|  |
| --- |
| PJATK |
| Dokumentacja techniczna do pracy inżynierskiej: „SuitGift” |
|  |

|  |
| --- |
| Piotr Jarzyński, Rafał Łagowski  1-31-2020 |

Spis treści

[Faza analizy 2](#_Toc41914834)

[Wstęp 2](#_Toc41914835)

[Cel 2](#_Toc41914836)

[Kontekst 2](#_Toc41914837)

[Zakres 2](#_Toc41914838)

[Wymagania funkcjonalne i niefunkcjonalne aplikacji „SuitGift” 2](#_Toc41914839)

[Wymagania funkcjonalne 2](#_Toc41914840)

[Wymagania niefunkcjonalne 3](#_Toc41914841)

[Aktorzy systemu 4](#_Toc41914842)

[Diagram przypadku użycia 4](#_Toc41914843)

[Diagram związków encji 5](#_Toc41914844)

[Faza projektowania i dokumentacja techniczna 6](#_Toc41914845)

[Wybór technologii 6](#_Toc41914846)

[Opis używanych technologii 6](#_Toc41914847)

[Architektura systemu 8](#_Toc41914848)

[Projekt interfejsu 8](#_Toc41914849)

[Diagram związków encji (projektowy) 8](#_Toc41914850)

[Szczegółowy opis encji i atrybutów 8](#_Toc41914851)

[Wymagania sprzętowe 10](#_Toc41914852)

[Diagramy biznesowe i przejścia między ekranami 10](#_Toc41914853)

# Faza analizy

## Wstęp

Często zdarza się nam dostawać nietrafione prezenty. Tak samo często o ile nie częściej nie wiemy co kupić swoim bliskim na różne uroczystości i święta. Również zdarza się nam zapomnieć o czyichś urodzinach bądź innych świętach. Na rynku nie ma obecnie żadnego rozwiązania tych problemów. My chcemy te problemy rozwiązać.

## Cel

Celem naszego projektu jest ułatwienie procesu wyboru trafionego prezentu dla naszych bliskich. Również chcemy ułatwić naszym znajomym kupno trafionego prezentu dla nas samych. Nasz system będzie ułatwiał również komunikację między znajomymi.

## Kontekst

Obecnie jeśli dostajemy zaproszenie od znajomego na daną okazję nie wiemy nic o reszcie osób również na nią zaproszonych ani o prezentach, które gospodarz pragnie otrzymać. Często kupujemy nietrafione lub zdublowane mniejsze prezenty, które nie znajdą zastosowania zamiast kupić jeden większy prezent wraz z innymi gośćmi zaproszonymi na daną imprezę. Komunikacja między uczestnikami wydarzenia jest niejasna, dochodzi do nieporozumień i niechcianych sytuacji.

## Zakres

W naszym projekcie planujemy stworzyć aplikację, która sprosta problemom opisanym w kontekście. Aplikacja będzie umożliwiać dodawanie swoich życzeń (prezentów) oraz zainteresowań do listy, która będzie widoczna dla naszych znajomych. Dodatkową funkcjonalnością aplikacji będzie możliwość stworzenia wydarzenia do którego użytkownik będzie mógł dodać swoich znajomych, aby ułatwić komunikację między nimi. Aplikacja ma zachować element niespodzianki dla obdarowywanego.

