

1. Wstęp

Obecna sytuacja rynkowa związana z istnieniem dużej ilości firm w zakresie każdej branży wymusza na przedsiębiorcach ciągłe zwiększanie swojej atrakcyjności oraz konkurencyjności w celu zdobycia potencjalnego klienta. W realizacji powyższego zadania konieczne jest wdrażanie nowoczesnych metod umożliwiających między innymi obniżenie kosztów pracy, co wprost przekłada się na niższą cenę produktu końcowego. Niższa cena dla klienta jest jednym z najważniejszych elementów, który determinuje wybór konkretnego usługodawcy. Aspekt ten dotyczy także branży hotelarskiej, która w sposób bezpośredni dotyczy naszej pracy.

Liczba hoteli w Polsce od kilkunastu lat nieprzerwanie rośnie¹. Implikuje to konieczność wzmożonej walki o klienta w celu utrzymania się na rynku. Obniżenie cen usług wynajmu pokoi, sal konferencyjnych oraz cateringów przy zachowaniu odpowiedniego poziomu dochodów jest problemem, z którym zmagają się każda sieć hotelowa. Przedsiębiorcy prześcigają się w znajdowaniu coraz to nowszych i efektywniejszych rozwiązań, które mają na celu rozwiązanie owego problemu. W XXI wieku dużą pomocą w tym zakresie okazuje się informatyzacja.

2. Geneza projektu

Podczas przeprowadzania transakcji dotyczącej organizacji szkolenia w jednej z sieci hotelowych, dostrzeżony został potencjalny problem wynikający z manualnego operowania wszelkimi danymi przez pracowników działu sprzedaży. Pracownicy Ci w czasie spotkania z klientem mają za zadanie ustalenie wszelkich wartości cenowych usług wybranych przez klienta. Usługi te bezpośrednio związane są z typem wydarzenia, które klient chce zorganizować. Przeważnie są to wszelkiego rodzaju konferencje, kilkudniowe szkolenia firmowe, ale także imprezy weselne, urodzinowe, czy spotkania rodzinne. Przez wzgląd na różną specyfikę powyższych przedsięwzięć sieć hotelowa dysponuje szeroką gamą usług z nimi związanych.

W czasie tworzenia przez sprzedawcę propozycji cenowej organizowanego wydarzenia, klient ma możliwość wyboru odpowiedniej dla jego potrzeb sali konferencyjnej lub bankietowej wraz z jej dodatkowym wyposażeniem, ustalenia aspektów gastronomicznych, takich jak liczba i rodzaj posiłków oraz napojów, a także wyborem odpowiedniego typu pokoju hotelowego. Po wstępnym wyborze zakresu usług, elementy te uzupełniane są o liczbę osób, które będą w wydarzeniu uczestniczyć, a także ilość dni jego trwania. Kolejnym etapem jest negocjacja rabatów oraz finalnie ustalenie formy zapłaty za organizowane przedsięwzięcie. Aktualnie operacja ta wraz z procesem tworzeniem dokumentu propozycji cenowej przeprowadzana jest za pomocą aplikacji

¹ <http://www.wykresy.net/liniowe/liczba-hoteli-w-polsce-ostatnie-12-lat.html>

Microsoft Excel jak również w wersji papierowej. Obydwie metodologie generują liczne problemy, które bezpośrednio wpływają na zwiększenie kosztów generowanych przez pracowników, a co za tym idzie ceny oferowanych usług. Pierwszy problem odnosi się do aspektu utrudnienia aktualizacji danych, który jest nieodłącznym elementem branży. Ceny pokoi hotelowych ulegają częstym modyfikacjom oraz uzależnione są od sytuacji rynkowej, organizowanych okolicznych wydarzeń rozrywkowych i kulturalnych oraz samego faktu wolnej ich ilości w danym czasie. Analogicznie sytuacja przedstawia się w przypadku produktów gastronomicznych oraz pozostałych wynajmowanych pomieszczeń.

