*Krzysztof Jackowski*

­­

Narzędzie do analizy działań marketingowych   
w organizacji AIESEC Polska.

1. Wstęp
   1. Analiza działań marketingowych

Jednym z głównych problemów działań marketingowych w ramach organizacji jest mierzenie ich efektywności. Każda inwestycja musi przynieść konkretne wyniki finansowe czyli tak zwany zwrot z inwestycji (ang. „Return Of Investment”). W tradycyjnych kampaniach promocyjnych jest to bardzo skomplikowane, ponieważ nie da się jednoznacznie określić ilości klientów pozyskanych na przykład z plakatów na popularnych bilbordach czy reklam w telewizji. Dzięki rewolucji technologicznej i popularyzacji rozwiązań e-commerce możliwe stało się dokładne zliczanie współczynników potrzebnych do dalszych analiz. Podstawowymi elementami są: ilość odwiedzin, czas spędzony na stronie, ilość sprzedanych produktów itp. Dzięki nim można wyliczyć wskaźniki statystyczne takie jak: ilość powracających klientów czy współczynnik konwersji (ang. „conversion rate”). Dzięki analizie konkretnych współczynników można wyciągnąć wnioski na temat wydatków na obecnie prowadzone kampanie marketingowe, zwrotów z inwestycji marketingowych oraz przygotować zalecenia na przyszłość.

* 1. Ścieżka klienta

Ścieżka klienta jest odwzorowaniem kolejnych interakcji pomiędzy klientem zainteresowanym danym produktem a organizacją go oferującą. Najczęściej definiuje się 3 główne etapy ścieżki:

1. Świadomość (ang. „Awareness”)
2. Namysł (ang. „Consideration”)
3. Decyzja (ang. „Decision”)

W ramach każdego z etapów oferuje się potencjalnemu klientowi różny zestaw treści oraz reklam odpowiadający jego świadomości odnośnie marki oraz intencji zakupowej.

* 1. Współczynniki konwersji wewnątrz ścieżki klienta

Współczynnik konwersji *CR* to wyrażony procentowo stosunek liczby pożądanych akcji podjętych przez użytkownika do liczby sytuacji, w jakich potencjalnie mogą one być zrealizowane (1). Dla przykładu stronę internetową odwiedziło 1000 osób w ciągu jednego dnia. Spośród nich 20 osób podjęło pożądaną przez nas akcję czyli kliknęło w przycisk kupna danego produktu. Pomiędzy tymi dwoma krokami możemy obliczyć współczynnik konwersji w następujący sposób:

W podanym powyżej przykładnie mamy dwa kroki pomiędzy którymi liczymy konwersję. Najważniejszym współczynnikiem konwersji jest moment przejścia z jednego etapu ścieżki klienta do następnego. Pomiędzy każdym ze zdefiniowanych kroków można policzyć współczynnik konwersji i określić, jak interakcje firmy z klientem wpływają na końcowy efekt sprzedażowy.

Ponadto w dobie rozwijających się sieci społecznościowych i różnorodności kanałów promocji w Internecie, pojawia się problem określania konwersji dla danego kanału oraz sposobu promocji w nim. Na przykład na portalu Facebook możliwe jest tworzenie treści promocyjnych w formie bezpłatnych postów lub płatnych reklam. W celu odpowiedniej dystrybucji środków finansowych konieczne jest sprecyzowanie, które kanały są najbardziej efektywne.

* 1. Zakres niniejszej pracy

Celem projektu jest stworzenie narzędzia do analizy w czasie rzeczywistym rezultatów działań marketingowych prowadzonych przez organizację AIESEC Polska. Istotą działania jest zbieranie podstawowych danych marketingowych na temat klientów organizacji w początkowych etapach zaangażowania oraz obliczenia konwersji. Narzędzie ma zbierać, na podstawie specjalnie generowanych linków, informacje o ilości unikalnych odwiedzin strony internetowej danego produktu organizacji oraz ilość osób zainteresowanych kupnem konkretnego produktu, poprzez formularz zgłoszeniowy. Dzięki temu mechanizmowi możliwa będzie prezentacja źródła z którego przyszedł dany klient oraz efektywność danego kanału promocji.

1. Opis organizacji AIESEC
   1. O organizacji

AIESEC to globalna organizacja prowadzona przez młodych ludzi, która od ponad 66 lat realizuje program praktyk i wolontariatów, docierając do 126 krajów i terytoriów. W Polsce jest obecna w 16 największych ośrodkach akademickich, gdzie poprzez realizowane projekty i pracę zespołową umożliwia studentom rozwój osobisty i profesjonalny. Kształtuje wśród nich postawę przedsiębiorczości oraz umiejętności komunikacyjne. Rozwija przyszłych liderów, którzy w odpowiedzialny sposób będą wpływali na swoje lokalne otoczenie. Zaraz po dołączeniu do organizacji student może spróbować swoich siły w pracy z zespołem czy budżetem w ramach wewnętrznych projektów. Dzięki globalnemu zasięgowi organizacji, członkowie budują międzynarodową sieć kontaktów. Wizją organizacji jest “Pokój i pełne wykorzystanie potencjału ludzkiego” (2)

Podstawą działalności AIESEC jest Międzynarodowy Program Praktyk – Global Talents oraz Międzynarodowy Program Wolontariatu – Global Citizen. Oba programy są dwustronne co oznacza, że AIESEC rekrutuje studentów i wspiera ich w procesie wyjazdu za granicę, ale również przyjmuje studentów z innych krajów, którzy przyjeżdżają pracować w polskich firmach oraz placówkach edukacyjnych. Ponadto prowadzone są projekty towarzyszące, mające na celu wykorzystanie zasobów organizacji jak i poprawienie jej stabilności finansowej. Są to:

* AIESEC University – kursy językowe prowadzone przez native speakerów. Głównym celem projektu jest pokazanie, że nauka języka może być bardzo atrakcyjna w szczególności gdy obcujemy bezpośrednio z przedstawicielem danej kultury.
* Global Host – jest to projekt skierowany do studentów ale również rodzin. Polega na możliwości goszczenia wolontariuszy z całego świata podczas ich pobytu w Polsce.

Członkowie organizacji pracują w ramach projektów skupiających się na realizacji poszczególnych programów i starają się dostarczyć jak najlepsze doświadczenie nie tylko praktykantom i wolontariuszom, ale również goszczącym ich firmom i organizacjom. Aby tego dokonać prowadzone są działania rekrutacyjne, sprzedażowe i logistyczne. Organizacja zmaga się z dużym problemem zachowania wiedzy w swoich strukturach, ponieważ członkowie obejmują pozycje w organizacji na krócej niż rok. Z tego powodu nie jest ona w stanie wykorzystywać skomplikowanych systemów do analiz działalności promocyjnych takich jak Google Analytics czy Hubspot i potrzebuje narzędzia, które w prosty sposób przedstawi wyniki podejmowanych działań. Członkowie organizacji w celu ewaluacji swojej pracy oraz określenia najlepszych kampanii marketingowych muszą być świadomi, które kanały generują największą ilość potencjalnych zainteresowanych oraz jaki jest rzeczywisty współczynnik konwersji w stosunku do końcowych uczestników programów.

* 1. Struktura organizacji i jej wpływ na procesy marketingowe

Pełny opis struktury AIESEC Polska został zawarty w Statucie organizacji dostępnym na stronie internetowej <http://aiesec.pl/dokumenty/>. Ze względu na jej zawiłość, na potrzeby tej pracy ograniczę się do przedstawienia wersji skróconej, która pozwoli zrozumieć jak wyglądają procesy związane z promocją produktów organizacji.

Jednostkami terenowymi Stowarzyszenia są Komitety Lokalne działające głównie przy uniwersytetach w największych miastach Polski. Każdy komitet posiada Radę Wykonawczą z Prezydentem Komitetu Lokalnego na czele, która sprawuje bezpośrednią władzę nad Komitetem. Organizuje również pracę Komitetu poprzez rekrutację członków, podział na zespoły projektowe oraz przydział obowiązków w ramach Komitetu. Rada Wykonawcza odpowiada bezpośrednio przed Komitetem Narodowym, który zarządza pracą Stowarzyszenia na terenie Polski. Kadencja członków na każdej z pozycji nie trwa dłużej niż rok, lecz nie jest praktykowane pełnienie tej samej funkcji dłużej niż dwa lata.

Na poziomie lokalnym oraz narodowym jedną z pozycji jest Wiceprzewodniczący do spraw Marketingu.

Rysunek 1 Uproszczona struktura Zarządu Narodowego AIESEC Polska z zależnością z Oddziałami Lokalnym

Oddziały Lokalne realizują swoją strategię marketingową na terenie swojej działalności, która musi być spójna komunikacyjnie ze strategią ogólnopolską. Każdy Wiceprzewodniczący ds. Marketingu posiada swój zespół odpowiedzialny za promocję każdego z produktów prowadzonych przez Oddział lokalny.

Zakładając że w każdym Oddziale Lokalnym średnio 5 osób realizujących zadania związane z promocją i marketingiem w sumie mamy ok 100 osób, których praca musi być zdewaluowana. Tak duża struktura w której pracują osoby dopiero uczące się jak prowadzić efektywne kampanie, potrzebuje prostego narzędzia, które w przystępny sposób pomoże zarządowi ocenę ich pracy oraz dostarczy informacji pomocnych do rozwoju umiejętności członków organizacji.

1. Opis funkcji systemu
   1. Ogólny opis systemu

Celem projektu jest stworzenie narzędzia do analizy w czasie rzeczywistym rezultatów działań marketingowych prowadzonych przez organizację AIESEC Polska. Istotą działania jest zbieranie podstawowych danych marketingowych na temat klientów organizacji w początkowych etapach zaangażowania oraz obliczenia konwersji. Narzędzie ma zbierać, na podstawie specjalnie generowanych linków, informacje o ilości odwiedzin strony internetowej danego produktu organizacji oraz ilość osób zainteresowanych konkretnym produktem, poprzez formularz zgłoszeniowy. Dzięki temu mechanizmowi możliwa będzie prezentacja źródła z którego przyszedł dany klient oraz efektywności danego kanału promocji.

* 1. Udziałowcy i użytkownicy
     1. Udziałowcy

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Opis** | **Odpowiedzialność** |
| Stowarzyszenie AIESEC Polska | Globalna organizacja pozarządowa zrzeszająca młodych ludzi na całym świecie | Wiceprzewodniczący do spraw Marketingu AIESEC Polska jest odpowiedzialny za:  - dostarczenie specyfikacji systemu  - akceptację kolejnych etapów projektu  - zatwierdzenie budżetu | |

Tabela 1 Udziałowcy systemu

* + 1. Użytkownicy

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Opis** | **Odpowiedzialność** |
| Administrator | Osoba posiadająca największe uprawnienia do zarządzania systemem. W szczególności członek Zarządu Narodowego AIESEC Polska. | - Zarządza użytkownikami  - zarządza aktywnymi kampaniami  - zarządza promowanymi produktami  - zarządza oddziałami lokalnymi organizacji |
| Użytkownik | Status użytkownika mają osoby z Oddziałów Lokalnych zaangażowane w działania promocyjne. | Osoba głównie korzystająca z systemu w celu generowania linków promocyjnych oraz ma dostęp do analiz prezentowanych przez system. |
| Kandydat | Osoba zainteresowana usługami AIESEC, która rejestruje się przez formularz na stronie internetowej | - |

Tabela 2 Użytkownicy systemu

* 1. Skrócony opis funkcji systemu na podstawie wywiadu wśród przyszłych użytkowników systemu

Istotą działania systemu jest zbieranie podstawowych danych marketingowych na temat klientów organizacji w początkowych etapach zaangażowania. Narzędzie ma zbierać informacje o ilości odwiedzin strony internetowej danego produktu organizacji oraz, poprzez formularz zgłoszeniowy, dane klienta. Następnie system będzie prezentował źródła, z którego przyszedł dany klient oraz efektywność danego kanału promocji. Dane o klientach mogą zostać pobrane w pliku CSV w celu dalszych analiz. System może zostać zintegrowany z innymi platformami poprzez udostępnione API w celu wsparcia dalszych procesów w organizacji.

Głównym ciałem systemu jest aplikacja internetowa, którą można w łatwy sposób zainstalować na swoim serwerze. Aplikacja obsługuje następujących użytkowników z funkcjami:

* Administrator – jest to osoba posiadająca największe uprawnienia do zarządzania systemem. W szczególności członek Zarządu Narodowego AIESEC Polska. Jego uprawnienia pozwalają na:
  + Zarządzenie użytkownikami.
  + Zarządzanie oddziałami lokalnymi.
  + Zarządzanie produktami.
  + Zarządzanie aktywnymi kampaniami promocyjnymi.
  + Podłączanie formularzy rejestracyjnych na stronie organizacji do systemu.
  + Wszystko to co może użytkownik.
* Użytkownik – jest to osoba, która ma dostęp do generatora linków oraz wszystkich analiz. Status użytkownika mają osoby z Oddziałów Lokalnych zaangażowane w działania promocyjne. Ich uprawnienia pozwalają na:
  + Generowanie linków promocyjnych.
  + Dostęp do wykresów analiz danych.
  + Filtrowanie analiz według kampanii oraz dat.
  + Pobieranie pliku CSV z danymi.

Ponadto system musi mieć zaimplementowaną integrację z globalnym systemem AIESEC w celu analizy ścieżki klienta już po rejestracji. System również poprzez otwarte API pozwala na integrację z zewnętrznymi systemami.

* 1. Scenariusze użycia (user stories)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Logowanie użytkownika | | | |
| **Numer scenariusza** | A-1 | | **Pracochłonność w punktach** | 2 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator. Użytkownik | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia logowanie użytkownika poprzez formularz. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | W celu przeprowadzenia autentykacji użytkownika wymagane jest podanie loginu w formie adresu e-mail oraz hasła. | | |
| Warunki Satysfakcji | | Poprawne wykonanie polega na zidentyfikowaniu użytkownika i wygenerowanie tokenu autoryzacyjnego na podstawie jego uprawnień. | | |

Tabela 3 Scenariusz - Logowanie użytkownika

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Wylogowanie użytkownika | | | |
| **Numer scenariusza** | A-2 | | **Pracochłonność w punktach** | 2 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator. Użytkownik | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia wylogowanie użytkownika po zalogowaniu z każdego miejsca w systemie. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Użytkownik który zakończył pracę w systemie powinien mieć możliwość wylogowania i unieważnienia zapisanego tokenu autoryzacyjnego. | | |
| Warunki Satysfakcji | | Poprawne wykonanie polega na usunięciu tokenu autoryzacyjnego z cache przeglądarki, unieważnienie tokenu w bazie danych oraz zaprezentowanie ekranu logowania użytkownikowi. | | |

Tabela 4 Scenariusz - Logowanie użytkownika

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Dodawanie użytkowników | | | |
| **Numer scenariusza** | A-3 | | **Pracochłonność w punktach** | 3 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia administratorowi dodawanie nowych użytkowników. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | W celu korzystania z systemu wymagany jest dostęp do konta przypisanego danej osobie. | | |
| Warunki Satysfakcji | | Poprawne wykonanie polega na udostępnieniu zasobów systemu nowemu użytkownikowi po zalogowaniu. | | |

Tabela 5 Scenariusz - Dodawanie użytkowników

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Usuwanie użytkowników | | | |
| **Numer scenariusza** | A-4 | | **Pracochłonność w punktach** | 2 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia administratorowi usuwanie kont użytkowników. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Administrator ma możliwość zablokowania dostępu do systemu istniejącym użytkownikom. | | |
| Warunki satysfakcji | | Usunięty użytkownik nie ma dostępu do zasobów systemu. | | |

Tabela 6 Scenariusz - Usuwanie użytkowników

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Dodawanie oddziałów lokalnych | | | |
| **Numer scenariusza** | B-1 | | **Pracochłonność w punktach** | 3 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia administratorowi dodawanie oddziałów lokalnych jako podstawowych jednostek organizacyjnych. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Każdy oddział lokalny prowadzi własne działania marketingowe i w związku z tym potrzebuje analizy przeprowadzane indywidualnie. | | |
| Warunki satysfakcji | | Poprawne wykonanie polega na utworzeniu reprezentacji oddziału lokalnego jako podstawowej jednostki organizacyjnej w systemie. | | |

Tabela 7 Scenariusz - Dodawanie oddziałów lokalnych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Usuwanie oddziałów lokalnych | | | |
| **Numer scenariusza** | B-2 | | **Pracochłonność w punktach** | 2 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia administratorowi usuwanie oddziałów lokalnych. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Oddziały lokalne są podstawowymi jednostkami organizacyjnymi AIESEC, które mogą zostać zamknięte decyzją Walnego Zebrania. | | |
| Warunki satysfakcji | | W momencie usunięcia oddziału lokalnego z systemu wszystkie dane do analiz są traktowane jako narodowe. | | |

Tabela 8 Scenariusz - Usuwanie oddziałów lokalnych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Dodawanie uniwersytetów | | | |
| **Numer scenariusza** | B-3 | | **Pracochłonność w punktach** | 3 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia administratorowi dodawanie uniwersytetów na których prowadzona jest działalność organizacji oraz powiązywanie ich z Oddziałami Lokalnymi | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Każdy oddział lokalny prowadzi działalność na wielu uniwersytetach. Czasami zdarza się że w jednym mieście działa więcej niż jeden Oddział Lokalny i w związku z tym uniwersytety muszą być podzielone pomiędzy oddziały. | | |
| Warunki satysfakcji | | Poprawne wykonanie polega na utworzeniu reprezentacji uniwersytetu oraz połączenie go z oddziałem lokalnym. | | |

Tabela 9 Scenariusz - Dodawanie uniwersytetów

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Usuwanie uniwersytetów | | | |
| **Numer scenariusza** | B-4 | | **Pracochłonność w punktach** | 2 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia administratorowi usuwanie uniwersytetów. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Uniwersytety w ramach jednego miasta mogą zostać wyłączone z obszaru działalności AIESEC. | | |
| Warunki satysfakcji | | Uniwersytet nie jest widoczny w systemie. | | |

Tabela 10 Scenariusz - Usuwanie uniwersytetów

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Dodawanie produktów organizacji | | | |
| **Numer scenariusza** | C-1 | | **Pracochłonność w punktach** | 3 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia dodawanie aktualnych produktów promowanych przez organizację. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Każdy oddział lokalny prowadzi działania promocyjne produktów organizacji. Każdy produkt musi mieć swoją reprezentację w systemie. Analizy generowane są dla efektywności promocji danych produktów. | | |
| Warunki satysfakcji | | Poprawne wykonanie polega na utworzeniu reprezentacji produktu w systemie oraz zbieraniu statystyk dla danego produktu. | | |

Tabela 11 Scenariusz - Dodawanie produktów organizacji

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Usuwanie produktów organizacji | | | |
| **Numer scenariusza** | C-2 | | **Pracochłonność w punktach** | 2 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia usuwanie produktów. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Każdy produkt może zostać zamknięty przez organizację. | | |
| Warunki satysfakcji | | Poprawne wykonanie usunięciu reprezentacji produktu z systemu ale dane klientów powiązane z produktem muszą pozostać nienaruszone. | | |

