профілю лікаря;
- меню вибору послуги;
- найближчі дні доступні для запису на прийом;
- час, доступний для запису на прийом.

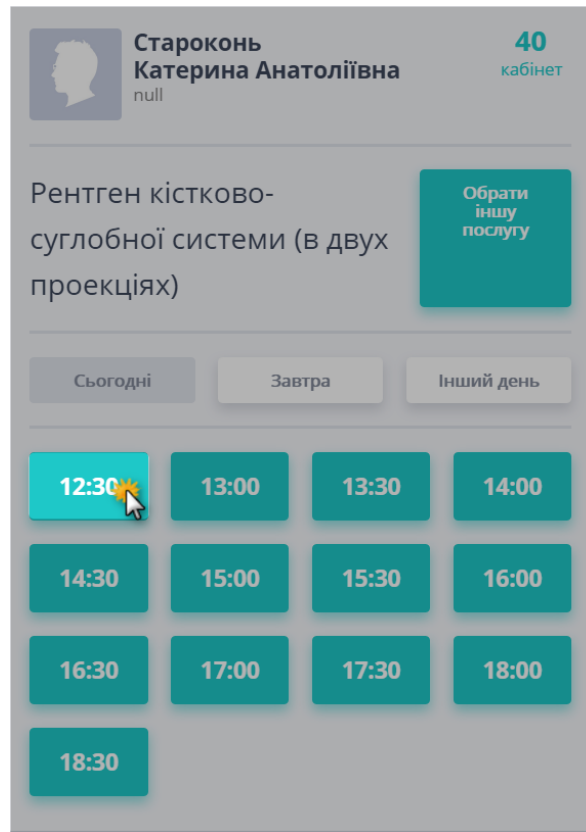


Рисунок 1.5 – Приклад розкладу лікаря в системі «Поліклініка без черг»

Проблеми існуючих ІСМЗ:

- необхідність лікаря вручну вводити дані до системи, що уповільнює процес документації;
- проблеми інтеграції ІСМЗ у системи медичних закладів;
- повільна обробка даних;
- відсутність авто генерації розкладу лікаря;
- неможливість користувача отримати дані у своєму обліковому записі.

Розглянуті медичні системи мають схожий базовий функціонал але відрізняються складністю структури: у системі «Поліклініка без черг»

реалізовано менший обсяг бізнес-процесів порівняно з системами Helsi та Medcard24, не надто зручний інтерфейс та обмежений функціонал, а розклад до лікаря можна подивитися тільки на найближчі дні, а не на місяць. Недоліком Helsi є відсутність пробного доступу до функцій системи. Недоліками Medcard24 є відсутність багатofакторної авторизації, підтримки повідомлення клієнтів.

У перелічених системах основною реалізованою функцією є запис до лікаря з можливістю переглянути його розклад на певний проміжок часу, з урахуванням тривалості робочого дня, вихідних та відпусток. Отже для кожного лікаря персонально повинен бути сформований розклад з урахуванням усіх факторів, а саме:

- час початку та кінця робочого дня;
- робочий конкретного лікаря;
- діюча номенклатура лікарських посад та типових штатів відповідних міжгалузевих та галузевих норм;
- нормативи праці (для окремих посад фахівців та робітників).

1.3 Аналіз сучасного стану досліджень з теми

Історія та сучасний стан проблеми був описаний у роботах [7-9]. У джерелах зачіпаються різні аспекти застосування інформаційних технологій у охороні здоров'я. Формулюються базові поняття в галузі медичного програмного забезпечення і, зокрема, медичних інформаційних систем (ІСМЗ), розглядаються основні функції та властивості цього класу систем, а також технологічні вимоги, яким мають відповідати сучасні ІСМЗ, та перспективи їх розвитку.

Для проведення дослідження необхідно вивчити існуючі види підходів, технологій та методів досліджуваної предметній області. Також необхідно вивчити поняття медичної інформаційної системи та принципи її роботи та функціонування, вплив на роботу медичного закладу, якість обслуговування пацієнтів. Методи запису, зберігання та обробки даних лікарів, пацієнтів, діагнозів та результатів медичних досліджень пацієнтів.

Деякі питання розвитку теорії розкладів розглянути в роботі [1]. Автори застосовують теорію розкладів при плануванні залізничних перевезень.

Класифікація та опис структурних компонентів та підсистем та функцій ІСМЗ [3-5]

Завдання складання розкладу для працівників підприємства відноситься до класу слабоформалізованих задач оптимізації. У її описі доводиться враховувати велику кількість параметрів та обмежень. Не зважаючи на велику кількість робіт, присвячених цьому завданню, оптимального рішення для автоматизації процесу складання розкладу досі не знайдено.

Існує багато постановок цього завдання, що відрізняються суворістю форматизації, запропоновані різні методи та алгоритми рішення [11-13, 16]. У кожного з них є свої переваги та недоліки, проте загальновизнаних і незаперечних моделей і методів немає.

Аналіз джерел доводить що автоматизована генерація розкладів для підприємств різних видів направленостей у різних ПО є актуальною темою, а для реалізації цієї функції для медичної інформаційної системи, ще не використовувався генетичний алгоритм, тому розробка та дослідження його ефективності є актуальними та мають наукову новизну.

1.4 Постановка задачі дослідження

Завданням є розробка та інтеграція компоненту автоматизованої генерації розкладу лікаря медичного закладу з метою зменшення витрат часу та підвищення ефективності. Задача складання розкладу працівника будь-якого підприємства у загальному вигляді полягає в розподілу заданого кінцевого набору подій у часі при заданих обмеженнях ресурсів та інших обмеженнях.

Вихідні дані для складання робочого розкладу лікарів медичного закладу складають:

- перелік лікарів що працюють у закладі;
- інформація про відпустки та лікарняні дні;
- робочий час медичного персоналу.

Обмеженнями під час створення розкладу є:

- час початку та час кінця робочого дня спеціаліста;
- кількість робочих днів на тиждень;
- максимальна кількість відвідувань.

Необхідно скласти розклад прийомів для кожного лікаря на кожен день тижня, розподілити кабінети на кожного лікаря, розподілити навантаження на лікарів на кожен день.

Сформовано перелік вимог до розкладу, що буде вважатися припустимим:

- кабінети лікарів можуть бути доступними та недоступними для відвідування в певний час;
- для кожної спеціальності лікаря є перелік допустимих та прийнятних кабінетів;
- доступність лікарів – лікар має робочий графік, що відображає присутність або відсутність лікаря на його робочому місці;
- у кабінеті лікаря під час медичної діагностики або лікувального процесу може перебувати лише один пацієнт у рамках відведеного для відвідування часу згідно розкладу.

Необхідно розробити вимоги до компоненту ІСМЗ для автоматичної генерації розкладу лікаря, створити функціональну модель, модель варіантів використання, діаграму класів, реалізувати та інтегрувати компонент у ІСМЗ, провести оцінку ефективності розробленого компоненту за критеріями: часу, що витрачено на створення розкладу та критерієм відносної кількості записів до лікарів за допомогою сайту ІСМЗ.

У результаті проведеної роботи та дослідження отримаємо працюючу компоненту для автоматичної генерації розкладу медичного лікаря та оцінку ефективності роботи розробленого компоненту.

2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ РОЗКЛАДУ

Найбільш використовувані методи генерації розкладів роботи працівників на підприємствах:

- табличний метод, що виконується вручну за допомогою допоміжних програм для роботи з електронними таблицями, (розклад складають у табличній формі в якій вказують лікарів, а також час початку та кінця робочого дня);
- автоматичне формування розкладу лікаря методом задоволення обмежень;
- генетичний алгоритм;
- мурашиний алгоритм;
- метод гілок та меж.

2.1 Табличний метод створення розкладу

Табличний метод є основним і застосовується на практиці найбільш широко при великій кількості лікарів. Розклад складають у табличній формі, у рядках якої наводять дані, що відповідають різним номерам кабінетів, а в стовпцях час початку та кінця робочого дня. Недоліками табличного методу складання розкладів є ненаочність відстеження інтервалів роботи лікаря. Застосування методу вимагає від укладача розкладу досвіду та навичок у роботі. Середня трудомісткість складання одного розкладу залежить від числа працюючих лікарів і коливається від 1 до 6 робочих днів. До переваг методу слід віднести наочність у роботі кожного кабінету протягом доби.

2.2 Автоматична генерація розкладу методом задоволення обмежень

Складання розкладу можна розглядати як задачу задоволення обмежень, на вирішення якої розроблено безліч алгоритмів. Виникло ціле напрям у програмуванні – програмування в обмеженнях (constraint programming). Програмування в обмеженнях тісно пов'язане із традиційним логічним програмуванням, у межах якого і сформувався. Автоматична генерація розкладу

методом задоволення обмежень являє собою вирішення комбінаторної задачі, яка включає в себе:

- змінні;
- набір можливих значень;
- список обмежень.

Більшість систем програмування в обмеженнях представляють собою звичайний інтерпретатор прологу з вбудованим механізмом розв'язання певного класу задач задоволення обмежень.

Програмування у таких системах називають логічним програмуванням в обмеженнях (Constraint Logic Programming або CLP). Правил всього два: для кожної змінної задається набір можливих значень - змінним можуть бути присвоєні будь-які значення, а не лише 1 або 0, також маємо список обмежень, яким задовольняють вхідні змінні. Вирішенням завдання задоволення обмежень буде знаходження всіх можливих значень, які можуть набувати змінних, з урахуванням існуючих обмежень. Основна ідея вирішення завдань така: програміст визначає деяку множину змінних x_1, \dots, x_n , та області їх значень

X_1, \dots, X_n , описує додаткові обмеження, яким мають задовольняти змінні, а система знаходить відповідні значення змінних, що задовольняють одночасно всім заданим обмеженням. Для ілюстрації наведемо невеликий приклад, що стосується досліджуваної предметної області. Розглянемо вуз, у якому працює M викладачів, у навчальному плані зафіксовано N предметів, аудиторний фонд складається із L аудиторій. Визначимо множину $P = 1, 2, \dots, D$, елементами якої є навчальні пари у вузі протягом тижня, а D - число всіх періодів навчання протягом тижня. Позначимо y_{ij} - період навчання, i - викладач j - предмет що викладається,

$y_{ij} \in P$. Тоді обмеження «кожний викладач у кожний момент часу може вести не більше одного заняття» набуде вигляду:

$$y_{ij} \neq y_{i'j'}, 1 \leq i \leq M, 1 \leq j \leq N, 1 \leq j' \leq N, j \neq j'$$

Нехай z_i - аудиторія, де проводить заняття i - викладач, $1 \leq z_i \leq L$. Тоді

обмеження «у кожній аудиторії у кожний момент часу може проводитися не більше одного заняття» набуде вигляду:

$$z_i \neq z_{i'}, 1 \leq i \leq M, 1 \leq i' \leq M, i \neq i'$$

Аналогічно складаються інші обмеження. В результаті роботи алгоритму буде отримано безліч значень кожної змінної, що задовольняють заданим обмеженням. При цьому область визначення змінних, що у строгих обмеженнях, може суттєво скоротитися або навіть містити єдине значення. Основна перевага, що отримується при використанні CLP, - це скорочення області пошуку, що досягається не шляхом оцінки кожного варіанту розкладу, а за рахунок того, що система сама виключає з розгляду безнадійні варіанти.

2.3 Генетичний алгоритм

Генетичні алгоритми дозволяють розв'язувати широке коло задач оптимізації шляхом випадкового підбору, комбінування та зміни параметрів моделювання способами, що подібні до біологічної еволюції. Генетичні алгоритми оперують сукупністю особин, які кодують один із можливих розв'язків задачі. Цим генетичний алгоритм відрізняється від більшості інших алгоритмів оптимізації, які оперують лише з одним розв'язком, який покращує його. Принцип роботи генетичного алгоритму: за допомогою функції пристосованості серед всіх особин популяції виділяють:

- найбільш пристосовані, які отримують можливість схрещуватися і давати нащадків;
- найгірші, які видаляються з популяції і не дають нащадків.

Отже, пристосованість нового покоління в середньому вище за попереднє.

