

**WYDZIAŁ PODSTAWOWYCH PROBLEMÓW TECHNIKI
POLITECHNIKI WROCŁAWSKIEJ**

ZARZĄDZANIE WYDATKAMI W GOSPODARSTWIE DOMOWYM

ARTUR PTASZEK

Praca inżynierska napisana
pod kierunkiem
dr Wojciecha Macyny

WROCŁAW 2010

Spis treści

1	Zagadnienia	3
1.1	SPA - Single Page Application	3
1.2	MVC	3
1.3	Logowanie tokenowe	3
2	Technologie	4
2.1	Serwer	4
2.2	Klient	5
3	Baza danych	5
3.1	Diagramy przedstawiające zależności między tabelami	5
3.2	Opis tabel	5

Wstęp

Praca swoim zakresem obejmuje stworzenie aplikacji do zarządzania wydatkami w gospodarstwie i wyświetlanie wielu statystyk z nimi związanymi. Ma ona na celu ułatwienie w takim zarządzaniu poprzez wyświetlanie licznych wykresów. Użytkownikiem docelowym jest każdy zarabiający/studujący chcący organizować swoje wydatki. Celem pracy zaprojektowanie i oprogramowanie aplikacji o następujących założeniach funkcjonalnych:

- Dostosowanie interfejsu do szerokiego wachlarzu urządzeń (telefony komórkowe, tablety, komputery)
- SPA (Single Page Application) czyli aplikacja działająca bez odświeżenia strony
- Prosty i intuicyjny interfejs
- Logowanie i rejestracja użytkowników
- Wprowadzanie i edycja przychodów i wydatków
- Tagowanie przychodów i wydatków
- Filtrowanie wydatków (po tagach i po czasie)
- Możliwość sprawdzenia czy można sobie pozwolić na jakiś większy wydatek w przyszłych miesiącach (Symulacja)
- Możliwość tworzenia wyzwań na tagi czyt. ustawienie sobie jakiegoś limitu na konkretną jednostkę czasu
- Możliwość tworzenia wykresów z podsumowaniami wydatków dla tagów
- Stworzenie predefiniowanych wykresów
 - Wykres kołowy ukazujący wykorzystanie tagowania w wydatkach
 - Wykres liniowy ukazujący stosunek przychodów do wydatków w ostatnim roku
 - Wykres liniowy ukazujący bilans przychodów do wydatków w ostatnim roku
 - Wykres liniowy ukazujący stosunek przychodów do wydatków w ostatnim miesiącu

Istnieje wiele aplikacji o podobnej funkcjonalności, które zazwyczaj są połączone z kontem bankowym:

- mBank - posiada interfejs do zarządzania wydatkami
- ING Bank - posiada interfejs do zarządzania wydatkami
- MoneyWiz - Personal Finance
- Money (with sync)
- Bills
- iFinance

lecz aplikacja będzie w formie strony internetowej, przez co będzie dostępna na wszystkie platformy. Także idea Responsive Web Design będzie zwiększała zasięg platformowy. Aplikacja mBanku i ING jest bardzo mało intuicyjna przez co można trafić do klientów tych aplikacji. Praca składa się z n rozdziałów ...

1 Zagadnienia

W pracy zostały wykorzystane liczne technologie, wzorce i zagadnienia, które poniżej zostaną omówione.

1.1 SPA - Single Page Application

Jest to koncepcja, która mówi, że strona powinna być załadowana raz, a każde przejście na kolejną stronę ma być wykonywane asynchronicznie. Wszystkie zmiany strony są pokazywane przez modyfikację drzewa DOM w dokumencie HTML. Uzasadnieniem tego typu stron jest rosnące wrażenia użytkownika przy korzystaniu z aplikacji (User Experience), a także zminimalizowanie transferu danych między przeglądarką a serwerem, przez co czas odpowiedzi serwera jest zdecydowanie szybszy i użytkownik szybciej zobaczy efekt. Koncepcja wiele czerpie z najnowszych technologii jakimi są HTML5, CSS3, JavaScript, AJAX.

