Szpital

Inżynieria oprogramowania

wtorek, 10:30

rok akademicki 2015/2016

O projekcie

Cel

Celem naszej grupy jest stworzenie systemu, z którego mogliby korzystać pracownicy różnych działów szpitala. System ten   
będzie ułatwiał pracownikom wykonywanie ich codziennych obowiązków i usprawniał pracę w szpitalu.

Założenia

Nasz system będzie się składał z 7 podstawowych komponentów:

* + apteka szpitalna
  + kuchnia
  + magazyn
  + finanse
  + rejestracja pacjentów
  + system użytkowników i uprawnień
  + zarządzanie dyżurami, urlopami i rezerwacją sal

Z systemu będzie korzystało 8 aktorów:

* + administratorzy
  + lekarze
  + pielęgniarki
  + pracownicy apteki
  + magazynierzy
  + kucharze
  + szef kuchni
  + dyrektor szpitala

W zależności od stanowiska, każdy będzie miał inne uprawnienia i dostęp do innych funkcjonalności systemu. Wszystkie moduły będą korzystały ze **wspólnej bazy danych.** Poszczególne komponenty będą ze sobą bezpośrednio powiązane.

Wymagania funkcjonalne

* System musi umożliwiać pracownikom logowanie się do systemu. Pracownicy, w zależności od stanowiska, muszą mieć inne uprawnienia, które będą umożliwiać korzystanie z konkretnych funkcjonalności.
* Administratorzy muszą mieć możliwość tworzenia, edycji   
  i usuwania kont użytkowników.
* Lekarze muszą mieć możliwość rejestrowania pacjentów, tworzenia, edycji i usuwania ich kart. Lekarze muszą również mieć możliwość zamawiania leków z apteki, rezerwowania sal dla pacjentów oraz ustalania swoich dyżurów/urlopów w systemie.
* Pielęgniarki muszą mieć możliwość zamawiania leków w aptece, ustalania swoich dyżurów/urlopów, ustalania menu dla pacjentów   
  i zamawiania dla nich posiłków. Pielęgniarki muszą mieć też możliwość edycji kart pacjentów.
* Pracownicy apteki muszą mieć dostęp do spisu leków, do instrukcji przygotowania leków oraz do archiwum zamówień zgłaszanych przez lekarzy. Muszą oni też mieć możliwość zamawiania brakujących leków do apteki z magazynu.
* Magazynierzy muszą mieć dostęp do archiwum zamówień zgłaszanych przez kucharzy i przez pracowników apteki. Muszą   
  oni też być w stanie udostępniać środki ze stanu magazynu oraz zamawiać brakujące środki u zewnętrznych dostawców.
* Kucharze muszą być w stanie zamawiać brakujące produkty spożywcze z magazynu, tworzyć menu i przeglądać zamówienia na posiłki, które składają pielęgniarki. Muszą mieć oni dostęp do całego zaopatrzenia kuchni.

Wymagania niefunkcjonalne

* Wszystkie moduły szpitala muszą pobierać dane bezpośrednio z bazy danych.
* Przesyłanie danych powinno być szybkie (max 2s po zapisie zmian) tak aby praca użytkowników nie była utrudniana.
* Użytkownicy powinni widzieć zmiany wprowadzone przez innych użytkowników systemu po odświeżeniu okna danej funkcjonalności.
* Gdy użytkownicy zaczynają edytować ten sam obiekt (np. kartę pacjenta) to:
  + Użytkownik który zapisze dane jako pierwszy nie otrzyma komunikatu o konflikcie, jego dane zostaną zapisane w systemie
  + Użytkownik który zapisze jako drugi otrzyma komunikat że ten sam rekord został wcześniej zmodyfikowany przez innego użytkownika, dane drugiego użytkownika nie będą zapisane w systemie i będzie musiał spróbować zapisać je jeszcze raz.

