**Zakład Informatyki Gospodarczej**

**Wydział Zarządzania**

**Politechnika Warszawska**

**PROJEKT SYSTEMU**

|  |  |  |
| --- | --- | --- |
| Temat | **Konwerter liczb** | |
| Wersja  dokumentu | **V** | |
| Zespół | *Grupa ISJ 5.2* | *Studia: Stacjonarne.*  *Kierunek: Zarządzanie i Inżynieria Produkcji.* |
| *Imię i NAZWISKO* | Ocena |
| *Kierownik* | *1. Emil Faliński* |  |
|  | *2. Bartłomiej Dymek* |  |
| *3. Konrad Dziejarski* |  |
| *4. Mariusz Bierzgalski* |  |
| *Podpis przekazującego* |  |
|  | | |

1. **Streszczenie**

Dokument zawiera informacje na temat aplikacji zamiany dowolnych liczb zapisanych w dowolnym systemie (od dwójkowego do dwudziestkowego a także systemu liczb rzymskich) na dowolny system liczbowy zawierający się w przedziale od kodu binarnego do systemu dwudziestkowego a także umożliwia konwersje liczby na system rzymski. Rozpoczęto od ogólnego przedstawienia funkcjonalności programu, jego zakresu oraz korzyści płynących z jego wykorzystania. Projekt opisuje pełen zakres czynności, które może wykonać użytkownik aplikacji. Przedstawiony został diagram klas obrazujące szczegółowy sposób działania programu poprzez jego klasy oraz zależności między nimi zachodzące (generalizacja – dziedziczenia, oraz bardziej ogólne zależności – dependency). Została wpisana także perspektywa logiczna klas oraz niezbędne atrybuty. W dokumencie wyspecyfikowano przybliżoną konfigurację sprzętową maszyn na których aplikacja ma pracować oraz określono oprogramowanie niezbędne do prawidłowego działania programu. Projekt zawiera wizualizację interfejsu pozwalającą na sterowanie aplikacją. W dokumencie można odnaleźć informacje o przewidywanych kosztach wdrożenia systemu, oraz rozpiskę dotyczącą harmonogramu prac zespołu projektowego. Końcowe rozdziały mówią o sposobie wdrożenia i implementacji systemu, ewentualnych problemach które mogą wystąpić w tym etapie oraz propozycje ich rozwiązania. Do projektu jest dołączona dokumentacja zawierająca dodatkowe informacje dotyczące projektowanego systemu.

1. **Zmiany w stosunku do wersji poprzedniej**

**brak**

1. **Spis treści**

[1. Streszczenie 2](#_Toc317356816)

[2. Zmiany w stosunku do wersji poprzedniej 2](#_Toc317356817)

[3. Spis treści 3](#_Toc317356818)

[4. Faza strategiczna 4](#_Toc317356819)

[4.1. Cel projektu 4](#_Toc317356820)

[4.2. Zakres 4](#_Toc317356821)

[**4.3.** **Zdefiniowanie obszaru** 5](#_Toc317356822)

[4.4. Charakterystyka obszaru podmiotowego 5](#_Toc317356823)

[4.5. Definicje, akronimy i skróty 5](#_Toc317356824)

[4.6. Referencje, odsyłacze do innych dokumentów 5](#_Toc317356825)

[4.7. Harmonogram 5](#_Toc317356826)

[4.8. Wstępne szacownie kosztów 6](#_Toc317356827)

[5. Określenie wymagań 6](#_Toc317356828)

[5.1. Ogólny opis 6](#_Toc317356829)

[5.1.1. Walory użytkowe i przydatność projektowanego systemu 6](#_Toc317356830)

[5.1.2. Ogólne możliwości projektowanego systemu 6](#_Toc317356831)

[5.1.3. Ogólne ograniczenia 6](#_Toc317356832)

[5.1.4. Charakterystyka użytkowników 6](#_Toc317356833)

[5.1.5. Środowisko operacyjne 6](#_Toc317356834)

[5.1.6. Założenia i zależności 7](#_Toc317356835)

[**6.** **Wymagania dotyczące funkcji systemu** 7](#_Toc317356836)

[**7.** **Inne wymagania** 9](#_Toc317356837)

[**7.1.** **Wymagania dotyczące wydajności systemu** 9](#_Toc317356838)

[**7.2.** **Wymagania dotyczące zewnętrznych interfejsów** 9](#_Toc317356839)

[**7.3.** **Wymagania dotyczące wykonywanych operacji** 9](#_Toc317356840)

[**7.4.** **Wymagania dotyczące wymaganych zasobów** 9](#_Toc317356841)

