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, Widows, 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 pseudoobiektowym), programowanie uogólnione i więcej styli 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).

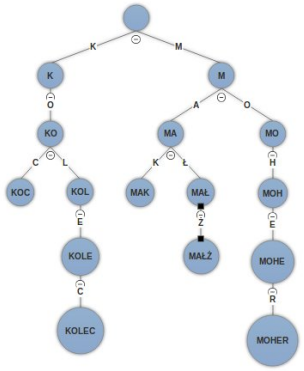
1.2.1 Architektura aplikacji

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

1.2.1.1 Sompresowane Drzewo Trie

Drzewo Trie jest drzewem służącym docelowo do sortowania wartości tesktowego 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 porządany ciąg. Biorą na przykład słownik składający się z następujących słow: mak, róża, kolec, małż, koc, moher otrzymujemy następujące drzewo 1.1. W ten sposób poszukując słow 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. autouzupełnianie albo korektę słów bierząco 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, zespala je i tworzy wszystkie możliwe końcówki (tu ćóraz łec"), które w połączeniu z szukaną frazą dają ąutouzupełnianeśłowa.



Rys. 1.1 – Przykładowe drzewo typu Trie.

1.2.2 Komunikacja aplikacji

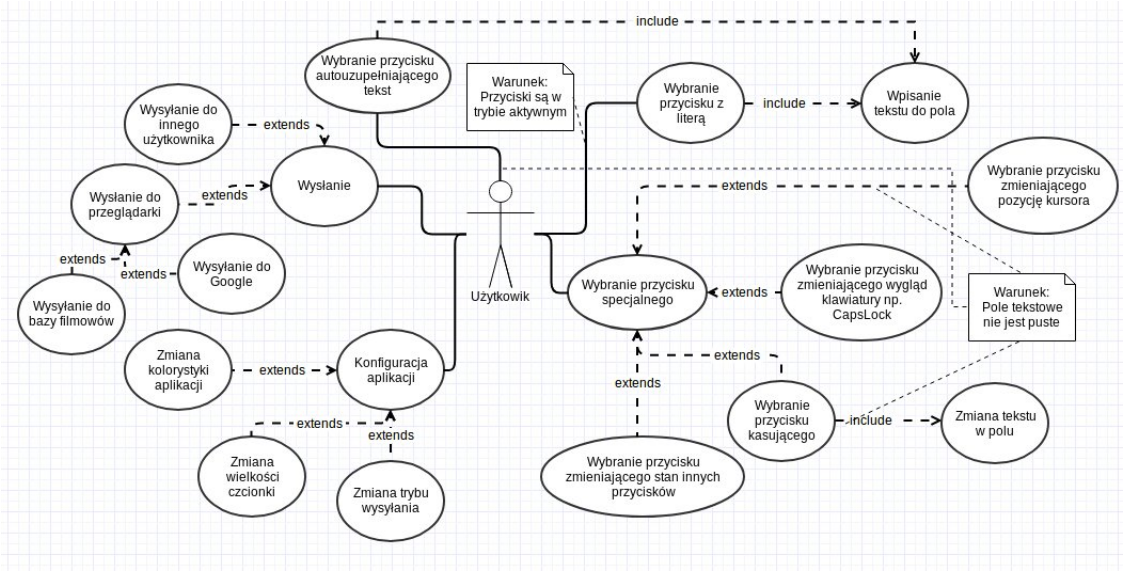
System umożliwia komunikację poprzez broadcast z innymi użytkownikami jak i poprzez specjalne API Google z przeglądarką internetową.

Broadcast jest metodą komunikacji (rozsyłania danych) między jednym nadawacą, a wieloma odbiorcami w tym samym czasie. Jest to dyfuzyjny, jednokierunkowy tryb transmijcji charakteystyczny 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. Nie dba on więc o to, czy wysłane ramki dotrą do odbiorcy w całości. UDP transmituje segmenty złożone z 8-bajtowego nagłówka oraz fragment przesyłanych danych, będący formą wiadomości. UDP zapewnia infomację 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 Seachr Engine występuje w dwóch wersjach - Custom Search Engine, która jest darmowa oraz Google Site Search, które jest werjsą płatną. Do potrzeb projektowych wystraczająca jest wersja darmowa, która umożliwia korzytsanie z dodatkowych API umożliwiających zwrócenie wyników wyszukiwania w postaci pliku XML czy też JSON. API te upraszaczją komunikację aplikacji z przeglądarką do zapytań RESTowych typu get w celu otrzymania uporządkowanej struktury danych. **?**

