

ZACHODNIOPOMORSKI UNIWERSYTET TECHNOLOGICZNY W SZCZECINIE

WYDZIAŁ INFORMATYKI



Konrad Zając

Kierunek Informatyka

Aplikacja internetowa wspomagająca ocenę usług gastronomicznych

Praca dyplomowa inżynierska
napisana pod kierunkiem
dr Małgorzaty Pelczar
w Katedrze Architektury
Komputerów i Telekomunikacji

Szczecin 2016

Web application for assisting in the evaluation of gastronomical points

Konrad Zajac
Supervisor: dr Małgorzata Pelczar

Abstract

The main aim of this transcript is how to develop a web application, which helps in the process of grading a restaurant. The program allows the user to grade a restaurant, to browse them, or to send a request to the administrator - who in turn, can invoke more actions. Another aim is the discussion of the technologies that are commonly used in creating a web application. The application is based on a client-server architecture and was created using many free, and open-source technologies, like: PHP, HTML, CSS, JavaScript, Jquery, BootStrap v.3 framework and MySQL. In order to create a responsive design I have used a combination of the first six of them and the last is for management of the database. In the first chapter basic concepts are presented, which are needed before the rest of the transcript can follow. The Second chapter describes the technologies, which are used to create a web application, and a database to support it. Also in this chapter there are definitions of the word used to describe those technologies, and a logical division of the whole application to the layers of the Domain-Driven Design methodology. The third chapter, with the knowledge taken from the previous ones, is about the final choice of the technologies being used to create this application. In the fourth chapter there is a description of the test to the application. It contains the results of the test, the interpretation of them, and examples of other uses of the application.

OŚWIADCZENIE

Oświadczam, że przekładaną pracę dyplomową kończącą studia napisałem samodzielnie. Oznacza to, że przy pisaniu pracy poza niezbędnymi konsultacjami, nie korzystałem z pomocy innych osób, a w szczególności nie zlecałem opracowania rozprawy lub jej części od innych osób. Potwierdzam też zgodność wersji papierowej i elektronicznej złożonej pracy. Mam świadomość, że poświadczenie nieprawdy będzie w tym przypadku skutkowało cofnięciem decyzji o wydaniu dyplomu.

Spis treści

Wstęp	5
1 Charakterystyka usług gastronomicznych	6
1.1 Punkt gastronomiczny	6
1.2 Podział punktów gastronomicznych	6
1.3 Cechy punktu gastronomicznego	6
2 Przegląd wybranych technologii tworzenia aplikacji webowych	7
2.1 Definicje podstawowych pojęć	7
2.1.1 Aplikacja web-owa	7
2.1.2 HTTP	7
2.1.3 URL	9
2.1.4 Architektura klient-serwer	9
2.1.5 Framework	10
2.1.6 Enkapsulacja	11
2.1.7 Krzywa uczenia	11
2.1.8 Search Engine Optimization	12
2.1.9 Domain-Driven Design	12
2.2 Technologie wykorzystywane przy tworzeniu FrontEndu aplikacji internetowych	15
2.2.1 Języki i paradygmaty oprogramowania	15
2.2.2 PHP	16
2.2.3 HTML	16
2.2.4 Połączenie CSS + HTML + JavaScript	16
2.2.5 ASP.net	18
2.2.6 Ruby	18
2.3 Technologie do implementacji BackEndu	19
2.3.1 Początek wyboru bazy danych	19
2.3.2 SQL	20
2.3.3 PostgreSQL	21
2.3.4 MySQL	22
2.3.5 SQLite	23
2.3.6 Microsoft SQL Server	23
2.4 Podsumowanie wyboru technologii	24
3 Aplikacja wspomagająca ocenę usług gastronomicznych	25
3.1 Definicje wymagań funkcjonalnych	25
3.1.1 Poziom 0	27
3.1.2 Poziom 1	27
3.1.3 Poziom 2	29
3.2 Cechy opracowanej aplikacji	29
3.3 Architektura opracowanej aplikacji	30

3.3.1	Podsumowanie wyboru technologii BackEndowych	30
3.3.2	Podsumowanie wyboru technologii FrontEndowych	31
3.3.3	Technologie w ujęciu Domain-Driven Design	32
3.3.4	Struktura bazy danych	32
3.3.5	Struktura kodu	34
3.3.6	Scharakteryzowanie procesu oceny	34
3.4	Możliwość wdrożenia	34
4	Testy i przykłady zastosowania opracowanej aplikacji	36
4.1	Prezentacja aplikacji	36
4.2	Narzędzie użyte do testowania	37
4.3	Przykłady stosowania aplikacji	38
4.3.1	Początkowy etap testów	38
4.3.2	Główny etap testów	39
4.3.3	Opis wyników testowania	40
Podsumowanie		41
Spis rysunków		42
Spis tabel		43
Literatura		44

Wstęp

Nie ulega wątpliwości, że branża gastronomiczna odgrywa istotną rolę w życiu ludzi. Jest ona ściśle związana z pierwotną potrzebą człowieka, co powoduje jej szybki rozwój. Potwierdzają to liczne doniesienia, np. artykuły ze strony www.portalspozywczy.pl [8], oraz serwisu forbes.pl [9]. Zjawisko to powoduje znaczny wzrost zainteresowania tą branżą międzynarodowych konglomeratów. Widoczny jest wzrost liczby nowych i rozwijających się punktów serwujących pożywienie.

Celem pracy jest zaprojektowanie i implementacja aplikacji internetowej do wspomagania oceny usług gastronomicznych. Głównym zadaniem aplikacji jest ułatwienie oceny wizyty w restauracji (poprzez dodanie pięciostopniowej oceny z opcjonalnym komentarzem), oraz zaplanowania następnej (wspomagając się ocenami innych użytkowników). Aplikacja ma wspomóc nie tylko klientów, ale również przedsiębiorców. Klientom zapewnia pomoc przy wyborze punktu w postaci informacji i opinii zebranych od wielu użytkowników. Dla samego prowadzącego punkt gastronomiczny dostarcza informacji zwrotnej, wskazującej zaniedbania i kierunki rozwoju. Dodatkowo aplikacja promuje punkty, które nie posiadają środków na reklamę. Wraz z rozwojem technologii, konsumenci coraz częściej przy decyzji o wyborze miejsca posiłku, wspomagają się informacjami zdobyтыmi za pośrednictwem Internetu. Jednak informacje te, zależą od doświadczenia użytkownika w używaniu urządzeń technologicznych, oraz posiadanego sprzętu - właśnie dlatego aplikacja web-owa dostępna poprzez WWW, jest najlepszym rozwiązaniem.

Praca składa się z czterech rozdziałów.

- W rozdziale pierwszym podane są podstawowe pojęcia, które dalej będą użyte w pracy, ponieważ są niezbędne, bez wcześniejszego ich zdefiniowania, opis pracy byłby niespójny.
- W rozdziale drugim przedstawiony jest przegląd dostępnych technologii tworzenia aplikacji webowych i projektowania baz danych. Znajdują się w nim również definicje pojęć, które są niezbędne przy dalszym rozwoju aplikacji. W tym rozdziale zaznaczony jest logiczny podział aplikacji na warstwy w metodologii Domain-Driven Design.
- Rozdział trzeci, posługując się wiedzą zaczerpniętą z poprzednich rozdziałów, mówi o ostatecznym wyborze narzędzi do tworzenia aplikacji. Znajduje się tam charakterystyka funkcjonalna, oraz sposób implementacji aplikacji wspomagającej ocenę punktów gastronomicznych.
- W rozdziale czwartym zawarty jest opis testowania wybranej aplikacji. Zawiera on wyniki testów, ich analizę, a także przykłady innych zastosowań.
- W podsumowaniu zostały przedstawione dalsze kierunki rozwoju aplikacji.

1 Charakterystyka usług gastronomicznych

1.1 Punkt gastronomiczny

Punkt gastronomiczny (*PG*) to miejsce, w którym można odpłatnie spożyć posiłek. Nie sprowadza się to jedynie do punktów, które od początku swojego istnienia serwowały posiłki. PG zawiera również punkty z innych branży, które z czasem rozszerzyły swą działalność o usługi gastronomiczne. W tym zbiorze zawierają się m.in. bary mleczne, stacje paliw, hotele, puby oraz czytelnie, w których serwowane są posiłki.

1.2 Podział punktów gastronomicznych

Dzisiejszy podział punktów gastronomicznych różni się od starszego poziomem skomplikowania. Starsza wersja mniej szczegółowo dzieliła PG, z powodu mniejszej ich ilości, co pociąga za sobą ich różnorodność. W dzisiejszych czasach, wraz ze wzrostem liczności tych punktów, podział stał się skomplikowany. Istotne jest zaprojektowanie odpowiedniego schematu, ponieważ aplikacja powinna analizować oceny względem podziału PG.

1.3 Cechy punktu gastronomicznego

Punkty gastronomiczne różnią się pod wieloma względami. Poprawna ocena powinna zawierać informacje, które są istotne z punktu widzenia oceny usług gastronomicznych. Jest to indywidualna preferencja, dlatego w tej aplikacji użytkownik ma pięć osobistych wag (w skali 1-5) dla różnych cech. Są nimi;

Cena jest to liczba oceniająca stosunek jak adekwatna jest cena do jakości spożytego posiłku.

Muzyka to wartość liczbową prezentującą jak w punkcie jest rozwiązana kwestia muzyki.

Różnorodność menu ta ocena mówi o zakresie dostępnych potraw.

Wystrój ta ocena stwierdza jak otoczenie wpłynęło na posiłek.

Smak pokazuje w jakim stopniu jedzenie smakowało.

Każdy PG opisany jest przy pomocy następujących danych:

Nazwa

Miasto

Ulica

Nr domu

Telefon - jest to telefon kontaktowy bądź adres email

Godziny otwarcia - są zmienne, ponieważ często zależą od preferencji właściciela

2 Przegląd wybranych technologii tworzenia aplikacji webowych

2.1 Definicje podstawowych pojęć

2.1.1 Aplikacja web-owa

Aplikacja web-owa jest narzędziem, które pomaga nam zrealizować określona czynność z użyciem Internetu. Uruchomienie jej sprowadza się jedynie do odwiedzenia określonej strony, tj. do wejścia na zadany URL z poziomu przeglądarki internetowej, do której mamy powszechny dostęp.

Takie podejście dostarcza licznych zalet. Pierwszą z nich jest łatwość użycia aplikacji. Spośród osób posługujących się urządzeniami z przeglądarką, większość użytkowników potrafi samodzielnie wykonać powyższą czynność. Łatwość użycia pociąga za sobą stale zwiększającą się liczbę użytkowników. Jak pokazują nam przykłady wielkich serwisów (m.in., Facebook, WP.pl, Onet.pl) popularne są te aplikacje, do których nie musimy instalować samodzielnego programu. Niektóre funkcje wymagają większej mocy obliczeniowej, ale ostatnie trendy pokazują, że wraz z postępem techniki jesteśmy w stanie zapewnić użytkownikowi to samo doświadczenie bez konieczności instalowania dodatkowego oprogramowania. Owszem istnieje taka możliwość, niemniej, nie jesteśmy zmuszeni z niej korzystać.

Wspomniana cecha związana jest z drugą z zalet. Jest nią oszczędność środków, które użytkownik musiałby przeznaczyć na zakup nowego urządzenia. Cechą ta korzystnie wpływa na sukces wielu nowych usług streamujących. Dlatego też główny nacisk powinien być położony na stronę WWW, która nie jest zasobochłonna, a dopiero później powinny być rozważane inne rodzaje oprogramowania. Dodatkowo rozwiązywanie wymagające jedynie przeglądarki daje największą wygodę oraz dostępność.

Trzecią zaletą jest dostęp do aplikacji w dowolnym czasie i miejscu. Jest to szczególnie istotne dla użytkownika, który planuje posiłek w określonej lokalizacji.

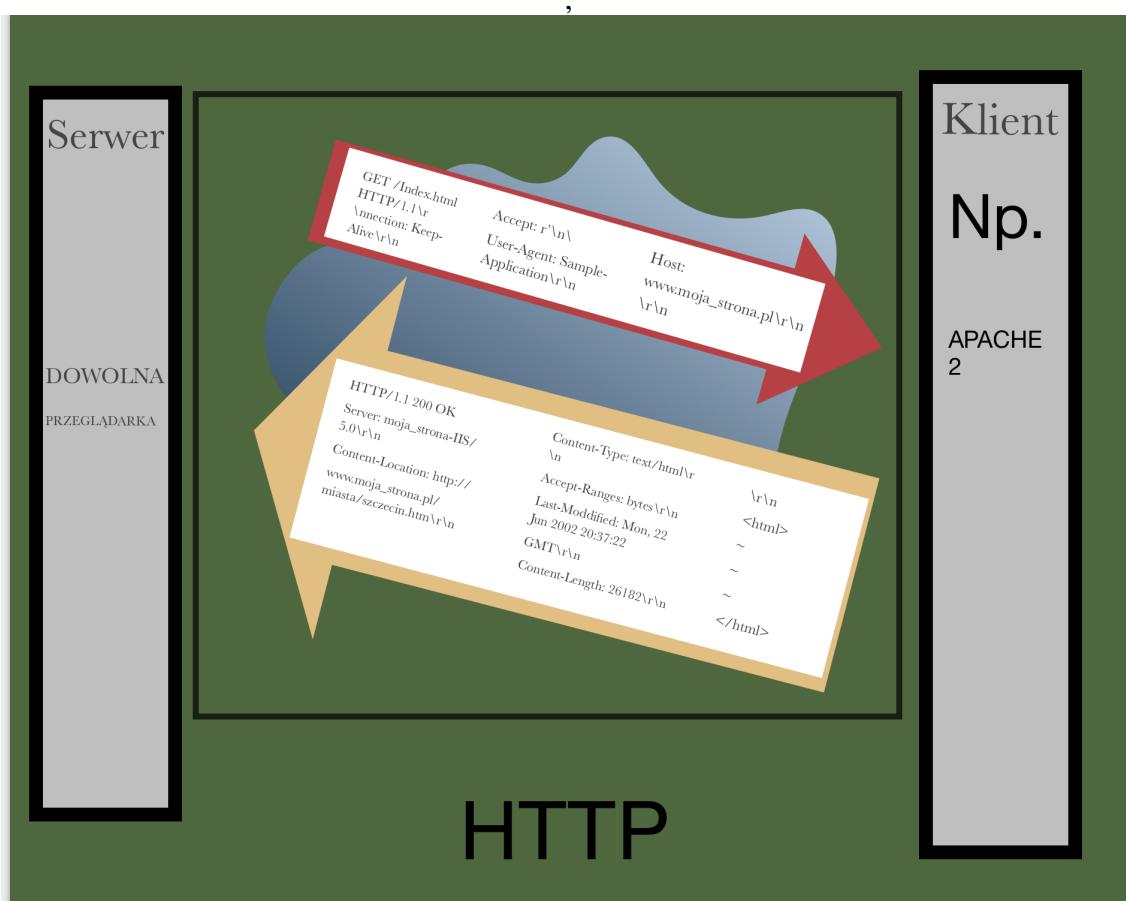
Czwartym pozytywem jest zysk z „metadanych”, czyli danych o danych. Lokalizację klienta oraz czas zalogowania można przetworzyć na wiele sposobów, usprawniając działanie aplikacji. Przykładem może być prowadzenie statystyk, pod względem czasu zalogowania, miejsca, z którego użytkownik się loguje, itp.

2.1.2 HTTP

HTTP (ang. Hypertext Transfer Protocol) jest jednym z wielu „protokołów internetowych”. Są one zbiorem formalnych kroków, jakie muszą podjąć maszyny, aby ustawić stabilne połączenie między sobą. Ten przypadek protokołu służy do przesyłania dokumentów hipertekstowych i działa na siódmej warstwie modelu OSI (aplikacji). Komunikacja za pomocą HTTP odbywa się za pomocą mechanizmu żądanie-odpowiedź. Żądanie http składa się z czterech elementów, nagłówka, wiersza żądania, pustych wierszy, i części głównej. Klient inicjuje komunikację, wysyłając żądanie na porcie 80, czasami 8080. Serwer „nasłuchiwa” na tych portach, a po otrzymaniu odsyła potwierdzenie otrzymania wiadomości,

oraz właściwą wiadomość lub komunikat o błędzie. Przeglądarki zapamiętują pewne elementy stron, aby uniknąć redundancji danych. Poniższy diagram prezentuje szczegóły funkcjonowania protokołu HTTP w przypadku, gdy żądany przez klienta zasób (przykładowo, konkretna strona WWW) został znaleziony. Na poniższym diagramie został opisany ten proces. W pierwszej części, po wpisaniu przez użytkownika adresu URL, przeglądarka

Rysunek 1: Etapy nawiązania połączenia http.



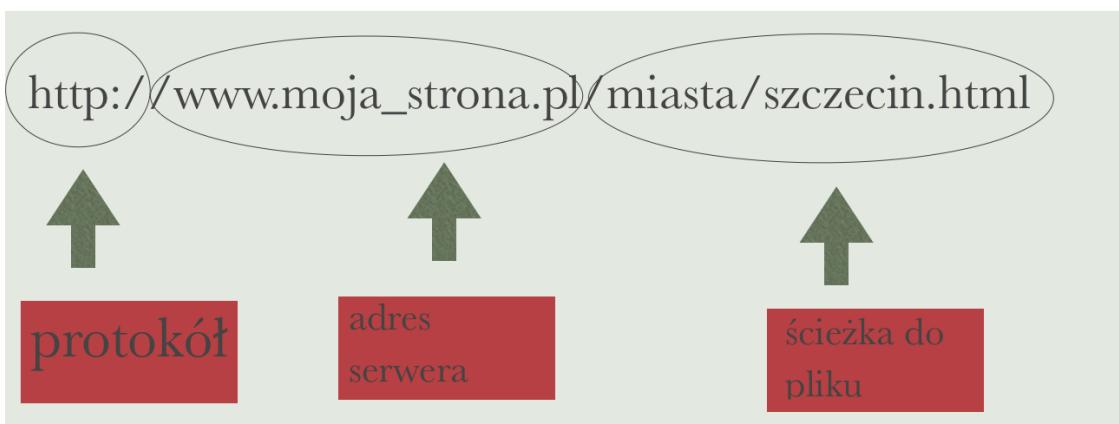
Źródło: Opracowanie własne.

internetowa wysyła zapytanie o udostępnienie zasobu, do serwera (czerwona strzałka). Jeżeli żądanie może być zrealizowane (walidacja się powiedzie), to serwer odsyła drugą wiadomość (żółta strzałka). Wadą tego protokołu jest element zachowania bezpieczeństwa - w odpowiedzi na ten problem powstał protokół https (przy użyciu SSL). Jednak nie jest to rozwiązanie zadowalające, ponieważ w X.509 sprawdzenie ścieżki poprawności może być nieprawidłowe. Główne i pośrednie centra weryfikacji nie mają wyszczególnionych konkretnych domen. Dlatego rozwiązaniem jest HTTP Public Key Pinning: [2] *Ten mechanizm pozwala mitygować ryzyko bezpośrednio związane z modelem X.509, które dowolnemu centrum certyfikacji pozwala wygenerować certyfikat dla dowolnej domeny.*

2.1.3 URL

URL (Uniform Resource Locator) jest z punktu widzenia użytkownika pewną pomocą w znajdowaniu zasobów. W zależności od rodzaju zasobu, używane są różne protokoły, czyli pewne przyjęte sposoby komunikacji. Jednym z nich jest HTTP. Jest to protokół, służący do przeglądania zasobów internetowych. Najczęściej spotykanym przypadkiem w sieci jest tłumaczenie adresów strony WWW. Tą konwersją adresów zajmuje się usługa DNS (Domain Name System). Jako użytkownikom łatwiej jest zapamiętać adres www.wp.pl niż 212.77.100.101, chociaż są to równoważne adresy, prowadzące na tę samą stronę. Aby zobrazować to zjawisko zaprezentuję poszczególne części na pełnym adresie: http://www.moja_strona.pl/miasta/szczecin.html.

Rysunek 2: Składniki URL.



Źródło: Opracowanie własne.

Pierwszym elementem jest protokół, który świadczy, o tym w jaki sposób, oraz jakiego rodzaju zasób chcemy osiągnąć. W tym przypadku jest to protokół http, czyli protokół służący do wysyłania żądania udostępnienia pliku. Kiedyś było wymagane wpisanie protokołu przez użytkownika, teraz przeglądarki domyślnie go dopisują do wpisywanego adresu.

Drugim jest host (adres serwera), w tym przypadku `www.moja_strona.pl`. Element ten mówi nam o tym, że zasób będzie pochodził z usługi internetowej www, oraz nazywa się `moja_strona`, i ma domenę przydzieloną Polsce.

Trzecim elementem jest ścieżka do pliku szukanego zasobu(resource), tutaj `/miasta/szczecin.html`, który zawiera interesującą nas treść.

2.1.4 Architektura klient-serwer

Architektura klient-serwer została zdefiniowana w książce Sieci Komputerowe w następujący sposób: [1] *Sieć klient-serwer stanowi implementacje dwuwarstwowej architektury, w której system serwerowy przesyła dane wykorzystywane przez system kliencki lub wiele systemów klienckich. Rozwiązania klient-serwer są obecnie najczęściej wdrażanymi formami rozproszonego przetwarzania danych, i znajdują zastosowanie w wielu aplikacjach sieciowych takich jak bazy danych, poczta elektroniczna, serwisy WWW i inne*

typowe usługi internetowe. Istnieją pewne przypadki, w których nie da się jasno wskazać podziału klient-serwer. Przykładem jest to, kiedy jedna maszyna jest jednocześnie klientem i serwerem. Obrazować to może sytuacja, kiedy serwer jest uruchomiony lokalnie, a na tym samym urządzeniu (na maszynie wirtualnej) działa klient. Ten podział należy więc rozumieć funkcjonalnie. Klient to narzędzie użytkownika, za pośrednictwem którego dokonuje on rozmaitych czynności, jak na przykład przeglądanie stron WWW. Serwer zaś to dostawca usług, treści i innych zasobów, które są od niego żądane przy użyciu oprogramowania klienckiego.

W przytoczonym przykładzie stron WWW, oprogramowanie klienckie to jedynie przeglądarka internetowa. Serwer to aplikacja udostępniająca strony internetowe za pośrednictwem odpowiednich protokołów rozumianych przez obie strony.

2.1.5 Framework

Framework jest pomocą często używaną przy tworzeniu aplikacji webowej. Jest to częściowe rozwiążenie problemu osadzonego w konkretnej domenie funkcjonalnej. Udostępnia interfejs do specjalizacji zachowania (najczęściej przez inwersję kontroli[7]). Inwersja kontroli jest pojęciem złożonym, ale jej głównym zadaniem jest zamiana zależności. Zwyczajowo to obiekt nadzędny decyduje, o tym jakich komponentów nowo utworzona instancja będzie używała. Natomiast stosując tę technikę decyduje o tym nowy obiekt. Przykładem jest MFC (Microsoft Foundation Classes) - pomaga w tworzeniu graficznych aplikacji na platformę PC. Nie jest to kompletny framework, a jedynie biblioteka programistyczna do Visual C++, ale był na tyle użyteczny, że stał się branżowym standardem. Raphal Saunier dobrze rozszerza to pojęcie, dlatego przytoczę cytat z jego książki pt. „Laravel 4” [6] *Dogłębne poznanie nowego frameworku - takiego jak Laravel - bywa wyzwaniem, ale jednocześnie przynosi dużo satysfakcji. Pozwala nie tylko poznać nowe sposoby na rozwiązywanie problemów, ale również podważyć zasadność wydawałoby się utartych schematów postępowania. Przede wszystkim nowy Framework ma zwiększyć szybkość realizacji projektu przy jednoczesnej poprawie jakości kodu.* Framework ma za zadanie wspomóc programistę. Robi to w wieloraki sposób, między innymi przez modularność. Frameworki zwiększają obiektywność języka poprzez enkapsulację, która jest opisana poniżej. Enkapsulacja pozwala zbudować stabilny interfejs. W kodzie, który jest napisany obiektywnie łatwiej jest zadbać o jakość aplikacji, poprzez lokalizowanie miejsc które zostały dotknięte zmianami. Dostarczy to informacji o działaniu dotyczących zmian, oraz informacje do budowania przyszłych. Drugą kwestią jest ponowne użycie kodu. Stabilne interfejsy dostarczane poprzez Framework mogą być wielokrotnie użyte. Dzięki tej możliwości możemy uniknąć powtarzania pewnych funkcjonalności. Ponowne użycie sprawdzonych części kodu, może zwiększyć produktywność i efektywność. Świadczy to o popularności tych rozwiązań - jeżeli tylu programistów ich używa, to przy najmniej warto się tym zainteresować. W dzisiejszych czasach informatyka rozrosła się na tyle, że tradycyjne budowanie aplikacji od podstaw jest zbyt czasochłonne i nieopłacalne. Przyszłość takiego podejścia do programowania jest obiecująca, o czym może świadczyć popularność takich frameworków, jak Laravel, Ruby on Rails, i inne.

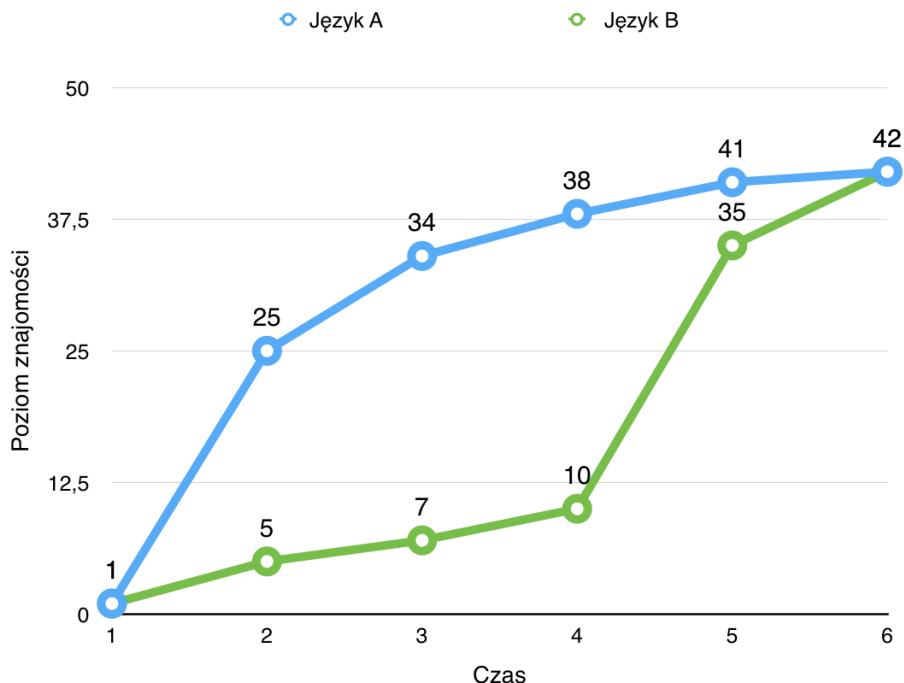
2.1.6 Enkapsulacja

Enkapsulacja jest jednym z kluczowych pojęć prezentowanej pracy. Zjawisko to charakteryzuje się tym, że obiekty mają różne poziomy dostępu do danych. Czasem obiekty mają inne poziomy dostępu do zawartych w sobie funkcji i danych. Jest to tylko jedna cecha języków obiektowo zorientowanych, jednak jest niezwykle istotna. Implementując aplikacje można stworzyć typ PG który będzie trzymał swoje wrażliwe funkcje, typu Usun_punkt, lub Edytuj_dane, prywatnie. Natomiast funkcje typu Dodaj_ocene_jedzenia, lub Dodaj_ocene_wystroju, będą publiczne dostępne dla każdego.

2.1.7 Krzywa uczenia

Na wstępie należy zaznaczyć, że „krzywa uczenia” jest pojęciem abstrakcyjnym i dla każdej osoby może mieć inne znaczenie. Jest to zależność poziomu znajomości względem czasu nauki.

Rysunek 3: Krzywa uczenia.



Źródło: opracowanie własne.

Przykładem będą tu dwa języki, A oraz B. Język A reprezentuje te języki w których łatwo przejść pierwsze etapy, ale późniejsze etapy doskonalenia są żmudne i czasochłonne. Dobrym przykładem takiego języka to Python. Natomiast język B cechuje się tym, że początkowe kroki są wymagające, ale kiedy już nauczymy się podstaw, dalsza wiedza przychodzi nam łatwiej. Obrazuje to język C++, dla osoby która nie miała wcześniej styczności z nowo dodaną funkcjonalnością, obecną w tym języku. Początkowo jest to bardzo trudny język, ale w miarę poszerzania wiedzy, okazuje się być językiem preferowanym. Dobra to widać w procesie starania się o pracę. Czasami ludzie tuż po ukończeniu studiów

twierdzą, że są ekspertami w danym języku. Z czasem podczas nauki danego języka poznajemy ile jest w nim funkcjonalności, której jeszcze nie zgłębiliśmy. Wtedy zmienia się samoocena względem danego języka. Języki programowania są jak narzędzia. Nie możemy porównywać młotka do wkrętarki, ale z całą pewnością możemy stwierdzić wyższość wkrętarki nad śrubokrętem podczas skręcania wielu elementów.

