

Jacek Pietrus

Laboratorium Inteligentnych Systemów Informatycznych, OPI

# Zastosowanie systemu siatki w aplikacji POL-on

Warszawa, 2012

Interfejs

Interfejs – zespół środków zapewniających dopasowanie oraz ustalających funkcjonalne relacje pomiędzy odrębnymi częściami systemu w celu wymiany informacji między nimi.

Interfejs użytkownika (UI) – sposób, w jaki dokonuje się interakcja pomiędzy człowiekiem a urządzeniem, mająca na celu sterowanie urządzeniem i otrzymywanie informacji zwrotnej o wyniku sterowania i stanie urządzenia.

W interakcji człowieka z komputerem interfejs odnosi się do systemu operacyjnego lub aplikacji, za pomocą których użytkownik może wykonywać zadania poprzez środowisko tekstowe lub graficzne (GUI).

# Projektowanie UI

Projektowanie interfejsu użytkownika to proces twórczy, który może składać się z następujących faz:

- zbieranie wymagań funkcjonalnych,
- analiza oczekiwań potencjalnych użytkowników,
- tworzenie architektury informacji,
- prototypowanie,
- testowanie użyteczności,
- tworzenie GUI.

Prawidłowo zaprojektowany interfejs powinien:

- być łatwy do nauczenia się przez nowego użytkownika i umożliwiać łatwe i szybkie wykonywanie typowych zadań,
- prezentować elementy funkcjonalne i dane w sposób zorganizowany i systematyczny,
- w czytelny sposób prezentować wszystkie dane i opcje potrzebne do wykonania konkretnego zadania,
- informować użytkownika o zmianach stanu, warunkach koniecznych do wykonania zadania, błędach,
- redukować możliwość popełniania błędów i być odporny na złe dane,
- być regularny i konsekwentny, powinien umożliwiać wielokrotne wykorzystywanie tych samych komponentów.

# Projektowanie graficzne

Projektowanie graficzne (graphic design) polega na dobieraniu i aranżowaniu elementów wizualnych takich jak tekst, zdjęcia, ilustracje, symbole, kolory.

Projektowanie graficzne nazywane jest również komunikacją wizualną (visual communication) lub projektowaniem zorientowanym na komunikację (communication oriented design).

Cele komunikacji wizualnej:

- stworzenie czytelnej, zrozumiałej informacji,
- przekazanie jej do odbiorcy,
- wywołanie pożądanego zachowania u odbiorcy po tym jak odbierze i zinterpretuje informację.



Projektowanie graficzne nie jest sztuką (w powszechnym rozumieniu słowa sztuka).

Sztuka wywołuje przede wszystkim wrażenia estetyczne oddziałuje na emocje. Oceniana jest poprzez oryginalność, maestrię artysty, dobór środków wyrazu.

Projekt wizualny ma przedstawić najlepszą graficzną reprezentację dla danej informacji i zrobić to w sposób efektywny i estetyczny.

Ze względu na funkcjonalny charakter, projekt graficzny może znacząco poprawić użyteczność interfejsu poprzez zwiększenie efektywności komunikacji z użytkownikiem.

Wybrane reguły w projektowaniu graficznym interfejsu:

jedność, subtelność, stosowność, przejrzystość, harmonia, umiar, grupowanie, hierarchia, relacje, równowaga, elastyczność, regularność, integralność, kompleksowość, stosowność

Wybrane typowe błędy występujące w interfejsie graficznym:

wizualny szum, przypadkowy układ, brak symetrii, błędna struktura, częściowe wyrównanie, nadmierne zagęszczenie, losowe lub niedopasowane rozmiary, niekonsekwentna prezentacja, różnorodność stylów

Przedstawione powyżej błędy pojawiają się w aplikacji.

# Badania użyteczności standardów

Throstrup, Nielsen (1991)

## Test laboratoryjny

26 studentów po kursie projektowania interfejsu miało za zadanie stworzyć dwustronicowy interfejs hipotetycznej firmy, na podstawie opisanego standardu.