## Wymagania funkcjonalne i niefunkcjonalne aplikacji „SuitGift”

### Wymagania funkcjonalne

1. Aplikacja ma umożliwiać zalogowanie się do aplikacji za pomocą danych z platformy „Facebook” lub „Twitter”. (Nie trzeba rejestrować się do systemu).
2. Aplikacja ma umożliwiać rejestrację konta za pomocą podania imienia, nazwiska, adresu email, oraz hasła.
3. Aplikacja ma umożliwiać edytowanie swojego konta (adres email, hasło, imię, nazwisko)
4. Aplikacja ma umożliwiać dodawanie, modyfikacje, usuwanie zainteresowań do odpowiedniego pola (dodawanie)
5. Aplikacja ma umożliwiać dodawanie, modyfikacje, usuwanie prezentów, które użytkownik chciałby dostać do odpowiedniego pola (dodawanie)
6. Aplikacja ma umożliwiać użytkownikowi wyszukanie, dodawanie, usuwanie znajomego; Odpowiednia wyszukiwarka działająca na podstawie imienia i nazwiska.
7. Aplikacja ma umożliwiać użytkownikowi dodawanie Okazji do Kalendarza wraz z dodaniem do niej znajomych, których chciałby zaprosić na wydarzenie.
8. Aplikacja ma umożliwiać znajomym użytkownika dodanych do Okazji na dodawanie wpisów na forum, do którego użytkownik zapraszający nie ma dostępu, a jedynie osoby zaproszone na daną okazję.
9. Aplikacja ma umożliwiać znajomym użytkownika sprawdzenie jego listy prezentów.
10. Aplikacja ma umożliwiać użytkownikowi sprawdzenie listy znajomego.
11. Aplikacja ma umożliwiać znajomym użytkownika zaznaczenie chęci spełnienia danego życzenia w sposób tajny (zachowując element niespodzianki).
12. Aplikacja ma umożliwiać znajomym zaproszonym na okazję zaznaczenie chęci spełnienia danego życzenia grupowo w sposób tajny (wiele znajomych może zaznaczyć jeden prezent na który chcą się „zbić”).
13. Aplikacja ma umożliwiać użytkownikowi wyszukiwanie wydarzeń na które został zaproszony.
14. Aplikacja ma wysyłać użytkownikowi alerty o zbliżających się wydarzeniach.
15. Aplikacja ma umożliwiać użytkownikowi dodanie awatara lub zdjęcia do profilu.

