1. Wymagania biznesowe

ServiceCMS to aplikacja webowa 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. Oprócz tego aplikacja umożliwia zarządzanie wszelakimi treściami takimi jak strony i aktualności, umożliwia zarządzanie plikami i udostępnia możliwość przeglądania statystyk odwiedzin panelu klienckiego. Oprogramowanie może obsługiwać osoba nie techniczna z podstawową znajomością obsługi komputera.

1. Słownik pojęć

**Administrator** – osoba specjalnie wyznaczona do zarządzania treścią wyświetlanej 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 administracyjny

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

**Strona** – typ treści który może zawierać wiele elementów takich jak obrazki 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 zarejestrowanych już usług
* Możliwość dodawania, usuwania i edytowania aktualności
* Możliwość dodawania, usuwania i edytowania dostawców usług
* Możliwość dodawania, usuwania i edytowania typów usług
* Możliwość organizacji przycisków 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 nie funkcjonalne

* 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 i Firefox

1. Warstwy aplikacji

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

* 1. **Rdzenia** – warstwa odpowiadająca za połączenie i obsługę bazy danych, obsługę i logowanie wyjątków oraz zawierająca klasy wykorzystywane w całej aplikacji.
  2. **Logiki biznesowej** – warstwa zawierająca implementacje algorytmów logiki biznesowej takiej jak zapisywanie, edytowanie i usuwanie modeli wykorzystywanych w aplikacji.
  3. **Modułów** – warstwa zawierająca wymienialne rozwiązania niezależne od reszty aplikacji np. obsługa zapisu i usuwania plików, łączenie z zewnętrznym serwisem lub obsługa kryptografii haseł.
  4. **Prezentacji** – 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 CodeFirst. Każda z tabel bazy posiada klucz główny które zapewniają możliwoć łączenie 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” poisada 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 zarejestrowane już usługi. 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 się zarejestrował.

**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 właśnie menu.

**News –** tabela podobna do tabeli **Pages**. Również posiada połączenie sama ze sobą co umożliwia przechowywanie historii. Oprócz tego za pośrednictwem tabeli **NewsNewsCategories** realizowana jest relacja wiele do wielu z tabelą **NewsCategories** która pozwala na filtracje aktualności za pomocą kategorii.

Poniżej przedstawiono pełny diagram bazy danych: