POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA (INF)

SPECJALNOŚĆ: INŻYNIERIA SYSTEMÓW INFORMATYCZNYCH (INS)

PRACA DYPLOMOWA

INŻYNIERSKA

System zarządzania restauracją

Restaurant managment system

AUTOR:

Olaf Krawczyk

PROWADZĄCY PRACĘ:

dr inż. Roman Ptak, W4/K-9

OCENA PRACY:

WROCŁAW, 2017

**Spis treści**

[Spis rysunków 4](#_Toc499489347)

[Spis tabel 5](#_Toc499489348)

[Spis listingów 6](#_Toc499489349)

[Skróty 7](#_Toc499489350)

[1. Wstęp 8](#_Toc499489351)

[1.1. Wprowadzenie 8](#_Toc499489352)

[1.2. Cel i zakres pracy 8](#_Toc499489353)

[1.3. Aplikacje typu SPA 9](#_Toc499489354)

[1.4. Aplikacja mobilna 9](#_Toc499489355)

[2. Istniejące rozwiązania 11](#_Toc499489356)

[2.1. OpenTable 11](#_Toc499489357)

[2.2. Zomato 12](#_Toc499489358)

[3. Projekt systemu 14](#_Toc499489359)

[3.1. Użytkownicy systemu 14](#_Toc499489360)

[3.1.1. Klient 14](#_Toc499489361)

[3.1.2. Właściciel 14](#_Toc499489362)

[3.2. Wymagania funkcjonalne 14](#_Toc499489363)

[3.2.1. Identyfikacja użytkowników 14](#_Toc499489364)

[3.2.2. Wyszukiwanie restauracji 15](#_Toc499489365)

[3.2.3. Dokonanie rezerwacji 15](#_Toc499489366)

[3.2.4. Statusy rezerwacji 15](#_Toc499489367)

[3.2.5. Dodawanie restauracji 16](#_Toc499489368)

[3.2.6. Zarządzanie restauracjami 16](#_Toc499489369)

[3.2.7. Obsługa rezerwacji 16](#_Toc499489370)

[3.3. Wymagania niefunkcjonalne 16](#_Toc499489371)

[3.3.1. Dostęp do serwisu 16](#_Toc499489372)

[3.3.2. Komunikacja pomiędzy składowymi systemu 16](#_Toc499489373)

[3.3.3. Zabezpieczenia 17](#_Toc499489374)

[3.3.4. Wykorzystane technologie 17](#_Toc499489375)

[3.4. Przypadki użycia 18](#_Toc499489376)

[3.4.1. Przypadki użycia dla użytkownika typu Klient 18](#_Toc499489377)

[3.4.2. Przypadki użycia dla użytkownika typu Właściciel 22](#_Toc499489378)

[3.5. Architektura systemu 23](#_Toc499489379)

[3.6. Kolejna sekcja poziomu 1 24](#_Toc499489380)

[3.6.1. Kolejna sekcja poziomu 2 24](#_Toc499489381)

[4. Podsumowanie i wnioski 26](#_Toc499489382)

[Literatura 27](#_Toc499489383)

[Dodatek A 28](#_Toc499489384)

# Spis rysunków

[Rys. 1. Przykład podpisu rysunku 22](#_Toc499481703)

# Spis tabel

[Tab. 1. Przykład podpisu tabeli 21](#_Toc499481730)

# Spis listingów

[Listing. 1. Początkowe żadanie HTTP 3](#_Toc465685644)

# Skróty

**JSON** (ang. JavaScript Object Notation)

**XML** (ang. *eXtensible Markup Language*)

**SOAP** (ang. *Simple Object Access Protocol*)

**WSDL** (ang. *Web Services Description Language*)

**UDDI** (ang. *Universal Description Discovery and Integration*)

**GIS** (ang. *Geographical Information System*)

**SDI** (ang. *Spatial Data Infrastructure*)

**ISO** (ang. *International Standards Organization*)

**WMS** (ang. *Web Map Service*)

**WFS** (ang. *Web Feature Service*)

**WPS** (ang. *Web Processing Service*)

**GML** (ang. *Geography Markup Language*)

**SRG** (ang. *Seeded Region Growing*)

**SOA** (ang. *Service Oriented Architecture*)

**IT** (ang. *Information Technology*)

1. Wstęp

Niniejsza praca dyplomowa jest opisem realizacji systemu aplikacji, którego celem jest dostarczenie rozwiązania ułatwiającego rezerwację stolików w restauracjach. System ten składa się z trzech aplikacji – mobilnej, internetowej oraz aplikacji dostarczającej interfejs dostępu do bazy danych.

* 1. Wprowadzenie

Znaczna część serwisów internetowych oraz aplikacji mobilnych powstaje, aby ułatwić użytkownikom codzienne czynności. Dla przykładu można wymienić aplikacje takie jak Any.do pozwalającą na tworzenie przejrzystych list zadań. Istnieją również aplikacje pozwalające w kilku kliknięciach zamówić taksówkę. Powstają również serwisy pozwalające zamówić jedzenie online. Podczas realizacji pracy dyplomowej stworzę serwis internetowy ułatwiający rezerwację stolików klientom restauracji za pośrednictwem przeglądarki internetowej oraz aplikacji mobilnej. Serwis ten będzie również wspierał właścicieli restauracji pozwalając im na dodawanie ich restauracji oraz zarządzanie pojawiającymi się rezerwacjami.

* 1. Cel i zakres pracy

Celem tej pracy jest zaprojektowanie oraz implementacja systemu składającego się z bazy danych, aplikacji mobilnej, aplikacji internetowej typu Single Page Application oraz aplikacji dostarczającej interfejs do komunikacji z bazą danych. Aplikacja mobilna zostanie stworzona z wykorzystaniem narzędzia Ionic, które pozwala na tworzenie aplikacji działających na różnych platformach mobilnych. Mobilna części serwisu będzie dedykowana klientom restauracji, za jej pomocą będą mogli dokonać rezerwacji stolika. Aplikacja internetowa będzie dostarczała interfejs klienta oraz właściciela restauracji. Ta część systemu będzie udostępniała funkcjonalności aplikacji mobilnej oraz interfejs dla właściciela restauracji. Ponieważ wymienione wyżej części składowe systemu uruchamiane będą na wielu różnych urządzeniach istotny jest interfejs komunikacji pomiędzy nimi, a bazą danych. Aby go zrealizować stworzę aplikację dostarczającą interfejs programistyczny wykorzystujący protokół http oraz REST API. Aplikacja ta zostanie stworzona z wykorzystaniem frameworka Spring. Ponieważ w serwisie będą przechowywane dane osobowe użytkowników komunikacja w systemie musi zostać zabezpieczona w taki sposób, aby do.

* 1. Aplikacje typu SPA

Aplikacje Single Page Applicaion powstały, aby zmniejszyć objętość zapytań http wykonywanych przez przeglądarki internetowe. Aplikacja tego typu jest ładowana do pamięci przeglądarki przy odwiedzeniu strony. W praktyce odwiedzający pobiera tylko raz plik index.html, wszelkie zmiany treści oraz wyglądu strony są realizowane w wyniku działań użytkownika. Przy korzystaniu z serwisów stworzonych z wykorzystaniem technologii SPA użytkownik nie doświadcza przeładowania strony. Zmianom podlegają tylko wybrane elementy modelu DOM, np. określone znaczniki. Nowe dane pobierane są wyłącznie, gdy istnieje taka potrzeba, a ich rozmiar można maksymalnie ograniczyć przez przesyłanie tylko istotnych informacji, bez dodatkowych danych takich jak strony HTML. Wiąże się to z oszczędnościami transferu oraz pozytywnym wpływem na odbiór strony przez użytkownika, gdy ładowane są nowe elementy strony użytkownik cały czas może korzystać z tych które są już załadowane. Znanym serwisem opierającym się o tę technologię jest aplikacja internetowa GMail.com

Do tworzenia aplikacji SPA wykorzystywane są powszechnie znane technologie takie jak JavaScript (oraz jego odmiany), HTML oraz CSS. Ponadto wymagają one frameworków takich jak Angular czy ReactJS.

Rozwiązania SPA mają wiele zalet, jednak posiadają również wady. Z uwagi na fakt iż, strona jest ładowana w całości wyłącznie raz podczas tego połączenia przesyłana jest duża ilość informacji – skompilowany plik JavaScript z aplikacją, pomocnicze biblioteki oraz arkusze stylów CSS. Do wad można również zaliczyć stopień skomplikowania frameworków do tworzenia tego typu stron, gdyż aplikacje tworzone za ich pomocą muszą mieć ściśle określoną architekturę, która nie zawsze jest intuicyjna. Należy również zauważyć, że witryny SPA ze względu na swój dynamiczny charakter nie zawsze są poprawnie indeksowane przez crawlery wyszukiwarek internetowych, chociaż w ostatnim czasie uległo to znacznej poprawie.

* 1. Aplikacja mobilna

Aplikacje mobilne to aplikacje, które można uruchomić na szeroko rozumianych urządzeniach mobilnych takich jak smartfony, tablety oraz smartwatche. Aplikacje te instalowane są bezpośrednio na urządzeniu oraz mają możliwość korzystania z jego zasobów np. kamery, głośników czy mikrofonu. Obecnie istnieje wiele sposobów tworzenia aplikacji mobilnych. Aplikacje te mogą być tworzone z wykorzystaniem technologii dostarczanych przez twórców systemu operacyjnego, jednak korzystanie z nich będzie możliwe jedynie na urządzeniu z dedykowanym systemem. Takie podejście wymaga napisania osobnych aplikacji na urządzenia z systemem Android oraz osobnej na urządzenia z systemem iOS. Istnieją również narzędzia pozwalające na tworzenie aplikacji z wykorzystaniem technologii znanych z rozwiązań internetowych (JavaScript, HTML, CSS), która następnie może zostać skompilowana do aplikacji dedykowanej na konkretny system. W praktyce programista pisze kod aplikacji raz, a następnie za pomocą odpowiednich komend framework buduje aplikacje na wybrane platformy. Jednym z takich rozwiązań jest Ionic, działający w oparciu o Angular oraz Apatche Cordova. Framework ten wykorzystałem do stworzenia aplikacji mobilnej na potrzeby niniejszej pracy. Główną zaletą pakietu Ionic jest ponowne wykorzystanie kodu. Posiadając działającą aplikację Ionic możemy za pomocą interfejsu dostarczonego wraz z pakietem zbudować aplikacje na urządzenia iOS, Android oraz Windows. Oczywiście rozwiązanie to nie jest idealne, aplikacje napisane w ten sposób mają ograniczony dostęp do natywnych funkcji systemu oraz działają zauważalnie wolniej od aplikacji dedykowanych na poszczególne platformy. Pomimo wad rozwiązanie to bardzo dobrze sprawdza się w przypadku w aplikacjach realizujących bardzo proste zadania takie jak dostęp do danych z zewnętrznej bazy danych.

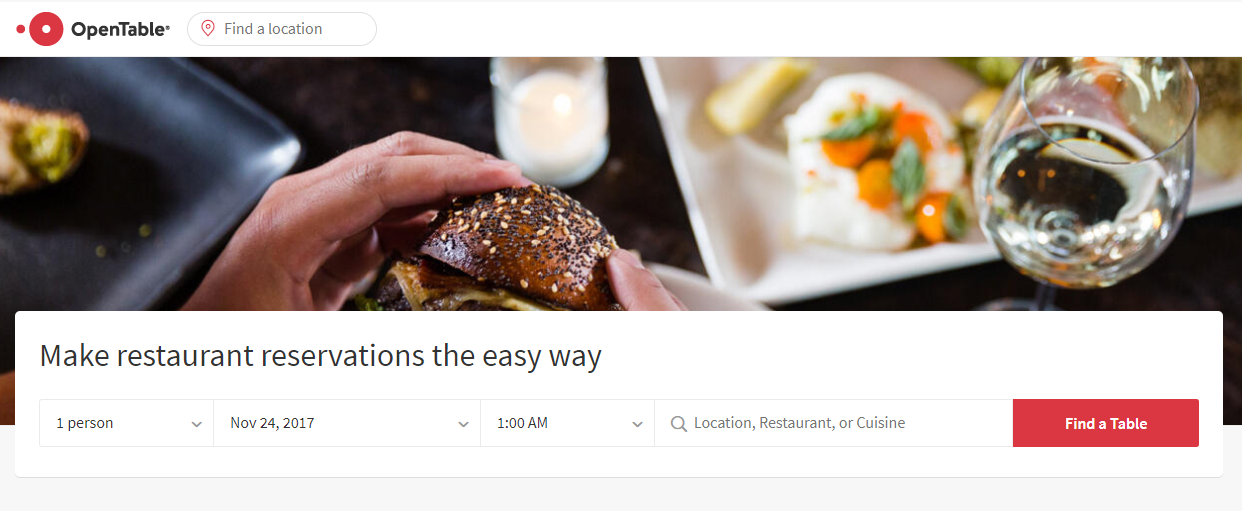
1. Istniejące rozwiązania

Na światowym rynku istnieje wiele rozwiązań pozwalających na rezerwację stolików online. Na polskim rynku jednak nie są one jeszcze powszechne.

* 1. OpenTable

Amerykański serwis OpenTable założony w 1998 roku daje możliwość rezerwacji miejsc w restauracjach na całym świecie. Według danych dostępnych w Wikipedii na dzień pisana tej pracy w serwisie znajduje się 40 tysięcy restauracji w wielu krajach świata. Niestety serwis nie działa w Polsce.

Użytkownicy mogą dokonać rezerwacji za pośrednictwem strony internetowej <http://opentable.com> lub aplikacji mobilnych. Aby zarezerwować stolik należy wypełnić formularz znajdujący się na stronie głównej serwisu.



Rysunek 1 Widok części strony głównej OpenTable

Jak widać na powyższym rysunku dane jakie należy podać to ilość osób dla jakiej ma zostać zarezerwowany stolik, datę rezerwacji, godzinę oraz lokalizację. Po wysłaniu formularza zostaje wyświetlona lista restauracji spełniających podane kryteria oraz możliwe godziny rezerwacji. Po wybraniu restauracji oraz godziny użytkownik zostaje przekierowany na stronę finalizującą proces w na której należy podać dane osobowe oraz numer telefonu kontaktowego. Rezerwacji stolika można dokonać również poprzez aplikację mobilną.

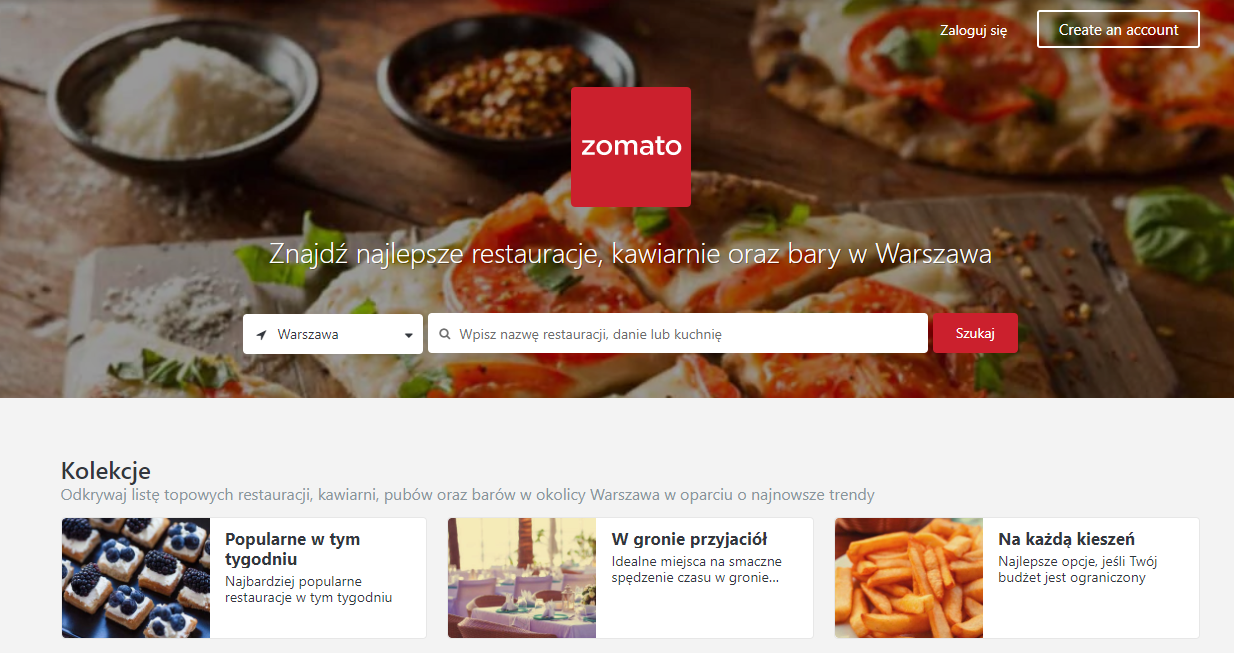
Serwis poza podstawowymi informacjami takimi jak nazwa restauracji oraz adres udostępnia użytkownikom inne dane pozwalające pomagające w wyborze restauracji. Są to na przykład aktualnie serwowane menu, galeria zdjęć prezentująca wystrój restauracji oraz przedział cenowy w jakim znajduje się restauracja.