[**7.5.** **Wymagania dotyczące sposobów weryfikacji** 10](#_Toc317356842)

[**7.6.** **Wymagania dotyczące sposobów testowania** 10](#_Toc317356843)

[**7.7.** **Wymagania dotyczące ochrony** 10](#_Toc317356844)

[**7.8.** **Wymagania dotyczące dokumentacji** 10](#_Toc317356845)

[**7.9.** **Wymagania dotyczące przenośności** 10](#_Toc317356846)

[**7.10.** **Wymagania dotyczące jakości** 10](#_Toc317356847)

[**7.11.** **Wymagania dotyczące niezawodności** 10](#_Toc317356848)

[**7.12.** **Wymagania dotyczące pielęgnacyjności** 10](#_Toc317356849)

[**7.13.** **Wymagania dotyczące bezpieczeństwa** 10](#_Toc317356850)

[7.14. Specyficzne wymagania 10](#_Toc317356851)

[8. Analiza 11](#_Toc317356852)

[**8.1.** **Model danych** 13](#_Toc317356853)

[**8.2.** **Modele dynamiczne systemu** 13](#_Toc317356854)

[9. Projekt systemu 14](#_Toc317356855)

[**9.1.** **Projekt rozwiązania w sieci** 14](#_Toc317356856)

[**9.2.** **Projekt konfiguracji i połączenia serwerów** 14](#_Toc317356857)

[**9.3.** **Projekt Interfejsu użytkownika** 15](#_Toc317356858)

[**9.3.1.** **Projekt nawigacji po aplikacji (systemie)** 15](#_Toc317356859)

[**9.3.2.** **Formularze (okna użytkowników)** 16](#_Toc317356860)

[**9.3.3.** **Raporty** 15](#_Toc317356861)

[**9.3.4.** **Komendy linii poleceń** 16](#_Toc317356862)

[10. Implementacja 16](#_Toc317356863)

[11. Testowanie 16](#_Toc317356864)

[12. Instalacja 16](#_Toc317356865)

[13. Eksploatacja 16](#_Toc317356866)

[14. Dokumentacja 17](#_Toc317356867)

1. **Faza strategiczna**

Nasza firma zajmuje się realizacją oprogramowania na konkretne zamówienie, dlatego projekt systemu powstaje w ścisłej współpracy z klientem. Po dokonaniu serii rozmów (wywiadów) z klientem, jesteśmy w stanie określić, czego oczekuje klient po systemie i jak według klienta powinien wyglądać efekt końcowy. Klient Indywidualny (rektor uczelni) zlecił wykonanie projektu konwertera liczb. Cele przedsięwzięcia z punktu widzenia klienta mogą wydawać się oczywiste, jednak dzięki serii rozmów z klientem możemy uniknąć ewentualnych nieporozumień. Klient oczekuje, że nasza firma, stworzy oprogramowanie, które pozwoli na dokonywanie konwersji liczb pomiędzy wersją arabską, rzymską, binarną a także z każdego z systemu dwójkowego, trójkowego, czwórkowego, piątkowego, szóstkowego, siódemkowego, ósemkowego, dziesiątkowego, dziesiętnego, jedenastkowego, dwunastkowego, trzynastkowego, czternastkowego, piętnastkowego, szesnastkowego, siedemnastkowego, osiemnastkowego, dziewiętnastkowego i dwudziestkowego na dowolny inny wymieniony powyżej. Co w jego mniemaniu pozwoli na naukę oraz sprawdzanie poprawności wykonywanych obliczeń przez studentów oraz wykładowców na uczelni. Odpowiednie określenie zakresu przedsięwzięcia pozwoli na późniejsze dokładne oszacowanie kosztów. Program obsługiwać będzie wielu użytkowników. Wyspecyfikowaliśmy procesy realizowane przez program i według wstępnej analizy powinien wykonywać następujące zadania:

* Wczytywanie liczb z plików tekstowych
* Zapis wyników pracy do plików tekstowych
* Zapisywanie historii wykonanych zadań
* Dokonywanie konwersji:

dwójkowy -> trójkowy

dwójkowy –> czwórkowy

dwójkowy -> piątkowy

dwójkowy -> szóstkowy

dwójkowy -> siódemkowy

dwójkowy -> ósemkowy

dwójkowy -> dziesiątkowy

**.**

**.**

**.**

dwójkowy -> dwudziestkowy

trójkowy -> dwójkowy

trójkowy -> czwórkowy

**.**

**.**

**.**

dwudziestkowy -> siedemnastkowy

dwudziestkowy -> osiemnastkowy

dwudziestkowy -> dziewiętnastkowy

Aplikacja zostanie stworzona w języku programowania C#. Podczas pracy nad programem zostanie wykorzystane oprogramowanie Microsoft Visual Studio Express Edition 2012.