Tabela 12 Scenariusz - Usuwanie produktów organizacji

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Uruchamianie kampanii marketingowej | | | |
| **Numer scenariusza** | D-1 | | **Pracochłonność w punktach** | 3 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia uruchomienie kampanii marketingowej równolegle dla wszystkich oddziałów lokalnych. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Kampanie prowadzone są równocześnie we wszystkich oddziałach lokalnych i trwają przez określony czas. | | |
| Warunki satysfakcji | | Analizy w systemie są kategoryzowany według kampanii. W ramach kampanii można promować wiele produktów. | | |

Tabela 13 Scenariusz - Uruchamianie kampanii marketingowej

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Archiwizacja kampanii marketingowej | | | |
| **Numer scenariusza** | D-2 | | **Pracochłonność w punktach** | 2 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia zarchiwizowanie kampanii marketingowej. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Zarchiwizowane kampanie nie mogą być wykorzystywane w działaniach marketingowych. | | |
| Warunki satysfakcji | | Nowi klienci przychodzący z promocji w ramach zarchiwizowanej kampanii są oznaczani jako ruch naturalny niepowiązany z żadną kampanią. | | |

Tabela 14 Scenariusz - Archiwizacja kampanii marketingowej

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Komunikacja z formularzami rejestracyjnymi na stronie www | | | |
| **Numer scenariusza** | E-1 | | **Pracochłonność w punktach** | 5 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia komunikację z formularzami rejestracyjnymi na stronie internetowej. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Proces rejestracji Kandydatów przeprowadzany jest za pomocą formularzy rejestracyjnych zamieszczanych na stronie internetowej. | | |
| Warunki satysfakcji | | Formularze muszą spełniać następujące wymagania:  - polami wymaganymi są: imię i nazwisko oraz email,  - formularz może mieć zmienną ilość pól | | |

Tabela 15 Scenariusz - Tworzenie formularzy rejestracyjnych na stronę www

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Generowanie linków promocyjnych | | | |
| **Numer scenariusza** | E-2 | | **Pracochłonność w punktach** | 1 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Użytkownik, Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System umożliwia generowanie adresów URL z parametrami śledzącymi na potrzeby promocji. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | System powinien wykorzystywać do śledzenia aktywnych linków promocyjnych tak zwanych tagów UTM wykorzystywanych między innymi w systemie Google Analytics. Użytkownik może wygenerować link na podstawie aktualnych kampanii, produktów oraz oddziałów lokalnych. | | |
| Warunki satysfakcji | | Na podstawie wygenerowanego linku można zbierać dane na temat działań promocyjnych prowadzonych przez Użytkowników. | | |

Tabela 16 Scenariusz - Generowanie linków promocyjnych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa scenariusza** | Prezentacja analiz marketingowych | | | |
| **Numer scenariusza** | E-3 | | **Pracochłonność w punktach** | 13 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Użytkownik, Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | System prezentuje analizy w formie wykresów na temat prowadzonych działań. | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Użytkownicy systemu oczekują analiz w czasie rzeczywistym odnośnie prowadzonych przez nich działań. Analizy powinny dostarczać informacji na temat efektywności aktualnych kampanii prowadzonych w różnych mediach internetowych dla różnych Oddziałów Lokalnych. | | |
| Warunki satysfakcji | | Wykresy prezentują informacje na temat ilości leadów oraz kandydatów generowanych przez poszczególne kanały. Ponadto wyliczana jest konwersja rejestracji Kandydatów. Analizy są podzielone na produkty oraz oddziały lokalne. System również umożliwia pobranie surowych danych do dalszych analiz w formacie CSV. | | |

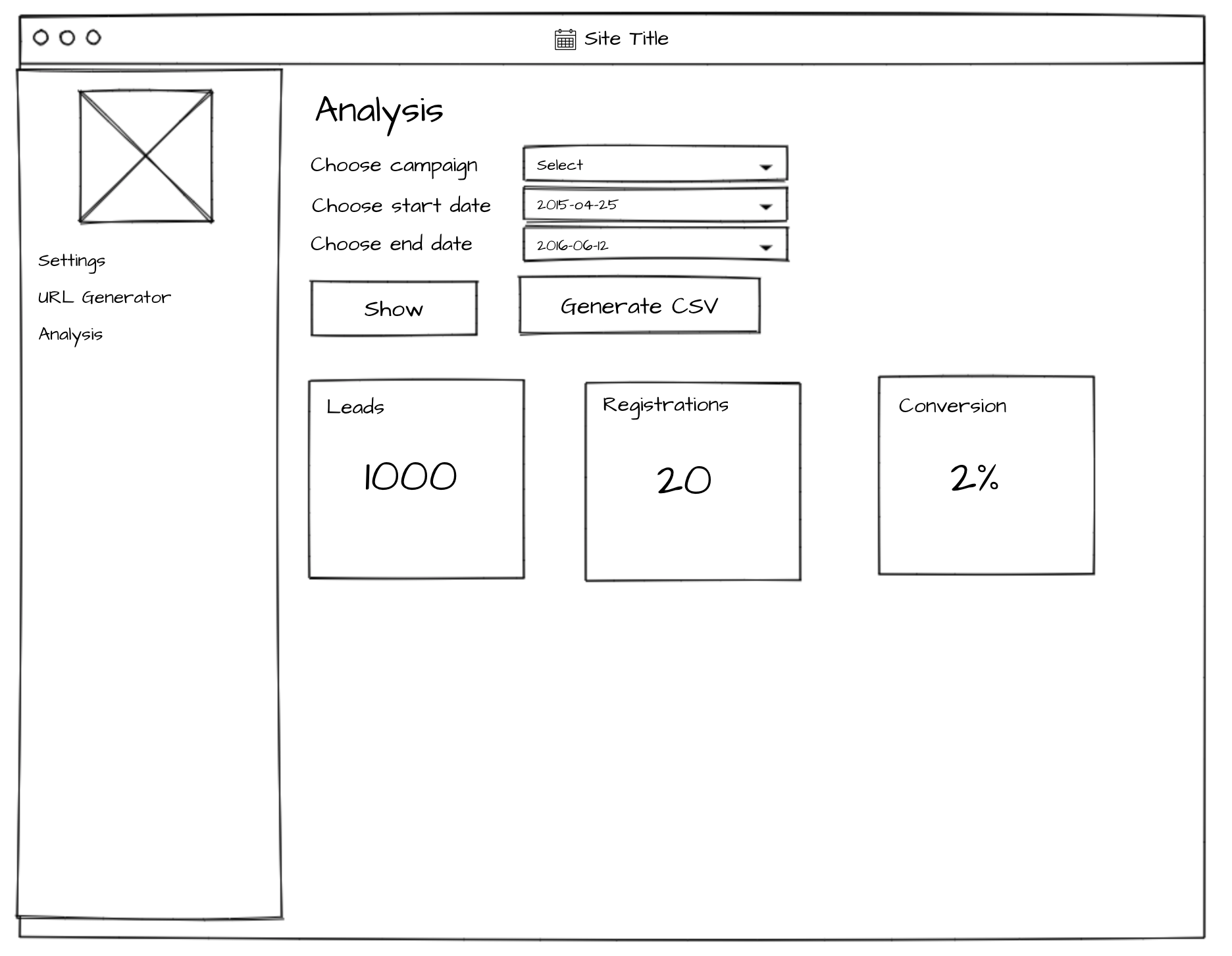
Tabela 17 Scenariusz - Prezentacja analiz marketingowych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numer scenariusza** | E-4 | | **Pracochłonność w punktach** | 5 |
| **TREŚĆ SCENARIUSZA** | | | | |
| **Rodzaj użytkownika systemu** | | Użytkownik, Administrator | | |
| Cel  (pożądana cecha lub problem, który trzeba rozwiązać) | | Integracja z globalnym systemem aiesec.org | | |
| Uzasadnienie  (korzyść płynąca z ukończenia story) | | Klienci którzy zarejestrują się poprzez formularz muszą być również zapisani w globalnym systemie organizacji AIESEC w celu aplikowania na praktyki lub wolontariaty. | | |
| Warunki satysfakcji | | Każdy zarejestrowany klient ma automatycznie założone konto w systemie aiesec.org. Integracja jest realizowana poprzez API dostarczone przez AIESEC International | | |

* 1. Projekt Interfejsu użytkownika

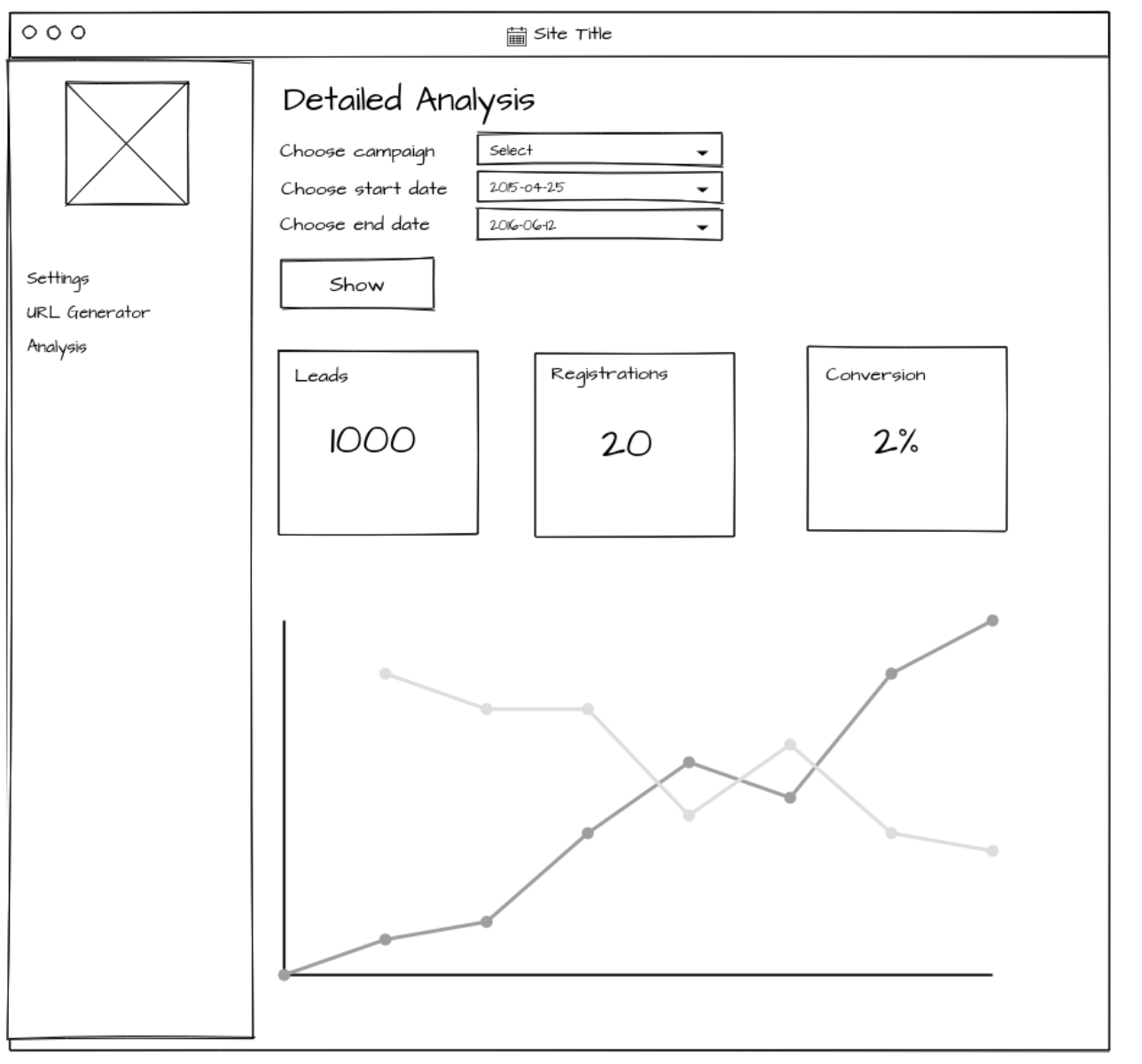
Interfejs użytkownika z założenia miał być prosty w obsłudze i intuicyjny. Do jego zbudowania został wykorzystany darmowy szablon *DASHGUM BASIC (*[*https://gridgum.com/themes/dashgum-basic-free-dashboard-template/)*](https://gridgum.com/themes/dashgum-basic-free-dashboard-template/))*. Został on podzielony na 4 główne części:*

1. *Ekran analiz*
2. *Ekran Analiz szczegółowych*
3. *Ekran ustawień*
4. *Ekran generowania adresów URL*
   * 1. Model szkieletowy ekranu analiz

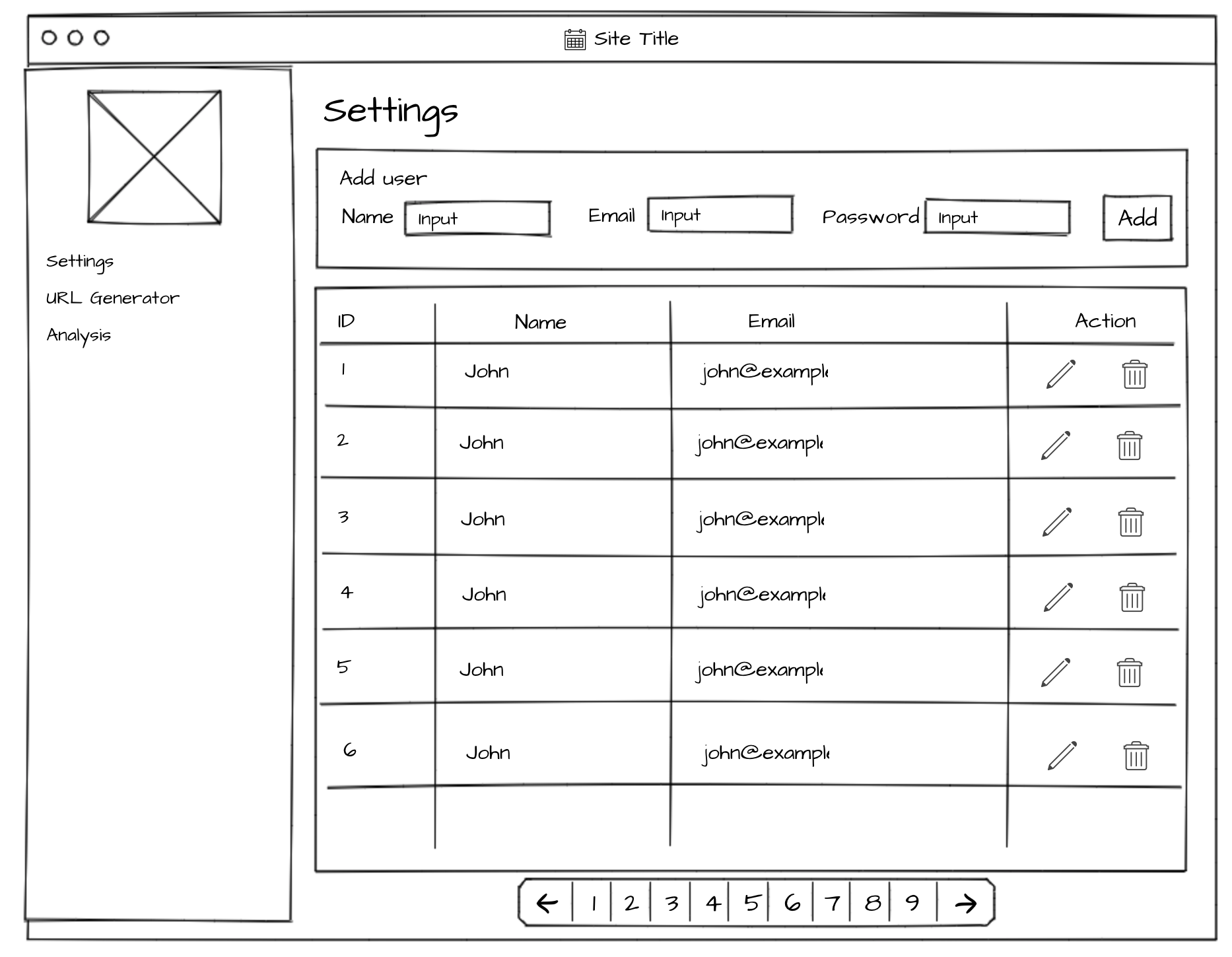


Rysunek 2 Model szkieletowy ekranu analiz

* + 1. Model szkieletowy ekranu analiz szczegółowych

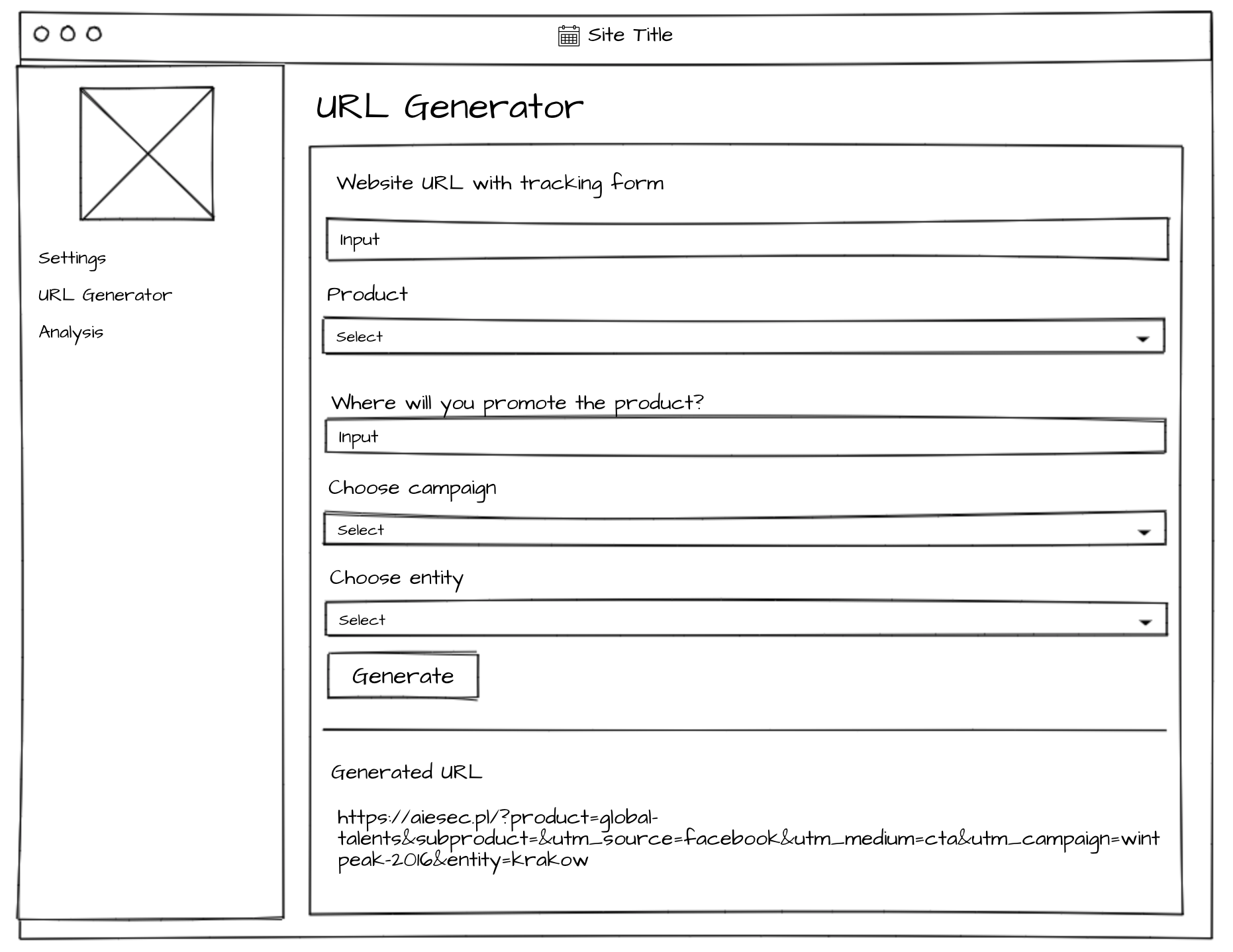


* + 1. Model szkieletowy ekranu ustawień



Rysunek 3 Model szkieletowy ekranu ustawień

* + 1. Model szkieletowy ekranu generowania adresów URL



Rysunek 4 Model szkieletowy ekranu generowania adresów URL

1. Opis techniczny systemu
   1. Wybór technologii

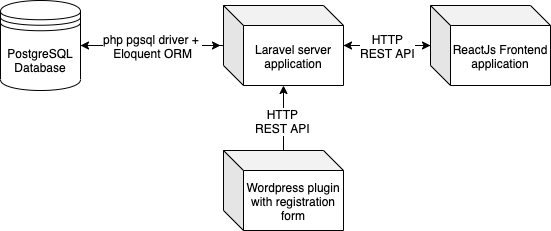
Najważniejszymi kryteriami, którymi kierowałem się przy wyborze technologii wykorzystanych przy implementacji systemu była moja znajomość danego rozwiązania oraz próg wejścia do nauczenia się nowego. Ponadto chciałem, żeby zachowane były podstawowe warstwy architektoniczne aplikacji (danych, logiki biznesowej oraz prezentacji), które są w pełni od siebie odseparowane i korzystają jedynie z dostarczonego API. W idealnym przypadku pożądane jest aby była możliwość instalacji każdej z warstw na innym serwerze w celu ochrony przed nieautoryzowanym dostępem.

