

Luna Designer

Zestawienie funkcji

Wersja 3.2

Spis treści

1. CEL DOKUMENTU	BŁĄD! NIE ZDEFINIOWANO ZAKŁADKI.
2. SŁOWNIK POJĘĆ	4
3. FUNKCJONALNOŚCI REALIZOWANE W RAMACH MODUŁU LUNA DESIGNER	5

1. Informacje ogólne

LUNA Designer jest narzędziem CASE umożliwiającym projektowanie i rozbudowę systemu przy użyciu notacji zbliżonej do UML. Narzędzie umożliwia wizualne tworzenie i modyfikowanie formularzy oraz wiązanie ich z modelem danych i przypadkami użycia. Pozwala na modelowanie dynamicznego modelu, przypadków użycia, formatek GUI, reguł, menu. Narzędzie na żądanie umożliwia wygenerowanie działającego kodu java i jego uruchomienie, pozwala to na sprawne tworzenie w pełni funkcjonalnych prototypów.

2. Słownik pojęć

Pojęcie	Wyjaśnienie
Czynność	Krok wykonywany w ramach akcji będącej elementem diagramu aktywności dla przypadku użycia. Czynności mogą być różnych typów.
Diagram aktywności	Diagram aktywności przypadku użycia. Określa sposób działania przypadku użycia poprzez zdefiniowanie węzłów diagramu i przejść między nimi.
Formatka	(Inne określenia: ekran, formularz). Interfejs programu z użytkownikiem). Służy między innymi do prezentacji, wprowadzania danych.
Kontrolka	Element umieszczony na formatce, związany z prezentacją lub wprowadzaniem danych lub uruchamianiem funkcjonalności. Przykładami typów kontrolki są: pole tekstowe, pole numeryczne, lista wyboru, zakładka, tabela, przycisk ekranowy
Przypadek użycia	Specyfikacja ciągów akcji i ich wariantów, które system lub inna jednostka może wykonać poprzez interakcję z aktorem tego systemu

3. Funkcjonalności realizowane przez LUNA Designer

L. p.	Nazwa funkcjonalności	Opis funkcjonalności
1.	Zarządzanie klasami	<ul style="list-style-type: none"> • Dodawanie, modyfikacja, usuwanie klasy • Umieszczenie definicji klasy w strukturze modułów i pakietów • Przeglądanie i wyszukiwanie klas zdefiniowanych w Designerze (kryteria wyszukiwania klas: nazwa klasy, nazwa pakietu, nazwa modułu) • Możliwość rozszerzania klas • Możliwość określenia, czy klasa jest persystowalna • Dodawanie komentarza dla klasy
2.	Zarządzanie atrybutami klas	<ul style="list-style-type: none"> • Dodawanie, modyfikacja, usuwanie atrybutów dla klasy • Możliwość oznaczenia atrybutu jako wymagalnego • Możliwość dodania komentarza dla atrybutu klasy
3.	Zarządzanie relacjami między klasami	<ul style="list-style-type: none"> • Graficzny edytor modelu danych – prezentacja interaktywnego diagramu klas w zakresie klas będących w relacji ze wskazaną klasą • Dodawanie, modyfikacja, usuwanie relacji między klasami • Strony w relacji - klasa „source” i klasa „destination” • Właściwości relacji: kierunkowości (relacja dwustronna lub jednostronna) oraz czy relacja jest kompozycją • Określanie krotności dla każdej ze stron relacji • Określanie nazwy roli dla każdego z końców relacji
4.	Zarządzanie formatkami	<ul style="list-style-type: none"> • Dodawanie, modyfikacja, usuwanie formatki • Umieszczenie definicji formatki w strukturze modułów i pakietów • Przeglądanie i wyszukiwanie formatek zdefiniowanych w Designerze (kryteria wyszukiwania: nazwa formatki, nazwa pakietu, nazwa modułu) • Dodawanie komentarza dla formatki • Możliwość wykorzystania jednej formatki w ramach różnych przypadków użycia
5.	Konfigurowanie właściwości formatki	<ul style="list-style-type: none"> • Konfigurowanie domyślnego trybu dostępu dla kontrolek na formatce • Określanie typu formatki (np.: standardowa, modalna, pływająca), jej szerokości, wysokości, położenia oraz innych właściwości
6.	Graficzne projektowanie formatek	<ul style="list-style-type: none"> • Graficzny interfejs użytkownika przy projektowaniu formatki (rozieszczanie kontrolek na formatce, modyfikacja rozmiaru kontrolek za pomocą myszki) • Responsywność projektowanych formatek – podział formatki zgodnie z bootstrap (struktura 12 kolumn) • Wybór różnych typów kontrolek z paska narzędziowego
7.	Konfigurowanie właściwości kontrolek na formatkach	<ul style="list-style-type: none"> • Konfigurowanie właściwości kontrolek (zestaw właściwości kontrolki jest zależny od typu kontrolki) • Sterowanie widocznością i dostępnością kontrolek na formatce