Wstępnie ustalona wycena usługi stworzenia aplikacji umożliwiającej konwersję liczb na około 20 tys. złotych. Firma i klient mają jednak świadomość, że to kwota bardzo orientacyjna i na pewno ulegnie ona zmianie.

* 1. **Cel projektu**

Celem projektu jest stworzenie w pełni sprawnego konwertera liczb, który umożliwi użytkownikom szybszą i sprawniejszą pracę z liczbami. Z punktu widzenia klienta celem przedsięwzięcia jest zautomatyzowanie wielu obliczeń wykonywanych dotąd manualnie, a także pomoc studentom i wykładowcom przy pracy na liczbach oraz ich konwersji,

* 1. **Zakres**

Proces przetwarzania danych liczbowych związany z zamianą liczb na poszczególne formy może zostać wydatnie zoptymalizowany poprzez jego automatyzację. Aplikacja pozwoli na pominięcie szeregu czynności jak np. w przypadku zamiany na system binarny:

721 na binarny

* 721:2 = 360, reszta **1**
* 360:2 = 180, reszta **0**
* 180:2 = 90, reszta **0**
* 90:2 = 45, reszta **0**
* 45:2 = 22, reszta **1**
* 22:2 = 11, reszta  **0**
* 11:2 = 5, reszta **1**
* 5:2 = 2, reszta  **1**
* 2:2 = 1, reszta **0**
* 1:2 = 0, reszta **1**

Co nam daje liczbę: **1011010001**

* 1. **Zdefiniowanie obszaru**

Aplikacja będzie narzędziem, którego będzie można używać na każdym komputerze wyposażonym w system Windows a także dodatek do systemu NET.Framework wersji od 3.0. Zarówno w biurze jak i w domu – to jest bardzo duża zaleta naszej aplikacji.

* 1. **Charakterystyka obszaru podmiotowego**

Obszarem podmiotowym dla projektowanej aplikacji będą laboratoria uczelniane. Aplikacja będzie również udostępniana do użytku studentom i wykładowcom w celach edukacyjnych.

* 1. **Definicje, akronimy i skróty**

**Generalizacja** – związek opisujący dziedziczenie po klasach. Na diagramie generalizację oznacza się za pomocą niewypełnionego trójkąta symbolizującego strzałkę (skierowaną od klasy pochodnej do klasy bazowej).

**Zależność** – najsłabszy związek znaczeniowy między klasami, gdy jedna z nich używa innych klas.

**.NET Framework**, w skrócie **.NET** (wym. *dot net*) – platforma programistyczna opracowana przez Microsoft, obejmująca środowisko uruchomieniowe (Common Language Runtime – CLR) oraz biblioteki klas dostarczające standardowej funkcjonalności dla aplikacji. Technologia ta nie jest związana z żadnym konkretnym językiem programowania, a programy mogą być pisane w jednym z wielu języków – na przykład C++/CLI, C#, F#, J#, Delphi 8 dla .NET, Visual Basic .NET.

**C#** (C Sharp, dosłownie "C-krzyżyk", "cis") – obiektowy język programowania zaprojektowany przez zespół pod kierunkiem Andersa Hejlsberga dla firmy Microsoft.Program napisany w tym języku kompilowany jest do języka Common Intermediate Language (CIL), specjalnego kodu pośredniego wykonywanego w środowisku uruchomieniowym takim jak .NET Framework, Mono lub DotGNU. Wykonanie skompilowanego programu przez system operacyjny bez takiego środowiska nie jest możliwe.

* 1. **Referencje, odsyłacze do innych dokumentów**

Brak.

* 1. **Harmonogram**

W załączeniu do projektu, plik „Harmonogram projektu” wykonany w programie Microsoft Project 2010.

* 1. **Wstępne szacownie kosztów**

Dokument Kosztorys zawarty wraz z projektem.

1. **Określenie wymagań**

Ze względu na charakterystykę i wbudowaną funkcjonalność projektowany system posiada szereg wymagań dotyczących sprzętu, środowiska operacyjnego. Wymagania sprzętowe należą do grupy wymagań funkcjonalnych natomiast środowisko operacyjne jest wymaganiem nie funkcjonalnym

* 1. **Ogólny opis**

Projektowana aplikacja jest w pełni autorskim rozwiązaniem, jej funkcjonalności umożliwiają sprawną pracę na liczbach.

