**Вступ**

Об’єктом дослідження даної курсової роботи є інформаційно-пошукова система, а також прогнозування поведінки користувача, для створення зрозумілого інтерфейсу.

Предметом дослідження є архітектури і принцип побудови пошукової системи ш її інтерфейсу на С#, реалізація обробки даних, управління базою даних.

Метою дослідження є створення пошукової системи для обробки, класифікації інформації та здійснення пошуку за даними, а також створення інтерфейсу та бази даних для взаємодії з системою.

Згідно зі свідченнями опитування, проведеному на Stack Overflow, C # займає одне із передових місць в списку найбільш популярних технологій. До переваг можна віднести те, що це об'єктно-орієнтована, проста і в той же час потужна мова програмування, яка дозволяє розробникам створювати багатофункціональні програми, вона об'єднує кращі ідеї сучасних мов програмування та завдяки великому розмаїттю синтаксичних конструкцій і можливості працювати з платформою .Net, C # дозволяє швидше, ніж на будь-який іншій мові, розробляти програмні рішення.

Обробка та робота з даними – завдання реалізація, яких лише за рахунок людської сили, вимагає багато часу та уважності. Розробка певного програмного продукту, що працює з наявними даними, значно полегшує роботу та запобігає виникненню механічних помилок. Особливо швидкість і коректність виконання цього завдання необхідно у медичній сфері, де від цього залежить благополуччя людини.

**РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ**

1.1 Аналіз задачі, засобів та методів її вирішення

Інформаційно –пошукові системи (ІПС) — це різновид автоматизованих інформаційних систем, що призначені для пошуку записів в сховищах (базах даних) за формальними характеристиками. Тому в роботі ІПС можна виділити такі основних етапи: перший — збір і зберігання інформації, другий — пошук і видача інформації користувачам, а також редагування, створення нових та видалення старих записів.

Для реалізації цього програмного продукту, перед нами стоять задачі:

1. Спроектувти та заповнити базу даних.

Для того, щоб досягти максимальної організованості структури даних, їх надійного збереження та полегшеного доступу для зміни і обробки взаємозалежної інформації великих обсягів, ми створимо БД та заповнимо її тестовими значеннями.

1. Створити основні класи

Для подальшої роботи створимо всі необхідні класи, на основі таблиць з нашої БД, починаючи від найпростіших, ні від кого не залежних структур та створимо основні поля та конструктори, для їх заповнення.

1. Обрати інтерфейс

Створимо проект на основі Windows Forms, що буде відповідальний за графічний інтерфейс користувача, для спрощення доступу до всіх реалізованих функцій.

1. Створити головну структуру інтерфейсу

Для запобігання перенавантаження інформацією на одній сторінці та для полегшення доступу до всього функціоналу створимо меню з декількома підрівнями та реалізуємо можливість вільно пересуватися між ними, завдяки відкриттю нових вікон. Додамо на форму основні інструменти, для роботи з інформацією.

1. Введення даних з клавіатури.

Використаємо стандартні текстові поля та випадіючі списки, для того, щоб користувач мав змогу створити новий запис, заповнивши всі необхідні поля, у випадку пацієнта, це: прізвище, ім'я, по батькові, дату народження, номер телефону, домашня адреса, номер медичної карти, обрати стан та можливі пільги.

1. Виведення даних

Для подальшого аналізу даних ми повинні реалізувати їх виведення, як для всіх елементів з таблиці БД у скороченому вигляді через data grid view (що також дозволить нам сортувати знайденні данні), так і повну інформацію про один запис, який буде об’єднувати у собі данні з різних таблиць нашої БД.

1. Пошук записів по параметрам

Нашою наступною задачею буде можливість пошуку запису у таблиці за певними, обраними через випадаючий список параметрами та відображення їх у вигляді структурованої таблиці.

1. Збереження нового або відредагованого запису

Якщо ми хочемо, щоб всі данні, що вводить користувач, запам’ятовувались, для їх подальшого відтворення та роботи, ми повинні реалізувати збереження записів у БД, за допомогою звернення до неї та передавання певного запиту.

1. Видалення даних

Також, створимо функцію для видалення певного, обраного користувачем, запису з БД.

1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту (роботи).

Перед початком роботи над програмним додатком, ми проаналізували та визначили переваги та недоліки інших схожих сервісів. Оцінювання ми проводили за такими критеріями, як: різноманітність даних, підтримка бази, інтерфейс, різноманітність функціоналу.

