

BAZA DANYCH Z INTERFEJSEM GRAFICZNYM

Autor: Damian Machnik
Akademia Górniczo-Hutnicza

Kraków (C) 2022

Spis treści

1. WSTĘP.....	3
2. FUNKCJONALNOŚĆ (<i>FUNCTIONALITY</i>)	4
3. ANALIZA PROBLEMU (<i>PROBLEM ANALYSIS</i>)	5
4. PROJEKT TECHNICZNY (<i>TECHNICAL DESIGN</i>)	6
5. OPIS REALIZACJI (<i>IMPLEMENTATION REPORT</i>)	8
6. PODRĘCZNIK UŻYTKOWNIKA (<i>USER'S MANUAL</i>).....	9
BIBLIOGRAFIA.....	12

1. Wstęp

Dokument dotyczy opracowania uniwersalnej bazy danych dla kół naukowych, zajęć dodatkowych czy innych grup zrzeszających większą ilość osób. Celem systemu jest umożliwienie kontroli i edycji listy członków spotkań wraz z potrzebnymi danymi. Prosty interfejs z niezbędnymi funkcjami pozwoli na bezproblemowe korzystanie z systemu bez wcześniejszego zapoznawania się z nim. Zastosowanie programu umożliwi rezygnację z dotychczasowej papierowej wersji zbierania informacji. Dzięki temu rozwiązaniu w łatwy i szybki sposób możemy uzyskiwać oraz modyfikować listę członków koła czy zajęć przy pomocy urządzenia elektronicznego niezależnie od systemu operacyjnego.

2. Funkcjonalność

Do funkcjonalności systemu należą:

- Zapis danych osobowych nowego członka koła do listy.
- Usunięcie danych osobowych z listy.
- Wypisanie listy wszystkich członków.
- Wyszukanie danych osobowych na podstawie imienia bądź nazwiska.
- Awansowanie członka zmieniając jego stopień umiejętności.

Dane użytkowników będą zapisywane do pliku „MemberList.txt” za pomocą odpowiedniego przycisku.

3. Analiza problemu

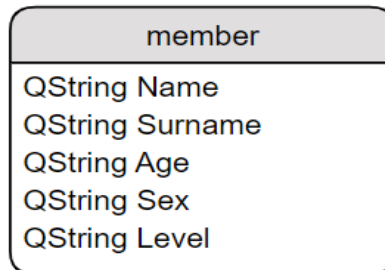
Do stworzenia interfejsu graficznego niezbędne było skorzystanie z przeznaczonego do tego środowiska. Moim wyborem było QT. By obsługiwać elementy graficzne należało zapoznać się z funkcjami oraz typami danych dedykowanymi dla tego środowiska. Wiele z nich znacząco różniło się od tych wcześniej poznanych, dlatego odnalezienie się wymagało poświęcenia czasu na badania. Przykładowo obsługa danych tekstowych wymagała użycia zmiennych „QString” natomiast obsługa plików tekstowych biblioteki „QFile”.

Dedykowanym kompilatorem dla środowiska QT jest qmake. Ze względu na jego ograniczenia pod względem wieloplatformowości, kompatybilności oraz wymagań projektowych został użyty CMake. Wymagało to wprowadzenia zmian w programach oraz projekcie. Po wielu próbach udało się z powodzeniem zbudować projekt w CMake.

Tworzony interfejs graficzny ma być prosty i przystępny dla każdego użytkownika. Z tego powodu zdecydowano się na użycie jedynie przycisków, rozwijanych list i pól tekstowych. Każdy z opisanych wcześniej elementów powinien być krótko opisany, by możliwie jak najbardziej uprościć korzystanie z programu. By zniwelować ewentualne błędy użytkownika program każdorazowo po wpisaniu danych i naciśnięciu na przycisk powinien w nowym oknie uzyskać potwierdzenie ze strony użytkownika o prawidłowo wprowadzonych danych. Zaimplementowane powinny zostać również okna informujące o błędach np. podczas wpisywania danych. W kodzie wprowadzone powinny zostać funkcje odpowiedzialne za analizę wpisanych tekstów np. usuwające spację przed lub po nazwach.

4. Projekt techniczny

W programie została użyta klasa „member”. Została ona przedstawiona na grafice 1.



Graf.1 - member class

Klasa umożliwia w prosty i szybki sposób zbierać, modyfikować dane konkretnego członka w jednym miejscu. Daje to możliwość prostego poruszania się po kodzie oraz zwiększa jego przejrzystość. Ułatwia również odczyt danych z wcześniej zapisanego członka przy pomocy prostej składni.

Dane odczytywane przez program z interfejsu graficznego są typu QString, dlatego też wszystkie zmienne w klasie są tego typu, umożliwiając ich łatwą obsługę.