OpenTable posiada również część społecznościową. Użytkownicy serwisu mogą umieszczać w nim opinie o odwiedzonych restauracjach oraz oceniać je przyznając punkty w postaci gwiazdek.

Restauratorzy mają możliwość zarządzania restauracją za pomocą aplikacji na urządzenia z systemem iOS dostarczanej przez OpenTable. Aplikacja ta pozwala na wprowadzenie stolików znajdujących się w restauracji oraz zarządzania nimi z jej poziomu. Możliwe jest również sprawdzanie statystyk restauracji oraz bieżących rezerwacji.

* 1. Zomato

Serwis Zomato (zomato.com) podobnie jak OpenTable jest bazą restauracji. Działa on w ponad 150 miastach na świecie. Pozwala na wyświetlenie restauracji znajdujących się w wybranej okolicy.



Rysunek 2 Widok strony głównej serwisu Zomato

Użytkownicy aplikacji mają dostęp do galerii, menu restauracji, wskazówek dojazdu oraz danych kontaktowych. Klienci mogą oceniać restauracje na podstawie skali gwiazdek oraz dodawać recenzje. Serwis ten jest bardzo podobny do OpenTable, różni się on jednak podejściem do rezerwacji online. Użytkownicy korzystający z Zomato mogą zarezerwować za jego pośrednictwem stolik tylko w wybranych restauracjach.

Podobnie jak OpenTable Zomato udostępnia swoim użytkownikom aplikacje mobilne na urządzenia działające pod kontrolą popularnych systemów operacyjnych. Aplikacja ta dostarcza takich samych funkcjonalności jak serwis internetowy.

Patrząc na ofertę serwisów OpenTable oraz Zomato możemy dostrzec wspólne cechy tych rozwiązań. Oba serwisy udostępniają użytkownikom możliwość znajdowania restauracji w wybranym mieście, dostarczają dane kontaktowe oraz informacje o serwowanym menu. Jak wiele obecnych na rynku usług udostępniają dostarczają aplikacje mobilne pozwalające na korzystanie z serwisu za pomocą urządzeń przenośnych. Tworząc system w ramach niniejszej pracy chciałbym odwzorować wymienione powyżej cechy oraz architekturę systemu.

1. Projekt systemu
   1. Użytkownicy systemu

System opisany w tej pracy będzie dostarczał funkcjonalności dwóm grupom użytkowników. Będą to osoby indywidualne nazywane, klienci restauracji nazywane dalej klientami oraz właściciele restauracji lub osoby przez nie upoważnione nazywane dalej właścicielami. Zarówno właściciel jak i klient będą identyfikowani za pomocą danych osobowych oraz adresu email.

* + 1. Klient

Klient to osoba wykorzystująca serwis do znalezienia restauracji znajdującej się w okolicy oraz zarezerwowania stolika. Każdy klient posiadał będzie indywidualne konto, za pośrednictwem którego będzie dokonywał nowych rezerwacji w restauracjach oraz będzie mógł przejrzeć ich historię.

* + 1. Właściciel

Jako właściciela w serwisie rozumie się osobę, która jest przedstawicielem restauracji. Konto właściciela pozwala na dodawanie nowych restauracji do serwisu oraz zarządzanie rezerwacjami.

* 1. Wymagania funkcjonalne
     1. Identyfikacja użytkowników

Każdy z użytkowników niezależnie od roli posiadać będzie indywidualne konto. Konto zostaje utworzone przez użytkownika przez podanie danych – imię, nazwisko, numer telefonu oraz hasło. Aby zalogować się do serwisu użytkownik musi podać adres email oraz hasło.

* + 1. Wyszukiwanie restauracji

Każdy klient będzie miał możliwość wyszukania restauracji spełniających sprecyzowane przez niego kryteria.

* + 1. Dokonanie rezerwacji

Klient będzie miał możliwość wysłania prośby o rezerwację stolika w wybranej restauracji. Rezerwacja może zostać anulowana przez klienta. Użytkownik będzie mógł wyświetlić wszystkie swoje rezerwacje. Dla uproszczenia przyjmujemy, że godzina na którą zostaje dokonana rezerwacja może być wyłącznie pełna, oraz czas trwania rezerwacji jest stały i wynosi 2 godziny.