Ogólny diagram przypadków użycia

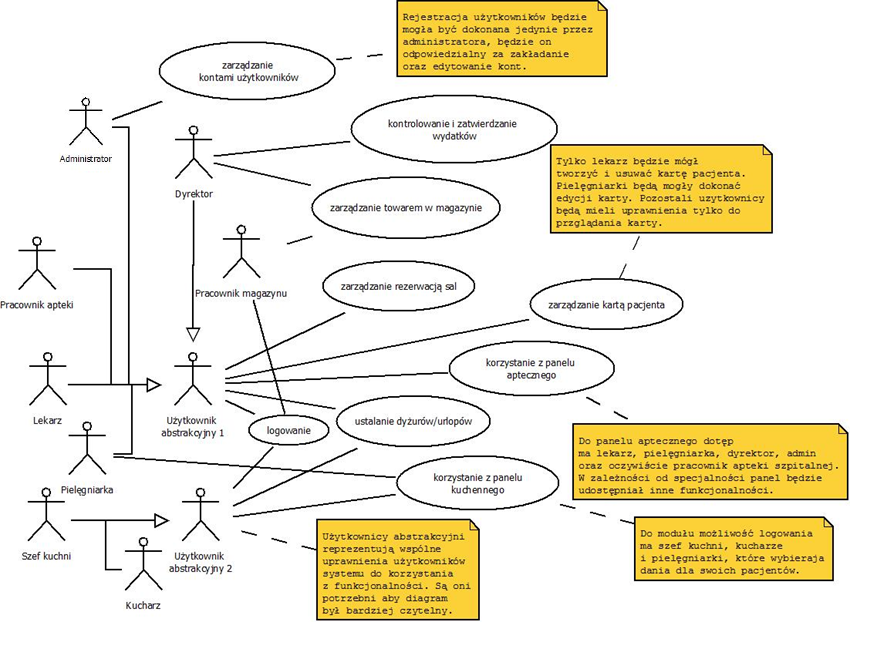


Diagram ten przedstawia ogólne przypadki użycia występujące w naszym systemie. Wszystkie te przypadki będą dokładniej opisane na diagramach poszczególnych modułów.

Ogólny diagram komponentów – interfejsy do podpisania

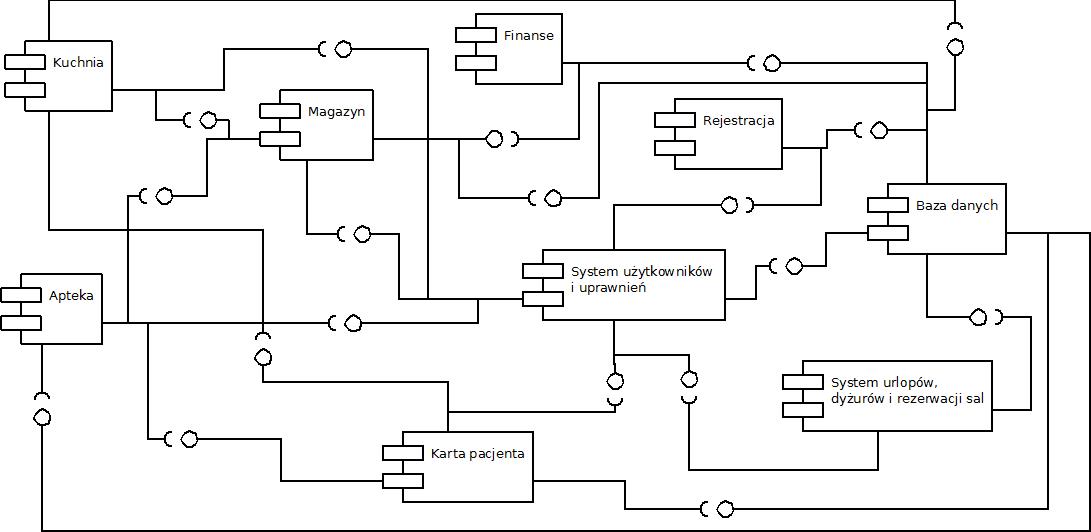


Diagram ten przedstawia ogólne powiązania między komponentami w naszym systemie. Jego części zostaną szczegółowo opisane w sekcjach związanych z poszczególnymi modułami.

Diagram wdrożeń

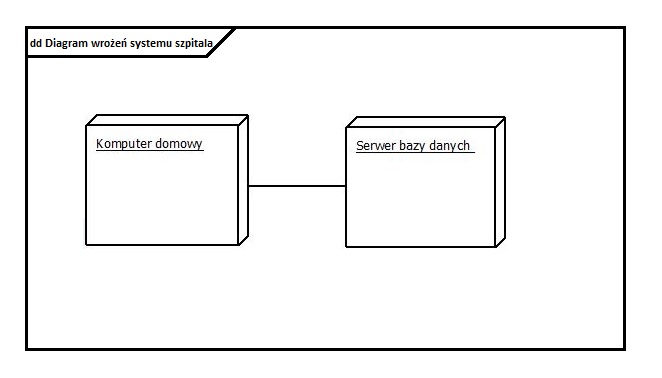


Diagram wdrożeń naszego systemu jest bardzo prosty. Z systemu tego można korzystać tylko na komputerach PC oraz laptopach. Nie używamy żadnych innych podsystemów. Nawiązujemy tylko połączenia z serwerem na którym znajduje się nasza baza danych.