Існує багато модифікацій генетичних алгоритмів. Але всім їм властива універсальність. Тобто від задачі, яку необхідно розв'язати, залежить визначення функції пристосованості й спосіб кодування рішень. Решта же етапів генетичних алгоритмів виконується однаково для будь-яких задач. Кожен крок алгоритму складається з трьох стадій:

- генерація проміжної популяції шляхом відбору поточного покоління;

- схрещування особин проміжної популяції шляхом кросовера, у результаті чого формується нове покоління;
- мутація нового покоління.

Мутація – генетичний оператор, який певною мірою змінює один або кілька «генів» у випадкових позиціях «хромосоми». Нащадки, що зазнали впливу генетичних операторів, утворюють нову популяцію – і в ній починається чергова ітерація генетичного алгоритму. Знову йде підрахунок функції придатності, відбувається природний відбір, а далі алгоритм або зупиниться, якщо задана умова виконана, або знову перейде до селекції. Тепер подивимося на переваги та недоліки цього методу.

Індивідуум (генетичний код, особина) - набір хромосом (варіант розв'язання задачі). Зазвичай особина складається з однієї хромосоми, тому надалі особина та хромосома ідентичні поняття.

Відстань – хеммінгова відстань між бінарними хромосомами.

Кроссинговер (кросовер) – операція схрещування, при якій дві хромосоми обмінюються своїми частинами.

Мутація – випадкова зміна однієї чи кількох позицій у хромосомі.

Інверсія – зміна порядку бітів у хромосомі або в її фрагмент.

Населення - сукупність індивідуумів.

Проміжна популяція – популяція, що формується з особин, які отримали право давати нащадків. Найбільш пристосовані особини можуть бути записані туди кілька разів, найменш пристосовані з великою ймовірністю туди не потраплять. Існує кілька способів реалізації цього відбору:

- *stochastic sampling*. Особини розташовуються на колесі рулетки так, що розмір сектора кожної особини пропорційний її пристосованості. N раз запускаючи рулетку, вибирається необхідна кількість особин для запису в проміжну популяцію;

- *remainder stochastic sampling*. Для кожної особини обчислюється відношення її пристосованості до середньої пристосованості популяції. Ціла частина цього відношення вказує, скільки разів потрібно записати особину в проміжну

популяцію, а дробова показує її ймовірність потрапити туди ще раз. Після відбору особини проміжної популяції випадковим чином розбиваються на пари, потім із деякою ймовірністю схрещуються, в результаті чого виходять два нащадка, які записуються до нового покоління, або не схрещуються, тоді в нове покоління записується сама пара.

У класичному генетичному алгоритмі застосовується одноточковий оператор кросовера: для батьківських «генотипів» випадковим чином вибирається точка розділу, нащадки формуються шляхом обміну відсіченими частинами. До отриманого в результаті відбору і схрещування нового покоління застосовується оператор мутації. При цьому кожен біт «генотипу» кожної особини популяції з деякою, досить малою ймовірністю, інвертується.

Такий процес еволюції може тривати до безкінечності. Критерієм зупинення може служити задана кількість поколінь або сходження популяції.

Класичний генетичний алгоритм складається з наступних етапів:

- ініціалізація, або вибір вихідної популяції хромосом;
- обчислення пристосованості хромосом у популяції;
- перевірка умов зупинення алгоритму;
- селекція хромосом(вибір батьків);
- схрещування;
- мутація;
- вибір «найкращої» хромосоми.

Ініціалізація, тобто формування вихідної популяції, полягає у випадковому виборі заданої кількості хромосом (особин), що представляються двійковими послідовностями фіксованої довжини.

Оцінювання пристосованості хромосом у популяції полягає у розрахунку функції пристосованості кожної хромосоми цієї популяції. Що більше значення цієї функції, то вище «якість» хромосоми. Форма функції пристосованості залежить від характеру завдання, що розв'язується. Передбачається, що функція пристосованості завжди набуває негативних значень і, крім того, що для вирішення оптимізаційної задачі потрібно максимізувати цю функцію. Якщо

вихідна форма функції пристосованості не задовольняє цим умовам, виконується відповідне перетворення (наприклад, задачу мінімізації функції можна легко звести до завдання максимізації).

Перевірка умови зупинення алгоритму. Визначення умови зупинки генетичного алгоритму залежить від його конкретного застосування. В оптимізаційних завданнях, якщо відомо максимальне або мінімальне значення функції пристосованості, то зупинка алгоритму може статися після очікуваного оптимального значення, можливо з заданою точністю. Зупинка алгоритму може статися у разі, коли його виконання не призводить до поліпшення вже досягнутого значення. Алгоритм може бути зупинений після закінчення певного часу виконання або після виконання заданої кількості ітерацій. Якщо умову зупинки виконано, відбувається перехід до завершального етапу вибору «найкращої» хромосоми. Інакше на наступному етапі виконується селекція.

Селекція хромосом – вибір (за розрахованими на другому етапі значенням функції пристосованості) тих хромосом, які братимуть участь у створенні нащадків для наступної популяції, тобто. для наступного покоління. Такий вибір проводиться відповідно до принципу природного відбору, яким найбільші шанси участь у створенні нових особин мають хромосоми з найбільшими значеннями функції пристосованості. Існують різні способи селекції. Найбільш популярним вважається так званий метод рулетки (roulette wheel selection). Кожній хромосомі можна порівняти сектор колеса рулетки, величина якого встановлюється пропорційною значенню функції пристосованості даної хромосоми. Тому що чим більше значення функції пристосованості, тим більше сектор на колесі рулетки. Все колесо рулетки відповідає сумі значень функції пристосованості всіх хромосом популяції, що розглядається. Кожній хромосомі, що позначається ch_i для $i = 1, 2, \dots, N$ (де N позначає чисельність популяції) відповідає сектор колеса $v(ch_i)$, виражений у відсотках згідно з формулою $v(ch_i) = p_s(ch_i) \times 100\%$, де

$$p_s(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)},$$

де $F(ch_i)$ – значення функції пристосованості хромосоми ch_i ;

$p_s(ch_i)$ – ймовірність селекції хромосоми ch_i .

В результаті процесу селекції створюється батьківська популяція, також звана батьківським пулом (mating pool) з чисельністю N , що дорівнює чисельності поточної популяції.

Потім на першому етапі схрещування вибираються пари хромосом із батьківської популяції. Це тимчасова популяція, що складається з хромосом, відібраних у результаті селекції та призначених для подальших перетворень з метою формування нової популяції нащадків.

Вибір "найкращої" хромосоми. Якщо умову зупинки алгоритму виконано, слід вивести результат роботи, тобто. уявити розв'язане завдання. Найкращим рішенням вважається хромосома з найбільшим значенням функції пристосованості. Для наочної демонстрації алгоритму наведена його схема на рисунку 2.1.

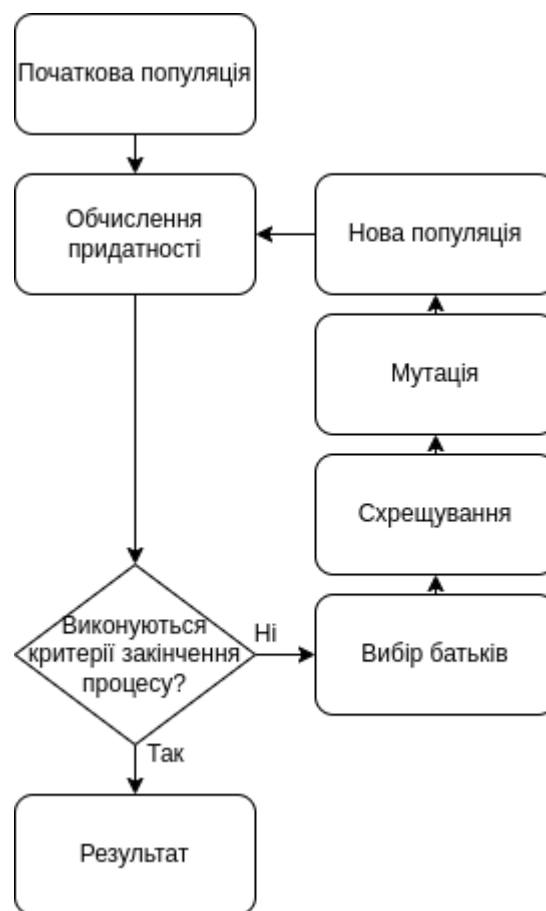


Рисунок 2.1 - Схема генетичного алгоритму

2.4 Мурашиний алгоритм

Мурашиний алгоритм – один із ефективних поліноміальних алгоритмів для знаходження наближених рішень задачі комівояжера, а також аналогічних завдань пошуку маршрутів на графах. Суть підходу полягає в використанні моделі поведінки мурах, що шукають шляхи від колонії до джерела живлення.

В основі алгоритму лежить поведінка мурашиної колонії – маркування більш вдалих шляхів великою кількістю феромону.

Мураха - це програмний агент, який є частиною колонії і використовується для вирішення проблеми. Мураха забезпечується набором простих правил, які дозволяють йому вибирати шлях у графі. Він підтримує список вузлів, які вже відвідав. Таким чином, мураха повинна проходити через кожен вузол лише один раз. Шлях між двома вузлами графа, яким мураха відвідав кожен вузол лише один раз, називається шляхом Гамільтона.

Феромон – речовина-маркер, яка залишається на шляху після проходження по ньому мурахи.

Робота починається з розміщення мурах у вершинах графа, потім здійснюється рух мурах. Напрямок руху визначається ймовірнісним методом. Дві мурахи з мурашника повинні дістатися до цілі, яка перебуває за перешкодою. Під час переміщення кожна мураха виділяє трохи феромону, використовуючи його як маркер. Кожна мураха вибере свій шлях. Перша мураха вибирає більший шлях, а друга - менший. Нижній шлях вдвічі коротший за верхній, друга мурашка досягне мети за час t_1 . Перша мураха в цей момент пройде лише половину шляху.

Коли одна мураха досягає їжі, вона бере один з об'єктів і повертається до мурашника тим же шляхом. За час t_2 друга мураха повернулася до мурашника з їжею, а перша мурашка тільки досягла їжі за цей час.

При переміщенні кожної мурахи на шляху залишається трохи феромону. Для першого мурахи за час $t_0 t_2$ шлях був покритий феромон тільки один раз. У той же час друга мураха покрила шлях феромонами двічі. За час t_4 перша мураха

повернулася в мурашник, а друга мураха вже встигла ще раз сходити до їжі і повернутися. При цьому концентрація феромону на нижньому коротшому шляху буде вдвічі вищою, ніж на верхньому. Тому перша мураха наступного разу вибере нижній шлях, оскільки там концентрація феромону вища. У цьому полягає базова ідея мурашиного алгоритму - оптимізація шляхом непрямої зв'язку між автономними агентами.

Припустимо, що довкілля для мурах є повний неорієнтований граф наведений на рисунку 2.1, кожне ребро якого має вагу, яка позначається як відстань між двома вершинами, з'єднаними ним. Граф двонаправлений, тому мураха може подорожувати за межею у будь-якому напрямку.

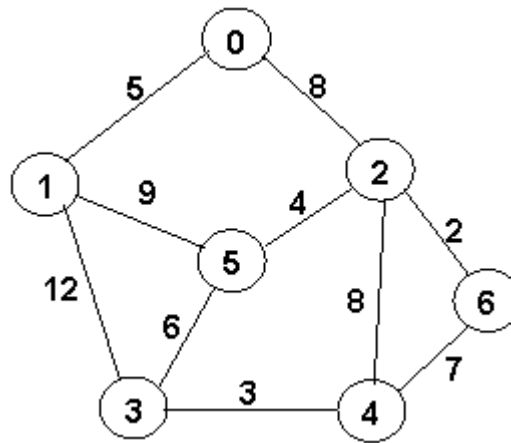


Рисунок 2.2 – Двонаправлений граф з вагами ребер

Імовірність включення ребра в маршрут окремої мурахи пропорційна кількості феромонів на цьому ребрі, а кількість феромона, що відкладається, пропорційно довжині маршруту. Чим коротший маршрут тим більше феромона буде відкладено на його ребрах, отже, більша кількість мурах включатиме його в синтез власних маршрутів. Моделювання такого підходу, що використовує лише позитивний зворотний зв'язок, призводить до передчасної збіжності – більшість мурах рухається локально-оптимальним маршрутом. Уникнути цього можна моделюючи негативно зворотний зв'язок у вигляді випаровування феромону. Причому, якщо феромон випаровується швидко, це призводить до втрати пам'яті

колонії і забування правильних рішень, з іншого боку, великий час випарів може призвести до отримання стійкого локального оптимального рішення.