Średni poziom zgodności wykonanej pracy z opisem wyniósł 71%.

Test przeprowadzony na grupie deweloperów duńskiej firmy programistycznej.

15 deweloperom przedstawiono dokument opisujący standardy. Na podstawie zbioru pytań zebrano opinie na temat standardów.

- 73% w pełni zgodziło się z tym, że standardy są potrzebne.
- 53% uznało, że są trudne do zapamiętania.

Tym samym osobom przedstawiono ekrany z interfejsem w 12 miejscach odbiegającym od standardu. Ankietowani wskazali średnio jedynie 4 różnice.

## Wnioski z badań

Standardy interfejsu użytkownika mogą być trudne do zastosowania przez developerów.

Aby zwiększyć użyteczność standardów zarekomendowano:

- użycie narzędzi deweloperskich lub szablonów, które będą wspierać implementację interfejsów,
- dołączenie większej ilości konkretnych przykładów poprawnego interfejsu,
- wymaganie 100% zgodności ze standardem,
- zgodność ze wcześniejszymi standardami lub dokładne oznaczenie i opisanie zmian.

System siatki

## Modernizm

- ograniczenie ornamentyki,
- prostota formy,
- jedność między sztuką a techniką,
- podejście funkcjonalne.

## Jan Tschichold

### “Nowa Typografia” (1928)

Istotą Nowej Typografii jest klarowność. Sytuuje ją to w świadomej opozycji do dawnej typografii, której celem było „piękno” i której klarowność nie osiąga wymaganego dziś maksimum.



# Szwajcarski Design / Międzynarodowy Styl Typograficzny

Główni przedstawiciele: Emil Ruder, Josef Muller-Brockmann.

Szwajcaria, lata 50-te XX wieku

Cechy stylu:

- rozumowe planowanie,
- podkreślenie czystości, schludności,
- pomniejszenie roli indywidualizmu i ekspresywności,
- czytelność, harmonia, prostota,
- użycie czcionek bezszeryfowych,
- preferowanie fotografii zamiast ilustracji,
- komunikacja z masowym odbiorcą.



# Siatka

Josef Muller-Brockmann – “Raster Systeme Fur Die Visuele Gestaltung” / “Grid Systems in Graphic Design” (1961)

Uniwersalny system siatki – dwuwymiarowa struktura składająca się z serii przecinających się pionowych i poziomych linii.

Cechy systemu:

- wsparcie dla projektantów,
- logiczna i konsekwentna organizacja informacji na stronie,
- przewidywalność tworzonych struktur,
- możliwość kreowania szablonów / modułów,
- różnorodność zastosowań (książka, plakat, logo, strona www).

1980-1981

1980-1981

1980-1981

1980-1981



1980-1981

1980-1981

1980-1981

1980-1981

1980-1981

1980-1981



1980-1981

1980-1981

# 960 Grid System

960.gs to zestaw stylów dających deweloperowi możliwość łatwego zaadaptowania na stronie www systemu siatki.

Cechy systemu:

- darmowy, można go legalnie modyfikować dla własnych potrzeb,
- przetestowany w wiodących przeglądarkach,
- dzięki dostępnemu generatorowi pozwala stworzyć siatkę zgodną z indywidualnymi wymaganiami,
- łatwy w użyciu, wymaga jedynie dołączenia pliku ze stylem do strony i zastosowania się do opisanej konwencji,
- posiada plik resetujący domyślne style i wsparcie typografii,
- szeroko stosowany,
- paczka zawiera szablony dla programów graficznych.

# Przykładowy kod strony www z użyciem 960.gs:

```
<h1><a href="http://960.gs/">960 Grid System</a></h1>
```

```
<div class="container_12">  
  <h2>12 Column Grid</h2>  
  <div class="grid_12">  
    <p>940</p>  
  </div>  
  <!-- end .grid_12 -->  
  <div class="clear"></div>  
  <div class="grid_3">  
    <p>220</p>  
  </div>  
  <!-- end .grid_3 -->  
  <div class="grid_9">  
    <p>700</p>  
  </div>  
  <!-- end .grid_9 -->  
</div>
```

# 960 Grid System w aplikacji POL-on



Aby spełnić wymagania stawiane przed interfejsem aplikacji wygenerowane zostały dwie siatki, które następnie poddałem modyfikacjom.