Moduły

Apteka szpitalna

Zadaniem modułu jest umożliwienie korzystania z wielu przydatnych funkcjonalności dotyczących funkcjonowania w aptece szpitalnej. Łatwy w obsłudze panel apteczny będzie pomagał użytkownikom pracę z dużą ilością danych. Wszystkie wprowadzane zmiany zapisywane będą w systemie. Wyeliminuje to w dużym stopniu błędy i niedopowiedzenia oraz pomoże rozwiązywać konflikty między pracownikami szpitala.

Założenia

Nasz moduł będzie dostępny dla pięciu różnych aktorów:

* + Pracownicy apteki
  + Dyrektor
  + Administrator
  + Pielęgniarki
  + Lekarze
* W zależności od stanowiska pracownicy będą mieli różne uprawnienia do korzystania z funkcjonalności modułu.
* Nasz moduł będzie pobierał dane z bazy danych, która jest wspólna dla wszystkich modułów szpitala.
* Niektóre informacje potrzebne aptece szpitalnej będą udostępniane tylko poprzez połączenie z innymi komponentami: magazyn, system użytkowników i uprawnień.

Wymagania funkcjonalne

Moduł apteki szpitalnej w zależności od stanowiska użytkownika musi udostępniać różne funkcjonalności:

* Pracownicy apteki muszą mieć dostęp do wszystkich funkcjonalności panelu.
* Administratorzy powinni mieć dostęp do funkcjonalności, za wyjątkiem zamawiania leków dla pacjentów, przeglądania instrukcji tworzenia leków oraz edytowanie kart pacjentów.
* Dyrektorzy muszą mieć dostęp do funkcjonalności, za wyjątkiem edytowania kart pacjentów (nie mają oni bezpośredniego kontaktu z pacjentami) i przeglądania instrukcji tworzenia leków.
* Pielęgniarki i lekarze powinni mieć dostęp do wszystkich funkcjonalności panelu oprócz przeglądania instrukcji tworzenia leków.

Panel apteczny powinien być powiązany z innymi komponentami systemu: magazyn, system użytkowników i uprawnień.

Wymagania niefunkcjonalne

* Moduł apteki szpitalnej musi pobierać dane bezpośrednio z bazy danych. Przesyłanie danych powinno być szybkie (max 2s po zapisie zmian) tak aby praca użytkowników nie była utrudniana i żeby nie było niepotrzebnych konfliktów.
* Użytkownicy powinni widzieć zmiany wprowadzone przez innych użytkowników systemu po odświeżeniu okna danej funkcjonalności.

Diagram przypadków użycia

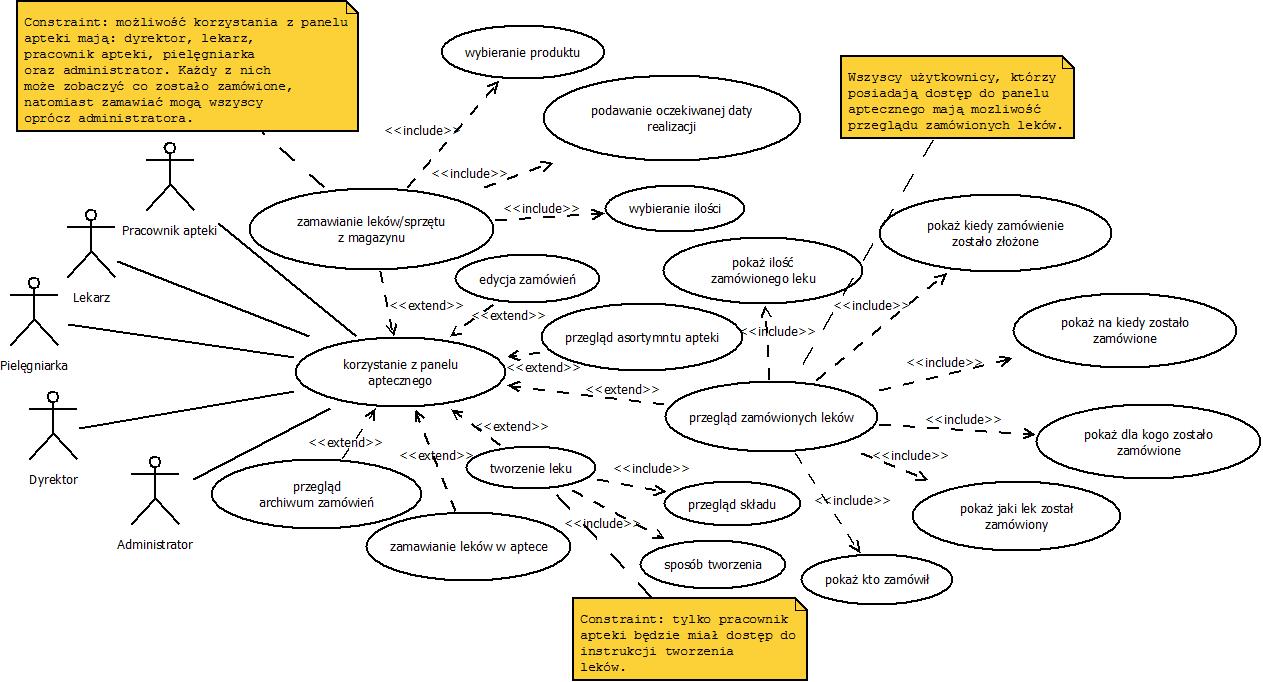
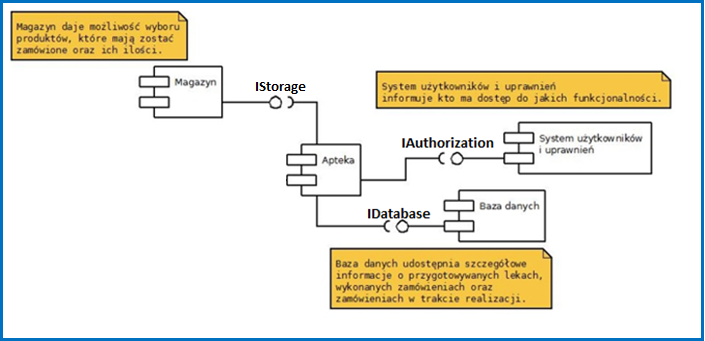


