

**Dokumentacja projektowa**

**“Super kalkulator 4k” (Sk4k)**

***Wykonanie: Maciej Pelc oraz Dawid Sopel***

Spis treści:

1. Opis założeń projektu:  
    Wymagania biznesowe  
    Identyfikacja wymagań
2. Folder dokumentacji
3. Opis techniczny projektu
4. Warstwa użytkowa projektu:  
    Aplikacja mobilna  
    Aplikacja desktop  
    Aplikacja web
5. System kontroli wersji
6. Raporty z testów jednostkowych
7. Literatura
8. Komentarz

Opis założeń projektu:

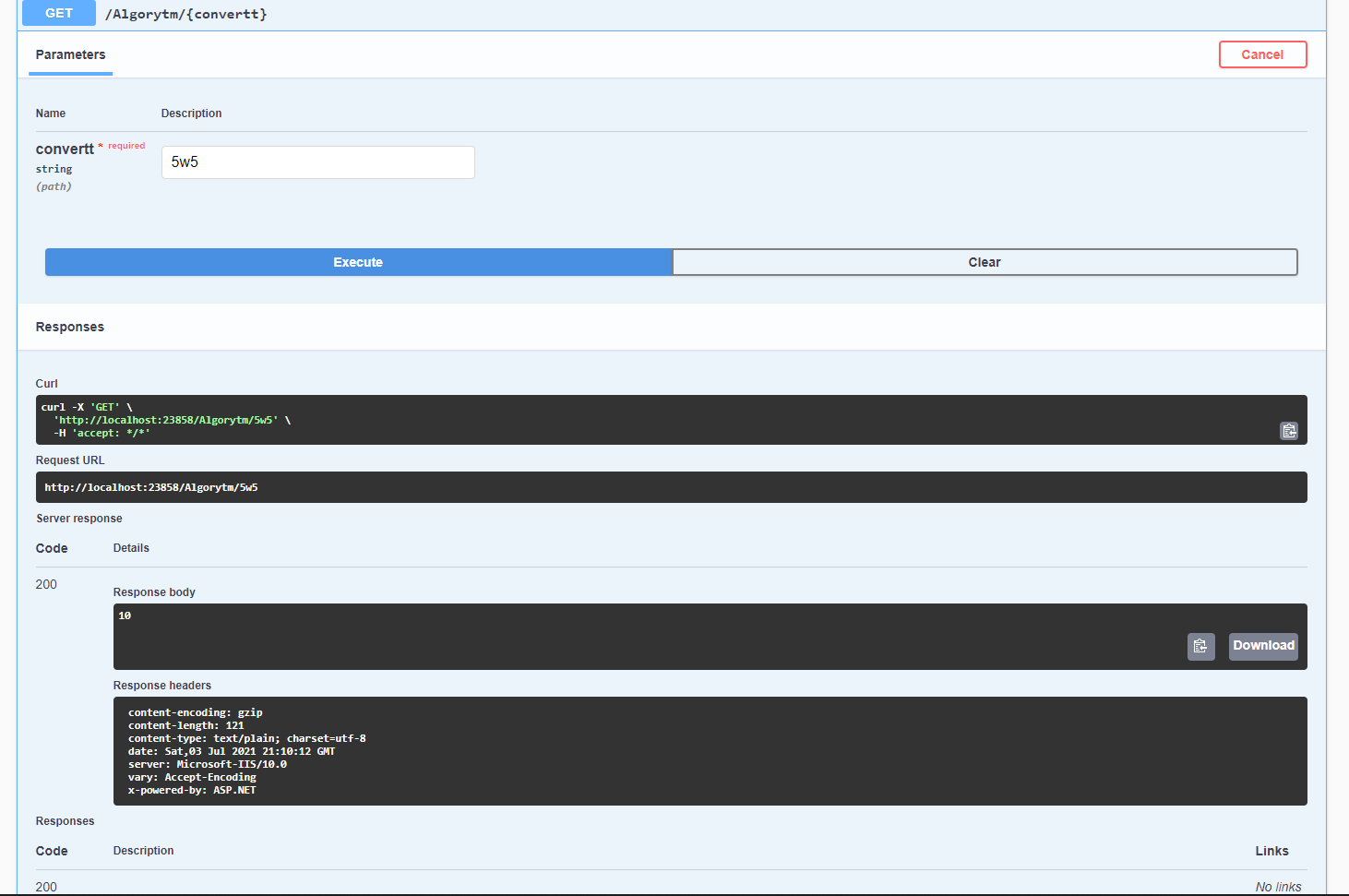
* Wymagania biznesowe:
  + Możliwość dodania do każdej platformy reklam.
  + Możliwość płatnej subskrypcji i dostępu do historii zapisów operacji matematycznych.
  + Przedsiębiorstwo korzystające z systemu Sk4k będzie miało dostęp do najlepszego środowiska kalkulatorowego dostępnego na rynku, co znacząco może się przyczynić do skrócenia czasu pracy z kalkulatorem dla każdego pracownika.
* Identyfikacja wymagań:
  + Funkcjonalne:
    - Dostęp do systemu z urządzeń mobilnych, komputerów stacjonarnych oraz przeglądarki internetowej,
    - Dostęp do historii operacji matematycznych i wyników z każdego urządzenia,
    - Obsługa nawet skomplikowanych operacji matematycznych zgodnie z kolejnością wykonywania zadań, z uwzględnieniem nawiasów.
  + Nie funkcjonalne:
    - System powinien mieć łatwą skalowalność systemu,
    - System powinien mieć możliwość skonstruowania zapytania matematycznego do przynajmniej dwóch tysięcy znaków,
    - System powinien być łatwy w obsłudze,
    - System powinien być zoptymalizowany pod kątem użycia na dowolnych urządzeniach, które tylko będą miały połączenie z Internetem.