Stworzone zostały dwa pliki css:

- 960gs\_cw30gw10cc24\_fixed.css

Plik ze stylami do wykorzystania w oknach dialogowych, wyświetlający siatkę o stałej szerokości. Siatka składa się z 24 kolumn, które zajmują szerokość 960px. Każda kolumna ma szerokość 30px z marginesem 10px;

- 960gs\_cw30gw10cc24\_fluid.css

Plik ze stylami do wykorzystania w filtrach, wyświetlający siatkę o szerokości dopasowującej się do rozmiaru strony.

Aby móc używać siatkę w sposób zbliżony do stosowania komponentu `panelGrid` z JSF, zostały stworzone komponenty:

- `grid:container` – kontener, w którym umieszczana jest siatka,
- `grid:item` – pojedynczy element, pozycjonowany na siatce.

Definicje komponentów znajdują się w plikach:

- `HtmlGridContainer.java`
- `GridContainerRenderer.java`
- `HtmlGridItem.java`
- `GridItemRenderer.java`

## Użycie komponentów w pliku xhtml.

```
<grid:container id="cont" size="16" type="fixed" showGrid="false">
  <!-- Tryb nadania -->
  <gridHelper:label
    for="trybprzyznania"
    value="#{resourceBundle['trybnadania.label']}:"
    size="4" align="right" required="true" first="true" />
  <grid:item size="6">
    <h:selectOneMenu id="trybprzyznania"
      value="#{form.degreeGrantingMode}" rendered="#{newDegree}">
      <opi:ajax flowAction="checkDegreeAwardType" event="change"
        </h:selectOneMenu>
    </grid:item>
  <gridHelper:label
    size="4" align="right" first="true" for="stopien"
    required="true" value="Stopień naukowy ujednolicony:" />
  <grid:item id="unifiedDegree" size="6">
    <h:inputText value="#{form.unifiedDegree}" disabled="true"
      converter="displayConverter" />
  </grid:item>
</grid:container>
```

```
<grid:container size="24" type="fluid" showGrid="false">
  <grid:item size="6"
    rendered="{not empty form.availableReportTypes}">
    <h:outputLabel for="reportType" value="Typ raportu:" />
    <h:selectOneMenu id="reportType" value="{form.reportType}">
    <f:selectItem itemLabel="-- wybierz --" itemValue="" />
    <f:selectItems value="{form.availableReportTypes}"/>
    </h:selectOneMenu>
  </grid:item>
  <grid:item size="6"
    rendered="{not empty form.availableReportKinds}">
    <h:outputLabel
      rendered="{form.reportType != null}"
      for="reportKind" value="Forma raportu:" />
    <h:selectOneMenu id="reportKind" value="{form.reportKind}"
      converter="enumConverter">
    <f:selectItem itemLabel="-- wybierz --" itemValue="" />
    <f:selectItems value="{form.availableReportKinds}"/>
    </h:selectOneMenu>
  </grid:item>
</grid:container>
```

## Dostępne szerokości kontenera:

1 col: 40px	9 col: 360px	17 col: 680px
2 col: 80px	10 col: 400px	18 col: 720px
3 col: 120px	11 col: 440px	19 col: 760px
4 col: 160px	12 col: 480px	20 col: 800px
5 col: 200px	13 col: 520px	21 col: 840px
6 col: 240px	14 col: 560px	22 col: 880px
7 col: 280px	15 col: 600px	23 col: 920px
8 col: 320px	16 col: 640px	24 col: 960px

## Proponowane szerokości okien

XS - 8 col 320px
S - 12 col 480px
M - 16 col 640px
L - 20 col 800px
XL - 24 col 960px