* + 1. Statusy rezerwacji

Każda rezerwacja w serwisie w zależności od wykonanych na niej operacji powinna posiadać swój status odzwierciedlający jej stan. Status może przyjmować jedną z poniższych wartości:

Tabela 1 Statusy rezerwacji

|  |  |  |
| --- | --- | --- |
| Lp. | Status | Opis |
| 1 | PENDING | Rezerwacja została wysłana przez użytkownika,  oczekuje na akcję właściciela restauracji |
| 2 | ACCEPTED | Rezerwacja została zaakceptowana przez  właściciela restauracji |
| 3 | REJECTED | Rezerwacja została odrzucona przez właściciela  restauracji |
| 4 | CANCELED | Rezerwacja została anulowana przez klienta |
| 5 | CLOSED | Zaakceptowana rezerwacja została automatycznie  zamknięta po upłynięciu terminu rezerwacji |

* + 1. Dodawanie restauracji

Użytkownik zalogowany jako właściciel może dodawać nowe restauracje do serwisu. Dodając restauracje użytkownik podaje dane takie jak nazwa, adres, opis, definiuje menu restauracji oraz stoliki w niej dostępne. Dodana restauracja przypisana jest do konta właściciela. Właściciel może mieć wiele restauracji.

* + 1. Zarządzanie restauracjami

Właściciel powinien mieć możliwość zmiany danych restauracji, jej menu oraz stolików za pośrednictwem przeglądarki internetowej.

* + 1. Obsługa rezerwacji

Właściciel restauracji powinien mieć możliwość akceptacji bądź odrzucenia rezerwacji wysłanych przez klientów. Właściciel powinien mieć możliwość wyświetlenia wszystkich rezerwacji dotyczących jego restauracji.

* 1. Wymagania niefunkcjonalne
     1. Dostęp do serwisu

Dostęp do serwisu dla klienta oraz właściciela powinien być możliwy za pomocą przeglądarki internetowej. Aplikacja internetowa powinna mieć intuicyjny oraz przyjazny użytkowników interfejs. Funkcjonalności dedykowane klientom powinny być również dostępne za pomocą aplikacji mobilnej działającej na urządzeniach z systemem Android.

* + 1. Komunikacja pomiędzy składowymi systemu

Ponieważ opisywany system będzie składał się z trzech aplikacji oraz bazy danych komunikacja pomiędzy poszczególnymi aplikacjami powinna być realizowana za pomocą protokołu HTTP z wykorzystaniem architektury REST. Implementacja takiego sposobu komunikacji pozwoli również na rozszerzanie systemu o kolejne aplikacje np. aplikację działającą na komputerach osobistych.

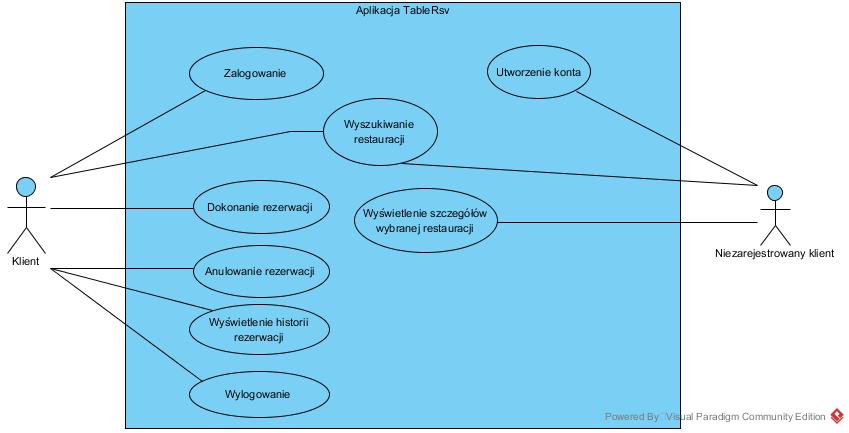
* + 1. Zabezpieczenia

Aby uzyskać dostęp do konta użytkownik musi zalogować się za pomocą adresu email oraz hasła podanego przy rejestracji, niezależnie od typu konta. Wszystkie zapytania do serwera udostępniającego REST API powinny być zabezpieczone tokenem potwierdzającym, że użytkownik ma dostęp do danych. Aby ułatwić użytkownikom korzystanie z serwisu token powinien być zapamiętany w pamięci przeglądarki.

* + 1. Wykorzystane technologie

Serwis powinien zostać stworzony z wykorzystaniem następujących technologii:

* Spring Boot
* Spring Security
* JSON Web Token
* Angular
* HTML5, CSS3
* Ionic
* PostgreSQL
* Bootstrap
  1. Przypadki użycia
     1. Przypadki użycia dla użytkownika typu Klient



Rysunek 3 Diagram przypadków użycia użytkownika typu Klient

Opis wymienionych przypadków użycia wymienionych na powyższym diagramie znajduje się w dalszej części podrozdziału. Powyższy diagram dotyczy klienta korzystającego z serwisu za pomocą przeglądarki internetowej. Klient korzystający z aplikacji mobilnej ma dostęp do tych samych funkcjonalności, z tą różnicą, że nie ma możliwości utworzenia nowego konta przez aplikację mobilną. Aplikacja mobilna daje dostęp do wymienionych funkcjonalności po uprzednim zalogowaniu się.

* + - 1. PU Utworzenie konta

Przypadek ten opisuje utworzenie nowego konta użytkownika. Funkcjonalność ta jest dostępna jedynie za pośrednictwem aplikacji WWW.