Вузли у списку поточного шляху розташовуються в тому порядку, в якому мураха відвідувала їх. Пізніше список використовується визначення протяжності шляху між вузлами. Справжня мураха під час переміщення по дорозі залишатиме за собою феромони. В алгоритмі мурашки агент залишає феромони на ребрах графа після подорожі.

Стартова точка, куди розміщується мураха, залежить від обмежень, що накладаються умовами задачі, тому що для кожного завдання спосіб розміщення мурах є визначальним. Або всі вони поміщаються в одну точку, або в різні з повтореннями або без повторень.

На етапі розміщення задається початковий рівень феромона. Він ініціалізується невеликим позитивним числом для того, щоб на початковому етапі ймовірності переходу в наступну вершину не були нульовими. Після створення населення мурах порівну розподіляється по вузлах мережі. Необхідний однаковий поділ мурах між вузлами, щоб усі вузли мали однакові шанси стати відправною точкою. Якщо всі мурахи почнуть рух із однієї точки, це означатиме, що ця точка є оптимальною для старту, але насправді ми цього не знаємо.

Рух мурашки ґрунтується на одному імовірнісному рівнянні. Якщо мураха ще не закінчила шлях, тобто не відвідала всі вузли мережі, для визначення наступної грані шляху використовується рівняння:

$$P = \frac{\tau(r,u)^a \times \eta(r,u)^\beta}{\sum_k \tau(r,u)^a \times \eta(r,u)^\beta},$$

де $t(r, u)$ – інтенсивність ферменту на межі між вузлами r і u ;

$h(r, u)$ – функція, яка представляє вимірювання зворотної відстані для грані;

a – вага ферменту;

β - коефіцієнт евристики.

Параметри a та β визначають відносну значущість двох параметрів, а також їх вплив на рівняння. Згадайте, що мурашка подорожує лише вузлами, які ще були

відвідані. Тому можливість розраховується тільки для граней, які ведуть до ще не відвіданих вузлів. Змінна k представляє грані, які ще були відвідані.

Пройдений мурахою шлях відображається, коли мураха відвідає всі вузли діаграми. Зверніть увагу, що цикли заборонені, оскільки алгоритм містить список табу. Після завершення довжина шляху може бути підрахована - вона дорівнює сумі всіх граней, якими подорожувала мурашка. Рівняння показує кількість ферменту, який залишився на кожній грані шляху для мурашки k . Змінна Q є константою.

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L^k(t)}$$

Результат рівняння є засобом вимірювання шляху - короткий шлях характеризується високою концентрацією ферменту, а більш довгий шлях - нижчою. Потім отриманий результат використовується у рівнянні 2.3, щоб збільшити кількість ферменту вздовж кожної грані пройденого мурахою шляху.

$$\tau_{ij}(t) = \Delta\tau_{ij}(t) + (\tau_{ij}^R(t) \times p)(2.3),$$

де p – константа зі значення між 0 та 1.

Це рівняння застосовується до всього шляху, кожна грань позначається ферментом пропорційно довжині шляху. Тому слід дочекатися, поки мурашка закінчить подорож і потім оновити рівні ферменту, інакше справжня довжина шляху залишиться невідомою.

На початку шляху кожна грань має шанс бути обраною. Щоб поступово видалити грані, що входять у найгірші шляхи в мережі, до всіх граней застосовується процедура випаровування ферменту. Використовуючи константу p з рівняння 2.3 ми отримуємо рівняння 2.4.

$$\tau_{ij}(t) = \tau_{ij}(t) \times (1 - p)(2.4).$$

Отже для випаровування ферменту використовується зворотний коефіцієнт поновлення шляху. Після того як шлях мурахи завершено, грані оновлені відповідно до довжини шляху і відбулося випаровування ферменту на всіх гранях, алгоритм запускається повторно. Список табу видаляється, і довжина колії

становиться нульовою. Мурахам дозволяється ходити по мережі, ґрунтуючи вибір межі на рівнянні 2.1. Цей процес може виконуватися для постійної кількості шляхів або до моменту, коли протягом кількох запусків не було зафіксовано повторних змін. Потім визначається найкращий шлях, який є рішенням.

Мурашиний алгоритм, незалежно від модифікацій, складається з наступних етапів, які виконуються циклічно до виконання умов виходу:

- створення мурах;
- пошук розв'язку;
- оновлення феромону.

На рисунку 2.3 поданий узагальнений алгоритм мурашиної колонії:

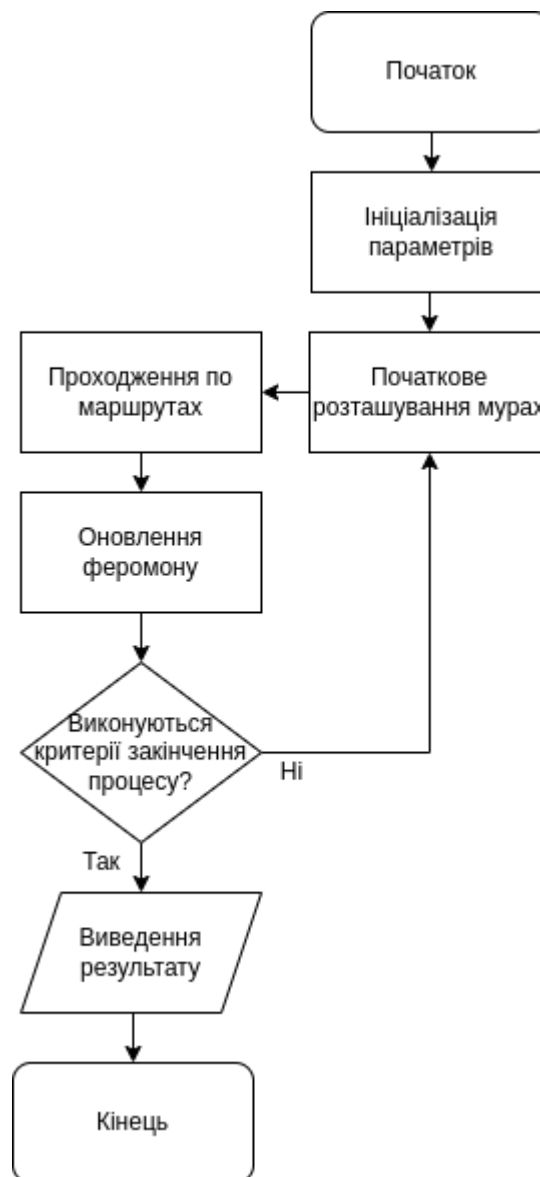


Рисунок 2.3 - Схема узагальненого мурашиного алгоритму

2.5 Метод гілок та меж

Метод гілок та меж є загальним алгоритмічним методом вирішення різноманітних оптимізаційних задач. Він широко застосовується для таких задач, як задача комівояжера. Цей метод є варіацією повного перебору, при якому відкидаються підмножини допустимих рішень, що не містять оптимальних значень цільової функції. В методі гілок та меж використовуються дві процедури: розгалуження та знаходження оцінок.

За допомогою процедури розгалуження множина допустимих рішень на кожному етапі розділяється на підмножини меншої розмірності. Ці множини стають вузлами дерева пошуку.

Процедура знаходження оцінок визначає верхні та нижні межі оптимального значення на підмножинах допустимих рішень. Якщо для знайденого вузла дерева пошуку верхня межа співпадає з нижньою, то це значення цільової функції є оптимальним і досягається на відповідній підмножині допустимих рішень.

Алгоритм, що реалізує метод гілок та меж, складається з послідовності однотипних кроків. На кожному кроці алгоритм починає роботу з перевірки елементів розбиття. Ці множини нумеруються числами від 1 до L , і починається новий крок алгоритму. При розв'язку конкретної задачі слід визначити способи побудови нижньої і верхньої оцінок, метод розгалуження, а також правило вибору перспективної множини для розбиття.

В основі методу гілок та кордонів лежить ідея послідовного розбиття безлічі допустимих рішень на підмножини. На кожному кроці методу елементи розбиття піддаються перевірці для з'ясування, містить дане підмножина оптимальне рішення чи ні. Перевірка здійснюється за допомогою обчислення оцінки знизу для цільової функції на даному підмножині. Якщо оцінка знизу не менша за рекорд – найкращий зі знайдених рішень, то підмножина може бути відкинута. Перевірюване підмножина може бути відкинута ще й у тому випадку, коли в ньому вдається знайти найкраще рішення. Якщо значення цільової функції на

знайденому рішенні менше за рекорд, то змінюється рекорд. Після закінчення алгоритму рекорд є результатом його роботи.

Якщо вдається відкинути всі елементи розбиття, то рекорд оптимальне рішення завдання. В іншому випадку, з невідкинутих підмножин вибирається найбільш перспективне (наприклад, з найменшим значенням нижньої оцінки), і воно розбивається. Нові підмножини знову піддаються перевірці тощо.

Обчислення нижньої межі є найважливішим елементом цієї схеми. Для найпростішого завдання розміщення один із способів її побудови полягає в наступному. Запишемо вихідне завдання у термінах цілісного лінійного програмування. Введемо такі змінні: y_i , дорівнює 1, якщо підприємство i обслуговує клієнта i , інакше дорівнює 0; x_{ij} , що дорівнює 1, якщо підприємство i відкрито, інакше дорівнює 0.

З використанням введених позначень найпростіше завдання розміщення записується наступним способом:

$$\min \left\{ \sum_{i \in I} c_i y_i + \sum_{i \in I} \sum_{j \in J} g_{ij} x_{ij} \right\}$$

$$\sum_{i \in I} x_{ij} = 1, j \in J$$

Подвійне завдання лінійного програмування має вигляд:

$$\max \sum_{j \in J} v_j = 1$$

$$\sum_{j \in J} w_{ij} \leq c_i^0, i \in I$$

Наближене рішення двоїстої задачі використовується як нижня оцінка.

Для скорочення розмірності завдання застосовується так званий блок попереднього відбраковування. Він заснований на застосуванні умов доповнюючої нежорсткості для лінійного програмування.

$$y_i \left(c_i - \sum_{j \in J} w_{ij} \right) = 0, i \in I$$

Якщо для оптимального розв'язання двоїстої задачі вираз у дужках позитивний для деякого $i \in I$, то скоріш за все у вихідній цілісній задачі $y_i = 0$, і розмірність можна зменшити. Зрозуміло, що цей евристичний прийом не завжди призводить до правильного рішення. Тому як поріг краще брати не 0, а деяку величину $\delta \geq 0$, вибір якої залежить від вихідних даних. Цю величину називають порогом відбраковування. Схема цього алгоритму подана на рисунку 2.4.