1.2 MVC

Złożony wzorzec architektoniczny służący do organizowania struktury aplikacji posiadających interfejsy graficzne. Wykorzystuje on 3 proste wzorce jakimi są Strategia, Obserwator, Kompozyt. MVC zakłada podzielenie aplikacji na poniżej wymienione części składowe:

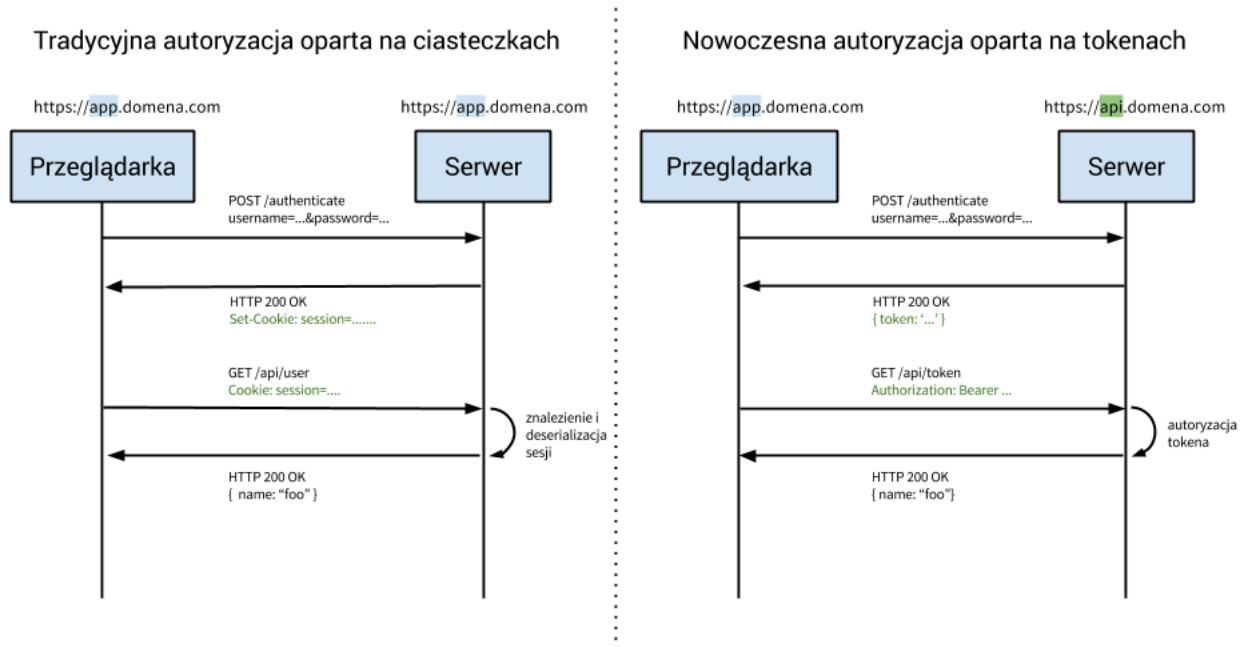
- Model - dane wymagane do utworzenia Widoku
- Widok - opisuje jak przedstawić część modelu użytkownikowi
- Kontroler - agreguje dane z wielu modeli i przygotowuje je aby przekazać do widoku

1.3 Logowanie tokenowe

Logowanie to działa na podobnej zasadzie jak logowanie przy pomocy ciasteczek, lecz znosi pewne ograniczenia jakie wcześniej wspomniano i najpopularniejszy w dzisiejszych czasach system identyfikacji sesji posiadał. Użytkownik, który chce się zalogować wpisuje swój login i hasło, na których podstawie jest generowany token (identyfikator sesji), który z kolei podajemy przy każdym wymagającym autentykacji żądaniu do serwera (poprzez dodanie w nagłówku żądania HTTP pola `Authentication: Bearer <token>`). Token jest wydawany jedynie na określony czas i służy do autentykacji tylko dla jednego użytkownika. Jedną z zalet tego logowania jest to, że możemy w aplikacjach SPA zalogować się bez odświeżenia strony i dodatkowo aplikacje klienckie możemy “hostować” na innej domenie. Nowoczesne strony wykorzystują już tego typu logowanie, idealnym przykładem jest protokół OAuth 2, który wykorzystują liczne strony:

- Facebook
- Google+

- Twitter



Rysunek 1: Diagram przepływu dla logowania tradycyjnego i tokenowego

2 Technologie

Aplikacja została podzielona na 2 części logiczne, którymi są:

- Serwer
- Klient

Każda z nich używa innych technologii, ponieważ działają w różnych środowiskach np. klient pracuje jedynie w przeglądarce.

