

<b>Nazwa i akronim projektu:</b> Szukanie ekstremów funkcji różnymi algorytmami / ESS (Extrema System Search)	<b>Zleceniodawca:</b> Projekt własny	<b>Zleceńbiorca:</b> PG, WFTiMS, zespół projektowy IO nr 001
<b>Numer zlecenia:</b> PG-WFTiMS-IO-2010-001	<b>Kierownik projektu:</b> Michał Jagielski	<b>Opiekun projektu:</b> Prof. dr hab. Józef E. Sienkiewicz

<b>Analiza i wizja systemu (AWS)</b>	<b>Nr wersji:</b> 1.0.0
<b>Odpowiedzialny za dokument:</b> Maciej Dąbrowski	<b>Data pierwszego sporządzenia:</b> 23-05-2018
	<b>Data ostatniej aktualizacji:</b> 23-05-2018

Historia dokumentu				
Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.0.0	wstępna wersja	całość	Maciej Dąbrowski	23-05-2018
{wersja}	{np. poprawka harmonogramu}	{np. punkt 3.3}	{nazwisko, imię}	{data zmiany}

## 1 Ogólne cele systemu

Oferowany przez nas produkt będzie udostępniany w formie freeware z reklamami i możliwością dotacji. System zostanie opublikowany w ramach licencji Berkeley Software Distribution. Oczekujemy, że środowiska naukowe takie jak uczelnie (studenci) będą chętnie korzystać z udostępnionej przez nas aplikacji z powodu braku środków na profesjonalne oprogramowanie. Chcemy skupić się przede wszystkim na dziedzinie architektury/budownictwa – aplikacja będzie umożliwiała np. zaprojektowanie takiego obiektu (np. śruby) aby jej budowa pochłaniała jak najmniej materiału, przy czym była jak najlżejsza i bardzo wytrzymała, a więc najbardziej optymalny obiekt przy szeregu warunków ograniczających. Wstępny wywiad wskazuje na niskie zainteresowanie daną dziedziną i tworzy lukę w której produkt staje się prekursorem.

## 2 Kontekst systemu

- Ludzie (osoby powiązane z środowiskiem naukowym i nie tylko)
- Systemy współpracujące (system operacyjny)
- Zdarzenia (techniczne lub fizyczne)
- Dokumenty (np. normy prawne, standardy, dokumentacja systemowa)

## 3 Zakres funkcjonalności

### Student/Uczeń: (użytkownik mniej doświadczony)

- Możliwość wyboru algorytmu - użytkownik ma dostęp do bardziej zaawansowanych jak i najprostszych algorytmów
- Możliwość wprowadzenia danych - użytkownik sam wprowadza dane po wyborze potrzebnego algorytmu
- Informacja o błędach - system podaje typ błędu oraz proponuje sposób prawidłowego wprowadzenia danych
- otrzymanie wiarygodnego i dokładnego wyniku - wynik podany z dobrą dokładnością, a także w sposób ciekawy
- przejrzystość interfejsu użytkownika oraz opisy pomocnicze - przejrzystość menu i dodatkowe opisy algorytmów przybliżające ich działanie
- szybkość i stabilność aplikacji - podstawowa wartość bardzo pożądana w dzisiejszym świecie

### Pracownik Naukowy: (użytkownik zaawansowany)

- Możliwość wyboru algorytmu - użytkownik ma szeroki dostęp do bardziej zaawansowanych algorytmów
- Możliwość wprowadzenia danych - użytkownik sam wprowadza dane po wyborze potrzebnego algorytmu
- Informacja o błędach - system podaje typ błędu oraz proponuje sposób prawidłowego wprowadzenia danych
- otrzymanie wiarygodnego i dokładnego wyniku - system podaje wyniki z dużą dokładnością liczbową, a jeżeli to możliwe, także graficzną
- przejrzystość interfejsu użytkownika oraz opisy pomocnicze - przejrzystość i intuicyjność menu zachęcająca do pracy z programem
- szybkość i stabilność aplikacji - podstawowa wartość bardzo pożądana w dzisiejszym świecie

