**Форма № 17**

Міністерство освіти і науки України

**Національний університет «Запорізька політехніка»**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кафедра програмних засобів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(найменування кафедри)

**КУРСОВИЙ ПРОЄКТ**

**(РОБОТА)**

з дисципліни «Об’єктно-орієнтоване програмування»

(назва дисципліни)

на тему: «Чат-бот пошуку стоматологічних послуг»

Студентів 2 курсу КНТ-219 групи

спеціальності 122 Комп’ютерні науки та технології

освітня програма (спеціалізація) Комп'ютерні науки

\_\_\_\_\_\_\_\_\_\_\_\_Жир Є.С.\_\_\_\_\_\_\_

(прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_Примаков В. В.\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_Кобяковський О. І.\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_ Пархоменко А. Д.\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник доцент, к.т.н., Табунщик Г. В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Табунщик Г. В.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Каплієнко Т. І.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Миронова Н. О.

(підпис) (прізвище та ініціали

2020 рік

**Форма № 25**

Міністерство освіти і науки України

**Національний університет «Запорізька політехніка»**

( повне найменування закладу вищої освіти)

Інститут, факультет ІІРЕ, ФКНТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра програмних засобів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ступінь вищої освіти      бакалавр**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Спеціальність 122 Комп’ютерні науки та технології **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(код і найменування)

Освітня програма (спеціалізація) Комп’ютерні науки**\_\_\_\_\_\_\_\_**

(назва освітньої програми (спеціалізації)

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри ПЗ, д.т.н, проф.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.О. Субботін**

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_року

З А В Д А Н Н Я

**НА КУРСОВИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

Жир Є.С., Примаков В. В., Кобяковський О. І., Пархоменко А. Д.

(прізвище, ім’я, по батькові)

1. Тема проєкту (роботи) «Чат-бот пошуку стоматологічних послуг»

керівник проєкту (роботи)\_ Табунщик Галина Володимирівна, к.т.н., доцент,

( прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від

2. Строк подання студентом проєкту (роботи)\_22 грудня 2020 року\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до проєкту (роботи) створити застосунок «Чат-бот пошуку стоматологічних послуг»

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області; 2. Аналіз програмних засобів; 3. Основні рішення з реалізації компонентів системи; 4. Керівництво програміста; 5. Керівництво користувача; 6. Додатки.

6. Консультанти розділів проєкту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада  консультанта | Підпис, дата | |
| завдання видав | прийняв  виконане завдання |
| 1-5 Основна частина | Каплієнко Т.І., доцент |  |  |
|  |  |  |  |
|  |  |  |  |

7. Дата видачі завдання\_\_20 вересня 2020 р.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів курсового  проєкту (роботи) | Строк виконання етапів проєкту  ( роботи ) | Примітка |
| 1. | Аналіз індивідуального завдання. | 1 тиждень |  |
| 2. | Аналіз програмних засобів, що будуть використовуватись в роботі. | 2 тиждень |  |
| 3. | Аналіз структур даних, що необхідно використати в курсової роботі. | 3 тиждень |  |
| 4. | Затвердження завдання | 4 тиждень |  |
| 5. | Вивчення можливостей програмної реалізації структур даних та інтерфейсу користувача. | 5-9 тиждень |  |
| 6. | Аналіз вимог до апаратних засобів | 9 тиждень |  |
| 7. | Розробка програмного забезпечення | 9-13 тиждень |  |
| 8. | Проміжний контроль | 10 тиждень | Розділи 1-2 ПЗ |
| 9. | Оформлення, відповідних пунктів пояснювальної записки. | 10-14 тиждень | Розділи 1-5 ПЗ |
| 10. | Захист курсової роботи. | 15 тиждень |  |

**Студент (ка) \_\_\_\_\_\_\_\_\_ \_\_\_** Жир Є.С.**\_\_\_\_\_\_\_\_\_\_\_**

( підпис ) (прізвище та ініціали)

**Студент (ка) \_\_\_\_\_\_\_\_\_ \_\_\_\_** Примаков В. В.**\_\_\_\_\_**

( підпис ) (прізвище та ініціали)

**Студент (ка) \_\_\_\_\_\_\_\_\_ \_** Кобяковський О. І.**\_\_\_\_\_**

( підпис ) (прізвище та ініціали)

**Студент (ка) \_\_\_\_\_\_\_\_\_ \_** Пархоменко А. Д.**\_\_\_\_\_**

( підпис ) (прізвище та ініціали)

**Керівник проєкту (роботи) \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_** Табунщик Г. В.**\_\_\_\_**

( підпис ) (прізвище та ініціали)

**РЕФЕРАТ**

Об’єкт дослідження – програмні засоби для створення програмного додатку.

Предмет дослідження – принципи створення програмного додатку (чат-бот пошуку стоматологічних послуг).

Мета роботи – створення програмної системи для пошуку стоматологічних послуг.

У даній роботі проведено дослідження структури бібліотеки контейнерів Qt. Розглянуто клас QList - для формування списку лікарів за заданими параметрами. Розглянуто особливості підключення JSON-файлів. Розглянуто класи інтерфейсу Qt Creator. Створення власних структур даних та алгоритмів для роботи з ними. Створення графічного додатку для пошуку стоматологічних послуг. Навчитися працювати з принципами ООП (інкапсуляція, поліморфізм і успадкування).

**ЗМІСТ**

С.

[Перелік скорочень та умовних познак](https://docs.google.com/document/d/1C2Xc73uPoA72PPNYx4l9OkvhRELdsH-3/edit#heading=h.1t3h5sf) 6

Вступ 7

1 Аналіз предметної області 8

2 Аналіз програмних засобів 11

3 Основні рішення з реалізації компонентів системи 12

3.1 Інтерфейс

3.2 Підключення файлів

3.3 Алгоритми та структури даних

3.4 Виключні ситуації

4 Керівництво програміста 17

5 Керівництво користувача 18

Висновки 21

Перелік джерел посилання 22

Додаток А Текст програми 23

[**ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**](https://docs.google.com/document/d/1C2Xc73uPoA72PPNYx4l9OkvhRELdsH-3/edit#heading=h.1t3h5sf)

Qt Creator –інтегроване середовище розробки, призначене для створення крос-платформових застосунків з використанням бібліотеки Qt;

JSON (JavaScript Object Notatio) - простий формат обміну даними, зручний для читання і написання як людиною, так і комп'ютером;

QList – один із загальних класів контейнерів Qt. Він зберігає елементи у списку, що забезпечує швидкий доступ на основі індексу та вставки та видалення на основі індексу;

