

Dokumentacja projektu

Spis treści

[1. Opis ogólny 3](#_Toc308516803)

[1.1. Poruszany problem 3](#_Toc308516804)

[1.2. Użytkownik docelowy 3](#_Toc308516805)

[1.3. Jak korzystać z programu? 3](#_Toc308516806)

[1.4. Możliwości programu 4](#_Toc308516807)

[2. Opis funkcjonalności 5](#_Toc308516808)

[2.1. Zarządzanie bazą produktów 5](#_Toc308516809)

[2.2. Zarządzanie bazą kontrahentów 5](#_Toc308516810)

[2.3. Przyjmowania faktur zakupowych, przeniesień magazynowych, produktów własnych 5](#_Toc308516811)

[2.4. Fakturowanie 5](#_Toc308516812)

[2.5. Drukowanie 6](#_Toc308516813)

[2.6. Zarządzanie kontami użytkowników 6](#_Toc308516814)

[2.7. Zarządzanie rozrachunkami 6](#_Toc308516815)

[2.8. Przypominanie o płatnościach 6](#_Toc308516816)

[2.9. Generowanie statystyk 6](#_Toc308516817)

[3. Role użytkowników w systemie 7](#_Toc308516818)

[4. Szczegółowy scenariusz działania programu 8](#_Toc308516819)

[4.1. Logowanie użytkownika 8](#_Toc308516820)

[4.2. Zarządzanie bazą kontrahentów 8](#_Toc308516821)

[4.3. Zarządzanie produktami 9](#_Toc308516822)

[4.4. Zarządzanie fakturami 10](#_Toc308516823)

[4.5. Zarządzanie rozrachunkami 11](#_Toc308516824)

[4.6. Zarządzanie stanem magazynowym 11](#_Toc308516825)

[4.7. Moduł drukowania 12](#_Toc308516826)

[5. Diagramy UML 13](#_Toc308516827)

[5.1. Diagram przypadków użycia 13](#_Toc308516828)

[5.2. Diagram encji 15](#_Toc308516829)

[5.2.1. Model uproszczony 15](#_Toc308516830)

[5.2.2. Model szczegółowy 16](#_Toc308516831)

[5.3. Diagram klas 17](#_Toc308516832)

[6. Przegląd wybranych narzędzi i technologii wykorzystanych przy realizacji i implementacji systemu magazynowania i fakturowania 21](#_Toc308516833)

[6.1. Narzędzie Visual Studio 2010 21](#_Toc308516834)

[6.2. Technologia ASP.NET 22](#_Toc308516835)

[6.3. Model obiektowy ADO.NET 22](#_Toc308516836)

[6.4. Technologia LINQ 23](#_Toc308516837)

[6.5. Wzorzec projektowy MVC w aplikacji ASP.NET 25](#_Toc308516838)

[7. Autorzy i podział prac 26](#_Toc308516839)

# Opis ogólny

## Poruszany problem

Założeniem projektu jest zaimplementowanie systemu informatycznego pozwalającego na:

* Księgowanie
* Zarządzanie stanem magazynowym,
* Fakturowanie
* Generowania statystyk sprzedaży
* Zarządzania płatnościami wobec kontrahentów

Aplikacja trafia w jakże wąską lukę rynkową, kompleksowych narzędzi księgowych opartych o architekturę webową.  
„Magazynier GT” jest niewątpliwie idealnym narzędziem skierowanym do szerokiego grona odbiorców.

## Użytkownik docelowy

System w założeniu jest aplikacją webową umożliwiającą zakładanie podmiotom gospodarczym indywidualnych kont identyfikujących daną firmę. Program jest uniwersalnym systemem informatycznym przeznaczonym zarówno dla małych przedsiębiorstw jak i dużych korporacji. Co więcej z systemu mogą korzystać klienci indywidualni rozpoczynający swą przygodę z rachunkowością.

## Jak korzystać z programu?

Ze względu na specyfikę zastosowanych technologii, obsługa systemu możliwa jest przy użyciu dowolnej przeglądarki spełniającej standardy W3C oraz posiadającej obsługę JavaScript. Dostęp do indywidualnego konta jest uwierzytelniany. Aplikacja odznacza się przenośnością ze względu na osadzenie jej w architekturze webowej. Dostęp do niej nie jest uzależniony od systemu operacyjnego, miejsca, architektury komputerowej. Jest to niewątpliwą zaletą wyróżniającą „Magazynier GT” na tle konkurencji.

## Możliwości programu

Podstawową zakładaną funkcjonalnością systemu jest zarządzanie bazą produktów, a na ich podstawie stanem magazynowym. Klient będzie miał możliwość przyjmowanie faktur zakupowych, przeniesień magazynowych, produktów własnych. Zdejmowanie towarów/usług ze stanu magazynowego będzie skutkiem fakturowania. Aplikacja umożliwia drukowanie dokumentów. Wszystkie wprowadzone do aplikacji dokumenty są składowane w bazie danych, dzięki czemu użytkownik ma podgląd do historii transakcji. Daje to możliwość analizy obrotów firmy oraz historii zakupów kontrahenta. Analizę ułatwia mechanizm sortowania danych po kolumnach tabel. Zarządzanie kontami użytkowników. W celu uniknięcia błędów użytkownika w aplikacji zaimplementowano zaawansowany system walidacji danych, który pozwoli zapobiec wprowadzeniu niepoprawnych danych. Aplikacja posiada intuicyjny system wyszukiwania, który w  łatwy sposób pozwala na przeglądanie danych znajdujących się w systemie.

# Opis funkcjonalności

Poniższy rozdział jest wyszczególnieniem funkcji zawartym w systemie wraz z ogólnym opisem ich przeznaczenia.

## Zarządzanie bazą towarów

Użytkownik ma możliwość określenia rodzaju asortymentu oraz wszystkich informacji niezbędnych do obrotu. W skład tych informacji wchodzą: nazwa, cena netto, ilość produktów, marża, stawka VAT, jednostka, producent.

## Zarządzanie bazą kontrahentów

Aplikacja pozwala na listowanie wszystkich przetrzymywanych w bazie danych kontrahentów. Ponadto można dodawać nowych, edytować istniejących oraz usuwać nieaktualne wpisy. Każda pozycja opisana jest poprzez pola: nazwa, opis, regon, nip, adres, nr konta, strona WWW, adres email, telefon kontaktowy.

## Zarządzanie bazą faktur

Pracownik może dodawać nowe faktury, edytować stare, usuwać. Podczas dodawania można dodać wiele informacji, dzięki czemu każda faktura będzie w należyty sposób zaksięgowana. Aplikacja pozawala na zarządzanie zarówno fakturami zakupowymi jak i sprzedażowymi.

## Tworzenie przeniesień magazynowych

Twórcy systemu przewidzieli potrzebę wycofań magazynowych na wypadek gdy jakiś produkt na magazynie, zostanie zniszczony, minie jego data ważności, bądź zostanie ukradziony.

## Fakturowanie

Funkcjonalność fakturowania pozwala klientowi na odejmowanie ze stanu magazynowego towarów jak również księgowanie świadczonych usług. Usunięcie bądź edycja wystawionego dokumentu jest jednoznaczne z aktualizacja stanu magazynowego.

## Drukowanie

Bogaty zbiór szablonów dokumentów jest niezwykle ważnym aspektem ułatwiającym klientowi kontakt z organami skarbowymi. Wśród przykładowych dokumentów istnieją wzory faktur i remanentów.

## Zarządzanie kontami użytkowników

Konta użytkowników mogą mieć dwa poziomy uprzywilejowania: użytkownik i administrator. Pierwszy poziom pozwala na pełne wykorzystanie wyżej wymienionych funkcjonalności. Drugi poziom ponadto pozwala na zarządzanie kontami użytkowników, w tym dodawanie oraz zmianę i resetowanie hasła.

## Zarządzanie płatnościami

Moduł umożliwia w bardzo szybki sposób zaksięgować zaległą płatność, można przeglądać ostatnie transakcje. W panelu okna znajdują się odnośniki do najbardziej popularnych banków, co w znaczący sposób pozwoli użytkownikowi zmniejszyć czas potrzebny na dostęp do konta internetowego.

## Przypominanie o płatnościach

Program automatycznie przedstawia aktualną listę wierzycieli, których ostateczny termin płatności minął, dzięki temu użytkownicy systemu wiedzą, kto zalega z płatnościami.

## Generowanie statystyk

Aplikacja posiada rozbudowany system do tworzenia statystyk na bazie informacji zawartych w bazie danych. Wyniki w czytelny sposób, są przedstawione za pomocą różnego rodzaju diagramów. Dzięki temu, każdemu, nawet mniej doświadczony użytkownik będzie miał możliwość bieżącej analizy stanu magazynowego oraz dynamiki sprzedaży.

# Przegląd wybranych narzędzi i technologii wykorzystanych przy realizacji i implementacji systemu magazynowania i fakturowania

Narzędziem użytym w celu implementacji systemu było Visual Studio 2010. Aplikacja została napisana przy użyciu technologii ASP.NET w oparciu o wzorzec projektowy MVC. Wybór tej technologii jest podyktowany olbrzymimi możliwościami jakie oferuje, obszerną dokumentacje oraz możliwość wykorzystania dostępnych technologii z platformy .NET. Wśród nich najważniejszymi są ADO.NET, LINQ oraz Entity Framework. Wykorzystano również system bazodanowy MSSQL. Połączenie wyżej wymienionych stwarza olbrzymie możliwości efektywnej i elastycznej implementacji systemu.

## Narzędzie Visual Studio 2010

Microsoft Visual Studio 2010 jest zaawansowanym, rozbudowanym środowiskiem programistycznym. W skład jego wchodzi wiele narzędzi umożliwiających między innymi łatwą i skuteczne projektowanie, refaktoryzacje i debugowanie kodu. Implementacje przy użyciu tego narzędzia skutecznie ułatwia mechanizm InteliSence, podpowiadający i sprawdzający składnie użytego języka programowania w czasie rzeczywistym. Używając go można tworzyć zarówno programy konsolowe, okienkowe, dynamiczne strony internetowe oraz gry. Możliwości te daje połączenie z tym narzędziem platformy .NET Framework. Sztandarową jej cechą jest kompilacja napisanego kodu do wspólnego środowiska uruchomieniowego CLR (Common Language Runtime). Głównymi zaletami takiego rozwiązania jest wysokie bezpieczeństwo, przenośność oraz elastyczność. Elastyczność ta objawia się tym, iż za pomocą narzędzia Microsoft Visual Studio 2010 możemy programować przy użyciu bardzo wielu języków w skład, których wchodzą między innymi C/C++, VB.NET, C#, F#. Wspierane i obsługiwane są również języki XML/XSLT, HTML/XHTML, JavaScript i CSS. Wszystko to sprawia, iż jest to jedno z najczęściej wykorzystywanych IDE (Integrated Development Environment) na świecie.

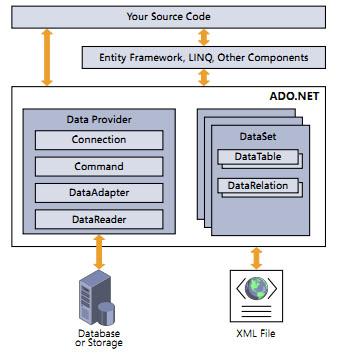
## Technologia ASP.NET

Platforma ASP.NET 3.5 jest frameworkiem aplikacji internetowych, służącym tworzeniu dynamicznych stron internetowych. Platforma ASP.NET 3.5 to obecnie bez wątpienia najszybsza, najefektywniejsza, najbardziej stabilna i najlepiej wspierana technologia służąca do tworzenia interaktywnych aplikacji. Pierwsza jej wersja powstała po wydaniu .NET Framework 1.0 w oparciu o ASP (Active Server Pages). Pozwalała ona na wykonywanie przez serwer sieciowy taki jak IIS, kodu aplikacji wewnątrz kodu HTML. Obecnie poprzez wykorzystanie form WWW, można skutecznie oddzielić warstwę prezentacji od warstwy logiki aplikacji, co znacznie wpływa na przejrzystość i możliwości refaktoryzacji. „Aplikacja sieciowa wykorzystująca ASP.NET składa się zazwyczaj z jednej lub kilku stron WWW, plików z kodem i plików konfiguracyjnych. Forma WWW jest przechowywana w pliku o nazwie z rozszerzeniem .aspx. Jest to zasadniczo plik HTML kilkoma tagami charakterystycznymi dla Microsoft .NET, definiującym układ i wygląd strony. „Formy WWW w ASP.NET są kompilowane i przechowywane na serwerze sieciowym, co ma korzystny wpływ na wydajność. Korzystanie z wielu języków dostępnych w platformie .NET jest możliwe dzięki kompilacji aplikacji do kodu pośredniego CLR. Każda kolejna wersja tej technologii jest wzbogaceniem jej o kolejne klasy i moduły zwiększające możliwości programisty przy jednoczesnym ograniczaniu potrzebnego nakładu czasowego. Tworzenie aplikacji może odbywać się przy użyciu zarówno wersji komercyjnych jak i bezpłatnych. ASP.NET jest silnie skorelowany z wzorcem projektowym MVC.

## Model obiektowy ADO.NET

Głównym celem dla ADO.NET jest zapewnienie pomostu między obiektami w ASP.NET a bazą danych. ADO.NET dostarcza zorientowanego obiektowo wglądu do bazy danych oraz jej właściwości. Dodatkowym, a w wielu przypadkach znacznie ważniejszym faktem jest, że obiekty ADO.NET hermetyzują i ukrywają szczegółowe informacje dotyczące dostępu do bazy danych. Jest to szczególnie pomocne korzystając z wielu systemów bazodanowych, których API nieczęsto się pokrywają. Aby więc obiekty aplikacji mogły współdziałać z obiektami ADO.NET, programista nie musi poznawać lub niepokoić się o szczegóły dotyczące przenoszenia danych z oraz do bazy danych.

.



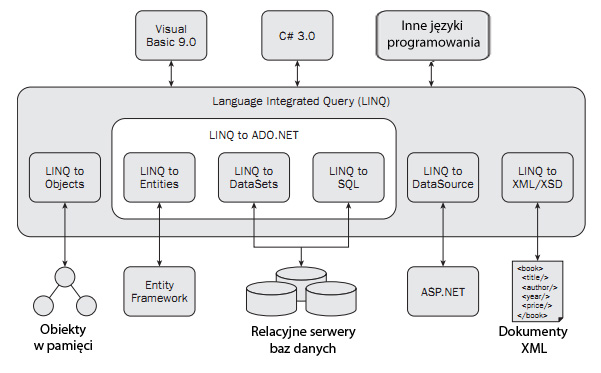
Rys. 3. Najważniejsze elementy ADO.NET

Na rysunku **Błąd! Nie można odnaleźć źródła odwołania.** przedstawiono kluczowe elementy technologii ADO.NET. Sercem całej biblioteki jest obiekt DataSet, który przechowuje obiekty DataTable będącymi odpowiednikami tabel relacyjnych systemów bazodanowych. Sposób operowania na danych jest bardzo elastyczny i różnorodny gdyż można odwoływać się zarówno za pomocą standardowych zapytań Sql, zunifikowanego języka LINQ, jak również za pomocą różnorodnych frameworków w tym Entity Framework. Obiekty DataTable można wypełniać danymi pochodzącymi z lokalnych bądź zewnętrznych systemów bazodanowych, a także plików XML.

## Technologia LINQ

W niniejszym rozdziale opisano podstawowe informacje na temat technologii LINQ, schemat jej działania oraz główne składowe.

Technologia LINQ zrewolucjonizowała metodologie programowania oraz zmieniła sposób komunikacji między programem a danymi. LINQ jest rozszerzeniem języków wchodzących w skład platformy .NET, które umożliwia odwoływanie się do dowolnych źródeł danych za pomocą zunifikowanej składni, podobnej do składni języka SQL. Bardzo rzadko zdarza się, że wszystkie wymagane dane znajdują się w jednym źródle. LINQ umożliwia łączenie tych danych i wykonywanie na nich standardowych operacji. Ogranicza to konieczność nauki nowych języków lub dialektów.

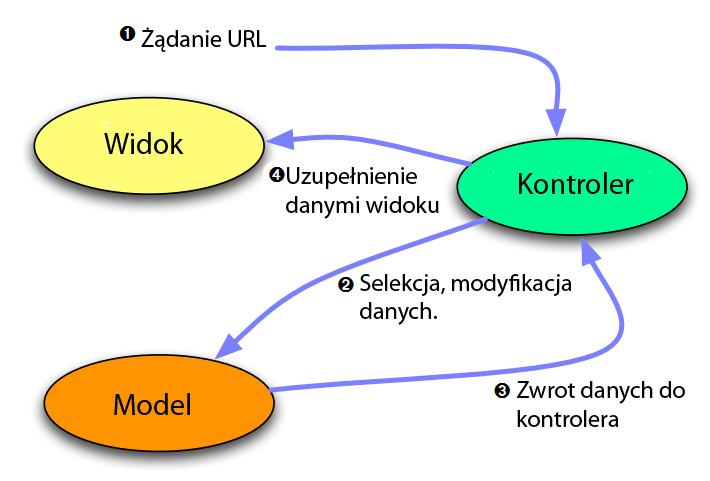


Rys. 1. Składowe technologii LINQ

Na rysunku 1 przedstawiono elementy składowe technologii LINQ. Model obiektowy ADO.NET znacznie ułatwia i ujednolica dostęp do różnorodnych systemów bazodanowych, natomiast technologia LINQ unifikuje sposób wyłuskiwania interesujących programistę danych. Ta sama składnia daje możliwość odwoływania się do obiektów w pamięci, zasobów strony ASP.NET oraz dokumentów XML. Ponadto istnieje pewna liczba zewnętrznych niezależnych dostawców LINQ, pozwalających na wykonywanie zapytań do wielu różnych źródeł danych, takich jak ORACLE, MySQL, Flickr, usługi sieciowe Amazon, NHibernate, usług SharePoint i innych.

## Wzorzec projektowy MVC w aplikacji ASP.NET

Wdrożenie wzorca MVC dla aplikacji ASP.NET sprawiło, iż tworzenie i testowanie aplikacji internetowych stało się znacznie łatwiejsze. W klasycznym spojrzeniu ASP.NET Web Forms starały się jakoby zamaskować fakt, iż protokół HTTP jest bezstanowy. Działo się to poprzez użycie kontrolek sieciowych i code – behind. Prowadziło to, do tego, iż informacja o stanie była przechowywana w czymś znanym, jako view – state. ASP.NET MVC rezygnuje z view – state na rzecz bardziej bezstanowej architektury.



Rys. 2. Przepływ sterowania modelu ASP.NET MVC

MVC kojarzy się głównie z wzorcem projektowania oprogramowania. Zakłada on wydzielenie niezależnych od siebie warstw Model – View – Controler odpowiadających kolejno za logikę aplikacji, wyświetlanie interfejsu użytkownika i kontrolę przepływu sterowania. ASP.NET MVC jest odrobinę odmienne od właściwego wzorca. Każde żądanie URL na podstawie parametrów jest tu przekazywane do konkretnego kontrolera. Kontroler wprowadza zmiany do modelu, po czym wybiera odpowiedni widok do wyświetlenia.

# Role użytkowników w systemie

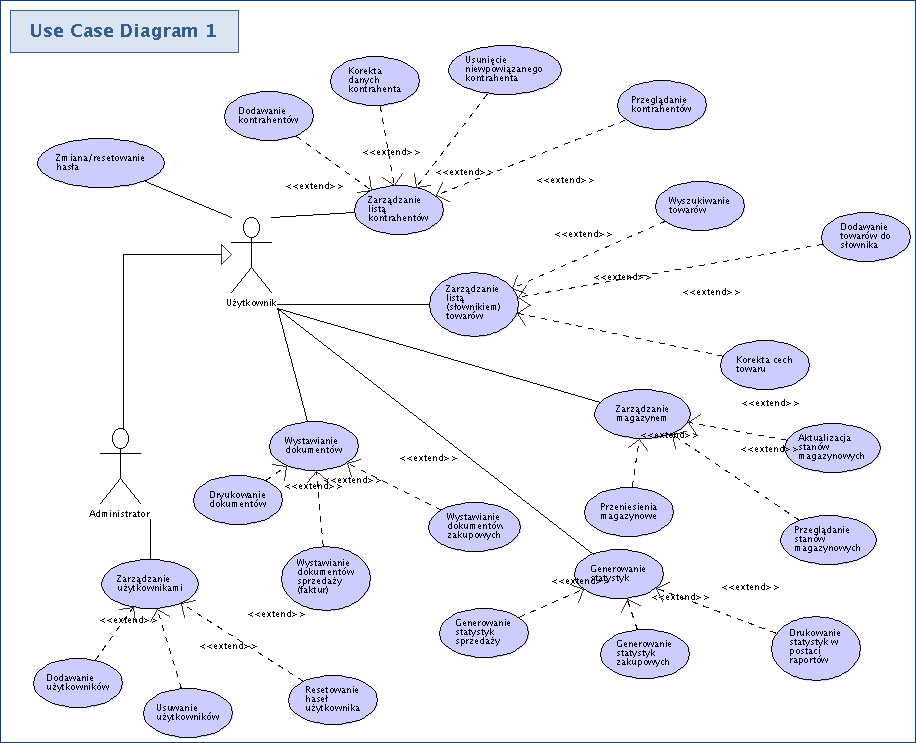
Aplikacja jest dostosowana do obsługi przez wiele podmiotów gospodarczych w ramach indywidualnych kont w systemie obsługi fakturowania „Magazynier GT”. W ramach działania aplikacji użytkownicy posiadają szereg ról w zależności od przywilejów jakie otrzymali. Rolami jakie użytkownik może otrzymać są:

* Administrator
  + Osoba o tym profilu dba ciągłość i stabilność działania aplikacji. Użytkownik o roli „Administratora” ma najwyższe uprawnienia w aplikacji pozwalające mu na dodawanie, poszczególnych kont użytkowników w systemie. Administrator jest użytkownikiem, który może zmieniać i resetować hasła w systemie, ma uprawnienia do modyfikowaniu danych o własnej firmie, w której pracują użytkownicy systemu. Administrator posiada wszystkie uprawnienia zwykłego pracownika.
* Użytkownik/Pracownik
  + „Pracownik” posiada możliwość wyłącznie zarządzania stanem magazynowym w ramach przydzielonego mu podmiotu gospodarczego. Rola będzie najczęściej wykorzystywanym profilem użytkownika w systemie. Pracownik może zarządzać towarami, fakturami, klientami, jednak nie ma możliwości edycji danych o własnej firmie i nie może dodawać nowych użytkowników do systemu.

# Diagramy UML

W niniejszym rozdziale przedstawiono różne rodzaje diagramów, które pozwolą na łatwe i klarowne przedstawienia zadań i możliwości serwisu.

## Diagram przypadków użycia



Rys. 1. Diagram przypadków użycia

Na rysunku 1 przedstawiono diagram przypadków użycia. Diagram ten nie uwzględnia roli “Firma”. Powodem tego jest dbałość o czytelność diagramu. Użytkownik o roli „Firma” rozszerza przypadki użycia pracownika o dodawanie, usuwanie, edycje pracowników, w ramach swojej działalności gospodarczej oraz może edytować

W przedstawionym powyżej diagramie przypadków użycia posiadamy aktora o nazwie „Użytkownik” który może:

Zmiana/resetowanie hasła – może zmienić swoje hasło dostępowe lub je zresetować jeżeli go zapomniał.

Zarządzanie listą kontrahentów, w skład którego wchodzą czynności:

Dodawanie kontrahentów.

Korekta danych kontrahenta.

Usunięcie niepowiązanego kontrahenta.

Przeglądanie kontrahentów.

Zarządzanie listą (słownikiem) towarów, w skład którego wchodzą czynności:

Wyszukiwanie towarów.

Dodawanie towarów do słownika.

Korekta cech towarów.

Zarządzanie magazynem, w skład którego wchodzą czynności:

` Aktualizacja stanów magazynowych.

Przeglądanie stanów magazynowych.

Przeniesienia magazynowe.

Generowanie statystyk,w skład którego wchodzą czynności:

Drukowanie statystyk w postaci raportów.

Generowanie statystyk zakupowych.

Generowanie statystyk sprzedaży.

Wystawianie dokumentów, w skład którego wchodzą:

Wystawianie dokumentów zakupowych.

Wystawianie dokumentów sprzedaży (faktur).

Drukowanie dokumentów.

Ponadto posiadamy aktora o nazwie „Administrator ” który może wykonywać te same czynności co „Użytkownik” i dodatkowo może:

Zarządzanie użytkownikami, w skład którego wchodzą czynności:

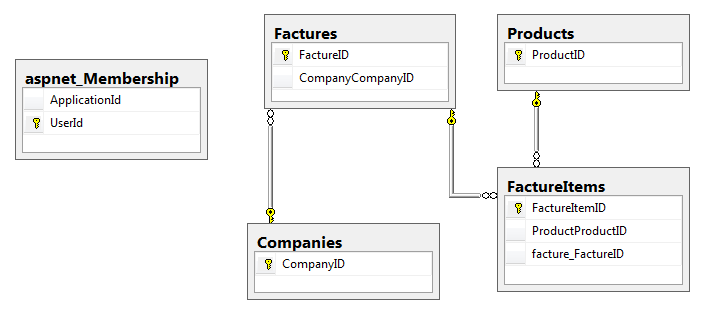
Resetowanie hasła użytkowników.

Usuwanie użytkowników.

Dodawanie użytkowników.

## Diagram encji

### Model uproszczony

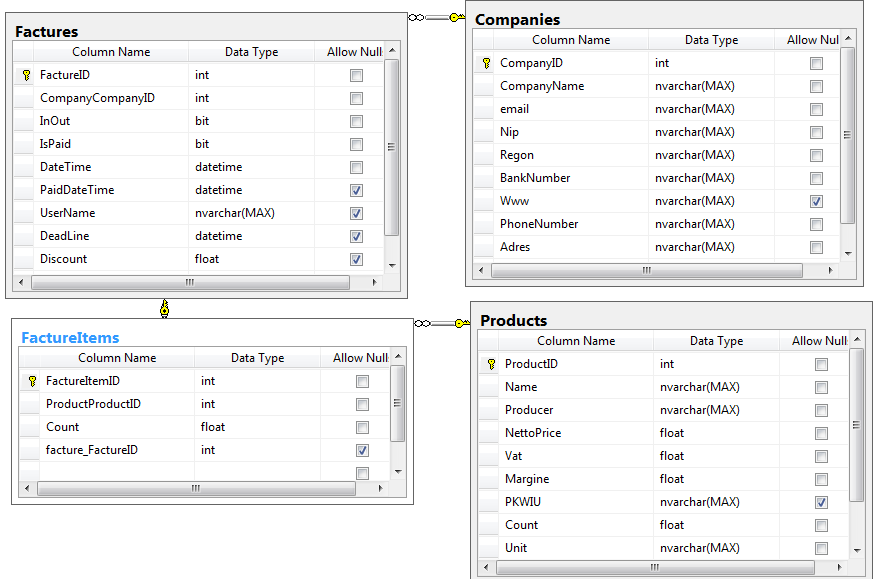


Rys. 2. Diagram encji – uproszczony

Powyższy diagram przedstawia uproszczony model danych, zostało na nim przedstawionych zaledwie pięć tabeli, a wyszczególnione pola to klucze główne i obce. Model uproszczony został zastosowany, aby należycie przedstawić związki, a nie zaciemniać obrazu zwykłymi polami.

Tabela „Products” to kontener przechowujący wszystkie informacja o elementarnym składniku systemu, jakim jest każdy przechowywany w magazynie towar. Wszystkie produkty posiadają swoje unikatowe pole „ProductID”, które jest kluczem głównym. Bardzo podobnie sytuacja wygląda z tabelą „Companies”, która przechowuje informacje o klientach magazynu i analogicznie posiada pole „CompanyID”, które jest kluczem głównym. Elementem systemu, do którego zadań należy połączenie powyższych struktur, jest tabela „Factures”, która przechowuje informacje o każdej fakturze. Oprócz klucza głównego występuje tutaj klucz obcy tabeli „Companies”, dzięki temu zrealizowano zależność między fakturą, a klientem, mianowicie, każda faktura może być wystawiona na rzecz tylko jednego klienta. Sytuacja z odzwierciedlaniem zależności towarów na fakturze jest inna, ponieważ niejednokrotnie występuje tam wiele towarów. Aby zrealizować to połączenie zastosowano tabele „FactureItems”, której zadaniem jest przechowywanie informacji o konkretnym wpisie na fakturze. Tabela oprócz swojego klucza głównego przechowuje klucze obce faktury, do której dodano wpis oraz klucz obcy towaru, którego wpis dotyczy. Do przechowywania informacji o użytkownikach systemu wykorzystano wbudowany w ASP.NET MVC mechanizm jakim jest ASP.NET MembershipProvider, który w bardzo praktyczny sposób pozwala zrealizować wszystkie zadania związane z użytkownikami, ich danymi oraz uprawnieniami. Tabela „Facture” posiada pole, przechowujące nazwę użytkownika, który wystawił fakturę.

### Model szczegółowy



Rys. 2. Diagram encji – szczegółowy

Powyższy diagram dostarcza szczegółowych informacji na temat wszystkich typów danych przechowywanych w bazie.

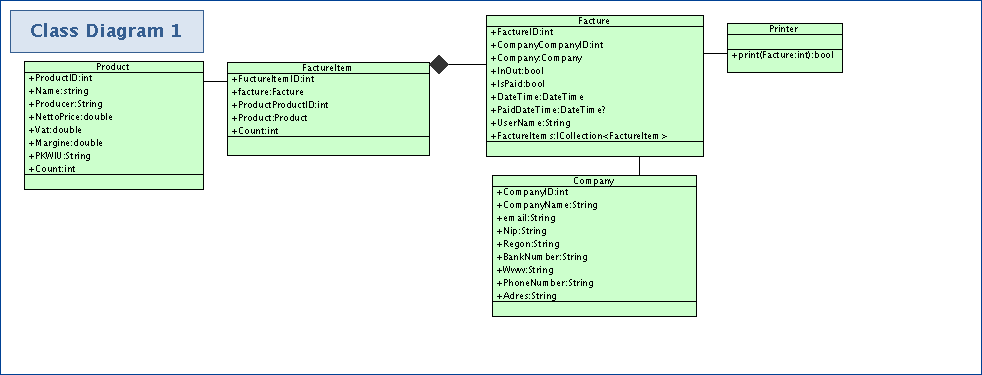
W tabeli „Companies” dodano takie kolumny jak Adres, NIP, Regon, Email oraz wiele innych, głównym celem tak zbudowanej tabeli jest przechowywanie jak najbardziej dokładnych i precyzyjnych informacji o każdym kliencie. Typ danych tutaj zastosowany to przeważnie nvachar(MAX), jego dużą zaletą jest elastyczność, dzięki czemu nie trzeba się martwić, że np. jakiś bardzo długi adres klienta spowoduje awarie całego systemu. Przewidziano również sytuacje, że jakaś firma może nie mieć adresu e-mail, czy strony www, dlatego zastosowano opcje umożliwiającą przechowywanie w bazie wartości NULL.

Produkty wpisane do bazy danych mogą być od razu wykorzystane w systemie, dlatego w większości przypadków w kolumnach tabeli „Products” zastosowano opcje uniemożliwiającą wpisania pustej wartości, dzięki temu w bazie danych będzie zachowana spójność i będzie można uniknąć wielu błędów. Dla kolumn Name, Producer oraz PKWiU zastosowano typ nvachar(MAX) z racji na trudność przewidzenia jakie dane tam mogą być wpisane. Kolumny Price, Vat czy Margine reprezentują wartości zmiennoprzecinkowe, dlatego wybrano typ float. Kolumna Count jest związana z liczbą określonego Towaru na magazynie, z racji tego że każdy towar posiada znormalizowaną jednostkę sprzedażową jak liczba sztuk, metry bieżące, kwadratowe, sześcienne, waga, objętość czy inne zastosowano tutaj typ całkowity.

Tabela „FactureItems” oprócz kluczy posiada kolumnę „Count”, która reprezentuje ilość konkretnego towaru na fakturze.

Tabela „Factures” posiada kolumny InOut i IsPaid typu bit, jest to typ zerojedynkowy lub inaczej boolowski. Kolumna InOut przechowuje informacje o tym czy faktura jest zakupowa czy sprzedażowa, natomiast kolumna IsPasd mówi o tym czy faktura została już zapłacona. Kolumny DateTime oraz PaidDateTime to odpowiednio informacje o dacie wystawienia faktury oraz o dacie zapłacenia, do przechowywania tych wartości zastosowano typ datetime. Ostatnią kolumną w tej tabeli jest UserName, dane w tej kolumnie to nazwy użytkowników, którzy wystawili bądź jako ostatni edytowali fakturę.

