CONTACT MANAGER

Dokumentacja Projektu

**• nr indeksów członków grupy**

148693,

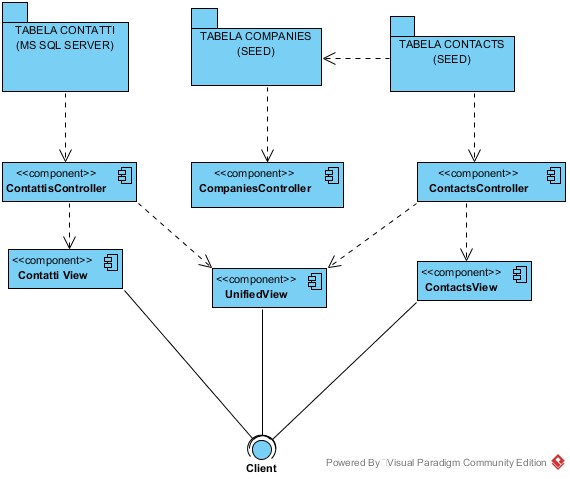
**• użyte technologie/ języki programowania**

Środowisko: Visual Studio 2015, ASP.NET Web Application

Język: C#

Bazy Danych: Microsoft SQL Server i Seed do lokalnej bazy danych MySql

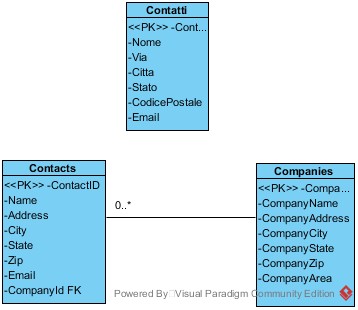
**• schemat oraz opis architektury systemu**



W projekcie są 3 tabele, z czego 1 (Contatti) jest tabelą stworzoną w MS SQL server połączoną z projektem oraz 2 (Companies i Contacts) zostały seedowane bezpośrednio z aplikacji.

Każda z tabel/modeli posiada własny controller ze swoimi metodami GET gdzie pobierane są odpowiednie dane, następnie są wyświetlane w View dla danego Controllera. Jest jeszcze Unified View który łączy dane ze wszystkich tabel.

**• opis dostawców, struktura encji**



Contatti: PK – ContactID

Contacts: PK – ContactID, FK – CompanyID

Companies: PK – CompanyID

Contatti jest główną tabelą z informacjami i danymi na temat kontaktów, Contacts jest odmianą tej tabeli posiadającą dodatkowo informacje o firmie w której każdy kontakt pracuje.

**• opis huba, przebieg integracji encji, struktura wynikowa encji**

Klient ma 3 widoki do wyboru: Contatti, Clients oraz Połączony widok, który przedstawia informacje ze wszystkich tabeli. Admin jest w stanie zmieniac i usuwać danę natomiast użytkownik nie. Integracja encji wykonana jest za pośrednictwem Controllerów i RESTful API, metodami get i set są aktualizowane na bieżąco (zmieniając dane w bazie danych SQL server zmiana następuję automatycznie w hubie).

**• napotkane problemy**

Wszystkie, przez automatyzacje w tym środowisku bardzo ciężko było zrobić coś indywidualnego, natomiast każda zmiana w strukturze danych wiążę się z wieloma procedurami i już nie jest aktualizowana automatycznie. Dużym problemem stanowiło połączenie wszystkich tabel w jedną, należało w tym celu dobrać odpowiedni model danych oraz odpowiedni typ przechowywania (IEnumerable czy lista bądź inny, ale za każdym razem wysypywał się błąd) a wszystkie informacje w internecie wskazywały na bezpośrednie użycie jednej tabeli zamiast dwóch lub więcej, więc trzeba było to zrobić metodą prób i błędów przez wiele godzin.

**• adres do repozytorium**

<https://github.com/Dante404/IAS>

**• podział pracy w grupie**

Angelo Pitarresi – Architektura systemu, programowanie

**dodatkowo: Co byśmy zmienili gdybyśmy robili ten projekt jeszcze raz**

środowisko, strukturę danych