Aplikacja składa się z 4 elementów:

1. Stronka HTML z prostym JavaScriptem, która realizuje następujące zadania:
   1. Autoryzuje użytkownika poprzez Cogito (<-co to konkretnie jest i jak działa?)
   2. Umożliwia wysłanie na API systemu formularza z zamówieniem:

Propozycja realizacji: POST z urlem /place\_order

* 1. Pobiera dane dotyczące zrealizowanych przez klienta zamówień

Propozycja realizacji: POST z urlem /results

1. Lambda podlinkowana w API do /place\_order , która:
   1. W wywołaniu pobiera numer liczby fibonacciego do obliczenia + dane identyfikacyjne użytkownika (<- co to konkretnie jest?, format, zakres danych)
   2. Wrzuca do kolejki zamówienie (<- co to konkretnie będzie SNS? SQS? Interfejs? Sposób obsług?)
   3. Wylicza unikalne id zamówienia i zwraca je zamawiającemu (do wyświetlenia na stronce HTML)
   4. Pod rozwagę: czy jednocześnie zakłada rekord w bazie danych w statusie „Processing?”, aby móc go przedsawiać na raportach ?
2. Lambda podlinkowana w API do /results , która:
   1. W wywołaniu pobiera dane indentyfikacyjne użytkownika (j.w. <- format? Zakres? )
   2. Pobiera dane z bazy danych (<-czym konkretnie będzie ta baza? Któraś z baz SQL, czy może DynamoDB?) dla danego użytkownika i zwraca je do wyświetlenia na stronce HTML
3. Oprogramowanie Workera, które:
   1. Pobiera kolejne zlecenie z kolejki do przetworzenia (<- czy to pobieranie to będzie push, czy pull, tzn. czy worker będzie okresowo odpytywał kolejkę, czy to koeljka będzie pukać do workerów, gdy coś będzie miała?)
   2. Po wyliczeniu liczby, zapisuje wynik do bazy

(Jeżeli zdecydujemy się na „push” z kolejki do workerów, to nie wykluczone, że będzie potrzebna jeszcze jedna lambda, którą posługiwała się będzie kolejka – bo po prawdzie nie wiem, jak taki push jest realizowany)