## Diagram klas



Rys. 2. Diagram klas

Na rysunku 2 przedstawiono diagram klas który odpowiada odzwierciedleniu modeli wzorca projektowego MVC. Kluczową klasą w systemie „Magazynier GT” jest „Product”. Odzwierciedla on elementarną jednostkę produktu w systemie. Klasa „FactureItem” odwzorowuje poszczególne pozycje na fakturze. Na podstawie „Product” opierają się klasa „FactureItem” i „Facture”. Klasa „Facture” korzystając z Klas „FactureItem” i „Company” przechowuje wszystkie potrzebne wartości do utworzenia faktury dzięki czemu za pomocą klasy „Printer” możemy wydrukować fakturę.

Klasa Product jest opisywana poprzez pola:

ProductID - zmienna publiczna o typie całkowitym.

Przechowuje niepowtarzalny identyfikator produktu.

Name - zmienna publiczna przechowująca ciąg znaków.

Przechowuje nazwę produktu.

Producer - zmienna publiczna przechowująca ciąg znaków.

Przechowuje nazwę producenta produktu.

NettoPrice - zmienna publiczna o typie zmiennoprzecinkowym podwójnej precyzji.

Przechowuje cenę jednostkowa produktu.

Vat - zmienna publiczna o typie zmiennoprzecinkowym podwójnej precyzji.

Przechowuje vat o jaki trzeba naliczyć dla produktu.

Margine - zmienna publiczna o typie zmiennoprzecinkowym podwójnej precyzji.

Przechowuje marże jaką trzeba nałożyć na produkt.

PKWIU - zmienna publiczna przechowująca ciąg znaków.

Przechowuje symbol Polskiej Klasyfikacji Wyrobów i Usług.

Count - zmienna publiczna o typie całkowitym.

Przechowuje liczbę produktów.

Klasa Company jest opisywana poprzez pola:

CompanyID - zmienna publiczna o typie całkowitym.

Przechowuje niepowtarzalny identyfikator firmy.

CompanyName - zmienna publiczna o typie ciąg znaków.

Przechowuje nazwę firmy.

email - zmienna publiczna przechowująca ciąg znaków.

Przechowuje Adres email firmy.

Nip - zmienna publiczna przechowująca ciąg znaków.

Przechowuje NIP firmy.

Regon - zmienna publiczna przechowująca ciąg znaków.

Przechowuje REGON firmy.

BankNumber - zmienna publiczna przechowująca ciąg znaków.

Przechowuje numer konta bankowego firmy.

Www - zmienna publiczna przechowująca ciąg znaków.

Przechowuje adres strony internetowej firmy.

PhoneNumber - zmienna publiczna przechowująca ciąg znaków.

Przechowuje numer telefonu firmy.

Adres - zmienna publiczna przechowująca ciąg znaków.

Przechowuje adres firmy.

Klasa FactureItem jest opisywana poprzez pola:

FactureItemID zmienna publiczna o typie całkowitym.

Przechowuje niepowtarzalny identyfikator faktury.

facture zmienna publiczn typu Facture

Przechowuje pojedynczą fakturę.

ProductProductID zmienna publiczna o typie całkowitym.

Przechowuje identyfikator produktu.

Product zmienna publiczn typu Product

Przechowuje zmienna typu Produkt.

Count zmienna publiczna o typie całkowitym.

Przechowuje ilość faktur.

Klasa Facture jest opisywana poprzez pola:

FactureID - zmienna publiczna o typie całkowitym.

Przechowuje niepowtarzalny identyfikator faktury.

CompanyCompanyID - zmienna publiczna o typie całkowitym.

Przechowuje niepowtarzalny identyfikator z klasy Company.

Company - zmienna publiczna o typu Company.

Przechowuje zmienna typu Company.

InOut - zmienna publiczna o typie logicznym.

Określa czy faktura jest zakupowa (true) czy sprzedażowa (false).

IsPaid - zmienna publiczna o typie logicznym.

Określa czy faktura jest zapłacona (true).

DateTime - zmienna publiczna przechowująca datę.

Przechowuje datę wystawienia faktury.

PaidDateTime - zmienna publiczna przechowująca datę.

Przechowuje datę zaksięgowania faktury.

UserName - zmienna publiczna przechowująca ciąg znaków.

Przechowuje użytkownika który ostatni modyfikował fakturę.

FactureItems - zmienna publiczna przechowujący kolekcję typu FactureItem.

Przechowuje listę wszystkich faktur.

Klasa Printer:

public bool print(Facture faktura) – metoda publiczna zwracająca wartość logiczną przyjmująca typ Facture jako argument.

* + Zwraca true jeżeli drukowanie się powiodło

# Szczegółowy scenariusz działania programu

Dostęp do aplikacji możliwy jest poprzez protokół „http” przy użyciu dowolnej przeglądarki spełniającej standardy W3C i posiadającej obsługę języka JavaScript. Po wyświetleniu ekranu powitalnego anonimowy użytkownik ma możliwość zalogowania się do systemu.

## Możliwości użytkownika

* Logowanie. Podczas procesu uwierzytelniania następuje sprawdzenie tożsamości użytkownika na podstawie indywidualnego loginu i hasła.
  + Sukces – przeniesienie na stronę główną systemu oraz nadanie uprawnień użytkownikowi na podstawie przydzielonej mu roli w systemie.
  + Porażka – na aktualnie widocznej stronie zostanie wyświetlona informacja, dlaczego logowanie nie przebiegło pomyślnie
* Zmiana hasła

Każdy użytkownik ma możliwość zmiany hasła, ale tylko administrator może zmieniać hasła innym użytkownikom. Jeżeli użytkownik chce zmienić hasło powinien kliknąć „zmień hasło” w dolnej części strony. Zostanie wyświetlone okno, w którym należy wpisać aktualne hasło oraz nowe i je powtórzyć, a następnie zatwierdzić

* + Sukces – zostanie wyświetlona odpowiednia informacja oraz nastąpi zmiana hasła.
  + Porażka – wyświetlenie informacji dlaczego hasło nie może zostać zmienione
* Wylogowanie

Po skończonej pracy, każdy pracownik, jeżeli nie będzie korzystał z systemu może się wylogować, dzięki temu zapewniona będzie gwarancja, że niepowołane osoby mogą wprowadzić jakieś niepoprawne dane. Opcja „wyloguj” widnieje na samym dole strony.

## Zarządzanie użytkownikami

Funkcja dostępna tylko dla administratorów, po zalogowaniu administratora w menu głównym pojawia się opcja „Pracownicy”. Po jej wybraniu administrator zobaczy listę użytkowników. Przy każdym rekordzie pojawią się dwie opcja „resetu hasło” i „zmień hasło”.