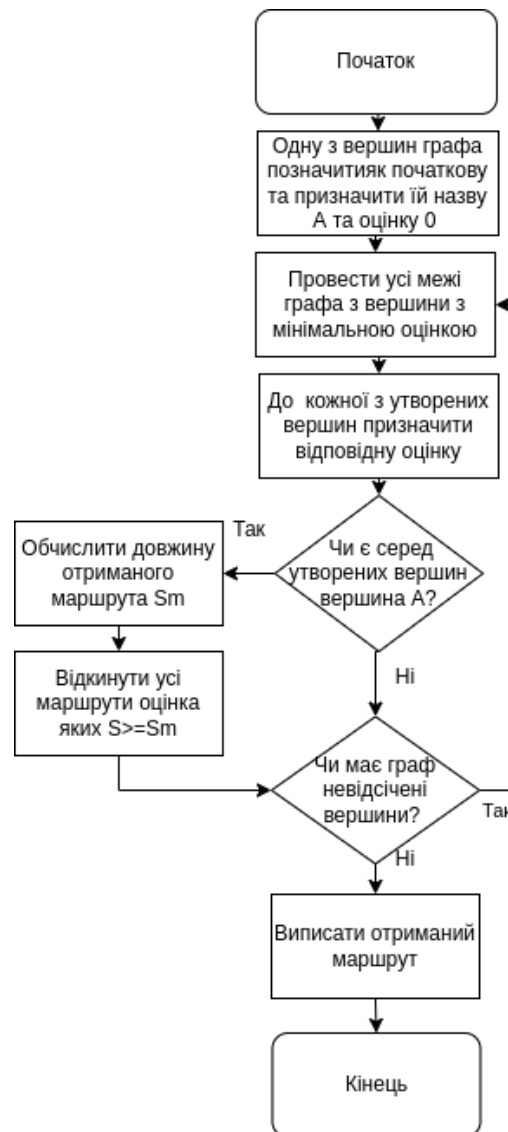


Рисунок 2.4 - Схема алгоритму гілок та меж

2.6 Вибір алгоритму для проектування та розробки компоненту генерації розкладу

Кожен з наведених методів забезпечує створення робочого розкладу медичних працівників, що використовується для забезпечення автоматизації

процесу запису до лікаря у ІСМЗ та дозволяє пацієнтам обирати конкретний час для відвідування спеціалістів, що в свою чергу дозволяє уникнути зайвих витрат часу при лікуванні та підвищує його якість. Але все ж кожен алгоритм має свої переваги та недоліки.

Недолік табличного методу у великих витратах часу що є неефективним та ризикованим через можливість виникнення помилок у процесі створення розкладу таким методом.

Головний недолік алгоритму методу гілок і меж полягає у необхідності повністю вирішувати задачі лінійного програмування, асоційовані з кожною з вершин багатогранника допустимих рішень. Для завдань великої розмірності це вимагає значних і певною мірою невиправданих з практичної точки зору витрат часу.

Недоліки мурашиного алгоритму полягають у складності теоретичного аналізу через зміну вірогідностей при ітераціях. Вирішення гарантується, але час збіжності не визначено. Зазвичай необхідно застосування додаткові методи, такі як локальний пошук. Сильно залежить від налаштованих параметрів, які підбираються лише виходячи з експериментів.

Для роботи був обраний генетичний алгоритм через його переваги такі як:

- концептуальна простота: основними кроками алгоритму є: ініціалізація, оцінка якості рішення за допомогою, ітеративна зміна популяції шляхом відбору особин та застосування генетичних операторів;
- при реалізації процедури пошуку генетичний алгоритм обробляє одночасно кілька варіантів рішення, а не переходить послідовно від варіанту до варіанту, як у традиційних методах;
- у процесі роботи генетичного алгоритму не використовують додаткової інформації крім даних про область допустимих значень параметрів та цільової функції у довільній точці, що підвищує швидкість їх роботи.
- для породження нових варіантів рішення одночасно генетичний алгоритм використовує як ймовірнісні, так і детерміновані правила, що дає значно більший ефект, ніж кожен із цих методів окремо.

3 РОЗРОБКА ВИМОГ ТА ПРОЕКТУВАННЯ КОМПОНЕНТУ ГЕНЕРАЦІЇ РОЗКЛАДУ

3.1 Проектування функціоналу компоненту автоматичного створення розкладу

Розроблений компонент ІСМЗ призначена для автоматизації генерації розкладу лікарів, що використовується для реалізації функції запису пацієнту до лікаря. Отже дані необхідні для медичного обслуговування, ті, що необхідні для відвідування лікаря у медичному закладі будуть збиратися за допомогою веб сайту в мережі Інтернет та зберігатися в БД медичного закладу.

Визначимо вимоги до розробленого компоненту цієї ІСМЗ:

- серверна частина реалізована у вигляді API (Application Programming Interface);
- клієнтська частина повинна бути виконана окремим компонентом та підключатися до API;
- реляційна БД на основі системи управління базами даних(СУБД) PostgreSQL та відповідати потребам в зберіганні та обробці даних;
- трансляція даних між API та інтерфейсом користувача виконана у форматі JSON.

Створена функціональна модель у веб-сервісі draw.io з використанням стандарту IDEF0 для опису функцій ІСМЗ. IDEF0 стандарт використовується для побудови функціональних моделей с описом процесів розробленої системи. За стандартом моделлю вважається сукупність ієрархічно впорядкованих діаграм: концептуальної діаграми та її декомпозицій. Концептуальна діаграма є блоком, що має сторони: «Вхід», «Вихід», «Механізми», «Управління».

Стрілки функціональних блоків відображають елемент системи, що обробляється блоками або впливають на процеси, що ці блоки відображають, наприклад, об'єкти, інформація, люди, які приймають участь у процесі. Кожна стрілка підписується.

Процес – послідовність пов’язаних між собою дій, що перетворюють входи на виходи і сприяють отримання кінцевого результату, що буде корисним кінцевому користувачу.

Концептуальна діаграма на рисунку 3.1 є узагальненою та зосереджується тільки на зовнішньому оточенні системи, отже для детального опису функцій проводиться декомпозиція концептуальної діаграми або блоків декомпозицій вищих рівнів.

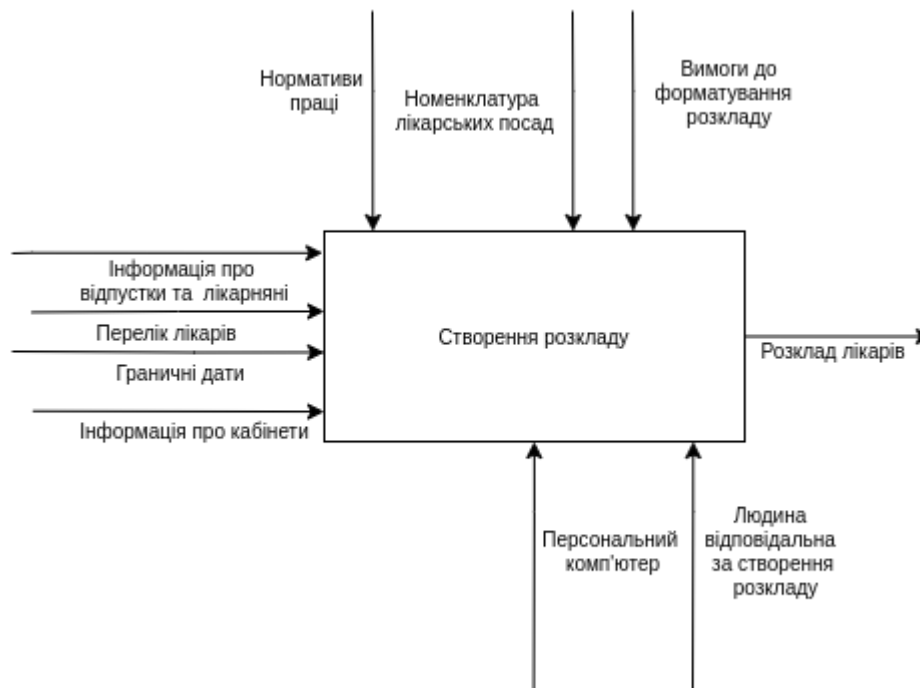


Рисунок 3.1 – Концептуальна діаграма функціональної моделі компоненту ІСМЗ

На вищезазначеній контекстній діаграмі створення розкладу табличним методом, як на діаграмі IDEF0 є виходи виходи, механізми та управління.

Вхідні дані процесу, які використовуються або перетворюється функціональним блоком для отримання результату:

- інформація про відпустки та лікарняні;
- граничні дати;
- перелік лікарів;
- перелік кабінетів.

Механізмами є ресурси, які виконують роботу у функціональному блоці та

визначають, відповідального за виконання функції. Механізмами концептуальної діаграми є: персональний комп'ютер працівника лікарні який є створює розклад, та сама людина.

Стрілками управління, як правилами або стандартами, що контролюють виконання процесу та входять у функціональний блок зверху є нормативи праці, номенклатура лікарських посад, та вимоги до форматування розкладу.

На виході отримуємо правильно відформатований розклад для лікарів на найближчий період часу.

Створена декомпозиція процесу створення розкладу лікарів та подана на рисунку 3.2:

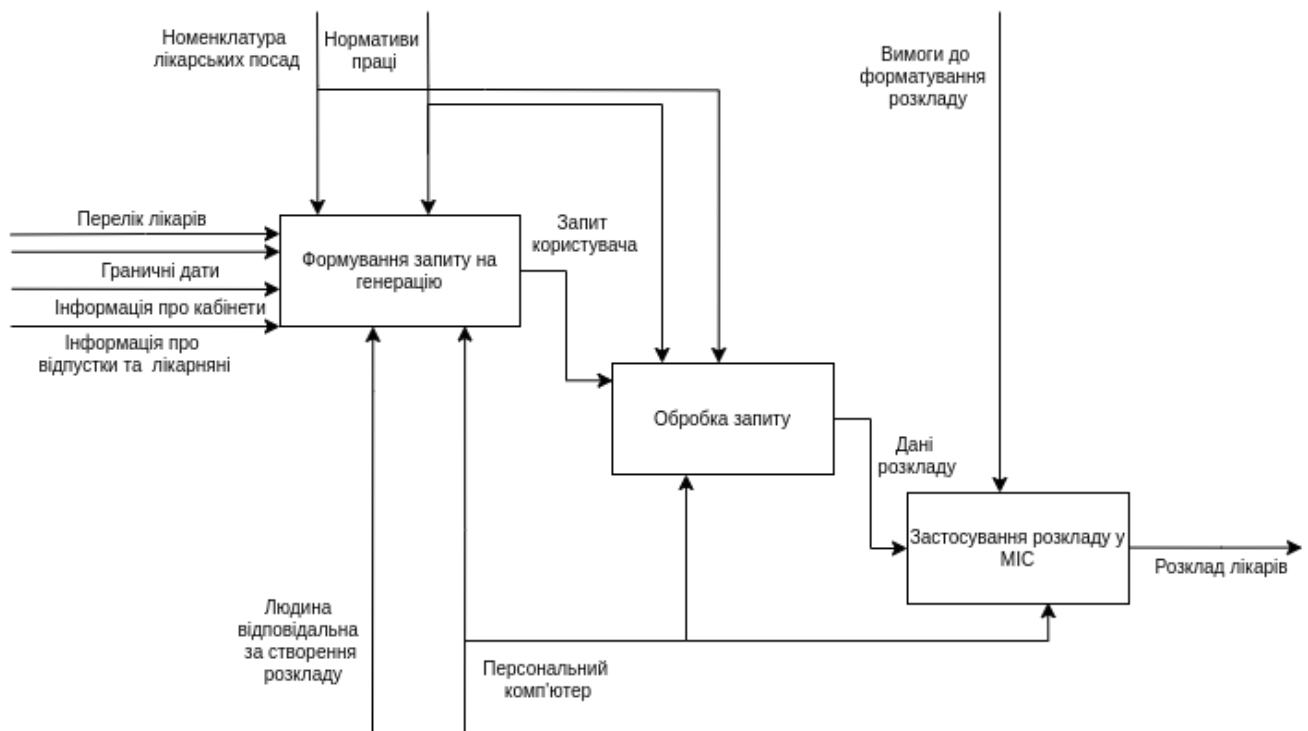


Рисунок 3.2 – Діаграма декомпозиції першого рівня

Основний блок створення розкладу при декомпозиції розкладається на три блоки: формування запиту на генерацію розкладу, обробку запиту та застосування розкладу у ІСМЗ.

З блоку формування запиту на виході отримуємо запит користувача, з блоку обробки запиту на виході отримуємо дані розкладу, з боку процесу застосування

розкладу у ІСМЗ на виході отримуємо сформований розклад лікарів, у вигляді у якому пацієнт його бачить на сайті.