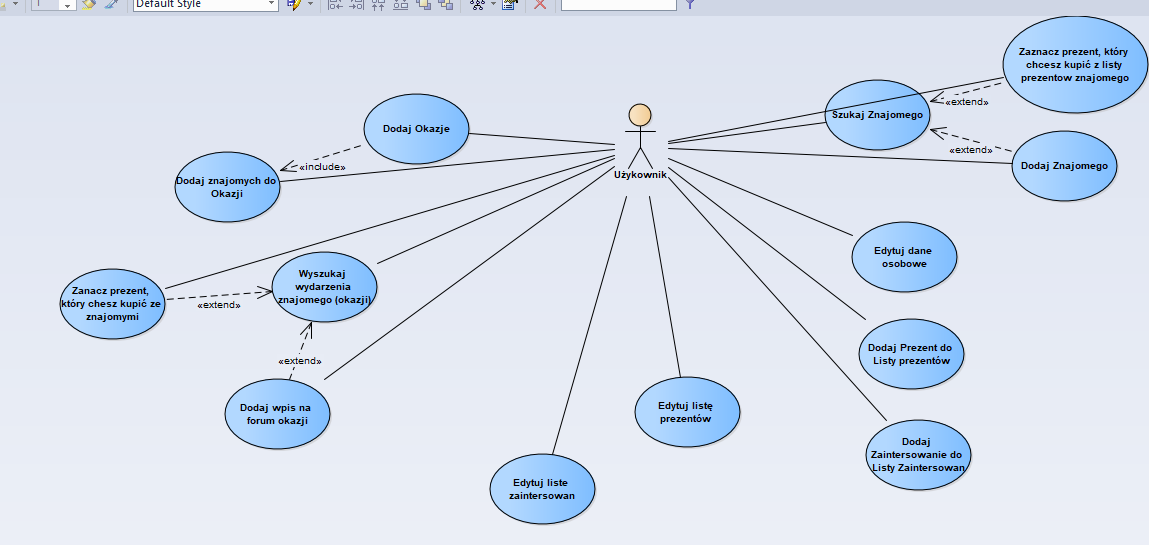
### Wymagania niefunkcjonalne

1. Aplikacja ma być obsługiwana przez obecne wersje najbardziej popularnych przeglądarek internetowych.
2. Aplikacja ma wykorzystywać protokół HTTPS.
3. Wyszukiwanie znajomych ma być realizowane szybciej niż 15 sekund.
4. Aplikacja ma mieć stały adres internetowy.
5. Rejestracja użytkownika ma przebiegać w prosty sposób, dla użytkownika niezaznajomionego z aplikacją w maksymalnie 5 minut.
6. Aplikacja ma umożliwiać ustawienie awatara o maksymalnej wielkości 10MB.
7. Interfejs użytkownika ma być przyjazny dla użytkownika, niezaznajomiony z nim użytkownik powinien wykonywać podstawowe funkcjonalności w maksymalnie 5 minut.

## Aktorzy systemu

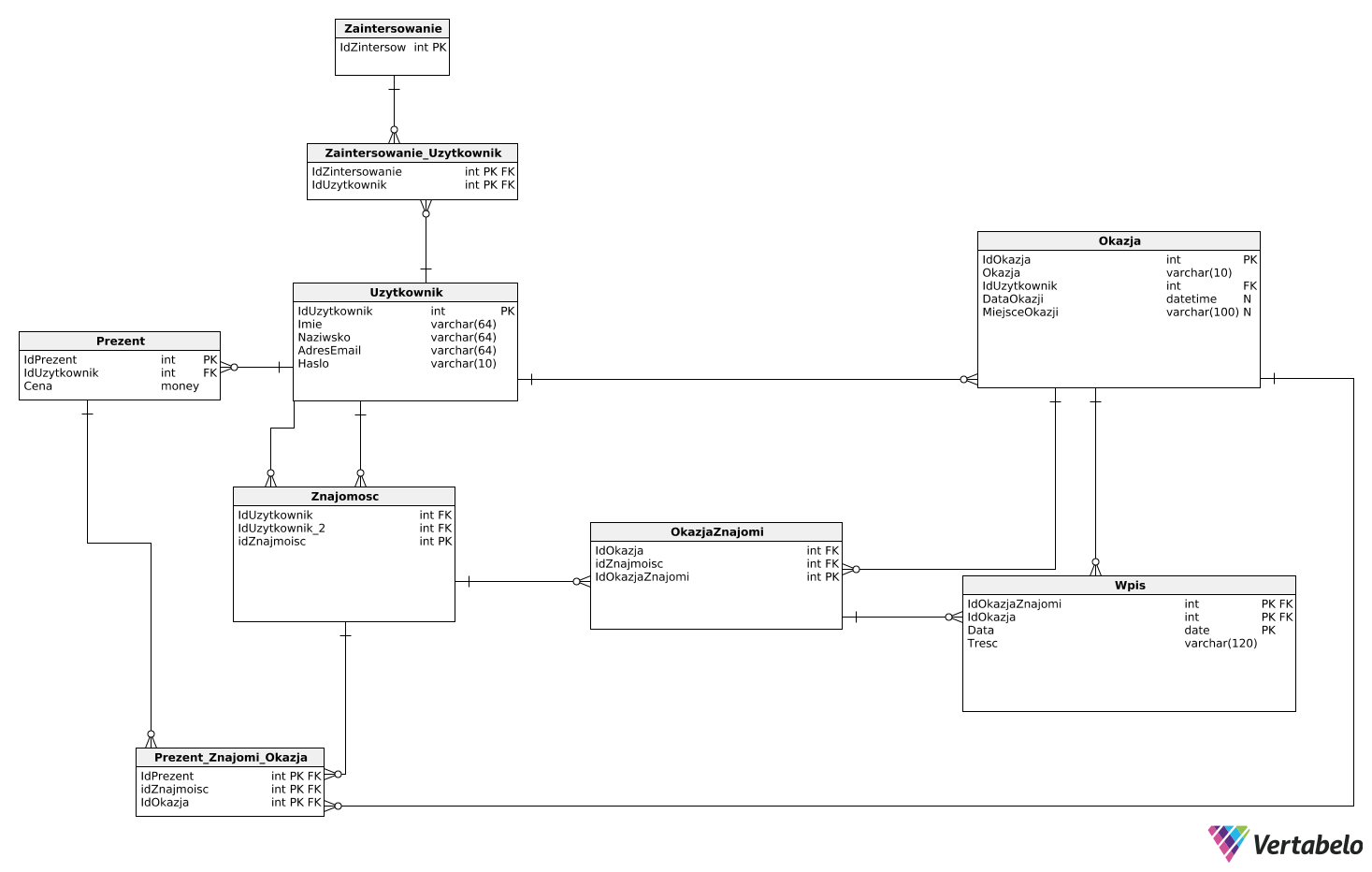
System będzie zawierał jednego aktora- Użytkownika. Użytkownik będzie miał możliwość stworzenia konta; zalogowania się na nie; dodawania znajomych; dodawania i edycji prezentów do swojej listy prezentów ;dodawania i edycji zainteresowań do listy zainteresowań; tworzenia okazji; dodawania wpisów do okazji znajomych oraz zaznaczania chęci spełnienia danego życzenia znajomego.