W folderze z dokumentacją znajduje się:

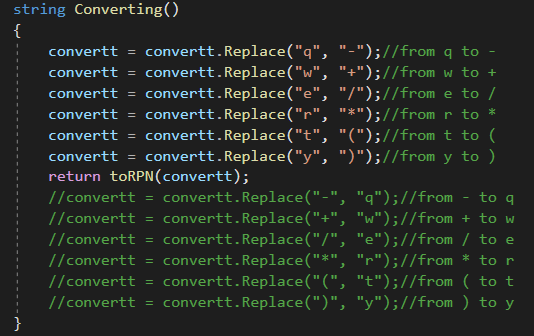
* Diagram przypadków użycia (Use case diagram),
* Dokumentacja DOXYGEN (paczka html.zip - ./index.html),
* Harmonogram realizacji projektu (diagram Gantta, w pliku .txt link),
* **Pseudokod obrazujący działanie algorytmu**.

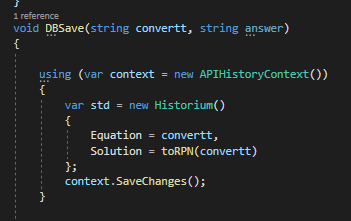
Opis techniczny projektu:

* Projekt został zrealizowany w technologiach .NET C# oraz Flutter. Działanie systemu Sk4k polega tworzeniu zapytania przez użytkownika z poziomu aplikacji mobilnej, webowej czy desktopowej. Zapytanie te jest kodowane, w sposób by serwer mógł przyjąć zapytanie po czym do niego wysyłane. Następnie jest ono dekodowane i przenoszone do funkcji, która to kolejno korzysta z algorytmu kolejkowego (Shunting-yard algorithm) Edsger’a Dijkstry. Następuje konwersja na z zapisu infiksowego ((2+5)/7+3/5) na zapis postfiksowy (2 5 + 7 / 3 5 +), inaczej odwróconej notacji polskiej. Z zapisu postfiksowego algorytm dokona obliczeń i zwrotną odpowiedź wysyła w postaci string. Do testowania metod Http Get przewidziano IDL Swagger. Urządzenie, z którego zapytanie zostało wysłane pobiera wynik, który jest wyświetlany. W przypadku błędu aplikacje go obsługują i powiadamiają użytkownika. Po dokonaniu obliczeń zapytanie oraz rozwiązanie zapisywane jest w bazie danych SQL, do której wgląd ma administrator. Serwer API, wykonany w technologii .NET ASP API(.NET 5.0), jest hostowany dla klientów za pomocą Windows Server 2019 Essentials w usłudze IIS. Zarządzanie odbywa się za pomocą łączenia z nim lokalnie poprzez tunel szyfrowany tunel (SHA1-AES256) OpenVPN, a z tego miejsca jest dopiero dostępny protokół RDP. Serwer z zewnątrz jest jedynie dostępny dla usług WEB oraz REST na dwóch osobnych portach, reszta połączeń jest objęta flagą drop, co niweluje możliwość nieautoryzowanego dostępu. Połączenia, firewall, trasowanie oraz serwer OpenVPN obsługiwane są za pomocą MikroTik RouterOS, zaś administrator może połączyć się ze środowiskiem do zarządzania dzięki odpowiednim certyfikatom.