ООП – об'єктно-орієнтоване програмува́ння;

STL (Standard Template Library) – стандартна бібліотека шаблонів в мові програмування C ++;

QtWidgets – містить класи для додатків на основі віджетів, модуль виділений з QtGui в Qt 5;

QString – є рядком символів Unicode. QString зберігає рядок 16-бітних QChar, де кожен QChar зберігає символ Unicode 4.0;

Unicode – це міжнародний стандарт, який підтримує більшість існуючих на сьогодні писемних систем;

ASCII (American standard code for information interchange) - назва таблиці (кодування, набору), в якій деяким поширеним друкованим і недрукованих символів зіставлені числові коди;

Latin-1 – кодова сторінка, призначена для західноєвропейських мов;

Set-ери – метод, який використовується в об'єктно-орієнтованому програмуванні для того, щоб привласнити яке-небудь значення інкапсульованому полю;

Get-ери – спеціальний метод, що дозволяє отримати дані, доступ до яких безпосередньо обмежений.

**ВСТУП**

У роботі необхідно виконати наступні завдання:

1) аналіз предметної області;

2) розробити відповідні структури даних;

* + - * хеш-таблиці
      * клас Стоматолог
      * клас Стоматологія

3) створити візуальний інтерфейс;

4) розробити програму;

5) провести тестування;

6) оформити пояснювальну записку.

**1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

Головне завдання програми - пошук стоматологів за певними критеріями. Перед розробкою потрібно подивитися структуру аналогічних програм. Як аналог було розглянуто веб-сервіс "helsi.me" виходячи з якого розробленi структури класів.

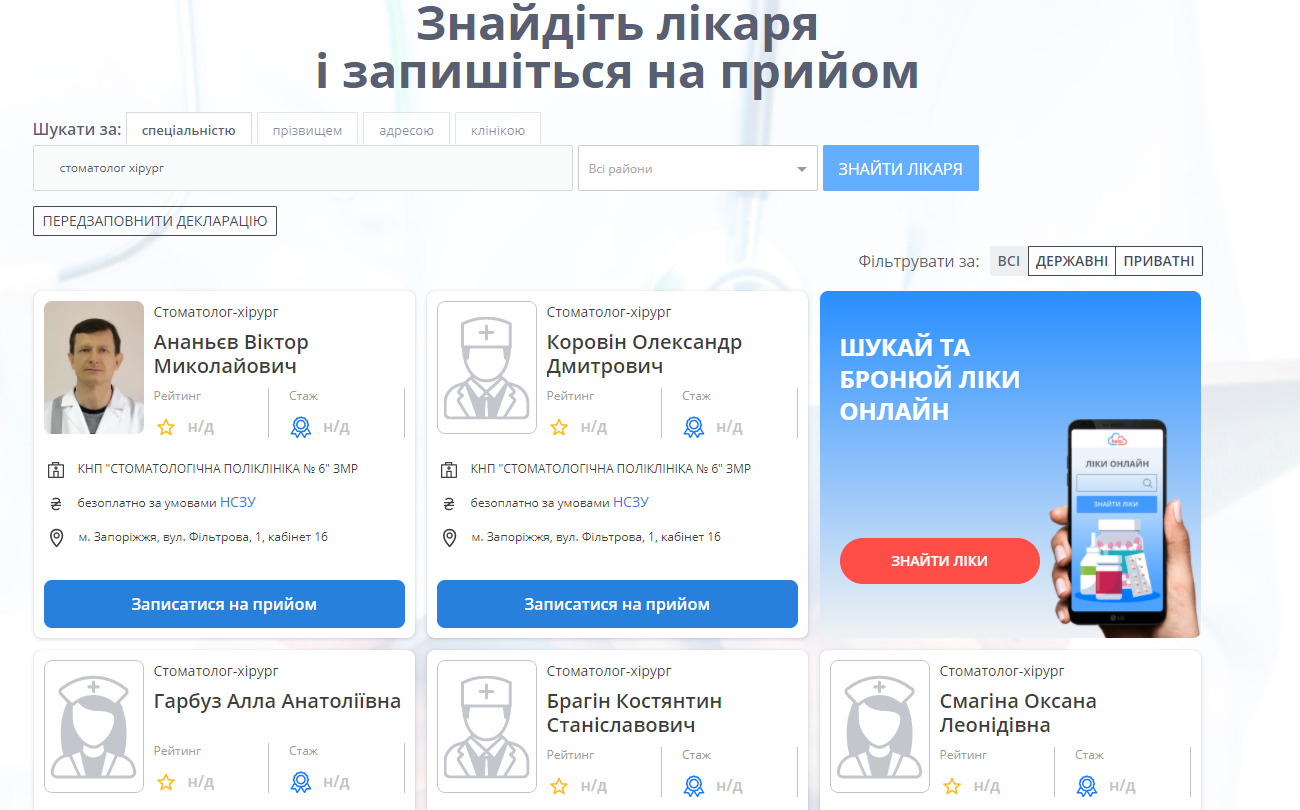


Рисунок 1 - Аналог – веб-сервіс пошуку стоматологів

Існують стоматологічні клініки до яких прихильність стоматологи, стоматологи мають свою спеціалізацію яка займається певними стоматологічними послугами. У свою чергу стоматологи повинні мати такі обов'язкові поля дані:

* ПIБ;
* спеціалізація;
* вік;
* стаж роботи;
* фото;
* контакти;
* стоматологія.

Також ми додали додаткові поля:

* про дантиста;
* коментарі;
* рейтинг.

Рейтинг підраховується виходячи з коментарів. Буде потрібно додатковий клас Коментар який має поля:

* ім'я відправника;
* сам коментар;
* оцінка.

Обов'язковою класом є стоматологія яка має такі дані:

* назва стоматології;
* адреса;
* робочий час;
* контакти;
* про стоматологію.

Виходячи з цих класів для компактного зберігання даних потрібно створити поля у стоматологів і стоматологій - ID стоматології, завдяки якому стоматолог буде прив'язаний до певної стоматології. А коментарі будуть прив'язані безпосередньо до певного стоматолога.

Значить функціонал програми повинен бути таким:

1. вміти шукати стоматологів за критеріями:
   * прізвище;
   * стоматологія;
   * спеціальність.
2. залишати коментарі до стоматолога;
3. автоматично підраховувати рейтинг лікаря виходячи з коментарів;
4. виведення всіх лікарів за певним критерієм;
5. сортування знайдених лікарів за рейтингом і за алфавітом.