Подальша декомпозиція процесів діаграми першого рівня декомпозиції подана на рисунках 3.3-3.5

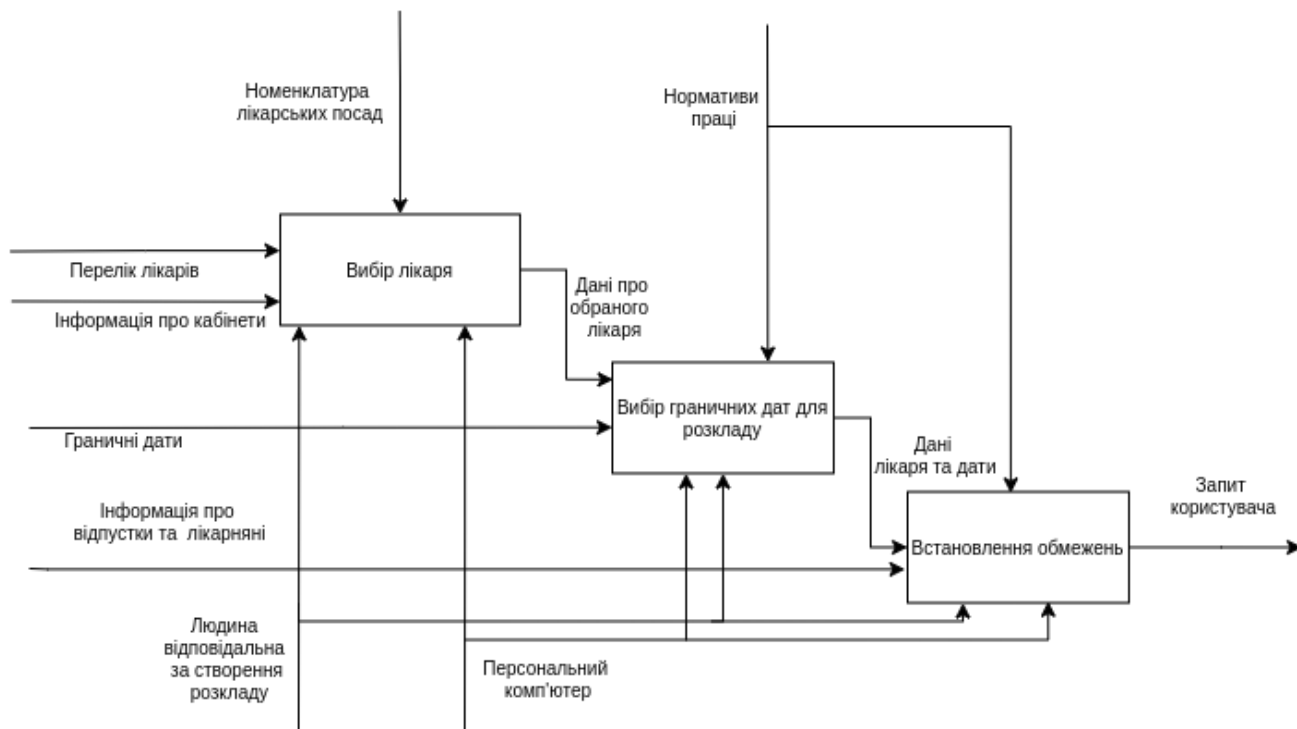


Рисунок 3.3 – Діаграма декомпозиції другого рівня для процесу «Формування запиту на генерацію»



Рисунок 3.4 – Діаграма декомпозиції другого рівня для процесу «Застосування розкладу у МІС»

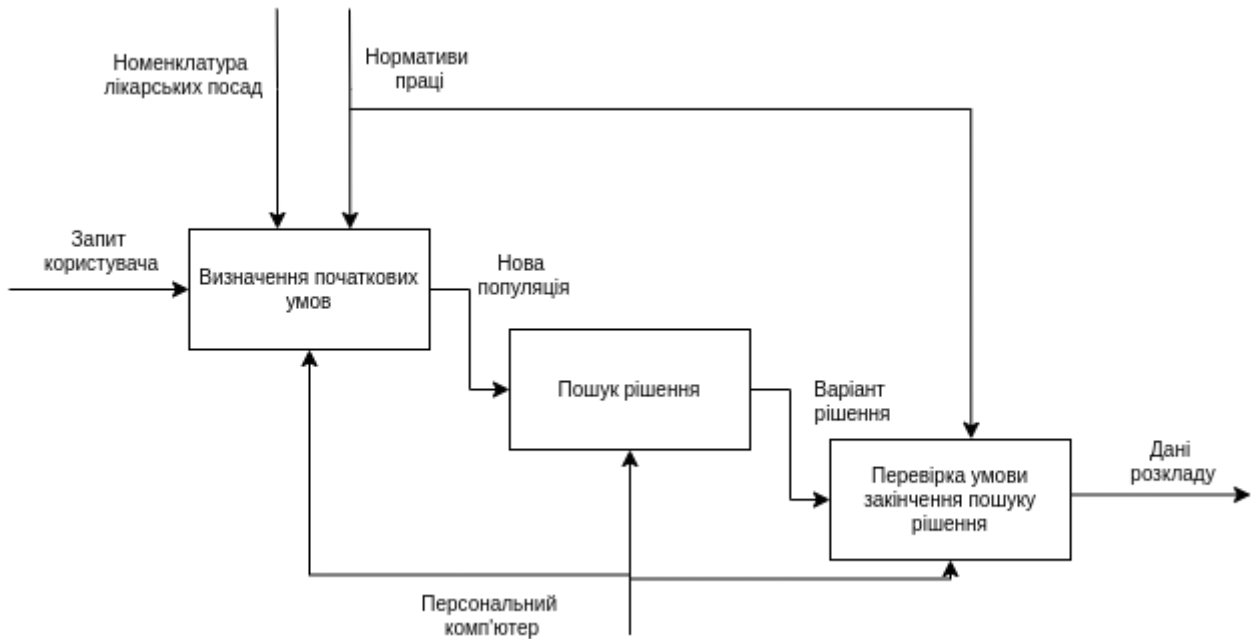


Рисунок 3.5 – Діаграма декомпозиції другого рівня для процесу «Обробка запиту»

3.2 Діаграми класів для компоненту, інтегрованого у ICM3

Для опису інформаційної системи що проектується з точки зору того, як класи системи пов'язані між собою створюється діаграма класів. Діаграми класів містять атрибути, операції класів. Є три види зв'язків у діаграмі класів:

- асоціація – відношення між класами;
- агрегація – зв'язок «частина-ціле»;
- композиція – агрегація при якій залежний клас не може існувати без іншого агрегуючого класу;
- наслідування – коли один клас має поведінку або структуру іншого класу.

Оскільки розроблений компонент автоматичної генерації розкладу розроблена для інтеграції у медичну систему для використання функції запису до лікаря та інформування пацієнтів, створено діаграму класів ICM3 з уже інтегрованим компонентом генерації розкладу. Діаграма класів подана на рисунку 3.6. На діаграмі наведено шість сутностей з іменами, їх властивостями, методами, що характеризують поведінку сутностей, а також відповідні відношеннями між класами.

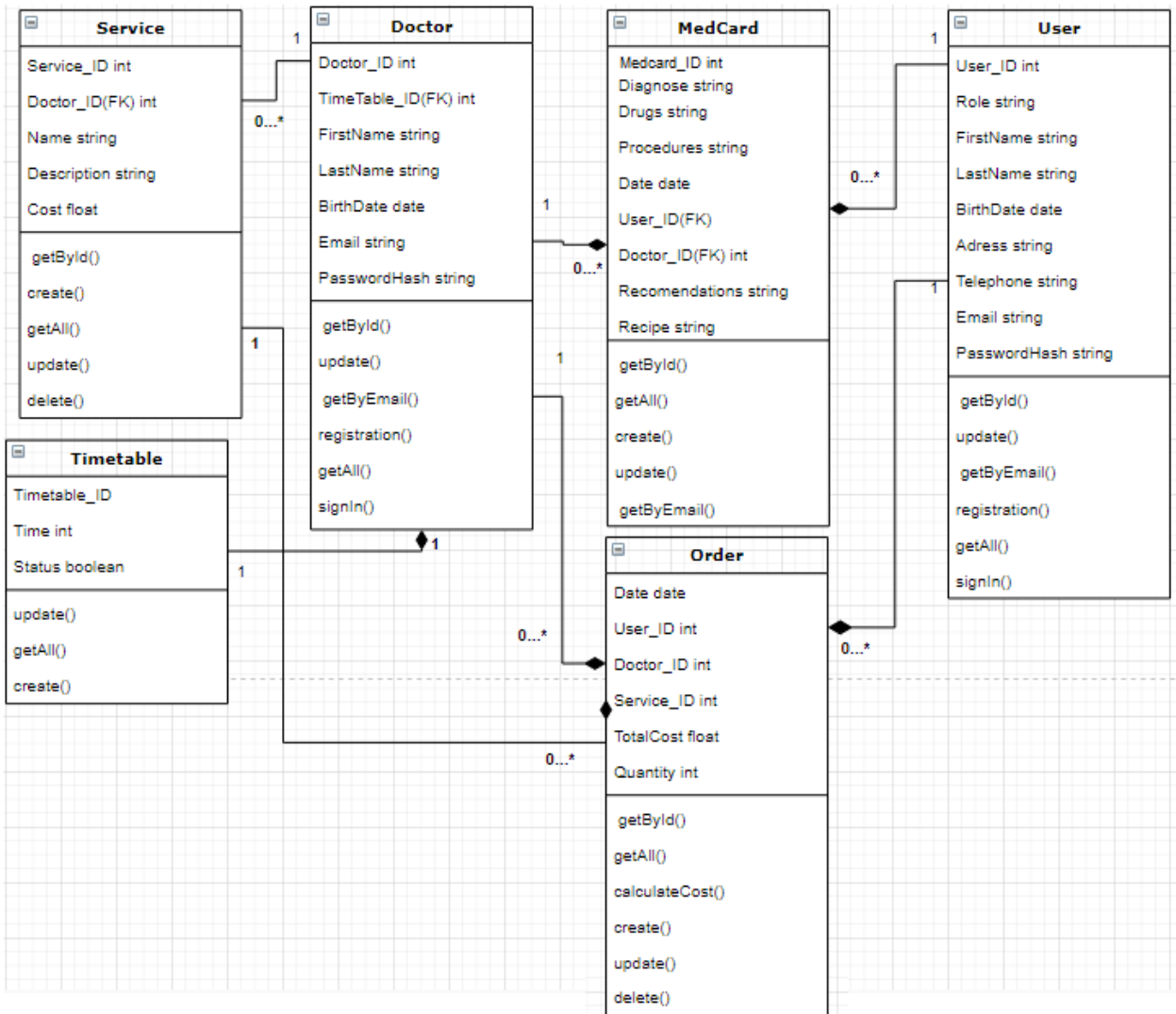


Рисунок 3.6 – Діаграма класів ІСМЗ

На діаграмі класів для ІСМЗ з інтегрованим компонентом генерації розкладу наведені такі класи:

а) Order – клас замовлення, зберігає інформацію про послуги, отримані у лікаря має у собі атрибути:

- Date – дата створення замовлення;
- User_ID – зовнішній ключ для того щоб пов'язати інформацію користувача з його замовленням;
- Service_ID – зовнішній ключ для зв'язку послуги з замовленням у якому вона міститься разом з іншими послугами лікаря;
- TotalCost – загальна вартість послуг, що отримує пацієнт;
- Doctor_ID – зовнішній ключ для зв'язку лікаря з замовленням.

Клас Order має такі операції:

- getById() – операція знаходження послуги за ідентифікаційним номером;
- getAll() – операція виведення послуг, що містяться у базі даних системи;
- create() – операція створення послуги;
- update() – операція оновлення послуги, що вже є у базі даних;
- calculateCost() – операція розрахунку загальної вартості послуг;
- delete() – операція видалення вже наявної але неактуальної послуги з бази даних лікарні.

б) Service – клас, що представляє послуги що надає лікар медичного закладу, цей клас має такі атрибути:

- Service_ID – ідентифікаційний номер;
- Doctor_ID(FK) – зовнішній ключ для зв'язку між послугою та лікарем, який її надає;
- Name – найменування послуги;
- Description – опис послуги;
- Cost – ціна послуги що потім розраховують у класі Order.

Клас Service схожі операції як і клас Order за виключенням відсутності операції для розрахунку загальної вартості.

