1. Wymagania biznesowe

ServiceCMS to aplikacja internetowa, dedykowana dla wszelakich firm usługowych. Aplikacja ma za zadanie wspomaganie w tworzeniu strony internetowej firmy. Najważniejszą funkcjonalnością aplikacji jest zautomatyzowanie rejestracji klientów na zdefiniowane wcześniej usługi. Ponadto aplikacja umożliwia zarządzanie treściami, które są reprezentowane przez strony i aktualności. Umożliwia również zarządzanie plikami i udostępnia możliwość przeglądania statystyk odwiedzin panelu klienckiego. Oprogramowanie może obsługiwać osoba nietechniczna z podstawową znajomością obsługi komputera.

1. Słownik pojęć

**Administrator** – osoba wyznaczona do zarządzania treścią, która jest wyświetlana w panelu klienckim.

**Klient** – osoba odwiedzająca panel kliencki

**Panel administracyjny** – strona internetowa umożliwiająca administratorowi zarządzanie treścią

**Panel kliencki** – strona internetowa która jest wynikiem ustawień ustalonych w panelu administracyjnym

**Aktualność** – prosty typ treści, wyświetlany na głównej stronie panelu klienckiego

**Strona** – typ treści który może zawierać wiele elementów takich jak ilustracje i załączniki w postaci plików. Do strony może prowadzić link w postaci przycisku z menu głównego.

**Dostawca usługi** – byt związany z typem usługi. Może to być osoba, stanowisko czy narzędzie wykonujące usługę.

**Typ usługi** – opis usługi wykonywanej w firmie. Każdy typ może posiadać wiele faz usługi.

**Popup** – wiadomość wyświetlana klientowi przy pierwszej wizycie w panelu klienckim.

**Odbiorca newslettera** – adres email na który może być wysłany newsletter

1. Wymagania funkcjonalne

Specyfikacja wymagań dla panelu administratora

* Możliwość dodawania, usuwania i edytowania aktualności
* Możliwość dodawania, usuwania i edytowania kategorii aktualności
* Możliwość dodawania, usuwania i edytowania artykułów/stron
* Możliwość przeglądania i usuwania usług zarejestrowanych przez klienta
* Możliwość dodawania, usuwania i edytowania aktualności
* Możliwość dodawania, usuwania i edytowania dostawców usług
* Możliwość dodawania, usuwania i edytowania typów usług
* Możliwość zarządzania przyciskami menu
* Możliwość dodawania, usuwania i edytowania odbiorców newslettera
* Możliwość kompozycji i wysyłania newslettera
* Możliwość dodawania, usuwania plików
* Możliwość dodawania, usuwania i edytowania popupów
* Możliwość przeglądania statystyk odwiedzin dotyczących panelu klienckiego
* Możliwość edytowania ustawień
* Możliwość zmiany języka na polski lub angielski

Specyfikacja wymagań dla panelu klienckiego

* Możliwość wyświetlania aktualności
* Możliwość wyświetlania stron
* Możliwość nawigowania za pomocą menu
* Możliwość rejestracji na daną usługę
* Wyświetlanie popupu przy pierwszej wizycie
* Zbieranie informacji o odwiedzinach

1. Wymagania niefunkcjonalne

* Aplikacja powinna mieć spójny i przejrzysty interfejs użytkownika
* Panel administracyjny powinien być w języku polskim jak i angielskim
* Zarówno panel kliencki jak i panel administracyjny powinien być poprawnie wyświetlany na rozdzielczościach nie mniejszych niż 1600x900
* Aplikacja powinna działać na najnowszych wersjach przeglądarek Chrome, Firefox i Opera

1. Warstwy aplikacji

Aplikacja została podzielona na 4 warstwy : rdzenia, modułów, logiki biznesowej i prezentacji. Dzielenie aplikacji miało na celu uporządkowanie struktury aplikacji i zmniejszenie redundancji kodu. Poniżej przedstawiono każdą z warstw:

* 1. **Rdzeń** – warstwa odpowiadająca za połączenie i obsługę bazy danych, obsługę i logowanie wyjątków oraz zawierająca klasy wykorzystywane w całej aplikacji.
  2. **Logika biznesowa** – warstwa zawierająca implementacje algorytmów logiki biznesowej takiej jak zapisywanie, edytowanie i usuwanie modeli wykorzystywanych w aplikacji.
  3. **Moduły** – warstwa zawierająca wymienialne rozwiązania niezależne od reszty aplikacji np. obsługa zapisu i usuwania plików, łączenie z zewnętrznym serwisem lub obsługa kryptografii haseł.
  4. **Prezentacja** – warstwa odpowiedzialna za interakcje z użytkownikiem aplikacji, zawiera wszystkie widoki wyświetlane użytkownikowi oraz akcje obsługujące działania użytkownik