## Opis atrybutów:

### Kontener:

- size – liczba od 1 do 24, ilość kolumn jakie zawiera kontener
- type – fixed/fluid, sposób renderowania (stała szerokość lub elastyczny)
- showGrid – true/false, włączenie/wyłączenie podglądu siatki

### Komórka

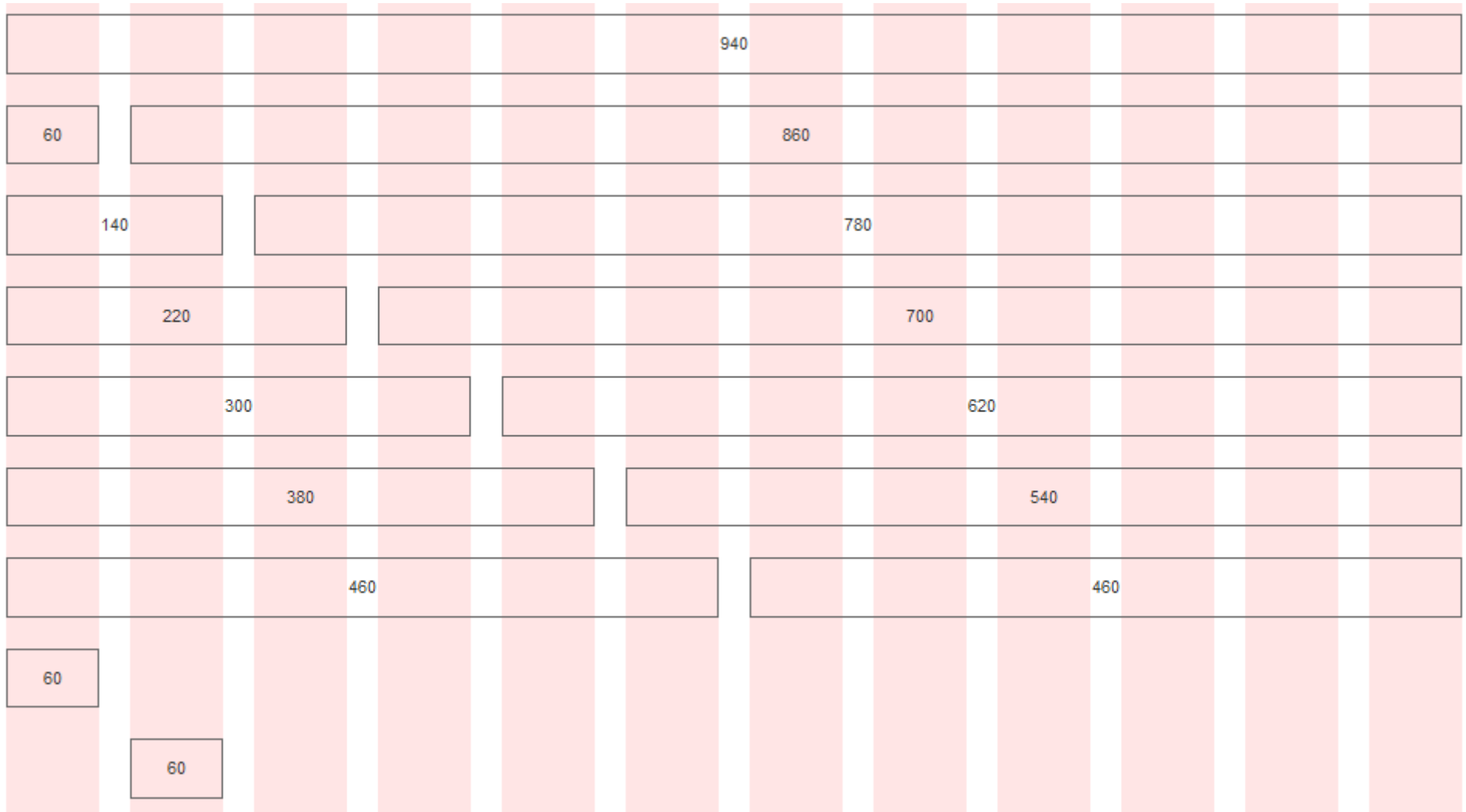
- size – liczba od 1 do 24, ilość kolumn jakie zajmie komórka
- first – true/false, jeśli true, komórka wyświetli się jako pierwsza w kolejnej linii
- align – left/right/center, wyrównanie tekstu w komórce
- required – true/false, dodaje klasę required (gwiazdka przy label)