2.1.8 Search Engine Optimization

Search Engine Optimization (SEO) jest optymalizacją pod kątem wyszukiwania. W każdej aplikacji internetowej, jest to ważny element - jednak w tej będzie kluczowy. Jest to bardzo rozległe zagadnienie, któremu poświęcono kilka książek. Głównym celem SEO jest stworzenie strony w taki sposób, aby link do niej był jak najbliżej początku w wyszukiwaniu np. w Google.pl . Aby dobrze je poznać, trzeba zrozumieć jak działa wyszukiwarka. Informacje, które chcemy przekazać powinny być poprawnie opisane, np. dobrą praktyką jest, aby atrybuty zdjęcia zawierały konkretny opis tego co ono przedstawia. Kolejnym przykładem jest inny wybór nazw dla przycisków, np. zamiast przycisku z nazwą "ok" używamy przycisku z nazwą "kupuję mieszkanie". Rzeczy niezależne od kodu to odnośniki i polecenia z innych stron. W tym przypadku jeżeli na stronie "Master Chef", będzie link do naszej strony będzie to bardziej znaczące niż ten sam link byłby na stronie tartak.pl .

2.1.9 Domain-Driven Design

Domain Driven-Design (DDD) jest jedną z metodologii. Każda metodologia to sposób ukierunkowania procesu tworzenia aplikacji. Materializuje stopnie, które trzeba podjąć, aby sukcesywnie ją stworzyć, a następnie wdrożyć. Co układa cały plan w pewnego rodzaju moduły - innymi słowami domeny. Wg. Erica Evansa [3] *Domena to zbiór wiedzy, zależności, lub aktywności (np. domena sztuki)*.

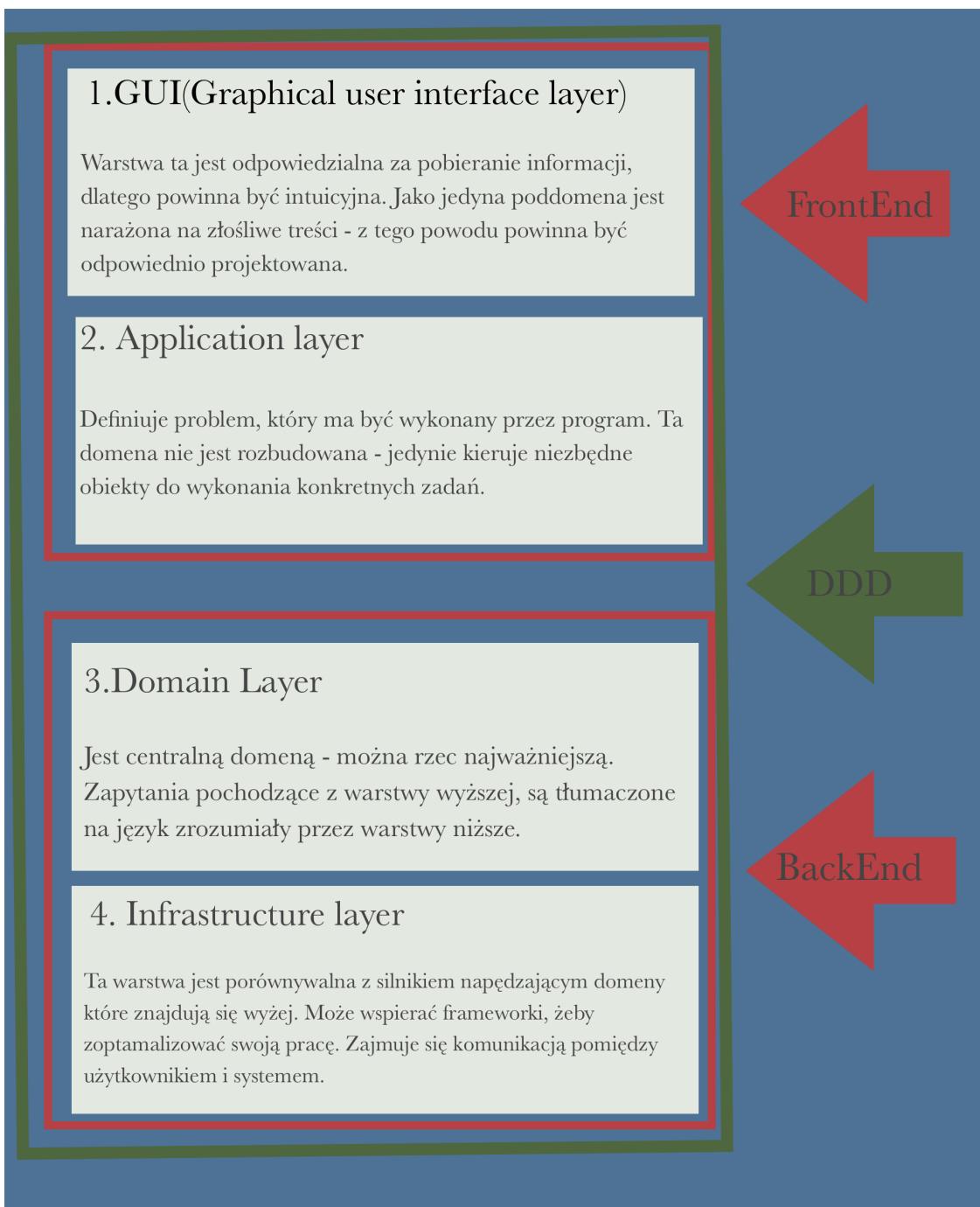
Z tych domen łatwiej jest zbudować cały system. W tym przypadku DDD jest bardziej rozwinięta, ponieważ kładzie większy nacisk na zrozumienie schematu projektu. Zazwyczaj schemat jest opisany językiem technicznym, a co za tym idzie, jedynie programiści mogą go zrozumieć. DDD w tym temacie się różni, dlatego każdy, może odczytać schemat całej aplikacji - co jest bardzo ważne. To ujednolicenie pozwala wszystkim precyzyjnie zaadresować swoje uwagi, jak i wskazać lepsze rozwiązania, co wpływa pozytywnie na całokształt projektu . Ponadto metodologie sprawne (Agile Metodologies), których częścią jest DDD - są nowym podejściem do procesu tworzenia aplikacji. Dotychczas stosowano metodologie kaskadowe, których jeden z kroków zatytułowany „Testowanie”. Nowsze techniki zakładają, że proces testowania idzie współbieżnie z procesem tworzenia, a nawet całego cyklu istnienia aplikacji. Ta możliwość szybkiej zmiany jest ważną częścią sprawnych metodologii, czasem w momencie projektowania niektóre aspekty nie są precyzyjnie zdefiniowane. Opcja zmiany w czasie rzeczywistym, rozwiązuje ten problem, może to nawet doprowadzić do powstania czegoś zupełnie nowego. Kolejną zaletą prowadzenia projektu w ten sposób jest pewnego rodzaju destylacja, która polega na oddzieleniu zbędnych pomysłów od tych kluczowych. Ma to w sobie jeszcze jedną korzyść, mianowicie łatwość podjęcia decyzji o technologii wybranej do tego projektu. Jest to też pewien

sposób na ujednolicenie wiedzy, która jest rozproszona pomiędzy wieloma miejscami. Sposób przekazania tej wiedzy, często może być ważniejszy od niej samej. Programista nie posiadający wiedzy o tym, jak dana funkcjonalność ma działać najprawdopodobniej zaimplementuje ją niepoprawnie. Dlatego wiedzę o projekcie w większości przypadków powinniśmy przekazać grafikami typu UML. Stwierdzenie „w większości przypadków” ma na celu podkreślenie, że niektóre funkcjonalności mogą być nieczytelne na diagramie UML. Może ograniczać to np. czcionka tekstu na diagramie. Problem ten można rozwiązać poprzez powiększanie, ale wtedy schemat straci na czytelności. Przekazanie wiedzy o projekcie powinno być jak najbardziej dostosowane do przypadku, z uwzględnieniem osoby, do której jest ona przekazywana. Jednym z najbardziej naturalnych sposobów przekazu dla człowieka, jest przekaz słowny. W tym przypadku ważna jest całkowita wiedza o całym projekcie, oraz umiejętność podziału skomplikowanej struktury na mniejsze - łatwiejsze do zrozumienia części. Cała aplikacja dzieli się na mniejsze fragmenty, zwane FrontEnd i BackEnd. Te części rozchodzą się na kolejne dwie, co przedstawiono na Rysunku 6.

Na diagramie widoczne są dwie, komplementarne części, tzw. „*Core domains*”, czyli części główne, które zawierają dwie poddomeny. są to FrontEnd i BackEnd, dalej zwane pierwsza i druga część. Pierwsza część aplikacji - jedyna z jaką użytkownik ma styczność, powinna być intuicyjnie zaprojektowana, oraz odporna na złośliwe dane. Przykładowa osoba nie jest skłonna poświęcić wiele swojego czasu na ocenę PG, dlatego ta część powinna być zaprojektowana efektywnie. Powinna nakłaniać użytkownika do dalszego uzupełniania np. hasłami typu „jeszcze tylko dwa pytania” czy „ukończono proces w 95%”. Realizacja jako GUI (Graphical User Interface), czyli część graficzna aplikacji, pozwoli na uzyskanie zamierzonych celów. Będzie ona służyć do sformułowania zapytania do dalszej części i wyświetlenia informacji zwrotnej.

Zadaniem drugiej części jest przetworzenie tego, co użytkownik wpisał w GUI aplikacji. W tej części będzie działał algorytm omawiany w rozdziale „wymagania funkcjonalne”. Akcja użytkownika zawsze powinna być skierowana, na uzyskanie wrażenia stabilności. Właśnie z tych powodów do wykonania tego projektu wybrano DDD.

Rysunek 4: Domain-Driven Design.



Źródło: opracowanie własne.

2.2 Technologie wykorzystywane przy tworzeniu FrontEndu aplikacji internetowych

2.2.1 Języki i paradigmaty oprogramowania

Chcąc wybrać odpowiedni język do odpowiedniego zadania musimy spojrzeć na najpopularniejszy podział na paradigmaty. Paradigmaty to pojęcia które pozwalają, na skategoryzowanie języka biorąc pod uwagę różnego typu właściwości, koncepcje i mechanizmy programistyczne, według których, każdemu językowi możemy przypisać jedną z par dwóch wykluczających się cech.

Kompilowane lub skryptowe. O języku skryptowym możemy mówić kiedy kod interpretowany jest bezpośrednio przez powłokę, bez komplikacji np.(Bash, Perl, Python, czy MATLAB). Natomiast o kompilowanym, kiedy wynik jest poddawany komplikacji - czyli kod jest tłumaczony na język maszynowy(uruchamialny w danym środowisku), przykładowo C, C++, Java, i Erlang.

Wysoko lub nisko poziomowe. Wysokopoziomowe to takie które charakteryzują się kompletnością funkcjonalną standardu, niezależnie od środowiska (Mac, Linux, lub Windows), z reguły te języki mają słabszą wydajność. Przeważnie towarzyszy im wirtualna maszyna, która instaluje się w środowisku jako interpreter kompilantu. Z drugiej strony mamy języki niskopoziomowe, które zazwyczaj charakteryzują się niekompletnością funkcjonalną standardu. Są silnie zależne od platformy (wymagają przynajmniej rekompilacji, lub nawet interwencji programistycznej), oraz mają wyższą wydajność, np. (Asembler, C, i C++).

Obiektowe lub strukturalne. -Obiektowe języki charakteryzują się tym że, spełniają paradymat OOP (object-oriented programming), czyli muszą posiadać:

1. Abstrakcje, co oznacza, że posiadają instancje reprezentujące obiekty rzeczywiste, które są strukturami danych, i które podlegają modyfikacji wielu funkcji.
2. Enkapsulację, która została opisana powyżej.
3. Polimorfizm, czyli specjalizację zachowań pod wspólnym interfejsem.
4. Dziedziczenie, rozumiane jako wsparcie polimorfizmu, czyli tworzenie obiektów hierarchicznie, które mają wyspecjalizowane funkcje, obrazuje to funkcja wirtualna.

Przykładem tych języków jest C++, Java, lub Erlang. Strukturalnymi są te języki, które nie spełniają OOP.

Funkcjonalne lub imperatywne (Typowość). Funkcjonalne języki są słabo, lub w ogóle nietypowane. Jest to programowanie oparte na implementacji funkcjonalności, bez określenia typów danych. Drugim rodzajem jest programowanie imperatywne, silnie stanowe i typowane. Funkcjonalność z reguły jest inna dla każdego z typów.

2.2.2 PHP

PHP jest językiem programowania. Charakteryzuje się przystępna, wspomnianą wcześniej krzywą uczenia. Dodatkowo jest bardzo popularny, a co za tym idzie, można zetknąć się z ogromnym wsparciem i pomocą w Internecie. Z oficjalnego źródła, które jest bardzo często aktualniane, oraz niespotykane jest tłumaczenie zawartości strony na tyle obcych języków - co stanowi o podejściu ludzi do programistów, Jak i ze stron innych użytkowników, dzielących się swoim doświadczeniem w tym języku. Ostatnio można zaobserwować pewną transformację tego języka ze strukturalnego, do obiektowego. W przyszłych latach wpłynie to korzystnie na popularność języka, ponieważ w większych projektach wymagania wobec używanego języka są rygorystyczne, a obiektowość to podstawa. Tę hipotezę o popularności, potwierdza artykuł na stronie w3techs.com - mówiący, że z przeprowadzonych badań, 81.9% stron używa PHP.

2.2.3 HTML

HTML (HyperText Markup Language) jest sposobem uporządkowania i wyświetlenia treści przez przeglądarki internetowe. W zrozumieniu tej definicji pomocny może być cytat z wydanej niedawno książki o HTML 5: [4] *Co rozumiemy poprzez HTML 5? - To pytanie na pozór może się wydawać niestosowne, choć istnieje coraz powszechniejsza tendencja wśród ekspertów standardu do wrzucania wszystkich nowych technologii webowych do jednego worka z napisem HTML 5. Na przykład spotkałem się z tym, że technologia SVG(Scalable Vector Graphics) określano jako „jedną z technologii z rodziny HTML5”, i pomimo że jest to niezależna specyfikacja graficzna organizacji W3C licząca już 6 lat teraz 14-dop. Aut.* Używa on znaczników do określania pozycji i formatowania elementów, takich jak `<h1></h1>`, oraz wielu innych. Wiele osób nie uważa go za język programowania, obecność słowa "language" wskazywałaby, że to jest język, ale słowo "Markup" jest tu kluczowe. Służy on jedynie do utworzenia informacji o prezentacji i strukturze treści. Potwierdza to fakt, że nie mówimy o kompilatorze HTML, a jedynie o interpreterze. Często jest używany jedynie do budowy obramowania, które następnie jest wypełniane, aby osiągnąć zamierzony cel. Przy budowie tego wstępnie szkieletu warto pamiętać o takich rzeczach jak dostępność. Wagę tego pokazują starsze elementy, oraz nowe możliwości w HTML 5. Ważna, w moim przypadku jest możliwość walidacji danych testowych już po stronie przeglądarki. W tej aplikacji będą dodawane znaczne ilości danych, dlatego powinny przedwcześnie zostać sprawdzone. Bardziej skomplikowane przypadki będą rozpatrzone przez administratora. Jednak należy do obowiązków projektanta, aby zminimalizować liczbę tych danych. Realizacja przedwstępniego sprawdzania danych jest dostępna po stronie samego kodu HTML w połączeniu z PHP. Możliwość połączenia z PHP, znacznie rozszerza dostępne funkcjonalności,

2.2.4 Połączenie CSS + HTML + JavaScript

Połączenie HTML + CSS + JS jest odpowiedzialne za znaczącą większość stron internetowych. Dlatego, dopiero po poznaniu przyjmniej trzech technologii tego typu możemy przystąpić do projektowania pełnej strony WWW. Zagadnienie tego połączenia

dobrze opisał Jon Duckett - popularny autor książek o HTML i CSS [4]: *Wszystkie witryny są tworzone przy użyciu języków HTML i CSS jednak systemy zarządzania treścią, platformy blogowe oraz platformy do prowadzenia handlu elektronicznego często uzupełniają tę parę o dodatkowe technologie*. Cytat ten mówi, o tym że wybór trzeciej technologii jest opcjonalny, jednak najczęstszym wyborem jest uzupełnienie o JavaScript. Kwestia samego HTML została opisana powyżej.