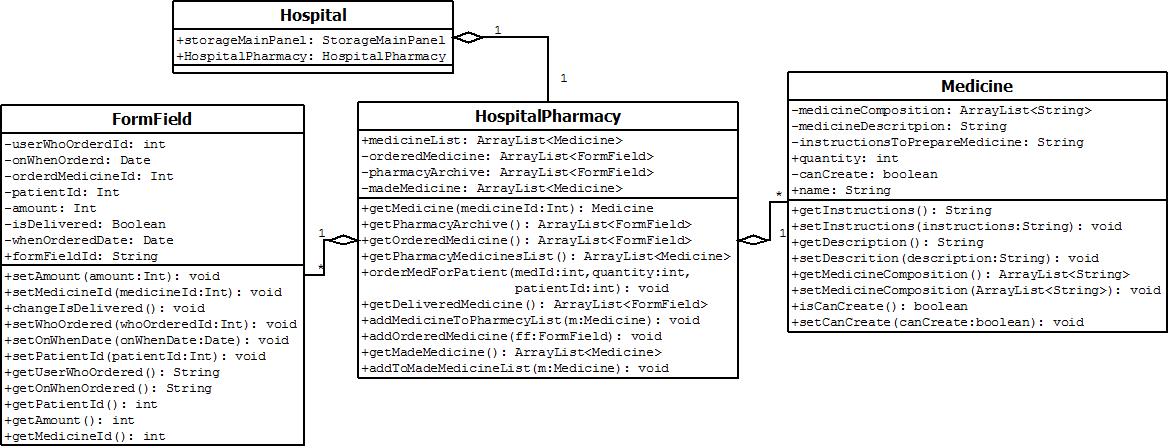
Diagram ten przedstawia wszystkie przypadki użycia występujące w module apteki szpitalnej. Moduł ten umożliwia zamawianie leków z magazynu do apteki wszystkim użytkownikom oprócz administratorowi systemu. Wszyscy użytkownicy oprócz administratora mogą również zamawiać leki w aptece. Każdy użytkownik ma możliwość zobaczenia tego co zostało zamówione oraz przeglądania archiwum zrealizowanych zamówień. W aptece istnieje możliwość stworzenia leku i umieszczenie go na liście leków dostępnych w aptece. Mogą to robić tylko pracownicy apteki. Użytkownicy sytemu mogą edytować zamówienia które zostały złożone rzez nich samych. Wszyscy użytkownicy mogą zobaczyć aktualny asortyment apteki.

Diagram komponentów



Moduł apteki szpitalnej pobiera dane z bazy danych wspólnej dla wszystkich modułów szpitala. Niektóre informacje udostępniane będą przez inne moduły: informacje o produktach dostępne będą w magazynie, a system użytkowników i uprawnień zapewnia, aby każdy użytkownik apteki miał dostęp do odpowiednich dla niego funkcjonalności.

Diagram klas



Klasa HospitalPharmacy jest główna klasą naszego modułu. Reprezentuje ona aptekę przechowując dane o lekach które aktualnie znajdują się w niej znajdują, o lekach które mogą być w niej zrobione oraz o zamówieniach które przychodzą do apteki i są realizowane przez aptekarzy. Klasa Medicine reprezentuje lek, a klasa FormField przechowuje informacje na temat aktualnych oraz już zrealizowanych zamówień.