Оскільки головне завдання додатку - пошук стоматологів значить при запуску програми повинна відкриватися форма в якій вже можна шукати стоматологів. Список знайдених стоматологів повинен виводиться окремим блоком і на головній формі повинна бути можливість вибору сортування і за яким критерієм буде працювати пошук і звичайно ж текстове поле для введення критерію (наприклад прізвища стоматолога). Якщо користувач не може визначиться з пошуком повинна бути функція виведення всіх стоматологів конкретної спеціалізації або стоматології.

Обов'язкова повинна бути форма "Допомога" в якій описується функціональність програми і як з нею працювати. Оскільки лікарі мають відносно багато даних, то при виводі списку лікарів повинні відображатися тільки основні поля для зручного перегляду для користувача. Для кожного знайденого лікаря повинна бути добавлена форма в якій буде виводиться повна інформація про стоматолога і стоматології в якій він працює.

Для зручності у кожного лікаря буде форма "Коментарі" в якій користувач зможе дивитися і додавати коментарі конкретного стоматолога.

**2 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ**

Основне завдання Qt Creator - спростити розробку програми за допомогою фреймворка Qt на різних платформах. Тому серед можливостей, властивих будь-якому середовищі розробки, є вбудований дизайнер інтерфейсів на QtWidgets.

Бібліотека Qt надає набір контейнерних класів загального призначення заснованих на шаблонах. Ці класи можуть використовуватися для зберігання елементів певного типу.

Ці контейнерні класи розроблені для легкого, безпечного і простого використання замість контейнерів STL.

Серед контейнерів для зберігання класів "Стоматолог" і "Стоматологія" використовується контейнер QList оскільки під час роботи програми постійно будуть додаватися елементи в структуру, то QList підходить для оптимізації додатку оскільки він моментально може додати елемент в контейнер на відміну від інших контейнерів, що значно прискорює працездатність програми.

Мова C ++ надає два типи рядків: традиційні рядки мови C - масиви символів, що завершуються символом '\ 0' і клас string. Qt надає набагато більш потужний клас QString. Він призначений для зберігання рядків з 16-ти бітними символами Unicode. Unicode містить набори символів ASCII і Latin-1 з їх звичайними числовими значеннями. Але оскільки кожен символ в QString представлений 16-ма бітами, він може містити тисячі інших символів. Завдяки використання класу QString можна зручно працювати з інтерфейсними класами QT, а так само створити багатомовний додаток в якому алгоритми будуть коректно працювати завдяки Unicode.

Клас QWidget надає базову можливість для відтворення на екрані і для обробки подій для користувача введення. Всі елементи призначеного для користувача інтерфейсу, що надаються Qt, є підкласами QWidget або використовуються в поєднанні з підкласом QWidget. Створення призначених для користувача віджетів виконується успадкуванням від QWidget або відповідного підкласу і перевизначення віртуальних обробників подій. Компонування - елегантний і гнучкий спосіб для автоматичного розміщення дочірніх віджетів всередині контейнера. Кожен віджет повідомляє компонувальнику свої вимоги до розміру за допомогою властивостей sizeHint і sizePolicy, а компоновщик відповідно розподіляє доступний обсяг простiру. За допомогою підкласів QWidget і компонувань буде реалізований графічний інтерфейс і функціональність програми.

**3 ОСНОВНІ РІШЕННЯ З РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ**

**3.1 Інтерфейс**

Прийнято рішення, щоб всі функції стосовно пошуку стоматологів розміщувалися на головній формі, а також блок в якому виводиться сам список. У кожного знайденого лікаря потрібно щоб створювалися дві кнопки – “Детальніше” та “Коментарі” при натисканні на які будуть відкриватися відповідні форми і в них передаватися дані про лікаря у якого були нажаті ці кнопки. У формі “Детальніше” виводиться повна інформація про лікаря та лікарню в якій він працює. У формі “Коментарі” виводяться всі коментарі конкретного лікаря та форма в якій можливо написати та відправити коментар. В блоці menubar знаходитимуться три меню-вкладки – “Меню” “Мова” “Показати всі”. В першій вкладці можливо відкрити форму “Допомога” та “Розробники”, де відповідно виведеться коротка інформація про програму та таблиця зі списком спеціалізацій стоматологів та послуг, які відносяться до певної спеціалізації. У другій вкладці можливо вибрати мову. А в третій вивести всіх стоматологів, стоматологій та спеціалізацій.

В усій програмі використовувалося два стилі елементів:

1. Перший стиль:

* color: black;
* background-color: rgb(199, 199, 199);
* font: 14pt "Arial".

1. Другий стиль:

* background-color: rgb(72, 60, 72);
* font: 12pt "Arial";
* color: white.

Перший стиль можливо побачити в блоці повідомлень для користувача на рис. 3.1, а другий в QLabel (синій колір на рис. 3.2). Використання двух стилів дає можливість створити цікаве рішення щодо дизайну програми.