CSS (Cascade Style Sheet) to plik lub pliki, tak zwane „arkusze stylów”, w których zapisane jest to, jak poszczególne elementy strony HTML będą wyglądać. Oczywiście możemy to rozwiązać w aktualnie pisanej stronie, jednak rozwiązywanie tego jako oddzielny plik .css jest bardziej efektywne w większych projektach, ponieważ zwalnia nas z obowiązku pisania tego samego kodu w wielu plikach, a jedynie linkujemy nasz arkusz stylów. Rozwiązanie to również jest bardziej eleganckie, kod tej samej strony składa się z mniejszej liczby wierszy, przez co jest bardziej czytelny. Modyfikuje poszczególne elementy różnymi parametrami, takimi jak rodzaj pozycjonowania elementu na ekranie względem innych, jego rozmiary, kolory i inne. Wybierając tą technologię ważne jest opanowanie i wykorzystanie nowych wartości, które są zawarte w nowym standardzie CSS3

JavaScript, jako jedyny z tej trójki może się poszczycić mianem języka. Z uwagi na możliwość przyporządkowania pewnych paradymatów programowania można stwierdzić, że jest to język skryptowy, wysoko poziomowy, imperatywny oraz funkcjonalny. Sama możliwość wskazania na te dane, jest dowodem na to, że nie jest to jedynie technologia. Służy nam do zaimplementowania wielu funkcji, które zwyczajnie nie są do osiągnięcia za pomocą samego HTML. Można ten związek porównać do czterech poziomów znaku, z przedmiotu Komunikacja Człowiek-Komputer. Jeżeli założymy że HTML pełni funkcje pragmatyczną, i odpowiada na pytanie "Coe", CSS pełni funkcję syntaktyczną i semantyczną, odpowiada na pytanie "W jaki sposób?", to w tym przypadku JavaScript spełnia funkcję empiryczną, odpowiada na pytanie "Jakimi metodami?".

Wybór tego połączenia z wielu powodów będzie efektywny. Są to popularne technologie, co pozwoli mi na skorzystanie z wielu materiałów dostępnych w sieci. Technologie te niezwykle szybko się rozwijają, wystarczy spojrzeć na to ile nowości jest w HTML5 i CSS3. To połączenie stało się standardem, w jakim budowane są strony internetowe. Świadczy to o tym, że w większości ogłoszeń poszukujących programisty - znajomość tych technologii jest co najmniej mile widziana. Natomiast w ogłoszeniach typu: "Poszukujemy Grafika", "Front-End Developer", wymagane jest już płynne posługiwanie się wspomnianymi tymi technologiami, które możemy rozpatrywać dwójako. Z jednej strony są one niezwykle rozwinięte, mają częste aktualizacje, wspaniałą "krzywą uczenia" - pojęcie to jest opisane wyżej. Nie można tego powiedzieć o technologii, która jest mało popularna. Jednak z drugiej strony wiele osób może stwierdzić, że miało z nią kontakt, ale niewiele może poszczycić się doskonałym opanowaniem. Można stwierdzić, że etap doskonalenia jest bardzo rozległy i niewielu programistom udaje się go przejść - co jest znane z C, C++, i Javy.

2.2.5 ASP.net

Asp.net to otwarty framework, oraz wiele narzędzi, dostarczonych przez firmę Microsoft. Z uwagi na to, że zarządza nim poważna firma, jest on często aktualizowany. Z każdą kolejną wersją rośnie numer wersji a nazwa wskazuje, że jest przystosowany do pisania stron internetowych, ponieważ ASP jest rozwijaną jako Active Server Pages. Pisane są w nim aplikacje webowe, dynamiczne strony internetowe, i serwisy internetowe. Zaletą dla mnie jest to, że na Wydziale Informatyki był przedmiot, przybliżający tą tematykę, tak więc mam pewne doświadczenie w pisaniu w tym języku. Za tym wyborem przemawia jego modularność, która pokrywa się z wcześniej zadeklarowanym projektem za pomocą Domain-Driven Design. Poprawia to co jest najważniejsze w procesie projektowania, czyli komunikację między pojedynczymi komórkami całego zespołu deweloperskiego. Daje to użytkownikowi końcowemu poczucie spójności całej aplikacji, co ostatnio zanika, a co jest tym, o co warto zadbać. Ważną cechą ASP.net jest łatwość rozpoczęcia, ponieważ interfejs jest zbliżony do środowiska znanego większości użytkowników, czyli do systemu Windows. Świadczy o tym m.in. sposób funkcjonowania przycisków oraz opisów w ASP.net.

2.2.6 Ruby

Ruby to stosunkowo nowy język. Ma on możliwości porównywalne z Perlem, jest to język w pełni obiektowy. Od niedawna cieszy się znaczącą popularnością, ze względu na popularny Framework o nazwie Ruby on Rails. W ostatnich czasach na liście bestsellerów dwie najpopularniejsze książki, w dziale programowanie, są o Ruby i Ruby on Rails.

Pierwszym z powodów, które skłaniają mnie do tego wyboru jest efektywność tego języka. Najczęściej spotykanym zdaniem o tym języku jest „Preferuje konwencje ponad konfiguracją”, co oznacza, że nauczenie się konwencji zaoszczędzi nam czas, który w innym przypadku byłby marnotrawiony na dostosowywanie poszczególnych funkcji do otrzymanych elementów.

Dodatkowo Ruby był tworzony z myślą o konwencji DRY (Don't Repeat Yourself), czyli poprzez użycie tego języka zyska się wiele czasu na implementacji.

Drugą przesłanką przemawiającą za tym wyborem jest fakt, że to stosunkowo młody język. Pierwsza wersja została wydana w 2005, natomiast teraz (po 12 iteracjach) mamy wersje 4.2. W 2013 mieliśmy pojedyncze uaktualnienie, natomiast w 2014 mamy dwie, co świadczy o rosnącym zainteresowaniu tym językiem.

Trzecim faktem potwierdzającym tę tezę jest współpraca. Nie tylko z wieloma architekturami tworzącymi strony WWW, których nie da się pominąć przy implementacji takiej aplikacji - jak HTML, CSS, i JavaScript. Cecha ta znacząco pomoże przy projektowaniu GUI aplikacji. Z założeń wstępnych wynika, że ta aplikacja musi posiadać wsparcie dla baz danych. Współpraca Ruby z wieloma bazami danych takimi jak: MySQL, PostgreSQL, SQLite, Oracle, SQL Server, DB2, lub Firebird. jest cechą kompatybilności, która przewija się w innych technologiach, ale tutaj wyjątkowo zwiększa atrakcyjność tego wyboru.

Czwartym faktem przemawiającym za wyborem tej technologii, to możliwość korzystania z wtyczek, których użycie uprości wiele zadań; takich jak logowanie, dodawanie nowego punktu gastronomicznego, i ocena.

Piątym faktem są nowe możliwości, który daje Framework Ruby on Raila. Mianowicie możliwość wykorzystania jednej z najbardziej sprawdzonego i popularnego wzorca architektury MVC(Model View Controller), lub innych.

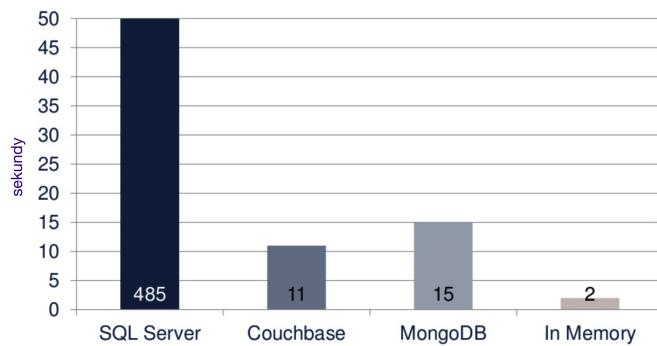
2.3 Technologie do implementacji BackEndu

2.3.1 Początek wyboru bazy danych

W każdym projekcie wybór bazy danych powinien być przeanalizowany. Najpierw trzeba zadecydować o typie, ponieważ ostatnio powstały nowe rodzaje realizacji koncepcji bazy danych. W tym przypadku jest to SQL.

Wybór ten wynika głównie z analizy systemów zarządzania bazami danych. Następnie wybór dotyczy konkretnej bazy, pod uwagę wzięto cztery: PostgreSQL, MySQL, SQLite, oraz Microsoft SQL Server. Do implementacji tego systemu najważniejszym czynnikiem wyboru bazy jest efektywność, dlatego brane pod uwagę są wyniki serializacji baz danych. Precyzując, jest to pewien plan na optymalizacje bazy danych, a w porównanie te ma na celu wybrać bazę do tej implementacji. Można mieć pewien zamysł na aplikacje, który znacznie zaoszczędzi czas, oraz zasoby. Nie każdy wpis musi posiadać pole, a jedynie różnice od domyślnych parametrów. Przykładowo istnieją trzy PG, w jednym nie można palić, w drugim można, a w trzecim tylko na zewnątrz. Ten pierwszy, nie będzie miał pola „palenie”. Drugi i trzeci, będą miały je zaznaczone innymi cyframi. W tym rozgraniczeniu zaoszczędzona została pamięć, która w innym przypadku byłaby poświęcona na wiele krotek w bazie, które miałyby w polu „palenie” NULL. Ponadto podejście to korzystnie wpłynie na szybkość odczytu danych na zapytanie „czy w PG dopuszczalne jest palenie i w jakim stopniu”. Wprawdzie użycie cyfr zamiast wartości boolowskich może bardziej obciążyć bazę, ale wpisywanie długiej wartości string z opisem wymaga znacznie więcej pamięci. Wartości boolowskie dają jedynie tę zaletę, a użycie cyfr daje nam stopniowanie przyzwolenia na palenie. Poniżej przedstawione są wyniki serializacji dla poszczególnych typów baz danych:

Rysunek 5: Porównanie wyników serializacji.



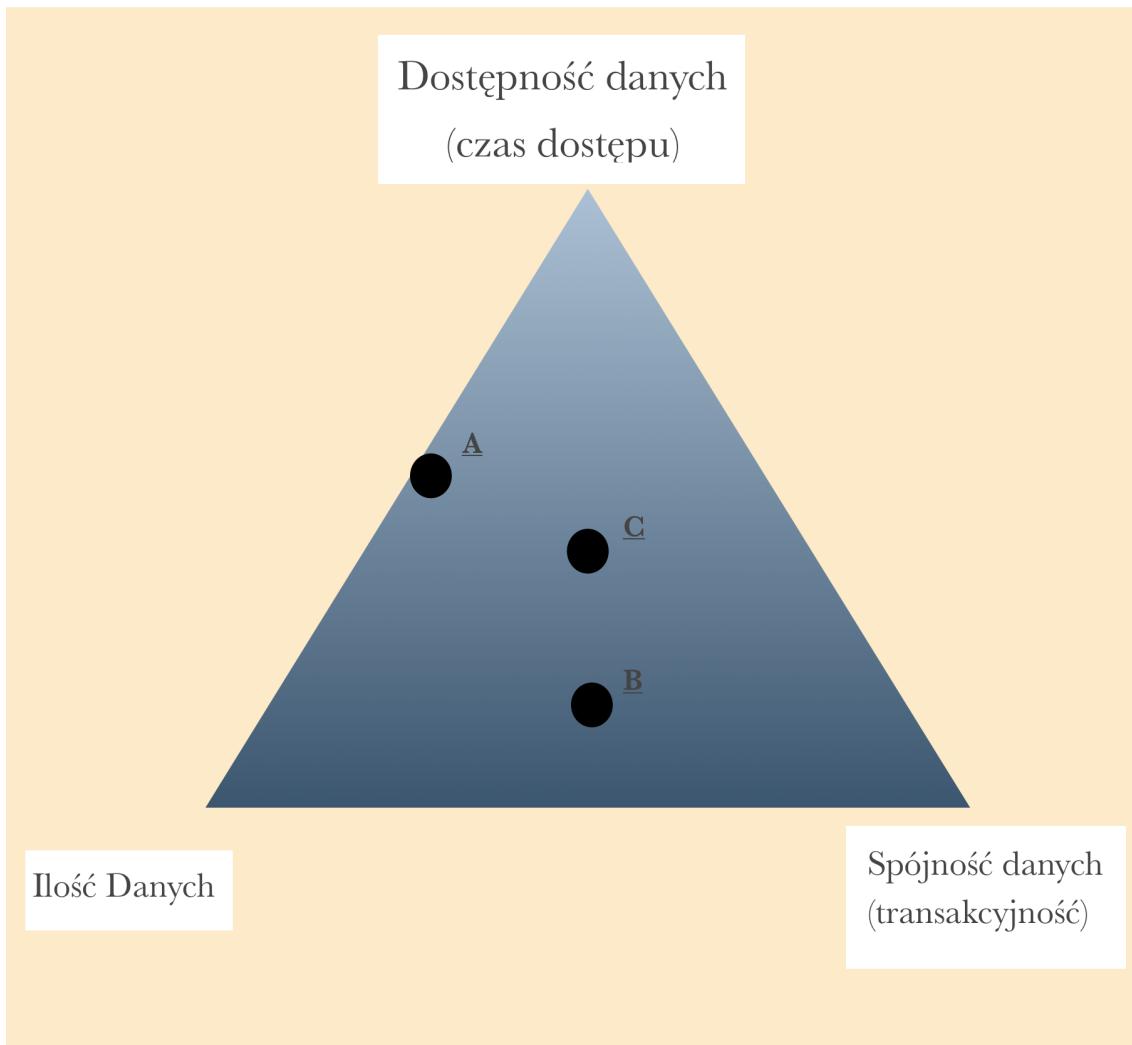
Źródło: google.pl (Dostęp: 28/01/2016)

Wyniki te pokazują jak wydajne są bazy sql przy użyciu serializacji. Z tego powodu cały wybór ograniczy się do baz tego typu.