Scenariusz nr 1 - Edycja zamówienia

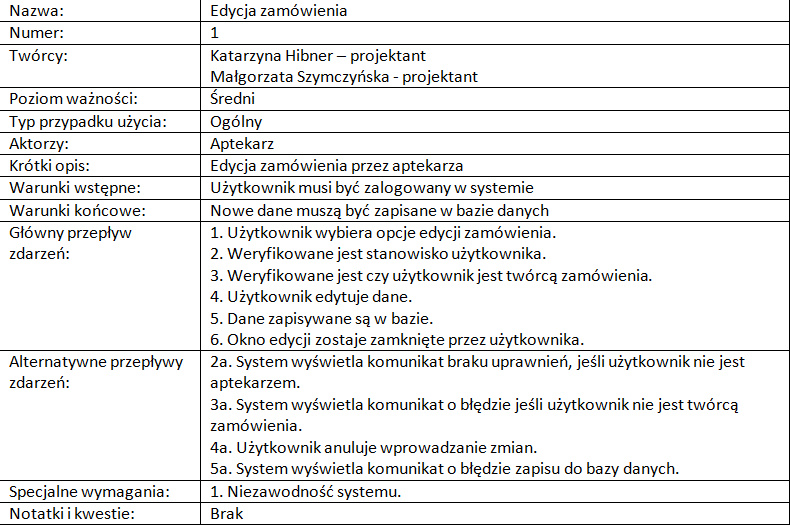


Diagram sekwencji nr 1 - Edycja zamówienia

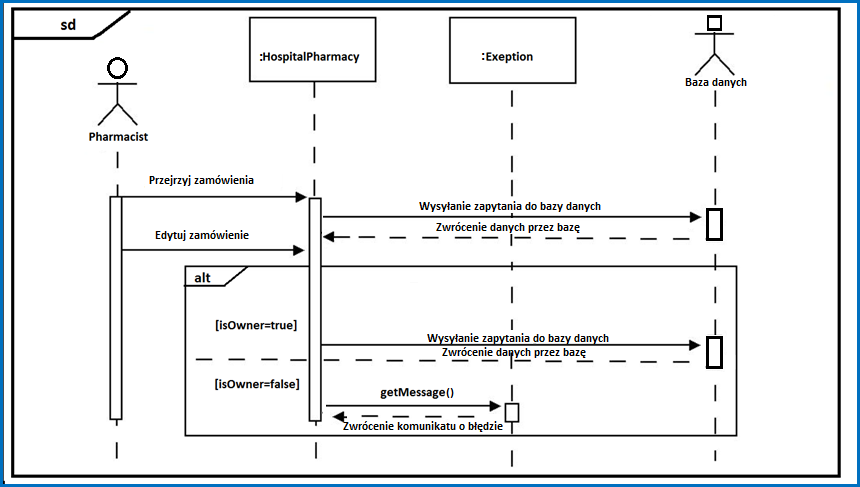


Diagram ten przedstawia sytuację w której użytkownik próbuje zobaczyć zamówione leki, a później edytować zamówienie. Aktorem w tym przypadku jest pracownik apteki. Gdy będzie on chciał zobaczyć zamówienia, zostanie wysłane zapytanie do bazy danych i dane zostaną zwrócone. Gdy spróbuje on edytować zamówienia sprawdzane jest czy dany użytkownik jest osobą która to zamówienie składała. Jeśli tak – dane zostaną pobrane z bazy i użytkownik będzie mógł dokonać edycji. Jeśli okaże się że użytkownik chce edytować zamówienie którego nie składał – zostanie wyświetlony komunikat z błędem. Użytkownicy nie mogą edytować zamówień składanych przez innych użytkowników. Mogą oni edytować tylko swoje własne.