в) User – клас що представляє дані користувача та має атрибути:

- User_ID – унікальний ідентифікаційний номер;
- Role – роль, необхідна для визначення серед звичайних користувачів системи адміністратора;
- FirstName – ім'я користувача;
- LastName – прізвище користувача;
- BirthDate – дата народження користувача для визначення віку;
- Adress – домашня адреса, може використовуватися у випадку якщо пацієнт робить виклик лікаря додому;
- Telephone – телефон користувача;
- Email – електронна поштова скринька користувача;
- PasswordHash – захешований пароль користувача.

Клас User має такі операції :

- getById() – пошук користувача за унікальним ідентифікаційним номером;
- getAll() – виведення усіх користувачів;
- registration() – операція реєстрації користувача;
- update() – операція оновлення;
- delete() – операція видалення;
- signIn() – операція авторизації користувача;
- getByEmail() – операція пошуку користувача за електронною поштою.

г) MedCard – клас, що представляє запис у медичній картці пацієнта, має такі атрибути:

- MedCard_ID – ідентифікаційний номер;
- Diagnose – поставлений лікарем діагноз;
- Drugs – ліки які виписав лікар;
- Procedures – процедури виписані лікарем для лікування хвороби;
- Date – дата створення запису у медичній картці пацієнта;
- User_ID(FK) – зовнішній ключ для створення зв'язку між пацієнтом та записом лікаря;
- Doctor_ID(FK) – зовнішній ключ для створення зв'язку між лікарем та записом лікаря;
- Recommendations – рекомендації лікаря;
- Recipe – рецепт.

Клас має такі схожі на методи класу замовлення.

д) Doctor – клас, що відображає дані та функції лікаря. Клас Doctor має такі атрибути:

- Doctor_ID – унікальний ідентифікаційний номер;
- TimeTable_ID(FK) – зовнішній ключ для зв'язку сутностей розкладу та лікаря;
- FirstName – ім'я лікаря ;
- LastName – прізвище лікаря;
- BirthDate – дата народження лікаря для встановлення віку;

- Email – поштова скринька для зв'язку з пацієнтами;
- PasswordHash – захеширований пароль лікаря.

Клас має такі самі методи класу користувача.

е) Timetable – клас що представляє сутність розкладу лікаря, він має такі атрибути:

- Timetable_ID – унікальний ідентифікаційний номер;
- Time – час у який лікар може прийняти одного пацієнта
- Status – статус часу: може бути зайнятий чи не зайнятий в залежності від

того записався хтось з пацієнтів на цей час чи ні.

Клас має такі операції:

- getAll() – виведення усіх користувачів;
- update() – операція оновлення;
- create() – операція створення.

3.3 Діаграма варіантів використання для компоненту ІСМЗ

Діаграма варіантів використання компонентів медичної ІС (Use case diagram) описує акторів та варіанти використання системи між якими існують відношення.

Опис елементів діаграми:

а) актор – сутність по відношенню до проектуємої системи є зовнішньою, вступає у відношення з інформаційною системою та відображає роль. Акторами можуть бути зовнішні системи або люди;

б) варіант використання - сутність, що є рядом робіт, які виконує система, після початку взаємодії. Сутність дії системи, але без уточнень;

в) відношення – зв'язок між акторами та варіантами використання, ця сутність відображає те, що актори взаємодіють з системою, у цій діаграмі існують такі типи відношень:

- включення – виконання основного процесу, який має в собі включення може вважатись виконаним тільки після виконання включного процесу;

– розширення – відношення, що використовується, коли варіант має не обов'язкове доповнення і для виконання основного процесу не потрібно виконувати доповнення.

На рисунку 3.7 подана діаграма варіантів використання ІСМЗ з інтегрованим компонентом автоматизації генерації розкладу працівників медичного закладу.

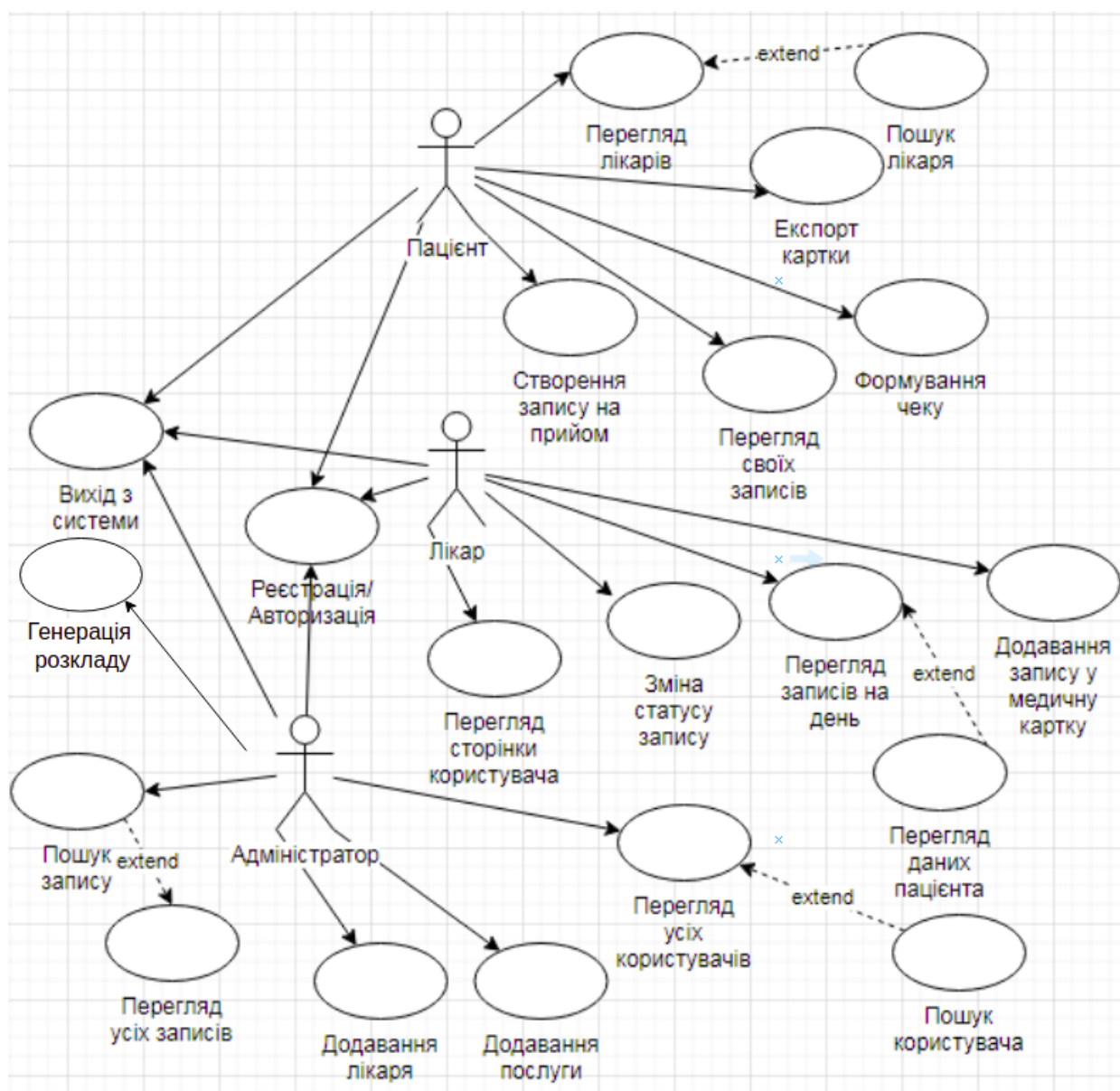


Рисунок 3.7 – Діаграма варіантів використання компоненту медичної ІС

Після інтеграції компоненту генерації розкладу у інформаційної системи з'явився варіант використання «Генерація розкладу» для адміністратора. Тепер адміністратор має можливість створювати запит на генерацію розкладу.

На представленій діаграмі варіантів використання ІСМЗ наведені три актори:

а) головний адміністратор, який має доступ до варіантів використання системи:

- перегляд записів на прийоми;
- пошук запису на прийом;
- додавання лікаря;
- додавання послуги;
- перегляд усіх користувачів;
- пошук користувача.

Перед використанням функцій системи адміністратор має зареєструватись як і інші користувачі системи.

б) лікар має доступ до варіантів використання:

- перегляд сторінки користувача;
- зміна статусу запису на прийом;
- перегляд записів на день до конкретного лікаря(до себе);
- додавання запису у медичну картку.

в) пацієнт, що має доступ до таких варіантів використання:

- перегляд лікарів;
- експорт картки;
- формування чеку;
- перегляд своїх записів;
- створення запису на
- прийом.

Отже, діаграма варіантів використання допомагає зрозуміти які ролі мають користувачі ІСМЗ та ряд функцій ІС, який призначений кожній ролі. У той час як користувачі з іншою роллю не можуть використовуватись цей функціонал, тому що система обмежує несанкціонований доступ до даних.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ КОМПОНЕНТУ ІСМЗ

4.1. Обґрунтування вибору мови програмування та СУБД

Для реалізації компоненту як і самої системи обрана документо-орієнтована СУБД MongoDB. Обрана СУБД саме цього типу тому що вона має швидку обробку даних та простішу структуру порівняно зі структурою реляційних баз даних. СУБД реалізована за допомогою NoSQL підходу, в її основі міститься сховище документів. У MongoDB сховища документів є колекціями файлів, що зберігають дані у форматі JSON. Особливістю колекцій є унікальний ідентифікаційний номер що надається йому при створенні нового документа.

Ще однією особливістю та перевагою MongoDB є гнучку мову для формування запитів та можливість створювати індекси для різних атрибутів, пов'язувати сутності за допомогою індексів. Також MongoDB має додаток з графічним інтерфейсом користувача, що дозволяє зручно взаємодіяти з даними БД, змінювати значення полів, видаляти файли.

NoSQL є нереляційним типом БД, виконуючий функції зберігання та виведення даних без підходу таблиць та відношень в реляційних БД, такі бази даних мають простий дизайн схеми та доступність до даних.

Для програмної реалізації запропонованого компоненту автоматизованого створення розкладу обрано мову програмування TypeScript та фреймворк Vue.js.

TypeScript пропонує багато механізмів, таких як інтерфейси, класи, типи утиліт, тому ці переваги є достатніми для того щоб обрати саме цю мову.

4.2 Розробка структури компоненту ІСМЗ

ІСМЗ розділена на два проекти: клієнтську частину та серверну. Клієнтська частина – це проект API, написаний на Typescript з використанням гнучкого веб-фреймворку Express та програми моделювання БД MongoDB і призначений для асинхронної роботи Mongoose. Mongoose – це бібліотека об'єктно-орієнтованого програмування JavaScript. NodeJS компілює Typescript у

Javascript, після чого проект запускається. Клієнтська частина реалізована мовою Vue.js.

При отриманні інформації від браузера клієнт перевіряє інформацію, валідує її, наприклад коли користувач вводить в поле поштової скриньки інформацію, що не відповідає формату адреси поштової скриньки, клієнт видає помилку. Також на клієнті зберігаються ресурси для використання функцій сайту, як користувацькі дані та токен JWT, який перевіряється на сервері.

JWT (JSON Web Token) – це особливий стандарт доступу токена на основі JSON, цей токен використовується для підтвердження користувачів.

Для кращого розуміння структури ІСМЗ наведена структурна схема ІСМЗ на рисунку 4.1, де наведена база даних, back-end або серверна частина з якої відправляються дані у форматі JSON до клієнтської front-end частини, яка через Інтернет з'єднуються з персональними комп'ютерами користувачів. Після обробки даних серверна частина, що відповідає за базовий функціонал системи, буде взаємодіяти з базою даних, та обробляти дані: оновлюватися або видалятися з БД.