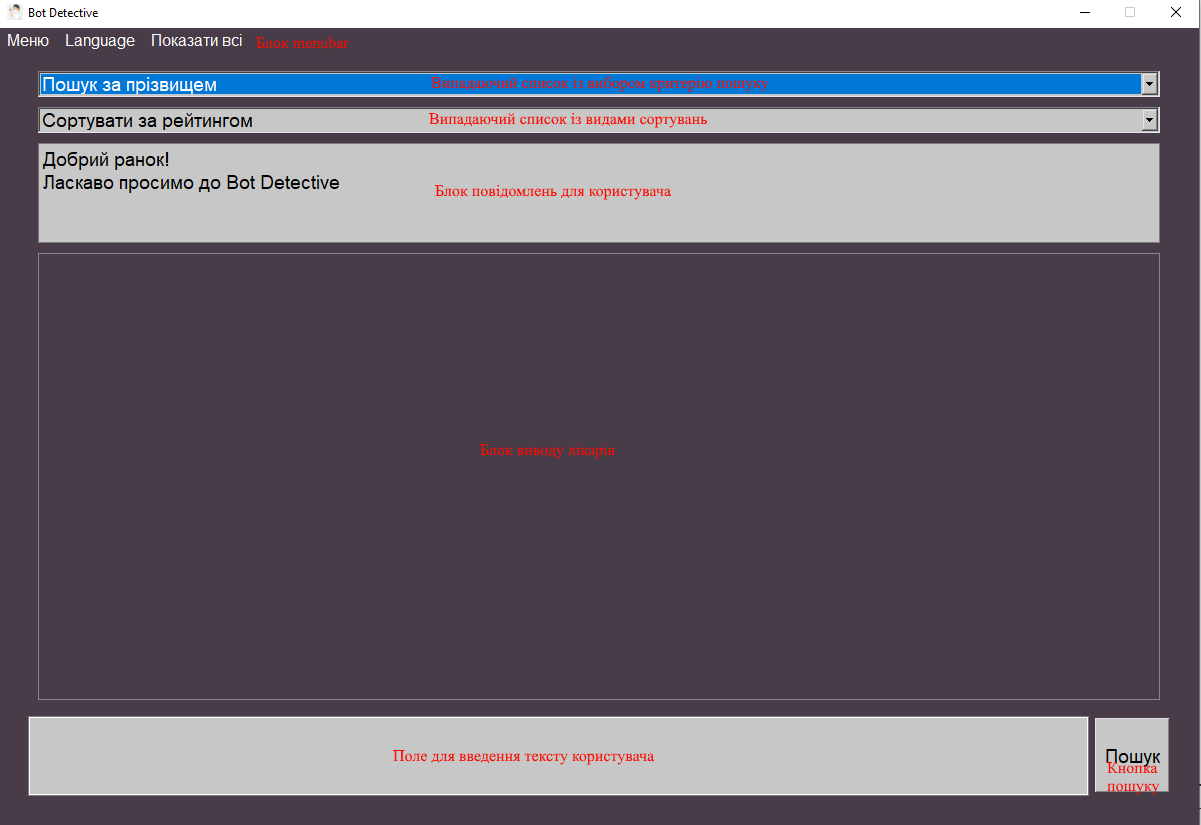


Рисунок 3.1 - Головна форма програми

На рис. 3.1 зображується структура головної форми програми. Випадаючі списки реалізовані за допомогою класу QComboBox. Використовуються його методи для додавання елементів до списку, повернення номеру вибраного елементу, як раз за допомогою цього методу в алгоритмах буде знаходитися за яким критерієм шукати та як сортувати список стоматологів. А також при зміні мови використовується метод очистки випадаючого списку. В блоці повідомлень використовується клас QTextBrowser, який дозволяє додавати до нього інформацію у вигляді рядків та редагувати текст за допомогою розмітки HTML та CSS-стилів. Блок виводу лікарів – це клас QScrollArea, який дозволяє встановлювати компонування та віджети, на основі цих методів створена функція для виводу стоматологів. Поле для введення тексту користувача – це клас QLineEdit, який дозволяє вводити текст та передавати його до програми. І звичайна кнопка QPushButton, яка має сигнал clicked і дозволяє виконувати функції пошуку в програмі. Блок menubar – це клас QMenuBar, який розділений на класи QMenu в яких знаходяться класи QAction, які вже дозволяються за допомогою сигналу triggered викликати певні функції.

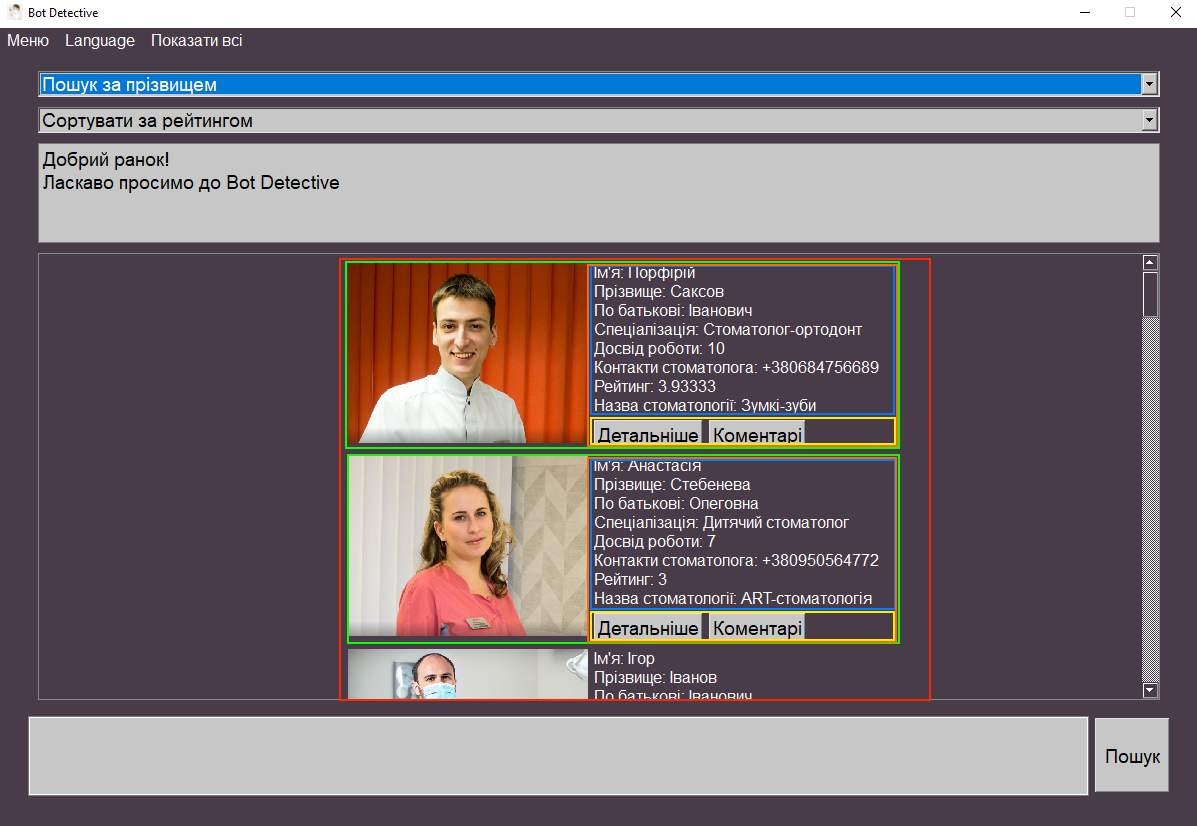


Рисунок 3.2 - Створення списку стоматологів

На рис. 3.2 схематично зображено, як створюється список знайдених лікарів. Спочатку до методу пересилається QList<QDentist\* >, створюється новий QWidget та головне компонування по вертикалі QVBoxLayout (червоний колір на рис. 3.2). Потім в циклі для кожного лікаря створюється компонування по горизонталі QHBoxLayout (зелений колір на рис. 3.2) до якого додається QLabel в якому зберігається фото лікаря, яке створюється за допомогою класу QPixmap і встановлюється в QLabel за допомогою методу setPixmap. Додатково створюється компонювання по вертикалі QVBoxLayout (помаранчевий колір на рис. 3.2), яке додається до зовнішнього QHBoxLayout (зелений колір на рис. 3.2), створюється QLabel (синій колір на рис. 3.2) до якого заносяться дані про лікаря і додається до зовнішнього QVBoxLayout (помаранчевий колір на рис. 3.2). Створюється компонування по горизонталі QHBoxLayout (жовтий колір на рис. 3.2) до якого додаються дві кнопки “Детальніше” та “Коментарі”, ці кнопки з’єднуються з методом для посилання лікаря до нової форми:

* connect(detailed, &QPushButton::clicked, [this, tmp](){ this->SendDetailed(tmp); });
* connect(comments, &QPushButton::clicked, [this, tmp](){ this->SendComments(tmp); });

QHBoxLayout (жовтий колір на рис. 3.2) додається до зовнішнього QVBoxLayout (помаранчевий колір на рис. 3.2). На цьому етапі створено QHBoxLayout (зелений колір на рис. 3.2) для лікаря і це компонування додається до головного QVBoxLayout (червоний колір на рис. 3.2) і так для кожного лікаря в списку. Після відпрацювання циклу для кожного лікаря в QWidget встановлюється головне компонування QHBoxLayout (зелений колір на рис. 3.2). І вже в кінці до блоку виводу лікарів (рис 3.1) встановлюється створений віджет.

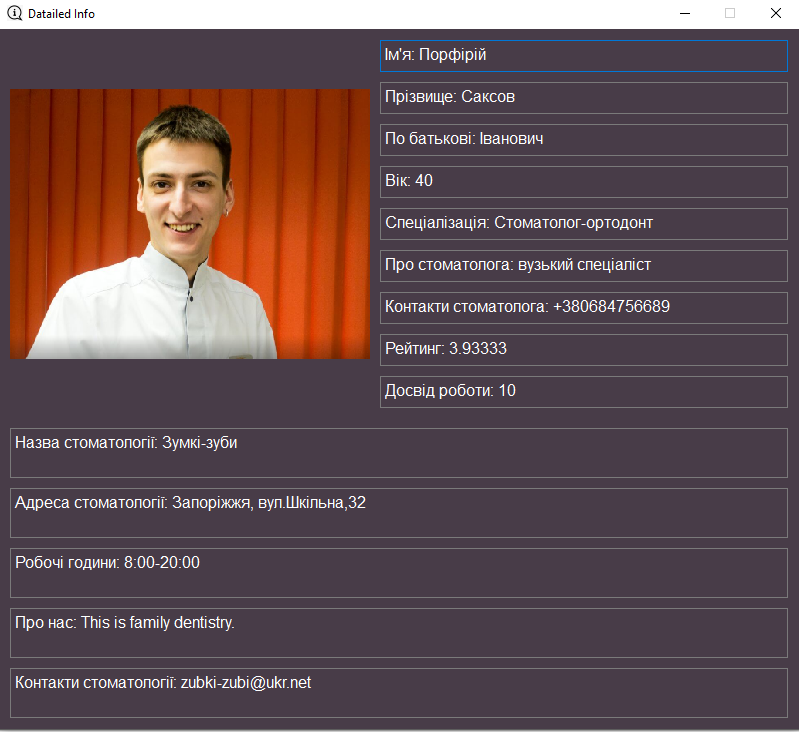


Рисунок 3.3 – Форма “Детальніше”

На рис. 3.3 показана форма “Детальніше”, яка відкривається під час натискання на кнопку “Детальніше” на рис3.2. В цій формі за допомогою компонувань та QTextBrowser-ів виводиться вся інформація про лікаря та стоматологію в якій він працює.

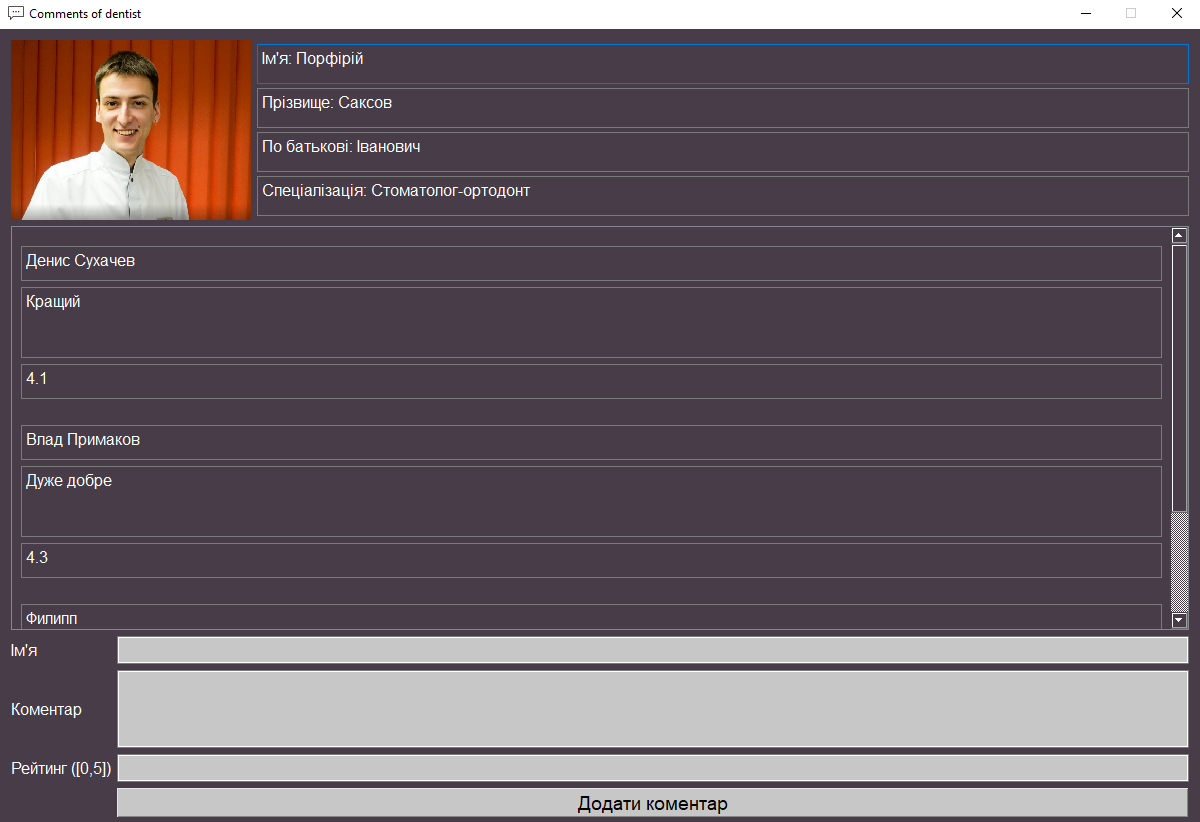


Рисунок 3.4 – Форма “Коментарі”

В цій формі виводиться ПІБ та спеціалізація лікаря, а також за аналогією створення списку стоматологів створюється список коментарів в QScrollArea а також форма в якій можна залишити коментар для конкретного лікаря.

