1. Wprowadzenie.
    1.1. Cel.
    1.2. Zakres.
    1.3. Definicje, akronimy i skróty.
    1.4. Referencje, odsyłacze do innych dokumentów.
    1.5. Krótki przegląd.
2. Ogólny opis.
    2.1. Walory użytkowe i przydatność projektowanego systemu.
    2.2. Ogólne możliwości projektowanego systemu.
    2.3. Ogólne ograniczenia.
    2.4. Charakterystyka użytkowników.
    2.5. Środowisko operacyjne.
    2.6. Założenia i zależności.
3. Specyficzne wymagania.
    3.1. Wymagania funkcjonalne (funkcje systemu).
    3.2. Wymagania niefunkcjonalne (ograniczenia).
4. Dodatki.
    4.1. Harmonogram prac nad projektem.

1. Wprowadzenie.
   1.1. Cel.
   Dokument ten opisuje wymagania jakie ma spełnić oprogramowanie i jego funkcję.
   Dokument przeznaczony dla programistów, testerów i użytkowników.
   1.2. Zakres.
   Produkt będzie szyfrował tekst w celu ochrony danych. Ze względu na liczne ataki hakerów i ataki na prywatność użytkownika, aplikacja okaże się przydatnym oprogramowaniem.
   Powstaje z myślą o osobach prywatnych chcących wykorzystać szyfrowanie w celach bezpieczeństwa bądź rozrywkowych.
   1.3. Definicje, akronimy i skróty.
   RSA - Algorytm Rivesta-Shamira-Adlemana – popularny algorytm kryptograficzny używający dwóch kluczy: klucz publiczny (szyfrujący) i klucz prywatny (deszyfrujący).
   Plik txt - plik tekstowy.
   Szyfrowanie / Kodowanie - zapisywanie informacji za pomocą szyfru / kodu.
   Deszyfracja / Odkodowanie - odczytywanie informacji zapisanej za pomocą szyfru / kodu.
   1.4. Referencje, odsyłacze do innych dokumentów.
   Nie dotyczy.
   1.5. Krótki przegląd.
   W rozdziale 2. został przedstawiony opis oprogramowania.
   W rozdziale 3. przedstawione są funkcje oprogramowania.
   W dodatkach przedstawiony będzie harmonogram do projektu.

2. Ogólny opis.
   2.1. Walory użytkowe i przydatność projektowanego systemu
   Program ma być w stanie skutecznie zabezpieczyć tekst przed niepowołanymi odbiorcami. Szyfrując algorytmem RSA przy odpowiednik ukryciu przez użytkownika klucza prywatnego szyfr jest na obecne czasy nie do złamania. Ilość obliczeń potrzebnych do złamania kodu bez klucza zajmuje dziesiątki lat. Jest to więc skuteczna metoda wykorzystywana na przykład przez banki. Użytkownik będzie mógł być spokojny o swoje dane.
   2.2. Ogólne możliwości projektowanego systemu.
   -Wytworzenie pary kluczy (klucza prywatnego i publicznego).
   -Szyfrowanie wprowadzonego teksu za pomocą klucza publicznego.
   -Szyfrowanie pliku txt za pomocą klucza publicznego.
   -Deszyfrowanie wprowadzonego teksu za pomocą klucza prywatnego.
   -Deszyfrowanie pliku txt za pomocą klucza prywatnego.
   2.3. Ogólne ograniczenia.
   -Liczby pierwsze użyte do wytworzenia kluczy muszą być usunięte.
   -Klucz prywatny musi być zabezpieczony.
   -Obliczenia wykonywane przy szyfrowaniu i deszyfrowaniu muszą być w zasięgu sprzętu na którym są wykonywane (dzisiejsze komputery).

2.4. Charakterystyka użytkowników.

Użytkownik – Osoba szyfrująca lub deszyfrująca tekst.

Przypadki użycia programu przez użytkownika:

-Tworzenie pary kluczy (prywatny i publiczny).

-Szyfrowanie danych.

-Odczytywanie zaszyfrowanych danych.

2.5. Środowisko operacyjne.

-Do prawidłowego funkcjonowania programu wymagany będzie system operacyjny zgodny z rodziną systemów Windows.

2.6. Założenia i zależności.

Komputer użytkownika musi być wyposażony minimalnie w:

-System operacyjny zgodny z rodziną systemów Windows.

-Procesor o taktowaniu 400 MHz.

-Pamięć RAM o wielkości 128MB.

-Ok. 200MB ilości wolnego miejsca na dysku twardym.

-Karta graficzna i monitor.

-Mysz, klawiatura.

Zależności: Brak zależności.

3. Specyficzne wymagania.

3.1. Wymagania funkcjonalne (funkcje systemu).

3.2. Wymagania niefunkcjonalne (ograniczenia).

4. Dodatki.

4.1. Harmonogram prac nad projektem.