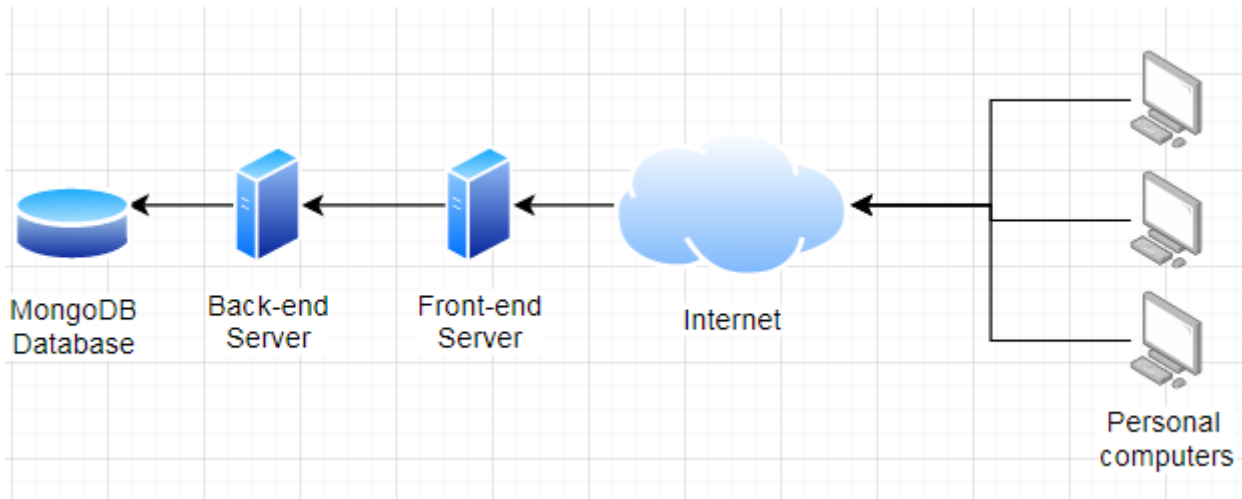


Рисунок 4.1 – Структурна схема ІСМЗ

Оскільки СУБД використовуваної розробки компонентів інформаційної медичної системи є NoSQL, база даних має досить просту, зручну і зрозумілу структуру. Загальна структура бази файлової бази даних подана на рисунку 4.2.

Після того як проаналізована структура БД NoSQL типу, побудована нова схема для структури бази даних ІСМЗ, подана на рисунку 4.3.



Рисунок 4.2 – Схема узагальненої структури NoSQL бази даних

На схемі наведена загальна структура файлової БД, що складається з:

- колекцій;
- документів;
- полей документів.

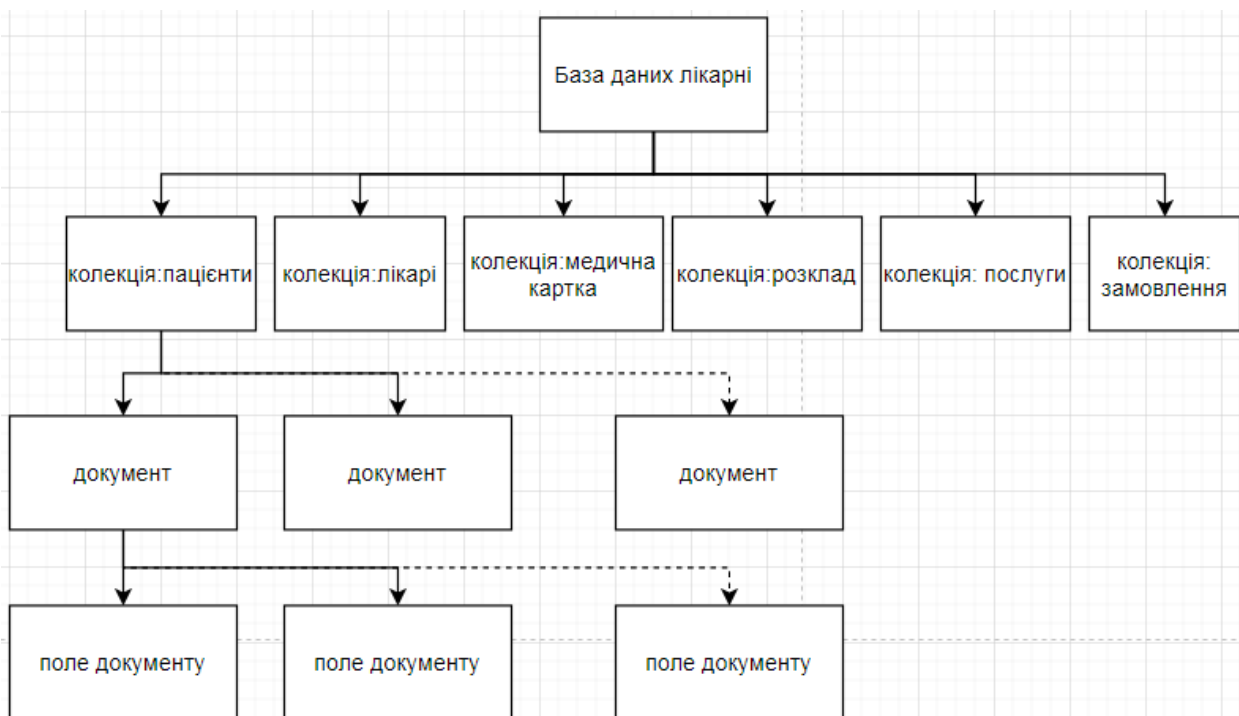


Рисунок 4.3 – Схема структури бази даних ІСМЗ

Структура БД є колекцією файлів, що містять дані у форматі JSON: поля з інформацією та індекси.

Створена БД медичного закладу має у собі такі колекції:

- лікарі;
- послуги (services);
- користувачі;
- медична картка;
- замовлення;
- розклад.

Розглянемо файли з колекцій користувачів(users) та послуг(services).

Структура файлу БД з колекції services у форматі JSON:

```
{
  "_id": {"$oid": "5e6f6dc2faca3e1590b0f322"},
  "doctor_ids": [
    {"$oid": "5e6b7dfdae5a8842cc7c218f"},
    {"$oid": "5e6b971f121cf92320baf8bc"}],
  "name": "someInfo",
  "description": "someInfo",
  "price": {"$numberInt": "1223"},
}
```

Опис полей файлу:

- "_id" – унікальний індекс що надається файлу при його створенні тому його не потрібно прописувати у схемі і створювати “вручну”.
- "doctor_ids" – масив індексів лікарів, що виконують дану послугу;
- "name" – найменування медичної послуги;
- "description" – опис послуги;
- "price" – ціна послуги.

Код схеми бази даних для створення документів для колекції послуги:

```
let ServicesSchema = new Schema({
  name: { type: String, required: true, unique: true },
  description: { type: String, required: false, unique: false },
  doctor_ids: [{ type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true }],
```

```
price: { type: Number, required: true },
}, { collection: 'services' });
```

В даному кодї можна побачити що у схемі вказується тип поля файлу: `type`, обов'язковість: `required` та унікальність заданого атрибуту `unique`. В даному випадку поле `description` має строковий тип `String`, не є обов'язковим для заповнення та не має бути унікальним в колекції.

Також у схемі обов'язково пишеться до якої колекції потрібно додати створений файл: `collection: 'services'`.

Код схеми бази даних для створення документів для колекцій лікарів:

```
{
  "_id": {"$oid": "5e6b7dfdae5a8842cc7c218f"},
  "timetable_Id": {"$oid": "5e6b7dfdae5a8842cc7c218f"},
  "email": "1Doctor@gmail.com",
  "firstName": "1DoctorFName",
  "lastName": "1DoctorLName",
  "passwordHash": "$2b$10$iJVA9wGRJMrfdPhDUG3c9OwapXyLDWPccYJ1Nmaj
bfu1Bi6U8Cae",
  "birthDate": "2020-02-02"
}
```

Опис полей файлу:

- `"_id"` – унікальний індекс що надається файлу при його створенні тому його не потрібно прописувати у схемі і створювати “вручну”.

- `"timetable_Ids"` – зовнішній ключ для розкладу лікаря;

- `"email"` – адреса поштової скриньки;

- `"firstName"` – ім'я користувача;

- `"lastName"` – прізвище;

- `"passwordHash"` – хеш пароля, пароль хешується, після чого його не розшифровують, а зберігають у вигляді хешу, при авторизації користувач вводить свій пароль, який теж хешується, після чого проводиться порівняння хешів;

- `"birthDate"` – дата народження.

4.3 Програмна реалізація генетичного алгоритму автоматизованої генерації розкладу роботи лікаря

Вхідними даними до програми для автоматизованої генерації робочого розкладу для медичного персоналу є:

- час початку та кінця роботи працівника медичного закладу;
- період на який буде створений розклад за замовчуванням періодом є один місяць, тому що немає сенсу створювати розклад на більший період, через зростання ймовірності помилок у розкладі, що призведе до майбутньої необхідності вручну виправляти всі помилки, що є зайвою витратою часу працівника, який займаються складанням розкладу лікарів;
- інформація про вихідні дні та дні відпустки;
- дані лікарів: ідентифікаційний номер;
- номери кабінетів.

Обмеженнями вирішення задачі є час початку та кінця роботи лікаря, тому що буде вважатися помилкою створення розкладу у разі якщо час роботи одного лікаря у одному кабінеті буде співпадати з часом роботи іншого лікаря у цьому ж кабінеті. Тобто умовою є те, що в одному кабінеті може працювати тільки один лікар у свій робочий час. Два лікарі не можуть працювати у одному кабінеті одночасно.

Програма видає розклад у форматі json, з якого потім з якого виділяється дата та час прийому лікаря, його ідентифікаційний номер та заноситься до БД.

Приклад json даних розкладу:

```
{ time: {"507f1f77bcf86cd799439011", '16:25:00 29/11/2021', '16:45:00 29/11/2021'}}
```

За результатами тестування швидкість генерації розкладу для кількості лікарів у розмірі 40 дорівнює 33,4 хвилини.

Отримані результати надходять до бази даних. У разі виникнення помилок у розкладі рекомендується вручну перевірити та виправити помилки за допомогою інтерфейсу або запустити повторна генерацію розкладу.

Повний код реалізації програми наведений у додатку Б. Приклад реалізації генетичного алгоритму на мові програмування typescript:

```
const solve = require('..').minCollisions
const multiple = (length, string) =>
  Array.apply([], { length }).map((a, i) => `${string}${i + 1}`)
const randomSet = (length, bookables) => {
  const bookablesCopy = bookables.slice()
  const result = []
  while (length--) {
    const randomIndex = Math.floor(Math.random() *
bookablesCopy.length)
    result.push(bookablesCopy[randomIndex])
    bookablesCopy.splice(randomIndex, 1)
  }
  return result
}
const days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday',
'Friday']
const slots = days.reduce((acc, day) => acc.concat(multiple(4,
day)), [])
const bookables = [].concat(multiple(10, Allergy))
.concat(multiple(8, 'Anesthesiology'))
.concat(multiple(8, 'Immunology'))
.concat(multiple(10, 'Diagnostic radiology'))
.concat(multiple(5, 'Emergency medicine'))
.concat(multiple(5, 'Family medicine'))
solve({ slots, bookables, constraints }, { skip: 1, size: 50 },
  (table, { fitness }) => console.log(table, fitness),
  (table, { generation, fitness }) => console.log(generation,
fitness))
```

4.4 Оцінка ефективності розробленого компоненту

Оцінка ефективності компоненту медичної системи для автоматизації генерації розкладу лікаря – це окрема та достатньо складна наукова задача, тому запропоновано проводити оцінку ефективності за такими критеріями:

- час генерації розкладу;
- відносна кількість записів до лікарів за допомогою сайту ІСМЗ та час створення розкладу.

На рисунку 4.4 наведено графік залежності кількості записів створених за допомогою сайту та кількості записів створених за допомогою реєстратури від часу, після інтеграції компоненти у систему. Записи створені за допомогою реєстратури зображені на графіку синім кольором, а записи створені за допомогою сайту зображення помаранчевим.