2.3.2 SQL

SQL (Search Query Language) jest popularnym językiem obsługi baz danych. Fakt że jest to język deklaratywny, spełnia ważną rolę, z uwagi możliwość konstruowania bardzo skomplikowanych zapytań do bazy. Dla tych skomplikowanych zapytań potrzebne są relacje między wieloma tabelami. Cechą ta jest powodem skłaniającym do użycia relacyjnych baz danych. Przedstawiony rysunek obrazuje fakt złożoności tego wyboru:

Rysunek 6: Wybór bazy danych.



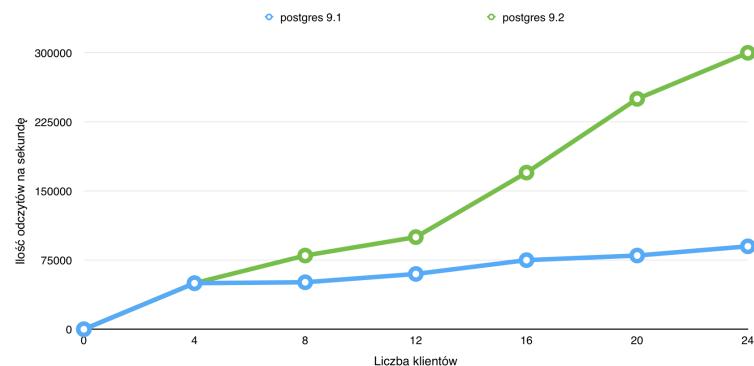
Źródło: opracowanie własne.

Na diagramie widać trzy kropki reprezentujące podejścia do zadania wyboru odpowiedniej bazy danych: A, B i C. Problem polega na tym, że w praktyce nie ma jedynego najlepszego rozwiązania. Podejście C jest zbalansowane, jednak trzeba ten wybór bardziej dostosować do projektowanego systemu. Przykładowo jeżeli projektujemy bazę danych dla aplikacji systemu dużego banku, najważniejsze są transakcyjność i ilość danych (podejście B). Jeżeli baza ma obejmować wybór PG, to najodpowiedniejszym rozwiązaniem jest to reprezentowane przez A. Wynika to z tego, że wymagana jest baza z szybkim dostępem i odpornością na dużą ilość danych.

2.3.3 PostgreSQL

PostgreSQL jest znanym systemem RDBMS. Tym co wyróżnia go od innych jest jego długa obecność na rynku, która nie oznacza stagnacji, a raczej stabilność (dotychczas ukazały się 24 wersje systemu zarządzania bazami danych PostgreSQL). Częścią tak dobrego wyniku jest jego zdolność przystosowania się do nowych warunków. Posiada on wiele podstawowych typów danych (jak INT, BOOL, Enum i inne), oraz daje nowe możliwości, jak castowanie, konwertowanie, przeładowane operatory, itd. Ciekawym przykładem tego może być informacja geograficzna z projektu PostGIS. Łatwość integracji z PHP, to coś, na co warto zwrócić uwagę podczas wyboru bazy danych. Tę integrację potwierdza istnienie narzędzia administracyjnego phpPgAdmin. Zaletą dla nowo powstałych firm jest to, że system ten jest realizowany jako OpenSource. Już sam ten fakt może być dwuznacznie interpretowany, ale trzeba przyznać, że jego obiektowość i możliwości modyfikacji w kodzie przemawia za jego wyborem. Odznacza się on również szerokim zakresem dostosowania, od niewielkich prrescia, składających kilka zapytań do bazy danych, aż do międzynarodowych konglomeratów, które tworzą bardzo zasobochłonne kwerendy. Poniższa grafika ilustruje różnice w czasach odczytu między innymi wersjami PostgreSQL:

Rysunek 7: Porównanie prędkości dwóch ostatnich wersji PostgreSQL.



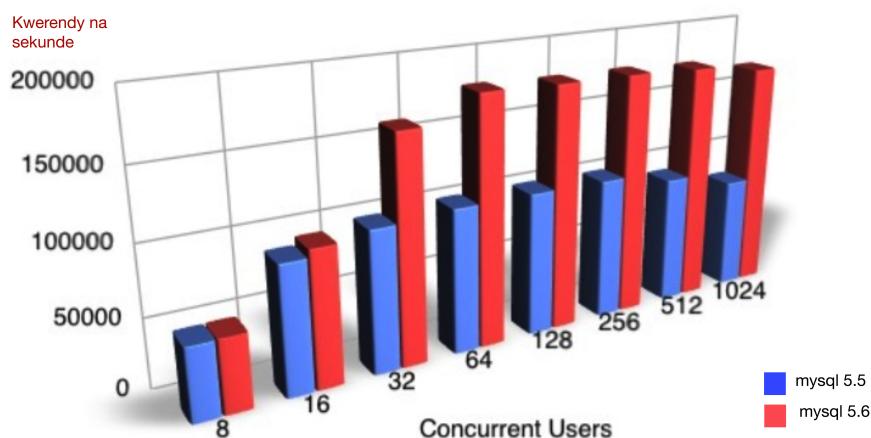
Źródło: Opracowanie własne na podstawie danych z Internetu.

Wynika z niego, że nowsza wersja znacznie wyprzedza poprzednika pod względem wydajności. Jest to fakt na który warto zwrócić uwagę, ponieważ w tym projekcie baza danych będzie zawierała wiele wpisów.

2.3.4 MySQL

Definicja pojęcia MySQL jest obecna m.in. w książce Larry'ego Ullmana [5]: *MySQL jest najpopularniejszą na świecie i, jak twierdzą niektórzy, najlepszą bazą danych o otwartym kodzie. W rzeczywistości, MySQL staje się coraz silniejszym konkurentem dla komercyjnych potęg, takich jak Oracle i Microsoft SQL Server. Za opracowanie i rozwijanie oprogramowania odpowiada szwedzka firma MySQL AB. MySQL jest systemem zarządzania bazami danych (DBMS - Database Management System), przeznaczonym dla relacyjnych baz danych (RDBMS - Relational DBMS). Z technicznego punktu widzenia MySQL stanowi aplikację zarządzającą plikami, zwanymi bazami danych. Powszechnie jednak terminu „baza danych” używa się w odniesieniu do obu tych elementów. Baza danych jest po prostu zestawem wzajemnie powiązanych danych w postaci tekstu, liczb czy plików binarnych, które są przechowywane i zarządzane przez DBMS.* Za wyborem tego systemu przemawia jego popularność i łatwość w użyciu. Najważniejsza w tym systemie jest prosta, która jest uzyskana przy pomocy dobrych narzędzi, takich OpenSource oraz phpMyAdmin. Wielu użytkowników miało z nim kontakt przez pakiety zgrupowane, jako odmiany 'AMP', z przedrostkiem; LAMP, MAMP, XAMP i inne. Fakt dobrego rozpowszechnienia wpływa pozytywnie na zjawisko skalowalności. Tak dystrybuowane pakiety są popularne w mniejszych firmach, co nie oznacza mniejszego wsparcia dużych przedsiębiorstw. Istnieją różnego rodzaju biblioteki wspierające MySQL do C, C++, a w tym przypadku pomocne będą biblioteki do PHP. MySQL jest flagowym przedsięwzięciem firmy Oracle, i dlatego jest stale rozwijany. Potwierdzeniem tej tezy jest grafika pokazująca różnice między starszą a nowszą wersją.

Rysunek 8: Porównanie dwóch wersji mySQL.



ORACLE

Źródło: oracle.com , 11.01.2016

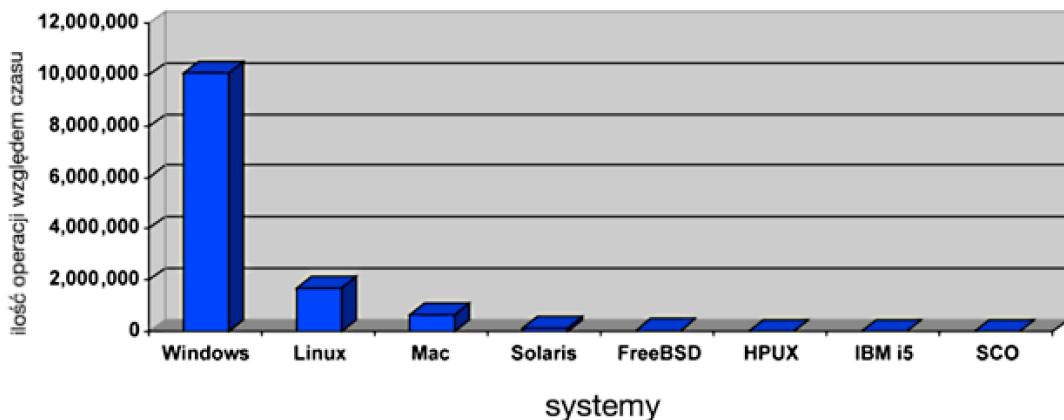
2.3.5 SQLite

SQLite jest bazą opartą na języku SQL, którą wyróżnia lekkie obciążenie systemu. Jest to zwykły plik binarny, a nie pełne oprogramowanie serwerowe. Podstawowa biblioteka oraz klient administracyjny baz danych SQLite zostały napisane w języku C. To co wyróżnia SQLite na tle innych, to konieczność implementacji na tej samej maszynie z uwagi na brak serwera. Przez to, że ten system nie potrzebuje znacznej ilości zasobów, jest w szczególności wykorzystywany do zastosowań, które wymagają od baz danych niewielkiego zużycia pamięci. Przykładem jest mobilny system Android firmy Google, czy system tego samego typu od Microsoftu. Warto tu wspomnieć o integracji z Ruby, ponieważ SQLite jest domyślną bazą danych dla tego języka. Najbardziej znane przeglądarki internetowe korzystają z pewnej jego odmiany. W systemach wbudowanych, często możemy spotkać tę bazę danych, ponieważ wspiera wiele z języków mało obciążających system. Przykładów jest wiele, a najbardziej znymi są: BASIC, C, C++, Free Pascal, Haskell, Java, JavaScript, Lua, Objective-C, Perl, PHP, Python, Swift, Visual Basic.

2.3.6 Microsoft SQL Server

Microsoft SQL Server jest zaprojektowany przez Microsoft, co raczej oznacza, że nie jest darmowy. Z tego również wynika, że jest to aplikacja działająca na innej maszynie tj. takiej, której celem jest przechowywanie i dostarczenie konkretnych rekordów. Aplikacja ta oferowana jest w wielu edycjach, co ma na celu spełnienie zapotrzebowan różnych grup. Czynnikiem przemawiającym za wyborem tej bazy danych jest integracja z ASP.net. Pomimo tych deklaracji, informacja[10] o tym, że ten system jest silnie zależny od platformy. W konsekwencji, na rodzimym systemie Windows będzie ona działać szybciej niż na innych. Jest to wiadomość ze strony konkurencji, ale deweloperzy zazwyczaj nie stosują praktyk zmyślania. Problem ten jest na tyle kluczowy, że dyskwalifikuje tę bazę z dalszego rozważania. Widać to na poniższej grafice:

Rysunek 9: Porównanie wydajność względem czasu na różnych platformach.



Źródło:Artykuł ze strony[10].

Grafika ta wskazuje na to, że ta baza danych działa najefektywniej w systemie Windows.

2.4 Podsumowanie wyboru technologii

Analiza obecnych wdrożeń ma na celu wyciągnięcie wniosku odnośnie technologii do tworzenia witryn internetowych. Najbardziej znane strony w internetowe, jak Facebook.com, Wikipedia.org, czy Twitter.com używają PHP[11] . Jest to informacja która stwierdza, że PHP jest popularnym wyborem w dziedzinie tworzenia aplikacji internetowych, oraz że radzi sobie z wielką ilością danych. Przy wyborze technologii warto również zwrócić uwagę na to, czym on skutkuje. Przykładowo zależność MS SQL od systemu Windows jest na tyle wysoka, że dyskwalifikuje to rozwiązanie. Powodem do użycia PHP jest łatwość połączenia pomiędzy nim a mySQL. Technologie są bardziej, lub mniej kompatybilne - trzeba to brać pod uwagę decydując się na specyficzną implementację.

3 Aplikacja wspomagająca ocenę usług gastronomicznych

3.1 Definicje wymagań funkcjonalnych

Wymaganiami funkcjonalnymi są akcje, które są udostępnione użytkownikowi przez dowolne aplikacje. W przypadku opracowanej aplikacji są nimi: „Przegląd PG dla zalogowanego użytkownika”, „Przegląd PG dla niezalogowanego użytkownika”, „Ocena PG”, „Dodanie PG”, „Usunięcie PG”, oraz „Zgłoszenie uwagi”. Operacje, która nie wpływają na bazę danych to „Zgłoszenie uwagi” i „Przegląd PG”. Pozostałe operacje, które są dostępne użytkownikom stworzonej aplikacji, mają różne poziomy dostępu:

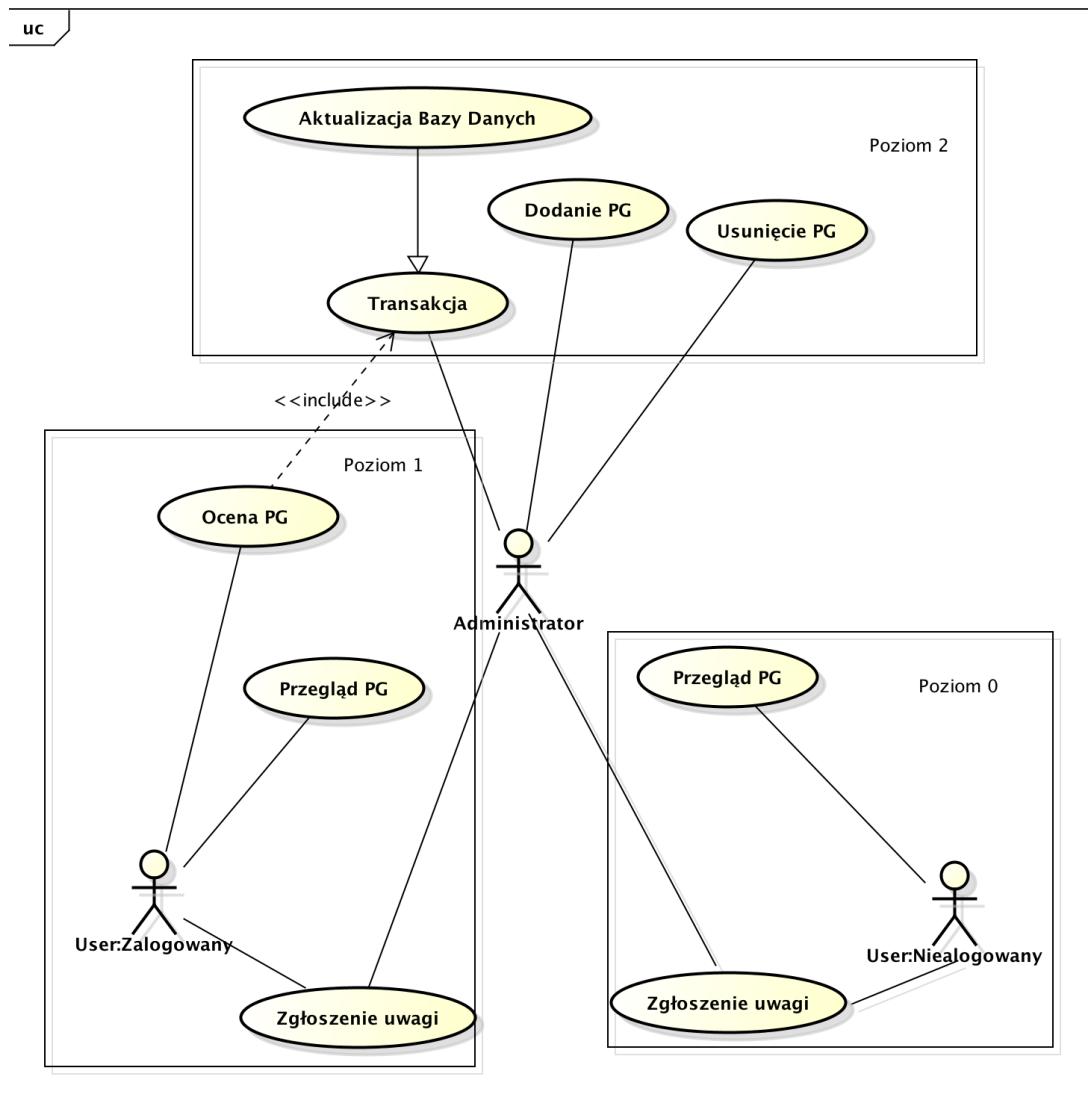
Poziom 0 Użytkownik, który nie jest zajogowany w aplikacji może jedynie zgłaszać żądania, oraz przeglądać listę punktów PG. Ma on dostęp do bazy ocen, nie są one jednak spersonalizowane do jego potrzeb.