* + 1. **Walory użytkowe i przydatność projektowanego systemu**

Walory użytkowe:

* Konwersja wielu typów liczb
* Wczytywanie liczb z plików zewnętrznych
* Zapisywanie wyników pracy do plików
* Zapisywanie historii działań
* Mobilność

Przydatność projektowanej aplikacji

* Aplikacja jest szczególnie przydatna do celów edukacyjnych dla studentów i wykładowców uczelni, dla której projektujemy aplikacje
  + 1. **Ogólne możliwości projektowanego systemu**

Aplikacja umożliwia konwersję liczb, oraz pracę na plikach.

* + 1. **Ogólne ograniczenia**

Minimalna specyfikacja komputera

|  |  |
| --- | --- |
| **System operacyjny:** | **Windows XP** |
| **Pamięć RAM:** | **1GB, DDR 3 1066Mhz** |
| **Dysk HDD:** | **10 MB wolnego miejsca** |
| **Procesor** | **Intel Pentium 4** |
| **Zasilacz** | **750W** |
| **Karta sieciowa:** | **Gbit Ethernet 10/100/1000 ze wsparciem technologii TCI/IP , iSCSI,** |
| **Karta graficzna** | **Intel HD 3000** |
| **Napęd dysków optycznych** | **CD/DVD** |

* + 1. **Charakterystyka użytkowników**

Wykładowcy - odbiorcy programu będą go używać w celach dydaktycznych. Pokażą studentom jak skorzystać z niego w praktyce oraz uczyć jego obsługi.

Studenci natomiast mogą wykonywać za pomocą programu proste konwersje liczbowe przydatne na studiach w celu sprawdzenia poprawności wykonanych ręcznie obliczeń.

* + 1. **Środowisko operacyjne**

Środowiskiem operacyjnym będzie komputer typu Desktop z oprogramowaniem Windows XP lub nowszym firmy Microsoft w zależności od wyboru użytkownika. Komputer powinien mieć zainstalowany najnowszy Service Pack i NET Framework (3.0 i wyższy)

* + 1. **Założenia i zależności**

System powinien:

* Nie sprawiać problemów w obsłudze.
* Natychmiastowo reagować na polecenia użytkownika.
* Być niezawodny (nie zawieszać się).
* Wyróżniać się dodatkowymi rozwiązaniami w porównaniu z konkurencją rynkową.
* Zostać wykonany zgodnie z projektem.

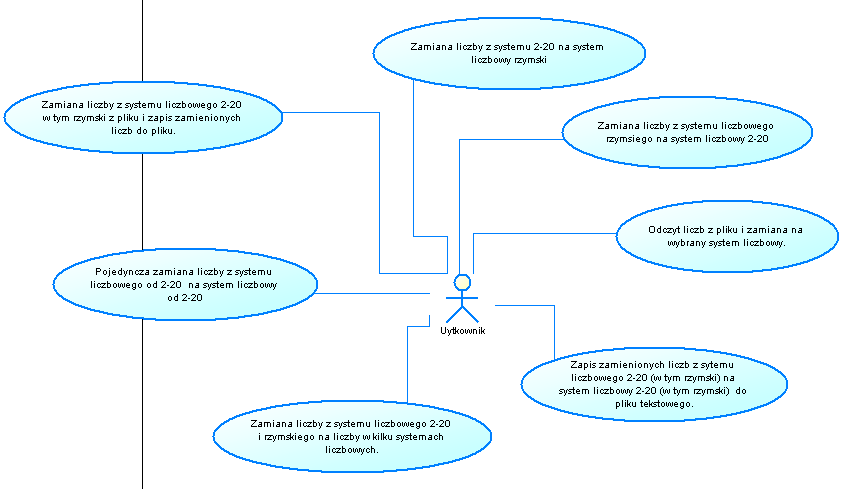
Zamawiający posiada minimalną konfigurację komputera w firmie opisaną w rozdziale 5.1.3 Zakłada się, że platforma ta jest wystarczająca do funkcjonowania aplikacji.

**Zależności pomiędzy akcją użytkownika, zdarzeniami losowymi i działaniem programu:**

* Jeżeli użytkownik wprowadzi błędny zapis liczby, w którymkolwiek z systemów liczbowych rozpoznawalnych w programie, program wyświetli komunikat o błędzie.
* Jeżeli podczas działania programu utracone zostanie zasilanie z sieci energetycznej, bądź baterii komputera wszystkie dane wprowadzone w obecnie działającej sesji programu zostaną utracone.
* Jeżeli podczas pracy programu, użytkownik zechce zamknąć okno programu wyświetlony zostanie komunikat o możliwości zapisu obecnego stanu sesji.
* Po każdej skończonej sesji programu wyniki pracy programu zostaną zapisane do pliku na dysku komputera na którym przeprowadzane były obliczenia.
* Jeżeli użytkownik nie określi przed rozpoczęciem obliczeń na jaki system liczbowy mają zostać zamienione wprowadzone liczby, liczby te zostaną konwertowane do każdego obsługiwanego systemu liczbowego.

