

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

<b>Projekt Systemu (PS)</b>	<b>Nr wersji:</b> 1.0.1
<b>Odpowiedzialny za dokument:</b> <i>Paul Bugaj, Krzysztof Bądkowski, Michał Fladziński</i>	<b>Data pierwszego sporządzenia:</b> 06-06-2018
	<b>Data ostatniej aktualizacji:</b> 07-06-2018

## Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.0.0	wstępna wersja	całość	Maciej Dąbrowski	06-06-2018
1.0.1	poprawka wszystkiego	całość	Paul Bugaj, Krzysztof Bądkowski, Michał Fladziński	13-06-2018

## 1 Cel i Zakres Dokumentu

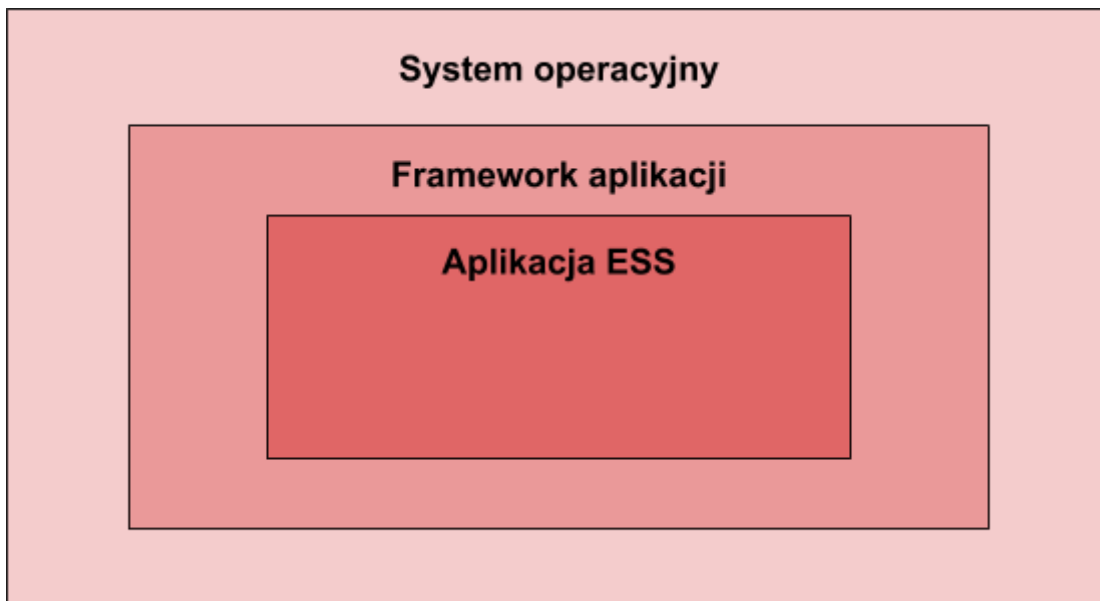
Celem dokumentu jest określenie w sposób ogólny projektowanego systemu.

Sprecyzujemy zakres oraz wstępną architekturę interfejsu użytkownika.

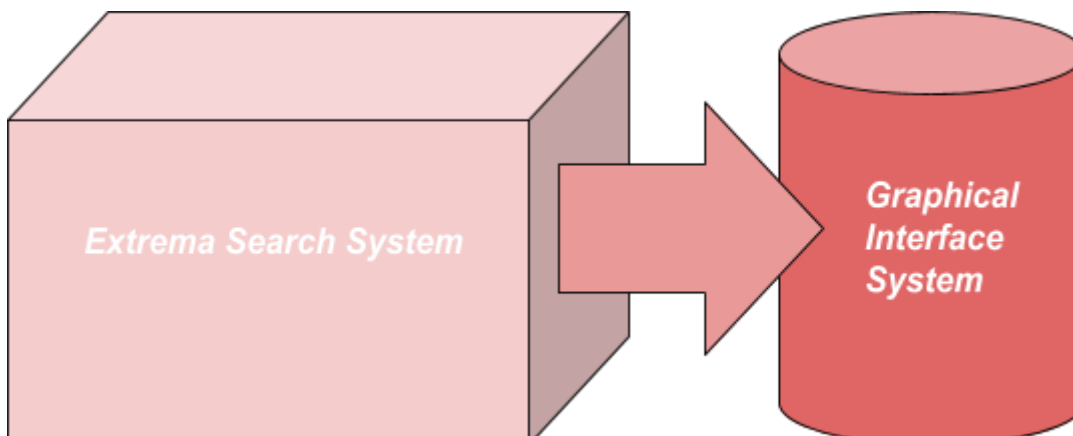
Określimy potrzebny czas w oparciu o dostępne zasoby i stworzymy kosztorys, który zostanie oparty o model COCOMO.