Dodatkowo chciałem wykorzystać nowoczesne praktyki budowania aplikacji internetowych wykorzystując takie wzorce jak Single-Page Application przy prezentacji danych czy elementy baz danych NoSql.

Bazując na powyższych wymaganiach wybrałem następujące technologie:

* Baza danych: PostgreSQL 9.4.10 wykorzystująca moduł hstore (3) do przechowywania danych w postaci klucz -> wartość
* Środowisko logiki biznesowej:
  + Laravel Framework 5.2.45
  + PHP 7.0.15
  + Autoryzacja: JSON Web Tokens oparta o standard RFC7519 (3)
  + Komunikacja: RESTful API oparta o dokumenty JSON
* Prezentacja:
  + JavaScript
  + React JS 15.0.2
  + Redux 3.5.2
  + ECMAScript 2015
  1. Opis architektury
     1. Wysokopoziomowy schemat architektury aplikacji

Aplikacja została podzielona na 3 niezależne warstwy: danych, logiki oraz prezentacji. Warstwy te mogą być zainstalowane na osobnych środowiskach a komunikują się między sobą za pomocą zdefiniowanych interfejsów.



Rysunek 5 Schemat architektury aplikacji

* + 1. Baza danych

Struktura relacyjnej bazy danych składa się z trzech grup tabel zawierających dane odnośnie logiki biznesowej, zarządzania użytkownikami oraz statycznych kluczy API.

* + - 1. Logika biznesowa

Rysunek 6 Schemat relacji tabel bazy danych odpowiadających za logikę biznesową

Tabele odpowiedzialne za przechowywanie danych logiki biznesowej aplikacji połączone są relacjami widocznymi na rysunku powyżej. Elementem o którym warto wspomnieć jest zastosowanie pola typu *hstore* w tabeli *customers.* Moduł *hstore* pozwala na przechowywanie danych w postaci klucz -> wartość w tradycyjnej relacyjnej bazie danych. Dzięki temu mechanizmowi można uzyskać elastyczność dokumentowej bazy danych razem z zachowaniem struktur relacyjnych.

Do zaprezentowanych tabel zostały również przygotowane dwa widoki (*lead\_customer* oraz *utm\_source\_medium*) ułatwiające odczyt danych w aplikacji.

* + - 1. Klucze API



Rysunek 7 Schemat tabeli przechowującej klucze API

* + - 1. Role i użytkownicy



Rysunek 8 Schemat relacji tabel bazy danych przechowujących informacje odnośnie użytkowników i ich ról

* + 1. Serwer

Aplikacja serwerowa została napisana we frameworku PHP Laravel. Jest to platforma pozwalająca na tworzenie w pełni funkcjonalnych aplikacji internetowych. W związku z tym że postanowiłem całkowicie odseparować warstwę prezentacji od warstwy logiki w ramach aplikacji Laravel skupiłem się na implementacji kontrolerów zawierających logikę biznesową aplikacji, komunikacji z bazą danych oraz wystawieniu interfejsu komunikacyjnego REST.

Architektura ogranicza się do definicji modeli danych wykorzystujących silnik Eloquent ORM, kontrolerów wykonujących operacje na danych oraz zdefiniowanego routingu do interfejsu REST.

* + - 1. Autoryzacja dostępu

Autoryzacja dostępu do poszczególnych elementów systemu przeprowadzana jest za pomocą tokenów JWT - JSON Web Token. Jest to standard stworzony przez organizację Internet Engineering Task Force (IETF) opisany w dokumencie RFC 7519. JWT Służy do generowania tokenów autoryzacji w systemach internetowych w formacie JSON. Tokeny są podpisywane za pomocą algorytmu HS256. Wewnątrz tokenu zapisane są między innymi informacje o dacie ważności czy dane serwera emitującego token. Po wygenerowaniu tokenu i przekazaniu do serwera, sprawdzana jest jego poprawność oraz poziom dostępu do danych.

* + - * 1. Przykładowy token JWT

**Przykładowy token wygenerowany przez serwer**

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyIwIjoic3ViIiwiMSI6IjU4ZGZhNTQ5OTQ4ZjUiLCJzdWIiOjMsImlzcyI6Imh0dHA6Ly92cHMzMzQzNzIub3ZoLm5ldDo4MDgwL2FwaS92MS9hcGkta2V5cyIsImlhdCI6MTQ5MTA1MTg0OSwiZXhwIjoxNTU0MDc2ODA5LCJuYmYiOjE0OTEwNTE4NDksImp0aSI6IjdiYmI1MjgwYmQzYTczOGM5YTdkY2RmNTgzYzE1Y2UwIn0.3MzSbbnChtmzi0Q6QUML-70atKPeV6N7-\_KzXhWFqTk

**Informacje odszyfrowane z tokenu**

Nagłówek: algorytm i typ tokenu

{

"typ": "JWT", //typ tokenu

"alg": "HS256" //algorytm szyfrujący

}

Blok danych:

{

"0": "sub",

"1": "58dfa549948f5",

"sub": 3, //numer tematu

"iss": "http://vps334372.ovh.net:8080/api/v1/api-keys", //adres serwera emitującego token

"iat": 1491051849, //data wydania tokenu UNIX time

"exp": 1554076809, //data ważności tokenu UNIX time

"nbf": 1491051849, //data od kiedy token jest ważny UNIX time

"jti": "7bbb5280bd3a738c9a7dcdf583c15ce0" //unikalny identyfikator tokenu

}

Podpis:

HMACSHA256(

base64UrlEncode(header) + "." +

base64UrlEncode(payload),

your-256-bit-secret

)

* + - 1. Routing zapytań interfejsu REST

<?php

Route::get('/', function () {

return App::abort(404);

});

/\*\*

\* Displays API docs by Swagger

\*/

Route::get("/api-docs", function() { return Redirect::to("/swagger-ui/dist/index.html"); });

/\*\*

\* Needs to be called to install app and specify configuration.

\*

\*/

Route::post('/install', 'InstallController@index');

Route::get('/run-migrations', 'InstallController@runMigrations');

Route::post('/init-permissions', 'PermissionsController@create');

Route::post('/init-roles', 'RolesController@create');

Route::group(['prefix' => 'api/v1'], function()

{

Route::resource('authenticate', 'AuthenticateController', ['only' => ['index']]);

Route::post('authenticate', 'AuthenticateController@authenticate');

Route::get('authenticate/user', 'AuthenticateController@getAuthenticatedUser');

Route::get('logout', 'AuthenticateController@logout');

//////////////\*\*\*\*\*\*\*\*\*\*USERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

//create new user

Route::post('/users', [

'uses' => 'UserController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.user']);

//delete user by id

Route::delete('/users/{id}', [

'uses' => 'UserController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.user']);

//get all users

Route::get('/users', [

'uses' => 'UserController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.user']);

//////////////\*\*\*\*\*\*\*\*\*\*LEADS\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

//create new lead

Route::post('/leads', [

'uses' => 'LeadController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.lead']);

//delete lead by id

Route::delete('/leads/{id}', [

'uses' => 'LeadController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.lead']);

//get all leads

Route::get('/leads/', [

'uses' => 'LeadController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.lead']);

Route::get('/leads/populate', [

'uses' => 'LeadController@populateLeads',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.lead']);

Route::post('/leads/count', [

'uses' => 'LeadController@count',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.lead']);

Route::post('/leads/leads-statistics', [

'uses' => 'LeadController@getLeadsAnalysis',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.lead']);

//////////////\*\*\*\*\*\*\*\*\*\*CAMPAIGNS\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

Route::post('/campaigns', [

'uses' => 'CampaignController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.campaign']);

Route::delete('/campaigns/{id}', [

'uses' => 'CampaignController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.campaign']);

Route::get('/campaigns/', [

'uses' => 'CampaignController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.campaign']);

Route::get('/campaigns/{id}', [

'uses' => 'CampaignController@viewById',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.campaign']);

Route::put('/campaigns/{id}', [

'uses' => 'CampaignController@update',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.campaign']);

//////////////\*\*\*\*\*\*\*\*\*\*CUSTOMERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

Route::post('/customers', [

'uses' => 'CustomerController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.customer']);

Route::delete('/customers/{id}', [

'uses' => 'CustomerController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.customer']);

Route::get('/customers/', [

'uses' => 'CustomerController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.customer']);

Route::post('/customers/csv', [

'uses' => 'CustomerController@download',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.customer']);

Route::post('/customers/count', [

'uses' => 'CustomerController@count',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.customer']);

Route::put('/customers/{id}', [

'uses' => 'CustomerController@update',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'update.customer']);

Route::post('/customers/customers-statistics', [

'uses' => 'CustomerController@getCustomersAnalysis',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.customer']);

//////////////\*\*\*\*\*\*\*\*\*\*ENTITY\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

Route::post('/entities', [

'uses' => 'EntityController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.entity']);

Route::delete('/entities/{id}', [

'uses' => 'EntityController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.entity']);

Route::get('/entities/', [

'uses' => 'EntityController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.entity']);

Route::put('/entities/{id}', [

'uses' => 'EntityController@update',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'update.entity']);

//////////////\*\*\*\*\*\*\*\*\*\*PRODUCT\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

Route::post('/products', [

'uses' => 'ProductController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.product']);

Route::delete('/products/{id}', [

'uses' => 'ProductController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.product']);

Route::get('/products/', [

'uses' => 'ProductController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.product']);

Route::get('/products/{id}', [

'uses' => 'ProductController@viewById',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.product']);

Route::put('/products/{id}', [

'uses' => 'ProductController@update',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'update.product']);

//////////////\*\*\*\*\*\*\*\*\*\*SUBPRODUCT\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

Route::post('/products/{product\_id}/subproducts', [

'uses' => 'SubproductsController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.product']);

Route::delete('/products/{product\_id}/subproducts/{id}', [

'uses' => 'SubproductsController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.product']);

Route::get('/products/{product\_id}/subproducts', [

'uses' => 'SubproductsController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.product']);

Route::put('/products/{product\_id}/subproducts/{id}', [

'uses' => 'SubproductsController@update',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'update.product']);

//////////////\*\*\*\*\*\*\*\*\*\*API keys\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

//create new user

Route::post('/api-keys', [

'uses' => 'ApiKeyController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.api\_key']);

//delete user by id

Route::delete('/api-keys/{id}', [

'uses' => 'ApiKeyController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.api\_key']);

//get all users

Route::get('/api-keys', [

'uses' => 'ApiKeyController@view',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.api\_key']);

//////////////\*\*\*\*\*\*\*\*\*\*Universities\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////////

Route::post('/universities', [

'uses' => 'UniversitiesController@create',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'create.university']);

Route::delete('/universities/{id}', [

'uses' => 'UniversitiesController@delete',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'delete.university']);

Route::get('/universities/', [

'uses' => 'UniversitiesController@view']);

Route::get('/universities/{id}', [

'uses' => 'UniversitiesController@viewById',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'view.university']);

Route::put('/universities/{id}', [

'uses' => 'UniversitiesController@update',

'middleware' => ['jwt.auth', 'acl'],

'can' => 'update.university']);

});

Fragment kodu 1 Definicja routingu w aplikacji serwerowej

* + 1. Prezentacja

Aplikacja warstwy prezentacji została napisana w języku JavaScript wykorzystując standaryzację ECMAScript 2015. Aplikacja nie bazuje na żadnym frameworku, implementuje jednak trzy podstawowe biblioteki narzucające schemat postępowania i projektowania aplikacji.

* + - 1. ReactJS

React jest biblioteką dla języka JavaScript służącą do tworzenia interfejsów graficznych aplikacji internetowych. Jedną z jej głównych zalet jest możliwość przeplatania kodu HTML razem z kodem JavaScript dzięki rozwiązaniu JSX. Pozwala on na znacznie szybsze i efektywne generowanie obiektowego modelu dokumentu (DOM).

Drugą z zalet jest komponentowa architektura aplikacji. Każdy element renderowany w przeglądarce może zostać zdefiniowany jako osobny komponent. Taka struktura pozwala na wielokrotne wykorzystywanie raz napisanego kodu w wielu miejscach aplikacji. Ponadto komponenty mają strukturę hierarchiczną i z łatwością komunikują się ze sobą przekazując sobie nawzajem zmiany stanu, poprzez mechanizm Props.

* + - 1. Redux

Redux jest ewolucją wzorca projektowego Flux stworzonego przez firmę Facebook. Jego głównym zadaniem jest zarządzanie przewidywalnym kontenerem stanu aplikacji internetowej.

* 1. Formularz rejestracyjny

Formularz rejestracyjny został stworzony jako plugin do popularnego systemu zarządzania treścią: Wordpress. Plugin składa się z 3 głównych części:

1. Szablonu HTML z edytowalnymi polami formularza
2. Logika generowania numeru identyfikacyjnego, zapisywania pliku cookie i zapisywania lead’a na serwerze
3. Logika identyfikacji klienta na podstawie pliku cookie i zapisywania danych z formularza na serwerze

Kluczowym elementem logiki formularza jest identyfikowanie klienta wchodzącego na stronę www. Aby odpowiednio przeprowadzić identyfikację wykorzystane zostało narzędzie Fingerprint.js (<https://fingerprintjs.com/)>. Fingerprint.js zbiera wszystkie unikalne funkcje z urządzenia oraz przeglądarki a następnie przepuszcza je przez funkcję hash aby zapewnić unikalny identyfikator użytkownika w postaci tokenu tekstowego. Fingerprint.js zbiera informacje takie jak: rozdzielczość ekranu, ustawienia WebGL, zainstalowane czcionki, typu urządzenia, typ przeglądarki oraz wiele innych, które pozwalają na odwzorowanie unikalnego profilu użytkownika.

Gdy „odcisk palca” zostanie wygenerowany jest on zapisywany w pliku cookie przeglądarki a następnie zapisywany na serwerze z informacją z jakiego źródła promocji przyszedł dany klient. Gdy osoba zdecyduje się na wysłanie swoich danych osobowych za pomocą formularza, wysyłany jest również jej „odcisk palca” odczytany z pliku cookie i wysłany razem z danymi osobowymi. Dzięki temu mechanizmowi możemy powiązać rzeczywistą osobę z pierwszym, unikalnym wejściem na stronę www oraz źródłem pierwszego kontaktu z marką za pomocą narzędzi marketingowych.

1. Prezentacja systemu
   1. Informacje podstawowe

Repozytorium z kodem aplikacji: <https://github.com/lycha/masters-thesis>

Środowisko testowe: [http://vps334372.ovh.net](http://vps334372.ovh.net/)

Przykładowy formularz rejestracyjny: <http://aiesecpl.home.pl/testing-wordpress/>

Dokumentacja API: <http://vps334372.ovh.net/docs/api_documentation.htm>

* 1. Instalacja systemu

Aby zainstalować system należy wykonać następujące kroki:

1. Przygotować serwer z konfiguracją zgodną ze specyfikacją opisaną w rozdziale 4.1
2. Skopiować kod źródłowy aplikacji z repozytorium GitHub

git clone git:// <https://github.com/lycha/masters-thesis>.git

1. Wykonać następujące komendy uzupełniając potrzebne dane konfiguracyjne:

curl -X POST -H "Content-Type: application/json" -H "Cache-Control: no-cache" '{

"db-host": "localhost",

"db-database":"homestead",

"db-username":"homestead",

"db-password":"secret"