## Diagram przypadku użycia



Rysunek Diagram przypadków użycia

## Diagram związków encji



Rysunek Diagram związków encji

# Faza projektowania i dokumentacja techniczna

## Wybór technologii

1. Baza danych:  
   Microsoft SQL Server (2019)  
   Microsoft SQL Server Tools- SQL Server Management Studio v18  
   Baza zapewnia lepszą wydajność, większe bezpieczeństwo i łatwość instalacji. Dodatkowo tworzy kompatybilne środowisko z Microsoft Visual Studio. Management Studio to wygodny w użytkowaniu graficzny interfejs ułatwiający wykonywanie zapytań na bazie.
2. Języki:  
   SQL dialekt Microsoft  
   C#  
   HTML, CSS, JavaScript  
   Node.js  
   C# cały czas jest rozwijany przez Microsoft, jest w nim wiele przydatnych bibliotek, które są ulepszane. Pozostałe języki wybraliśmy ze względu na doświadczenie przy pracowaniu z nimi.
3. Backend:  
   Microsoft Visual Studio (2019)  
   Umożliwia bezpośrednie połączenie oraz wykonanie zapytań w jednym środowisku programistycznym. Dodatkowo posiada wygodne narzędzia do debugowania kodu. Umożliwia łatwe przeprowadzanie testów jednostkowych, integracyjnych, e2e.
4. Frameworks:  
   Microsoft ASP.NET Core  
   Entity Framework Core  
   Entity Framework SQL Server Core  
   Entity Framework Core Tools  
   Łatwość wykorzystania w projekcie oraz łatwa instalacja przy użyciu Visual Studio.
5. Frontend:  
   Microsoft Visual Studio Code (2019)  
   Łatwość instalacji potrzebnych bibliotek.
6. Biblioteki:  
   Angular  
   Biblioteki te są „opensource” i umożliwiają wykonywanie pożądanych funkcjonalności.
7. Dodatkowe narzędzia:  
   caweno.com – tworzenie diagramów biznesowych w notacji BPMN.  
   vertabelo.com – umożliwia tworzenie diagramów związków encji w graficznym interfejsie oraz generowanie kodu SQL.  
   MS Word  
   Git – system kontroli wersji.  
   GitHub.com – platforma umożliwiająca kontrolę nad projektem.