1.3 Wymagania funkcjonalne

1.3.1 Przypadki użycia

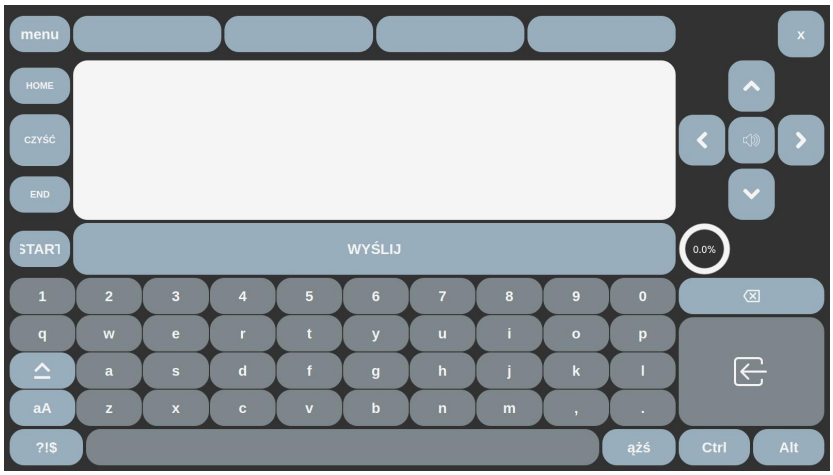


Rys. 1.2 – Wykres przypadków użycia stworzony za pomocą strony Gliffy.com

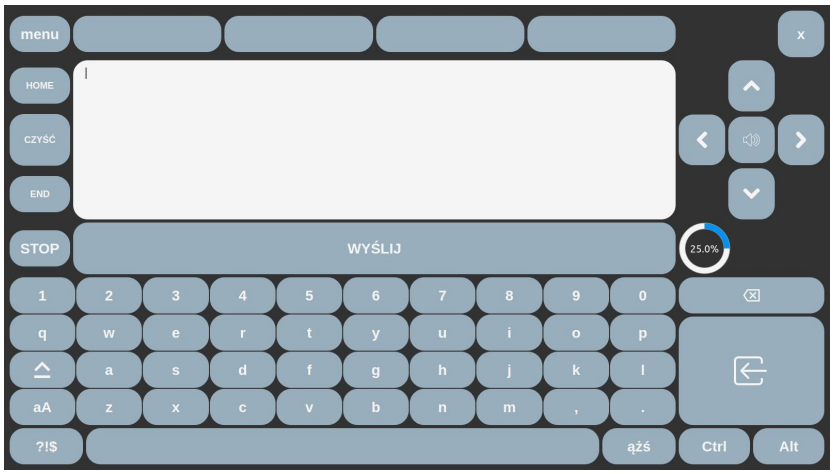
1.4 Propozycja rozwiązania

1.4.1 Zakres projektu

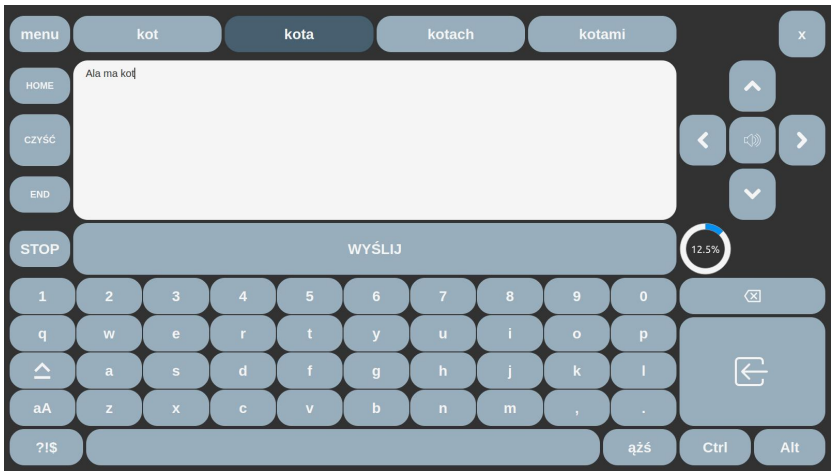
1.4.2 Interfejs oprogramownia



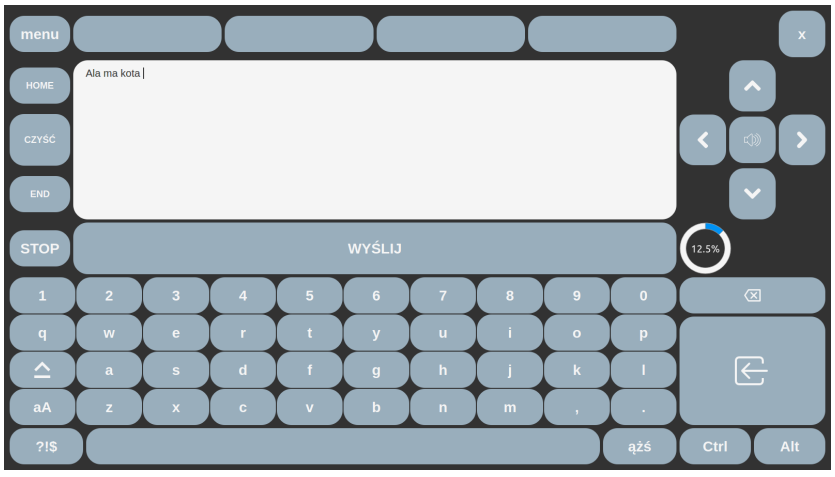
Rys. 1.3 – Początkowy wygląd aplikacji z nieaktywnymi przyciskami.



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