



# AGH

**AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA  
W KRAKOWIE**

## Wzorce Projektowe

Implementacja aplikacji wykorzystującej specyfikację HTML5 pozwalającej na przechowywanie informacji w bazie danych przeglądarki oraz synchronizację z bazą centralną.

Autorzy:

Konrad Gębczyński

Mateusz Wiater

Rafał Krzyś

Przemysław Michałek

## 1. Wprowadzenie

Poniższa dokumentacja opisuje architekturę oraz styl zaprojektowania systemu dodawania osób fizycznych do bazy danych w postaci aplikacji webowej w standardzie HTML5 [1]. Technologia HTML5 powstała w 2014 roku i w dobie urządzeń mobilnych jest aktualnie najbardziej wspieranym standardem dla tworzenia stron internetowych, wartymi wspomnienia zaletami HTML5 są:

- (a) **Wideo** - HTML5 pozwala na dodanie pliku wideo bezpośrednio na stronę, bez potrzeby używania wtyczki w przeciwieństwie do standardu HTML4, gdzie najczęściej rozwiązywano ten problem dodając powszechnie znanego Flash'a - ta zaleta ma szczególne znaczenie dla użytkowników Apple'a który znany był z konsekwentnego blokowania wtyczki Flash na swoich produktach [2].
- (b) **Geolokacja** - Bardzo ważna cecha w dzisiejszych czasach (choć nie zawsze mile widziana), standard HTML5 pozwala serwerowi zlokalizować użytkownika zarówno po adresie IP (dla komputerów stacjonarnych) jak i sygnale GPS, połączeniu Wi-Fi czy Bluetooth [3].
- (c) **Canvas** - Pozwala na lepsze zarządzanie i manipulowanie grafiką bezpośrednio na stronie internetowej. Używa JavaScriptu do dynamicznego rysowania obrazów, jest to tak naprawdę kolejne zastąpienie niemiłe widzianego Flash'a, więcej o Canvas w artykule [4].

Standard HTML5 został po raz pierwszy zaprezentowany w 2007 roku i otrzymał status rekomendowanego języka w październiku 2014 roku i jest ciągle wspierany jako główna technologia tworzenia stron internetowych, jego najnowsza wersja HTML 5.2 została zatwierdzona w 2017, w planach jest już wersja 5.3, więcej informacji w artykule [5].

## 2. Cele projektu

Celem projektu jest stworzenie aplikacji webowej pozwalającej na modyfikowanie bazy danych znajdującej się po stronie serwera z uwzględnieniem sytuacji braku połączenia z siecią: aplikacja powinna odpowiednio dostosować się do sytuacji i w razie braku połączenia dalej zapisywać zmiany w swojej lokalnej bazie danych i w momencie odzyskania połączenia uaktualnić serwerową bazę, podobnie jak jest to rozwiązane w technologii GIT (możliwość wysyłania commit'ów offline i wywoływania komendy *push* w momencie posiadania połączenia z internetem).

## 3. Zasady działania systemu

Wszystkie operacje operujące na bazie danych wykonywane są po stronie użytkownika i zapisywane w indexedDb. Naciśnięcie przycisku synchronizacji wywołuje proces synchronizacji lokalnej bazy z centralną na zasadzie który rekord ma wcześniejszą datę zapisu.

## 4. Opis funkcjonalności

Użytkownik ma możliwość dodawania/uaktualniania/usuwania oraz dodawania zależności dla rekordów w prostym systemie CRM. System zawiera 4 zakładki: Account, Contact, Asset, Opportunity, daje to możliwość prostego zarządzania przedsięwzięciami w firmie.

## 5. Wykorzystane technologie

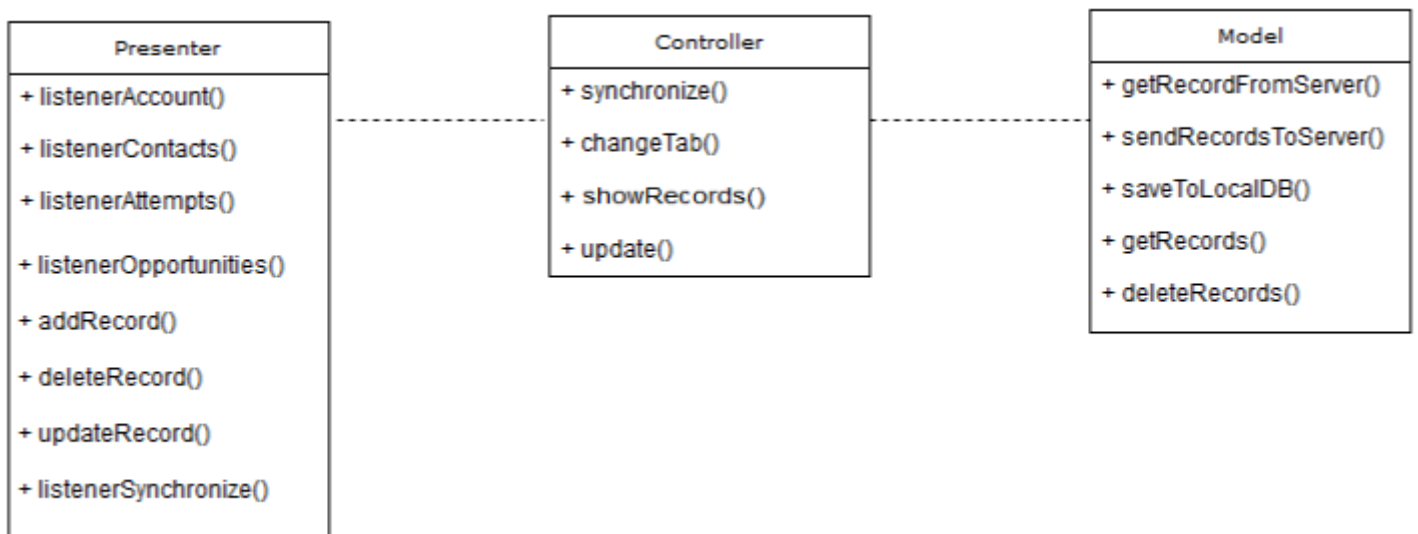
Do wykonania projektu wykorzystano następujące technologie:

- Spring - wykorzystany w celu udostępnienia API po stronie serwera, które służy do udostępniania i odbierania rekordów naszego systemu CRM.
- JavaScript - wykorzystany po stronie klienta do obsługi bazy danych przeglądarki oraz do łączenia z serwerem.
- TypeScript - wykorzystywany w celu wprowadzenia obiektowości do projektu i wykorzystania wzorca Buildera.
- HTML5
- API IndexedDB - baza danych przeglądarki na której operuje klient

## 6. Wybrane wzorce

- Model View Presenter

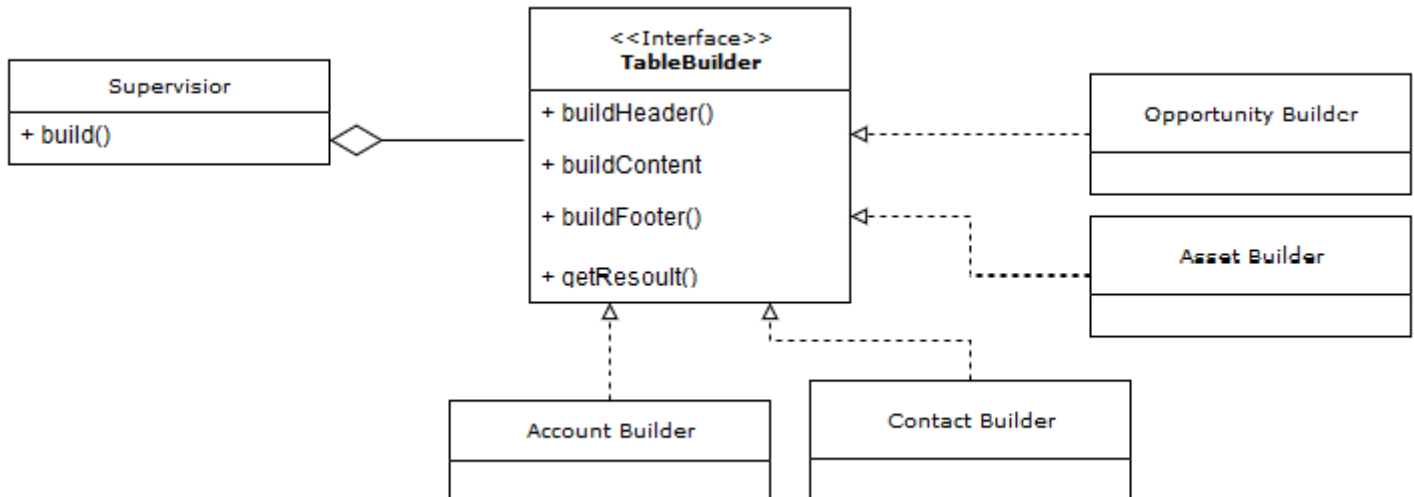
Model View Presenter jest wzorcem powstałym na bazie wzorca MVC. Dane nie są przekazywane bezpośrednio z modelu do widoku jak to ma miejsce w MVC. Prezener wysyła zapytanie do modelu, model zwraca dane do prezentera, prezenter przetwarza otrzymane dane i przekazuje do widoku.



Rys. 1: MVP - diagram klas

- Builder

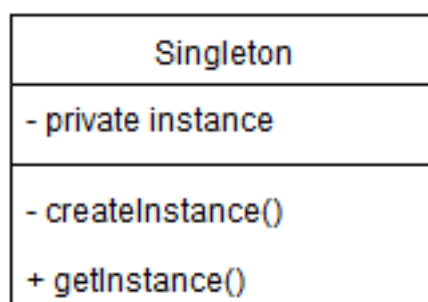
Wzorzec Builder pozwala na podział skomplikowanego procesu tworzenia obiektu, na kilka mniejszych etapów, gdzie każdy z nich może być implementowany na różne sposoby. Jest on interfejsem, który pozwala budować części takiego obiektu.