## 2 Dekompozycja systemu

Model warstwowy podsystemów projektu:

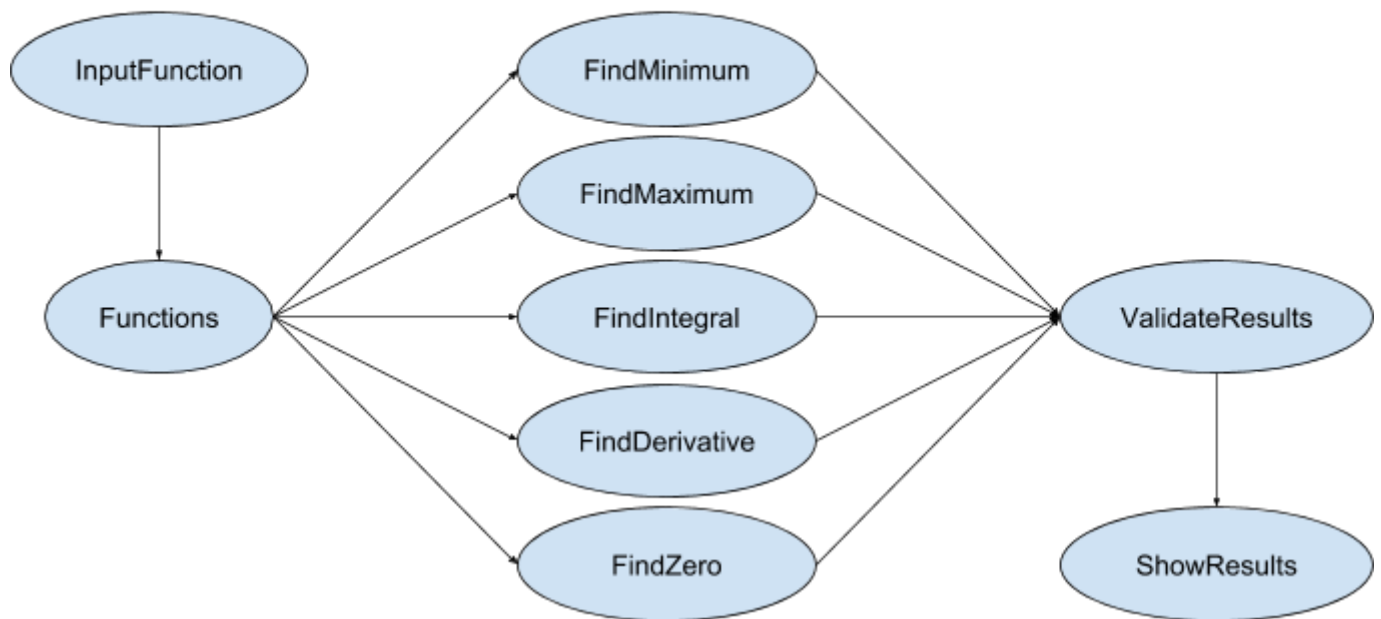


Model przepływowy podsystemów projektu:



## 3.1. Back End (klasy)

## Diagram Klas



<b>KLASA</b>	<b>InputFunction</b>
<b>OPIS KLASY</b>	Klasa zawierająca metody pozwalające dodać nową funkcję zdefiniowaną przez użytkownika do zbioru funkcji (klasa Functions)
<b>ATRYBUTY</b>	- Brak -
<b>METODY</b>	<ul style="list-style-type: none"> <li>• <b>Save()</b> – Zapisanie funkcji podanej przez użytkownika do zbioru w programie</li> </ul>
<b>ZWIĄZKI</b>	-Związek z klasą Functions

<b>KLASA</b>	<b>Functions</b>
<b>OPIS KLASY</b>	Klasa zawierająca zbiór funkcji
<b>ATRYBUTY</b>	<ul style="list-style-type: none"> <li>• <b>Michalewicz</b> - funkcja Michalewicza</li> <li>• <b>Rosenbrock</b> - funkcja Rosenbrocka</li> <li>• <b>Rastrigin</b> - funkcja Rastrigina</li> <li>• <b>Levy</b> - funkcja Levy'ego</li> </ul>
<b>METODY</b>	<ul style="list-style-type: none"> <li>• <b>Delete(function)</b> - Usuwanie danej funkcji ze zbioru</li> <li>• <b>Optimize(function)</b> - Optymalizowanie danej funkcji ze zbioru</li> </ul>
<b>ZWIĄZKI</b>	- Klasa InputFunction

<b>KLASA</b>	<b>FindMinimum</b>
<b>OPIS KLASY</b>	Klasa obliczająca minimum danej funkcji
<b>ATRYBUTY</b>	<ul style="list-style-type: none"> <li>• - Brak -</li> </ul>
<b>METODY</b>	<ul style="list-style-type: none"> <li>• <b>Calculate (function)</b> – Obliczanie minimum funkcji przesłanej jako argument</li> </ul>
<b>ZWIĄZKI</b>	- Związek z klasą Functions

<b>KLASA</b>	<b>FindMaximum</b>
<b>OPIS KLASY</b>	Klasa obliczająca maksimum danej funkcji

ATRYBUTY	<ul style="list-style-type: none"> <li>- Brak -</li> </ul>
METODY	<ul style="list-style-type: none"> <li><b>Calculate (function)</b> – Obliczanie maksimum funkcji przesłanej jako argument</li> </ul>
ZWIĄZKI	- Związek z klasą Functions

KLASA	<b>FindZero</b>
OPIS KLASY	Klasa obliczająca miejsca zerowe danej funkcji
ATRYBUTY	<ul style="list-style-type: none"> <li><b>Calculate (function)</b> – Obliczanie miejsca zerowe przesłanej jako argument</li> </ul>
METODY	<ul style="list-style-type: none"> <li><b>Calculate ()</b> – {Opis}</li> </ul>
ZWIĄZKI	-Związek z klasą Functions