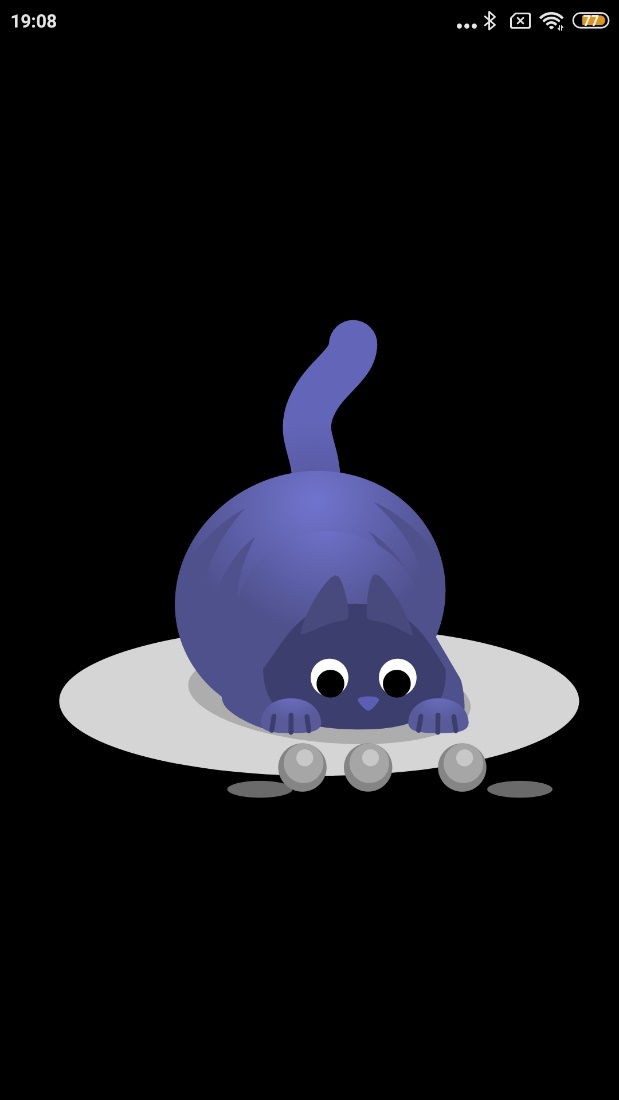


Encoding q, w, e, r, t, y = -, +, /, \*, (, ).  
(5+5 = 5w5)

