

УДК 616.082

Ковалишин О.С.

Національний університет «Львівська політехніка»

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ РОЗКЛАДІВ ВІДНОВЛЮВАЛЬНОЇ ТЕРАПІЇ

Ковалишин О.С. Інформаційна технологія оптимізації розкладів відновлювальної терапії. В роботі встановлено, що для побудови розкладів використовують ряд методів, починаючи від ручного планування і математичного програмування з обмеженнями, закінчуючи використанням засобів штучного інтелекту. Розроблено інформаційну технологію автоматизованої побудови та оптимізації розкладів відновлювальної терапії. Обґрунтовано вибір методів і засобів для реалізації програмного комплексу. Охарактеризовано архітектуру системи, структуру бази даних для збереження результатів роботи системи, а також основні компоненти графічного користувацького інтерфейсу.

Ключевые слова: клініка, розклад, побудова розкладу, оптимізація розкладу

Ковалишин О.С. Информационная технология оптимизации учреждений восстановительной терапии . В работе установлено, что для построения расписаний используют ряд методов, начиная от ручного планирования и математического программирования с ограничениями, заканчивая использованием средств искусственного интеллекта. Разработана информационная технология автоматизированного построения и оптимизации расписаний восстановительной терапии. Обоснован выбор методов и средств для реализации программного комплекса. Охарактеризованы архитектуру системы, структуру базы данных для хранения результатов работы системы, а также основные компоненты графического пользовательского интерфейса.

Ключові слова: клініка, расписание, построение расписания, оптимизация расписания

Kovalyshyn O.S. Information technology of rehabilitation institutions schedules optimisation. In the work it is established that for the construction of schedules, there are a number of methods, from manual planning and mathematical programming with constraints, ending with the use of means of artificial intelligence. The information technology of automated construction and optimization of regenerative therapy schedules is developed. The choice of methods and means for implementation of the software complex is substantiated. The architecture of the system, the structure of the database for maintaining the results of the system as well as the main components of the graphical user interface are described.

Key words: clinic, schedule, scheduling, optimization of the schedule

Вступ. Функціонування клінік, а також якісне обслуговування в них пацієнтів значною мірою залежить від розкладу їх роботи.

Складання розкладу функціонування клініки є надзвичайно трудомістким процесом, що враховує низку обмежень та факторів: наявність кваліфікованих лікарів; відповідність отриманих медичних послуг послідовності плану терапії пацієнтів; наявність відповідного медичного обладнання на момент проведення лікування; необхідність спеціалізованих приміщень для проведення процедур тощо. Розклад можна назвати оптимальним за умови повної реалізації обмежень предметної області, ефективного використання наявних ресурсів, врахування побажань пацієнтів та персоналу клініки, відповідності процесу лікування сформованим планам [1].

Якщо під час його складання розкладу не враховані в повному обсязі вимоги до планів реабілітації, або не задоволені обмеження якості лікування знижується. Це в свою чергу істотно позначається на якості медичних послуг. Поряд з наведеними вище обмеженнями доцільним є задоволення побажань пацієнтів [2].

Беручи до уваги той факт, що в більшості українських клінік складання розкладу відбувається вручну, а також враховуючи надзвичайну складність урахування всіх обмежень, велику увагу необхідно приділяти автоматизації процесу його складання.

Постановка наукової проблеми. Складання та оптимізації розкладів відноситься до класу NP-повних задач [3,4]. Для їх розв'язку за поліноміальний час залежно від розмірності не існує ефективних детермінованих методів вирішення. Тому для такої категорії завдань використовуються наближені методи, які дозволяють скласти субоптимальний розклад [4,5], зокрема: метод імітації відпалу металу; метод розфарбування графу; метод генетичного алгоритму тощо. Значна кількість існуючих методів, а також можливості їх конфігурації значною мірою впливають на результати роботи, пов'язані із складанням розкладу. Через це розробка універсальної інформаційної технології вирішення задачі побудови та оптимізації розкладів є актуальною проблемою і потребує ефективного вирішення.

Виклад основного матеріалу та обґрунтування отриманих результатів дослідження.

Задачу побудови розкладу розглядають як процес, що реалізує функцію розподілу деяких процедур у часі. Оптимальний розклад дозволить отримати максимальний ефект від

раціонального використання наявних в клініки ресурсів, збільшити кількісні та якісні показники лікування, підвищити рівень задоволеності працівників та пацієнтів.