}' "http://api.domain-name.dev/install"

curl -X GET -H "Cache-Control:no-cache" "http://api.domain-name.dev/run-migrations"

curl -X POST -H "Cache-Control: no-cache" "http://api.domain-name.dev/init-permissions"

curl -X POST -H "Content-Type: application/json" -H "Cache-Control: no-cache" '{

"name": "username",

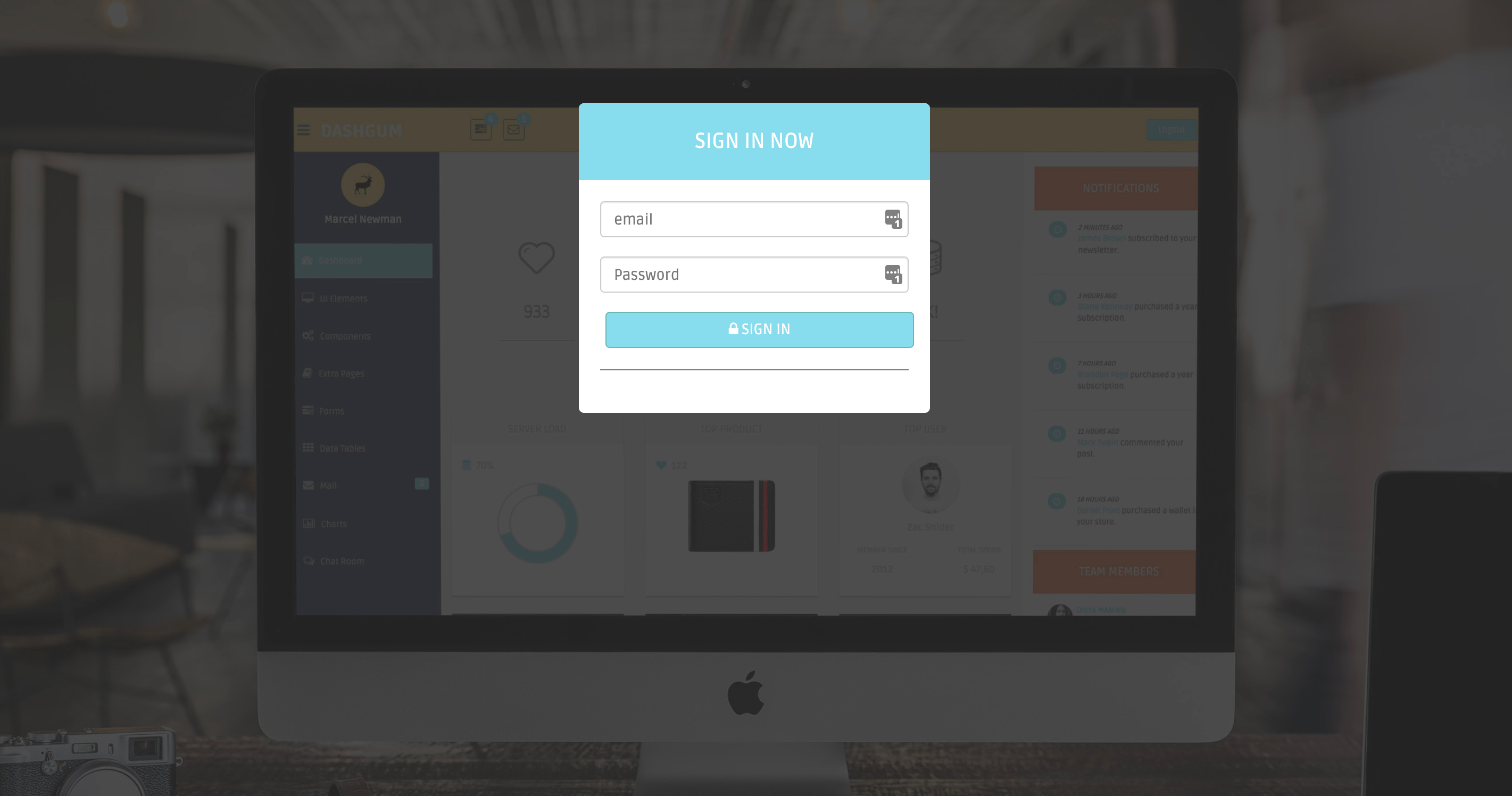
"email": "user@example.com ",

"password": "secret"

}' "http://api.domain-name.dev.dev/init-roles"

1. Aplikacja powinna zostać poprawnie zainstalowana na serwerze
   1. Prezentacja wyników implementacji według scenariuszy użycia
      1. A-1 Logowanie użytkownika

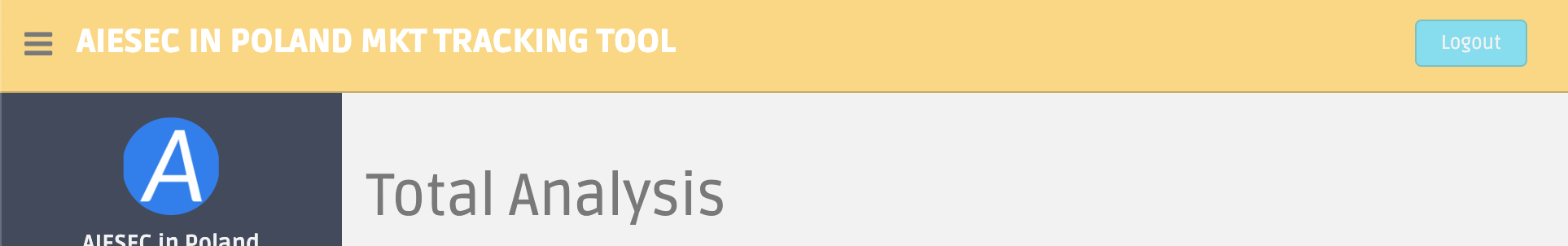
Po wejściu na stronę główną systemu prezentowany jest formularz logowania użytkownika. Aby uzyskać dostęp do systemu należy podać adres email użytkownika oraz hasło zdefiniowane podczas instalacji oraz nacisnąć przycisk *Sign in*.



Rysunek 9 Ekran logowania

* + 1. A-2 Wylogowanie użytkownika

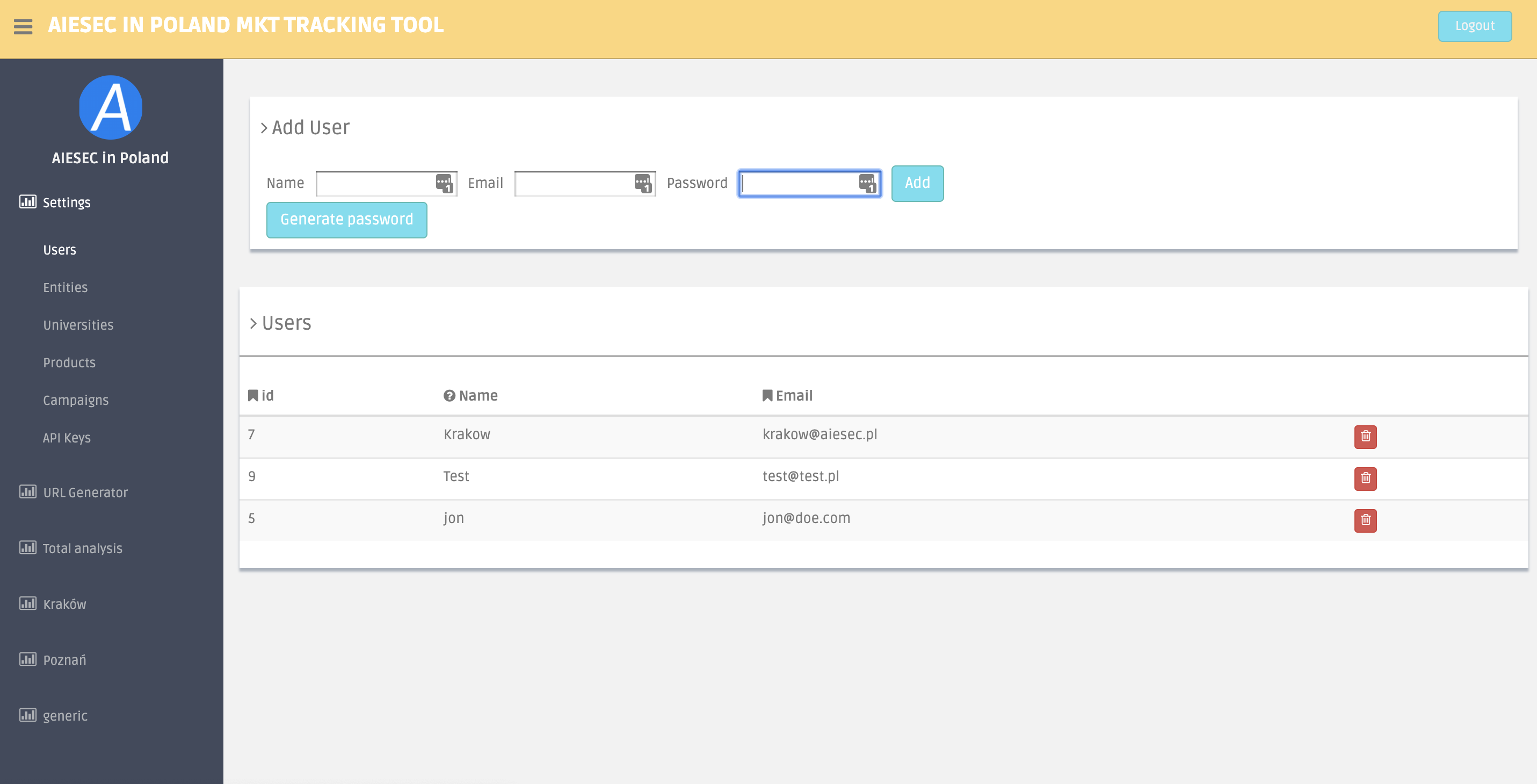
Wylogowanie użytkownika możliwe jest w każdym widoku systemu. W prawym górnym rogu umiejscowiony jest przycisk *Logout*.



Rysunek 10 Przycisk wylogowania użytkownika

* + 1. A-3 Dodawanie użytkowników

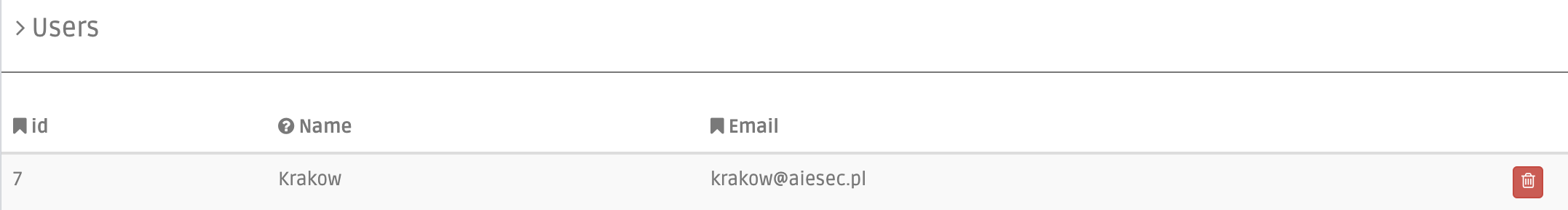
Aby dodać nowego użytkownika należy wybrać z bocznego menu *Settings*  a następnie *Users*. Na ekranie pojawi się formularz dodawania oraz lista obecnych użytkowników. W celu dodania użytkonika należy wypełnić pola formularza podając nazwę, adres e-mail oraz hasło i nacisnąć przycisk *Add*. Dla ułatwienia został dodany również przycisk *Generate pasword* który umożliwia Administratorowi wygenerowanie tajnego dziesięcioznakowego hasła tymczasowego, które zostanie wysłane uzytkownikowi w wiadomości e-mail.



Rysunek 11 Dodawanie nowego użytkownika

* + 1. A-4 Usuwanie użytkowników

Usuwanie użytkowników możliwe jest z tego samego miesjca co ich dodawanie. Na liście wszystkich użytkowników po prawej stronie każdego rekordu umiejscowiony jest czerwony przycisk usuwania. Należy go kliknąć w celu usunięcia konkretnego uzytkownika.

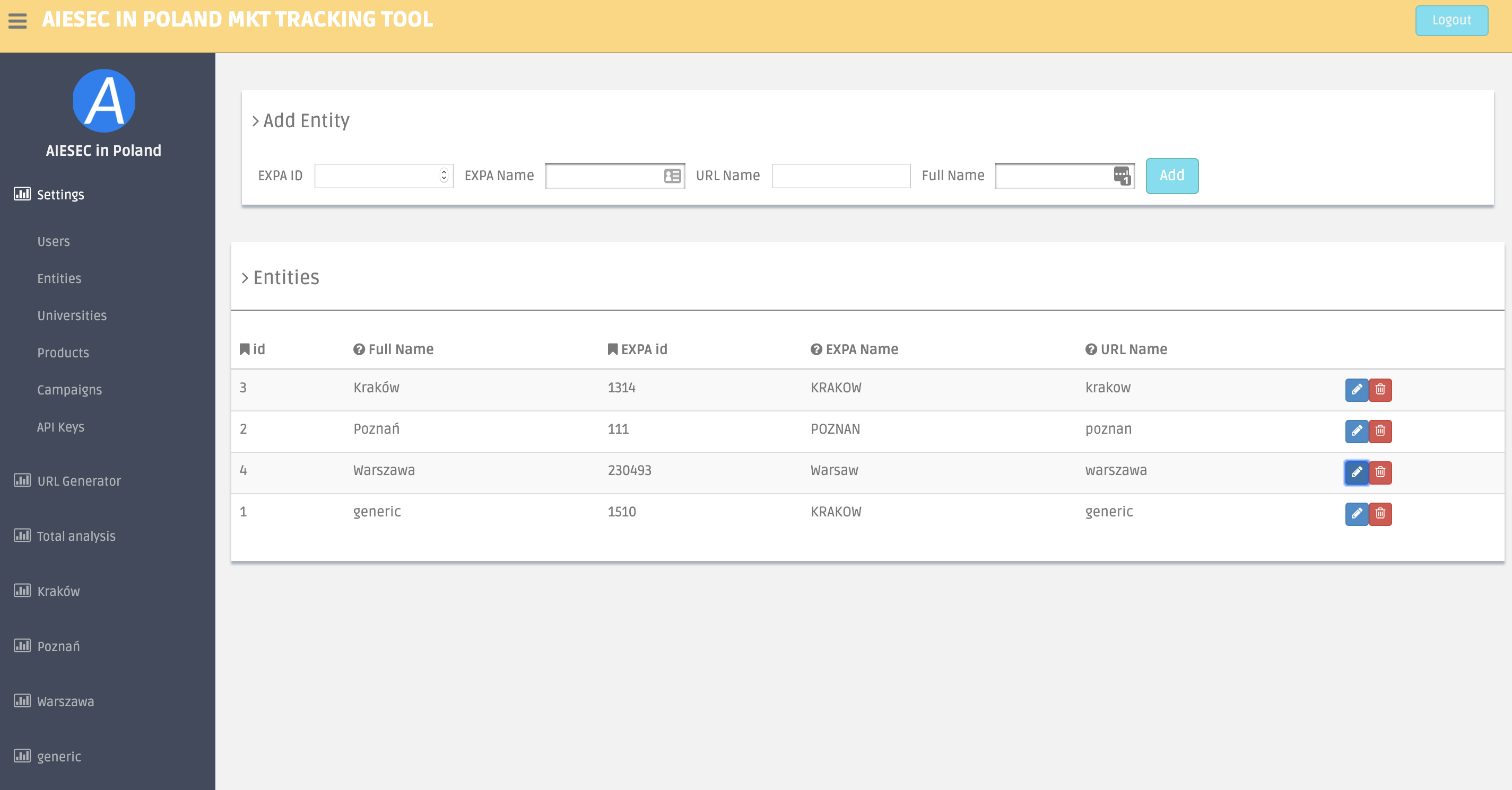


Rysunek 12 Usuwanie użytkowników

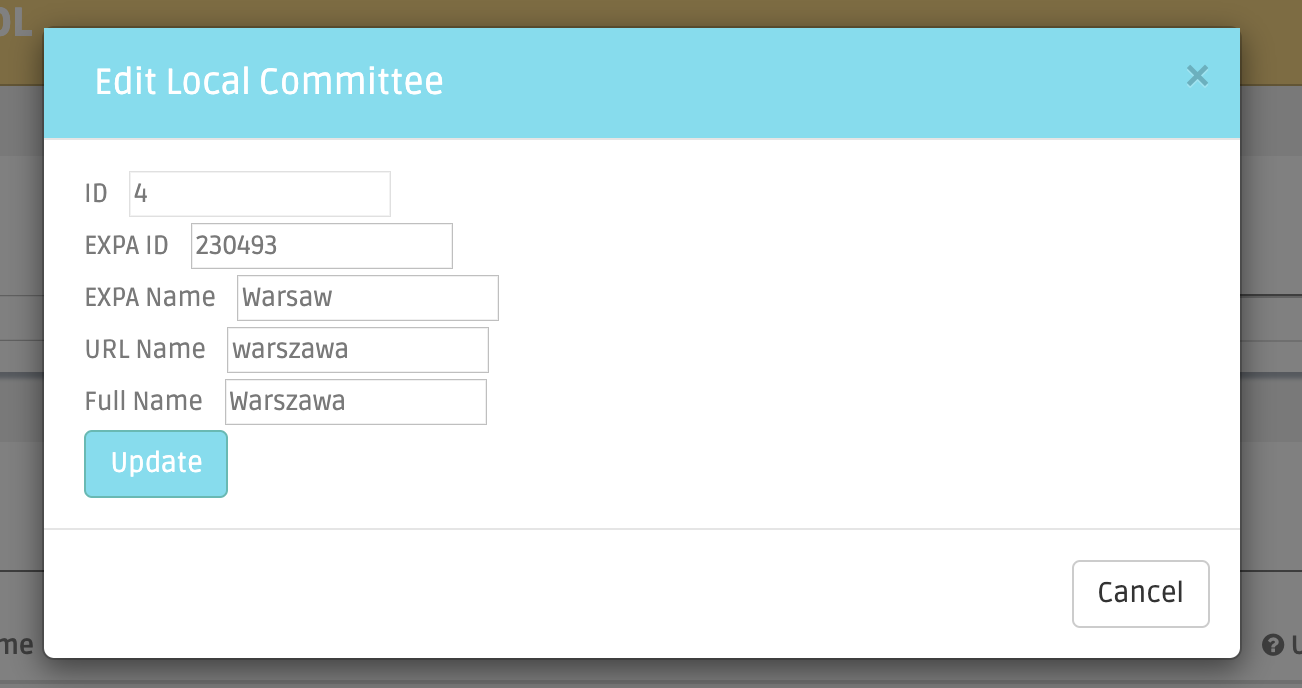
* + 1. B-1 Dodawanie oddziałów lokalnych

W celu dodania nowego odziału lokalnego organizacji należy z menu wybrać *Settings* a następnie *Entities*. Aby dodać nowy oddział należy wypełnić formularz na górze strony a następnie kliknąć *Add*. Po dodaniu nowego oddziału lokalnego pojawi się on również w menu bocznym.

Możliwe jest również edytowanie poprzednio dodanych oddziałów. W tym celu należy kliknąć niebieski przycisk *Edit* znajdujący się po prawe stronie rekordu na liście oddziałów. Następnie pojawi się okno dialogowe (Rysunek 14 Edytowanie oddziału lokalnego) w którym należy zmienić odpowiednie elementy i nacisnąć przycisk *Update*. Aby wyjść z okna dialogowego bez zapisywania należy nacisnąć przycisk *Cancel*.