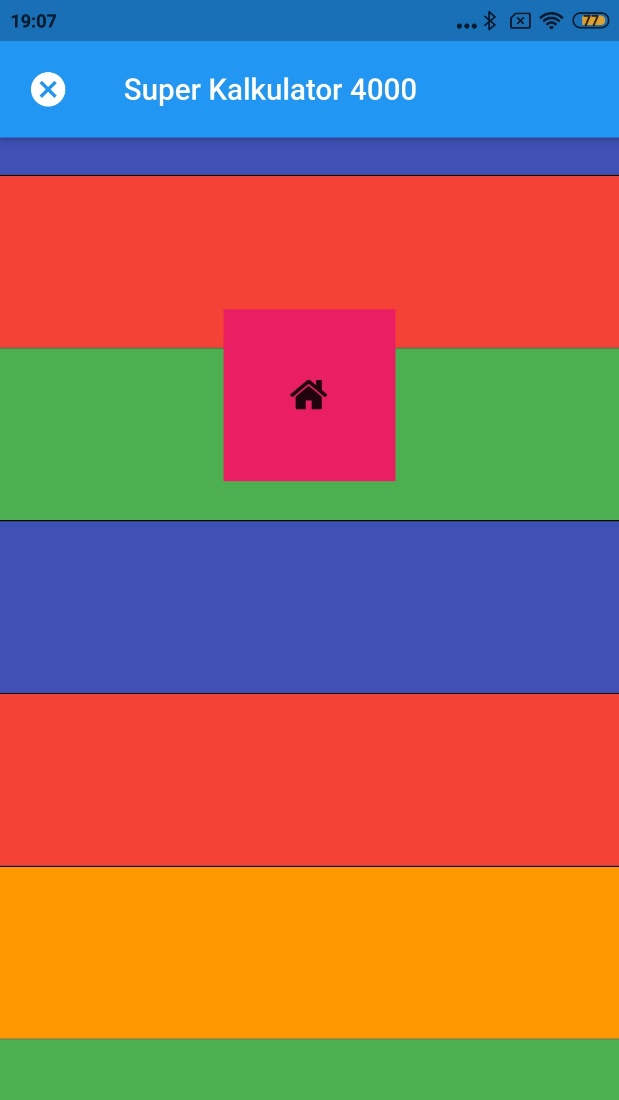




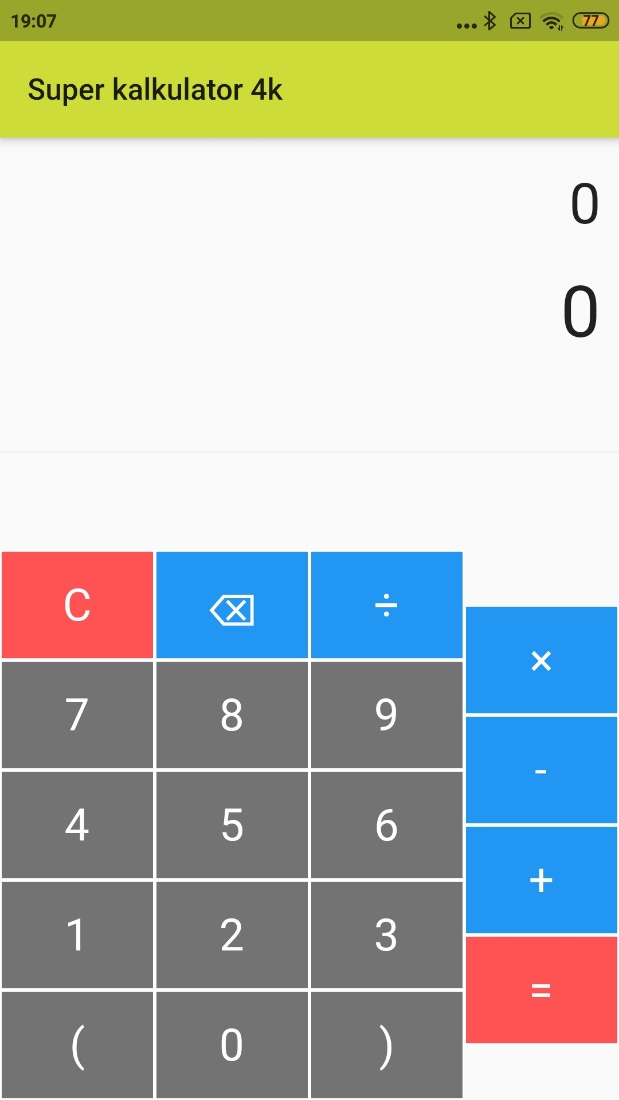
Prezentacja warstwy użytkowej projektu:

Aplikacja mobilna:  


Na zdjęciu widać animację lottie, przy starcie programu podczas ładowania.