Do pliku xhtml należy dodać odpowiednią deklarację namespace:

- `xmlns:grid="http://opi.org.pl/jsf/grid"`
- `xmlns:gridHelper="http://java.sun.com/jsf/composite/components/grid"`

Ekraný





• Rodzaj zatrudnienia:

• Miejsce pracy  Podstawowe miejsce pracy  Dodatkowe miejsce pracy


• Instytucja:


• Forma zatrudnienia:

• Rodzaj umowy:

• Wymiar czasu pracy:  Część:  Całość:

• Nauczyciel akademicki - typ:


• Data nawiązania stosunku pracy:  

Data rozwiązania /  
wygaśnięcia stosunku  
pracy:  

• Stanowisko:

• Data objęcia stanowiska:  

Zgoda rektora na  
dodatkowe zatrudnienie:

Data wydania zgody:  

• Rodzaj zatrudnienia:	Nauczyciel akademicki	
• Miejsce pracy	<input checked="" type="radio"/> Podstawowe miejsce pracy <input type="radio"/> Dodatkowe miejsce pracy	
• Instytucja:	Wydział Fizyki i Informatyki Stosowanej	
• Forma zatrudnienia:	Mianowanie	
• Rodzaj umowy:	Czas nieokreślony	
• Wymiar czasu pracy:	Pełny Część: 1 Całość: 1	
• Nauczyciel akademicki - typ:	Naukowo-dydaktyczny	
• Data nawiązania stosunku pracy:	1999-07-01	
• Data rozwiązania / wygaśnięcia stosunku pracy:		
• Stanowisko:	Adiunkt	
• Data objęcia stanowiska:	2000-07-01	
Zgoda rektora na dodatkowe zatrudnienie:	<input type="checkbox"/>	
• Data wydania zgody:		

• Tryb nadania:


• Stopień naukowy:


• Stopień naukowy ujednolicony:



• Dziedzina, dyscyplina:

Specjalność:

• Instytucja nadająca:  Instytucja istniejąca  Instytucja historyczna

Data wszczęcia przewodu:  

• Data uzyskania stopnia:  



• Tryb nadania:	<input type="text" value="Stopień krajowy"/>
• Stopień naukowy:	<input type="text" value="-- Wybierz --"/>
• Stopień naukowy ujednolicony:	<input type="text"/>
• Dziedzina, dyscyplina:	<input type="button" value="Dodaj klasyfikacje"/> Pusta lista klasyfikacji
Specjalność:	<input type="text"/>
• Instytucja nadająca:	<input checked="" type="radio"/> Instytucja istniejąca <input type="radio"/> Instytucja historyczna <input type="button" value="Wybierz"/>
Data wszczęcia przewodu:	<input type="text"/> 
• Data uzyskania stopnia:	<input type="text"/> 

Tryb nadania: Stopień krajowy

Stopień naukowy: kwalifikacje II stopnia

Stopień naukowy ujednolicony: doktor habilitowany


Dziedzina, dyscyplina: Dodaj klasyfikacje


Klasyfikacja	Dziedzina	Dyscyplina	
Dyscypliny CKSST	Dziedzina nauk biologicznych	Biochemia	 Usun
Dyscypliny historyczne	Automatyka i robotyka	Nauki techniczne	 Usun

Specjalność:

Instytucja nadająca:  Instytucja istniejąca  Instytucja historyczna

Wybrano: Instytut Ekologii Terenów Uprzemysłowionych

Data wszczęcia przewodu:  

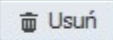
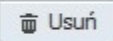
Data uzyskania stopnia:  