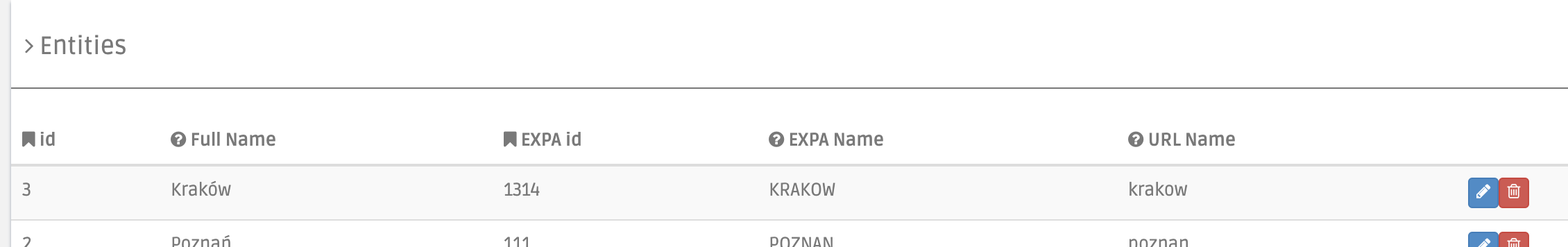
Rysunek 13 Dodawanie oddziałów lokalnych



Rysunek 14 Edytowanie oddziału lokalnego

* + 1. B-2 Usuwanie oddziałów lokalnych

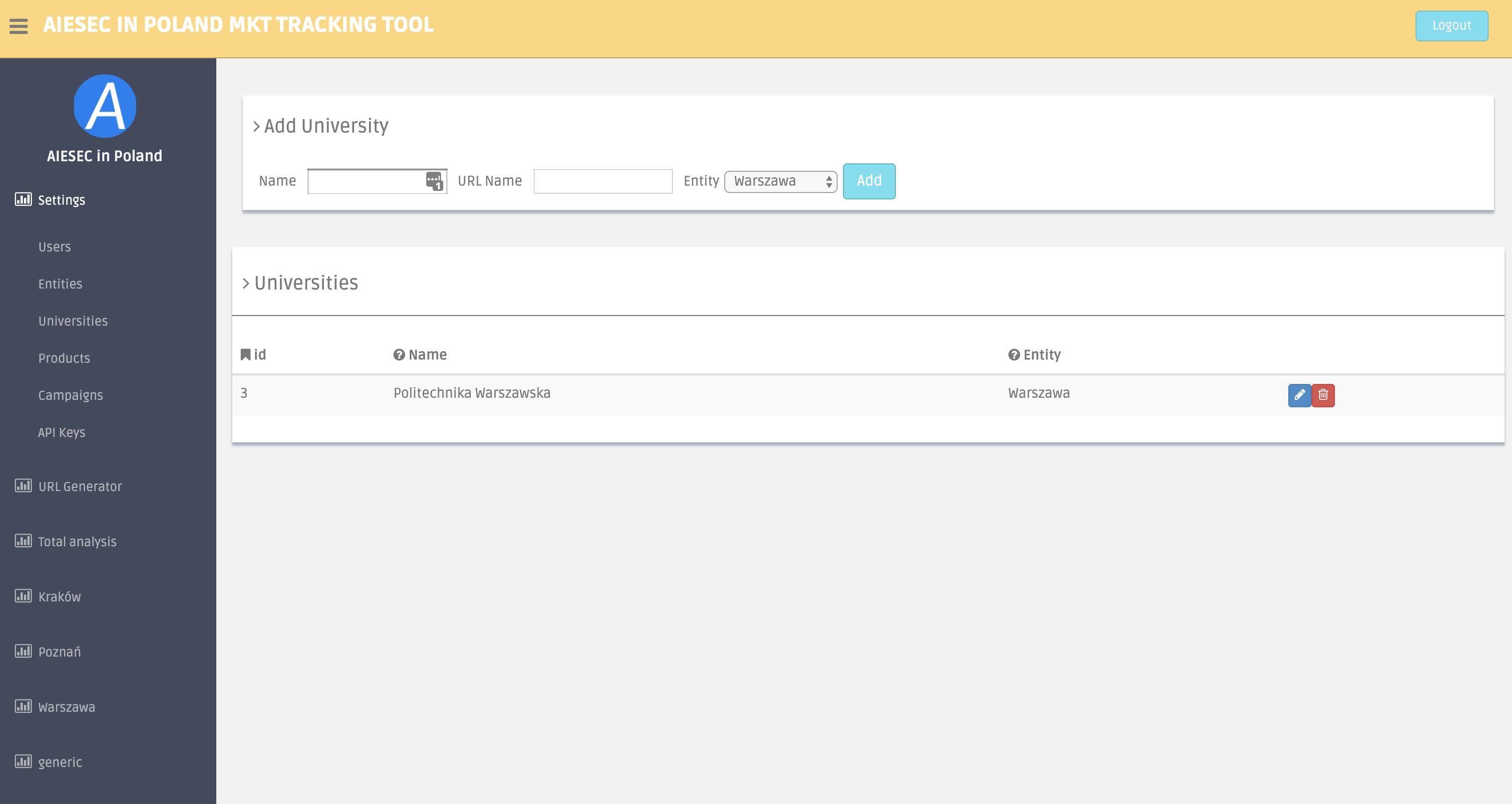
Aby usunąć oddział lokalny należy wejść z poziomu menu w *Settings* a następnie *Entities.* Na liście dostępnych oddziałów lokalnych należy kliknąć czerwony przycisk *Delete* znajdujący się po prawej stronie wybranego elementu.



Rysunek 15 Usuwanie oddziałów lokalnych

* + 1. B-3 Dodawanie uniwersytetów

Aby dodać nowy uniwersytet należy z menu bocznego wybrać *Settings* a następnie *Universities*. Następnie należy podać jego nazwę, nazwę skróconą na potrzeby adresu URL oraz wybrać z listy rozwijanej oddział lokalny działający na danej uczelni.



Rysunek 16 Dodawanie uniwersytetów

* + 1. B-4 Usuwanie uniwersytetów

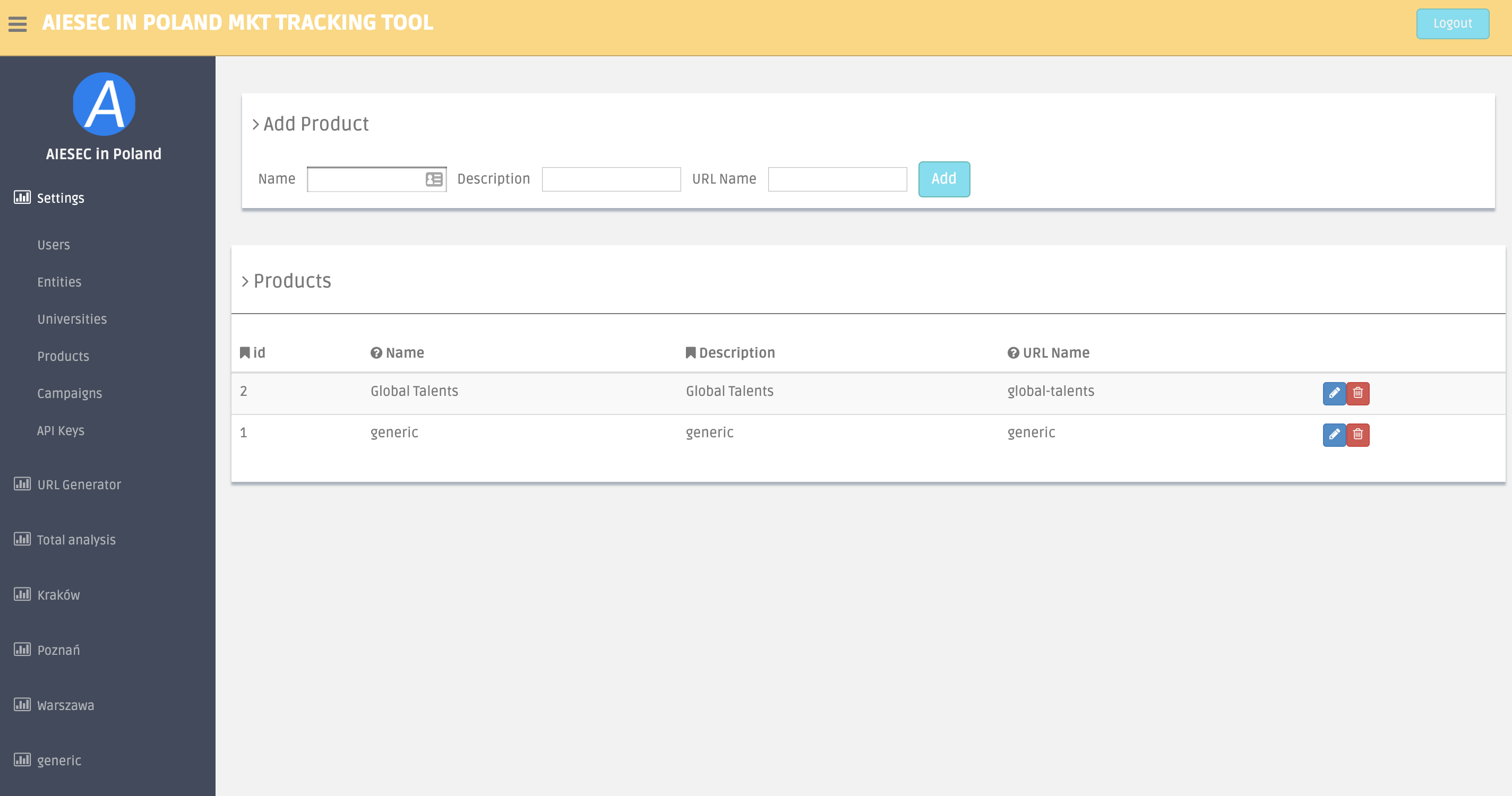
Aby usunąć Uniwersytet należy z menu bocznego wybrać *Settings* a następnie *Universities*. Na ekranie pojawi się lista Uniwersytetów, w celu usunięcia rekordu z listy należy kliknąć w czerwony przycisk *Usuń.*



Rysunek 17 Usuwanie uniwersytetów

* + 1. C-1 Dodawanie produktów organizacji

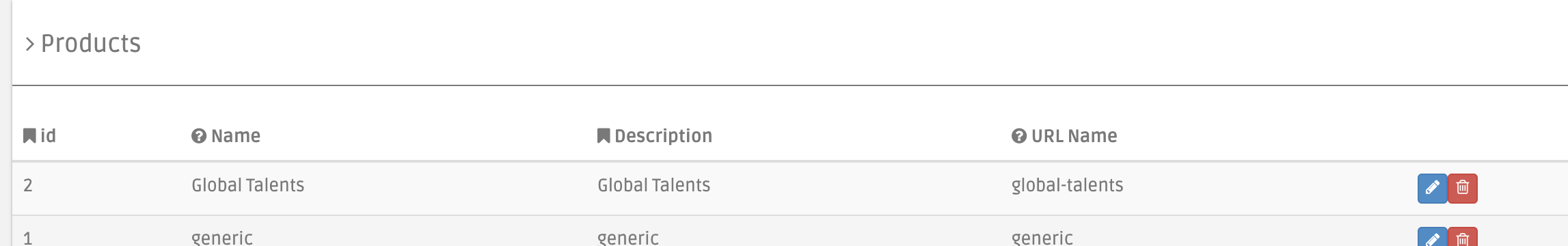
W celu dodania nowego produktu z menu bocznego należy wybrać *Settings* a następnie *Products*. W następnym kroku należy wypełnić nazwę, krótki opis oraz uproszczoną nazwę na potrzeby adresu URL i nacisnąć przycisk *Add*.



Rysunek 18 Dodawanie produktów organizacji

* + 1. C-2 Usuwanie produktów organizacji

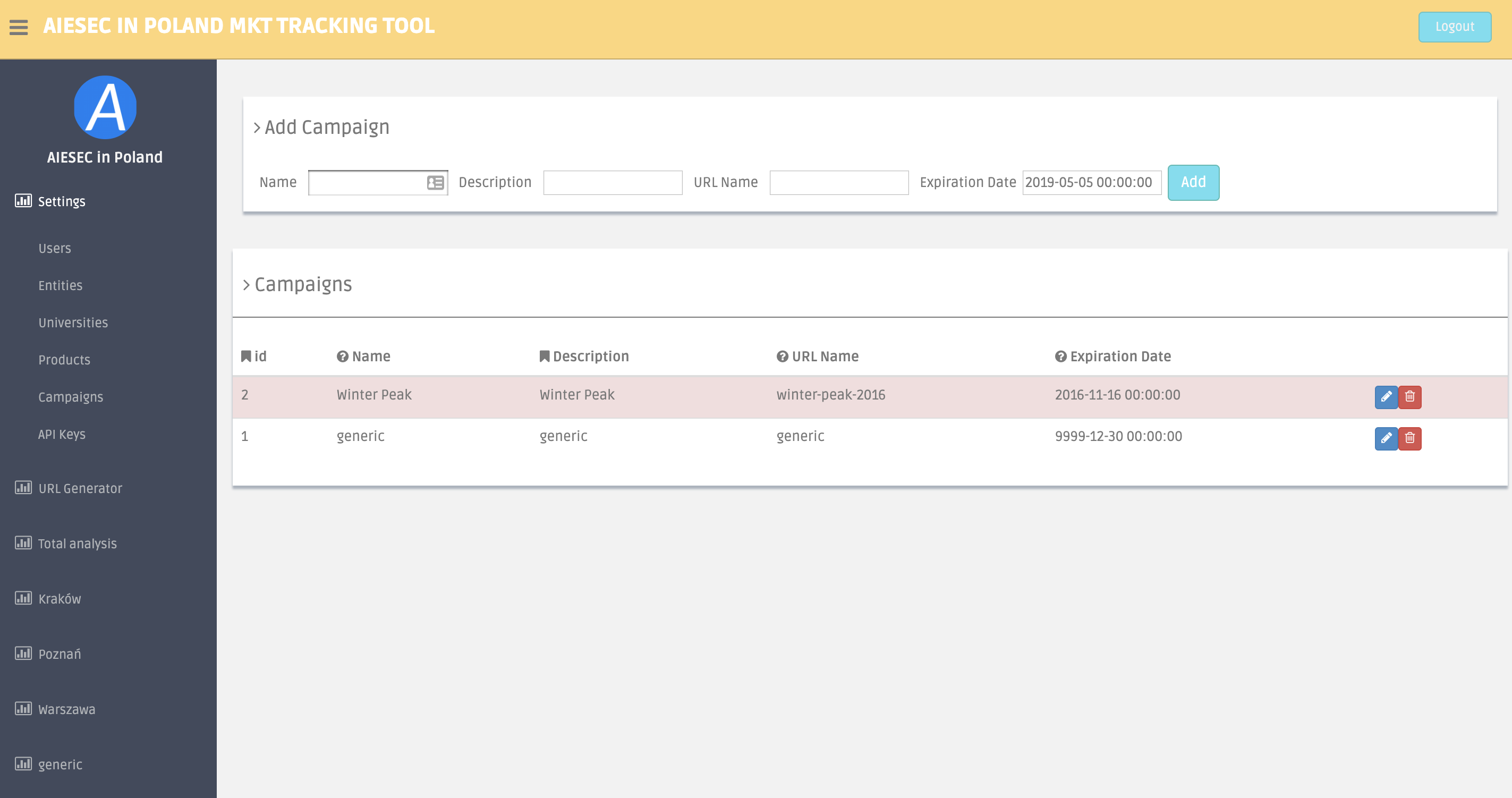
Aby usunąć produkt należy kliknąć w menu bocznym *Settings* a następnie *Products*. Na liście produktów należy nacisnąć przycisk *Usuń*.



Rysunek 19 Usuwanie produktów organizacji

* + 1. D-1 Uruchamianie kampanii marketingowej

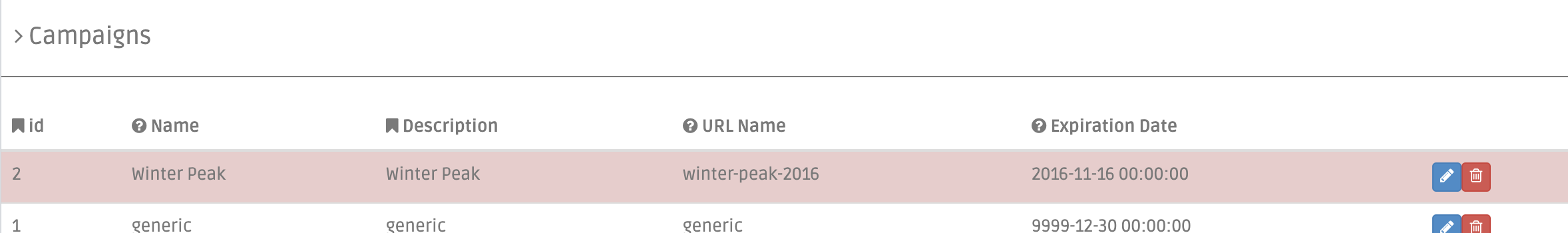
W celu uruchomienia nowej kampanii marketingowej należy z menu bocznego wybrać *Settings* a następnie *Campaigns*. W formularzy w górnej części strony należy podać informacje na temat nowej kampanii takie jak: nazwa, opis, uproszczona nazwa oraz datę końcową czasu trwania kampanii a następnie nacisnąć przycisk *Add*.



Rysunek 20 Uruchamianie kampanii marketingowej

* + 1. D-2 Archiwizacja kampanii marketingowej

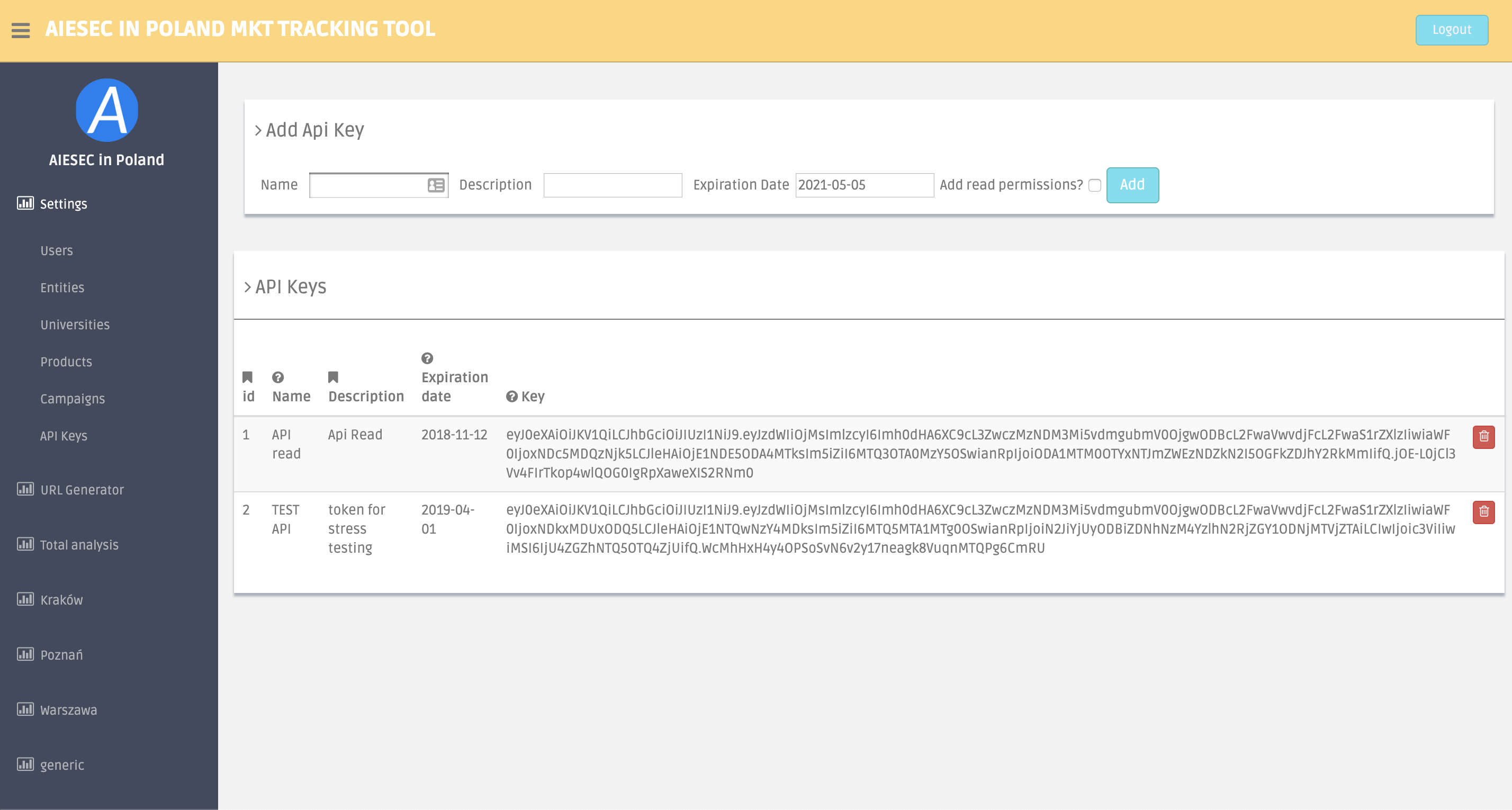
Aby zarchiwizować kampanię marketingową należy z menu bocznego należy wybrać *Settings* a następnie *Campaigns.* Na liście aktywnych kampanii należy wybrać kampanię do usunięcia a następnie kliknąć czerwony przycisk *Usuń*.