Rys. 2: Builder - diagram klas

- Singleton

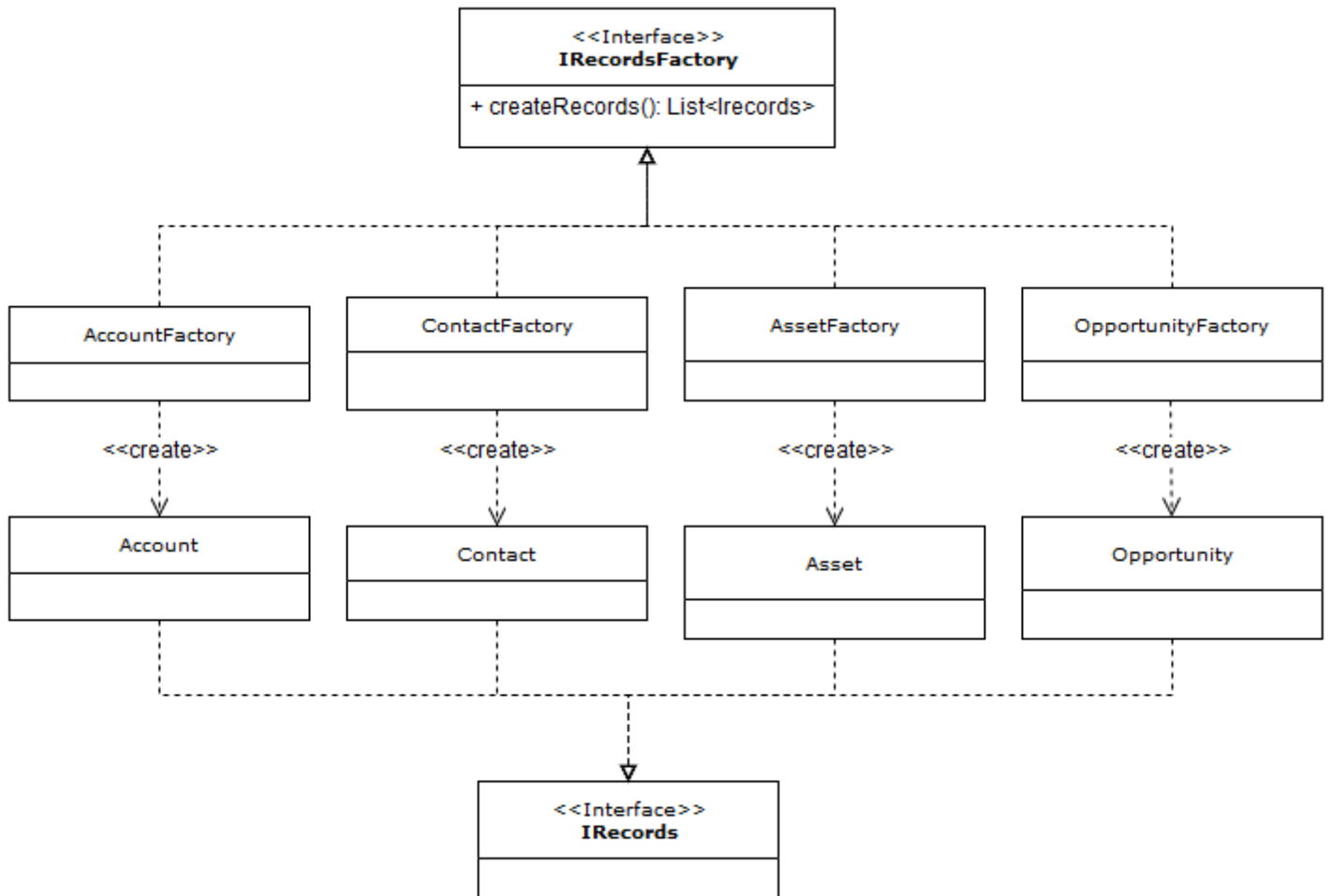
Umożliwia stworzenie tylko jednej instancji obiektu danej klasy. Tylko ten obiekt będzie używany w ramach całej aplikacji.



Rys. 3: Singleton - diagram klas

- Abstract Factory

Pozwala tworzyć powiązane lub zależne obiekty bez specyfikowania ich konkretnych klas. Tworzone obiekty zwykle implementują ten sam interfejs. Wzorec ten kładzie nacisk na tworzenie obiektów konkretnej rodziny, a nie na sposób w jaki te obiekty są tworzone.



Rys. 4: Abstract Factory - diagram klas

## Literatura

- [1] Matthew B Hoy. Html5: a new standard for the web. *Medical reference services quarterly*, 30(1):50–55, 2011.
- [2] Jamie Johnson, Stephen Willes, William Travis Smith, and Alan Morine. Flash isolation for device covers, March 10 2015. US Patent 8,973,752.
- [3] Anthony T Holdener. *HTML5 geolocation*. Ó'Reilly Media, Inc.", 2011.
- [4] w3schools. Html canvas reference, 2014.
- [5] w3schools. Html 5.3 editor's draft, 2018.