2.1 Serwer

Została napisana w języku C# wykorzystując ASP.NET MVC 5 i ASP.NET Web Api 2. Statyczne pliki są udostępniane przy pomocy MVC, a reszta serwera została udostępniona w formie Web Service'u.

Silnik bazy danych, który został wykorzystany jest to Microsoft SQL Server 2012 w wersji Express

LocalDB. Cała baza danych została zaprojektowana i wygenerowana przy pomocy Entity Framework 6, który także wspomaga wykonywanie zapytań do bazy danych. Zapytania w tej bibliotece nie wykonuje się bezpośrednio przy pomocy języka strukturalnego SQL, lecz pisząc kod i operując na encjach te zapytania są generowane w locie. Wielką zaletą tego typu rozwiązania jest możliwość przeniesienia aplikacji na inny silnik bazy danych bez zmiany jakiegokolwiek fragmentu kodu. Jediną rzeczą, którą trzeba będzie wykonać to zmienić parametry połączenia. Jest to typowe rozwiązanie mapowania obiektowo-relacyjnego w skrócie ORM.

2.2 Klient

Klient został w większości przy pomocy biblioteki AngularJS. Jest to narzędzie napisane w JavaScript, które umożliwia tworzenie aplikacji za pomocą wzorca projektowego MVC, a także ułatwia tworzenie według koncepcji Single Page Application.

Interfejsu graficzny wykorzystuje bibliotekę Bootstrap, która nie tylko ładnie i schludnie wygląda, lecz przy zastosowaniu pewnych reguł zapewnia nam kompatybilność z innymi urządzeniami tj. telefony komórkowe, tablety, czytniki itd. Tego typu zgodność jest nazywana Responsive Web Design i jest coraz częściej stosowana w internecie.

Wykresy są generowane przy pomocy HighCharts.js, który sprawia, że wykresy są proste w generowaniu.

Aplikacja została napisana modułowo i każdy takich modułów da się przetestować testami jednostkowymi, a wszystko dzięki narzuconemu schematowi przez AngularJS.

3 Baza danych

Jak wyżej zostało napisane baza danych to Microsoft SQL Server 2012 Express LocalDB Poniżej projekt bazy danych.

3.1 Diagramy przedstawiające zależności między tabelami

3.2 Opis tabel

ExpenseModels Przechowuje wydatki i przychody

- Id int IDENTITY (Identyfikator)
- Comment nvarchar(max) (Komentarz)
- Cost decimal(18,2) (Wartość wydatku/przychodu)
- UserId nvarchar(max) (Identyfikator użytkownika)
- DateAdded datetime (Data dodania wydatku/przychodu)
- DateOfExpense datetime (Data wydatku/przychodu wprowadzona przez użytkownika)
- Title nvarchar(max) (Nazwa wydatku)
- DateModified datetime (Data modyfikacji wydatku/przychodu)

LimitModels Przechowuje dane wyzwań

- Id int IDENTITY (Identyfikator)
- Name nvarchar(max) (Nazwa wyzwania)
- Amount decimal(18,2) (Wartość wyzwania)
- From datetime (Data, od której zaczyna się wyzwanie)
- To datetime (Data, do której trwa wyzwanie)
- UserId nvarchar(max) (Identyfikator użytkownika)

LimitModelTagModels Tabela łącząca tabele wyzwań z tagami

- LimitModelId int (Id wyzwania)
- TagModelId int (Id tagu)

SummaryModels Przechowuje dane podsumowań

- Id int IDENTITY (Identyfikator)
- Name nvarchar(max) (Nazwa podsumowania)
- From datetime (Data, od której zaczyna się wyzwanie)
- To datetime (Data, do której trwa wyzwanie)
- Type int (Typ podsumowania)
- Scope int (Zasięg podsumowania np. roczne, miesięczne itd.)
- UserId nvarchar(max) (Identyfikator użytkownika)

TagModelExpenseModels Tabela łącząca tabele wydatków/przychodów z tagami

- ExpenseModelId int (Id wydatku/przychodu)
- TagModelId int (Id tagu)

TagModels Tabela przechowująca tagi

- Id int IDENTITY (Identyfikator)
- Name nvarchar(max) (Nazwa tagu)
- UserId nvarchar(max) (Identyfikator użytkownika)
- SummaryModelId int (Identyfikator podsumowania)

Pozostałe tabele są wykorzystywane przez standardowy mechanizm autentykacji ASP.NET MVC.