* Wybranie przycisku „Dodaj nowego pracownika” z menu bocznego
  + Użytkownik wypełnia pola, które będą opisywać nowego pracownika
  + Walidacja wprowadzonych danych
    - Sukces – dodanie pracownika do bazy danych, automatyczna aktywacja konta
    - Porażka – Wyświetlenie komunikatu o błędzie
* Wybranie przycisku „Resetuj hasło” , następuje zresetowanie hasła, a nowe, automatycznie wygenerowane hasło zostaje przesłane pracownikowi na jego adres e-mail, który został przez administratora podczas zakładania konta.
* Wybranie przycisku „Zmień hasło” , powoduje pojawienie się strony, na której administrator może ustalić dla swojego pracownika konkretne hasło.
  + Użytkownik wypełnia pola
  + Walidacja wprowadzonych danych
    - Sukces – zmiana hasła pracownika
    - Porażka – Wyświetlenie komunikatu o błędzie

## Zarządzanie magazynem

W systemie Magazynier GT przewidziano możliwość modyfikacji oraz podglądu szczegółowych danych o firmie. Możliwości w zakresie edycji ma tylko administrator, a odczytywać te informacje może, każdy zalogowany pracownik. Aby korzystać z tej możliwości należy kliknąć na głównym menu zakładkę firma, w zależności od poziomu uprzywilejowani użytkownika zostanie wyświetlona, albo strona z informacjami, albo strona umożliwiająca ponadto ich edycje.

## Zarządzanie bazą kontrahentów

Zarządzanie bazą kontrahentów aktualnego podmiotu gospodarczego jest możliwe po wybraniu w menu głównym zakładki „Klienci”. Skutkiem jest wylistowanie kontrahentów wraz z ich opcjami edycyjnymi. W kolejnym kroku użytkownik ma możliwość:

* Wybranie przycisku „Dodaj nowego klienta” z menu bocznego
  + Użytkownik wypełnia następujące pola: nazwa, opis, regon, nip, adres, nr konta, strona WWW, adres email, telefon kontaktowy
  + Walidacja wprowadzonych danych
    - Sukces – dodanie klienta do bazy danych
    - Porażka – Wyświetlenie komunikatu o błędzie
* Wybranie przycisku „Usuń klienta”
  + Wyświetlenie okna z zapytanie o potwierdzenie usunięcia
    - Sukces ‑ Usunięcie klienta z bazy danych
    - Porażka ‑ Anulowanie operacji
* Wybranie przycisku „Modyfikuj klienta”
  + Użytkownik modyfikuje wybrane pola: nazwa, opis, regon, nip, adres, nr konta, strona WWW, adres email, tel kontaktowy
  + Walidacja wprowadzonych danych
    - Sukces – modyfikacja klienta w bazie danych
    - Porażka – Wyświetlenie komunikatu o błędzie
* Wybranie przycisku „Szczegóły klienta”
  + Użytkownik może odczytać interesujące go dane o kliencie

## Zarządzanie produktami

W celu wylistowania produktów znajdujących się obecnie w bazie danych użytkownik powinien przejść do zakładki „Towary” w menu głównym programu. Podobnie jak w przypadku klientów, tutaj również pojawiają się opcje przy każdym rekordzie. Moduł ten daje poniższe możliwości:

* Wybranie przycisku „Dodaj nowy Towar” z menu bocznego
  + Użytkownik wypełnia następujące pola: nazwa, cena netto, marża, VAT, itd
  + Walidacja wprowadzonych danych
    - Porażka – Wyświetlenie komunikatu o błędzie
    - Sukces – dodanie produktu do bazy danych
* Wybranie przycisku „Usuń produkt”
  + Wyświetlenie okna z zapytanie o potwierdzenie usunięcia
    - Sukces ‑ Usunięcie produktu z bazy danych
    - Porażka ‑ Anulowanie operacji
* Wybranie przycisku „Modyfikuj produkt”
  + Użytkownik modyfikuje wybrane pola: nazwa, cena netto, marża, VAT, opis, uwagi
  + Walidacja wprowadzonych danych
    - Sukces – modyfikacja produktu w bazie danych
    - Porażka – Wyświetlenie komunikatu o błędzie
* Wybranie przycisku „Szczegóły produkt”
  + Użytkownik może przeglądać szczegółowe informacje o produkcie

## Zarządzanie fakturami

Główna funkcjonalność aplikacji jest dostępna po wybraniu zakładki „Faktury” w głównym menu. Na stronie zostaje wyświetlona lista faktur z podstawowymi informacjami, obok każdej faktury widnieją opcje. Modół daje poniższe możliwości:

* Wybranie przycisku „Dodaj nową fakturę” z menu bocznego
  + Użytkownik wybiera klienta
  + Użytkownik opcjonalnie zaznacza box‑a „zapłacono”
  + Użytkownik wybiera czy faktura jest zakupowa, czy sprzedażowa
  + Weryfikacja danych
    - Sukces – Dodanie faktury do bazy danych, przeniesienie użytkownika do okna, w którym będzie dodawał produkty do faktury
    - Porażka – Wyświetlenie komunikatu o błędzie
* Wybranie przycisku „Szczegóły”
  + Użytkownik ma możliwość przeglądania dokładnych informacji o fakturze, w menu bocznym widnieje opcja umożliwiająca edycje tych danych.
* Wybranie przycisku „Edytuj Fakturę”
  + Użytkownik opcjonalnie modyfikuje „datę”
  + Użytkownik opcjonalnie modyfikuje „klient”
  + Użytkownik opcjonalnie modyfikuje box‑a „Zapłacono”
  + Weryfikacja danych
    - Sukces – Modyfikacja faktury w bazie danych
    - Porażka – Wyświetlenie komunikatu o błędzie
* Wybranie przycisku „Usuń Fakturę”
  + Wyświetlenie okna z zapytanie o potwierdzenie usunięcia
    - Sukces ‑ Usunięcie faktury z bazy danych
    - Porażka ‑ Anulowanie operacji

Utworzenie, edycja faktury powoduje automatyczne wywołanie skutku magazynowego, czyli jego aktualizacje. System weryfikuje dane względem ilości, jaką może zostać sprzedana względem tej, która aktualnie znajduje się w magazynie. Niezaznaczenie opcji zapłacone spowoduje wyświetlenie faktury w płatnościach niezapłaconych. W przypadku usunięcia faktury użytkownik może zadecydować czy chce wywołać skutek magazynowy, przewidziano sytuacje omyłkowego utworzenia faktury bez pokrycia w towarze.

