Salceson - dokumentacja techniczna

Jakub Bondyra

8 czerwca 2016

Spis treści

1 Wprowadzenie

Niniejszy dokument stanowi ogólną specyfikację techniczną projektu internetowej gry polegającej na zadawaniu pytań zamkniętych użytkownikowi i próbie odganięcia (powszechnie znanej) osoby, o której początkowo pomyślał użytkownik. Projekt nosi roboczą nazwę "Salceson". Poniżej zostanie przedstawiona charakterystyka użytych technologii, ogólny opis architektury systemu oraz informacje stricte implementacyjne - podział na projekty i moduły i ich krótki opis.

2 Użyte technologie. Ogólny schemat projektu

Aplikacja widoczna po stronie użytkownika to typowa aplikacja webowa, wykonana w technologii ASP.NET WebForms, na języku znaczników HTML5 i arkuszach CSS. Polecenia użytkownika wynikające z interakcji z elementami strony przesyłane są za pomocą języka skryptowego JavaScript i frameworka jQuery technologią AJAX do usługi WCF po stronie serwera. Usługa ta komunikuje się z komponentami napisanymi już w czystym .NET 4.5, które operują na zbiorach danych za pomocą technologii LINQ. Niezbędne zbiory danych pozyskiwane są z bazy danych MS SQL za pomocą mapowania obiektowo-relacyjnego dostarczonego przez Entity Framework.

3 Architektura, szczegółowy opis warstw

Z racji dość szerokiego zakresu prowadzonych prac, aplikacja została podzielona na sześć oddzielnych warstw, które komunikując się między sobą zapewniają określone funkcjonalności. Wynika to z faktu, że oprócz stworzenia sensownej bazy danych i odpowiedniego się z nią łączenia, należy jeszcze rozpatrzeć oddzielne moduły gromadzące wiedzę wynikającą z kontekstu rozgrywki lub odpowiednio ją modyfikujące do przyszłego użytku. Na to należy jeszcze dołożyć aplikację

internetową, która komunikuje się z odpowiednim modułem w celu zasięgnięcia informacji, co w danej chwili należy użytkownikowi wyświetlić. W konsekwencji powoduje to dość złożony podział aplikacji. Poniżej następuje przedstawienie wszystkich warstw w systemie wraz z podstawowym opisem ich działania.

3.1 Warstwa danych

Warstwa danych - czyli baza danych, będąca podstawą całej aplikacji i jej rozgrywek. Stworzona została baza SQL w technologii MS SQL w podejściu Code First z wykorzystaniem mapowania relacyjno - obiektowego dostarczonego przez technologię Entity Framework.

3.2 Warstwa pozyskująca dane

Warstwa ta została stworzona w celu zapewnienia odpowiedniego łącza do danych i zwiększenia testowalności kodu, zapewnia generyczny interfejs pozyskujący rzeczywiste rekordy z bazy na żądania algorytmów. Implementację tej warstwy stanowi klasa *DbRepository*. Pozwala ona na jednorazowe połączenie z bazą danych i załadowanie potrzebnych danych do pamięci programu, by nie obciążać zanadto serwera SQL. Odpowiednie wyższe warstwy komunikują się z bazą danych poprzez stworzenie instacji tej klasy i wykonywanie sekwencji LINQ na kolekcjach w instancji.

3.3 Warstwa algorytmiczna

Warstwa algorytmiczna (rozgrywki) operuje na danych, zapewnia w miarę inteligentne algorytmy wyznaczania pytań i zgadywania osób pozyskując wiedzę z warstwy łącza, ale aktualizuje też wiedzę po zakończonych rozgrywkach, operując bezpośrednio na bazie danych. Takie podejście było niezbędne w celu zachowania spójności i względnej prostoty systemu. Algorytmy zaimplementowane zostały bez użycia dodatkowych technologii, opierają się na danych pobranych z warstwy niższej i dostarczają całość logiki gry. W projekcie niniejszej warstwie odpowiada klasa *DataModule*, wraz ze znaczną ilością klas wykorzystywanych w algorytmie.

3.4 Warstwa interakcyjna

Warstwa interakcyjna zapewnia generyczny interfejs pozwalający na decydowanie o kolejnych ruchach w rozgrywce, operuje na warstwie algorytmicznej odpowiednio nią zarządzając. Z tego powodu użytkownik, czy to korzystający z aplikacji internetowej czy prostej konsoli, wie jedynie jak wymusić reakcję systemu na przykładowo wprowadzoną odpowiedź czy też żądanie dodania pytania do bazy danych, jednakże nie ma wglądu do logiki gry. W projekcie niniejszej warstwie odpowiada klasa *InteractionModule* wraz z interfejsem *IInteractive*.

3.5 Warstwa usług

Warstwa usług - dostępna w przypadku aplikacji internetowej - to z nią łączą się przeglądarki w celu pobrania danych. Zarządza modułami interakcyjnymi dla każdego użytkownika sytemu. Została ona zaimplementowana w postaci prostej usługi WCF, która dostarcza odpowiednie funkcjonalości w oddzielnych funkcjach.

3.6 Frontend

Frontend jest rzeczywistym wyglądem strony wraz z logiką odwołań do warstwy usług. Warstwa dotyczy również aplikacji konsolowej - jest ona mikroskopijna i służy jedynie do prostej kontroli rozgrywki. W przypadku aplikacji konsolowej, frontend aplikacji jest zrobiony na zasadzie obiektu-pluginu, który można stworzyć i uruchomić. Z kolei w przypadku aplikacji internetowej, została przygotowana cała strona internetowa (dostępna w chwili obecnej na stronie

`www.salceson.azurewebsites.net`) wykonana w technologii ASP.NET WebForms. Wszelkie modyfikacje strony wraz z komunikacją z warstwą usług zostały wykonane za pomocą frameworka jQuery.

4 Instalacja, dostęp do źródeł

Projekt jest dostępny na publicznym repozytorium git (adres `https://github.com/JakubBondyra/person-guesser`). Baza danych MS SQL umieszczona jest na serwerze *www.pgdatabase.database.windows.net*, z użytkownikiem SQL *kleofas88* oraz hasłem *Azbest69*.

W celu przygotowania środowiska do pracy nad modyfikacjami projektu, należy ściągnąć źródło z repozytorium o adresie jak wyżej. Tłumaczenie w jaki sposób korzysta się z systemu wersjonowania git mija się z celem niniejszej specyfikacji. Do prawidłowej kompilacji i wykonywania kodu wymagane jest, aby na stacji roboczej był zainstalowany framework **.NET 4.5**. Po wykonaniu niezbędnych kroków aplikacja powinna poprawnie się budować i poprawnie działać. W przeciwnym przypadku nie nastąpiła automatyczna instalacja nugetów, zatem należy doinstalować je ręcznie - wymaganymi nugetami są Entity Framework oraz mechanizm migracji do niego (polecenie Enable-Migrations w package manager).

Aplikacja ma dostęp do bazy poprzez connection string zdefiniowany w konfiguracji *web.config*. Powinien on być poprawny, jednak jeżeli tak nie jest, należy podmienić poszczególne jego części na te wyspecyfikowane na początku niniejszego rodziału. Do bazy danych użytkownik może się również dostać bezpośrednio, poprzez SQL Server Management Studio. Przy łączeniu z serwerem bazodanowym, należy wpisać wymienione wyżej dane (dla użytkownika sql-owego).