1. **Wymagania dotyczące funkcji systemu**

Diagram przypadków użycia systemu przez użytkownika:

****

**Funkcje przedstawione w formie tabeli:**

|  |  |
| --- | --- |
| **Nazwa funkcji** | **Konwersja liczb** |
| **Opis funkcji** | Funkcja pozwala na dokonanie konwersji pomiędzy wcześniej wymienionymi typami liczb. |
| **Dane wejściowe** | Liczba mająca zostać poddana konwersji oraz rodzaj oczekiwanych wyników |
| **Źródła danych wejściowych** | Klawiatura i mysz |
| **Dane wyjściowe** | Przekonwertowane liczby |
| **Warunki** | Podanie liczby w odpowiedniej postaci; zaznaczenie chociaż jednego typu docelowego |
| **Efekty uboczne** | Jeśli liczba zostanie podana w sposób nieprawidłowy system wyświetli komunikat o błędzie; jeśli nie zostanie zaznaczony chociaż jeden typ docelowy, aplikacja wyśwetli komunikat o błędzie |
| **Powód** | Powodem jest nierozpoznanie wprowadzonej liczby |

|  |  |
| --- | --- |
| **Nazwa funkcji** | **Wczytywanie plików z liczbami** |
| **Opis funkcji** | Funkcja pozwala na wczytanie pliku tekstowymi z liczbami, które mają zostać poddane konwersji |
| **Dane wejściowe** | Plik tekstowy |
| **Źródła danych wejściowych** | Dysk twardy, dysk usb itp |
| **Dane wyjściowe** | Przekonwertowane liczby |
| **Warunki** | Odpowiedni format pliku i odpowiednie sformatowanie tekstu w pliku |
| **Efekty uboczne** | Jeśli liczba zostanie podana w sposób nieprawidłowy system wyświetli komunikat o błędzie; |
| **Powód** | Powodem jest nierozpoznanie wprowadzonej liczby |

|  |  |
| --- | --- |
| **Nazwa funkcji** | **Zapisywanie plików z liczbami** |
| **Opis funkcji** | Funkcja pozwala na zapisanie pliku z liczbami, które zostały poddane konwersji |
| **Dane wejściowe** | Przekonwertowane liczby; nazwa pliku |
| **Źródła danych wejściowych** | Program |
| **Dane wyjściowe** | Plik tekstowy o podanej nazwie |
| **Warunki** | Musi być miejsce w lokalizacji docelowej pliku |
| **Efekty uboczne** | W przypadku braku miejsca aplikacja wyświetli komunikat o błędzie |
| **Powód** | Brak miejsca w lokalizacji docelowej uniemożliwia zapis danych |

|  |  |
| --- | --- |
| **Nazwa funkcji** | **Zapisywanie plików z historią działań** |
| **Opis funkcji** | Funkcja pozwala na zapisanie pliku z liczbami, które zostały poddane konwersji podczas sesji pracy z programem |
| **Dane wejściowe** | Przekonwertowane liczby; nazwa pliku |
| **Źródła danych wejściowych** | Program |
| **Dane wyjściowe** | Plik tekstowy o podanej nazwie |
| **Warunki** | Musi być miejsce w lokalizacji docelowej pliku |
| **Efekty uboczne** | W przypadku braku miejsca aplikacja wyświetli komunikat o błędzie |
| **Powód** | Brak miejsca w lokalizacji docelowej uniemożliwia zapis danych |

1. **Inne wymagania**
   1. **Wymagania dotyczące wydajności systemu**

Czas odpowiedzi aplikacji na pytanie użytkownika nie powinien przekraczać 1,5s.

* 1. **Wymagania dotyczące zewnętrznych interfejsów**

Nie dotyczy.

* 1. **Wymagania dotyczące wykonywanych operacji**
* Weryfikacja danych
  1. **Wymagania dotyczące wymaganych zasobów**

System musi posiadać następujące bazy danych:

* Baza liczb arabskich
* Baza liczb rzymskich
  1. **Wymagania dotyczące sposobów weryfikacji**

Podczas wprowadzania danych do aplikacji system powinien na bieżąco sprawdzać poprawność semantyczna i syntaktyczną danych. Weryfikacja poprawności danych będzie następować poprzez inteligentne porównywanie wprowadzanych danych do wzorców zapisanych w bazie.