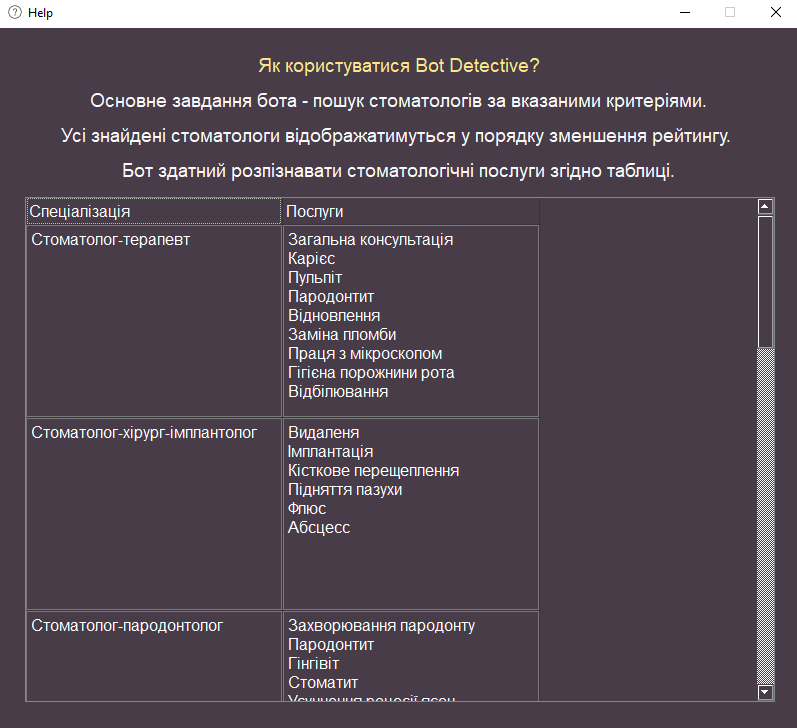


Рисунок 3.5 – Форма “Допомога”

В блоці menubar на рис. 3.1 можна відкрити форму “Допомога” в якій знаходиться інформація про користування програмою та таблиця зі спеціалізаціями та відповідними їм послугами, за якими можливий пошук лікарів.

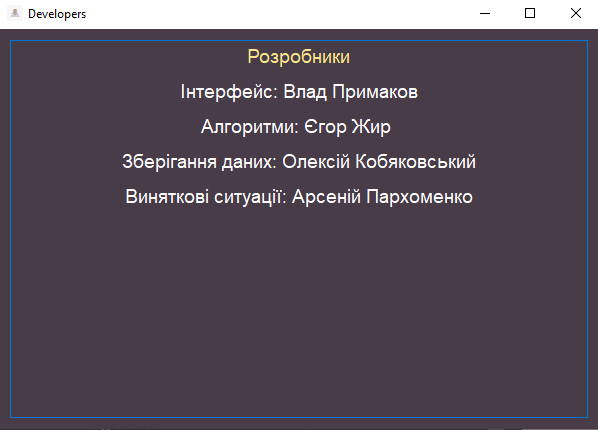


Рисунок 3.6– Форма “Розробники”

В тому ж блоці menubar на рис. 3.1 можна відкрити форму “Розробники” в якій виводяться безпосередньо розробники цієї програми

**3.2 Підключення файлів**

JSON (англ. JavaScript Object Notation, укр. Запис об'єктів*JavaScript*, вимовляється *джейсон*) — це текстовий формат обміну даними між комп'ютерами.

Ми обрали цей формат, тому що з ним зручно працювати в IDE Qt Creator. Qt Creator має спеціальні класи для JSON, такі як: QJsonObject, QJsonParseError, QJsonArray. Ми написали файли JSON, на двох мовах: українській та англійській. На рисунку 2, файли з закінченням eng написані на англійській, а ukr – на українській.

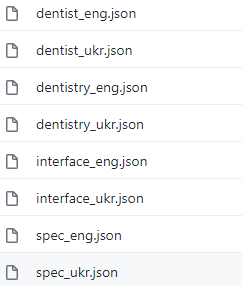


Рисунок 3.7- каталог JSON-файлів.

Структура JSON-файлів.

На рисунку приклад одного JSON-об’єкту в масиві. Дані в JSON записуються таким чином: "key" : "value". Цифрою 2 позначені ключі, 1 і 3 –значення. 1-початок массиву.

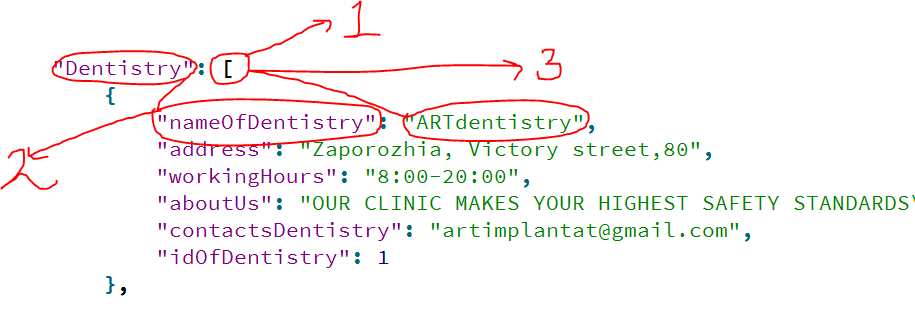


Рисунок 3.8 - Структура JSON-файлів.

Робота з даними в проекті ведеться за допомогою ключів. Знизу приклад.



Рисунок 3.9 –приклад роботи з ключами в С++.

Рейтинг встановлюються з урахуванням коментарів, нижче фрагмент коду.

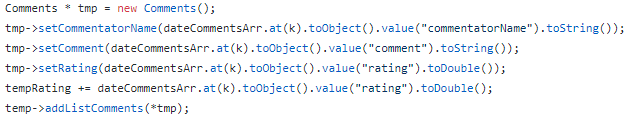


Рисунок 3.10 – фрагмент коду, де через коментарі ведеться підрахування рейтингу.

