Wedding Planner

# Założenia wstępne

## Cel projektu

Celem projektu ma być przygotowanie systemu umożliwiającego tworzenie listy prezentów ślubnych przez narzeczonych. Goście powinni mieć możliwość rezerwowania prezentów z listy życzeń ślubnych państwa młodych. Konieczne musi być wtedy podanie swojego imienia, nazwiska oraz maila.

Serwer główny będzie synchronizował się z serwerami zawierającymi bazy danych z listami prezentów, w celu zarezerwowania odpowiedniej liczby produktów w sklepie.

## Zakres projektu

### moduł główny

organizuje wesela. łączy się z serwerem lokalnym trzykrotnie (gdy para młoda wypełni wszystkie najistotniejsze informacje na serwerze lokalnym, wraz z końcem terminu potwierdzania obecności gości na weselu oraz 2 tyg. przed weselem).

* ma przechowywać końcową listę prezentów - ile należy zarezerwować w sklepie prezentów
* liczbę osób, które potwierdziło obecność (dzieci do 7 lat, młodzież do 18 lat, dorośli) - dla firmy cateringowej, aby wiedziała ile jedzenia przygotować oraz do rezerwacji odpowiedniej wielkości sali weselnej
* termin wesela - do rezerwacji sali weselnej

### moduł lokalny

przechowuje dane konkretnej pary narzeczonych:

* ich imiona i nazwiska
* imienną listę gości wraz z przedziałem wiekowym (dzieci/młodzież/dorośli)
* listę prezentów, które chcą dostać młodzi
* możliwość rezerwacji prezentów
* termin wesela

## Kontekst biznesowy

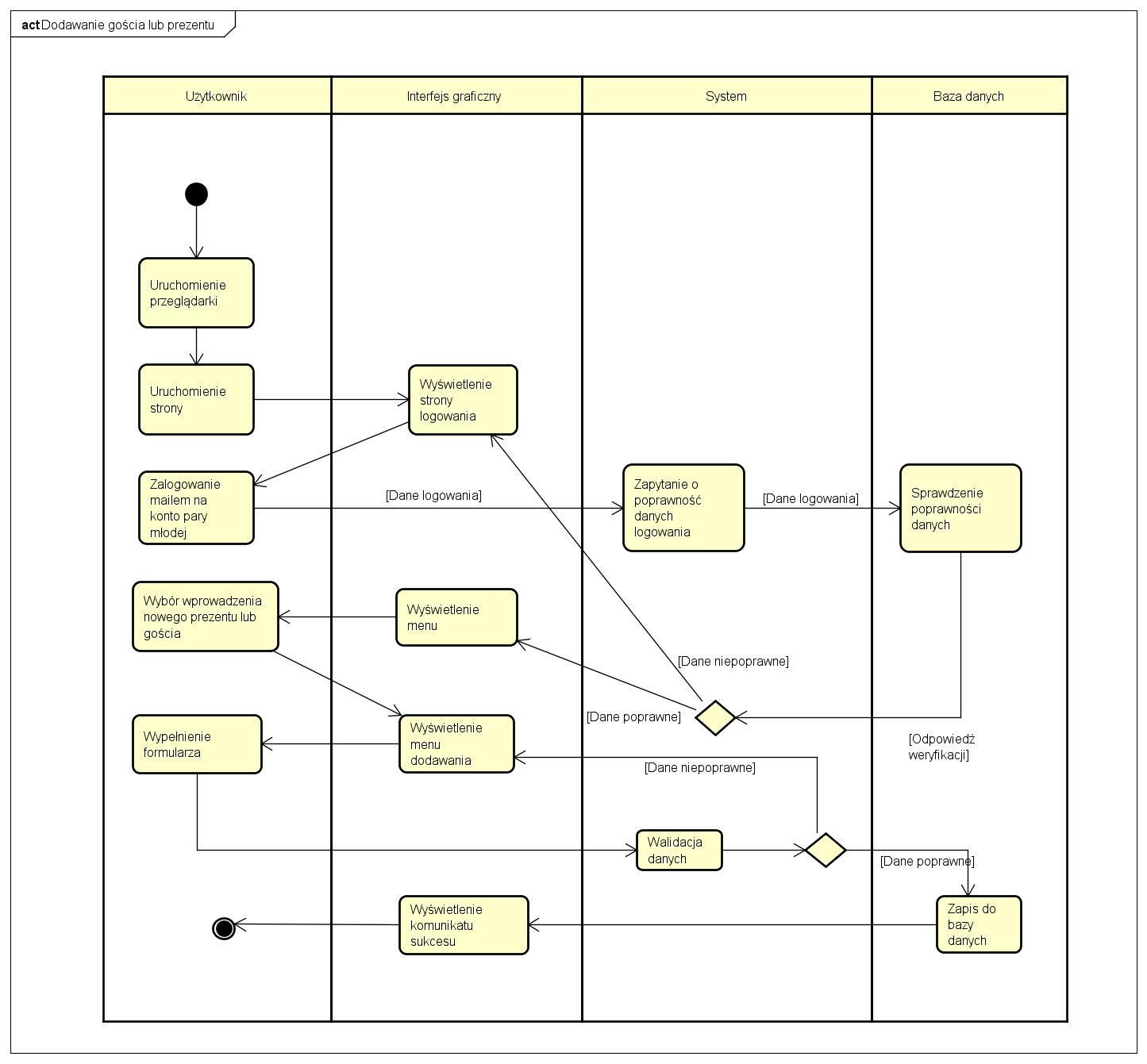
Specjalnością Weddings Planner będzie kompleksowa organizacja ślubów i wesel.

Aplikacja będzie pomagała przede wszystkim w kreacji i projektowaniu idealnej uroczystości ślubnej. Stworzona zostanie zatem koncepcja w oparciu o nowe trendy. W odniesieniu do oczekiwań, zaoferowane będą obiekty zlokalizowane na terenie całej Polski, zapewniające wysoki standard obsługi, wyróżniające się spójnym stylem oraz ładnym otoczeniem.

Ponadto umożliwione będzie internetowe potwierdzanie przez gości obecności. Nie będzie konieczności dzwonienia i dopytywania się, czy aby na pewno zaproszony gość będzie mógł się pojawić na weselu. W odpowiednim czasie dostanie on maila z prośbą o potwierdzenie obecności.

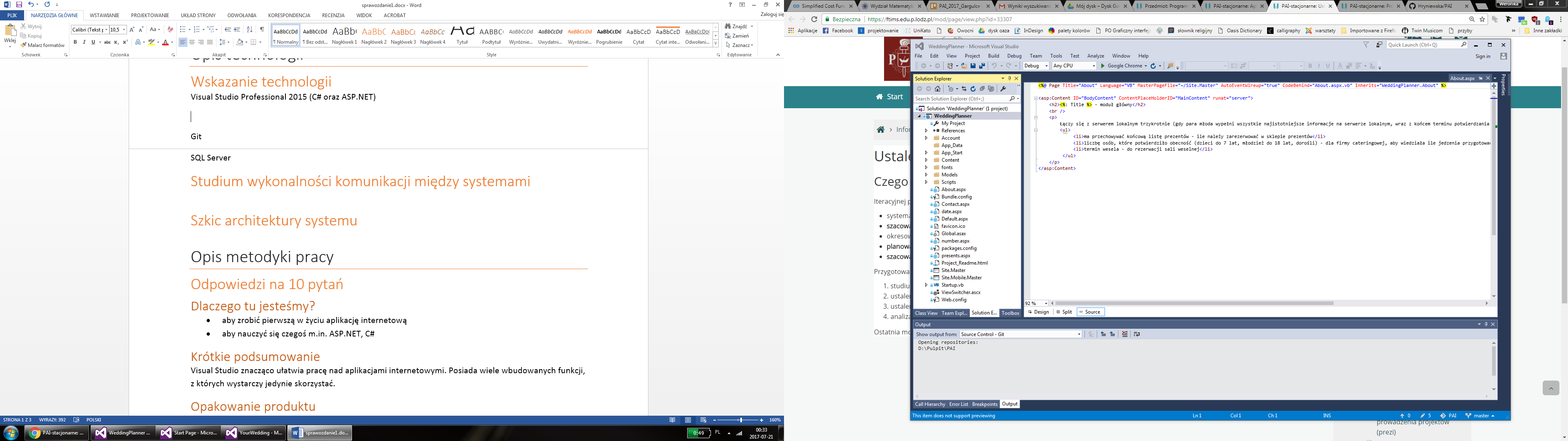
Nie będzie również problemu z nietrafionymi prezentami lub sytuacjami, gdy kilka osób kupi ten sam prezent. Państwo młodzi mogą ułożyć listę prezentów, z której zaproszeni goście będą mogli sobie zarezerwować prezent, który chcieliby kupić.