Klasa ta posłużyła przy budowie bazy danych. Wektor który pełni tę funkcję jest typu „member”.

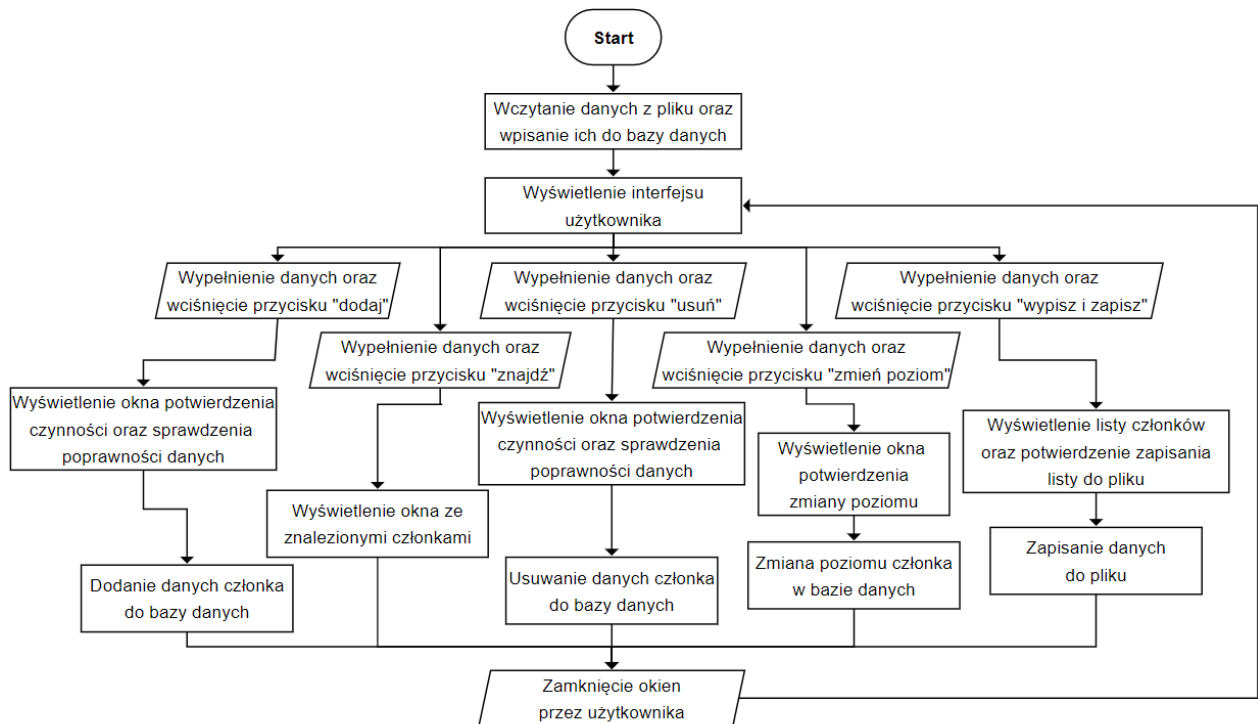
W programie w każdym polu tekstowym interfejsu graficznego użyte zostało zabezpieczenie przed użyciem spacji polegające na usunięciu ich całkowicie z tekstu wpisanego przez użytkownika.

Pozwala to na prawidłowe kodowanie danych do pliku oraz usuwa błędy podczas wyszukiwania konkretnych członków.

By zmniejszyć ryzyko popełnienia błędów program każdorazowo dopytuje użytkownika w nowym oknie czy dane które zostały wpisane są poprawne jednocześnie prezentując mu je, by mógł spojrzeć na nie ponownie.

Podczas wpisania złych danych program wyświetla w nowym oknie informacje o błędzie użytkownika by ten otrzymał informacje zwrotną programu. Pozwoli to uniknąć uczucia zawieszonego programu, który nie wykonuje przedstawionych mu poleceń.

Schemat blokowy programu wygląda następująco:



Graf. 2 – Schemat blokowy programu

5. Opis realizacji

Projekt pisany oraz testowany był w systemie Windows 10 64 bitowym. Narzędziem z którego korzystano to „QT Creator 9.0.1 Community”. Wersja QT to 6.4.1 natomiast CMake 3.25.1.

Projekt został rozpoczęty 26 grudnia 2022 w Visual Studio 2022. Projekt rozwinął się znacząco, niestety na skutek niepowodzeń z interfejsem graficznym prace zostały przerwane dnia 30 grudnia 2022. Prace nad projektem zostały ponowione 11 stycznia 2023. Zostało zmienione środowisko na QT Creator a sam program został napisany na nowo. Prace nad projektem odbywały się regularnie do 16 stycznia 2022, gdy projekt zakończono. Tego samego dnia wykonane zostały testy które przebiegły pomyślnie. Projekt został przesłany na GitHub do repozytorium „JPO_Damian_Machnik_Baza_Danych”.

https://github.com/MachnikDamian/JPO_Damian_Machnik_Baza_Danych

6. Podręcznik użytkownika

Po uruchomieniu programu, przed użytkownikiem pojawia się okno przedstawione na Obr.1.