Rysunek 21 Archiwizacja kampanii marketingowej

* + 1. E-1 Komunikacja z formularzami rejestracyjnymi na stronie www

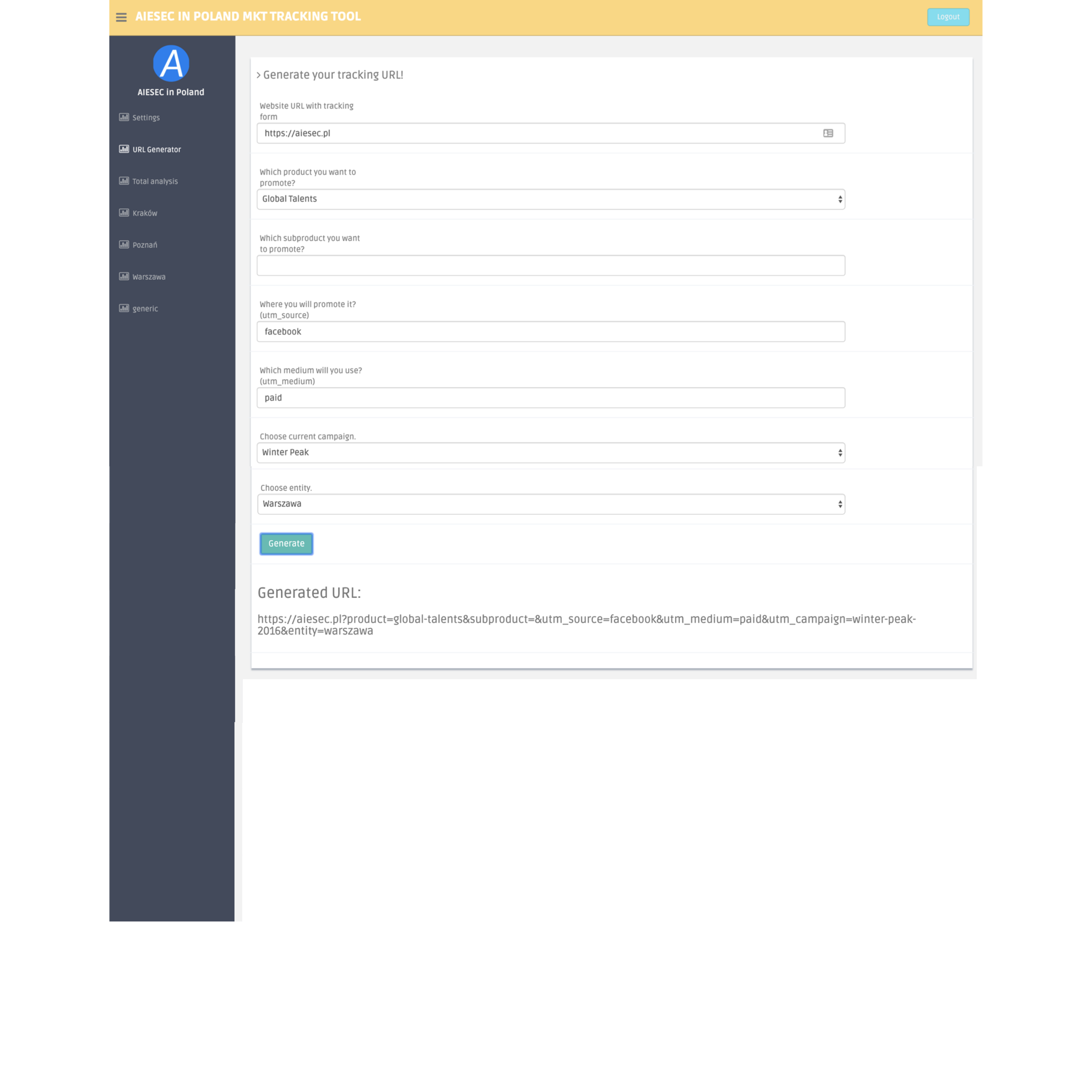
Komunikacja pomiędzy formularzem na stronie www a systemem analiz jest przeprowadzona za pomocą odpowiednich zapytań http z wykorzystaniem statycznych kluczy API umożliwiających zapis nowych danych. Aby wygenerować nowy statyczny klucz należy z menu bocznego wybrać *Settings* a następnie *API Keys.* W celu wygenerowania nowego klucza należy podać nazwę klucza, opis z informacją gdzie będzie używany oraz termin ważności. Opcjonalnie można wygenerować klucz z uprawnieniami do odczytu danych ale ze względów bezpieczeństwa należy ograniczyć takie klucze bo bezwzględnego minimum.



Rysunek 22 Komunikacja z formularzami rejestracyjnymi na stronie www

* + 1. E-2 Generowanie linków promocyjnych

Aby wygenerować nowy link promocyjny z odpowiednimi tagami UTM należy z menu bocznego wybrać *URL Generator*. Następnie w formularzu podać wszystkie wymagane dane potrzebne do identyfikacji działań marketingowych w ramach których dany link będzie używany. Po wypełnieniu wymaganych pól należy nacisnąć przycisk *Generate* a poniżej pojawi się wygenerowany adres. Należy pamiętać że system będzie działał poprawnie tylko gdy pod wygenerowanym adresem znajduje się formularz połączony z systemem.



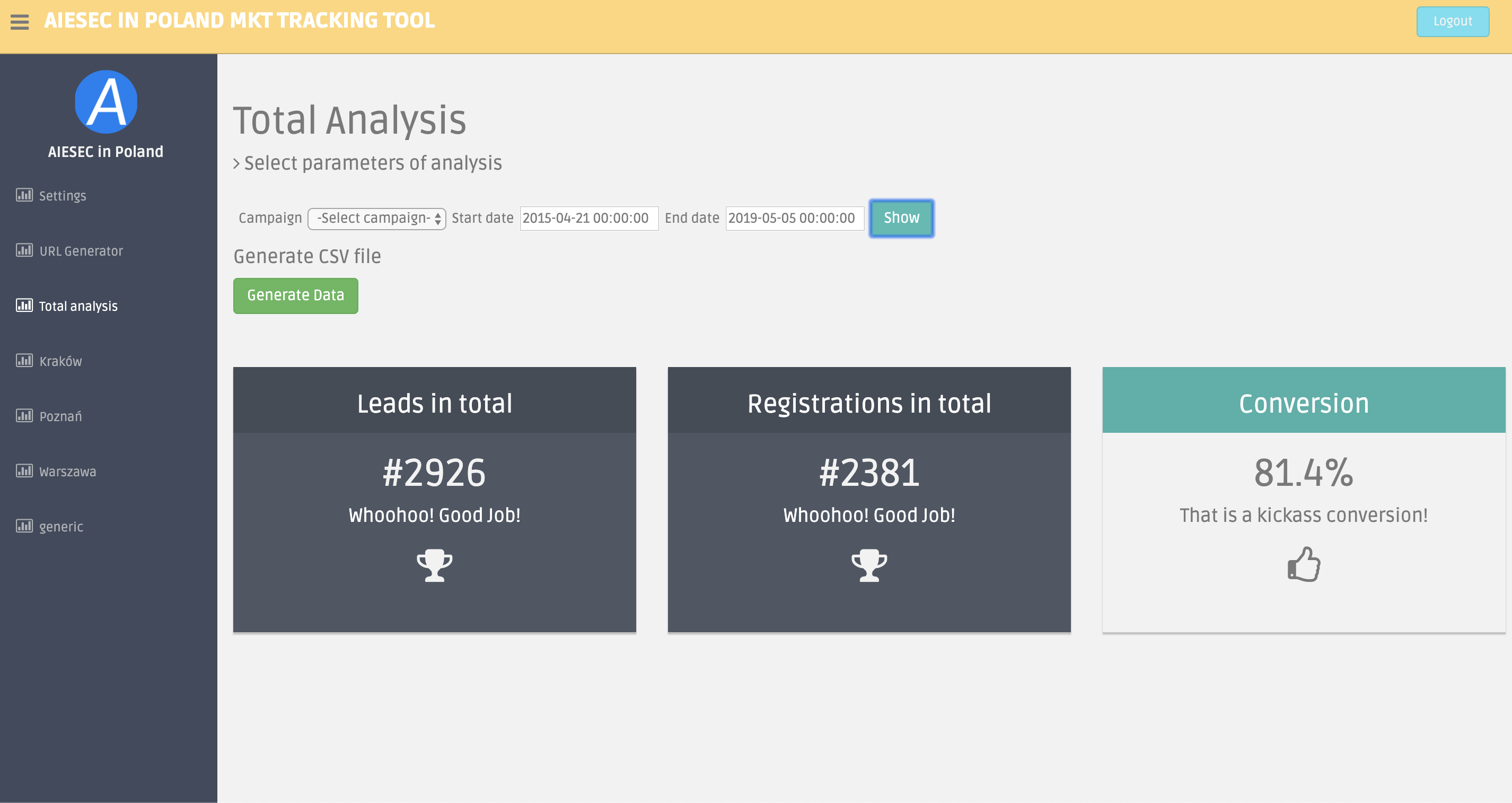
Rysunek 23 Generowanie linków promocyjnych

* + 1. E-3 Prezentacja analiz marketingowych

Analizy marketingowe podzielone są na dwie główne części:

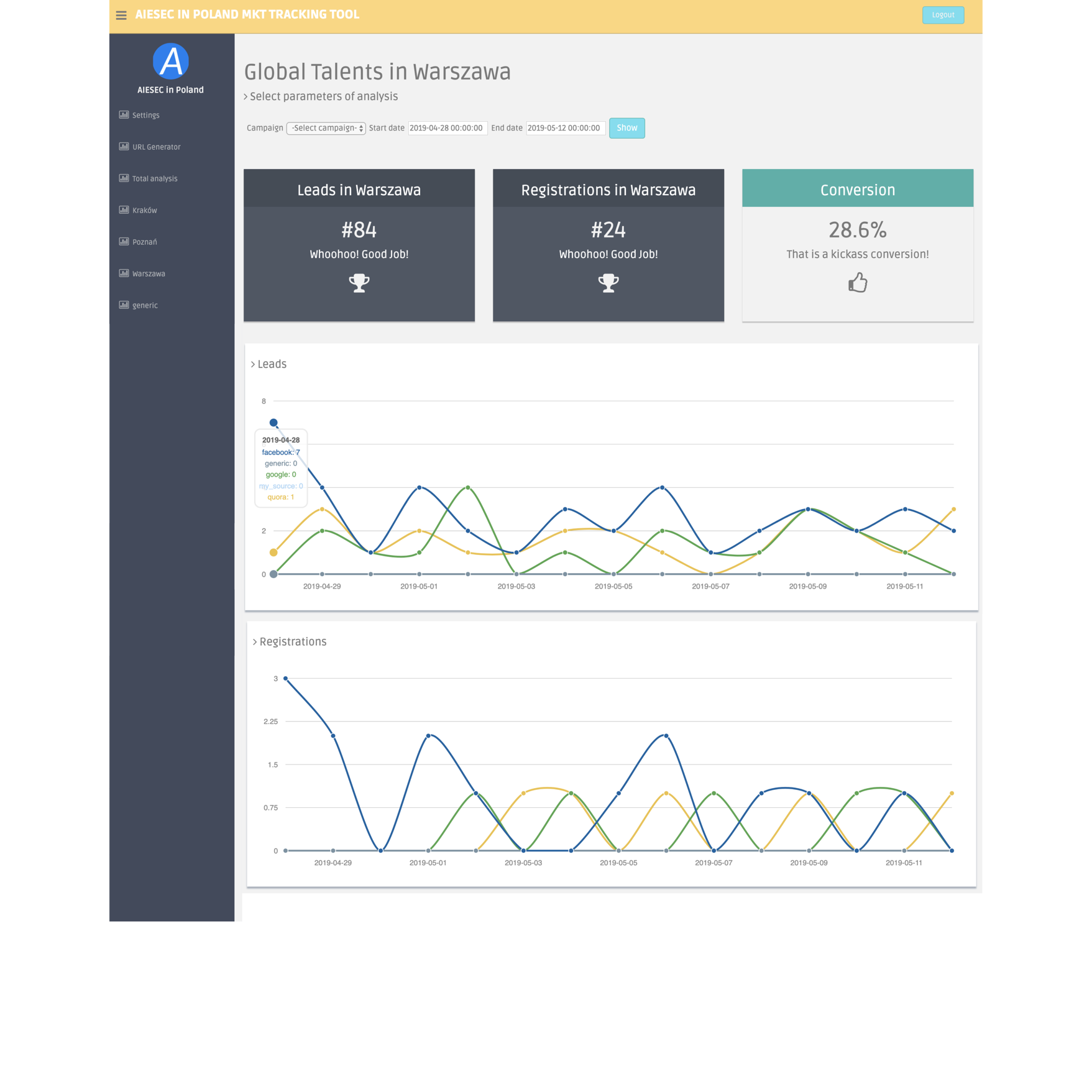
* Analizę ogólną - Rysunek 24 Prezentacja analiz marketingowych - analiza zbiorcza
* Analizę szczegółową dla każdego oddziału lokalnego z podziałem na produkty - Rysunek 25 Prezentacja analiz marketingowych - analiza szczegółowa

Analiza ogólna pojawia się jako główny ekran systemu oraz jest dostępna z menu bocznego po kliknięciu w *Total Analysis*. Zawiera podstawowe informacje o ilości leadów, klientów oraz konwersji. W górnej części ekranu umiejscowiony jest również formularz służący do filtracji wyników na podstawie kampanii marketingowej oraz przedziału czasowego. Możliwe jest również wygenerowanie pliku CSV ze szczegółowymi danymi statystycznymi.



Rysunek 24 Prezentacja analiz marketingowych - analiza zbiorcza

Aby wyświetlić analizę szczegółową należy z menu bocznego wybrać oddział lokalny a następnie produkt prowadzony przez oddział na przykład „Global Talents” w Warszawie. Na ekranie zaprezentowane są podstawowe statystyki podobnie jak na poprzednim ekranie (Rysunek 24 Prezentacja analiz marketingowych - analiza zbiorcza) oraz dwa wykresy. Wykres pierwszy prezentujący zmianę ilości leadów w czasie z podziałem na źródła oraz wykres drugi prezentujący zmianę ilości rejestracji klientów z podziałem na źródła.



Rysunek 25 Prezentacja analiz marketingowych - analiza szczegółowa

1. Analiza wydajności systemu
   1. Cel i hipoteza

Przedmiotem przeprowadzonego badania jest symulacja równoczesnych odwiedzin systemu przez użytkowników oraz analiza jak system się zachowa w sytuacji większego obciążenia.

Głównym założeniem jest, że system będzie potrzebował więcej czasu na reakcję w raz ze wzrostem użytkowników. Ponadto w związku z tym że infrastruktura na której został zainstalowany system jest dosyć podstawowa, możliwe jest że część użytkowników nie zostanie poprawnie obsłużona.

* 1. Infrastruktura testowa

System został zainstalowany na infrastrukturze firmy OVH z planem: Serwer VPS 2016 - SSD 2. Parametry wykupionego hostingu są następujące:

* Procesor Intel Core Haswell minimum 2GHz
* Detykowany jeden rdzeń procesora
* 4 GB RAM
* Dysk: 40GB
* Przepustowość: 100 Mbps - Nielimitowany transfer
* System operacyjny: Ubuntu 16.04 Server
* Serwer http: nginx/1.10.3
* Wersja silnika PHP: 7.0.33-0ubuntu0.16.04.4
* Baza danych: PostgreSQL 9.4.10

Do analizy wydajności systemu zostało użyte narzędzie wrk (<https://github.com/wg/wrk)> uruchamiane na prywatnym komputerze: MacBook Pro z procesorem 3,1 GHz Intel Core i5 oraz pamięcią RAM 16 GB.

Narzędzie wrk pozwala na zrównoleglone wysyłanie zapytań HTTP do serwera oraz analizę odpowiedzi. Informacje zbierane to:

* Czas oczekiwania na odpowiedź serwera (ang.: Latency)
* Ilość zapytań na sekundę
* Ilość poprawnych odpowiedzi
* Ilość odpowiedzi z błędem podzielonych na rodzaje (connection, read, write, status, timeout)
* Rozmiar danych odebranych
  1. Metoda badawcza

Badaniu zostały poddane najbardziej wrażliwe na przeciążenie punkty końcowe (ang.: endpoints) systemu:

* POST /api/v1/leads – zapisanie nowego lead’a
* GET /api/v1/leads? date\_from={date}&date\_to={date} – pobranie listy leadów
* POST /api/v1/leads/leads-statistics – pobranie statystyki leadów do wyświetlenia na wykresie
* POST /api/v1/customers – zapisanie nowego klienta
* GET /api/v1/customers?date\_from={date}&date\_to={date} – pobranie listy klientów
* POST /api/v1/customers/customers-statistics – pobranie statystyki klientów do wyświetlenia na wykresie

Badanie każdego punktu końcowego zostało podzielone na 11 kroków trwających 30 sekund każdy. Co krok zwiększana była ilość równoległych połączeń z systemem od 1 do 50. Ustawiony został również górny limit czasu przeznaczonego na pojedynczą odpowiedź na 10 sekund.

* 1. Analiza wyników badań
     1. Zapisanie nowego lead’a

Wykres 1. Czas oczekiwania na odpowiedź serwera dla punktu końcowego: POST /api/v1/leads

Źródło: opracowanie własne

Analizując Wykres 1. Czas oczekiwania na odpowiedź serwera dla punktu końcowego: POST /api/v1/leadsmożna zauważyć że zgodnie z przewidywaniami wraz ze wzrostem liczby równoległych połączeń rośnie czas oczekiwania na odpowiedź serwera. Minimalna wartość przy pojedynczym połączeniu wyniosła 0,06 sekundy co jest praktycznie niezauważalne dla użytkownika. Maksymalny wynik przy 50 równoległych połączeniach wyniósł 2,3 sekundy. Średni czas oczekiwania przy 50 połączeniach wyniósł 1,3 sekundy.

Wykres 2. Ilość zapytań na sekundę dla punktu końcowego: POST /api/v1/leads

Źródło: opracowanie własne

Minimalna ilość zapytań na sekundę przy jednym połączeniu wyniosła 6 a maksymalna 20. Co ciekawe maksymalna ilość zapytań na sekundę nie rosła dla połączeń od 5 do 50. Średnia ilość zapytań dla jednego połączenia wyniosła 15,4 a dla 50 połączeń 40,2. Zatrzymanie wzrostu maksymalnej ilości zapytań najprawdopodobniej wynika z przepustowości serwera.