Aktualnie wszelkie modyfikacje cenników pokoi hotelowych, sal możliwych do wynajęcia oraz dostępnych towarów dokonywane są przez menadżera sprzedaży, który za pośrednictwem poczty elektronicznej przekazuje uaktualnione wersje arkuszy kalkulacyjnych oraz listy cen określonym sprzedawcom. Częste zmiany zobowiązują pracowników działu sprzedaży do ciągłej kontroli skrzynki elektronicznej oraz powodują presję spowodowaną posiadaniem potencjalnie nieaktualnych danych. Zaawansowane arkusze kalkulacyjne wykonane w aplikacji Excel podatne są na błędy ludzkie, wymuszają na pracownikach dobrą znajomość oprogramowania oraz umiejętność jego obsługi, co z kolei niekorzystnie przekłada się na nowych pracowników oraz pracodawcę, który zobowiązany jest przeprowadzać długie i kosztowne szkolenia w tym zakresie. Kolejnym dostrzeżonym problemem jest zagadnienie czasochłonności wyszukiwania danych dotyczących produktów znajdujących się na hotelowej restauracji i ich cen. Pracownik otrzymuje rozbudowaną listę produktów wraz z przyporządkowanymi im cenami i we własnym zakresie zobligowany jest znaleźć interesującą go pozycję w czasie przygotowywania propozycji cenowej.

Dostrzegając powyższe problemy zdecydowano się wykonać dedykowaną aplikację bazodanową, której celem jest ich rozwiązanie. Założono, że konieczne by było, aby każdy ze sprzedawców posiadał możliwość uruchomienia na swoim służbowym komputerze programu, który dzięki połączeniu z bazą danych i wykorzystaniu jej funkcjonalności umożliwiłby przyspieszenie procesu przeprowadzania transakcji z klientem poprzez automatyczną aktualizację cenników oraz przedstawienie listy produktów gastronomicznych w skategoryzowany sposób. Dzięki przechowywaniu wszystkich propozycji cenowych w bazie danych, zanikałaby konieczność samodzielnej ich kategoryzacji i dbania o bezpieczeństwo na lokalnym komputerze. Umożliwiłoby to pracownikom pracę na różnych maszynach, co w znaczny sposób zwiększyłoby ich elastyczność.

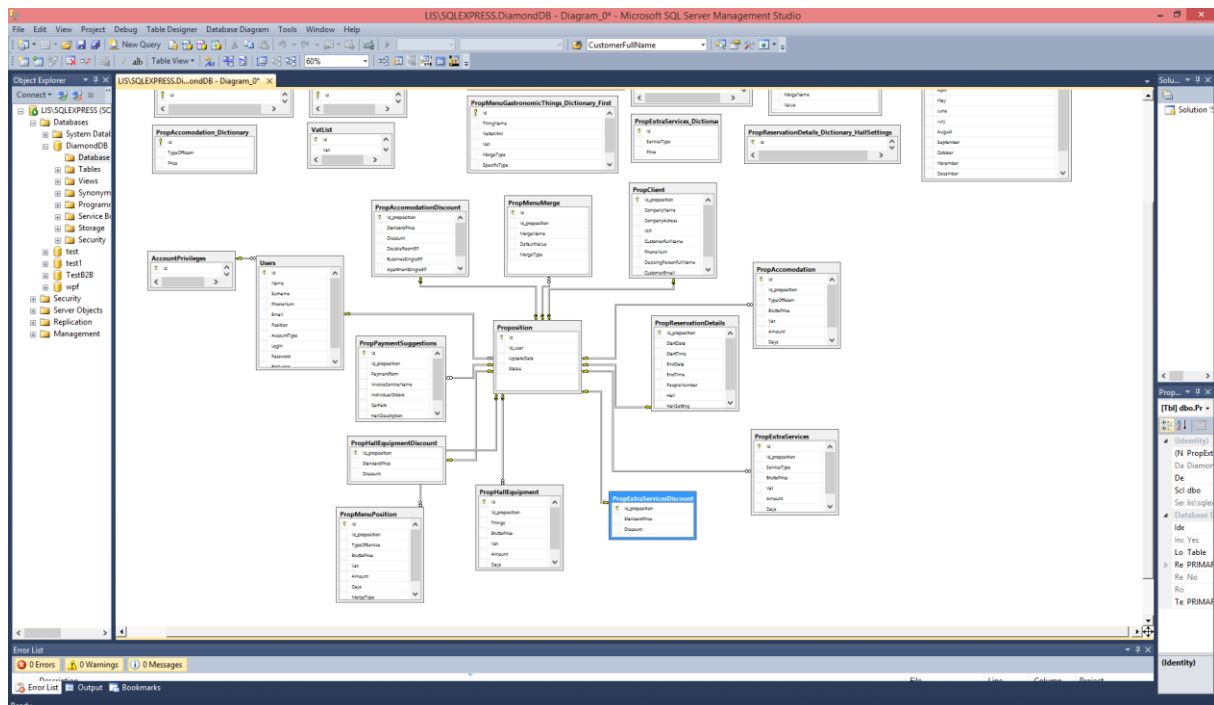
Kolejnym atutem przedstawionego rozwiązania byłoby umożliwienie kierownictwu z poziomu kont administratorskich kontroli pracowników poprzez zdalny wgląd w przygotowywane przez nich propozycje cenowe, a co za tym idzie ustalanych z klientem cen oraz zniżek. Menadżer sprzedaży za pomocą swojego uprzywilejowanego konta posiadałby również możliwość aktualizacji odpowiednich tabel danych, dzięki czemu zmiany natychmiastowo byłyby widoczne na kontach sprzedawców. Zmiany te