1. Projekt bazy danych

Aplikacja wykorzystuje relacyjną bazę danych SQL wygenerowaną przy pomocy Entity Framework zgodnie z metodologią CodeFirst. Każda z tabel bazy posiada klucz główny który zapewnia możliwość łączenia ze sobą dwóch lub więcej tabel. Baza danych została znormalizowana do 3 postaci normalnej, aby zapobiec rozrastaniu i nadmiarowości danych. W celu przyspieszenia przeszukiwania tabel zostały nałożone indeksy na każdy klucz główny.

Poniżej przedstawiono najważniejsze tabele i ich relacje:

**ServiceTypes –** tabela odwzorowująca typ usługi. Oprócz klucza głównego „Id” posiada kolumnę „Name” przechowującą nazwę danego typu usługi. Jest połączona relacją jeden do wielu z tabelą **ServicePhases** (typ usługi ma wiele faz).

**ServicePhases –** tabela przedstawiająca fazy usług. Posiada kolumnę „Order” które przechowuje informację o kolejności wykonywania. Tabela przechowuje również informacje o czasie trwania danej fazy (kolumna „DurationInMinutes”) oraz czasie opóźnienia (kolumna „DelayInMinutes”). Posiada również klucz obcy tabeli ServiceType, umożliwiający relację typu „jeden do wielu”.

**ServiceProviders –** przechowuje informacje o dostawcach usług. Jest połączona z tabelą **ServiceTypes** relacją wiele do wielu (dostawca usługi może wykonywać wiele usług a usługi mogą być wykonywane przez wielu dostawców) przy pomocy dodatkowej tabeli **ServiceProviderServiceTypes.**

**RegistratedServices –** odwzorowuje usługi zarejestrowane przez klientów. Połączona jest z tabelą **ServiceTypes** relacją jeden do wielu (typ usługi może być w wielu zarejestrowanych usługach) oraz z tabelą **ServiceProviders** również relacją jeden do wielu. Oprócz tego przechowuje podstawowe informacje o kliencie który zarejestrował się na usługę.

**Pages –** przedstawia strony edytowane przez administratora. Przy pomocy tabeli **PageFiles** tworzy relację wiele do wielu z tabelą **Files.** Kolumna „RestorePageId” odpowiada za połączenie tej tabeli z samą sobą. Dzięki temu baza danych umożliwia przechowywania historii modyfikacji danych stron.

**MenuButtons –** tabela przedstawiająca przyciski menu. Łączy się sama ze sobą dzięki czemu uzyskujemy strukturę drzewiastą która doskonale odwzorowuje menu. Oprócz tego posiada połączenie jeden do wielu z tabelą **Pages** co pozwala na nawigowanie pomiędzy stronami dzięki stworzonemu menu.

**News –** tabela podobna do tabeli **Pages**. Posiada połączenie wewnętrzne, co umożliwia przechowywanie historii. Oprócz tego za pośrednictwem tabeli **NewsNewsCategories** realizowana jest relacja wiele do wielu z tabelą **NewsCategories** która pozwala na filtrowanie aktualności za pomocą kategorii.

**Settings** – tabela przechowująca informacje o ustawieniach aplikacji. Posiada pole „Value” które może przechowywać dowolną wartość możliwą do zapisania oraz pole InputType dzięki któremu wiemy jaki typ kontrolki należy wyświetlić do edytowania danego ustawienia

Poniżej przedstawiono pełny diagram bazy danych:

Poniżej przedstawiono diagramy sekwencji najważniejszych akcji podejmowanych przez użytkownika:

