

Rozdział 1

Analiza projektowa

Rozdział poświęcony analizie wybranych technologii oraz wymagań projektu. Przedstawiono również wstępną analizę dotyczącą przypadków użycia oraz metod testowania produktu.

1.1 Wybrane technologie

Krótki opis teoretyczny wybranych technologii - C++ oraz bibliotek QT.

1.1.1 QT

QT jest zespołem przenośnych bibliotek oraz narzędzi programistycznych stworzonych w C++ do tworzenia aplikacji desktopowych, zagnieżdżonych, jak również mobilnych. Wspiera systemy takie jak: Linux, OS X, Windows, xWorks, QNX, Android, iOS, BlackBerry, Sailfish OS, dzięki czemu jest nadzwyczaj uniwersalnym narzędziem kompatybilnym z następującymi językami programowania: C++, QML (QT Modeling Language), Python, Ring, Go, Rust, PHP i Java. ?? QT zapewnia listę dodatkowych możliwości rozszerzających C++. Są to między innymi:

- mechanizmy pozwalające na komunikację między obiektami zwane sygnałami i otworami (slots)
- wyjątkowa możliwość edycji wyglądu i responsywności obiektów
- swobodna możliwość edycji zachowań w przypadku różnego rodzaju zdarzeń
- kontekstowe tłumaczenie stringów do internacjonalizacji
- hierarchiczne i responsywne drzewa obiektowe, organizujące strukturę obiektów
- automatyczna zmiana wartości wskaźników na 0 w przypadku zniszczenia obiektu, w przeciwieństwie do wskaźników w C++, które stając się zawieszonymi (dangling pointers) ?

1.1.2 C++

C++ jest powszechnie stosowanym językiem programistycznym, będącym potomkiem języka C, w którym wprowadzono szereg udogodnień. W porównaniu z C, C++ zapewnia dokładniejsze sprawdzanie typów danych, wspiera abstrakcje, programowanie obiektowe (z tego względu mówi się o nim jako o języku pseudo-obiektowym), programowanie uogólnione i więcej stylów programistycznych. Do wspieranych założeń programowania obiektowego należą polimorfizm, enkapsulacja i dziedziczenie. Nowsza wersja C posiada również bardzo dużą ilość bibliotek, których wykorzystanie znacznie ułatwia jego wykorzystanie.

1.2 Architektura systemu

System składa się z warstwy aplikacji oraz warstwy transportowej. Komunikacja zrealizowana jest za pomocą broadcastu przy pomocy protokołu UDP (user data protocol).

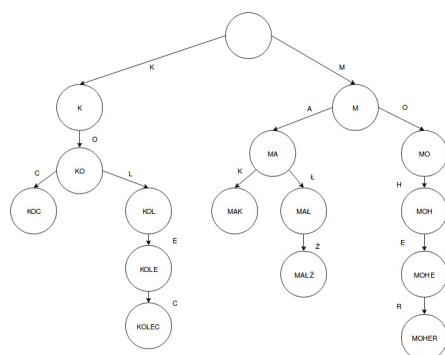
1.2.1 Architektura aplikacji

Aplikację zrealizowano w wyżej wymienionym języku C++ z wykorzystaniem wielu bibliotek QT np. QWidget, QString, QtGui, QPainter itd.. Do funkcji auto uzupełniania tekstu posłużono się algorytmem wykorzystującym skompresowane drzewo trie.

1.2.1.1 Sompresowane Drzewo Trie

Drzewo Trie jest drzewem służącym docelowo do sortowania wartości tekstowego typu danych, w którym każdemu węzłowi przypisany jest wspólny prefix(fragment klucza). Wartości tekstowe zapisywane są w liściach drzewa. (?) Do drzewa wczytywane zostają wszystkie słowa z danego mu słownika, a następnie każde rozkłada się na litery i rozmieszcza w drzewie, tak by czytając znaki od korzenia do liścia tworzyły pożądaną ciąg. Biorąc na przykład słownik składający się z następujących słów: mak, róża, kolec, małż, koc, moher otrzymujemy następujące drzewo 1.1. W ten sposób poszukując słów zaczynających się na "ko" w bardzo prosty sposób możemy określić, że zaliczają się do nich koc oraz kolec. Drzewo to w znaczący sposób skraca czas przeszukiwania dużych słowników (w wypadku tego projektu 3639970 wyrazów) i umożliwia np. auto uzupełnianie albo korektę słów bieżąco wpisywanych przez użytkownika.