- ### 5.1. Wykorzystane narzędzia

- ### 5.1.1. Visual Studio 2013

- ### 5.1.2. SQL Server Management Studio

W trakcie prac nad projektem zostało wykorzystane narzędzie do zarządzania bazą danych SQL Server Management Studio, które ułatwiło nam prace i proces projektowania całej bazy danych. Dzięki przejrzystemu interfejsowi użytkownika, można w bardzo łatwy sposób tworzyć nawet bardzo skomplikowane struktury, bez konieczności znajomości języka tworzącego bazę danych, wykorzystując ten program wspomogło, co proces tworzenia wymaganej przez sieć hotelową bazy. Istnieje również możliwość generowania skryptów, których zadanie jest odtworzenie całej struktury bazy danych. W program jest również możliwość tworzenia diagramu wszystkie jak również tylko wybranych encji, można to zaobserwować na poniższym zrzucie ekranu, co umożliwiło diagnozowanie problemu braku połączeń niektórych tabel z resztą struktury.



Wykorzystując wbudowane mechanizmy możliwe jest również łatwe wpisywanie początkowych danych takich jak użytkownicy systemu, cennych produktów jak również ich edycja na poziomie bazy również, co umożliwiło na sprawdzenie czy baza działa z naszymi oczekiwaniami. Mogliśmy śledzić w

trakcie działania naszego programu czy wszystkie elementy prawidłowo działają i czy w poprawny sposób jest zapisywana dana propozycja.

5.1.3.Trello

5.1.4.SourceTree + GitHub

Aktualnie zespoły programistyczne nie są jedno osobowe, lecz wieloosobowe wnikała więc potrzeba by ułatwić pracę wielu osobą nad jednym projektem, to zadanie spełniają repozytoria. Istnieje wiele typów systemu kontroli wersji scentralizowane, rozproszony i lokalne. Z racji potrzeby posiadania systemu kontroli wersji spowodowany ilością osób i stylem tylko dwa z nich spełniały wymagania projekty wybór padł na rozproszony, ponieważ spowodowane to było możliwością tworzenia własnych ścieżek rozwojowych, co pozwoliło zabezpieczenie główną ścieżkę przed błędami i umożliwiło tworzenie funkcjonalności, które zostały wprowadzone w późniejszych etapach projektu. Wymaganie, które stawialiśmy przed tego typu serwisem były głównie dwa: musiało być darmowe i musiał być rozproszony. Wybór padł na Github'a z powodu możliwości tworzenia prywatnego repozytorium, aby nikt nie mógł w nie ingerować. Dzięki wbudowane mechanizmy zaimplementowane w repozytorium mogliśmy z łatwością cofnąć się do działającej wersji lub też wykorzystać rzeczy, które zostały nadpisane lub usunięte, istnieje również możliwość rozwiązywania konfliktów, jeśli dany członek zespołu ma zmiany w tym samym pliku i danej linijce, co na serwerze to istnieje możliwość rozwiązania konfliktu bez konieczności większej ingerencji w kod programu.