Obr.1 - Ekran główny programu

Poszczególnymi cyframi oznaczone zostały funkcje:

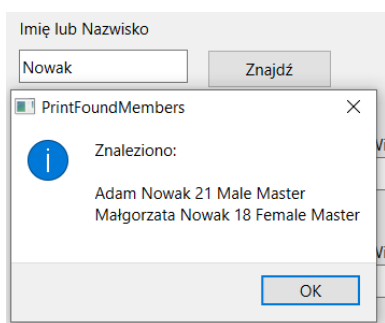
1. Dodanie danych do bazy danych
2. Wyszukanie członka z bazy danych
3. Usunięcie członka z bazy danych
4. Zmiana poziomu członka
5. Wyświetlenie oraz zapisanie listy członków do pliku

Po wypełnieniu pierwszej linii, która odpowiada za funkcję dodawania (oznaczona literą 1 na Obr.1) oraz kliknięciu przycisku pojawia się okno pierwsze przedstawione na Obr.2. Jest to okno, które prosi o potwierdzenie czy dane wpisane są poprawnie. W jego opisie znajduje napis „czy chcesz dodać członka?”, a poniżej widnieją dane wpisane przez użytkownika. Następnie po wciśnięciu opcji „Yes” pojawia się drugie okno przedstawione na Obr.3. Informuje ono, że dane zostaną dodane do bazy danych. Gdy w oknie pierwszym zostanie kliknięta zostanie opcja „No” dane nie zostaną dodane, a okno zniknie.

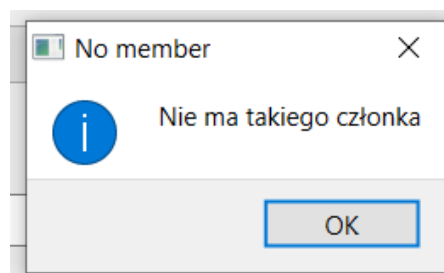
Obr.2 - Okno dodawania 1

Obr.3 – Okno dodawania 2

W funkcji wyszukiwania jest możliwość wpisania imienia bądź nazwiska członka. Program przeszukuje bazę danych, a następnie wyszukuje i wypisuje wszystkich członków, do których dane będą pasować. Lista zostanie wyświetlona w osobnym oknie przedstawionym na Obr.4. Gdy program nie znajdzie członka o podanym imieniu lub nazwisku wyświetli okno przedstawione na Obr.5.

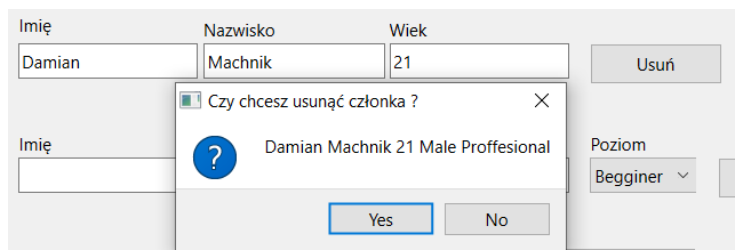


Obr.4 - Okno szukania 1

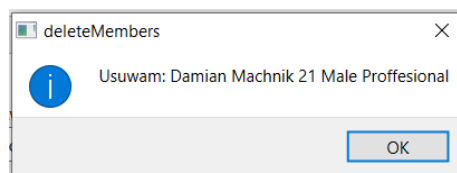


Obr.5 – Okno szukania 2

By usunąć członka należy poprawnie wpisać jego dane w linii 3 w oknie głównym (Obr.1), a następnie przycisnąć przycisk „usuń”. Po poprawnym wypełnieniu wszystkich danych pojawi się okno pytające, czy na pewno dany członek powinien zostać usunięty, przedstawione na Obr.6. Po wciśnięciu „Yes” członek zostanie usunięty z bazy danych oraz pojawi się okno przedstawione na Obr.7. Gdy wybrana zostanie opcja „No” program przejdzie do okna głównego. W momencie gdy dane zostaną wprowadzone niepoprawnie program wyświetli okno przedstawione na Obr. 5.