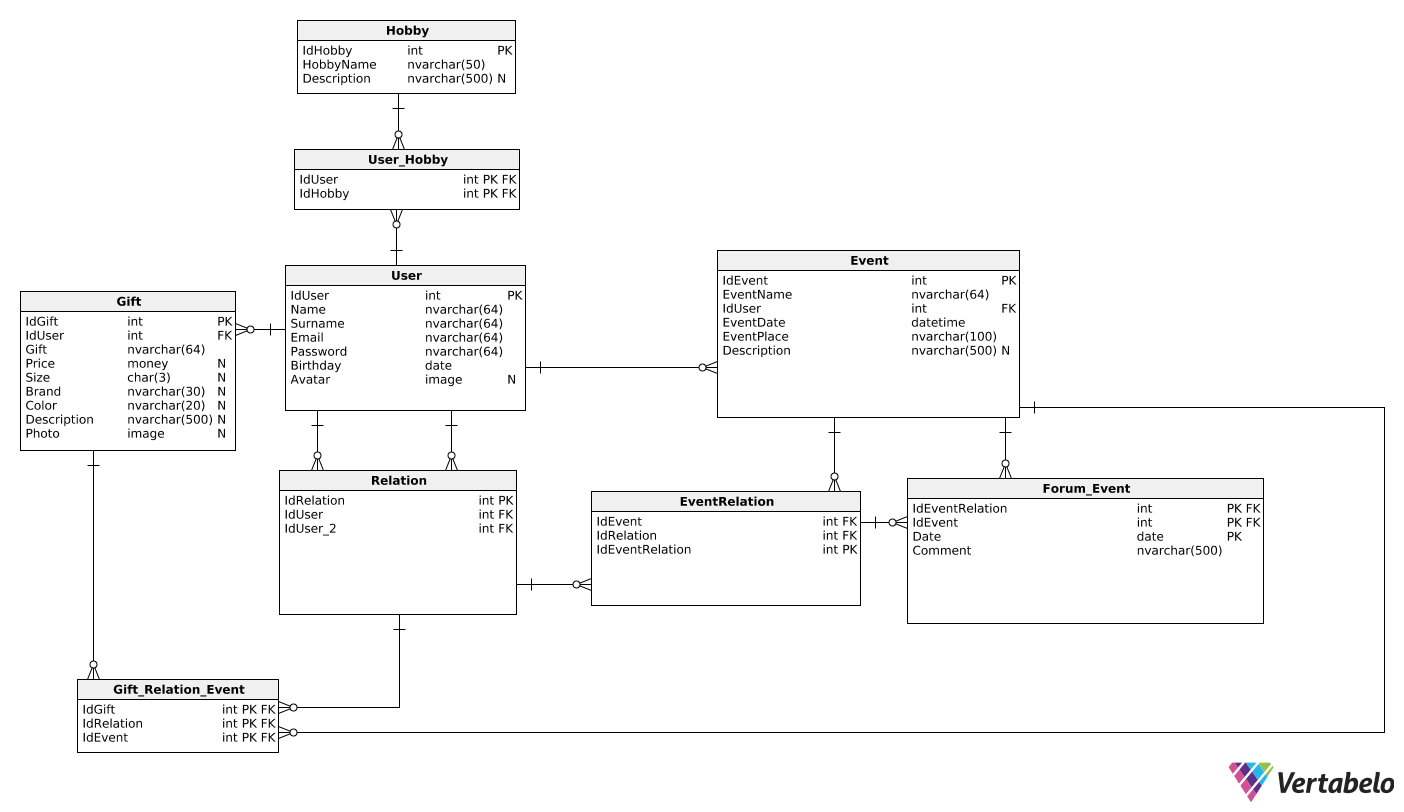
## Opis używanych technologii

1. Microsoft SQL Server (2019)  
   System zarządzania bazą danych, wspierany i rozpowszechniany przez firmę Microsoft. Jest to główny produkt bazodanowy tej korporacji, który cechuje się tym, iż jako język zapytań używany jest przede wszystkim Transact-SQL, który stanowi rozwinięcie standardu ANSI/ISO. MS SQL Server jest platformą bazodanową typu klient-serwer. Odznacza się lepszą wydajnością, niezawodnością i skalowalnością. Przede wszystkim są tu zaimplementowane wszelkie mechanizmy wpływające na bezpieczeństwo operacji (m.in. procedury wyzwalane).
2. SQL Server Management Studio v18  
   Zintegrowane środowisko do zarządzania wszystkimi komponentami (baza danych, usługi analityczne, usługi raportowe itd.), wchodzącymi w skład Microsoft SQL Server. Zawiera narzędzia do konfiguracji, monitorowania i administrowania instancjami SQL Server. Umożliwia budowę zapytań i skryptów, zawiera zarówno edytor skryptów jak i narzędzia graficzne. Główną cechą aplikacji jest Object Explorer, który pozwala na przeglądanie, wybieranie i wykonywanie różnych działań na obiektach serwera.
3. SQL  
   SQL jest to język zapytań (Structured Query Language), który zapewnia komunikację między użytkownikiem lub aplikacją, a bazą danych.
4. C#  
   Jest to wysokopoziomowy obiektowy język programowania ściśle zintegrowany z platformą .NET, która jest zarówno frameworkiem i środowiskiem uruchomieniowym. C# jest używany do tworzenia aplikacji webowych po stronie serwera za pomocą frameworka ASP.NET.
5. HTML, CSS, JavaScript  
   HTML jest to język znaczników wykorzystywany w tworzeniu aplikacji webowych.  
   CSS jest to język służący do opisu formy prezentacji (wyświetlania) stron WWW. Arkusz stylów CSS to lista dyrektyw (tzw. reguł) ustalających w jaki sposób ma zostać wyświetlana przez przeglądarkę internetową zawartość wybranego elementu (lub elementów) HTML.   
   JavaScript jest to skryptowy język programowania, stosowany na stronach internetowych. Dynamicznie typowany język wysokiego poziomu zapewniający interaktywność stron oraz obsługę zdarzeń, walidację formularzy, budowanie elementów nawigacyjnych.
6. Node.js  
   Jest to platforma umożliwiająca uruchomienie kodu JavaScript poza przeglądarką. Udostępnia API, którego najszerszym zastosowaniem jest tworzenie stron internetowych opartych na programowaniu sterowanymi zdarzeniami. Zawiera wiele modułów.