Niestety możliwe jest wykorzystanie tego typu rozwiązania z poziomu wiersza poleceń, dlatego zastosowaliśmy nakładkę graficzną SourcTree by móc szybciej i sprawniej przełączać się między ścieżkami rozwoju dzięki przejrzystemu interfejsowi mogliśmy też rozwiązywać konflikty w plikach, łączyć różne ścieżki rozwoju. Dzięki jednemu scentralizowanemu narzędziu istnieje również możliwość podłączanie różnych repozytoriów z innych projektów.

Resharper

5.1.5.Jenkins

- 5.1.5.1. Użyte pluginy
- 5.1.5.2. Konfiguracja Jenkins
- 5.1.5.3. Instrukcja obsługi

5.2. Biblioteki zewnętrzne

- 5.2.1.Extended WPF Toolkit
- 5.2.2.PDFsharp & MigraDoc
- 5.2.3.[GalaSoft.MvvmLight](#)
- 5.2.4.[EntityFramework](#)

6. Implementacja

6.1. Architektura aplikacji

- 6.1.1.Przeznaczenie wzorców architektonicznych
- 6.1.2.Zastosowany wzorzec architektoniczny – MVVM
- 6.1.3.Model aplikacji /klasy/

- 6.2. Architektura bazy danych
 - 6.2.1. Definicja bazy danych
 - 6.2.2. MS-SQL – Zastosowany system zarządzania bazą danych
 - 6.2.3. Budowa bazy danych użytej w projekcie
 - 6.2.3.1. Model relacyjny bazy danych
 - 6.2.3.2. Tabele słownikowe
- 6.3. Komunikacja bazy danych z projektem programistycznym
 - 6.3.1. Narzędzia ORM
 - 6.3.2. Zastosowane oprogramowanie mapowania obiektowo relacyjnego – Entity Framework
 - 6.3.3. Dostosowanie Entity Framework do wymagań związanych ze wzorcem MVVM.
 - 6.3.4. Użycie bazy danych ze strony projektu C#
- 6.4. Automatyzacja wydawania kolejnych wersji programu
 - 6.4.1. Continuous Integration – definicja
 - 6.4.2. Continuous Deployment – definicja
 - 6.4.3. Continuous Delivery – definicja
 - 6.4.4. Schemat budowania programu (build, static code analysis, e-mails, zip into server? pass)
- 7. Specyfikacja zewnętrzna
 - 7.1. Logowanie
 - 7.2. Tworzenie nowej propozycji cenowej
 - 7.2.1. Opis klienta
 - 7.2.2. Szczegóły rezerwacji
 - 7.2.3. Usługi gastronomiczne
 - 7.2.4. Usługi noclegowe
 - 7.2.5. Usługi dodatkowe i forma płatności
 - 7.3. Edycja istniejącej propozycji cenowej
 - 7.4. Tworzenie pliku PDF istniejącej propozycji cenowej
 - 7.5. Modyfikacja kont **użytkowników/sprzedawców**
 - 7.5.1. Dodawanie konta
 - 7.5.2. Edycja konta
 - 7.5.3. Usuwanie konta
 - 7.6. Modyfikacja słowników cenowych
 - 7.7. Zarządzanie kontem sprzedawcy oraz administratora (obsługa)
- 8. Specyfikacja wewnętrzna
 - 8.1. Logowanie
 - 8.1.1. Funkcja haszująca
 - 8.1.2. SHA-2
 - 8.1.3. Pierwsze logowanie
 - 8.2. Konto administratora i użytkownika – ListView
 - 8.3. Tworzenie propozycji cenowej
 - 8.3.1.
 - 8.4. Edycja słowników oraz kont użytkowników
 - 8.4.1. DataGridView a dynamiczna komunikacja z Bazą danych
 - 8.5. Tworzenie szablonu PDF propozycji cenowej
- 9. Perspektywy rozwoju oprogramowania