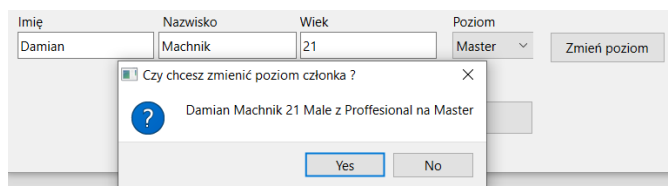


Obr.6 - Okno usuwania 1

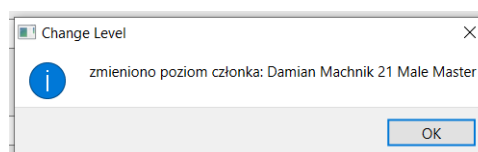


Obr.7 - Okno usuwania 2

W linii 4 należy wypełnić imię, nazwisko, wiek oraz nowy poziom członka. Po poprawnym wypełnieniu tych rubryk program wyświetli okno przedstawione na Obr.8. Okno ma za zadanie uzyskać potwierdzenie od użytkownika, że dane powinny zostać zmienione. Po wciśnięciu “Yes” pojawi się okno przedstawione na Obr.9, a poziom członka zostanie zmieniony w bazie danych. Gdy klikniemy “No” program przeniesie nas do okna głównego. Jeśli wpisujemy nieprawidłwe dane pojawi się okno przedstawione na Obr.5.

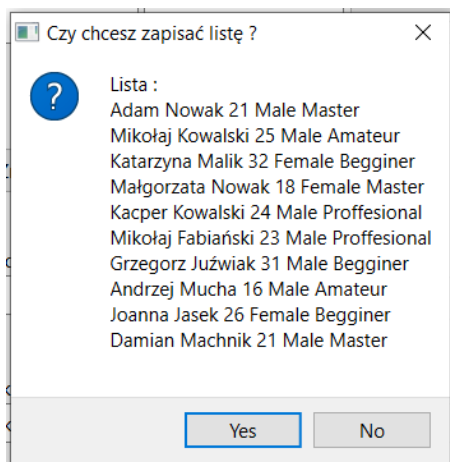


Obr.8 – Okno zmiany poziomu 1

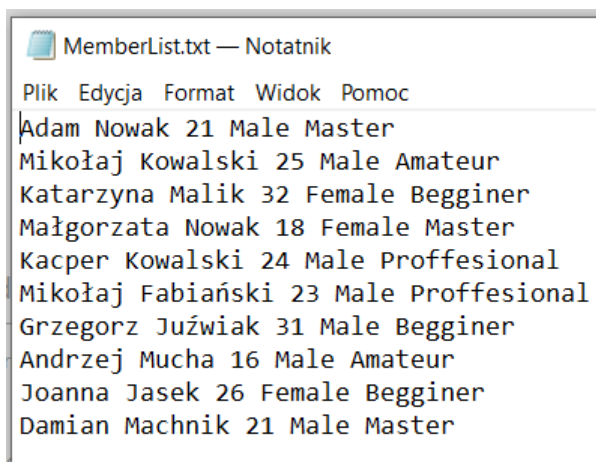


Obr.9 – Okno zmiany poziomu 2

Ostatnią funkcjonalnością jest wypisanie i zapisanie danych do pliku. Po kliknięciu na przycisk oznaczony na Obr.1 cyfrą 5 pojawi się okno przedstawione na Obr.10. Wszystkie dane które zostały kiedykolwiek dodane zostają wyświetlone w oknie. Okno poza wypisaniem listy członków ma za zadanie uzyskać odpowiedź od użytkownika czy dane powinny zostać zapisane. Wybierając opcję „Tak” lista ta zostaje zapisana do pliku. Dane w pliku prezentują się w sposób widoczny na Obr.11. Wybierając opcję „Nie” program przenosi nas do okna głównego przedstawionego na Obr.1



Obr. 10 – Okno wypisania



Obr. 11 – Dane z pliku tekstowego

7. Bibliografia

Podczas tworzenia projektu korzystano z:

<https://stackoverflow.com/questions/50436877/create-cmake-qt-project-in-visual-studio-2017>

<https://decovar.dev/blog/2019/05/11/vscode-cmake-qt/>

<https://doc.qt.io/qt-5/cmake-manual.html>

<https://doc.qt.io/qt-6/qstring.html>

<https://doc.qt.io/qt-6/qfile.html>

<https://doc.qt.io/qt-6/qmessagebox.html>

<https://doc.qt.io/qtcreator/creator-writing-program.html>

https://www.bogotobogo.com/Qt/Qt5_QStringList.php

<https://cpp.hotexamples.com/examples/-/QFile/remove/cpp-qfile-remove-method-examples.html>

serii filmów poradnikowych:

<https://www.youtube.com/watch?v=EkjaiDsiM-Q>

[Q&list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA&ab_channel=ProgrammingKnowledge](https://www.youtube.com/watch?v=EkjaiDsiM-Q&list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA&ab_channel=ProgrammingKnowledge)