Tabela 2 PU Utworzenie konta klienta

|  |  |
| --- | --- |
| WS | Otwarcie strony WWW serwisu |
| WK | Utworzenie nowego konta użytkownika lub komunikat o błędzie |
| Cel | Rejestracja nowego użytkownika typu Klient |
| Przebieg | 1. Przejście do formularza rejestracji klienta 2. Podanie danych osobowych 3. Podanie adresu email oraz hasła 4. Weryfikacja czy wszystkie pola zostały wypełnione 5. Weryfikacja czy w serwisie nie istnieje już klient   o tym samym adresie email   1. Utworzenie nowego konta klienta lub komunikat   o błędzie |

* + - 1. PU Zalogowanie

Przypadek użycia zalogowanie opisuje zachowanie aplikacji podczas logowania się klienta do serwisu zarówno za pomocą przeglądarki internetowej, jak i aplikacji mobilnej. Mechanizm autoryzacji tokenem wygenerowanym poniżej zostanie opisany w części omawiającej implementację systemu.

Tabela 3 PU Zalogowanie klienta

|  |  |
| --- | --- |
| WS | Otwarcie strony WWW serwisu lub aplikacji mobilnej |
| WK | Zalogowanie klienta lub komunikat o błędzie |
| Cel | Zalogowanie użytkownika oraz utworzenie tokena sesji  autoryzacji |
| Przebieg | 1. Przejście do formularza logowania klienta 2. Podanie adresu email oraz hasła 3. Weryfikacja czy klient o podanym adresie   email istnieje w serwisie oraz czy skrót hasła  zapisanego w bazie zgadza się ze skrótem hasła  wprowadzonego podczas logowania   1. Jeżeli podane dane nie są poprawne wyświetlenie   komunikatu o błędzie, w przeciwnym przypadku  wygenerowanie tokena.   1. Wysłanie odpowiedzi z tokenem |

* + - 1. PU Wylogowanie

Wylogowanie jest możliwe wyłącznie w podczas korzystania z internetowej części serwisu, ponieważ tylko w tej wersji token użytkownika zostaje zapisany w pamięci urządzenia.

Tabela 4 PU Wylogowanie klienta

|  |  |
| --- | --- |
| WS | PU Zalogowanie |
| WK | Usunięcie tokena z pamięci przeglądarki |
| Cel | Wylogowanie klienta poprzez usunięcie tokena sesji |
| Przebieg | 1. Naciśnięcie przycisku „Logout” z menu 2. Usunięcie tokena z pamięci przeglądarki 3. Przekierowanie użytkownika na stronę główną |

* + - 1. PU Wyszukiwanie restauracji

Przebieg przypadku użycia „Wyszukiwanie restauracji” jest identyczny dla serwisu internetowego oraz aplikacji mobilnej. Dla aplikacji mobilnej warunkiem startowym jest „PU Zalogowanie”

Tabela 5 PU Wyszukiwanie restauracji

|  |  |
| --- | --- |
| WS | Otwarcie strony WWW serwisu lub aplikacji mobilnej |
| WK | Zwrócenie listy restauracji spełniających podane kryteria  lub wyświetlenie komunikatu o błędzie |
| Cel | Wyświetlenie listy restauracji w których klient może  zarezerwować stolik na wybraną godzinę |
| Przebieg | 1. Przejście do głównej strony 2. Wybór miasta 3. Wybór liczby gości 4. Wybór daty oraz godziny 5. Wybór rodzaju kuchni 6. Wyświetlenie restauracji spełniających kryteria   lub komunikatu z informacją, że żadna z restauracji  nie spełnia podanych kryteriów. |

* + - 1. PU Wyświetlenie szczegółów restauracji

Ten przypadek użycia jest identyczny dla aplikacji mobilnej oraz serwisu internetowego.

Tabela 6 PU Wyświetlenie szczegółów restauracji

|  |  |
| --- | --- |
| WS | PU Wyszukiwanie restauracji |
| WK | Przeniesienie użytkownika do widoku szczegółów restauracji |
| Cel | Umożliwienie klientowi zapoznanie się z ofertą restauracji |
| Przebieg | 1. Klient wybiera restaurację z listy otrzymanej podczas   wyszukiwania   1. Przeniesienie użytkownika do widoku szczegółów   restauracji |

* + - 1. PU Dokonanie rezerwacji

Tabela 7 PU Dokonanie rezerwacji

|  |  |
| --- | --- |
| WS | PU Wyświetlenie szczegółów restauracji, PU Zalogowanie |
| WK | Utworzenie nowej rezerwacji dla wybranej restauracji |
| Cel | Rejestracja nowego użytkownika typu Klient |
| Przebieg | 1. Należy kliknąć przycisk „View tables” 2. Wybór jednego ze stolików z listy dostępnych   poprzez naciśnięcie przycisku „Book table”   1. Utworzenie nowej rezerwacji o statusie PENDING 2. Przekierowanie klienta do widoku historii jego   rezerwacji |

* + - 1. PU Wyświetlenie historii rezerwacji

Tabela 8 Wyświetlenie historii rezerwacji

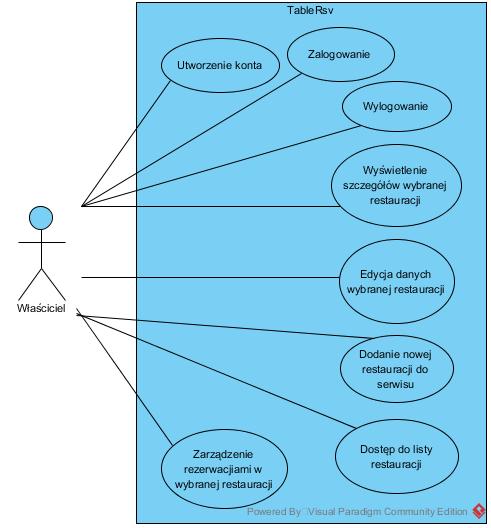
|  |  |
| --- | --- |
| WS | PU Zalogowanie |
| WK | Wyświetlenie wszystkich rezerwacji klienta |
| Cel | Wyświetlenie wszystkich rezerwacji użytkownika |
| Przebieg | 1. Wybór widoku „Reservations” 2. Pobranie rezerwacji z bazy danych 3. Przekierowanie użytkownika do widoku   pokazującego dokonane przez niego rezerwacje |