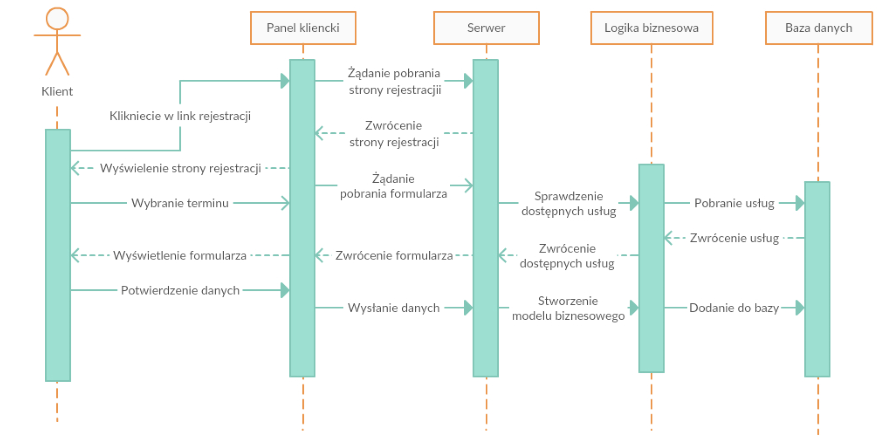
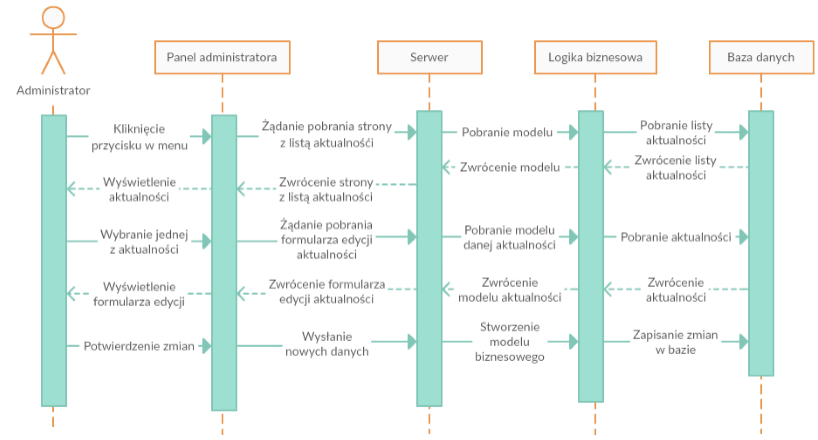
1. Rejestracja na usługę

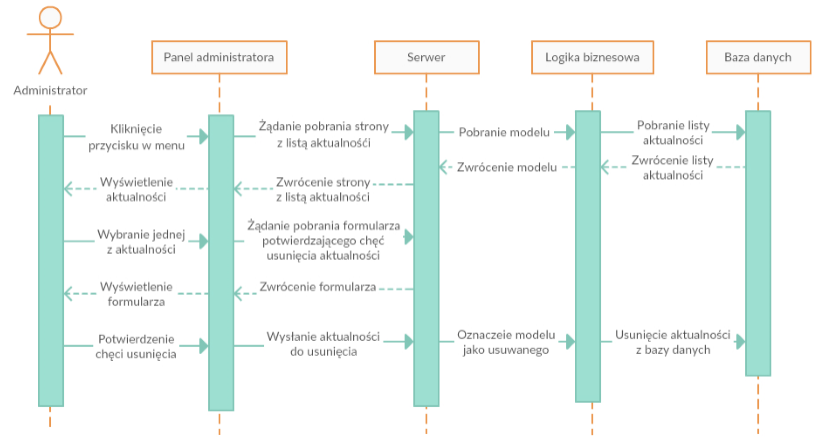
Diagram przedstawia podstawową funkcjonalność aplikacji, jaką jest rejestracja klienta na usługę. Użytkownik poprzez kliknięcie w odpowiedni link sygnalizuję chęć rejestracji. Wyświetlana jest strona z kalendarzem i dostępnymi terminami usług. Użytkownik wybiera odpowiadający mu termin. Następnie wysyłane jest żądanie pobrania formularza z dostępnymi w tym terminie usługami. Kolejnym krokiem jest potwierdzenie przez użytkownika danych i zarejestrowanie usługi w bazie danych.



1. Edycja aktualności

Ze względu na podobieństwo wszystkich funkcjonalności edycji i dodawania, aktualności, stron itd. ograniczono się do przedstawienia jednego przykładu w postaci edycji aktualności.

Użytkownik poprzez menu wybiera typ danych który chce edytować. Wyświetlana jest strona z listą wybranych wcześniej encji. Użytkownik wybiera jedną z encji którą chce edytować i wyświetlany jest mu formularz edycji z danymi. Następnie użytkownik potwierdza dokonanie zmian co skutkuje zapisaniem danych w bazie.