Zastosowany algorytm wykorzystuje fakt występowania wspólnych węzłów i zapamiętuje jedynie numer ostatniego węzła związanego z wyszukiwanym słowem np. dla frazy "ko" - węzeł nr. 2, a następnie pobiera wartości wszystkich dzieci tego węzła, zspala je i tworzy wszystkie możliwe końcówki (tu "c" oraz "lec"), które w połączeniu z szukaną frazą dają auto uzupełnianie słowa.



Rys. 1.1 – Przykładowe drzewo typu Trie.

1.2.2 Komunikacja aplikacji

System umożliwia komunikację poprzez broadcast z innymi użytkownikami jak i poprzez specjalne API Google z przeglądarką internetową.

Broadcast jest metodą komunikacji (rozsyłania danych) między jednym nadawcą, a wieloma odbiorcami w tym samym czasie. Jest to dyfuzyjny, jednokierunkowy tryb transmisji charakterystyczny dla sieci LAN. Protokołem regulującym dany ruch sieciowy jest w tym wypadku UDP (User Datagram Protocol). Jest to bezpołączeniowy protokół, który pozwala aplikacjom dobudować własne - niezbędne do poprawnego działania, protokoły. Zapewnia on aplikacji możliwość wysyłania enkapsulowanych datagramów IP bez potrzeby wytworzenia połączenia, co jest jednocześnie dużo szybszą metodą komunikacji, niż tryb synchroniczny. Nie dba on o to, czy wysłane ramki dotrą do odbiorcy w całości, co jest typowe dla komunikacji asynchronicznej. UDP transmituje segmenty złożone z 8-bajowego nagłówka oraz fragment przesyłanych danych, będący formą wiadomości. UDP zapewnia informację o portach źródłowych i docelowych. Port źródłowy jest przydatny w momencie, gdy urządzenie odbiorcy zechce odesłać odpowiedź na otrzymany segment. Rozmiar ramki wynosi od 8-65 515 bajtów. ?

Google Custom Search umożliwia stworzenie własnej wyszukiwarki umożliwiającej na przeszukiwanie zarówno stron internetowych jak i obrazów. Możliwe jest zawężenie i personalizacja wyników wyszukiwania np. do wyników pochodzących z konkretnej strony, lub zawierających sprecyzowaną frazę. Google Search Engine występuje w dwóch wersjach - Custom Search Engine, która jest darmowa oraz Google Site Search, które jest wersją płatną. Do potrzeb projektowych wystarczająca jest wersja darmowa, która umożliwia korzystanie z dodatkowych API umożliwiających zwrócenie wyników wyszukiwania w postaci pliku XML czy też JSON. API te upraszczają komunikację aplikacji z przeglądarką do zapytań RESTowych typu get w celu otrzymania uporządkowanej struktury danych. ?

1.3 Wymagania funkcjonalne

Po przeanalizowaniu problemu klawiatury ekranowej oraz potrzeb osób chorych na ALS stwierdzono następujące wymanagania funkcjonalne:

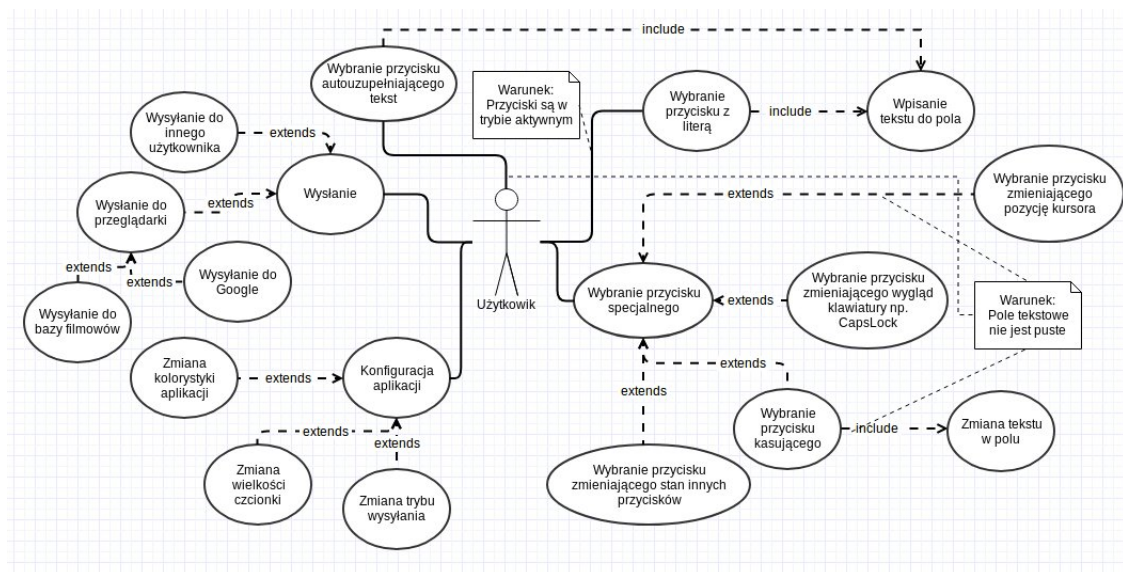
- Klawiatura ekranowa reagująca na sygnał wejściowy będący współrzędnymi punktu fiksacji, które określane są z dokładnością jednego stopnia. Zakładając, że odległość użytkownika od ekranu to 60cm, wyliczono iż dokładność wyznaczania pozycji to obszar ok. 3,2 cm powierzchni ekranowej. Dla komputera o ekranie 14 cal (monitor testowy), gdzie rozmiar pojedynczego piksela to ok. 0,2269 mm - to, w przybliżeniu, 141px dla każdego przycisku.
- Możliwość auto uzupełniania słów za pomocą słownika języka polskiego.
- Możliwość wprowadzenia przycisków w stan nieaktywny.
- Wykorzystywanie polskich znaków w tekście.
- Wysyłanie wiadomości za pomocą broadcastu.
- Wyszukiwanie stron w przeglądarce Google.
- Możliwość konfigurowania wyglądu aplikacji - różne stopnie kontrastu oraz kolorystyki, a także rozmiaru czcionki tekstu wpisywanego.

Szczegółowe przypadki użycia klawiatury zobrazowano w rozdziale "Przypadki użycia" 1.3.1.

1.3.1 Przypadki użycia

Na zamieszczonym rysunku 1.2 przedstawiono diagram przypadków użycia przedstawiający możliwe interakcje użytkownika z aplikacją.

Aby rozpocząć pracę z klawiaturą należy odblokować pisanie znaków wybierając przycisk "start". Główną funkcjonalnością, a zarazem najczęściej zachodzącym przypadkiem użycia będzie wybór przycisku z literą w celu wypisania tekstu na ekranie. Użytkownik ma do wyboru klawiaturę podstawową z 28 znakami oraz cyframi z zakresu 0-9 oraz 2 tablice przycisków ze znakami specjalnymi oraz jedną poświęconą jedynie polskim znakom. Każdy z przycisków reaguje na zmianę położenia punktu fiksacji - w momencie, w którym wykryte zostanie położenie nad przyciskiem naliczany zostaje czas, którego narastanie wizualizują się za pomocą pasku postępu. Gdy 100% postępu zostanie osiągnięte dana litera pojawia się na ekranie w przeznaczonym obszarze i w określonej, za pomocą kursora, pozycji w tekście. W przypadku jeśli użytkownik zechce zmienić wielkość liter może wybrać między opcją CapsLock (stałym powiększeniem liter wpisywanych), bądź Shift (zwiększającą kolejną literę dodaną do tekstu, następnie zmieniając wielkość liter na małe). W obu wypadkach następuje zmiana wyglądu całej klawiatury. Po tekście można poruszać się za pomocą strzałek - ruch w prawo (do początku tekstu) i lewo (na koniec tekstu) o jeden znak, lub specjalnych przycisków "home" oraz "end", które przenoszą kursor odpowiednio na początek i koniec tekstu. Przyciski Enter oraz Spacja traktowane jako zwykłe



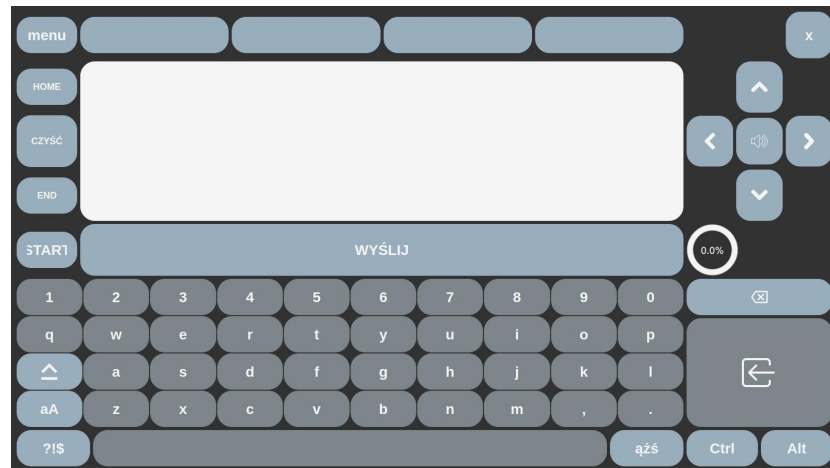
Rys. 1.2 – Wykres przypadków użycia stworzony za pomocą strony Gliffy.com

znaki. Fiksacja na przycisku Backspace usuwa ostatni wpisany znak, a na przycisku "czyść" usunięcie wszystkiego co znajduje się w polu tekstowym. Wyróżnia się też innego rodzaju przyciski specjalne, zmieniające wygląd klawiatury - są to przyciski do znaków specjalnych oraz polskich liter. Pierwszy z wymienionych w pierwszym kroku prezentuje wszystkie znaki specjalne powszechnego użytku np. wykrzyknik, cudzysłów, średnik, a przy powtórnym wciśnięciu na klawiaturze pojawiają się popularne emotikony, które znalazły tam swoje miejsce, ze względu na możliwość wysyłania wiadomości do innych użytkowników i mają za zadanie ułatwienie reprezentacji emocjonalnego przekazu wiadomości. W przypadku gdy użytkownik wpisał już jakiś fragment tekstu może posłużyć się przyciskiem odpowiedzi, które działają na zasadzie auto uzupełniania tekstu na podstawie słów znalezionych w słowniku. Aby zmienić wygląd aplikacji lub tryb wysyłania należy skorzystać z przycisku menu. To pozwoli na wybranie schematu kolorystycznego aplikacji oraz na dobór wielkości czcionki. Użytkownik może wybrać również jeden z trzech trybów rozsyłania zapisanego przez niego tekstu - są to: Google Search, Filmweb oraz Broadcast (w celu czatu). W zależności od wybranego trybu, przycisk "wyślij" powoduje wywołanie odpowiedniej akcji.

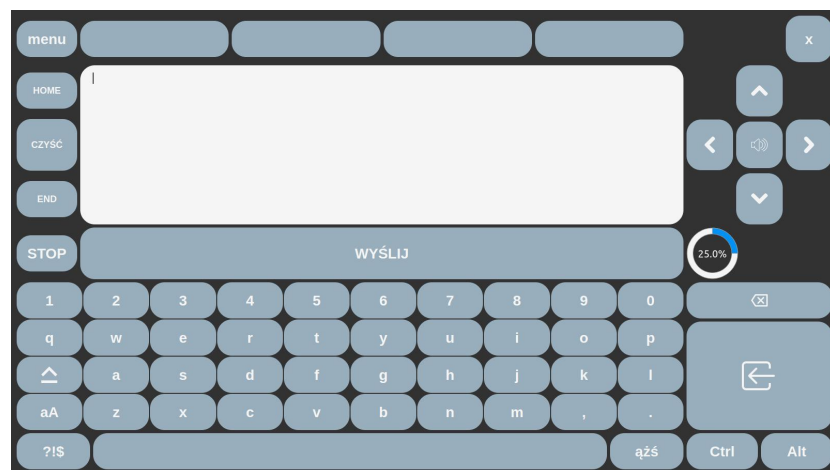
1.4 Propozycja rozwiązania

1.4.1 Zakres projektu

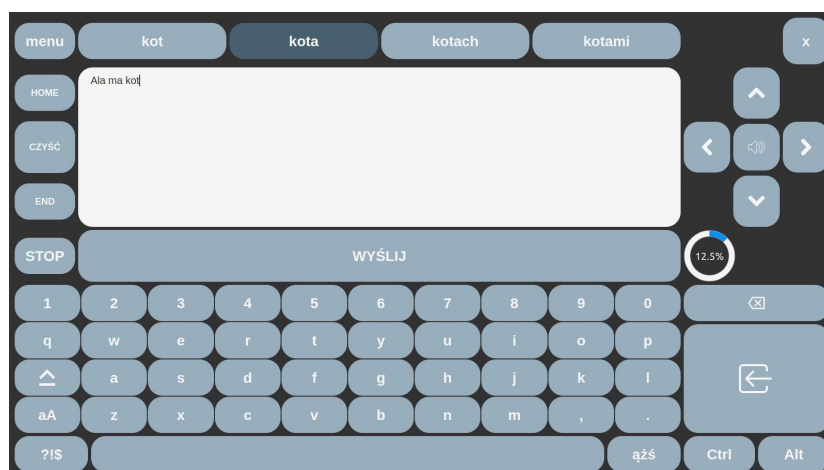
1.4.2 Interfejs oprogramowania



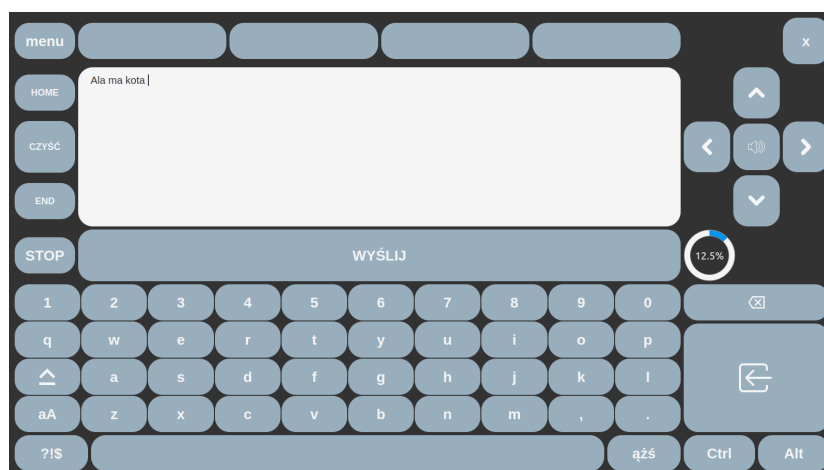
Rys. 1.3 – Początkowy wygląd aplikacji z nieaktywnymi przyciskami.



Rys. 1.4 – Aktywny wygląd aplikacji.



Rys. 1.5 – Przykład wyświetlanych opcji autouzupełniania.



Rys. 1.6 – Tekst i podpowiedzi po wybraniu sugerowanego tekstu.

1.5 Projekt testów

1.6 Cele biznesowe

1.6.1 Możliwość dalszego rozwoju

1.7 Wymagania pozafunkcjonalne

1.7.1 Ergonomia

1.7.2 Ograniczenia

1.8 Podsumowanie