Scenariusz nr 2 - Odczytanie składu leku

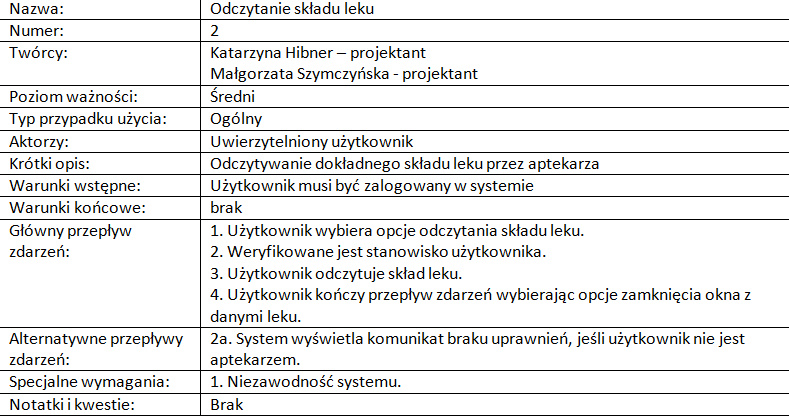


Diagram sekwencji nr 2 - Odczytanie składu leku

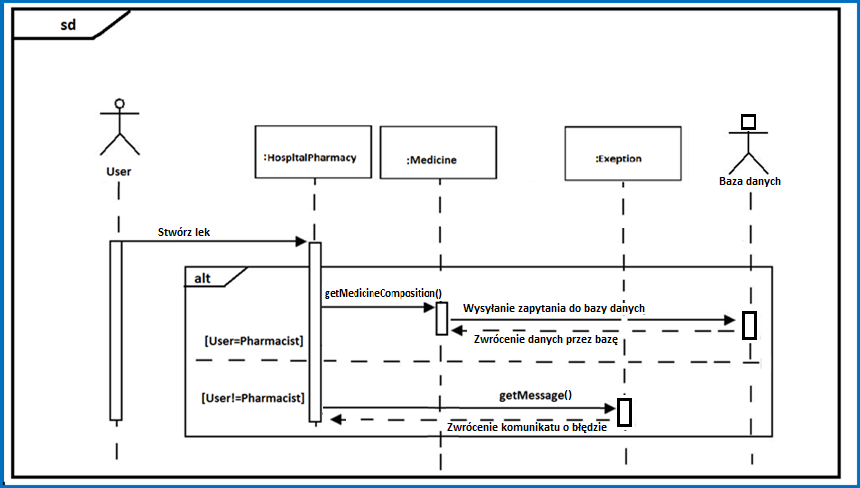


Diagram ten przedstawia próbę odczytania składu leku przez użytkownika. Jeżeli użytkownik będzie pracownikiem apteki to zapytanie do bazy danych zwróci mu skład danego leku. Jeśli użytkownik nie będzie pracownikiem apteki wtedy zostanie mu zwrócony odpowiedni komunikat z błędem.

Diagram czynności

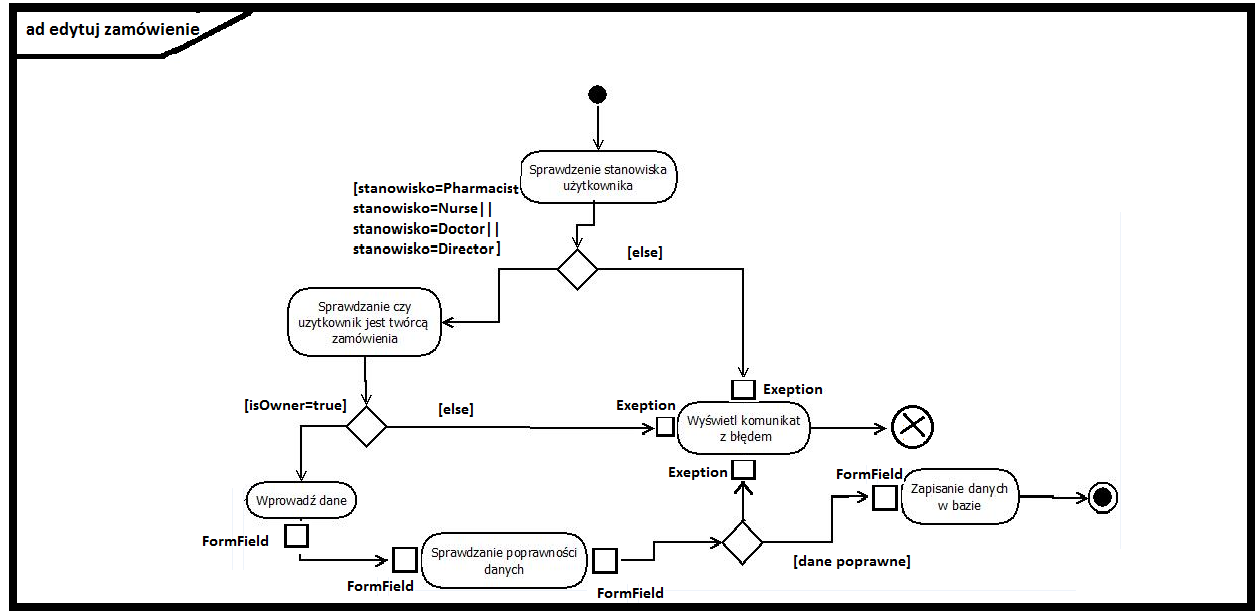


Diagram ten przedstawia czynności wykonywane przez moduł apteki w momencie gdy użytkownik będzie próbował edytować zamówienie złożone w aptece. Na początku sprawdzana jest funkcja jaką użytkownik pełni w szpitalu a później czy dany użytkownik stworzył zamówienie które wybrał do edycji.