medpoisk.ru

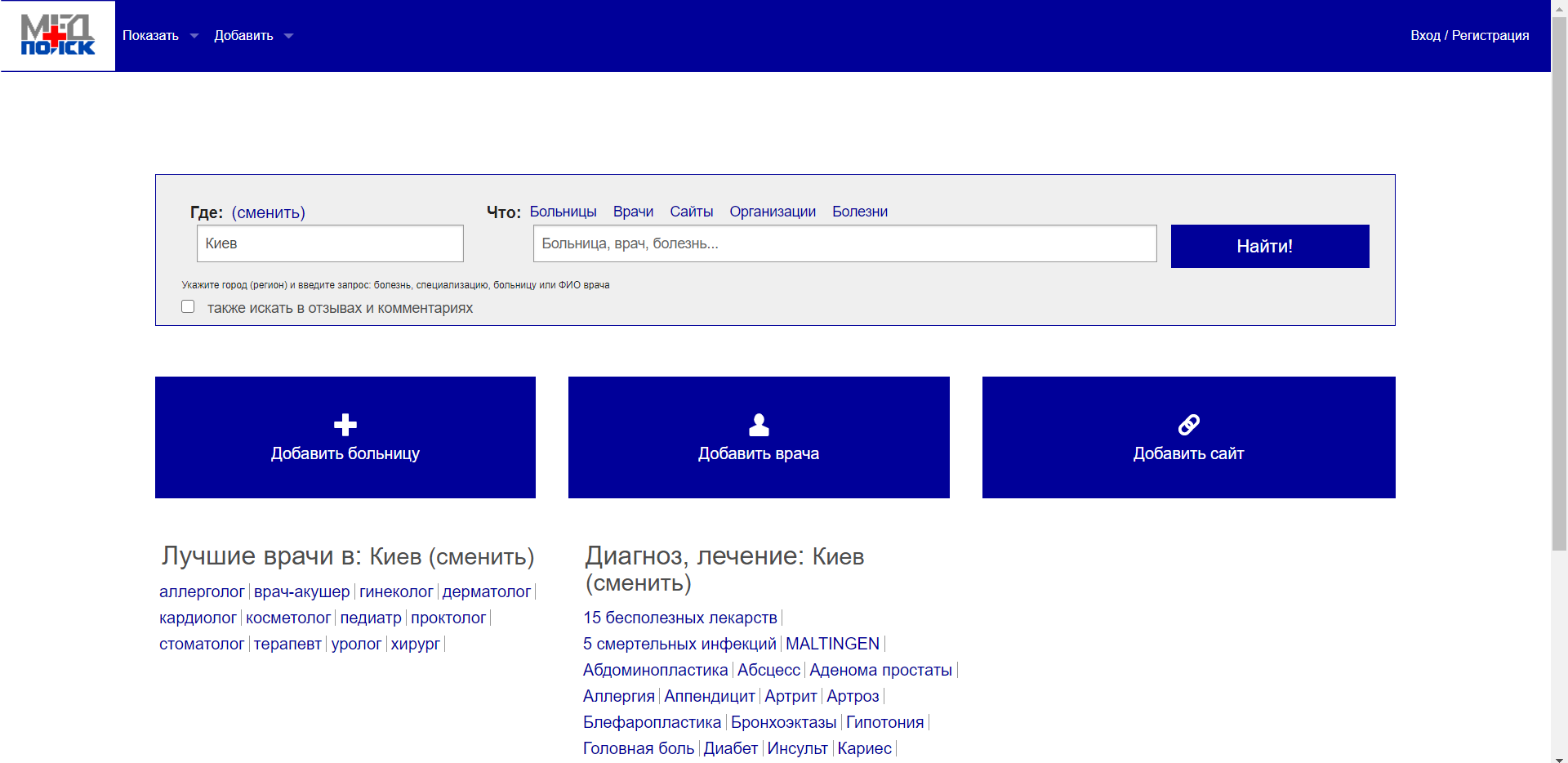
Посилання: <https://www.medpoisk.ru/>

Медичний онлайн-сервіс medpoisk. ru надає можливість шукати лікарів з різних міст, поліклінік та лікарень та різних фахів. Дозволяє оцінити можливості спеціаліста, записатися до нього на прийом та залишити відгук.

Для модуля пошуку ліків розроблена система обміну даними з різними обліковими системами аптек, щоб користувачі мали завжди актуальні ціни на ліки і їх наявність в конкретній аптеці.

З переваг цього сайту можна відзначити велику кількість даних, зрозумілий та простий інтерфейс.

До недоліків, можемо віднести, можливість додати будь-яку людину у ролі лікаря без перевірки даних.

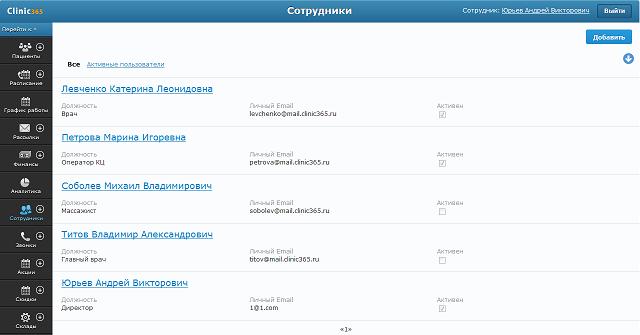


**Clinic365**

Clinic365 дозволяє вести облік персоналу клініки (кадрову інформацію). Для кожного співробітника в Clinic365 реєструється графік робіт, який в подальшому використовується для запису пацієнтів. Заповнення картки візиту. Створення історії хвороби. Залежно від завдань конкретної клініки Clinic365 дозволяє налаштувати необхідні типи документів. Направлення на додаткові консультації, діагностику та обстеження.