• Tryb nadania: Stopień krajowy

• Stopień naukowy: kwalifikacje II stopnia

• Stopień naukowy ujednolicony: doktor habilitowany

• Dziedzina, dyscyplina: Dodaj klasyfikacje

Klasyfikacja	Dziedzina	Dyscyplina	
Dyscypliny CKSST	Dziedzina nauk biologicznych	Biochemia	
Dyscypliny historyczne	Automatyka i robotyka	Nauki techniczne	

Specjalność:

• Instytucja nadająca:  Instytucja istniejąca  Instytucja historyczna

Wybrano: Instytut Ekologii Terenów Uprzemysłowionych

Data wszczęcia przewodu:

• Data uzyskania stopnia:

```
[-] <div id="popupWindow1_subview:form:cont" class="gs c_16 fixed clearfix showgrid">
  [+ <div class="grid_4 first right label">
  [+ <div class="grid_4">
  [+ <div class="grid_4 right label">
  [+ <div class="grid_4">
  [+ <div class="grid_4 first right label">
  [+ <div id="popupWindow1_subview:form:unifiedDegree" class="grid_12">
  [+ <div class="grid_4 first right label">
  [+ <div class="grid_3">
    <div class="grid_4 first"></div>
  [+ <div class="grid_12">
  [+ <div class="grid_4 first right label">
  [+ <div id="popupWindow1_subview:form:specjalnosci" class="grid_12">
  [+ <div class="grid_4 right label">
  [+ <span id="popupWindow1_subview:form:j_idt56">
  [+ <div class="grid_4 first right label">
  [+ <div class="grid_4">
  [+ <div class="grid_4 right label">
  [+ <div class="grid_4">
```



• Tryb nadania:

• Stopień naukowy:


• Stopień naukowy ujednolicony:


• Dziedzina, dyscyplina:

Klasyfikacja	Dziedzina	Dyscyplina	
Dyscypliny CKSST	Dziedzina nauk biologicznych	Biochemia	<input type="button" value="Usuń"/>

Specjalność:

• Instytucja nadająca:  Instytucja istniejąca  Instytucja historyczna

Data wszczęcia przewodu:  

• Data uzyskania stopnia:  

```
<div id="popupWindow1_subview:form:cont" class="gs c_16 fixed clearfix showgrid">
  <div class="grid_4 first right label">
  <div class="grid_6">
  <div class="grid_4 first right label">
  <div class="grid_6">
  <div class="grid_4 first right label">
  <div id="popupWindow1_subview:form:unifiedDegree" class="grid_6">
  <div class="grid_4 first right label">
  <div class="grid_3">
    <div class="grid_4 first"></div>
  <div class="grid_12">
  <div class="grid_4 first right label">
  <div id="popupWindow1_subview:form:specjalnosci" class="grid_12">
  <div class="grid_4 right label">
  <span id="popupWindow1_subview:form:j_idt56">
  <div class="grid_4 first right label">
  <div class="grid_4">
  <div class="grid_4 first right label">
```

**Podsumowanie**

Zastosowanie systemu siatki w aplikacji POL-on daje następujące korzyści:

- nadaje wizualny porządek formularzom w oknach dialogowych,
- standaryzuje rozmiary i położenie komponentów,
- wprowadza możliwość unormowania rozmiarów okien,
- nowy sposób tworzenia układu komponentów w formularzu jest łatwy do nauczenia się,
- efekt pracy jest przewidywalny,
- istnieje możliwość sprawdzania na bieżąco tworzonego układu poprzez wyświetlenie podglądu siatki,
- jest to rozwiązanie modułarne, możliwe do zastosowania w wielu miejscach w aplikacji,
- modyfikacje wymagają zmian w css i komponentach.

## Trudności, ograniczenia, problemy:

- wymaga nauki, przyzwyczajenia
- aby nadać spójny wygląd całej aplikacji należałoby zmienić już gotowe okna,
- tworząc kontener w oknie popup trzeba ręcznie uaktualnić szerokość okna (zawsze +40px),
- testowany przez małą liczbę developerów.

Dziękuję