Poziom 1 Użytkownik zalogowany, ma dostęp do tych samych informacji, do których uprawnienia ma użytkownik niezalogowany. Dodatkowo widziane oceny są zmodyfikowane według jego preferencji, oraz może on wystawiać oceny PG.

Poziom 2 Administrator posiada dostęp do wszystkich opcji. Może on dodawać PG, przeglądać PG, oceniać PG, usuwać PG oraz zgłaszać uwagi.

Te funkcjonalności są opisane na poniższym rysunku:

Rysunek 10: Przypadki użycia aplikacji.



Źródło: Opracowanie własne.

3.1.1 Poziom 0

Kiedy niezalogowany klient odwiedzi stronę, to ma możliwość wyboru trzech opcji: przeglądania, zgłoszenia uwagi, oraz rejestracji. Pierwsza funkcjonalność, przeglądanie ocen punktów jest najważniejszą z czynności, jakie ten program udostępnia - dlatego dzieli się na dwie części: dla zalogowanego użytkownika, oraz dla niezalogowanego. Jeżeli użytkownik jest zarejestrowany w systemie, przy prezentowaniu ocen punktów, uwzględniane są wagi użytkownika (wartości liczbowe, które podał przy rejestracji). Druga funkcjonalność - czyli „zgłoszenie uwagi” polega na podaniu: imienia, nazwiska, emaila, oraz treści uwagi, jest to specyficzna funkcjonalność, ponieważ pośrednio przez nią wszystkie inne funkcjonalności są dostępne. Trzecia funkcja to rejestracja. Umożliwia ona zapisanie się do systemu poprzez wypełnienie dwóch obowiązkowych pól (login i hasło), oraz zmodyfikowania swoich pięciu osobistych wag dotyczących: ceny, muzyki, jedzenia, wystroju, oraz różnorodności menu (w zakresie 1-5),

3.1.2 Poziom 1

Ocena wystawiona na poziomie pierwszym jest poprawniejsza od ocen wystawionych przez użytkowników którzy nie są zarejestrowani w systemie. Wygląda ona następująco: użytkownik wybiera nazwę punktu, a następnie wystawia ocenę w pięciu kategoriach - wystrój, muzyka, smak jedzenia, różnorodność menu, cena, oraz ewentualny komentarz. Ocena wystawiana jest w sześciostopniowej skali, aby nie zezwolić na bezmyślny wybór średniej. W ocenie ważne jest, aby wyłapać próby dodania fałszywych recenzji, wystawionych przez konkurencję danego punktu. Dlatego, na takie przypadki wyczulony jest skrypt, który nadzoruje działanie systemu. Kiedy akcje są poprawne składniowo i przechodzą systemową wstępna validację, zapytanie zostane zadane. Jednak kiedy zapytanie w jakikolwiek inny sposób jest niepoprawne, zostaje przekazane do tzw. „poczekalni”. Kiedy zbierze się w niej odpowiednio dużo zapytań, wysyłane są do administratora.

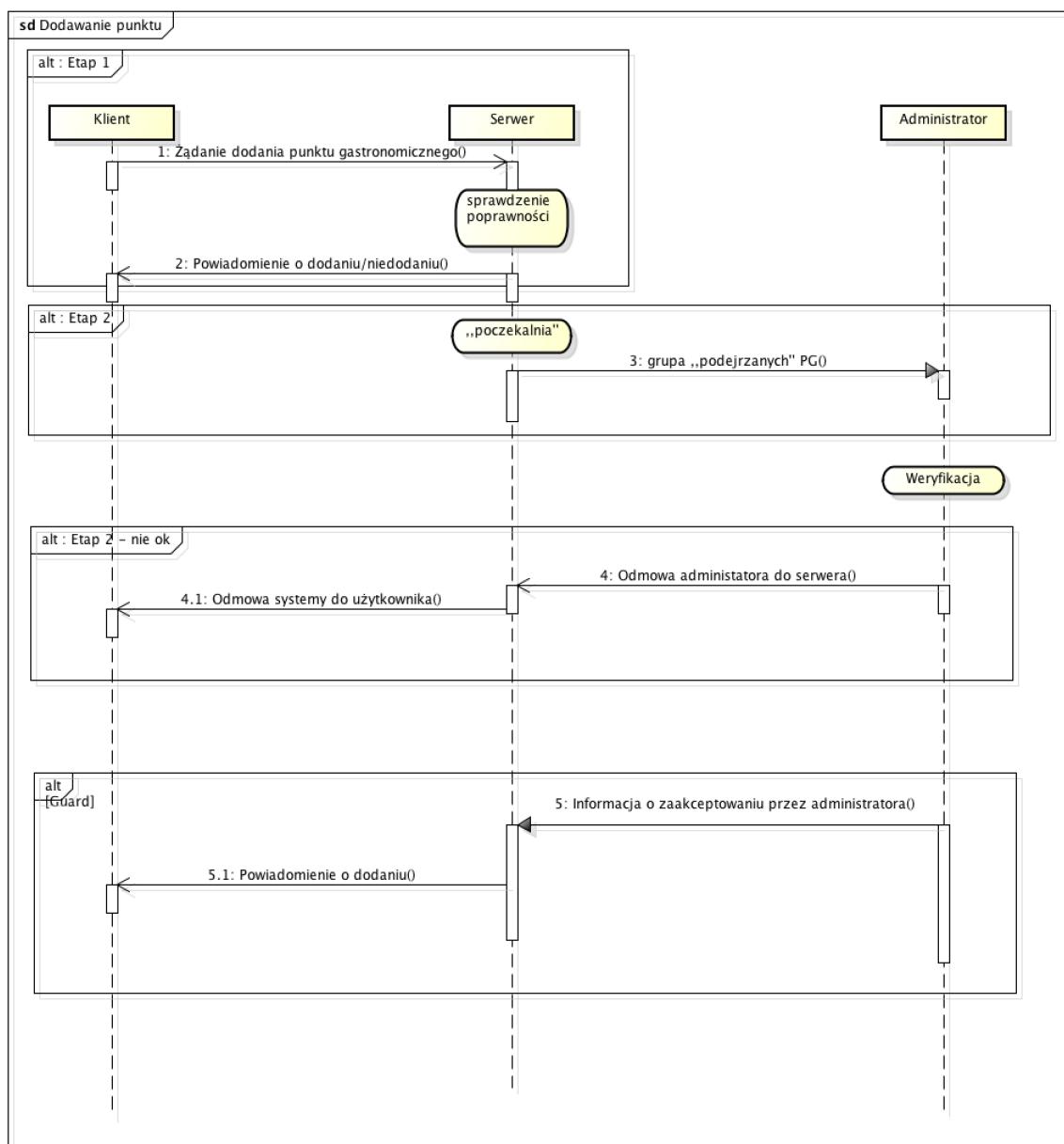
Ocena PG jest najważniejszym działaniem aplikacji wspomagającej ocenę punktów gastronomicznych. Jest ona krytyczna dla poprawnego działania całości, dlatego została podzielona na następujące części:

- Ocena przez użytkownika 1-1-n - jeden użytkownik daje jedną opinię, jednemu punktowi, wiele razy. Nie zakładam tu możliwości modyfikacji, ponieważ zwiększy to prawdopodobieństwo długich, i nieczytelnych komentarzy. Na przykład jeden zły "Mięso było suche", a drugi dobry "Zmieniono szefa kuchni i teraz mięso jest dobre". Daje to więcej informacji niż niewidoczna dla innych użytkowników zmiana złego na dobry komentarz.

- Ocena tylko dla zalogowanych - jest to tak zrealizowane, aby wyeliminować automaty. Wpływają one w negatywny sposób na bazę, publikując wiele informacji, co zniekształca dane. Dodatkowo mogą one generować tak wielki ruch, na który serwer może nie być przygotowany.

- Wspominany algorytm weryfikujący, uczy się od administratora. Ponowne zgłoszanie tego samego problemu (jeżeli administrator oznaczy problem jako bezpieczne działanie), jest oznaczane jako poprawne, i przy następnych wystąpieniach nie jest już wyróżnione. Ten punkt mówi o tym, że algorytm musi być stale udoskonalany.

Rysunek 11: Dodawanie opinii o punkcie gastronomicznym.



Źródło: Opracowanie własne.

3.1.3 Poziom 2

Drugi poziom uprawnień dostępny jest jedynie dla administratora aplikacji. Poziom ten zawiera dwie dodatkowe funkcjonalności: dodawanie PG oraz usuwanie PG. Dodawanie punktu wymaga od administratora kilku pól, takich jak: nazwa, miasto, ulica, i godzina otwarcia/zamknięcia (jeżeli PG charakteryzuje się nieregularnymi godzinami, dodający zaznacza odpowiednie pole) jest to czynność, od której zależą inne funkcjonalności, z tego powodu powinno być bardziej zabezpieczone. Zawartość bazy jest wyświetlana jako tabela, po wybraniu id, wyświetlane jest powiadomienie, po którego zaakceptowaniu punkt jest usuwany. Obie funkcjonalności mogą być nadużyte, dlatego są dostępne wraz z największym poziomem uprawnień. Wszystkie informacje o PG, które nie zostały opublikowane, a które okażą się interesujące można zasugerować przez przycisk „Zgłoszenie żądania”. Przykładowo, jeśli użytkownik uzna, że aplikacja powinna zawierać informacje o potrawach zawierających alergeny, to może zakomunikować to, poprzez użycie tej funkcji.

3.2 Cechy opracowanej aplikacji

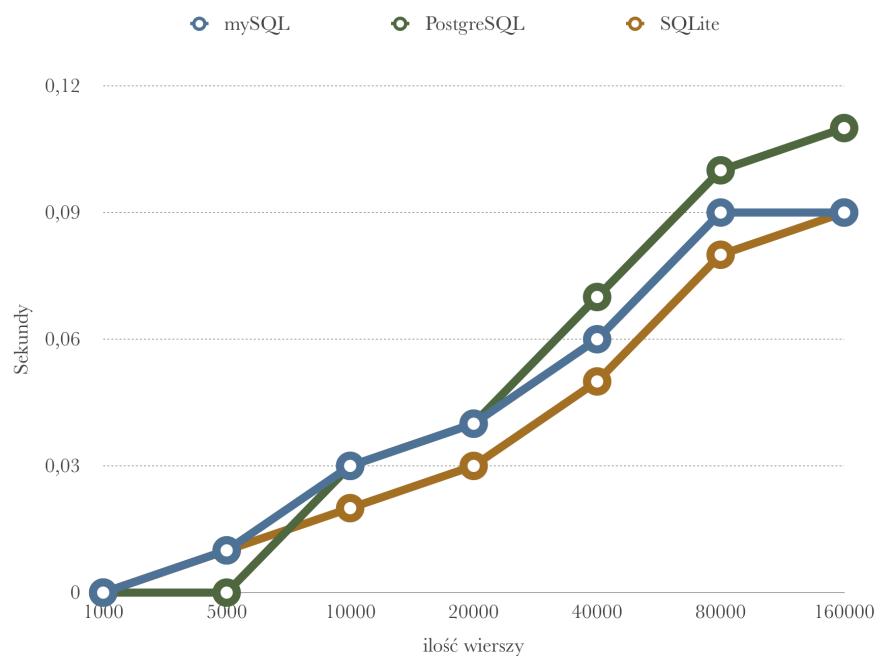
Opracowaną aplikację wyróżnia rozmiar – jest planowana jedynie na terenie Polski, co może być wadą, jak i zaletą. Komentarze dla punktów w innych aplikacjach wystawiane są w wielu językach, przez co tracą w kilku istotnych kwestiach. Fakt ten negatywnie wpływa na spełnienie cech projektowania w metodologii „Plastera miodu Morville'a”. Tracą na tym wszystkie cechy opisane w tej metodologii, takie jak: przydatność, łatwość w użyciu, łatwość w odnalezieniu, dostępność, wiarygodność informacji i atrakcyjność wizualna. Opracowana Aplikacja internetowa wspomagająca ocenę punktów gastronomicznych, jest w pełni polskojęzyczna. Angielski jest niezwykle popularny w naszym kraju wśród młodszej pokolenia, nie zapominajmy jednak o pozostałych członkach społeczeństwa. Kolejnym aspektem jest estetyka - komentarze w wielu językach nie przedstawiają się dobrze. Prezentacja całego systemu jest na pierwszym miejscu. Powyższe cechy pokazują kierunek rozwoju prezentowanej aplikacji. Inne programy tego typu, zawsze próbują zająć się dodatkowymi funkcjonalnościami, niż tylko oceną PG. Stara prawda mówi, że „Jak coś jest do wszystkiego, to jest do niczego”. Dodatkowym elementem wyróżniającym Aplikację internetową wspomagającą ocenę usług gastronomicznych jest rozwój użytkownika. Wszyscy ludzie chętniej wykonują daną czynność kiedy zostają za nią nagrodzeni. Dlatego w tej aplikacji zbieranie są punkty „doświadczenia”, za wystawianie komentarzy. Będzie to podwójnie korzystne, ponieważ zmotywuje to odbiorców aplikacji do wystawiania ocen, oraz doda to dane do dalej opisanego algorytmu i wyróżni bardziej aktywnych klientów.