* 1. **Wymagania dotyczące sposobów testowania**

Testy powinny być częścią cyklu projektowania tzn. system jest tworzony wg metodologii Agile, która polega na tym, że zespoły produkcyjne, organizują swoją pracę w modelu iteracyjnym, tak, by co 2-4 tygodnie dostarczyć gotowe, działające oprogramowanie.

Testowanie

Projektowanie

Kodowanie

Analiza

Testowanie

Projektowanie

Kodowanie

Analiza

Testowanie

Projektowanie

Kodowanie

Analiza

Rysunek 1: Schemat przedstawiający sposób testowania oprogramowania

Testy powinny się odbywać wraz z użytkownikami na makietach funkcjonalnych w trakcie każdej iteracji.

* 1. **Wymagania dotyczące ochrony**

Nie dotyczy.

* 1. **Wymagania dotyczące dokumentacji**

W dokumentacji projektowanej aplikacji, która będzie dostarczana klientowi wraz z gotowym produktem powinny znaleźć się następujące elementy:

* Krótki opis systemu

Przedstawienie ogólnych możliwości systemu i jego funkcjonalności na sprzyjającym zwykłemu użytkownikowi poziomie szczegółowości.

* Informacje o Gwarancji
* Informacje dotyczące konserwacji
* Wzór formularza do zgłaszania błędów i wymagań
  1. **Wymagania dotyczące przenośności**

Oprogramowanie powinno działać na sprzęcie o minimalnej specyfikacji opisanej w punkcie 5.1.3.

* 1. **Wymagania dotyczące jakości**

Nie dotyczy.

* 1. **Wymagania dotyczące niezawodności**

Aplikacja powinna działać płynnie i się nie zawieszać (obsługa wyjątków try - catch)

* 1. **Wymagania dotyczące pielęgnacyjności**

W trakcie używania aplikacji producent będzie wypuszczał poprawki w postaci patch’ów dostępnych do pobrania na stronie producenta

* 1. **Wymagania dotyczące bezpieczeństwa**

Nie dotyczy.

* 1. **Specyficzne wymagania**
* Łatwość instalacji i konfiguracji systemu w różnych środowiskach zapewniona musi być poprzez m.in. precyzyjną dokumentację instalacji i konfiguracji aplikacji, przetestowane pakiety instalacyjne, itd.

1. **Analiza**

|  |  |
| --- | --- |
| **Nazwa** | *Zamiana liczby z systemu 2-20 na system liczb rzymskich* |
| **Numer** | 1 |
| **Twórca** | *Konrad Dziejarski, Emil Faliński, Bartłomiej Dymek, Mariusz Bierzgalski* |
| **Poziom ważności** | *Ważny* |
| **Typ przypadku użycia** | *Szczegółowy* |
| **Aktorzy** | Użytkownik |
| **Krótki opis** | *Wykonanie zamiany liczb z systemu od 2 do 20 na system liczb rzymskich* |
| **Warunki wstępne** | *Użytkownik musi wpisać liczbę w poprawnym formacie* |
| **Warunki końcowe** | *brak* |
| **Główny przepływ zdarzeń** | 1. *Użytkownik wpisuje liczbę w dowolnym formacie* 2. *Użytkownik wskazuje system lub systemy docelowe zaznaczając odpowiednie pola wyboru* 3. *Użytkownik wybiera polecenie konwertuj* 4. *System sprawdza poprawność wpisanej liczby i wykonuje konwersję* |
| **Alternatywne przepływy zdarzeń** | *4a) Gdy użytkownik nie wpisze poprawnie liczb np. w polu znajdą się litery nie wchodzące w skład liczb rzymskich* |
| **Specjalne wymagania** | 1. *Funkcje sprawdzające poprawność* |