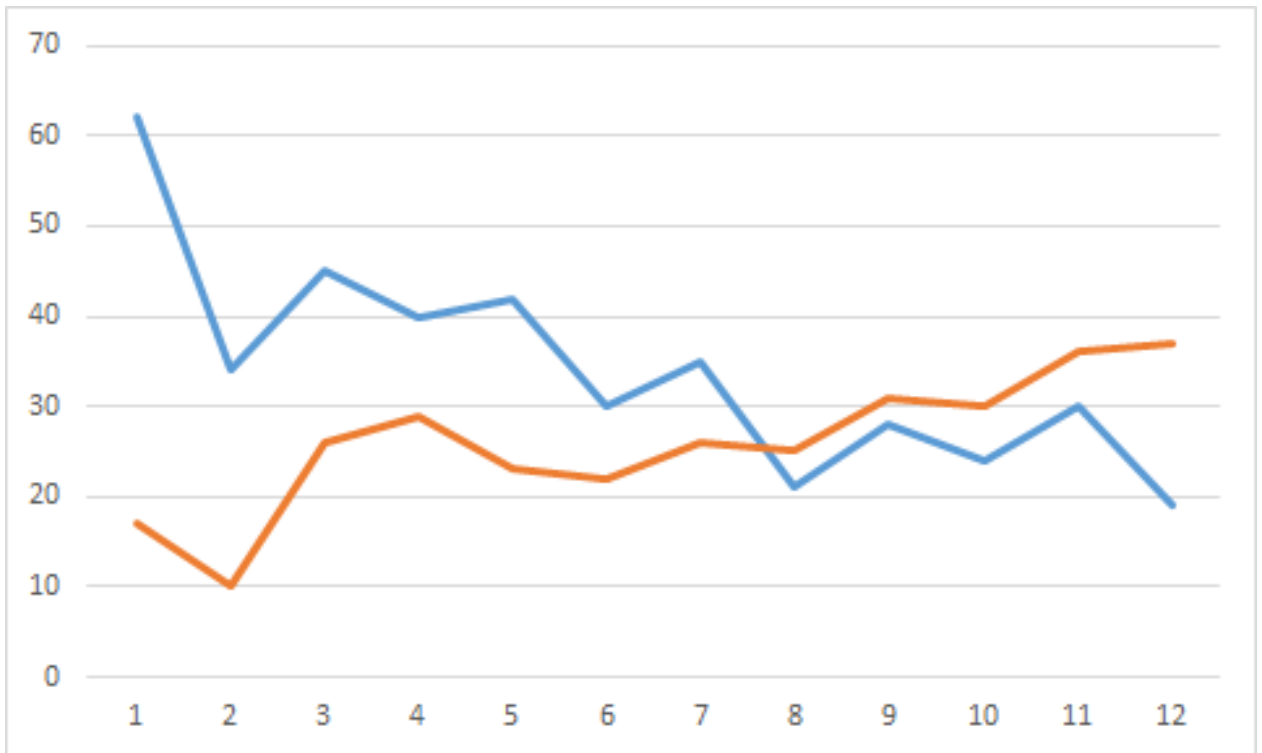


Рисунок 4.4 – Графік залежності кількості записів від часу після інтеграції у систему модуля автоматичної генерації розкладу

Позначимо відносну кількість записів до лікарів за допомогою сайту ІСМЗ E , що розраховується за формулою:

$$E = \frac{N_s - N_t}{N},$$

де N_s – це кількість записів за допомогою сайту;

N_t – це кількість записів до лікаря через реєстратуру;

N – загальна кількість записів до лікаря.

За цією формулою побудований графік показника ефективності та наведений на рисунку 4.5.

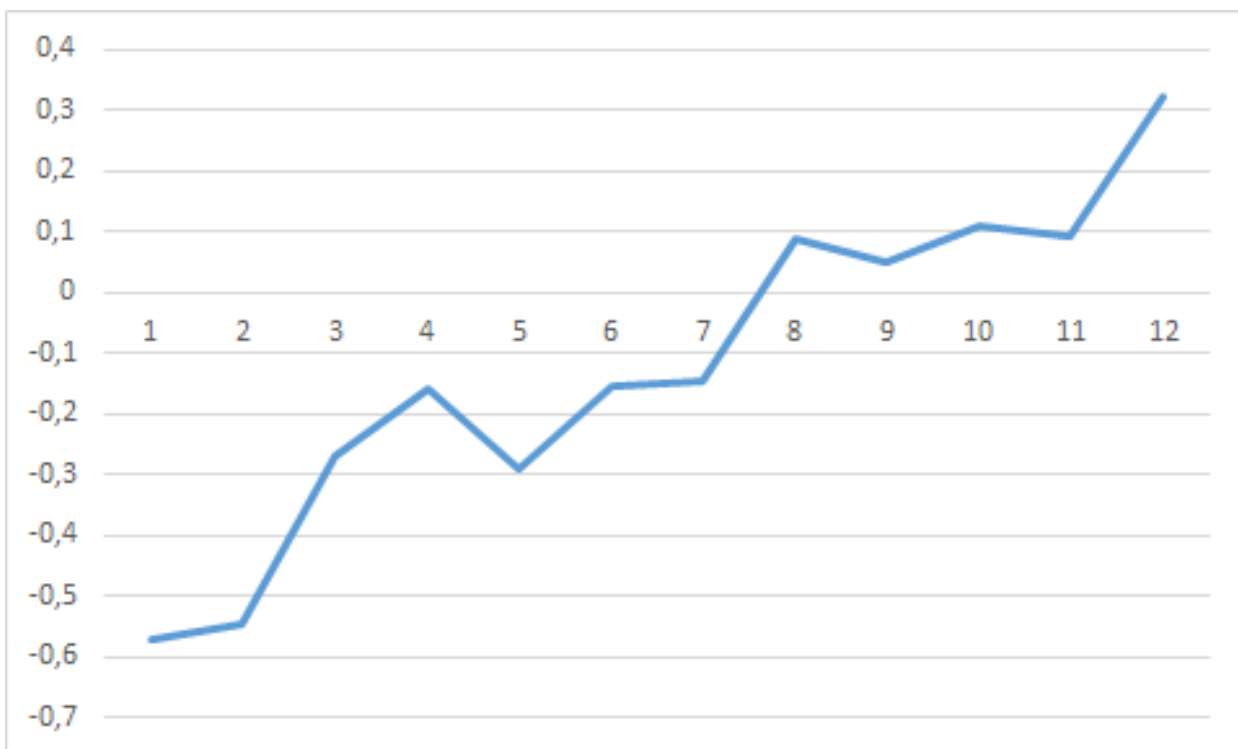


Рисунок 4.5 – Графік ефективності розробленого компоненту автоматичної генерації розкладу

За наведеним графіком можна зробити висновки про те, що інтеграція компоненту автоматизації генерації розкладу медичного персоналу у ІСМЗ було вдалим рішенням, що підтверджується зростом ефективності.

ВИСНОВКИ

Проведено дослідження та порівняння працюючих медичних інформаційних систем та виявлені їх проблеми, основні переваги та недоліки. Виділені основні послуги, що ці системи надають користувачеві. Найбільш важливою функцією кожної з розглянутих систем є функція створення запису до лікаря, для цієї функції є критичними наявність коректного розкладу лікаря. Виконане планування розробки компоненту системи: визначені основні системні вимоги до модулю автоматичного створення розкладу. Розглянуті підходи та методи вирішення задачі автоматичної побудови розкладів для лікарів.

Проведено дослідження та порівняння найбільш використовуваних алгоритмів для автоматизації процесу генерації розкладу лікарів:

- генетичного алгоритму;
- мурашині алгоритми;
- метод гілок та меж.

На основі дослідження обраний генетичний алгоритм та спроектована компонента генерації розкладу лікаря, спроектована система з інтегрованою у неї компонентою.

Виконана програмна реалізація генетичного алгоритму автоматизованої генерації розкладу роботи лікаря. Для визначення ефективності системи після інтеграції компоненту генерації розкладу запропоновані такі критерії оцінки ефективності: час, що витрачається на створення робочого розкладу та відносна кількість записів до лікарів за допомогою сайту ІСМЗ. Побудований графік залежності кількості всіх записів до лікаря за допомогою реєстратури та кількості записів до лікаря за допомогою сайту інформаційної системи медичного закладу. За проведеним аналізом отриманих результатів введення компоненту автоматизації генерації розкладу був вдалим рішенням та підвищення ефективності роботи медичного закладу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Igor Grebennik, Rémy Dupas, Oleksandr Lytvynenko, Inna Urniaieva, "Scheduling Freight Trains in Rail-rail Transshipment Yards with Train Arrangements", International Journal of Intelligent Systems and Applications (IJISA), Vol.9, No.10, pp.12-19, 2017. DOI: 10.5815/ijisa.2017.10.02 Scopus
2. В. А. Павлиш, Л. К. Гліненко, Н. Б. Шаховська. Основи інформаційних технологій і систем: підручник / Львівська політехніка, 2018. – 620 с. – ISBN 966-941-264-5.
3. Т.М. Басюк, Н.О. Думанський, О.В. Пасічник Основи інформаційних технологій: навчальний посібник, Львів, 2020.
4. Гужва В. М. Г 93 Інформаційні системи і технології на підприємствах: Навч. посібник. – К.: КНЕУ, 2001. – 400 с.
5. Класифікація інформаційних систем та інформаційних технологій. URL: https://stud.com.ua/21185/informatika/klasifikatsiya_informatsiynih_sistem_informatsiynih_tehnologiy
6. Медична інформаційна система. URL: <https://res.in.ua/medichna-informacijna-sistema-mis.html>
7. Перспективи розвитку медичних інформаційних систем. URL: <http://studcon.org/perspektyvy-rozvytku-medychnyh-informaciynyh-system>
8. Нетепчук, В. В. (2014) Управління бізнес-процесами. НУВГП, Рівне. URL: <http://ep3.nuwm.edu.ua/id/eprint/8812>
9. Сервіс запису до лікарів та зберігання медичних даних. URL: <https://helsi.me/>
10. Medcard24. URL: <https://img.medcard24.net/>
11. Поліклініка без черг. URL: <https://pb4.com.ua/>
12. Бурнасов П.В. Математическая постановка задачи составления расписания занятий, Вестник ИрГТУ. 2014.
13. Бульонков М.А., Емельянов П.Г., Пак Е.В. К стандартизации данных для составления расписания в учебных заведениях. Открытое образование. 2010.
14. Снитюк В.Є. Сіпко Є.Н. Про особливості формування цільової функції та обмежень в задачі складання розкладу занять//Математичні машини і системи, 2014.

15. Дворянкин А.М. Чалышев В.С. Обзор методов составления расписания вузов. Изв. ВолгГТУ. Серия Актуальные проблемы управления, вычислительной техники и информатики в технических системах: межвуз. сб. науч. ст. 2011.
16. Томашевський В.М., Новіков Ю.Л., Камінська П.А. Складання розкладів занять у дистанційних системах навчання. Вісник НТУУ «КПІ» Інформатика, управління та обчислювальна техніка, 2010.
17. Jha S.K. Exam Timetabling Problem using Genetic algorithm. International Journal of Research in Engineering and Technology. 2014. Vol.3, No5.
18. Астахова И.Ф., Фирас А.М. Составление расписания учебных занятий на основе генетического алгоритма. Вестник ВГУ, серия: Системный анализ и информационные технологии. 2013. No 2.
19. Юрчак І.Ю., Москович Т.Р. Дослідження генетичних алгоритмів та їх застосування їхнього в автоматизованій системі розподілу навантаження для викладачів і студентів. URL: <http://eom.lp.edu.ua/sntk/doc/ksm2018/moskovytch.pdf> (дата звернення 01.10.2021).
20. Thepphakorn T, Pongcharoen P, Hicks C. An ant colony based timetabling tool. International Journal of Production Economics, 2014, No 149(3).
21. 1С: Автоматизированное составление расписания. Университет. URL: https://solutions.1c.ru/catalog/asp_univer (дата звернення: 05.11.2021).
22. Verma O.P., Garg. R., Bisht V.S., Optimal Time-Table Generation by Hybridized Bacterial Foraging and Genetic Algorithm // Proceedings of International Conference on Communication Systems and Network Technologies (CSNT'12), (2012), Pp. 919-923.
23. ITC 2019: International Timetabling Competition. URL: <https://www.itc2019.org/home> (дата звернення: 05.11.2021).
24. Безгинов, А.Н Трегубов С.Ю. Обзор существующих методов составления расписаний // Информационные технологии в программировании: межвуз. сб. ст. 2005. Вып. 2(14). С.5-19.
25. Титов Ю.П. Модификации метода муравьиных колоний для разработки программного обеспечения решения задач многокритериального управления поставками // Современные информационные технологии и ИТ-образование, 2017. Том 13. No 2.