7. Microsoft Visual Studio (2019)  
   Zintegrowane środowisko programistyczne firmy Microsoft. Jest używane do tworzenia oprogramowania konsolowego wraz z graficznym interfejsem użytkownika.
8. Microsoft ASP.NET Core  
   Szablony projektów do pisania aplikacji internetowych i interfejsów API platformy ASP.NET Core dla systemów Windows, Linux i MacOS przy użyciu .NET Core lub .NET Framework.
9. Entity Framework Core  
   Lekka, rozszerzalna, otwarta i wieloplatformowa wersja popularnej technologii dostępu do danych. Służy jako maper obiektowo-relacyjny (ORM) umożliwiając developerom pracę z bazą danych ograniczając ilość kodu.
10. Entity Framework SQL Server Core/ Tools  
    Jest to część technologii Entity Framework Core odpowiedzialna za dostarczenie narzędzi potrzebnych do obsługi MS SQL Server.
11. Microsoft Visual Studio Code (2019)  
    Darmowy desktopowy edytor programistyczny kodów źródłowych z kolorowaniem składni dla wielu języków. Stworzony przez Microsoft na wolnej licencji.
12. Angular  
    Jest to JavaScriptowa biblioteka służąca do tworzenia interfejsów użytkownika, ułatwia tworzenie interaktywnych UI opartych na komponentach.

## Architektura systemu

## Projekt interfejsu

## Diagram związków encji (projektowy)



Rysunek Diagram związków encji (projektowy)

Diagram projektowy w stosunku do analitycznego różni się nazwami tabel, które zostały zmienione na angielskie oraz typem danych „varchar” zamienionym na „nvarchar”. Atrybuty niektórych tabel również uległy zmianie.

## Szczegółowy opis encji i atrybutów

### Encja „User”

Opisuje cechy użytkownika zarejestrowanego do aplikacji „SuitGift”. Zawiera atrybuty takie jak:   
IdUser – jest to klucz główny (PK), autogenerowany, służy do łatwiejszego wyszukiwania użytkowników dla administratora bazy danych; użytkownik nie ma dostępu do tego atrybutu; na tej kolumnie jest założony indeks pogrupowany.  
Name, Surname, Email, Password, Birthday – są to atrybuty podawane przez użytkownika podczas rejestracji; definiują użytkownika poprzez imię, nazwisko, email oraz datę urodzenia; widoczne na jego profilu przez niego i innych użytkowników; użytkownik może edytować swoje dane.  
Avatar – jest to atrybut opcjonalny; użytkownik może dodać awatar w formie zdjęcia (format image)