### Diagram czynności UML[[1]](#footnote-1)



# Opis technologii

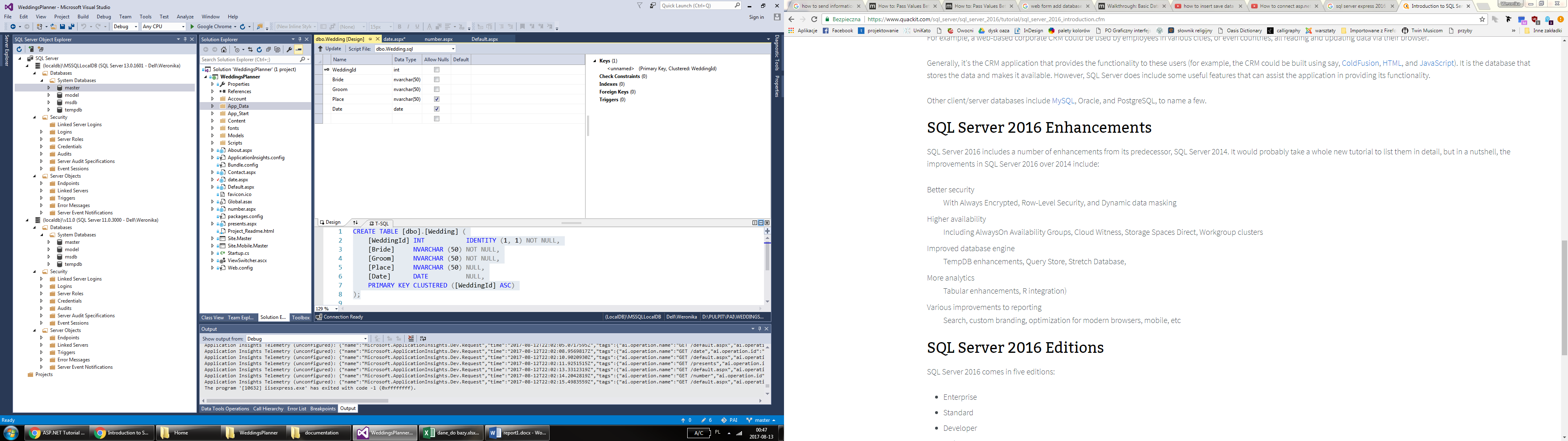
## Wskazanie technologii

Visual Studio Professional 2015 (C# oraz ASP.NET)

Git (z wykorzystaniem programu TortoiseGit)

Adres dostępu do repozytorium z kodem: [github.com/Hryniewska/PAI](https://github.com/Hryniewska/PAI)



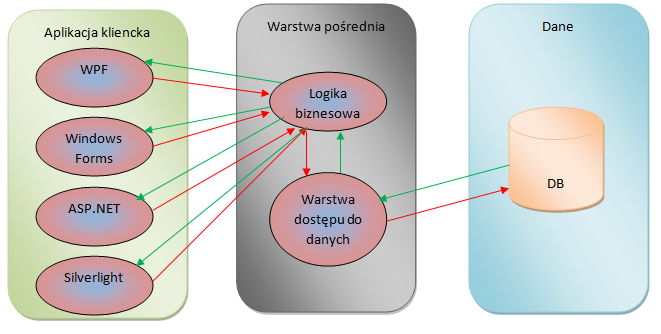
SQL Server 2016

## Studium wykonalności komunikacji między systemami

Istnieje kilka możliwości, aby dane z aplikacji klienckiej mogły być pobierane przez serwer główny. Zostały one opisane na stronie internetowej w temacie o tytule „How to transfer data from one website to another in ASP.Net”[[2]](#footnote-2)

## Szkic architektury systemu[[3]](#footnote-3)

ASP.NET cechuje się architekturą trójwarstwową (three-tier architecture), która pozwala na rozdzielenie systemu informatycznego na składowe zwane warstwami . Dzięki takiemu podziałowi zyskuje się przede wszystkim łatwość wymiany modułów, bez potrzeby aktualizowania pozostałych.