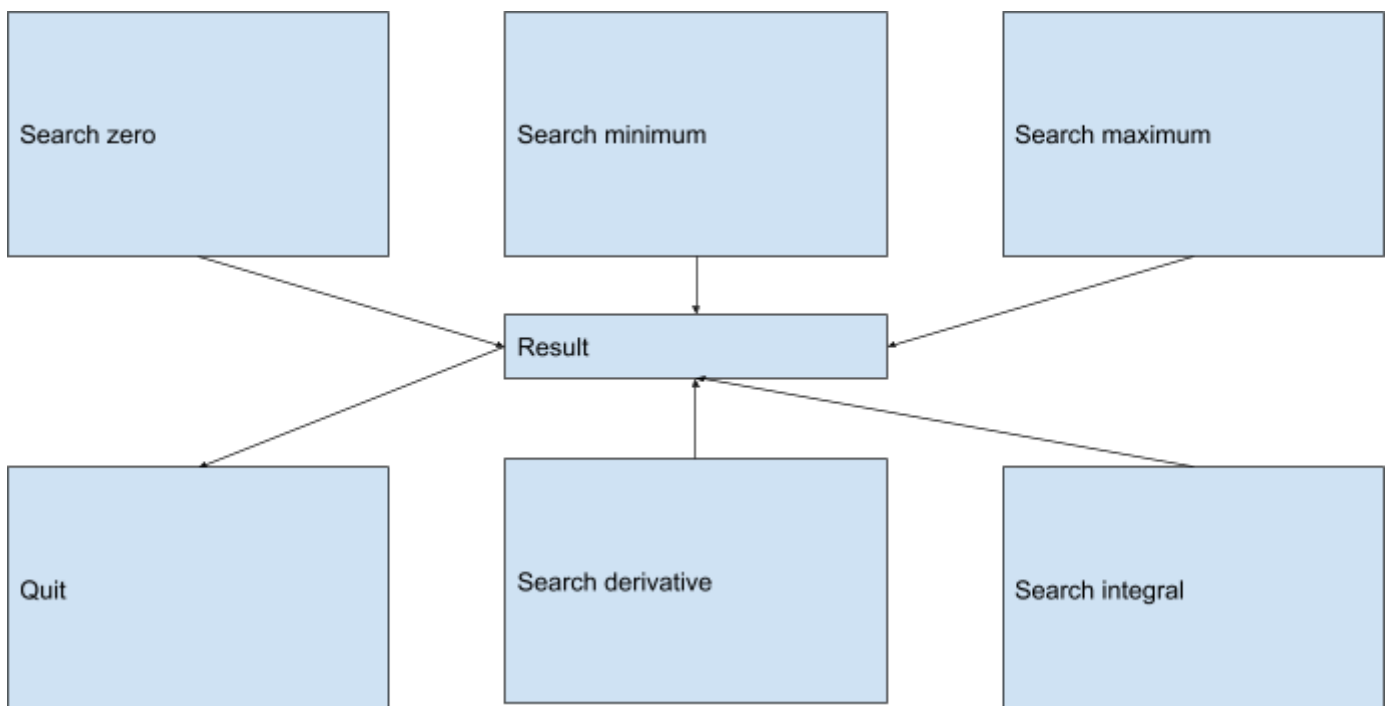
KLASA	<b>FindDerivative</b>
OPIS KLASY	Klasa obliczająca pochodnej danej funkcji
ATRYBUTY	<ul style="list-style-type: none"> <li>- Brak -</li> </ul>
METODY	<ul style="list-style-type: none"> <li><b>Calculate (function)</b> – Obliczanie pochodnej funkcji przesłanej jako argument</li> </ul>
ZWIĄZKI	- Związek z klasą Functions

KLASA	<b>FindIntegral</b>
OPIS KLASY	Klasa obliczająca całki danej funkcji
ATRYBUTY	<ul style="list-style-type: none"> <li>- Brak -</li> </ul>
METODY	<ul style="list-style-type: none"> <li><b>Calculate (function)</b> – Obliczanie całki funkcji przesłanej jako argument</li> </ul>
ZWIĄZKI	- Związek z klasą Functions