### Encja „Hobby”

Przetrzymuje informacje o zainteresowaniach. Zawiera atrybuty:  
IdHobby – klucz główny (PK); użytkownik nie ma dostępu do tego atrybutu; zawiera indeks pogrupowany; autogenerowany;  
HobbyName, Description – atrybuty opisujące dane zainteresowanie poprzez jego nazwę i opis (opcjonalny przy zainteresowaniach bardzo szczegółowych).

### Encja „User\_Hobby”

Encja asocjacyjna, która umożliwia związek wiele do wiele między tabelą „User” i „Hobby”. Przetrzymuje informacje o hobby, które ma dany użytkownik. Składa się z kluczy obcych z tabel „User” i „Hobby” i są to jej klucze główne (PK). Zapewnia spójność więzów referencyjnych.

### Encja „Gift”

Przetrzymuje informacje o prezentach, które pragnie otrzymać użytkownik poprzez klucz obcy (IdUser) wraz z zapewnieniem więzów spójności referencyjnej. Zawiera atrybuty:  
IdGift- klucz główny (PK); autogenerowany; użytkownik nie ma do niego dostępu.  
Gift – atrybut wymagany, opisuje nazwę prezentu; podawany przez użytkownika.  
Price, Size, Description, Photo, Color, Brand – atrybuty opcjonalne, które użytkownik może podać w celu doprecyzowania prezentu, który chce otrzymać.

### Encja „Event”

Przechowuje informacje o wydarzeniach lub okazjach stworzonych przez użytkownika. Zawiera atrybuty:  
IdEvent – klucz główny (PK); autogenerowany; użytkownik nie ma do niego dostępu.  
IdUser – klucz obcy (FK); zawiera więzy spójności referencyjnej;   
EventName, EventDate, EventPlace, Description – atrybuty opisujące wydarzenie: nazwa, miejsce, czas, opis (opcjonalnie).

### Encja „Relation”

Przechowuje informacje o znajomych użytkownika. Encja asocjacyjna łącząca dwóch użytkowników w relację. Kluczem głównym relacji jest IdRelation, a nie suma kluczy obcych. Składa się z atrybutów:  
IdRelation – klucz główny (PK); autogenerowany; użytkownik nie ma do niego dostępu  
IdUser, IdUser\_2 – klucze obce (FK) z tabeli User; zawierają więzy spójności referencyjnej.

### Encja „EventRelation”

Przechowuje informacje na temat użytkowników (znajomych) zaproszonych na dane wydarzenie przez użytkownika. Encja asocjacyjna łącząca tabele Event i Relation. Kluczem głównym jest IdEventRelation, a nie suma kluczy obcych. Składa się z atrybutów:  
IdEventRelation – klucz główny (PK); autogenerowany; użytkownik nie ma do niego dostępu  
IdEvent, IdRelation – klucze obce (FK) z tabel Event i Relation; zawierają więzy spójności referencyjnej

### Encja „Forum\_Event”

Przechowuje informacje o postach wstawianych przez znajomych na forum wydarzenia. Użytkownik tworzący wydarzenie nie ma dostępu do forum (zachowanie elementu niespodzianki). Encja asocjacyjna łącząca tabele Event i EventRelation. Zawiera atrybuty:  
Date – klucz główny (PK).  
IdEvent, IdEventRelation – klucze obce (FK) oraz klucze główne (PK)  
Comment – zapisuje informacje o treści komentarza dodanego przez użytkownika.

### Encja „Gift\_Relation\_Event”

Encja asocjacyjna łącząca tabele Gift, Event i Relation. Zawiera atrybuty:  
IdGift, IdEvent, IdRelation – są to klucze obce (FK) i główne (PK).

## Wymagania sprzętowe

## Diagramy biznesowe i przejścia między ekranami