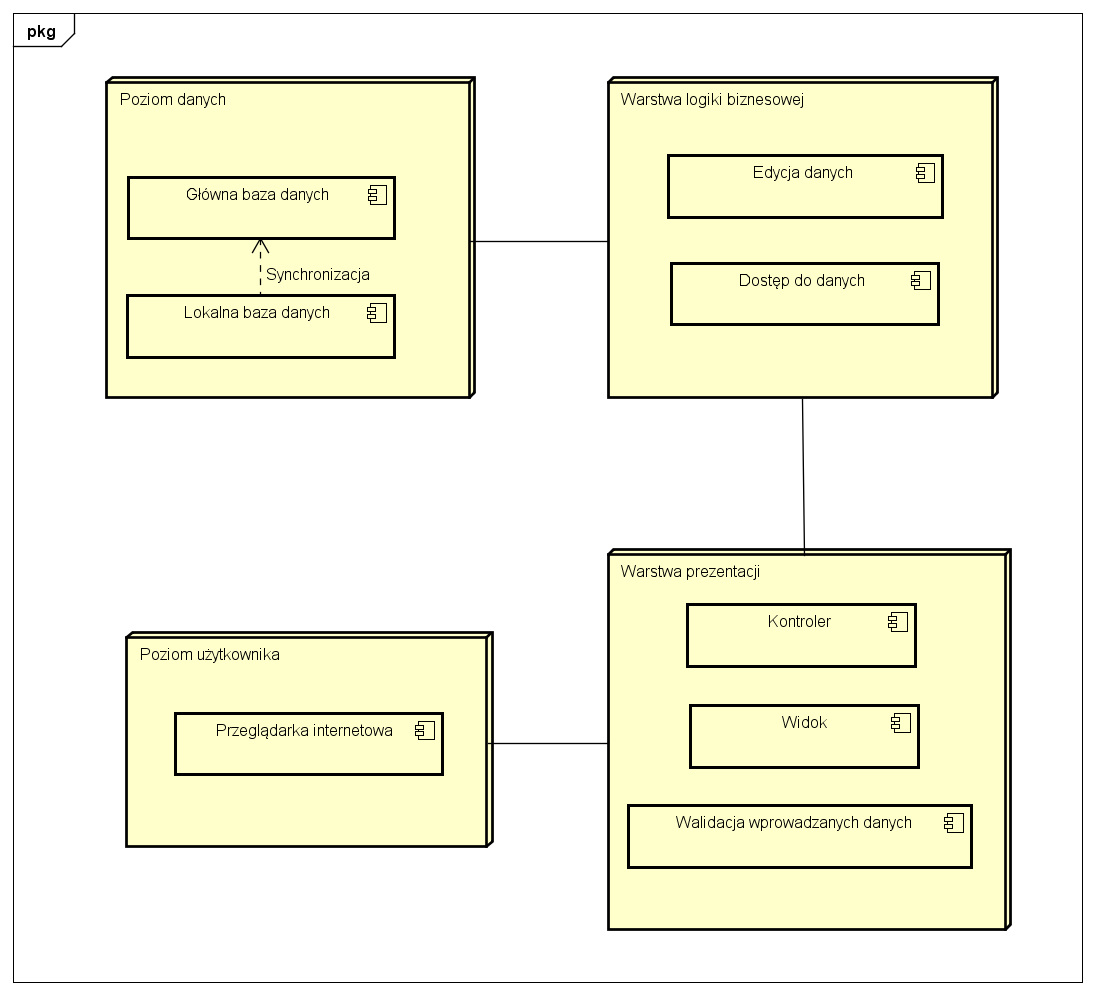


Rys. Diagram prezentujący warstwy aplikacji[[4]](#footnote-4)2

Aplikacja trójwarstwowa składa się z następujących warstw:

* warstwy prezentacji - jest to najwyższa z warstw w tym modelu i to z nią prowadzi interakcję użytkownik aplikacji. Warstwa ta posiada najczęściej walidację wprowadzanych przez użytkownika danych oraz referencję do warstwy biznesowej. Nie wie natomiast nic na temat zasad działania systemu, wykorzystywanej bazy danych itp.
* warstwa logiki biznesowej - warstwa ta przetwarza zapytania użytkownika. Można sobie ją wyobrazić jako "mózg" całej aplikacji. Warstwa ta jest "pośrednikiem" między warstwą prezentacji a warstwą dostępu do danych. Warstwa ta nie wie w jaki sposób "wydobywać" informacje z bazy danych.
* warstwa danych - jest to jedyna warstwa aplikacji, która wie w jaki sposób pobierać, zapisywać i uaktualniać informacje w bazie danych. Warstwa ta nie wie kto pobiera informacje lub komu je przekazuje.

### Diagram UML wdrożenia[[5]](#footnote-5)



# Opis metodyki pracy

## Odpowiedzi na 10 pytań

### Dlaczego tu jesteśmy?

* aby zrobić pierwszą w życiu aplikację internetową
* aby nauczyć się czegoś - ASP.NET, C#
* aby nauczyć się w miarę szybko wdrażać w nowe technologie

### Krótkie podsumowanie

Visual Studio znacząco ułatwia pracę nad aplikacjami internetowymi. Posiada wiele wbudowanych funkcji, z których wystarczy jedynie skorzystać.

### Opakowanie produktu

* dostarczone oprogramowanie ma być zachęcające pod względem wizualnym
* interfejs powinien być przyjazny użytkownikowi i kolorystyką nawiązywać do tematyki ślubnej

### Lista "NIE"

* nie kopiować kodu
* nie poddawać się

### Otoczenie projektu

* instalacja oprogramowania
* oglądanie tutoriali
* optymalizacja wybieranych rozwiązań
* czytanie literatury
* próba realizacji projektu na wiele sposobów

### Szkic rozwiązania

* zrobić widoki
* połączyć widoki z bazą danych
* dodać funkcjonalności typu: dodaj, usuń, modyfikuj
* umożliwić przesyłanie danych między serwerami

### Nocne koszmary

* problem z zainstalowaniem oprogramowania
* konieczność zmiany technologii w połowie projektu
* niezdążenie przed czasem

### Oszacowania

* trzeba poświęcić na ten projekt dużo czasu ze względu na nieznajomość ASP.NET

### Czego dostarczymy?

* działającej aplikacji
* czegoś, co może się komuś z nas potem w życiu przyda ;)

### Czego potrzebujemy?

* nauczyć się robić backend do aplikacji internetowych
* zaliczyć przedmiot
* zainstalować środowisko pracy

## Organizacja pracy

Do pracy zostanie wykorzystane Trello. Zespół będzie składał się z trzech osób: Weronika Hryniewska, Michał Lis, Sebastian Andrzejczak. Kod będzie w całości umieszczony i przechowywany w repozytorium Gita (<https://github.com/Hryniewska/PAI>). Procedura uruchamiania aplikacji zostanie opisana w pliku *README.md*. Główne historie użytkownika zostały określone w rejestrze produktu na Trello. Oszacowania przygotowano dzięki wtyczce *Scrum for Trello*. Ze względu na konieczność ciągłych zmian w podejściu do projektu zastosowano podejście zwinne.

1. <https://github.com/johny1614/paiDiag/blob/master/Dodawanie%20go%C5%9Bcia%20lub%20prezentu.png> [↑](#footnote-ref-1)
2. [www.aspsnippets.com/Articles/How-to-transfer-data-from-one-website-to-another-in-ASPNet.aspx](http://www.aspsnippets.com/Articles/How-to-transfer-data-from-one-website-to-another-in-ASPNet.aspx) [↑](#footnote-ref-2)
3. [patryknet.blogspot.com/2010/06/architektura-trojwarstwowa-w-aspnet.html](http://patryknet.blogspot.com/2010/06/architektura-trojwarstwowa-w-aspnet.html) [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)
5. <https://github.com/johny1614/paiDiag/blob/master/Diagram%20wdro%C5%BCenia.png> [↑](#footnote-ref-5)