**3.3 Алгоритми та структури даних**

Клас Dentistry:

У класі Dentistry є такі поля:

* QString nameOfDentistry;
* QString address;
* QString workingHours;
* QString aboutUs;
* QString contactsDentistry;
* int idOfDentistry; //ми використовуємо це поле для находження клініки лікаря.

У класі Dentistry є такі методи :

* Set-ери для кожного поля;
* Get-ери для кожного поля.

Клас Dentist:

У класі Dentist є такі поля:

* Dentistry dentistry; // ми використовуємо структуру Dentistry для зберігання даних про стоматологію.
* QString firstName;
* QString lastName;
* QString patronymic;
* QString specialization;
* QString aboutDentist;
* QString contactsDentist;
* int years;
* double rating;
* int workExperience;
* int idOfDentistry; // ми використовуємо це поле для находження клініки лікаря.
* QString photo; //ми використовуємо це поле для зберігання назви фотографії,

наприклад: “dentist.png”.

* QList<Comments> listComments;// ми використовуємо QList для зберігання списку коментарів типу Comments

У класі Dentist є такі методи :

* Set-ери для кожного поля;
* Get-ери для кожного поля.

Клас Comments:

У класі Comments є такі поля:

* QString commentatorName;
* QString comment;
* double rating;

У класі Comments є такі методи :

* Set-ери для кожного поля;
* Get-ери для кожного поля.

Хеш-таблиці:

При зчитуванні даних з JSON ми додаємо їх в хеш-таблицю. Хеш-таблиця існує для швидкого пошуку за ключовими словами. Існує пошук лікаря за прізвищем, спеціалізації та клініці. Так само є алгоритм знаходження відповідного лікаря за симптомами.

При виведенні даних на екран ми маємо можливість вибрати варіант сортування (за рейтингом, за алфавітом, проти алфавіту). У вікні виводу для нас відкритий вибір дій: докладніше про лікаря і залишити коментар. З вибірки коментарів складається рейтинг для кожного стоматолога.

Існують декілька хеш-таблиць:

* HashTableByName;
* HashTableByNameOfDentistry;
* HashTableBySpecialization;

Усі вони використовують відкриту адресацію. Вони в основному схожі, але їх пошук відрізняються, через які не можна використовувати одну хеш-таблицю для всіх випадків. Хешування відбувається відбувається за трьома першими літерами

for (int i = 0; i < key.size(); i++)

{

hash\_result = (hash\_result + key[i].unicode()) % size;

}

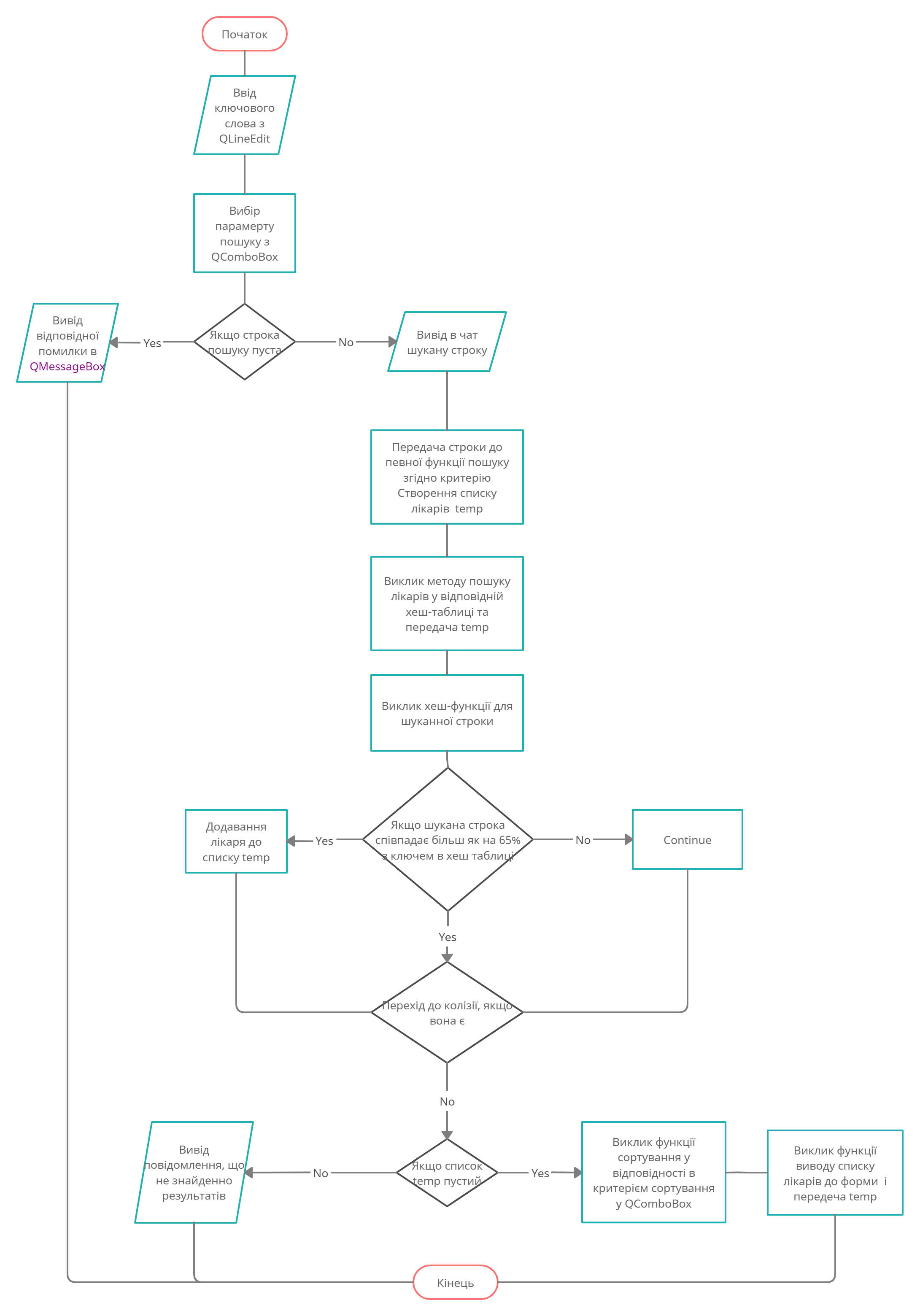


Рисунок 3.1 – схема алгоритму пошуку лікарів за певним критерієм

**Бульбашкова сортування**

У коді є три методи сортування:

* по рейтингу (sortByRating);
* по алфавіту (sortByAscending);
* проти алфавіту(sortByDescending).