Для його побудови необхідно використовувати методи, які в тому чи іншому вигляді потребують наявності коректної першопочаткової (опорної) вибірки. Часто її генерація виконується випадковим чином, що в даній ситуації не може бути виконано, зважаючи на набір обов'язкових обмежень, накладених предметною областю [6,7].

З іншої сторони, множина обмежень є детермінованою, набір необхідних операцій для створення розкладу та їх порядок є визначеними.

Відповідно до цього стає реальністю розробка системи, що дозволить задовольнити базові обмеження та дасть можливість побудувати опорну вибірку. Беручи до уваги необхідність використання системи великою кількістю користувачів, доцільним є застосування розподіленої архітектури програмного комплексу. Для реалізації таких систем існують два розповсюджені способи: використання "товстого" та "тонкого" клієнта.

Спосіб "тонкого" клієнта має низку переваг, зокрема:

- можливість легкої інтеграції зі сторонніми додатками;
- можливість масштабування за вимогою (під час використання хмарних ресурсів);
- кращий рівень безпеки та вища стійкість проти відмов;
- простіша підтримка клієнтів
- незалежність від апаратної архітектури та операційної системи клієнтів;
- необхідність у підтримці лише однієї операційної системи та набору сторонніх допоміжних ресурсів і додатків;
- простота у додаванні нового функціоналу та розмежування доступу клієнтів до різних частин ужитку;
- висока швидкість розробки.

З огляду на наведений вище перелік переваг та особливості даного програмного продукту було зроблено вибір на користь використання саме "тонкого" клієнта.

Програмний комплекс системи оптимізації планів відновлювальної терапії представляє собою трирівневу архітектуру, що складається із клієнтського середовища, сервера застосувань, а також сервера бази даних.

Компонентами програмного комплексу системи оптимізації планів відновлювальної терапії є: web-сервер; сервер бази даних; сервер безперервної інтеграції; система контролю версій; сервер застосувань; сервер представлення клієнтської інформації/

Web-сервер. В даній роботі використано хмарний сервіс OpenShift, що надається компанією RedHat. Він дозволяє швидко налаштувати середовище для розгортання програмних продуктів шляхом вибору картриджів (віртуальних робочих станцій) для мови програмування та пов'язаних ужитків. Картриджі будуються на основі готових шаблонів, що забезпечує швидке розгортання програмних комплексів. В залежності від апаратних та програмних вимог картриджі можна конфігурувати чи компонувати з вже існуючи нові картриджі.

Сервер бази даних. Для роботи з даними обрано сервер MongoDB, що дозволяє швидко та зручно маніпулювати з даними великого об'єму. Перевага використання даної бази в тому, що вона є компактною і надає можливість дублікування записів, що зменшує ризик втрати даних, оскільки її структура є розподіленою та зберігається в декількох місцях. Також MongoDB забезпечує можливість зміни структури даних в процесі розгортання нових версій програмних комплексів за рахунок гнучкої схеми записів.

Сервер безперервної інтеграції. Для безперервної інтеграції обрано сервер Jenkins, що забезпечує легкість здійснення розгортання нових версій програмного комплексу. Перевагою використання саме цього продукту є підтримка його від спільноти, що забезпечує стабільне оновлення версій. Також Jenkins забезпечує гнучку конфігурацію та розширення власного функціоналу за рахунок використання плагінів.

Система контролю версій. Для зберігання вихідного програмного коду використано GIT та сервер контролю версій Bitbucket. GIT займає лідируючі позиції та зарекомендував себе як надійна системи контролю версійності коду. Bitbucket є одним із найбільших репозиторіїв для спільної розробки програмного забезпечення. Він дозволяє переглядати збережений код, має функцію підсвічування синтаксису для різних мов програмування.

Сервер застосувань. В його основу покладені методи побудови оцінки та оптимізації розкладів.

Сервер представлення клієнтської інформації. В даній роботі використано сервер Node JS, що дозволяє швидко побудову функціональних клієнтських представлень реального часу.

Принцип роботи архітектури програмного комплексу на рівні адміністратора представлено на рис 1.