Diagram stanów

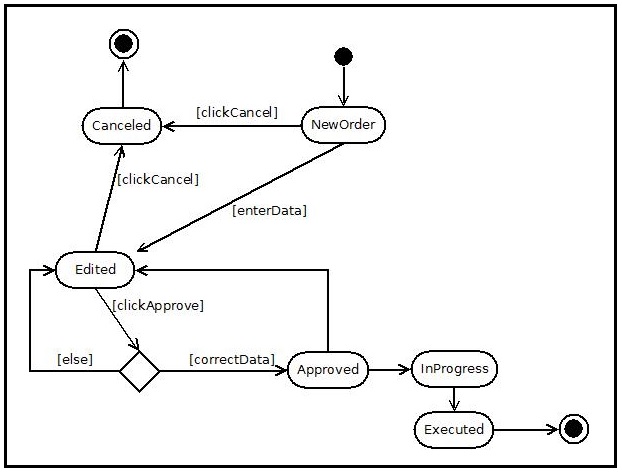


Diagram stanów przedstawia stany w jakich znajduje się zamówienie złożone przez użytkownika systemu. Dane będą poddawane walidacji zanim zostaną zapisane w bazie.

Diagram pakietów

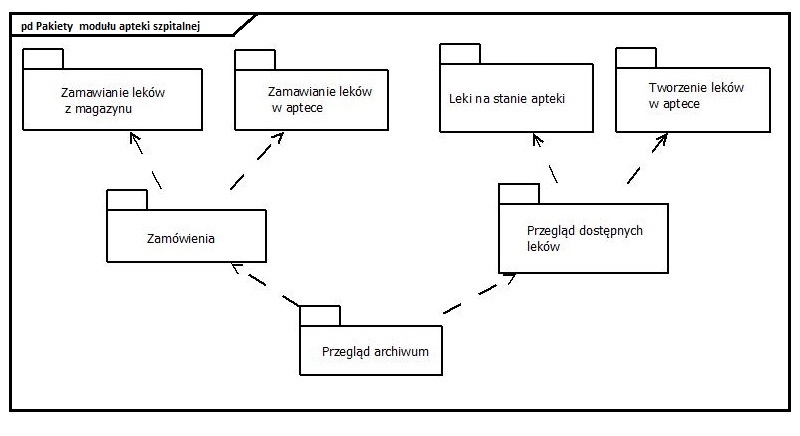


Diagram ten przedstawia podział modułu apteki na pakiety. Każdy pakiet opisuje grupę powiązanych ze sobą zadań systemu.