### Klient Pracownik: (zastosowanie komercyjne)

- Możliwość wyboru algorytmu - użytkownik ma szeroki dostęp do bardziej zaawansowanych algorytmów
- Możliwość wprowadzenia danych - użytkownik sam wprowadza dane po wyborze potrzebnego algorytmu
- Informacja o błędach - system podaje typ błędu oraz proponuje sposób prawidłowego wprowadzenia danych
- otrzymanie wiarygodnego i dokładnego wyniku - system podaje wyniki z dużą dokładnością liczbową, a jeżeli to możliwe, także graficzną
- przejrzystość interfejsu użytkownika oraz opisy pomocnicze - przejrzystość i intuicyjność menu zachęcająca do pracy z programem, oraz dodatkowe opisy algorytmów przybliżające ich działanie
- szybkość i stabilność aplikacji - podstawowa wartość bardzo pożądana w dzisiejszym świecie

### Klient Hobbysta: (zastosowanie domowe)

- Możliwość wyboru algorytmu - użytkownik ma szeroki dostęp do bardziej zaawansowanych algorytmów
- Możliwość wprowadzenia danych - użytkownik sam wprowadza dane po wyborze potrzebnego algorytmu
- Informacja o błędach - system podaje typ błędu oraz proponuje sposób prawidłowego wprowadzenia danych
- otrzymanie wiarygodnego i dokładnego wyniku - system podaje wyniki z dobrą dokładnością liczbową
- przejrzystość interfejsu użytkownika oraz opisy pomocnicze - przejrzystość menu i dodatkowe opisy algorytmów przybliżające ich działanie
- szybkość i stabilność aplikacji - podstawowa wartość bardzo pożądana w dzisiejszym świecie

### 3.1. Diagram przypadków użycia



### 3.2. Opis aktorów



**Aktor1** – Pracownik Naukowy, aktywny, główny



**Aktor2** –Uczeń, aktywny, główny



**Aktor3** – Klient Pracownik, aktywny, główny



**Aktor4** – Klient Hobbysta, aktywny, główny

## 3.3. Opis przypadków użycia

<b>UC01</b>	<b>Wybór Algorytmu</b>
<b>Aktorzy</b>	Wszyscy
<b>Streszczenie</b>	Użytkownik na samym starcie programu wybiera jeden z dostępnych algorytmów
<b>Warunki początkowe</b>	Uruchomiona aplikacja w menu głównym
<b>Zdarzenie inicjujące</b>	Wybór przycisku "Oblicz Ekstremum"
<b>Struktura</b>	Przypadek użycia jest rozszerzany przez przypadek użycia [UC02]
<b>Przebieg podstawowy</b>	1. Przed wyborem algorytmu może przeczytać sposób jego działania 2. Kolejne kroki zgodnie z przypadkiem użycia [UC02]
<b>Sytuacje wyjątkowe</b>	Użytkownik nie wybiera żadnego algorytmu wybiera przycisk "Cofnij"
<b>Warunki końcowe</b>	{opis stanu systemu po zakończeniu scenariusza}
<b>Komentarz</b>	-

<b>UC02</b>	<b>Wprowadzenie danych</b>
<b>Aktorzy</b>	Wszyscy
<b>Streszczenie</b>	Po wybranym Algorytmie użytkownik wprowadza dane
<b>Warunki początkowe</b>	Uruchomiona aplikacja w menu głównym
<b>Zdarzenie inicjujące</b>	Wybór przycisku Oblicz Ekstremum
<b>Struktura</b>	Przypadek użycia używa przypadku użycia [UC01]
<b>Przebieg podstawowy</b>	1. Użytkownik wybiera algorytm z podanej listy 2. Wyświetla się okno algorytmu z polami do wypełnienia 3. Użytkownik wypełnia pola i zatwierdza przyciskiem "Wynik" 4. Wyświetla się okno z rezultatem w postaci liczbowej i graficznej(opcjonalnie) 5. <i>Wymyślamy jakiś sposób zapisu?</i>
<b>Sytuacje wyjątkowe</b>	Błędne wprowadzenie danych przez użytkownika cofa go do listy algorytmów i wyświetla komunikat o rodzaju błędu
<b>Warunki końcowe</b>	Użytkownik wraca do listy algorytmów
<b>Komentarz</b>	-