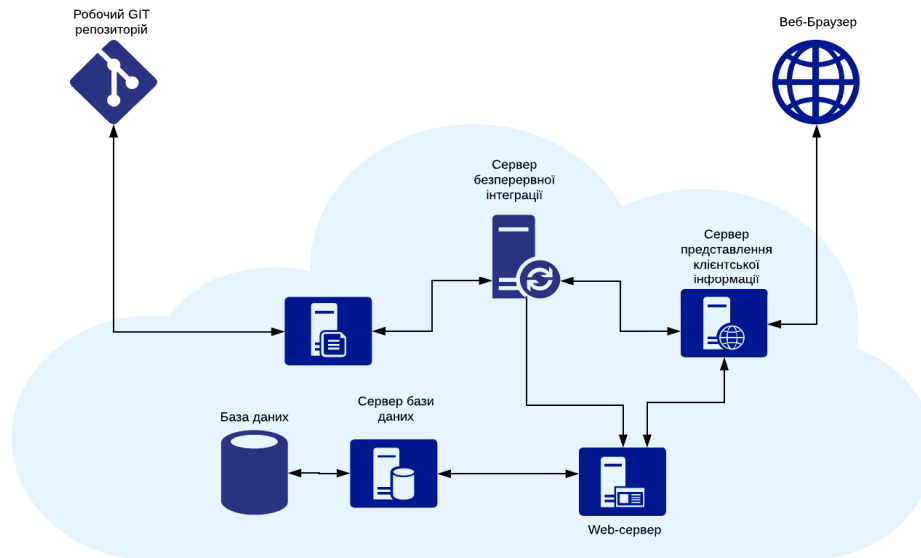


Рис. 1. Архітектура програмного комплексу оптимізації планів відновлювальної терапії

Робота клієнтів з програмним комплексом починається із запитів до сервера побудови представлення, що буде графічний інтерфейс для клієнтів. Запити зроблені через графічний інтерфейс і надходять до сервера застосувань через web-сервер. Дані, які оброблюються сервером застосувань, передаються до сервера баз даних для зберігання.

Сервер застосувань включає в себе наступні структурні рівні: рівень представлення даних; рівень бізнес логіки; рівень доступу до даних.

Рівень представлення даних відповідає за конвертування кінцевої інформації для представлення її на стороні клієнта. Цей рівень реалізований за допомогою REST-контролерів, що є частиною програмної екосистеми Spring Framework.

Бізнес рівень реалізує основну логіку роботи програмного комплексу. Він реалізований за допомогою сервісів Spring Framework.

Рівень доступу до даних містить в собі реалізацію методів доступу до колекцій бази даних та класи безпосередньої роботи з ними.

Рівень доступу до даних та бізнес рівень використовують наскрізну функціональність, яка містить в собі основні рішення, що впливають на роботу декількох рівнів (Cross-cutting).

Обґрунтування вибору технологій. В ході реалізації програмного комплексу оптимізації планів відновлювальної терапії було використано наступні технології:

- мова програмування високого рівня Java версії 1.8;
- NoSQL база даних MongoDB;
- платформа Java Enterprise Edition;
- Spring Framework.

В якості інтегрованого середовища розробки було використано IntelliJ Idea версії 2016. Дане середовище надає можливість автоматизованого генерування діаграм класів та має широкий спектр різних засобів інтеграції з аналогічними для розробки програмного забезпечення як системи контролю версій, web-сервера, покриття коду тощо.

Для побудови проекту та узгодження залежностей між зовнішніми бібліотеками використано систему автоматичної збірки проектів Gradle.

Java – об'єктно-орієнтована мова програмування з C-подібним синтаксисом. Основною перевагою Java є незалежність від типів операційних систем та їх архітектури, тобто, код

написаний на Java, можна запускати на більшості з наявних платформ без змін у початковому коді та їх перекомпіляції.

Платформа Java Enterprise Edition надає методи і засоби виконання для розробки та запуску корпоративного програмного забезпечення, такого як мережеві сервіси, web-сервіси, системи розподіленого типу тощо.

Spring Framework – універсальний комплекс програмних бібліотек з відкритим вихідним кодом для Java-платформи. Spring MVC – модуль Spring Framework, що базується на сервлетах та дає можливість розробляти web-орієнтовані прикладні програми і REST-сервіси. Його архітектура поділяє систему на три частини: модель даних, вигляд даних та керування. Ця архітектура застосовується для відокремлення даних від інтерфейсу користувача, щоб його зміни мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача. Spring Data – модуль Spring Framework для здійснення роботи з серверами баз даних. Перевагою його використання є абстракція від кінцевого типу бази даних, що використовується програмним комплексом.