Безумовними перевагами цього сервісу є широкий функціонал і, завдяки біль вузькому колу інформації, можливість легше підтримувати базу даних.

Недоліками є перенавантажений інтерфейс програмного додатку.

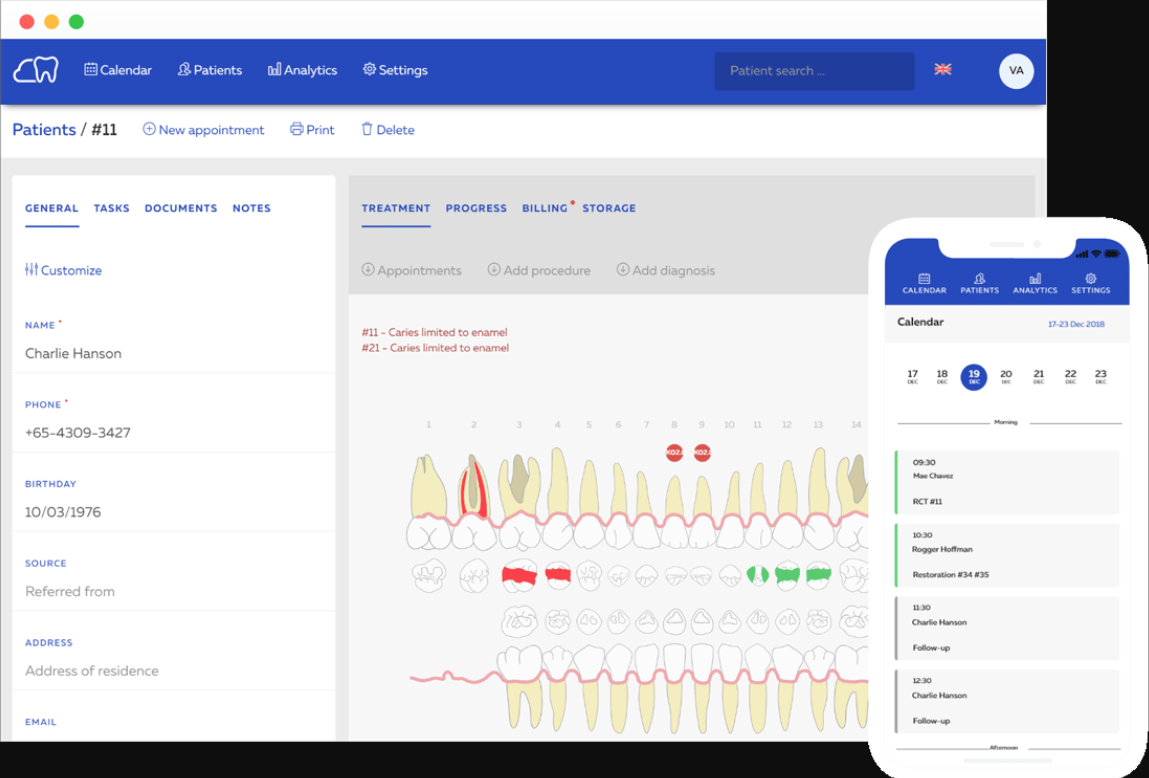


Dentaltap

DentalTap - безкоштовна хмарна платформа управління клінікою і практикою лікаря. Продукт поєднує у собі три ключових напрямки роботи - управління офісними завданнями, лікуванням і комунікації з пацієнтами. Ключові розділи платформи підвищують ефективність роботи стоматологічної практики через автоматизацію управління роботи з пацієнтами (CRM), планування лікування, спільну роботу, документообіг, рахунків і оплат, матеріалів, налаштування прав доступу, даних і аналітики. Особливу увагу при розробці приділили плануванню лікування і зробили його простим і інформативним.

До переваг цього продукту можемо віднести зручний інтерфейс, легку підтримку БД та широкий функціонал.

Вузьку направленість цієї хмарної платформи, можна визначити, як один із недоліків.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **medpoisk.ru** |  |  |  |  |
| **Clinic365** |  |  |  |  |
| **Dentaltap** |  |  |  |  |

2.1 Проектування загального алгоритму роботи програми

Навести загальний алгоритм роботи програми та схеми.

2.2 Розробка функціональних алгоритмів роботи програми

2.3 Розробка програмного забезпечення

Перед початком проектування пошуково-інформаційної системи, ми зібрали та проаналізували всі данні, які необхідні для повноцінної роботи реєстратури поліклініки. Далі за допомогою онлайн сервісів ми змоделювали нашу структуру даних, створили основні таблиці та визначили зв’язки між ними.

Після завершення концептуального проектування ми створюємо БД на основі СУБД SQLitle. Для спрощення та візуалізації роботи використовуємо DB Browser for SQLite.

Створюємо основні таблиці, визначаємо зв’язки між ними та заповнюємо тестовими данними.