* + - 1. PU Anulowanie rezerwacji

Tabela 9 Wyświetlenie historii rezerwacji

|  |  |
| --- | --- |
| WS | PU Wyświetlenie historii rezerwacji |
| WK | Zmiana statusu wybranej rezerwacji na CANCELED |
| Cel | Anulowanie wybranej przez klienta rezerwacji |
| Przebieg | 1. Należy kliknąć przycisk „Cancel”   znajdujący się przy wybranej rezerwacji   1. Zmiana statusu rezerwacji w bazie danych 2. Odświeżenie widoku rezerwacji |

* + 1. Przypadki użycia dla użytkownika typu Właściciel



Rysunek 4 Diagram przypadków użycia – właściciel

Funkcjonalności serwisu dedykowane właścicielom restauracji są dostępne jedynie z aplikacji internetowej. Wszystkie opisane poniżej przypadki użycia zakładają, że użytkownik korzysta z aplikacji internetowej.

* + - 1. PU Utworzenie konta

Tabela 10 PU Utworzenie konta właściela

|  |  |
| --- | --- |
| WS | Otwarcie strony WWW serwisu |
| WK | Utworzenie nowego konta właściciela lub komunikat o błędzie |
| Cel | Rejestracja nowego użytkownika typu Właściciel |
| Przebieg | 1. Przejście do formularza rejestracji właściciela 2. Podanie danych osobowych 3. Podanie adresu email oraz hasła 4. Weryfikacja czy wszystkie pola zostały wypełnione 5. Weryfikacja czy w serwisie nie istnieje już właściciel   o tym samym adresie email   1. Utworzenie nowego konta właściciela lub komunikat   o błędzie |

* + - 1. PU Zalogowanie

Tabela 11 PU Logowanie właściciela

|  |  |
| --- | --- |
| WS | Otwarcie strony WWW serwisu |
| WK | Zalogowanie właściciela lub komunikat o błędzie |
| Cel | Zalogowanie właściela oraz utworzenie tokena sesji do  autoryzacji |
| Przebieg | 1. Przejście do formularza logowania właściciela 2. Podanie adresu email oraz hasła 3. Weryfikacja czy właściciel o podanym adresie   email istnieje w serwisie oraz czy skrót hasła  zapisanego w bazie zgadza się ze skrótem hasła  wprowadzonego podczas logowania   1. Jeżeli podane dane nie są poprawne wyświetlenie   komunikatu o błędzie, w przeciwnym przypadku  wygenerowanie tokena.   1. Zapisanie tokena w pamięci przeglądarki 2. Ustawienie praw właściciela dla zalogowanego   użytkownika   1. Wyświetlenie widoku panelu właściciela |

* + - 1. PU Wylogowanie właściciela

Tabela 12 PU Wylogowanie właściciela

|  |  |
| --- | --- |
| WS | PU Zalogowanie właściciela |
| WK | Usunięcie tokena z pamięci przeglądarki, usunięcie  praw właściciela dla aktywnej sesji |
| Cel | Wylogowanie właściciela |
| Przebieg | 1. Naciśnięcie przycisku „Logout” z menu 2. Usunięcie tokena z pamięci przeglądarki 3. Usunięcie uprawnień właściciela 4. Przekierowanie użytkownika na stronę główną |

* + - 1. PU Dostęp do listy restauracji

Tabela 13 Dostęp do listy restauracji

|  |  |
| --- | --- |
| WS | PU Zalogowanie |
| WK | Wyświetlenie widoku panelu właściciela |
| Cel | Wyświetlenie listy restauracji danego właściciela |
| Przebieg | 1. Naciśnięcie przycisku „Dashboard” w menu 2. Wyświetlenie widoku panelu właściciela |

* + - 1. PU Dodanie nowej restauracji do serwisu

Tabela 14 PU Dodanie nowej restauracji do serwisu

|  |  |
| --- | --- |
| WS | PU Dostęp do listy restauracji |
| WK | Utworzenie nowej restauracji w serwisie |
| Cel | Dodanie nowej restauracji do serwisu |
| Przebieg | 1. Wyświetlenie formularza dodawania   nowej restauracji   1. Podanie podstawowych informacji o restauracji:    * nazwa restauracji    * adres    * telefon, adres email    * rodzaj serwowanej kuchni    * godziny otwarcia 2. Podanie opisu restauracji 3. Podanie informacji na temat karty dań:    * nazwa dania    * opis    * cena 4. Weryfikacja czy wszystkie pola zostały wypełnione 5. Zapisanie restauracji w bazie danych 6. Wyświetlenie widoku restauracji |

* + - 1. PU Wyświetlenie szczegółów wybranej restauracji

Tabela 4 PU Wylogowanie klienta

|  |  |
| --- | --- |
| WS | PU Dostęp do listy restauracji |
| WK | Wyświetlenie szczegółowego widoku wybranej restauracji |
| Cel | Dostęp do szczegółów restauracji |
| Przebieg | 1. Wybranie restauracji z listy restauracji właściciela 2. Pobranie informacji o restauracji z bazy danych 3. Wyświetlenie widoku szczegółów restauracji |