MongoDB – нереляційна (NoSQL) система управління базами даних. Вона була створена у відповідь на обмеження традиційної технології реляційних баз даних. У порівнянні з ними бази даних MongoDB дозволяють краще масштабування та забезпечують вищу продуктивність, їх модель усуває недоліки реляційної моделі та характеризується наступними перевагами:

- висока документоорієнтованість, оскільки MongoDB не вимагає наявності чітко визначеної схеми даних. Кількість полів, вміст та розмір документа може відрізнятись від одного документа до іншого;

- не вимагає розробки складних операторів об'єднання даних з різних таблиць;

- підтримує динамічні запити до документів, використовуючи мову запитів таку ж потужну, як SQL;

- усуває необхідність використання систем об'єктно-реляційного відображення, тим самим підвищує швидкодію;

- забезпечує можливість використання індексів, побудованих на будь-яких полях документів, що пришвидшує пошук та запис даних.

MongoDB використовується для реалізації складних корпоративних систем з потребами високої швидкодії, наприклад: систем аналізу фінансових даних; моніторингу ринків інвестицій; міжнародних платіжних платформ; систем медичної інформації тощо.

Процес побудови та оптимізації розкладу.

Інтерфейс користувача (рис. 2) програмного комплексу містить: меню користувача програми (1); головне меню програми (2); мультиекран (3), який залежно від конфігурації (4) може складатись зі списку усіх пацієнтів клініки або тільки пацієнтів поточного користувача; обладнання, закріплене за поточним користувачем; розклад функціонування клініки або лише процедуру поточного користувача. Головне меню програми дозволяє навігацію від мультиекрану поточного користувача до конкретних функціональних екранів системи. Побудова та оптимізація розкладу здійснюються на екрані «Розклад», що складається з форм побудови розкладу для заданих відрізків часу, його оптимізації для певних часових відрізків, списку лікувальних приміщень, а також календаря для наочної репрезентації робочого графіку клініки. Перехід до екрану «Розклад» виконується з головного меню програми. Побудова та оптимізація розкладу здійснюється на основі даних, попередньо введених на екран «Пацієнти» за допомогою внутрішніх алгоритмів та обмежень, введених у систему.

Для початку побудови розкладу необхідно ввести дати початку та закінчення часового відрізка побудови розкладу, форми побудови розкладу та натиснути лівою клавішею миші на кнопку «Збудувати». Після закінчення роботи алгоритму результати будуть відображені на вбудованому календарі системи.

Для оптимізації розкладу необхідно ввести дати початку та закінчення часового відрізка оптимізації розклад та натиснути лівою клавішею миші на кнопку «Оптимізувати». Після закінчення роботи алгоритму оптимізації результуючі зміни в розкладі відобразяться на календарі.