* Wybranie przycisku „Drukuj Fakturę”
  + Automatyczne generowanie pliku pdf gotowego do druku, na dokumencie są wyszczególnione dane o fakturze oraz wszystkie informacje na temat produktów, które się na niej znajdują.

## Zarządzanie towarami na fakturze

Nieodłącznym elementem w procesie tworzenia faktury jest dodawanie do niej towarów. W systemie Magazynier GT zastosowano innowacyjną technologie, która polega na rozdzieleniu danych faktury, oraz towarów które się na niej znajdują. Po poprawnym wypełnieniu danych na fakturze pracownik magazynu zostaje przeniesiony na stronę, na której może dodawać produkty do faktury. Mechanizm dodawania produktów został zbudowany na technologii Ajax, dzięki czemu w bardzo szybki i przejrzysty sposób można dodawać towary.

* Dodawanie towarów do faktury
  + Użytkownik wybiera nazwę produktu
  + Użytkownik wybiera ilość jaka została zakupiona/sprzedana
  + Naciśnięcie przycisku „Dodaj” i Weryfikacja danych
    - Sukces – dodanie towaru do faktury, odświeżenie elementu strony odpowiedzialnego za przedstawienie listy towarów na fakturze
    - Porażka – wypisanie informacji dlaczego towar nie może zostać dodany
* Edycja towarów na fakturze
  + Po dodaniu towaru do faktury przy każdym rekordzie pojawia się opcja „Edytuj”, po jej wybraniu następuje przerenderowanie elementu strony, w którym wcześniej był panel do dodawania nowych towarów, w jego miejscu pojawia się panel do edycji wybranego towaru, stare wartości produktu zostają automatycznie przedstawione w odpowiednich polach.
  + Po wprowadzeniu ewentualnych modyfikacji i ich zatwierdzeniu następuje walidacja danych
    - Sukces – edycja towaru na fakturze, odświeżenie elementu strony odpowiedzialnego za przedstawienie listy towarów na fakturze, przerenderowanie widoku w którym była edycja, z powrotem pojawi się tam panel do dodawania nowych towarów
    - Porażka – wypisanie informacji dlaczego towar nie może zostać edytowany
* Usuwanie towarów z faktury
  + Podobnie jak opcja edycji obok każdego towaru na fakturze pojawia się również opcja „usuń”.
  + Po jej naciśnięciu pojawi się okno z pytaniem skierowanym do użytkownika, czy na pewno chce usunąć ten towar.
    - Sukces – rekord zostaje usunięty z faktury
    - Porażka – nie wprowadzono zmian

## Zarządzanie rozrachunkami

W celu zarządzania płatnościami wobec kontrahentów należy wybrać opcję „Płaności” w memu głównym. Moduł ten daje poniższe możliwości użytkownikowi:

* Wyświetlenie nieuregulowanych oraz uregulowanych faktur
* Istnieje możliwość filtrowania faktur
* Wybranie przycisku „Uregulowano”
  + Sukces – Modyfikacja faktury w bazie danych
  + Porażka – Wyświetlenie komunikatu o błędzie

Powyższa funkcjonalność w znaczący sposób ułatwia kontrolę bieżącego salda płatności. W celu skrócenia czasu na poszukiwanie witryny banku, system wyświetla najważniejsze linki partnerów bankowych.

## Zarządzanie wycofaniami magazynowymi

Wycofania magazynowe to moduł, który ma zapewnić elastyczność systemu na nieprzewidziane sytuacje tj. kradzież, uszkodzenie towaru, przekroczenie daty ważności. Aby użyć tej opcji należy wybrać z menu głównego zakładkę „wycofania”. Magazynier GT nie zapisuje przeprowadzonych wycofań, pozwala tylko i wyłącznie wycofać z magazynu określoną ilość konkretnego towaru.

* Dodawanie wycofania
  + Należy wybrać towar oraz jego ilość
  + Po naciśnięciu przycisku „wycofaj”
    - Sukces – z bazy zostaje usunięta określona ilość towaru
    - Porażka – nie wprowadzono zmian, zostaje wypisany komunikat

## Moduł drukowania i Remanent

Każdą fakturę można wydrukować poprzez wybranie polecenia drukuj znajdującego w wierszu każdej faktury wylistowanej poprzez menu „Faktury”.

System oprócz drukowania każdej faktury pozwala na utworzenie i wydrukowanie aktualnego stanu magazynowego, dzięki temu w łatwy i szybki sposób można uzyskać namacalną listę towarów znajdujących się na magazynie.

Aby skorzystać z tej opcji wystarczy kliknąć zakładkę „Remanent” na głównym menu, zostanie automatycznie stworzony wielostronicowy plik pdf gotowy do druku.

# Autorzy i podział prac

//Dane osobiste usunięte

|  |  |  |
| --- | --- | --- |
| Lp. | Czynność | Osoba odpowiedzialna |
|  | Projektowanie i przygotowanie |  |
| 1 | Model danych | 1, 2, 3, 4 |
| 2 | Budowanie schematu bazy danych - mapowanie | 2, 4 |
| 3 | Uzupełnienie baz danych | 2 |
|  |  |  |
|  | Implementacja silnika |  |
| 1 | Kontroler zarządzania użytkownikami | 4 |
| 2 | Kontroler zarządzania klientami | 4 |
| 3 | Kontroler zarządzania towarami | 4 |
| 4 | Kontroler zarządzania fakturami | 4 |
| 5 | Kontroler zarządzania towarami na fakturze | 4 |
| 6 | Kontroler zarządzania wycofaniami | 4 |
| 7 | Kontroler zarządzania płatnościami | 2 |
| 8 | Pierwszy moduł drukowania | 1 |
| 9 | Drugi moduł drukowania | 1,2 |
| 10 | Moduł statystyk | 1,2 |
| 11 | Wsparcie programistyczne dla 4 | 1,2 |
| 12 | Sortowanie (klienci, towary) | 1 |
|  |  |  |
|  | Budowa Designu |  |
| 1 | Schemat podstawowy | 1,2,3,4 |
| 2 | Budowa właściwego Layout’u | 3 |
| 3 | Tworzenie grafiki | 3 |
| 4 | Formatowanie i prezentacja danych | 3 |
|  |  |  |
|  | Testowanie |  |
| 1 | Testowanie silnika | 5 |
|  |  |  |
|  | Dokumentacja |  |
| 1 | Dokumentacja –opracowanie ogólne | 1, 3, 5 |
| 2 | Opis funkcjonalności | 1 |
| 3 | Narzędzia i technologie | 1 |
| 4 | Diagramy | 2, 4, 5 |
| 5 | Szczegółowy scenariusz działania | 1, 4 |
|  |  |  |
|  | Instrukcja obsługi |  |
| 1 | Cała Instrukcja | 5 |