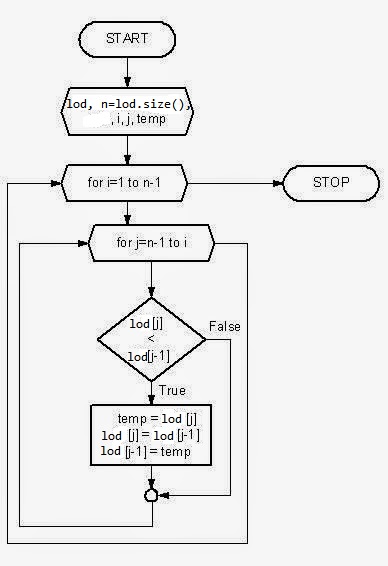


Рисунок 3.2 - схема алгоритму сортування бульбашкою

**3.4 Виключні ситуації**

**4 КЕРІВНИЦТВО ПРОГРАМІСТА**

**4.1 Призначення програми**

Призначення програми полягає в пошуку стоматолога по імені, спеціалізації, клініки, а також за ключовими знаходження лікарів.

**4.2 Перерахування файлів, в яких знаходиться програма**

Folder:

* Date;
* Icons;
* ImageDentists.

.pro:

* BotDetective.pro.

.user:

* BotDetective.pro.user.

.h:

* commentsdentist.h;
* dentist.h;
* dentistry.h;
* detaileddentist.h;
* developers.h;
* hashtablebyname.h;
* hashtablebynameofdentistry.h;
* hashtablebyspecialization.h;
* help.h;
* mainwindow.h.

.cpp:

* commentsdentist.cpp;
* dentist.cpp;
* dentistry.cpp;
* detaileddentist.cpp;
* developers.cpp;
* hashtablebyname.cpp;
* hashtablebynameofdentistry.cpp;
* hashtablebyspecialization.cpp;
* help.cpp;
* main.cpp;
* mainwindow.cpp.

.ui:

* commentsdentist.ui;
* detaileddentist.ui;
* developers.ui;
* help.ui;
* mainwindow.ui.

.qrc:

* resources.qrc.
  1. **Вхідні і вихідні дані**

1. Date:
   * + dentist\_eng.json;
     + dentist\_ukr.json;
     + dentistry\_eng.json;
     + dentistry\_ukr.json;
     + interface\_eng.json;
     + interface\_ukr.json;
     + spec\_eng.json;
     + spec\_ukr.json.
2. ImageDentists:
   * + фотографії стоматологів.
3. Icons:
   * + іконки з програми.

**5 КЕРІВНИЦТВО КОРИСТУВАЧА**

**5.1 Призначення програми**

Програма призначена для пошуку стоматологів за певним критерієм або ключовим словом. Доступні критерії:

* пошук за прізвищем;
* пошук за спеціалізацією або послугою, яка відноситься до цієї спеціалізації;
* пошук за назвою стоматології.

Також можливо виводити знайдених стоматологів за критерієм сортування:

* сортування за рейтингом;
* сортування за прізвищем (за зростанням);
* сортування за прізвищем (за спаданням).

Можливо додавати коментарі до певного лікаря, таким чином вибудовується рейтинг у стоматологів.

**5.2 Умови виконання програми**

Програма виконується, якщо вдалося відкрити всі обов’язкові файли з даними стоматологів, стоматологічних послуг та стоматологій. Якщо файли інтерфейсу пошкодженні то програма може працювати некоректно.

**5.3 Виконання програми**

Виконання програми відбувається з відкриття обов’язкових файлів, якщо не вдасться відкрити певний файл, то з’явиться повідомлення який саме файл не вдалося відкрити. Під час завершення програми дані про стоматологів зберігаються у відповідний файл, якщо програма некоректно завершиться, то всі зміни які відбувалися в програмі не зберігуться.

**5.4 Повідомлення користувачу**

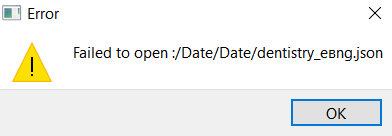
****

Рисунок 5.1 - Повідомлення, якщо не вдалося відкрити певний файл

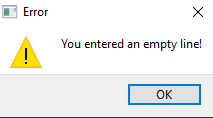


Рисунок 5.2 - Повідомлення, якщо введений пустий рядок

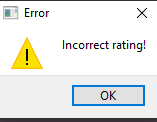


Рисунок 5.3 - Повідомлення, якщо при додаванні коментаря введен некоректний рейтинг

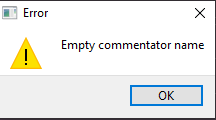


Рисунок 5.4 - Повідомлення, якщо не введено ім'я коментатора



Рисунок 5.5 - Повідомлення, якщо не знайдено стоматологів за заданим критерієм

**ВИСНОВКИ**

У цій програмі були досить вдало застосовні хеш-таблиці для зберігання даних. Були добре зроблені структури такі як Dentist, Dentistry. Так само не можна не помітити відмінне використання JSON.

Із недоліків не зовсім коректна сортування на українському (сортування відбувалася при перекладі в unicode і проблема полягала в тому, що “і” менший unicode ніж у “а”).

Другим недоліком можна вважати отстутсвіі збереження прогресу использвания програми при переході на іншу мову (всі знайдені лікарі зникали).

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. QT [Електронний ресурс]: Вікіпедія. Вільна енциклопедія. – Режим доступа: <https://uk.wikipedia.org/wiki/Qt>
2. ООП [Електронний ресурс]: Вікіпедія. Вільна енциклопедія. – Режим доступа: https://uk.wikipedia.org/wikiОб%27єктно-орієнтоване\_програмування
3. QList [Електронний ресурс]: doc.qt.io. Документація. – Режим доступа: <https://doc.qt.io/qt-5/qlist.html>
4. Базові контейнери QT [Електронний ресурс]: doc.crossplatform.ru. Вільна енциклопедія. – Режим доступа: <http://doc.crossplatform.ru/qt/4.3.2/containers.html>
5. Класи QString і QVariant [Електронний ресурс]: opennet.ru. Вільна енциклопедія. – Режим доступа: <https://www.opennet.ru/docs/RUS/qt3_prog/x5518.html>
6. Віджети та компоновка QT [Електронний ресурс]: doc.crossplatform.ru. Вільна енциклопедія. – Режим доступа: <http://doc.crossplatform.ru/qt/4.7.x/widgets-and-layouts.html>
7. Хеш-таблиці [Електронний ресурс]: habr.com. Пост. – Режим доступа: <https://habr.com/ru/post/509220/>

**ДОДАТОК А**

**Текст програми**