Wykres 3. Ilość poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/leads

Źródło: opracowanie własne

Na Wykresie 3. Możemy zaobserwować jaka była zbiorcza ilość wszystkich poprawnych odpowiedzi z serwera wraz ze wzrostem równoległych połączeń. Maksymalny wynik został osiągnięty przy 20 połączeniach o wartości 1218 odpowiedzi. Widać również tak samo jak na Wykresie 2. Zatrzymanie wzrostu odpowiedzi dla ponad 10 połączeń. Wszystkie zapytania dostały odpowiedź zakończoną sukcesem. Dla pojedynczego połączenia wyniosła 470 a przy 50 połączeniach 1170. W sumie podczas badania wykonano 12304 zapytań.

Wykres 4. Procent poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/leads

Źródło: opracowanie własne

Wykres 5. Ilość odpowiedzi błędnych dla punktu końcowego: POST /api/v1/leads

Źródło: opracowanie własne

Wykres 6. Procent odpowiedzi błędnych dla punktu końcowego: POST /api/v1/leads

Źródło: opracowanie własne

Wykres 7. Rozmiar danych na sekundę odebranych dla punktu końcowego: POST /api/v1/leads

Źródło: opracowanie własne

Ilość danych odebrana z serwera jest bezpośrednio skorelowana z ilością zapytań. Średnia przepustowość utrzymywała się na poziomie 24,1 KB/s.

* + 1. Pobranie listy lead’ów

Wykres 8. Czas oczekiwania na odpowiedź serwera dla punktu końcowego: GET /api/v1/leads

Źródło: opracowanie własne

Wykres 9. Ilość zapytań na sekundę dla punktu końcowego: GET /api/v1/leads

Źródło: opracowanie własne

Wykres 10. Ilość poprawnych odpowiedzi dla punktu końcowego: GET /api/v1/leads

Źródło: opracowanie własne

Wykres 11. Procent poprawnych odpowiedzi dla punktu końcowego: GET /api/v1/leads

Źródło: opracowanie własne

Wykres 12. Ilość odpowiedzi błędnych dla punktu końcowego: GET /api/v1/leads

Źródło: opracowanie własne

Wykres 13. Procent odpowiedzi błędnych dla punktu końcowego: GET /api/v1/leads

Źródło: opracowanie własne

Wykres 14. Rozmiar danych odebranych dla punktu końcowego: GET /api/v1/leads

Źródło: opracowanie własne

* + 1. Pobranie statystyki lead’ów

Wykres 15. Czas oczekiwania na odpowiedź serwera dla punktu końcowego: POST /api/v1/leads/leads-statistics

Źródło: opracowanie własne

Wykres 16. Ilość zapytań na sekundę dla punktu końcowego: POST /api/v1/leads/leads-statistics

Źródło: opracowanie własne

Wykres 17. Ilość poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/leads/leads-statistics

Źródło: opracowanie własne

Wykres 18. Procent poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/leads/leads-statistics

Źródło: opracowanie własne

Wykres 19. Ilość odpowiedzi błędnych dla punktu końcowego: POST /api/v1/leads/leads-statistics

Źródło: opracowanie własne

Wykres 20. Procent odpowiedzi błędnych dla punktu końcowego: POST /api/v1/leads/leads-statistics

Źródło: opracowanie własne

Wykres 21. Rozmiar danych odebranych dla punktu końcowego: POST /api/v1/leads/leads-statistics

Źródło: opracowanie własne

* + 1. Zapisanie nowego klienta

Wykres 22. Czas oczekiwania na odpowiedź serwera dla punktu końcowego: POST /api/v1/customers

Źródło: opracowanie własne

Wykres 23. Ilość zapytań na sekundę dla punktu końcowego: POST /api/v1/customers

Źródło: opracowanie własne

Wykres 24.Ilość poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/customers

Źródło: opracowanie własne

Wykres 25. Procent poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/customers

Źródło: opracowanie własne

Wykres 26. Ilość odpowiedzi błędnych dla punktu końcowego: POST /api/v1/customers

Źródło: opracowanie własne

Wykres 27. Procent odpowiedzi błędnych dla punktu końcowego: POST /api/v1/customers

Źródło: opracowanie własne

Wykres 28. Rozmiar danych odebranych dla punktu końcowego: POST /api/v1/customers

Źródło: opracowanie własne

* + 1. Pobranie listy klientów

Wykres 29. Czas oczekiwania na odpowiedź serwera dla punktu końcowego: GET /api/v1/customers

Źródło: opracowanie własne

Wykres 30. Ilość zapytań na sekundę dla punktu końcowego: GET /api/v1/customers

Źródło: opracowanie własne

Wykres 31. Ilość poprawnych odpowiedzi dla punktu końcowego: GET /api/v1/customers

Źródło: opracowanie własne

Wykres 32. Procent poprawnych odpowiedzi dla punktu końcowego: GET /api/v1/customers

Źródło: opracowanie własne

Wykres 33. Ilość odpowiedzi błędnych dla punktu końcowego: GET /api/v1/customers

Źródło: opracowanie własne

Wykres 34. Procent odpowiedzi błędnych dla punktu końcowego: GET /api/v1/customers

Źródło: opracowanie własne

Wykres 35. Rozmiar danych odebranych dla punktu końcowego: GET /api/v1/customers

Źródło: opracowanie własne

* + 1. Pobranie statystyki klientów

Wykres 36. Czas oczekiwania na odpowiedź serwera dla punktu końcowego: POST /api/v1/customers/customers-statistics

Źródło: opracowanie własne

Wykres 37. Ilość zapytań na sekundę dla punktu końcowego: POST /api/v1/customers/customers-statistics

Źródło: opracowanie własne

Wykres 38. Ilość poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/customers/customers-statistics

Źródło: opracowanie własne

Wykres 39. Procent poprawnych odpowiedzi dla punktu końcowego: POST /api/v1/customers/customers-statistics

Źródło: opracowanie własne

Wykres 40. Ilość odpowiedzi błędnych dla punktu końcowego: POST /api/v1/customers/customers-statistics

Źródło: opracowanie własne

Wykres 41. Procent odpowiedzi błędnych dla punktu końcowego: POST /api/v1/customers/customers-statistics

Źródło: opracowanie własne

Wykres 42. Rozmiar danych odebranych dla punktu końcowego: POST /api/v1/customers/customers-statistics

Źródło: opracowanie własne

1. Dokumentacja API

Komunikacja z systemem możliwa jest poprzez udostępniony interfejs którego dokumentacja znajduje się poniżej. Odpowiedzi z serwera są zgodne ze specyfikacją Json API. (3)

Dokumentacja dostępna jest również w ramach systemu pod adresem: <http://vps334372.ovh.net/docs/api_documentation.htm>

Zapytania mogą zwracać następujące wyjątki:

* DATE\_NOT\_VALID
* SLUG\_NOT\_UNIQUE
* OBJECT\_DOES\_NOT\_EXIST
* INVALID\_PAYLOAD
* INVALID\_CREDENTIALS
* TOKEN\_CREATION\_FAILED
* USER\_NOT\_FOUND
* RESOURCE\_NOT\_AVAILABLE
* WRITE\_FILE\_ERROR
* DATABASE\_ERROR
* MIGRATIONS\_ERROR
* CREATE\_PERMISSIONS\_FAILED
* OBJECT\_DUPLICATED
* OBJECT\_CREATION\_FAILED
* EMAIL\_NOT\_UNIQUE
* INTERNAL\_VIOLATION
* ID\_NOT\_UNIQUE
  1. Klucze API
     1. Tworzenie klucza

Generuje nowy klucz API w formie tokenu dla formularza. Uprawnienia odczytu muszą być urzywane z wysoką rozwagą.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/api-keys | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "Forms",

"description": "API key for forms",

"expiration\_date": "25-04-2202",

"read\_permissions": "true"

}

Przykładowa odpowiedź:

{

"name": "Forms",

"description": "API key for forms",

"expiration\_date": "25-04-2202",

"key": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjIsImlzcyI6Imh0dHA6XC9cL2FwaS5tYXN0ZXItdG9vbC5kZXZcL2FwaVwvdjFcL2FwaS1rZXlzIiwiaWF0IjoxNDc3ODYwMzkzLCJleHAiOjE0Nzc4ODU1OTMsIm5iZiI6MTQ3Nzg2MDM5MywianRpIjoiY2Q4ZGE4ODljNDlkZTJkNTY2MjU1NjZiM2U5OTE4ZjgifQ.aZdchMD5\_1gxKtGRupL-ZhOIMn2fhf0eqfGaBxYPglU",

"updated\_at": "2016-10-30 20:46:33",

"created\_at": "2016-10-30 20:46:33",

"id": 2

}

* + 1. Usuwanie klucza

Usuwa i dezaktualizuje klucz o danym numerze id.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/api/v1/api-keys/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |

Zwraca status 200 jeśli ukończono z powodzeniem.

* + 1. Pobieranie listy kluczy

Pobiera listę wszystkich kluczy które są obecnie dostępne.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/api/v1/api-keys | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": 2 ,

"key": "eyJ0eXAiOiJKLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjIsImlzcyI6Imh0dHA6XC9cL2FwaS5tYXN0ZXItdG9vbC5kZXZcLFcL2FwaS1rZXlzIiwiaWF0IjoxNDc3ODYwMzkzLCJleHAiOjE0Nzc4ODU1OTMsIm5iZiI6MTQ3Nzg2MDM5joiY2Q4ZGE4ODljNDlkZTJkNTY2MjU1NjZiM2U5OTE4ZjgifQ.aZdchMD5\_1gxKtGRupL-ZhOIMn2fhf0eqfGaBxYPglU",

"name": "Forms",

"description": "API key for forms",

"expiration\_date": "25-04-2202",

"created\_at": "2016-10-30 20:46:33",

"updated\_at": "2016-10-30 20:46:33",

"deleted\_at": null

}

]

* 1. Autoryzacja
     1. Informacje o użytkowniku

Pobiera informacje o zalogowanym użytkowniku na podstawie tokenu.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/api/v1/authenticate/user | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

{

"user": {

"id": 1,

"name": "Kris",

"email": "kris@kris.pl",

"created\_at": "2016-10-14 18:51:42",

"updated\_at": "2016-10-14 18:51:42",

"roles": [

{

"id": 1,

"name": "Admin",

"slug": "admin",

"description": "manage administration privileges",

"created\_at": "2016-10-14 18:51:42",

"updated\_at": "2016-10-14 18:51:42",

"pivot": {

"user\_id": 1,

"role\_id": 1,

"created\_at": "2016-10-14 18:51:42",

"updated\_at": "2016-10-14 18:51:42"

}

}

]

},

"roles": {

"1": "admin"

}

}

* + 1. Logowanie nowego użytkownika

Przeprowadza autentykację nowego użytkownika oraz zwraca token autoryzacyjny.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/authenticate/user | |
| Nagłówki | **Content-Type** | application/json |

Ciało zapytania:

{

"email": "username@domain.pl",

"password": "secret"

}

Przykładowa odpowiedź:

{

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEsImlzcyI6Imh0dHA6XC9cL2FwaS5tYXN0ZXItdG9vbC5kZXZcL2FwaVwvdjFcL2F1dGhlbnRpY2F0ZSIsImlhdCI6MTQ3Nzg2MTc0MSwiZXhwIjoxNDc3ODg2OTQxLCJuYmYiOjE0Nzc4NjE3NDEsImp0aSI6IjEwZmExMTBkYThhYTE0ZGI2YjlkNDBiMGEwODRkNmQwIn0.NuZhAqSSeBs6kx6XbSq8DiZaFat0xZCG5yqnb9fhJEg"

}

* 1. Zarządzanie kampaniami
     1. Nowa kampania

Tworzy nową kampanię na podstawie przekazanych danych.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/ api/v1/campaigns | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "My super campaign",

"description": "This is description of super campaign",

"slug": "my-campaign-2",

"expires\_on": "2016-12-22 00:00:00"

}

Przykładowa odpowiedź:

{

"name": "My super campaign",

"description": "This is description of super campaign",

"slug": "my-campaign-1",

"expires\_on": "2016-12-22 00:00:00",

"updated\_at": "2016-10-30 21:22:48",

"created\_at": "2016-10-30 21:22:48",

"id": 2

}

* + 1. Usuwanie kampanii

Usuwa kampanię o danym numerze id. Zwraca status 200 w wypadku udanej operacji.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/ api/v1/campaigns/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

* + 1. Pobieranie listy kampanii

Pobiera listę otwartych kampanii.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/ api/v1/campaigns | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": 1,

"name": "generic",

"description": "generic",

"slug": "generic",

"expires\_on": "9999-12-30 00:00:00",

"created\_at": "2016-10-14 18:51:35",

"updated\_at": "2016-10-14 18:51:35"

},

...

{

"id": 3,

"name": "My super campaign",

"description": "This is description of super campaign",

"slug": "my-campaign-2",

"expires\_on": "2016-12-22 00:00:00",

"created\_at": "2016-10-30 21:26:30",

"updated\_at": "2016-10-30 21:26:30"

}

]

* + 1. Aktualizacja kampanii

Aktualizuje istniejącą kampanię.

|  |  |  |
| --- | --- | --- |
| Typ | PUT | |
| URL | {url}/api/v1/campaigns{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "My super campaign",

"description": "This is description of super campaign",

"slug": "my-campaign-2",

"expires\_on": "2016-12-22 00:00:00"

}

Przykładowa odpowiedź:

{

"name": "My super campaign",

"description": "This is description of super campaign",

"slug": "my-campaign-1",

"expires\_on": "2016-12-22 00:00:00",

"updated\_at": "2016-10-30 21:22:48",

"created\_at": "2016-10-30 21:22:48",

"id": 2

}

* + 1. Pobieranie informacji o kampanii

Pobiera informacje o kampanii na podstawie numeru id.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/api/v1/campaigns{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

{

"id": 1,

"name": "generic",

"description": "generic",

"slug": "generic",

"expires\_on": "9999-12-30 00:00:00",

"created\_at": "2016-10-14 18:51:35",

"updated\_at": "2016-10-14 18:51:35"

}

* 1. Zarządzanie klientami
     1. Nowy klient

Zapisuje informacje o nowym kliencie dla danego id leadu.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/customers | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"lead\_id": "a9aa445312aab6f75b4c98c455b8f820",

"email": "aa@bbab.pl",

"entity\_id": 1,

"fields": {

"name": "Jack",

"surname": "Doe",

"phone\_number": "+998877"

}

}

Przykładowa odpowiedź:

{

"lead\_id": "a9aa445312aab6f75b4c98c455b8f820",

"email": "aa@bbab.pl",

"entity\_id": 1,

"fields": "\"name\"=>\"Jack\",\"surname\"=>\"Doe\",\"phone\_number\"=>\"+998877\"",

"updated\_at": "2016-11-14 20:46:56",

"created\_at": "2016-11-14 20:46:56",

"id": 5

}

* + 1. Lista klientów

Pobiera listę klientów pozyskanych w danym okresie czasu. Jeśli parametry date\_from oraz date\_to nie zostały podane zwracana jest lista wszystkich klientów.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/api/v1/customers?date\_from={date}&date\_to={date} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": 2,

"lead\_id": "a9aa445312aab6f75b4c98c455b8f820",

"email": "aa@bbba.pl",

"entity\_id": 1,

"fields": {

"name": "Kris",

"source": "friend",

"surname": "Jack"

},

"created\_at": "2016-11-14 20:44:08",

"updated\_at": "2016-11-14 20:44:08",

"lead": {

"id": "a9aa445312aab6f75b4c98c455b8f820",

"utm\_source": "generic",

"utm\_campaign\_id": 1,

"utm\_medium": "generic",

"utm\_content": "",

"utm\_term": "",

"entity\_id": 1,

"product\_id": 1,

"subproduct": "n\_d",

"created\_at": "2016-11-12 22:04:47",

"updated\_at": "2016-11-12 22:04:47"

}

}

]

* + 1. Liczenie klientów

Zwraca ilość klientów zarejestrowanych w danym okresie czasu oraz odpowiadających określonym parametrom. Pola "entity", "product" oraz "utm\_campaign" są optionalne.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/ api/v1/customers/count | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"date\_from": "2016-09-30",

"date\_to": "2016-11-15",

"entity": "generic",

"product": "generic",

"utm\_campaign": "generic"

}

Przykładowa odpowiedź:

[

{

"count": 3

}

]

* + 1. Statystyki klientów

Zwraca statystykę klientów zarejestrowanych w danym okresie czasu oraz odpowiadających określonym parametrom. Pola "entity" oraz "utm\_campaign" są optionalne.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/ api/v1/customers/customers-statistics | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"date\_from": "2016-11-12",

"date\_to": "2016-11-15",

"product": "generic",

"entity": "generic",

"utm\_campaign": "generic"

}

Przykładowa odpowiedź:

[

{

"date": "2016-11-12",

"generic": 0

},

{

"date": "2016-11-13",

"generic": 0

},

{

"date": "2016-11-14",

"generic": 3

},

{

"date": "2016-11-15",

"generic": 0

}

]

* + 1. Usuwanie klienta

Usuwa klienta o danym numerze id. Zwraca status 200 w wypadku udanej operacji.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/ api/v1/customers/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

* + 1. Generowanie pliku CSV

Generuje plik CSV z informacjami o kandydatach.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/ api/v1/customers/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"date\_from": "2016-10-31",

"date\_to": "2016-11-14",

"utm\_campaign": "generic"

}

* 1. Zarządzanie oddziałami
     1. Nowy oddział

Zapisuje informacje o nowym oddziale.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/entities | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "Kraków",

"expa\_name": "KRAKOW",

"expa\_id": "1000",

"slug": "krakow"

}

Przykładowa odpowiedź:

{

"name": "Kraków",

"expa\_name": "KRAKOW",

"expa\_id": "1000",

"slug": "krakow",

"updated\_at": "2016-11-14 21:24:14",

"created\_at": "2016-11-14 21:24:14",

"id": 2

}

* + 1. Usuwanie oddziału

Usuwa oddział o danym numerze id. Zwraca status 200 w wypadku udanej operacji.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/ api/v1/entities/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

* + 1. Lista oddziałów

Pobiera listę oddziałów w organizacji.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/api/v1/entities | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": 1,

"name": "generic",

"expa\_name": "KRAKOW",

"expa\_id": 1510,

"slug": "generic",

"created\_at": "2016-11-12 22:02:03",

"updated\_at": "2016-11-12 22:02:03"

},

{

"id": 2,

"name": "Kraków",

"expa\_name": "KRAKOW",

"expa\_id": 1000,

"slug": "krakow",

"created\_at": "2016-11-14 21:24:14",

"updated\_at": "2016-11-14 21:24:14"

}

]

* + 1. Aktualizacja oddziału

Aktualizuje informacje o oddziale z danym numerem id.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/entities{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "Poznań",

"slug": "poznan",

"expa\_id": "111",

"expa\_name": "POZNAN"

}

Przykładowa odpowiedź:

{

"id": 2,

"name": "Poznań",

"expa\_name": "POZNAN",

"expa\_id": "111",

"slug": "poznan",

"created\_at": "2016-11-14 21:24:14",

"updated\_at": "2016-11-14 21:29:53"

}

* 1. Zarządzanie leadami sprzedażowymi
     1. Nowy lead

Zapisuje informacje o nowym leadzie na podstawie danych pobranych ze śledzącego adresu URL.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/leads | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"id": "5a551d504775546b473d813beb09816d",