* + - 1. PU Zarządzanie rezerwacjami restauracji

Tabela 15 PU Wylogowanie klienta

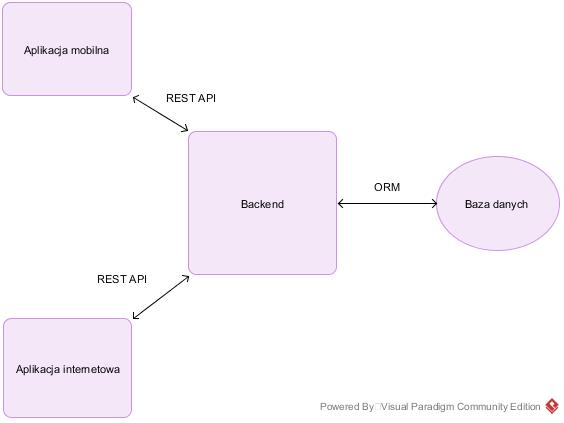
|  |  |
| --- | --- |
| WS | PU Wyświetlenie szczegółów wybranej restauracji |
| WK | Wyświetlenie listy wszystkich rezerwacji w systemie powiązanych  z wybraną restauracją |
| Cel | Zarządzanie rezerwacjami |
| Przebieg | 1. Wciśnięcie przycisku „All reservations” w widoku   szczegółów restauracji   1. Wyświetlenie widoku listy rezerwacji powiązanych z daną   restauracją posortowanych według daty utworzenia oraz  przyciskami umożliwiającymi zmianę statusu rezerwacji |

* + - 1. PU Edycja danych wybranej restauracji

Tabela 4 PU Wylogowanie klienta

|  |  |
| --- | --- |
| WS | PU Wyświetlenie szczegółów wybranej restauracji |
| WK | Aktualizacja danych restauracji |
| Cel | Zarządzanie danymi restauracji |
| Przebieg | 1. Naciśnięcie przycisku „Edit” w widoku szczegółów   restauracji   1. Wyświetlenie formularza edycji restauracji 2. Wprowadzenie zmian w danych restauracji 3. Weryfikacja czy wszystkie pola zostały poprawnie   wypełnione   1. Naciśnięcie przycisku „Save” 2. Aktualizacja danych restauracji 3. Przekierowanie użytkownika do widoku   szczegółów restauarcji |

* 1. Architektura systemu



Rysunek 5 Schemat logiczny systemu

Powyższy rysunek przedstawia schemat logiczny systemu. W jego centrum jest aplikacja nazwana „Backend” jest ona warstwą dostępu do bazy danych dla innych aplikacji. Backend realizuje funkcje grubego klienta, do zadań tej aplikacji należy walidacja danych otrzymanych z innych części serwisu. Obsuga logowania użytkowników oraz przyznawania im dostępu do poszczególnych funkcjonalności. Komunikacja z bazą danych realizowana jest za pośrednictwem narzędzi ORM. Pozwala to na efektywne przesyłanie obiektów pomiędzy aplikacjami w systemie oraz dodatkowo zabezpiecza aplikację przed atakami typu SQL Injection.

Aplikacjami klienckimi w tym systemie jest aplikacja internetowa napisana z wykorzystaniem technologii Angular oraz aplikacja mobilna stworzona w technologii Ionic. Obie aplikacje wykorzystują do komunikacji z aplikacją Backend protokół HTTP oraz REST API. Komunikacja poprzez REST opiera się o wysyłanie zapytań na odpowiednie adresy internetowe nazywane czasem „endpoints”. Dane przesyłane pomiędzy za pomocą REST API przesyłane są w formacie JSON.

* + - 1. Sekcja poziomu 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Tab. 1. Przykład podpisu tabeli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. Kolejna sekcja poziomu 1
     1. Kolejna sekcja poziomu 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Rys. 1. Przykład podpisu rysunku

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Listing. 1. Początkowe żądanie HTTP

GET /script/Articles/Latest.aspx HTTP/1.1  
Host: www.codeproject.com  
Connection: keep –alive  
Cache -Control: max-age=0  
Accept: text/html ,application/xhtml+xml,application/xml|  
User -Agent: Mozilla/5.0 ...  
Accept -Encoding: gzip ,deflate ,sdch  
Accept -Language: en-US...  
Accept -Charset: windows -1251,utf -8...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Przykład listy numerowanej:

1. Podsumowanie i wnioski

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus. Aliquam erat volutpat. Integer maximus est turpis, ut bibendum ligula accumsan et. Ut eget vestibulum libero. Aliquam erat volutpat. Nullam placerat mauris a lectus tincidunt, et aliquet turpis aliquam. Etiam in malesuada lacus. Proin dignissim augue sit amet auctor elementum. Suspendisse potenti. Vivamus suscipit vulputate massa ac molestie. Suspendisse a justo porttitor, commodo mi at, placerat risus. Integer lobortis augue ac neque suscipit, vel sodales lacus fringilla.

# Literatura

[1] M. Bickley, C. Slominski. A MySQL-based data archiver: preliminary results. Proceedings of ICALEPCS07, Paz. 2007. http://www.osti.gov/scitech/servlets/purl/922267 [dostęp dnia 20 czerwca 2015].

[2] J. Jędrzejczyk, B. Sródka. Segmentacja obrazów metodą drzew decyzyjnych. Raport instytutowy, Politechnika Wrocławska, Wydział Elektroniki, 2007.

# Dodatek A