L. p.	Nazwa funkcjonalności	Opis funkcjonalności
		<ul style="list-style-type: none"> • Dla wybranych typów kontrolek możliwość wyzwalania zdarzeń formatkowych, których sposób obsługi jest definiowany w ramach przypadku użycia • Menu kontekstowe dostępne dla każdej z kontrolek osadzonych na formatce z poziomu interfejsu graficznego
8.	Model danych dla formatki	<ul style="list-style-type: none"> • Możliwość zdefiniowania modelu dla formatki (albo poprzez wskazanie na jedną klasę albo poprzez zdefiniowanie wewnętrznego modelu dla formatki, w ramach którego można wskazać wiele elementów z określeniem krotności każdego z nich) • Możliwość powiązania kontrolek na formatce z elementami modelu formatki (dostępny system podpowiedzi klas, atrybutów klas i ról w relacjach)
9.	Warianty dla formatki	<ul style="list-style-type: none"> • Możliwość zdefiniowania wielu wariantów dla jednej formatki, co pozwala na różnicowanie warunków dostępności i widoczności kontrolek w ramach różnych wariantów tej samej formatki, z zachowaniem ich pozostałych właściwości • Sterowanie widocznością i dostępnością kontrolek na formatce w zależności od wariantu formatki i warunków logicznych
10.	Zarządzanie przypadkami użycia	<ul style="list-style-type: none"> • Dodawanie, modyfikacja, usuwanie przypadku użycia • Umieszczenie definicji przypadku użycia w strukturze modułów i pakietów • Przeglądanie i wyszukiwanie przypadków użycia zdefiniowanych w Designerze (kryteria wyszukiwania przypadków użycia: nazwa klasy, nazwa pakietu, nazwa modułu) • Dodawanie komentarza dla przypadku użycia
11.	Parametryzacja przypadku użycia	<ul style="list-style-type: none"> • Możliwość określenia parametrów wejściowych i wyjściowych dla przypadku użycia • Systemowa weryfikacja przekazywania wartości dla parametrów przypadku użycia
12.	Model danych dla przypadku użycia	<ul style="list-style-type: none"> • Możliwość zdefiniowania modelu danych dla przypadku użycia (definiowanie elementów modelu, które swoim typem odwołują się do klas lub do wewnętrznego modelu danych formatki) • Operowanie na modelu danych w ramach przypadku użycia (operacje odczytu, zapisu, usuwania danych) • Przekazywanie modelu danych z przypadku użycia do formatki przy wywołaniu formatki z poziomu przypadku użycia
13.	Diagram aktywności dla przypadku użycia	<ul style="list-style-type: none"> • Działanie przypadku użycia określone przez diagram aktywności • Projektowanie diagramu aktywności dla przypadku użycia z użyciem węzłów typu „Węzeł początkowy”, „Węzeł końcowy”, „Akcja”, „Wywołanie innego przypadku użycia” oraz przejść między węzłami.