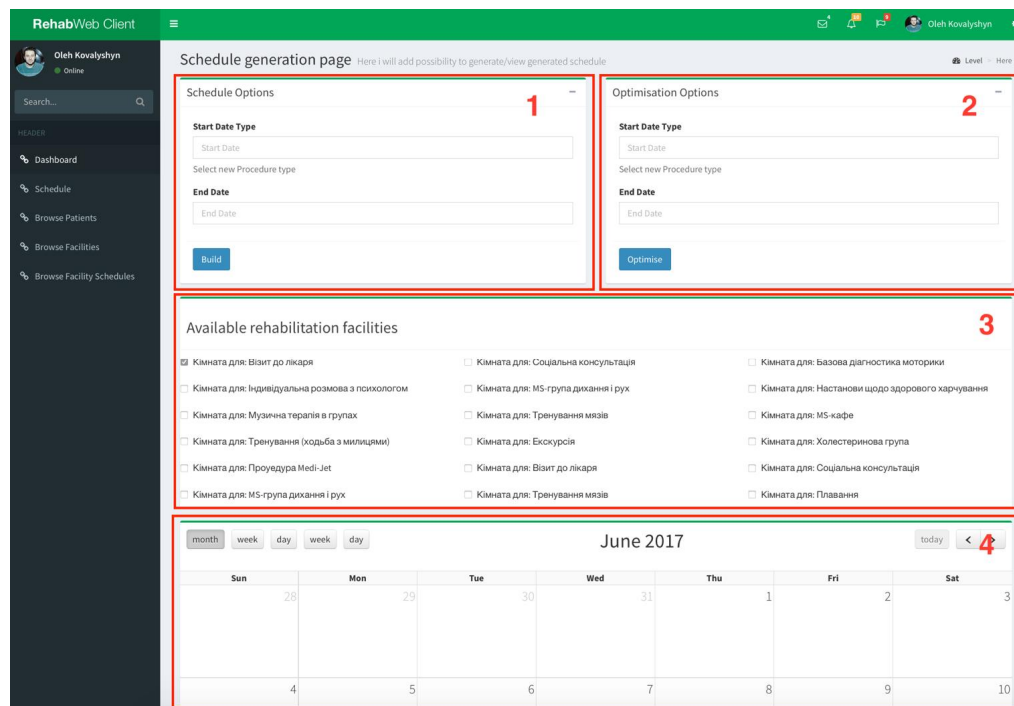


Рис. 2. Графічний користувацький інтерфейс побудови та оптимізації розкладів відновлювальної терапії

Розклади отримані в процесі побудови та оптимізації автоматично зберігаються до бази даних, та стають автоматично доступними всім користувачам системи з відповідними правами доступу.

Висновки. 1. Враховуючи надзвичайну складність урахування всіх обмежень до планів реабілітації клінік, підвищення рівня вдовolenості їх пацієнтів від отриманих медичних послуг особливу увагу необхідно приділяти автоматизації процесу його складання.

2. Запропонований програмний комплекс є основою інформаційної технології автоматизованої побудови та оптимізації розкладів відновлювальної терапії. В ході його реалізації було використано наступні технології: мова програмування високого рівня Java, нереляційна база даних MongoDB, платформа Java Enterprise Edition, Spring Framework.

3. Використана для реалізації програмного комплексу архітектура "тонкого клієнта" забезпечує його універсальність, незалежність від апаратної архітектури та операційних систем клієнтів, простоту додавання нового функціоналу, а також високу безпеку та стійкість проти відмов.

4. Наведений опис основних вікон та елементів інтерфейсу користувача служить для клієнтів рекомендаційним та методичним матеріалом, який дозволяє їм використовувати розроблений програмний комплекс.

1. Епифанов В. А., Епифанов А. В. Реабилитация в неврологии/ В.А. Епифанов.- ГЭОТАР-Медиа, 2014.- 416 с.
2. Іпатов. А. В. Основні показники інвалідності та діяльності медико-соціальних експертних комісій України за 2014 рік. / А. В. Іпатов, О. М. Мороз, В. А. Голик, Р. Я. Перепелична, І. Я. Ханюкова, Ю. І. Коробкін, Р. М. Молчанов, Г. М. Маловичко, Н. О. Гондуленко, Н. А. Саніна // Основні показники інвалідності та діяльності медико-соціальних експертних комісій України за 2014 рік: Аналітико-інформаційний довідник / За ред. начальника відділу медико-соціальної експертизи Департаменту медичної допомоги МОЗ України С. І. Черняка. – Дніпропетровськ: Роял- Принт, 2015. – 167 с.
3. Бурдюк В.Я., Шкурба В.В. Теория расписаний. Задачи и методы решений / В.Я. Бурдюк // Кибернетика. – 1971. – № 1. – С. 89-102.
4. Гафаров Е.Р., Лазарев А.А. Доказательство NP-трудности частного случая задачи минимизация суммарного запаздывания для одного прибора/ Е.Р. Гафаров //Известия АН: Теория и системы управления. – 2006.– № 3.– С. 120–128.
5. Лазарев А.А., Кварацхелия А.Г., Гафаров Е.Р. Алгоритмы решения NP-трудной проблемы минимизации суммарного запаздывания для одного прибора / А.А. Лазарев // Доклады Академии Наук. – 2007.– Т. 412.– № 6. – С. 739–742.
6. Коффман Э.Г. Теория расписаний и вычислительные машины // Э.Г. Коффман – М.: Наука, 1984. – 335 с.

7. Сидорин А. Б. Методы автоматизации составления расписания занятий. Часть 2. Эвристические методы оптимизации /А.Б. Сидорин, Л.В. Ликучева, А.М. Дворянкин // Известия ВолгГТУ. – Волгоград, 2009. – №12(60). – С. 120-123.