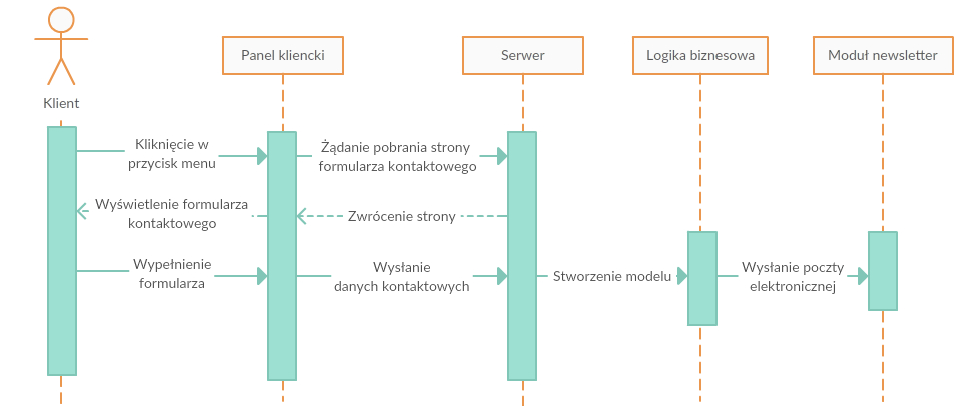
1. Usuwanie aktualności

Usuwanie każdego typu danych jest realizowane w podobny sposób. Tak jak w przypadku edycji czy usuwania użytkownik wybiera typ danych do usunięcia. Następnie wyświetlana jest strona z listą encji. Użytkownik wybiera encję którą chce usunąć, wyświetlany jest mu formularz który pozwala na potwierdzenie chęci usunięcia. Po potwierdzeniu wysyłane jest żądanie usunięcie aktualności, która jest usuwana z bazy danych.

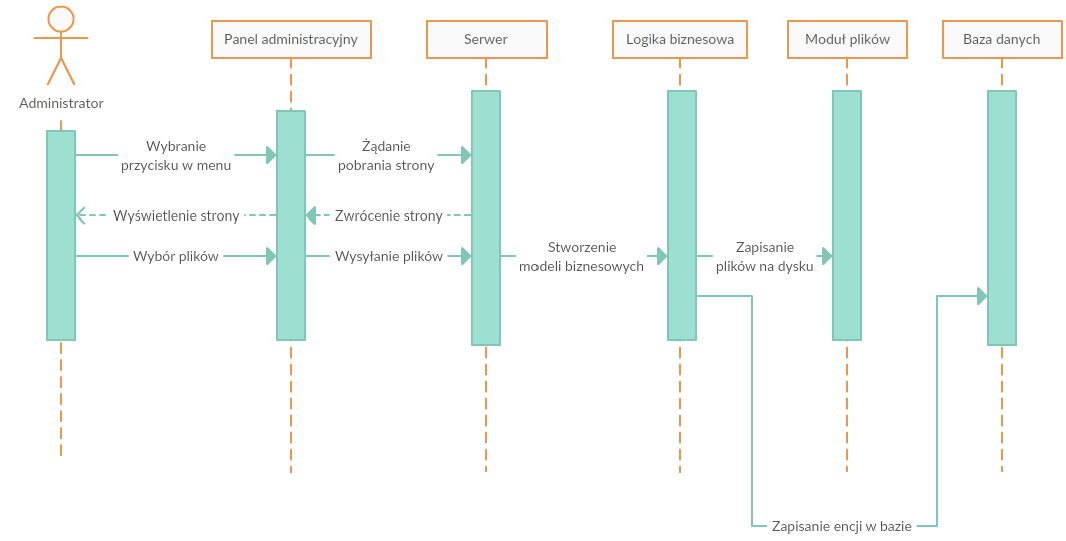
1. Wyświetlenie statystyk



Administrator sygnalizuje chęć przeglądania statystyk odwiedzin, poprzez wybranie w menu odpowiedniego przycisku. Wysyłane jest żądanie pobrania strony z statystykami, następnie serwer pobiera odpowiednie dane z bazy danych i wyświetlana jest strona z danymi przedstawionymi w postaci odpowiedniego diagramu.



Klient wybiera w menu odpowiedni przycisk. Wyświetlana jest mu strona z formularzem kontaktowym. Następnie klient wypełnia formularz danymi i potwierdza wysyłanie wiadomości. System przekazuje dane do modułu newslettera który tworzy i wysyła wiadomość poprzez serwer SMTP.



Administrator wybiera odpowiedni przycisk w menu. Następnie wybiera pliki które chce przekazać na serwer i potwierdza ten wybór. Pliki są zapisywane na dysku oraz tworzone są encje opisujące te pliki do zapisania w bazie danych .