<b>UC03</b>	<b>Kontakt z Supportem</b>
<b>Aktorzy</b>	Wszyscy
<b>Streszczenie</b>	Użytkownicy mogą zgłaszać uwagi/błędy/propozycje bezpośrednio z aplikacji
<b>Warunki początkowe</b>	Uruchomiona aplikacja w podstawowym menu
<b>Zdarzenie inicjujące</b>	Wybór przycisku "Napisz Do Nas"
<b>Struktura</b>	brak powiązań z innymi przypadkami użycia
<b>Przebieg podstawowy</b>	1. Wyświetla się konto z trzema polami (rodzaj sprawy/temat/opis) 2. Użytkownik wybiera/wypełnia pola i zatwierdza przyciskiem "Wyślij"
<b>Sytuacje wyjątkowe</b>	Wysyłanie spamu przez użytkowników
<b>Warunki końcowe</b>	Użytkownik wraca do menu głównego
<b>Komentarz</b>	-

## 4 Model obiektowy systemu

### 4.1. Diagram(y) klas

KLASA	Wykresy Funkcji
OPIS KLASY	<i>Rysowanie funkcji (1,2) zmiennych</i>
ATRYBUTY	<ul style="list-style-type: none"> <li>• nazwa : Typ – {Opis}</li> <li>•</li> </ul>
METODY	<ul style="list-style-type: none"> <li>• nazwa () – {Opis}</li> <li>•</li> </ul>
ZWIĄZKI	-

KLASA	Prognozowanie Wykresów
OPIS KLASY	<i>Prognoza wykresu funkcji na podstawie podanych przez użytkownika punktów.</i>
ATRYBUTY	<ul style="list-style-type: none"> <li>• nazwa : Typ – {Opis}</li> <li>•</li> </ul>
METODY	<ul style="list-style-type: none"> <li>• nazwa () – {Opis}</li> <li>•</li> </ul>
ZWIĄZKI	-

KLASA	Ekstrema Funkcji
OPIS KLASY	<i>Oblicza ekstremum funkcji za pomocą podanych algorytmów</i>
ATRYBUTY	<ul style="list-style-type: none"> <li>• nazwa : Typ – {Opis}</li> <li>•</li> </ul>
METODY	<ul style="list-style-type: none"> <li>• nazwa () – {Opis}</li> <li>•</li> </ul>
ZWIĄZKI	- Algorytm XYZ

KLASA	Algorytm XYZ
OPIS KLASY	<i>Zaimplementowany algorytm XYZ</i>
ATRYBUTY	<ul style="list-style-type: none"> <li>• nazwa : Typ – {Opis}</li> <li>•</li> </ul>
METODY	<ul style="list-style-type: none"> <li>• nazwa () – {Opis}</li> <li>•</li> </ul>
ZWIĄZKI	-

KLASA	ALL Algorytm XYZ
OPIS KLASY	<i>Oblicza i porównuje kolejno ekstrema funkcji zaimplementowanych algorytmów XYZ</i>
ATRYBUTY	<ul style="list-style-type: none"> <li>• nazwa : Typ – {Opis}</li> <li>•</li> </ul>
METODY	<ul style="list-style-type: none"> <li>• nazwa () – {Opis}</li> <li>•</li> </ul>
ZWIĄZKI	<ul style="list-style-type: none"> <li>- Algorytm XYZ</li> <li>- Ekstrema Funkcji</li> </ul>

### Diagram maszyny stanowej dla każdego opisu klasy

{W następnym etapie projektu należy tu wkleić rysunek – diagramy maszyny stanowej dla opisu każdej klasy (pod każdym diagramem klas odpowiedni diagram maszyny stanowej)}

### 4.2. Diagram(y) czynności

{W następnym etapie projektu należy tu wkleić rysunek – diagram(-y) czynności}

## 5 Wymagania jakościowe i eksploatacyjne

*Ochrona kodu systemu operacyjnego - System powinien być zabezpieczony przed nieuprawnionym odczytem i modyfikowaniem danych oraz kodu.*

*Ograniczenie odpowiedzialności - Twórcy nie odpowiadają za niezgodne z prawem lub nieprawidłowe wykorzystanie i modyfikowanie stworzonego oprogramowania.*

*Płynność i stabilność działania aplikacji - Czas wykonywania programu oraz jego użytkowanie powinno być płynne i jak najszybsze. Alokacja zasobów powinna być ograniczona do minimum, a wykorzystane zasoby powinny być zwolnione niezwłocznie po zamknięciu aplikacji.*

*Obsługa błędów - Użytkownik jest informowany o wystąpieniu sytuacji krytycznej, zgłoszeniu wyjątku lub dostarczeniu niepoprawnych danych*