L. p.	Nazwa funkcjonalności	Opis funkcjonalności
		<p>Możliwość rozmieszczania elementów na diagramie i konfigurowania właściwości elementów wskazywanych na diagramie.</p> <ul style="list-style-type: none"> • Możliwość kojarzenia przejść między węzłami na diagramie aktywności ze zdarzeniami formatkowymi, pochodzącymi z wywoływanych formatek • Możliwość stosowania wyrażeń logicznych warunkujących wykonanie przejść pomiędzy węzłami przypadku użycia
14.	Parametryzacja akcji	<ul style="list-style-type: none"> • Możliwość określenia wielu parametrów wejściowych dla akcji • Możliwość wywoływania danej akcji w ramach przypadku użycia z różnymi wartościami parametrów • Systemowa weryfikacja przekazywania wartości dla parametrów akcji przy jej wywoływaniu w ramach diagramu aktywności przypadku użycia
15.	Czynności dla akcji	<ul style="list-style-type: none"> • Definiowanie sekwencji czynności wykonywanych w ramach akcji • Wykorzystywanie predefiniowanych typów czynności • Konfigurowanie właściwości czynności (zestaw właściwości czynności jest zależny od typu czynności) • Możliwość określenia warunku logicznego dla każdej czynności, warunkującego jej wykonanie
16.	Zdarzenia obsługiwane w ramach akcji	<ul style="list-style-type: none"> • Definiowanie sposobu obsługi zdarzeń, wywoływanych w ramach formatek
17.	Zarządzanie regułami	<ul style="list-style-type: none"> • Dodawanie, modyfikacja, usuwanie reguły • Umieszczenie definicji reguły w strukturze modułów i pakietów • Przeglądanie i wyszukiwanie reguł (kryteria wyszukiwania reguł: nazwa reguły, nazwa pakietu, nazwa modułu)
18.	Graficzne projektowanie reguł	<ul style="list-style-type: none"> • Projektowanie reguł z użyciem graficznego interfejsu użytkownika • Projektowanie reguł poprzez zestawianie ze sobą w ramach graficznego interfejsu gotowych elementów i ich parametryzowanie • Możliwość deklarowania lokalnych zmiennych i stałych na potrzeby przetwarzania danych w ramach reguły oraz inicjowania ich wartości • Dostępne operatory sterujące przetwarzaniem w pętlach • Dostępne operatory warunkowe • Dostępne operatory odpowiedzialne za pobieranie, filtrowanie i sortowanie danych • Dostępne operatory matematyczne i logiczne
19.	Parametryzacja reguł	<ul style="list-style-type: none"> • Możliwość określenia parametrów wejściowych i wyjściowych dla reguły • Możliwość wywoływania reguły z przekazaniem wartości parametrów

L. p.	Nazwa funkcjonalności	Opis funkcjonalności
20.	Wykorzystanie reguł w projektowaniu formatek	<ul style="list-style-type: none"> Wykorzystywanie reguł do sterowania widocznością i dostępnością kontrolek na formatce
21.	Wykorzystanie reguł w projektowaniu przypadków użycia	<ul style="list-style-type: none"> Wykorzystanie reguł do odczytywania, filtrowania i sortowania danych Wykorzystywanie reguł jako warunków logicznych warunkujących wykonanie czynności w ramach akcji przypadku użycia
22.	Zarządzanie elementami menu	<ul style="list-style-type: none"> Możliwość zaprojektowania wielopoziomowego menu dla projektowanej aplikacji w ramach każdego modułu Element menu może być elementem grupującym elementy niższego poziomu lub elementem wywołującym wskazany przypadek użycia
23.	Tworzenie Przypadków Użycia dla Integracji	<ul style="list-style-type: none"> Wizualne projektowanie przepływu usług sieciowych w tym: obsługa błędów, transformacja komunikatów, routing, filtrowanie, kanonizacja, odwołanie do innych usług wewnętrznych i zewnętrznych
24.	Generowanie kodu Java	<ul style="list-style-type: none"> Generowanie kodu Java reprezentującego przypadek użycia Generowanie kodu reprezentującego formatkę Generowanie kodu reprezentującego model danych wraz z kodem odpowiedzialnym za odczyt i zapis z/do bazy danych
25.	Kompilacja i uruchamianie przypadków użycia	<ul style="list-style-type: none"> Automatyczna kompilacja kodu dla wskazanego przypadku użycia wraz ze wszystkimi jego podprzypadkami oraz innymi artefaktami niezbędnymi do prawidłowego ich działania Możliwość uruchomienie wybranego przypadku użycia wraz z przypadkami użycia zależnymi