3.3 Architektura opracowanej aplikacji

3.3.1 Podsumowanie wyboru technologii BackEndowych

W celu wyłonienia dobrego rozwiązania dla prezentowanej aplikacji, wykonano eksperyment, który testuje ważną dla takiego systemu metrykę, tj. czas dostępu. Eksperyment ten został przeprowadzony w taki sposób, że zostały utworzone coraz większe tabele, w trzech systemach (mySQL, PostgreSQL, i SQLite). Następnie wyniki czasowe wykonania kwerendy zostały naniesione na grafikę. Grafika ta dostarcza informacji o efektywności. Zapytanie ma postać, `SELECT * FROM PG_10 WHERE palenie = 1 and psy = 1 HAVING (cena_obiadu) < 100.` Przykładowa baza zawiera, nazwę, miasto, ulicę, średnią cenę, informacje o tym, czy można palić, oraz czy można przyjść z psem. Informacje te są wystarczające do wyboru odpowiedniej bazy danych. Choć informacje w końcowej aplikacji mogą być bardziej rozbudowane, to eksperyment został przeprowadzony dla najprostszych zapytań. Chcąc uzyskać pełne dane można ekstrapolować pozyskane wcześniej dane. Wyniki te mają za zadanie jedynie porównać czas dostępu między wybranymi systemami RDBMS. Na poniższym diagramie przedstawione są wyniki tego eksperymentu.

Rysunek 12: Wyniki eksperymentu.



Źródło: Opracowanie własne w oparciu o wiedzę z Internetu.

Wyniki te świadczą o nieznacznych różnicach pomiędzy wspomnianymi systemami. Dostarcza to informacji o tym, że nie ma potrzeby wybierania bazy danych przez pryzmat szybkości dostępu. W takim razie do realizacji tego projektu odpowiednią jest baza MySQL. Wynika to z tego, że jest ona automatycznie dodawana na wielu serwisach hostinguowych. Może to zredukować koszt wdrożenia.

3.3.2 Podsumowanie wyboru technologii FrontEndowych

Poniżej przedstawiono tabelę subiektywnych wyników porównania różnych technologii. W wierszach znajduje się 5 kategorii, każda ma swoją wagę ($x?=?$, gdzie znak zapytania to liczba od 1 do 5.), oraz porównanie ogólne, czyli średnią ważoną. W kolumnach znajdują się nazwy porównywanych ośmiu technologii: PHP, HTML, ASP.net, Ruby, PostgreSQL, MySQL, SQLite, MS SQL Server.

Tabela 1: Pierwsza część tabeli wspomagającej wybór.

technologie \ parametry	Znajomość $x1=5$	Łatwość $x2=4$	Popularność $x4=3$	Krzywa uczenia $x3=2$
PHP	5	4	5	5
HTML	5	3	5	3
ASP.net	3	4	4	4
Ruby	1	5	3	5
PostgreSQL	4	4	3	2
MySQL	5	4	5	5
SQLite	1	4	3	2

Źródło: Opracowanie własne.

Tabela 2: Druga część tabeli wspomagającej wybór.

technologie	średnia ważona
PHP	5
HTML	4
ASP.net	4
Ruby	3
PostgreSQL	4
MySQL	5
SQLite	2

Źródło: Opracowanie własne.

Każda technologia w danej kategorii dostała ocenę (1-5), gdzie 1 to bardzo źle a 5 to bardzo dobrze.

Wyniki spersonalizowanego przydzielania ocen prezentują się następująco:

1. PHP i MySQL
2. HTML
3. ASP.net
4. PostgreSQL
5. Ruby
6. SQLite

Za wyborem końcowych technologii przemawiają trzy argumenty:

- ogólne wyniki powyższych rozważań,
- spersonalizowany wynik z tabelki,
- dotychczasowe doświadczenie.

Z powyższej tabeli oraz ze wcześniejszych analiz wynika, że najodpowiedniejszymi technologiami są PHP, MySQL, i HTML. Wybór tej pierwszej pociąga za sobą wybór HTML'a, ponieważ kod napisany w PHP jest i tak przetwarzany przez HTML. Te dwie technologie najłatwiej zintegrować z bazą MySQL.

3.3.3 Technologie w ujęciu Domain-Driven Design

Domain-Driven Design jest rodzajem modelowania, które intuicyjnie dzieli aplikację na części. Logika biznesowa aplikacji prezentuje się następująco:

-GUI jest zrealizowane za pomocą HTML, PHP, CSS, i JavaScript.

-Application layer specyfikuje to, jak FrontEnd komunikuje się z BackEndem. Jest to zrealizowane za pomocą PHP.

-Domain layer mówi o tym, gdzie i w jaki sposób będą rozwiązywane kwestie konfliktowe. Do tego zadania również użyto PHP.

-Infrastructure layer zajmuje się pojedynczymi obiektami.

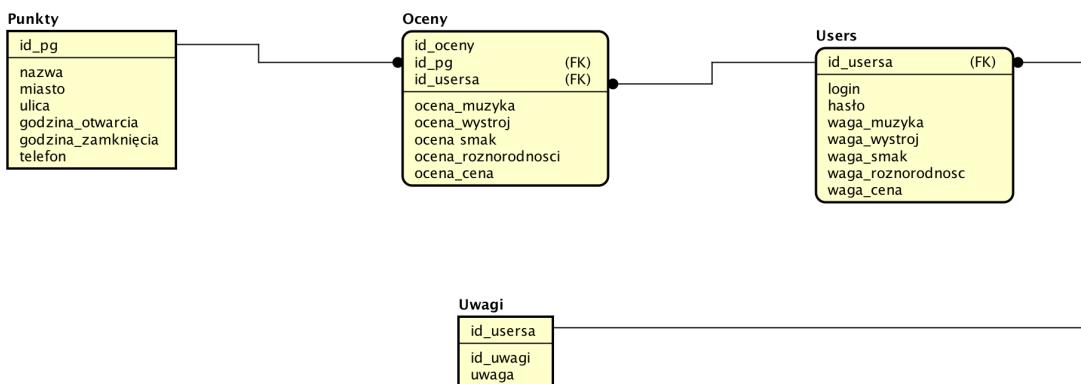
3.3.4 Struktura bazy danych

Na rysunku 13. przedstawione zostały cztery tabele, które powiązane są relacjami. Relacje te polepszają efektywność całej bazy danych. Znaczenie tych pól zostało opisane w podrozdziale 1.4.

Tablica „punkty” posiada sześć atrybutów: nazwa, miasto, ulica, godzina_otwarcia, godzina_zamknięcia, oraz telefon. Jej klucz główny jest kluczem obcym dla tabeli „Ocena”. Tabela „Ocena” jest najważniejsza w całym systemie/ Posiada ona trzy klucze: główny id_oceny, i dwa obce, id_pg oraz id_usera. Funkcja nadawania oceny przyjmuje od użytkownika informacje o pięciu wartościach – jak ocenił muzykę, wystrój, smak, i różnorodność menu. Informacje te przechowywane są w pięciu atrybutach. Wspomniany klucz

obcy dla tej tabeli (id_usera) jest kluczem głównym dla tabeli users, która ma pola: login, hasło i pięć znajdujących się wyżej wag. Na klucz główny tej encji referuje się kolejna tabela „uwagi”. Posiada ona id_usera, id_uwagi, oraz samą uwagę. Struktura bazy została zaprojektowana lokalnie. Przeniesienie jej sprowadziło się do użycia funkcji mysqldump, która tworzy zrzut do pliku .sql, zawierający kompletną strukturę bazy. Następnie został on załadowany do programu phpMyAdmin. Zrzut ten zawiera cztery wspomniane tabele oraz adekwatne wpisy.

Rysunek 13: Model bazy danych.



Źródło: Opracowanie własne.

Zzaprezentowano model bazy danych wykorzystany w aplikacji. W celu skrócenia czasu przeszukiwania, tabela została znormalizowana jedynie do trzeciej postaci normalnej. Model bazy jest w trzeciej postaci normalnej.

Pojęcie to wyjaśnia następująca definicja: [5] *Baza danych ma trzecią postać normalną, jeżeli jest zgodna z drugą postacią normalną i jeżeli jej każda kolumna, która nie jest kluczem, pozostaje niezależna od innych kolumn niebędących kluczami. Innymi słowy, wszystkie pola tabeli, które nie są kluczami, powinny być względem siebie niezależne.*

3.3.5 Struktura kodu

Cały projekt jest napisany w dziewięciu plikach .php. Zostały one napisane za pomocą edytorów vi i xcode, i przetestowane na lokalnym serwerze apache 2. Kod jest zrealizowany stukaturalnie z użyciem obiektów, które jedynie rozszerzają możliwości funkcjonalności. Sama struktura strony została stworzona przy pomocy kilku frameworków, m.in. bootstrap, który cieszy się wielką popularnością pośród wielkich serwisów. Są one zastosowane, aby skrócić proces pisania samej strony i pozwolić programistie skupić się na innych kwestiach, które nie są udostępniane w Internecie. Połączenie z bazą danych jest realizowane poprzez obiekt PDO (PHP Data Objects). Rozszerzenie to definiuje stały interfejs udostępniający kontakt z bazą danych w PHP. Wynika to z tego, że strukturalne metody nie są już zalecane. Dane logowania do bazy różnią się między maszyną lokalną a stroną WWW. Z tego powodu są zapisane w oddzielnym pliku .php, co ułatwia proces przeniesienia kodu. Do uzyskania przejścia przy stronie tytułowej i zegara przy dodawaniu punktu, użyto języka JavaScript i jQuery. Oby nie powtarzać tych samych instrukcji w kodzie samej strony, formatowanie zostało przeniesione do oddzielnego arkusza .css. Rozwiązanie takie udostępnia wiele interesujących selektorów, dzięki temu takie podejście jest prostsze, szczególnie kiedy elementy znajdujące się na stronie zostaną zaprezentowane w modelu DOM (Document Object Model). Konfiguracja była niezwykle szybka, wymagała jedynie wspomnianego zrzutu ze struktury bazy oraz przeniesienia plików klientem ftp.

3.3.6 Scharakteryzowanie procesu oceny

Na potrzeby aplikacji stworzony został dwuetapowy podział oceny. Niezalogowany użytkownik widzi jedynie średnie arytmetyczne z wielu ocen. Do celów prezentacyjnych, widoczne są jedynie trzy wpisy dla jednego punktu. Jest to zestaw sześciu ocen (1-6), na tej podstawie obliczana jest średnia arytmetyczna (kolor czerwony). Aplikacja przelicza oddzielne średnie w tle, a widoczne są pojedyncze wyniki oraz średnia arytmetyczna. Kiedy zalogowany użytkownik wyszukuje punkty, na ocenę końcową (kolor zielony) składa się średnia punktu (kolor czerwony), wymnożona ze spersonalizowanymi wagami użytkowników. W tabeli 3 i 4 przedstawiono proces przydzielania oceny. Podsumowując tą realizację procesu oceniania, wiele osób wspólnie buduje bazę ocen, która jest dostępna dla każdego.

3.4 Możliwość wdrożenia

Możliwości wdrożenia aplikacji są niezwykle zmienne i zależne od wielu czynników, Wybór hostingu jest złożony. Najważniejszym aspektem jest cena, ponieważ od niej zazwyczaj zależy cała reszta, dokładniej podział na darmowe i różne odmiany płatnych hostingów - z tego wynikają kolejne punkty. Przykładowo, wiadomo że adres aplikacja.pl będzie łatwiejszy w zapamiętaniu, niż aplikacja.xyz.zyx.com.pl - aby mieć prostszy adres url trzeba dodatkowo zapłacić. Następna sprawa to ilość udostępnianego miejsca przez hosting. W większości przypadków spotykane są opcje nieograniczonego miejsca, jednak warto na to zwrócić uwagę, że przy wyborze darmowej opcji dochodzi problem przy-

Tabela 3: Pierwsza część procesu przydzielenia oceny.

parametry \ oceny	osoba 1	osoba 2	osoba 3	średnia arytmetyczna
Wystrój	1	2	3	2
Muzyka	1	5	6	4
Ocena posiłku	4	3	2	3
Różnorodność menu	5	4	3	4
cena	5	6	4	5

Źródło: Opracowanie własne.

Tabela 4: Druga część procesu oceny.

parametry \ oceny	średnia arytmetyczna	wagi użytkownika	ocena końcowa
Wystrój	2	2	2
Muzyka	4	3	6
Ocena posiłku	3	5	8
Różnorodność menu	4	4	8
cena	5	1	3

Źródło: Opracowanie własne.

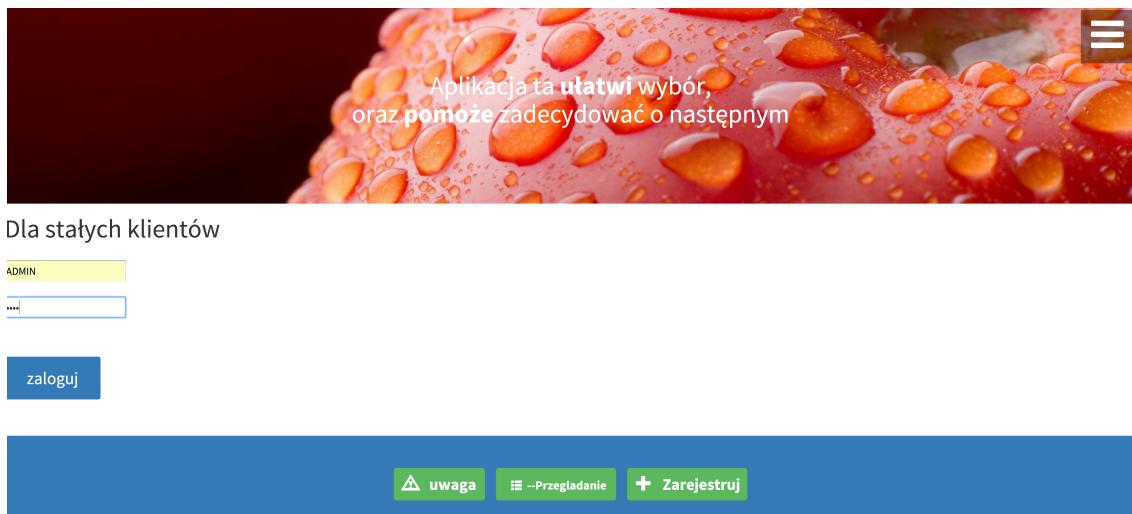
musowych reklam. Kolejna sprawa to rzeczy które muszą być wspierane, takie jak np. dostęp do bazy danych. Ta kwestia też jest zależna od ceny, ponieważ obsługa bazy może się odbywać przez aplikacje udostępniane przez hosting, lub połączenie zdalne. Teraz Aplikacja internetowa wspomagająca ocenę usług gastronomicznych wykorzystuje bazę MySQL, lecz w przyszłości inna technologia może okazać się potrzebna. Jest to ważny wybór, ponieważ dane zebrane w aplikacji są unikalne i jest potrzeba zadbania o ich bezpieczeństwo - warto wybrać redundantny hosting który kopiuje dane do różnych ośrodków gromadzenia danych. Instalacja aplikacji wymaga jedynie dostępu do klienta FTP - jednak dobrze mieć opcje przesyłania danych na hosting przez SFTP.