"utm\_source": "my-source",

"utm\_campaign": "my-campaign",

"utm\_medium": "paid",

"utm\_content": "my-content",

"utm\_term": "my-term",

"entity": "poznan",

"product": "gc"

}

Przykładowa odpowiedź:

{

"id": "5a551d504775546b473d813beb09816d",

"utm\_source": "my\_source",

"utm\_campaign\_id": 1,

"utm\_medium": "paid",

"utm\_content": "my\_content",

"utm\_term": "my\_term",

"entity\_id": 2,

"product\_id": 1,

"subproduct": "",

"updated\_at": "2016-11-14 21:41:39",

"created\_at": "2016-11-14 21:41:39"

}

* + 1. Usuwanie leada

Usuwa lead o danym numerze id. Zwraca status 200 w wypadku udanej operacji.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/ api/v1/leads/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

* + 1. Pobieranie leadów

Pobiera listę wszystkich leadów.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/ api/v1/leads/ | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": "a9aa445312aab6f75b4c98c455b8f820",

"utm\_source": "generic",

"utm\_campaign\_id": 1,

"utm\_medium": "generic",

"utm\_content": "",

"utm\_term": "",

"entity\_id": 1,

"product\_id": 1,

"subproduct": "n\_d",

"created\_at": "2016-11-12 22:04:47",

"updated\_at": "2016-11-12 22:04:47"

},

{

"id": "7162b2baf14068c7b61aed199123db9a",

"utm\_source": "generic",

"utm\_campaign\_id": 1,

"utm\_medium": "generic",

"utm\_content": "",

"utm\_term": "",

"entity\_id": 1,

"product\_id": 1,

"subproduct": "n\_d",

"created\_at": "2016-11-14 22:03:44",

"updated\_at": "2016-11-14 22:03:44"

}

]

* + 1. Liczenie leadów

Zwraca ilość leadów zapisanych w danym okresie czasu oraz odpowiadających określonym parametrom. Pola "entity", "product" oraz "utm\_campaign" są optionalne.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/ api/v1/leads/count | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"date\_from": "2016-09-30",

"date\_to": "2016-11-15",

"entity": "generic",

"product": "generic",

"utm\_campaign": "generic"

}

Przykładowa odpowiedź:

[

{

"count": 3

}

]

* + 1. Statystyki leadów

Zwraca statystykę leadów zapisanych w danym okresie czasu oraz odpowiadających określonym parametrom. Pola "entity" oraz "utm\_campaign" są optionalne.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/ api/v1/leads/leads-statistics | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"date\_from": "2016-11-12",

"date\_to": "2016-11-15",

"product": "generic",

"entity": "generic",

"utm\_campaign": "generic"

}

Przykładowa odpowiedź:

[

{

"date": "2016-11-12",

"generic": 0

},

{

"date": "2016-11-13",

"generic": 0

},

{

"date": "2016-11-14",

"generic": 3

},

{

"date": "2016-11-15",

"generic": 0

}

]

* 1. Zarządzanie produktami
     1. Nowy produkt

Zapisuje informacje o nowym produkcie. Obecnie pod produkty nie posiadają bezpośredniej reprezentacji w bazie danych tylko są elementem typu string w tabeli leads.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/products | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "Global Citizen",

"description": "Global Citizen",

"slug": "gc"

}

Przykładowa odpowiedź:

{

"name": "Global Citizen",

"description": "Global Citizen",

"slug": "gc",

"updated\_at": "2016-11-14 22:27:23",

"created\_at": "2016-11-14 22:27:23",

"id": 2

}

* + 1. Usuwanie produktu

Usuwa produkt o danym numerze id. Zwraca status 200 w wypadku udanej operacji.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/ api/v1/products/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

* + 1. Pobieranie produktów

Pobiera listę wszystkich produktów.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/ api/v1/products/ | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": 1,

"name": "generic",

"description": "generic",

"slug": "generic",

"created\_at": "2016-11-12 22:02:03",

"updated\_at": "2016-11-12 22:02:03"

},

{

"id": 2,

"name": "Global Citizen",

"description": "Global Citizen",

"slug": "gc",

"created\_at": "2016-11-14 22:27:23",

"updated\_at": "2016-11-14 22:27:23"

}

]

* + 1. Pobieranie produktu

Pobiera informacje o produkcie z danym numerem id.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/ api/v1/products/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

{

"id": 1,

"name": "generic",

"description": "generic",

"slug": "generic",

"created\_at": "2016-11-12 22:02:03",

"updated\_at": "2016-11-12 22:02:03"

}

* + 1. Aktualizacja produktu

Aktualizuje informacje o produkcie z danym numerem id.

|  |  |  |
| --- | --- | --- |
| Typ | PUT | |
| URL | {url}/ api/v1/products/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "Global Talents",

"slug": "global-talents",

"description": "Global Talents"

}

Przykładowa odpowiedź:

{

"id": 2,

"name": "Global Talents",

"description": "Global Talents",

"slug": "global-talents",

"created\_at": "2016-11-14 22:27:23",

"updated\_at": "2016-11-16 20:58:35"

}

* 1. Zarządzanie uniwersytetami
     1. Nowy uniwersytet

Zapisuje informacje o nowym uniwersytecie. Uniwersytet jest bezpośrednio w relacji z oddziałem.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/universities | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "AGH Akademia Górniczo-Hutnicza",

"slug": "agh-akademia-gorniczo-hutnicza",

"entity\_slug": "generic"

}

Przykładowa odpowiedź:

{

"id": 2,

"name": "AGH Akademia Górniczo-Hutnicza",

"slug": "agh-akademia-gorniczo-hutnicza",

"entity\_slug": "generic",

"created\_at": "2016-11-16 21:00:04",

"updated\_at": "2016-11-16 21:00:04",

"entity": {

"id": 1,

"name": "generic",

"expa\_name": "KRAKOW",

"expa\_id": 1510,

"slug": "generic",

"created\_at": "2016-11-12 22:02:03",

"updated\_at": "2016-11-12 22:02:03"

}

}

* + 1. Usuwanie uniwersytetu

Usuwa uniwersytet o danym numerze id. Zwraca status 200 w wypadku udanej operacji.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/ api/v1/universities/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

* + 1. Pobieranie listy uniwersytetów

Pobiera listę wszystkich uniwersytetów.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/ api/v1/universities/ | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": 1,

"name": "Uniwersytet Adama Mickiewicza",

"slug": "uam",

"entity\_slug": "poznan",

"created\_at": "2016-11-16 20:35:28",

"updated\_at": "2016-11-16 20:35:38",

"entity": {

"id": 2,

"name": "Poznań",

"expa\_name": "POZNAN",

"expa\_id": 111,

"slug": "poznan",

"created\_at": "2016-11-14 21:24:14",

"updated\_at": "2016-11-16 20:35:16"

}

},

{

"id": 2,

"name": "AGH Akademia Górniczo-Hutnicza",

"slug": "agh-akademia-gorniczo-hutnicza",

"entity\_slug": "generic",

"created\_at": "2016-11-16 21:00:04",

"updated\_at": "2016-11-16 21:00:04",

"entity": {

"id": 1,

"name": "generic",

"expa\_name": "KRAKOW",

"expa\_id": 1510,

"slug": "generic",

"created\_at": "2016-11-12 22:02:03",

"updated\_at": "2016-11-12 22:02:03"

}

}

]

* + 1. Pobieranie uniwersytetu

Pobiera informacje o uniwersytecie z danym numerem id.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/ api/v1/universities/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

{

"id": 1,

"name": "Uniwersytet Adama Mickiewicza",

"slug": "uam",

"entity\_slug": "poznan",

"created\_at": "2016-11-16 20:35:28",

"updated\_at": "2016-11-16 20:35:38",

"entity": {

"id": 2,

"name": "Poznań",

"expa\_name": "POZNAN",

"expa\_id": 111,

"slug": "poznan",

"created\_at": "2016-11-14 21:24:14",

"updated\_at": "2016-11-16 20:35:16"

}

}

* + 1. Aktualizacja uniwersytetu

Aktualizuje informacje o uniwersytecie z danym numerem id.

|  |  |  |
| --- | --- | --- |
| Typ | PUT | |
| URL | {url}/api/v1/universities/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "Politechnika Warszawska",

"slug": "politechnika-warszawska",

"entity\_slug": "warszawa"

}

Przykładowa odpowiedź:

{

"id": 2,

"name": "Politechnika Warszawska",

"slug": "politechnika-warszawska",

"entity\_slug": "warszawa",

"created\_at": "2016-11-16 21:00:04",

"updated\_at": "2016-11-16 21:00:04",

"entity": {

"id": 1,

"name": "Warszawa",

"expa\_name": "WARSZAWA",

"expa\_id": 1510,

"slug": "warszawa",

"created\_at": "2016-11-12 22:02:03",

"updated\_at": "2016-11-12 22:02:03"

}

}

* 1. Zarządzanie użytkownikami
     1. Nowy użytkownik

Tworzy nowego użytkownika systemu.

|  |  |  |
| --- | --- | --- |
| Typ | POST | |
| URL | {url}/api/v1/users | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Ciało zapytania:

{

"name": "Jon",

"email": "jon@doe.com",

"password": "1234"

}

Przykładowa odpowiedź:

[

{

"id": 5,

"name": "jon",

"email": "jon@doe.com",

"created\_at": "2016-11-16 21:06:31",

"updated\_at": "2016-11-16 21:06:31"

}

]

* + 1. Usuwanie użytkownika

Usuwa użytkownika o danym numerze id. Zwraca status 200 w wypadku udanej operacji.

|  |  |  |
| --- | --- | --- |
| Typ | DELETE | |
| URL | {url}/ api/v1/users/{id} | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

* + 1. Pobieranie listy użytkowników

Pobiera listę wszystkich użytkowników.

|  |  |  |
| --- | --- | --- |
| Typ | GET | |
| URL | {url}/ api/v1/users/ | |
| Nagłówki | **Authorization** | Bearer {{token}} |
| **Content-Type** | application/json |

Przykładowa odpowiedź:

[

{

"id": 6,

"name": "Michael",

"email": "michael@brown.com",

"created\_at": "2016-11-16 21:17:03",

"updated\_at": "2016-11-16 21:17:03"

},

{

"id": 5,

"name": "jon",

"email": "jon@doe.com",

"created\_at": "2016-11-16 21:06:31",

"updated\_at": "2016-11-16 21:06:31"

}

]

1. Możliwości rozwoju

- dodanie kreatora formularza w formie drag and drop

- integracja ze sklepami internetowymi typy Woocomerce

Bibliografia

1. **Frontczak Tomasz.** *Marketing internetowy w wyszukiwarkach.* Gliwice : Onepress, 2006.

2. **AIESEC Polska.** *AIESEC Polska - przewodnik marki.* Warszawa : AIESEC Polska, 2015.

3. **The PostgreSQL Global Development Group.** Dokumentacja systemu PostgreSQL 9.4. *Dokumentacja modułu hstore.* [Online] https://www.postgresql.org/docs/9.4/static/hstore.html.

4. **Internet Engineering Task Force.** Internet Engineering Task Force Tools. *Opis standardu RFC7519.* [Online] https://tools.ietf.org/html/rfc7519.

5. **Steve Klabnik Yehuda Katz, Dan Gebhardt, Tyler Kellen, Ethan Resnick.** Json API. [Online] [Zacytowano: 22 11 2016.] http://jsonapi.org/.

Spis Treści

[1. Wstęp 1](#_Toc8580786)

[1.1 Analiza działań marketingowych 1](#_Toc8580787)

[1.2 Ścieżka klienta 1](#_Toc8580788)

[1.3 Współczynniki konwersji wewnątrz ścieżki klienta 1](#_Toc8580789)

[1.4 Zakres niniejszej pracy 2](#_Toc8580790)

[2. Opis organizacji AIESEC 2](#_Toc8580791)

[2.1 O organizacji 2](#_Toc8580792)

[2.2 Struktura organizacji i jej wpływ na procesy marketingowe 3](#_Toc8580793)

[3. Opis funkcji systemu 5](#_Toc8580794)

[3.1 Ogólny opis systemu 5](#_Toc8580795)

[3.2 Udziałowcy i użytkownicy 5](#_Toc8580796)

[3.2.1 Udziałowcy 5](#_Toc8580797)

[3.2.2 Użytkownicy 5](#_Toc8580798)

[3.3 Skrócony opis funkcji systemu na podstawie wywiadu wśród przyszłych użytkowników systemu 6](#_Toc8580799)

[3.4 Scenariusze użycia (user stories) 7](#_Toc8580800)

[3.5 Projekt Interfejsu użytkownika 11](#_Toc8580801)

[3.5.1 Model szkieletowy ekranu analiz 12](#_Toc8580802)

[3.5.2 Model szkieletowy ekranu analiz szczegółowych 13](#_Toc8580803)

[3.5.3 Model szkieletowy ekranu ustawień 14](#_Toc8580804)

[3.5.4 Model szkieletowy ekranu generowania adresów URL 15](#_Toc8580805)

[4. Opis techniczny systemu 15](#_Toc8580806)

[4.1 Wybór technologii 15](#_Toc8580807)

[4.2 Opis architektury 16](#_Toc8580808)

[4.2.1 Wysokopoziomowy schemat architektury aplikacji 16](#_Toc8580809)

[4.2.2 Baza danych 16](#_Toc8580810)

[4.2.2.1 Logika biznesowa 17](#_Toc8580811)

[4.2.2.1 Klucze API 17](#_Toc8580812)

[4.2.2.1 Role i użytkownicy 18](#_Toc8580813)

[4.2.3 Serwer 18](#_Toc8580814)

[4.2.3.1 Routing zapytań interfejsu REST 18](#_Toc8580815)

[4.2.4 Prezentacja 23](#_Toc8580816)

[4.2.4.1 ReactJS 23](#_Toc8580817)

[4.2.4.2 Redux 24](#_Toc8580818)

[4.3 Formularz rejestracyjny 24](#_Toc8580819)

[5. Prezentacja systemu 24](#_Toc8580820)

[5.1 Informacje podstawowe 24](#_Toc8580821)

[5.2 Instalacja systemu 25](#_Toc8580822)

[5.3 Prezentacja wyników implementacji według scenariuszy użycia 25](#_Toc8580823)

[5.3.1 A-1 Logowanie użytkownika 25](#_Toc8580824)

[5.3.2 A-2 Wylogowanie użytkownika 26](#_Toc8580825)

[5.3.3 A-3 Dodawanie użytkowników 26](#_Toc8580826)

[5.3.4 A-4 Usuwanie użytkowników 27](#_Toc8580827)

[5.3.5 B-1 Dodawanie oddziałów lokalnych 27](#_Toc8580828)

[5.3.6 B-2 Usuwanie oddziałów lokalnych 28](#_Toc8580829)

[5.3.7 B-3 Dodawanie uniwersytetów 29](#_Toc8580830)

[5.3.8 B-4 Usuwanie uniwersytetów 29](#_Toc8580831)

[5.3.9 C-1 Dodawanie produktów organizacji 30](#_Toc8580832)

[5.3.10 C-2 Usuwanie produktów organizacji 30](#_Toc8580833)

[5.3.11 D-1 Uruchamianie kampanii marketingowej 30](#_Toc8580834)

[5.3.12 D-2 Archiwizacja kampanii marketingowej 31](#_Toc8580835)

[5.3.13 E-1 Komunikacja z formularzami rejestracyjnymi na stronie www 31](#_Toc8580836)

[5.3.14 E-2 Generowanie linków promocyjnych 32](#_Toc8580837)

[5.3.15 E-3 Prezentacja analiz marketingowych 33](#_Toc8580838)

[6. Dokumentacja API 35](#_Toc8580839)

[6.1 Klucze API 59](#_Toc8580840)

[6.1.1 Tworzenie klucza 59](#_Toc8580841)

[6.1.2 Usuwanie klucza 60](#_Toc8580842)

[6.1.3 Pobieranie listy kluczy 60](#_Toc8580843)

[6.2 Autoryzacja 61](#_Toc8580844)

[6.2.1 Informacje o użytkowniku 61](#_Toc8580845)

[6.2.2 Logowanie nowego użytkownika 62](#_Toc8580846)

[6.3 Zarządzanie kampaniami 62](#_Toc8580847)

[6.3.1 Nowa kampania 62](#_Toc8580848)

[6.3.2 Usuwanie kampanii 63](#_Toc8580849)

[6.3.3 Pobieranie listy kampanii 63](#_Toc8580850)

[6.3.4 Aktualizacja kampanii 64](#_Toc8580851)

[6.3.5 Pobieranie informacji o kampanii 65](#_Toc8580852)

[6.4 Zarządzanie klientami 65](#_Toc8580853)

[6.4.1 Nowy klient 65](#_Toc8580854)

[6.4.2 Lista klientów 66](#_Toc8580855)

[6.4.3 Liczenie klientów 67](#_Toc8580856)

[6.4.4 Statystyki klientów 68](#_Toc8580857)

[6.4.5 Usuwanie klienta 69](#_Toc8580858)

[6.4.6 Generowanie pliku CSV 69](#_Toc8580859)

[6.5 Zarządzanie oddziałami 70](#_Toc8580860)

[6.5.1 Nowy oddział 70](#_Toc8580861)

[6.5.2 Usuwanie oddziału 70](#_Toc8580862)

[6.5.3 Lista oddziałów 70](#_Toc8580863)

[6.5.4 Aktualizacja oddziału 71](#_Toc8580864)

[6.6 Zarządzanie leadami sprzedażowymi 72](#_Toc8580865)

[6.6.1 Nowy lead 72](#_Toc8580866)

[6.6.2 Usuwanie leada 73](#_Toc8580867)

[6.6.3 Pobieranie leadów 73](#_Toc8580868)

[6.6.4 Liczenie leadów 74](#_Toc8580869)

[6.6.5 Statystyki leadów 75](#_Toc8580870)

[6.7 Zarządzanie produktami 76](#_Toc8580871)

[6.7.1 Nowy produkt 76](#_Toc8580872)

[6.7.2 Usuwanie produktu 76](#_Toc8580873)

[6.7.3 Pobieranie produktów 77](#_Toc8580874)

[6.7.4 Pobieranie produktu 77](#_Toc8580875)

[6.7.5 Aktualizacja produktu 78](#_Toc8580876)

[6.8 Zarządzanie uniwersytetami 78](#_Toc8580877)

[6.8.1 Nowy uniwersytet 78](#_Toc8580878)

[6.8.2 Usuwanie uniwersytetu 79](#_Toc8580879)

[6.8.3 Pobieranie listy uniwersytetów 80](#_Toc8580880)

[6.8.4 Pobieranie uniwersytetu 81](#_Toc8580881)

[6.8.5 Aktualizacja uniwersytetu 81](#_Toc8580882)

[6.9 Zarządzanie użytkownikami 82](#_Toc8580883)

[6.9.1 Nowy użytkownik 82](#_Toc8580884)

[6.9.2 Usuwanie użytkownika 83](#_Toc8580885)

[6.9.3 Pobieranie listy użytkowników 83](#_Toc8580886)

[7. Możliwości rozwoju 84](#_Toc8580887)

[Bibliografia 86](#_Toc8580888)

[Spis Treści 87](#_Toc8580889)