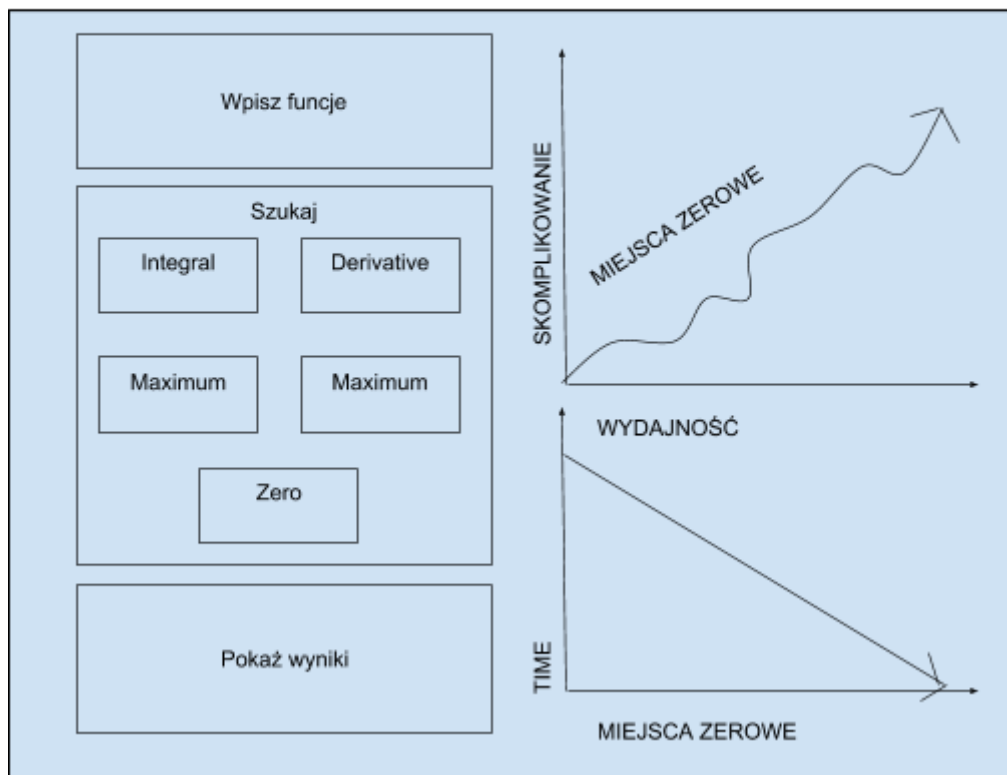
KLASA	<b>ShowResults</b>
OPIS KLASY	Klasa wypisująca na ekran wyniki w postaci liczbowej i/lub wykresu po uprzednim sprawdzeniu poprawności wyniku
ATRYBUTY	<ul style="list-style-type: none"> <li>- Brak -</li> </ul>
METODY	<ul style="list-style-type: none"> <li><b>Show (function)</b> – Wypisuje wyniki</li> <li><b>Plot (function)</b> - Pokazuje wykres funkcji</li> </ul>
ZWIĄZKI	- Związek z klasą ValidateResult

KLASA	<b>ValidateResult</b>
OPIS KLASY	Klasa sprawdzająca czy dany wynik, który chcemy wyświetlić jest zgodny z oczekiwanym rezultatem
ATRYBUTY	<ul style="list-style-type: none"> <li>- Brak -</li> </ul>
METODY	<ul style="list-style-type: none"> <li><b>Validate (result)</b> – sprawdzanie poprawności wyniku</li> </ul>
ZWIĄZKI	- Związek ze wszystkimi klasami oprócz klasy InputFunction

### 3.2. Interface użytkownika (możliwe wybory)



## Interface użytkownika



## 4 COCOMO

Software project	$a_b$	$b_b$	$c_b$	$d_b$
Organic	1.78	1.01	2.5	0.38

KLoC- 2.2 (Czyli 2,200 linijek)

Effort Applied ( $E$ ) =  $a_b(KLOC)^{b_b}$  [man-months]

Development Time ( $D$ ) =  $c_b(Effort Applied)^{d_b}$  [months]

People required ( $P$ ) = Effort Applied / Development Time [count]

$$E = 1.78(2.2)^{1.01} \Rightarrow E = 3.95$$

$$D = 2.5(3.95)^{0.38} \Rightarrow D = 4.21$$

$$P = \frac{3.95}{4.21} \Rightarrow P = .94$$

## 5 Podsumowanie

Realizacja projektu nie powinna być problemem dla wyznaczonej grupy w wyznaczonych terminach z możliwością ciągłego rozwoju projektu. Przewidywane jest, że każdemu członkowi grupy uda się wywiązać ze swoich obowiązków w wyznaczonym czasie co spowoduje powstanie pełnoprawnej aplikacji w wyznaczonym terminie.