4 Testy i przykłady zastosowania opracowanej aplikacji

4.1 Prezentacja aplikacji

Kiedy strona została załadowana, można przejść do trzech funkcjonalności: uwagi, przeglądania, oraz rejestracji.

Rysunek 14: Strona widoczna dla niezalogowanego użytkownika.



Źródło: Opracowanie własne.

Alternatywnie, kiedy użytkownik posiada już konto może się zalogować wpisując login i hasło, np. ADMIN i 1234, lub USER 1234. W zależności od tego kto jest zalogowany widoczne są dwie wykluczające się możliwości.

Rysunek 15: Zmiany na stronie po zalogowaniu się użytkownika USER.

Zalogowano jako USER

wyloguj



Źródło: Opracowanie własne.

Rysunek 16: Zmiany na stronie po zalogowaniu się użytkownika ADMIN.

Zalogowano jako ADMIN

wyloguj



Źródło: Opracowanie własne.

4.2 Narzędzie użyte do testowania

Aby mieć pewność, że aplikacja działa poprawnie, trzeba przeprowadzić statyczne oraz dynamiczne testy najważniejszych funkcjonalności. Testy statyczne są zakończone sukcesem, ponieważ strona działa prawidłowo, co jest zaprezentowane poniżej

Przed właściwym testowaniem warto zaznaczyć, że testy zostały przeprowadzone na maszynie lokalnej. Aby mieć pewność, że aplikacja działa poprawnie, trzeba przeprowadzić statyczne oraz dynamiczne testy najważniejszych funkcjonalności. Testy statyczne są zakończone sukcesem, ponieważ strona działa prawidłowo, co jest zaprezentowane poniżej

Teraz zaprezentowany zostanie przebieg testu dynamicznego. Z uwagi na to, że przeprowadzenie testów składa się z części, w tym miejscu został zawarty opis całej procedury. Testy są przeprowadzone aplikacją JMeter. Jest to aplikacja open source napisana przy pomocy języka java. Pomaga to w wykonaniu danej funkcjonalności oraz zmierzniu wyników ich wykonania. Oryginalnie aplikacja JMeter została stworzona do testowania aplikacji webowych, jednak rozszerzyła swoją działalność. Apache JMeter może być wykorzystany do testowania statycznych jak i dynamicznych zasobów (Usług sieciowych (SOAP/REST), stron napisanych w dynamicznych językach - PHP, Java, ASP.net, c, obiektów Java, baz danych, zapytań do nich, serwerów FTP, i innych). Może być użyty do generowania dużego ruchu na serwerze, bądź w większych sieciach lub obiektach, aby przetestować ich wytrzymałość, lub przeanalizować parametry pod różnymi rodzajami obciążenia. Można tego użyć do graficznej analizy jakości lub przetestowania zachowania dowolnego serwera, skryptu, lub obiektu pod wielkim obciążeniem.

Apache JMeter umożliwia wygenerowanie ruchu i sprawdzenie parametrów za pomocą wielu różnych metod:

- Protokołów internetowych - HTTP, HTTPS.
- Protokołu wymiany plików FTP.
- Bazy danych poprzez JDBC.
- Protokołu dostępu do danych LDAP.
- Message-oriented middleware (MOM) przez JMS.

- Usług i protokołów pocztowych - SMTP(S), POP3(S) and IMAP(S).
- Baz MongoDB (NoSQL).
- Natywnych komend w języku bash, lub skryptów.
- Transmisyjnego protokołu TCP.
- Gwarantuje całkowitą implementację w języku JAVA.
- Pełny wielowątkowy framework, który zapewnia współbieżne próbkowanie przez wiele wątków podzielonych na inne grupy.
- Precyzyjnie zaprojektowany interfejs graficzny pomaga w szybszym projektowaniu planów, przez co skraca cały etap testowania.
- Analiza wyników w trybie offline.
- Mocno rozszerzalny rdzeń.
- Rozszerzalne możliwości próbkowania pozwalają na większą liczbę testów.
- Kilka statystyk o przebiegu testów może być pobrane za pomocą dodatkowych funkcji typu timer.
- Rozszerzenia wizualizujące dane, pozwalają znacznie rozszerzyć aplikacje, oraz spersonalizować wyniki.
- Funkcje pozwalające na dynamiczne wejścia dla testów.

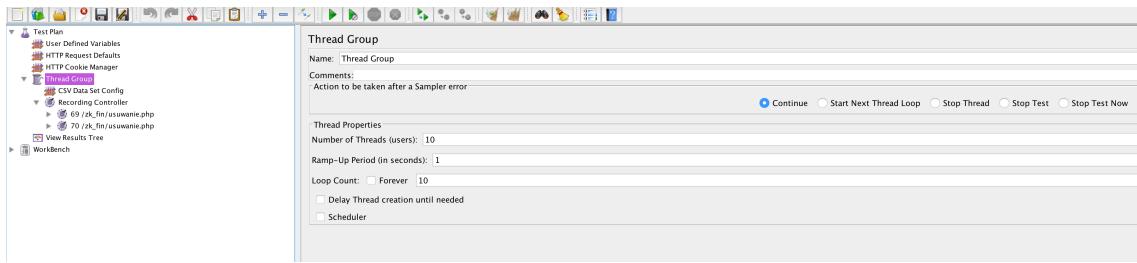
Wadą aplikacji JMeter jest fakt, że jej działanie nie jest zbliżone do przeglądarki internetowej. Przez inny serwer jest widziana jak wiele klienckich przeglądarek, jednak JMeter nie wykonuje wszystkich akcji wspieranych przez przeglądarki. Najbardziej znaczącym przykładem jest to, że JMeter nie wspiera języka Javascript, który jest często spotykany na stronach korzystających z HTML. W aplikacji znajdują się siedem funkcjonalności:

4.3 Przykłady stosowania aplikacji

4.3.1 Początkowy etap testów

Pierwszy etap to przygotowanie zapytania do programu JMeter. Wprawdzie jest to możliwe przez przekazanie pliku .xml, jednak aby lepiej to zaprezentować, proces ten jest pokazany w bardziej czytelnym graficznym interfejsie: Na rysunku widoczne są ustawienia odnośnie wątków. Najważniejsze parametry są ustawione na 10. Jest to liczba wątków, oraz liczba iteracji. Rezultatem będzie stu wirtualnych aktywnych użytkowników którzy są zidentyfikowani jako realni klienci.

Rysunek 17: Okno programu JMeter z podanymi danymi.

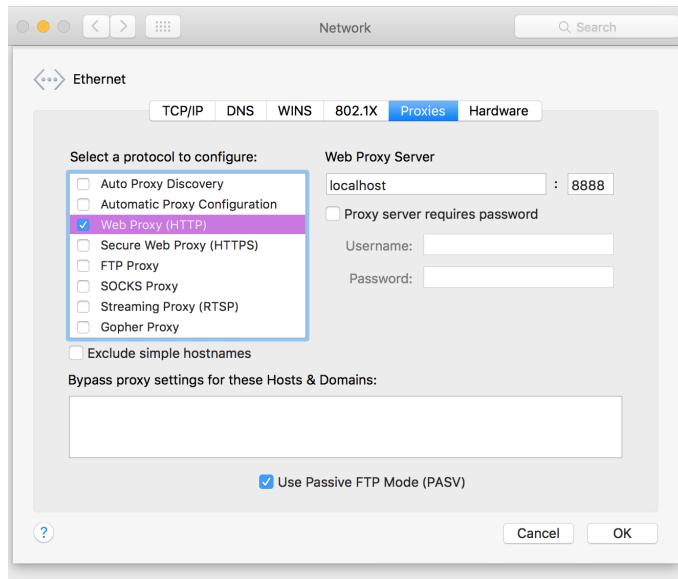


Źródło: Opracowanie własne.

4.3.2 Główny etap testów

Ważnym etapem testów jest konfiguracja. Stanowi ona punkt centralny całego procesu. Rynika to m.in. z tego, że zależą od niej rozmiar, oraz wygląd rezultatów. Konfigurację można przeprowadzić zarówno przez interfejs graficzny (nagrywanie akcji, oraz ręczną konfigurację), jak i poprzez skonstruowanie pliku xml i podanie za pośrednictwem linii komend do programu. Dla lepszego zrozumienia zostanie zaprezentowana wersja z nagrywaniem poprzez interfejs graficzny. Nagrywanie akcji zaczyna być możliwe po włączeniu serwera proxy na serwerze lokalnym(localhost) i porcie 8888. Po ustawieniu serwera

Rysunek 18: Ustawianie serwera proxy, zależne od systemu i przeglądarki.



Źródło: Opracowanie własne.

proxy, cały ruch internetowy przepływa przez kontroler o nazwie HTTP(S) Script Recorder. W taki sposób zarejestrowana jest pierwsza akcja, następnie możemy przejść do wspomnianych testów wielowątkowych.

4.3.3 Opis wyników testowania

Testy wielowątkowe mogą być niezwykle czasochłonne oraz mogą wymagać mniej więcej maszyny. Najpierw wymagane jest wybranie ścieżki dostępu do pliku, gdzie ma zostać zapisany rezultat. Po tych krokach możliwe jest uruchomienie aplikacji.

Rysunek 19: Wyniki zwrócone przez programu JMeter.

```
153151 1453668248285,72,70 /zk_fin/usuwanie.php,200
153152 1453668234321,14124,69 /zk_fin/usuwanie.php,
153153 1453668248343,48,69 /zk_fin/usuwanie.php,200
153154 1453668248374,55,70 /zk_fin/usuwanie.php,200
153155 1453668248337,58,69 /zk_fin/usuwanie.php,200
153156 1453668248287,75,69 /zk_fin/usuwanie.php,200
```

Źródło: Opracowanie własne.

Powyżej zaprezentowana jest jedynie część wyników, co w zupełności wystarczy, aby je zrozumieć. Wyniki pomiędzy testami różnych funkcjonalności są zbliżone, dlatego jedynie wyniki z dowolnej funkcjonalności zostaną zinterpretowane - w tym przypadku usuwanie punktów gastronomicznych. Pierwszy wniosek to, że strona wytrzymała pod takim obciążeniem. Kolejny mówi o tym, że wszystkie zapytania miały kod potwierdzenia 200, co znaczy że komunikacja była prawidłowa. Trzecim wnioskiem jest informacja o ramkach. Ich wielkość to 15974 bajtów (ok. 15MB), mimo to docierały z prędkością około trzech sekund.

Podsumowanie

Celem prezentowanej pracy było stworzenie aplikacji wspomagającej ocenę punktów gastronomicznych. Realizacja tego celu oparta została na trzech krokach:

- Początkowo został opracowany schemat tego jak powinien działać system aplikacji wspomagającej ocenę punktów gastronomicznych.
- Następnie, szczegółowo zostały przeanalizowane technologie do tworzenia aplikacji webowych. Zostało to osiągnięte za pomocą wiedzy zaczerpniętej z wielu źródeł, m.in. z literatury przedmiotu.
- W oparciu o wyniki analiz zawartych w rozdziałach pierwszym oraz drugim wskazany został sposób wdrożenia aplikacji wspomagającej ocenę punktów gastronomicznych.

Skupiając się na tej implementacji, widoczne kierunki rozwoju zależą od popularności Aplikacji internetowej wspomagającej ocenę usług gastronomicznych. Niektóre funkcjonalności mogą być dodawane wraz z pojawiением się na nie zapotrzebowania. Przykładem tego jest wprowadzenie możliwości modyfikacji danych. W chwili obecnej udostępnione są opcje oceny, przeglądania dla użytkownika niezalogowanego, przeglądania dla użytkownika zalogowanego, dodawania, usuwania, zgłoszenia uwagi, rejestracji. Jednak przy wzroście danych wygodniejsze może to się okazać wprowadzanie nowych informacji poprzez modyfikację informacji już wprowadzonych.

Spis rysunków

1	Etapy nawiązania połączenia http.	8
2	Składniki URL.	9
3	Krzywa uczenia.	11
4	Domain-Driven Design.	14
5	Porównanie wyników serializacji.	19
6	Wybór bazy danych.	20
7	Porównanie prędkości dwóch ostatnich wersji PostgreSQL.	21
8	Porównanie dwóch wersji mySQL.	22
9	Porównanie wydajność względem czasu na różnych platformach.	23
10	Przypadki użycia aplikacji.	26
11	Dodawanie opinii o punkcie gastronomicznym.	28
12	Wyniki eksperymentu.	30
13	Model bazy danych.	33
14	Strona widoczna dla niezalogowanego użytkownika.	36
15	Zmiany na stronie po zalogowaniu się użytkownika USER.	36
16	Zmiany na stronie po zalogowaniu się użytkownika ADMIN.	37
17	Okno programu JMeter z podanymi danymi.	39
18	Ustawianie serwera proxy, zależne od systemu i przeglądarki.	39
19	Wyniki zwrócone przez programu JMeter.	40

Spis tabel

1	Pierwsza część tabeli wspomagającej wybór.	31
2	Druga część tabeli wspomagającej wybór.	31
3	Pierwsza część procesu przydzielenia oceny.	35
4	Druga część procesu oceny.	35

Literatura

- [1] Barie Sosinsky, *Sieci komputerowe - Biblia*, HELION, Gliwice, 2011
- [2] Michał Bentkowski „*Programista*”, nr 3(35) 20155
- [3] Eric Evans, *Domain-Driven Design. Zapanuj nad złożonym systemem informatycznym* HELION, 15.04.2004
- [4] Jon Duckett, *HTML i CSS - Zaprojektuj i zbuduj własną witryne WWW*, HELION, Gliwice, 2014
- [5] Larry Ullman, *MySQL*, HELION, Gliwice, 2006
- [6] Raphael Saunier, *Laravel 4 - Podstawy tworzenia aplikacji w PHP*, HELION, Gliwice, 2014
- [7] Simone Busoli , *What is Inversion of Control?*, <http://stackoverflow.com/questions/3058/what-is-inversion-of-control> dostęp: 31.07.2015
- [8] Rzeczpospolita/PAP, „2015 rok przyniesie wzrost rynku gastronomicznego”, <http://www.portalspozywczy.pl/horeca/wiadomosci/2015-rok-przyniesie-wzrostu-rynu-gastronomicznego,109365.html>, dostęp: 31.07.2015
- [9] Forbes.pl, „Rynek gastronomiczny w Polsce – od czego zależy sukces restauracji”, <http://www.forbes.pl/rynek-gastronomiczny-w-polsce-od-czego-zalezy-sukces-restauracji,artykuly,184437,1,2.html>, dostęp: 31.07.2015
- [10] dev.mysql.com, „Why Move to MySQL from Microsoft SQL Server?”, dostęp: 9.11.2015
- [11] phpro.org/articles/Companies-Using-PHP.html, „Companies using php”, dostęp: 27.01.2016