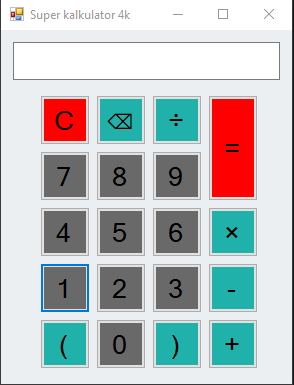


Ekran powitalny aplikacji mobilnej (kolory się poruszają). Ekran jest niesamowicie skalowany, by przejść do kalkulatora, należy kliknąć w ikonę.



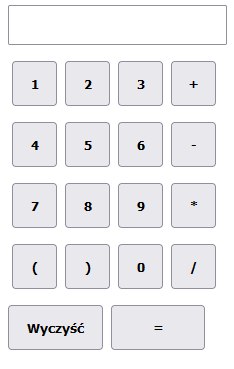
Frontend aplikacji mobilnej, C usuwa cały kod, znak backspace jedną literkę z zapisu. Pozostałe są tym co pokazują ich obrazki.

Aplikacja desktop:



Jej działanie jest identyczne jak aplikacji mobilnej.

Aplikacja web:

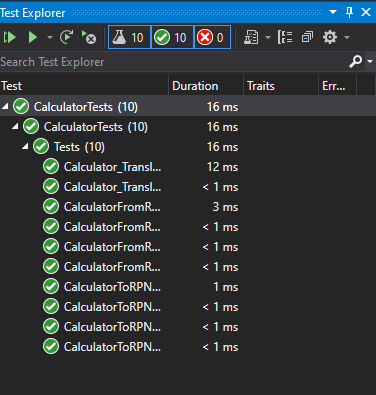


Działanie jak powyżej.

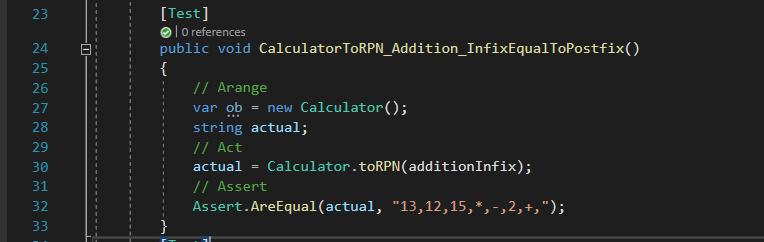
System kontroli wersji:

* <https://github.com/tejf/DesktopCalc>
* <https://github.com/tejf/MobileCalc>
* <https://github.com/tejf/WebCalc>

Raporty z testów jednostkowych:



Przykładowy Unit test:



Test stworzony dla przypadku skrajnego, gdzie uwzględniałem w teście ostatni operator dodawania.

Nazwa testu zrobiona zgodnie z dokumentacją NazwaKlasyLubFunkcji\_CoTestPrzedstawia\_WarunekPrawdy

Testy są stworzone w formacie arrange, act, assert. Wszystkie powyższe elementy zwiększają czytelność kodu.

Literatura potrzebna do stworzenia projektu:

<http://www.math.bas.bg/bantchev/place/rpn/rpn.c%23.html>

https://www.szkolazpasja.pl/onp-np-notacja-infiksowa/

<https://api.flutter.dev/flutter/dart-core/Uri/encodeFull.html>

<https://flutter.dev/docs/cookbook/networking/fetch-data>

https://flutter.dev/docs/cookbook/networking/fetch-data

https://www.youtube.com/watch?v=ub3P8c87cwk

https://www.youtube.com/watch?v=MP4zatl3jF8 https://www.youtube.com/watch?v=w1zEiTR-7pA

<https://www.youtube.com/watch?v=vN9NRqv7xmY>

https://www.youtube.com/channel/UCvtT19MZW8dq5Wwfu6B0oxw

Dokumentacja stworzona przez system komentarzy dokumentujących (XML Comments). Dokumentacja jest w postaci kodu HTML, wygenerowana przy pomocy programu Doxygen. Stworzona na podstawie kodu z “Testowego Kalkulatora”. Znajduje się w folderze dokumentacji.