|  |  |
| --- | --- |
| **Nazwa** | *Odczyt liczb z pliku i zamiana na wybrany system liczbowy* |
| **Numer** | 2 |
| **Twórca** | *Konrad Dziejarski, Emil Faliński, Bartłomiej Dymek* |
| **Poziom ważności** | *Ważny* |
| **Typ przypadku użycia** | *Szczegółowy* |
| **Aktorzy** | Użytkownik |
| **Krótki opis** | *Wczytanie liczb z pliku tekstowego i zamiana ich na wskazane systemy liczbowe* |
| **Warunki wstępne** | *Użytkownik musi podać prawidłową ścieżkę dostępu do pliku* |
| **Warunki końcowe** | *Plik tekstowy musi być odpowiednio sformatowany* |
| **Główny przepływ zdarzeń** | 1. *Użytkownik podaje ścieżkę dostępu do pliku tekstowego z liczbami do zamiany* 2. *Użytkownik wskazuje system lub systemy docelowe zaznaczając odpowiednie pola wyboru* 3. *Użytkownik wybiera polecenie konwertuj* 4. *System sprawdza poprawność formatowania pliku i wykonuje konwersję* |
| **Alternatywne przepływy zdarzeń** | *1a) Gdy użytkownik poda nieprawidłową ścieżkę dostępu do pliku*  *3a) Gdy plik jest niewłaściwie sformatowany* |
| **Specjalne wymagania** | 1. *Funkcje sprawdzające poprawność* |

|  |  |
| --- | --- |
| **Nazwa** | *Zapis zamienionych liczb do pliku* |
| **Numer** | 3 |
| **Twórca** | *Konrad Dziejarski, Emil Faliński, Bartłomiej Dymek* |
| **Poziom ważności** | *Ważny* |
| **Typ przypadku użycia** | *Szczegółowy* |
| **Aktorzy** | Użytkownik |
| **Krótki opis** | *Po wykonaniu zamiany liczb dokonuje się zapisu tychże liczb do pliku tekstowego* |
| **Warunki wstępne** | *Użytkownik musi podać prawidłową ścieżkę dostępu do pliku* |
| **Warunki końcowe** | *Na dysku musi być wystarczająca ilość miejsca* |
| **Główny przepływ zdarzeń** | 1. *Użytkownik wpisuje liczbę w dowolnym formacie* 2. *Użytkownik wskazuje system lub systemy docelowe zaznaczając odpowiednie pola wyboru* 3. *Użytkownik wybiera polecenie konwertuj* 4. *System sprawdza poprawność wpisanej liczby i wykonuje konwersję* 5. *Użytkownik wybiera opcję z menu rozwijanego ZAPISZ* 6. *Użytkownik podaje ścieżkę gdzie mają zostać zapisane zamienione liczby* 7. *Użytkownik wybiera polecenie ZAPISZ* 8. *System sprawdza dostępne miejsce na dysku i zapisuje plik* |
| **Alternatywne przepływy zdarzeń** | *4a) Gdy użytkownik nie wpisze poprawnie liczb np. w polu znajdą się litery nie wchodzące w skład liczb rzymskich*  *3a) Na dysku nie ma wystarczającej ilości miejsca* |
| **Specjalne wymagania** | 1. *Funkcje sprawdzające poprawność* |

* 1. **Model danych**

Nie dotyczy.

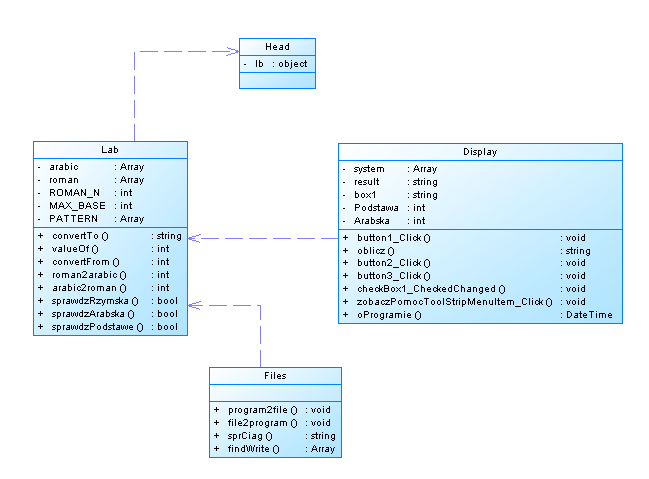
* 1. **Modele dynamiczne systemu**

Diagram klas:

1. **Projekt systemu**
   1. **Projekt rozwiązania w sieci**

Nie dotyczy.

* 1. **Projekt konfiguracji i połączenia serwerów**

Nie dotyczy.

* 1. **Projekt Interfejsu użytkownika** 
     1. **Projekt nawigacji po aplikacji (systemie)**

PODAJ LICZBĘ

Otwórz…

Zapisz

Plik

Wyjście

Konwertuj

Historia

Wybierz systemy liczb

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

Opcja

**Wyniki:**

Opcja

Opcja

Opcja

**I**nterfejs aplikacji został tak zaprojektowany, aby nawigacja po konwerterze nie sprawiała problemów. Dostęp do operacji na plikach jest umieszczony wg powszechnie panujących konwencji w górnym pasku funkcyjnym, pod hasłem „Plik”. Przycisk funkcyjne, kluczowy z punktu widzenia aplikacji został wyróżniony na tle reszty elementów okna aplikacji. Miejsce umieszczenia przycisku konwertuj zostało ustalone w taki sposób, aby użytkownik „po drodze” do przycisku przeszedł przez wszystkie kroki wykonywania konwersji liczb. Pozwoli to na zredukowanie potencjalnych błędów użytkownika polegających na braku wpisania, lub zaznaczenia jakiejś opcji.

* + 1. **Formularze (okna użytkowników)**

Nie dotyczy.

* + 1. **Raporty**

Nie dotyczy.

* + 1. **Komendy linii poleceń**

Nie dotyczy.

1. **Implementacja**

Implementacja polega na wprowadzeniu w życie i realizacji projektu systemu. Środowisko implementacji musi dostarczać takich zasobów, aby można było w nim program umieścić, uruchomić i użytkować.

Zastosowana zostanie całościowa strategia wdrażania aplikacji, a nie włączanie części aplikacji po kolei. Ta strategia jest najbezpieczniejsza i jednocześnie najmniej kosztowna.

Aplikacja wymagała będzie zastosowania komputera stanowiącego jej inicjacje :

* Wykonującego program aplikacji o wymaganiach minimalnych podanych w rozdziale 5.1.3.

Aplikacja jest programem o dość prostej budowie co nie wymaga zróżnicowanego podejścia do zagadnienia programowania, dlatego też zaleca się stosowanie języka programowania wysokopoziomowego Java/C#.

Do programowania aplikacji posłuży nam jednak platforma Visual Studio 2012 Express Edition ze względu na szybką możliwość tworzenia graficznych interfejsów i możliwie najskuteczniejsze zaprogramowanie podanych przez klienta funkcji. Język C# jest w pełni obiektowy i zgodny z narzędziem użytym do modelowania diagramu klas (Power Designer) co pozwala na skrócenie czasu wypisywania klas, atrybutów oraz metod do kodu źródłowego programu.

Przedstawione w poprzednich podpunktach projektu wymagania dotyczące tworzonej aplikacji, determinują jej architekturę.

1. **Testowanie**

Mierzenie czasu wykonania ustalonych scenariuszy użycia systemu. Analiza ścieżki użytkownika. jego przemyśleń. Test powinien być przeprowadzony w przestronnym pokoju imitującym warunki korzystania z systemu. Prowadzący badanie powinien siedzieć w takim miejscu, aby dobrze widzieć reakcje badanego, a jednocześnie zminimalizować presję na badanym. Stanowisko badawcze będzie wyposażone w: kamerę, która będzie rejestrowała badanego oraz ekran, a także urządzenie śledzące ruch gałek ocznych.

Test łatwości użycia dotyczy interfejsu użytkownika. Testowane będą wszystkie udostępniane funkcjonalności w ramach każdej iteracji.

1. **Instalacja**

Instalacja nie jest wymagana, jest to wersja portable. Wymagane natomiast jest posiadanie Net. Frameworka 3.0 lub nowszego a także systemu Microsoft Windows od XP.

1. **Eksploatacja**

Eksploatacją programu zajmują się jego użytkownicy. Wszelka eksploatacja programu polegająca na ingerencji w jego kod źródłowy jest zabroniona i narusza prawo patentowe. Eksploatacja programu konwertującego liczby polegać będzie na sekwencji działań prowadzących w efekcie do zamiany zapisu liczb na zadany system liczbowy. Użytkownicy mają możliwość wpisania zestawu liczb zapisanych w dowolnym obsługiwanym przez program systemie liczbowym ręcznie bądź wczytania ich z pliku .txt do programu. Następnym krokiem jest określenie na jaki zapis (w którym systemie liczbowym) zadane liczby mają zostać przekonwertowane i rozpoczęcie działanie programu. Wyniki mogą zostać wyświetlone na monitorze, zapisane do pliku .txt na dysku komputera, bądź obie opcje jednocześnie. Jeśli użytkownicy zauważą jakikolwiek problem w poprawności działania programu, mogą liczyć na pomoc twórców programu, którzy zobowiązują się do wydawania patchy uaktualniających wersję programu. Konwerter liczb nie jest złośliwym oprogramowaniem i twórcy nie biorą odpowiedzialności za nieodpowiednie używanie programu przez użytkowników i wszelkie szkody nim spowodowane.­­­­

1. **Dokumentacja**

* Niniejszy dokument
* Rejestr spotkań roboczych zespołu
* Harmonogram projektu
* Harmonogram zespołu
* Wstępny kosztorys