Diagram interakcji

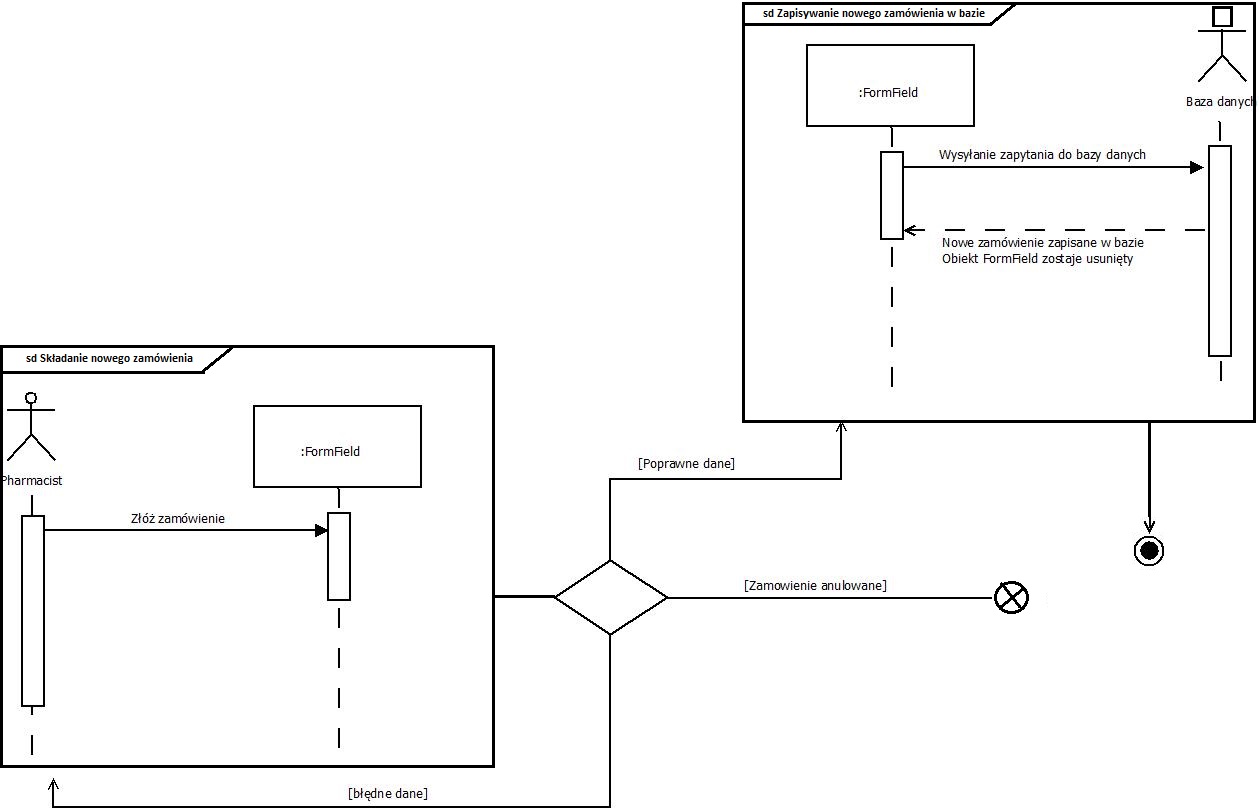


Diagram przedstawia proces składania nowego zamówienia. Początkowo tworzy się nowy obiekt FormField, następnie sprawdzana jest poprawność danych. Jeżeli typy danych są poprawne, zostaje wysłane zapytanie do bazy danych, które powoduje utworzenie nowego rekordu z danymi zamówienia. Jeżeli dane nie są poprawne, zapytanie do bazy nie jest wysyłane. Użytkownik ma również możliwość anulowania swojego zamówienia.

Diagram strukturalny

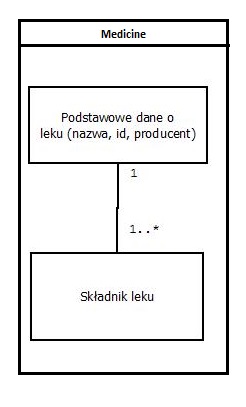


Diagram strukturalny pokazuje tutaj, że klasa Medicine składa się z różnych części. Każdy lek może się składać z wielu składników. Łącznik (connector) pokazuje związek między częściami klasy.

Diagram harmonogramowania

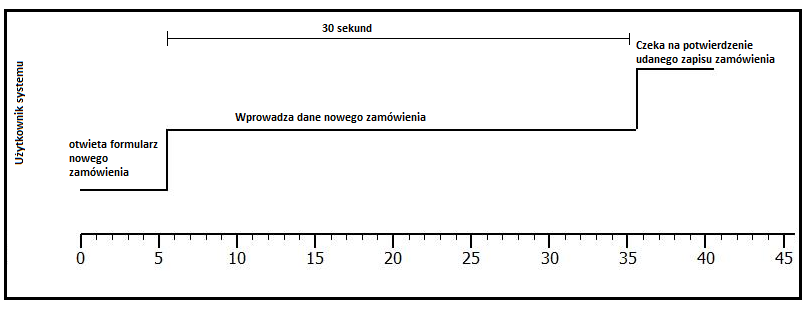


Diagram przebiegu czasowego (harmonogramowania) modułu apteki przedstawia ile czasu zajmuje użytkownikowi wykonanie danych czynności w systemie. Na diagramie został przedstawiony proces dodawania nowego zamówienia przez użytkownika.

Kuchnia

Cel i założenia

Celem naszej grupy jest zaprojektowanie jednego z modułów systemu dla szpitala, odpowiedzialnego za usprawnienie działania szpitalnej kuchni.

Moduł dostępny będzie tylko dla wybranych grup pracowników szpitala. Poza administratorem użytkownikami będą:

* szef kuchni,
* kucharze,
* pielęgniarki.

Każda osoba, w zależności od pełnionego stanowiska, będzie posiadała inny dostęp do modułu, a co za tym idzie, będzie mogła korzystać z innych funkcjonalności.

Wymagania

**Funkcjonalne:**

* Sprawdzenie stanu zasobów w kuchni (ilości produktów, daty ważności).
* Składanie zamówień do magazynu, w przypadku braku produktów do wytworzenia odpowiedniej ilości produktów (rejestracja zwiększenia zasobów po odbiorze zamówienia).
* Generowanie dziennego menu zawierającego dziennie po 3 posiłki, z czego każdy z nich będzie miał 2-3 opcje wyboru dania (w wyborze posiłków do menu, będzie mógł sprawdzać stan zasobów kuchni, by wyeliminować marnowanie żywności).
* Dodawanie i usuwanie posiłków z bazy (dostarczanie bądź wykreślanie przepisów).
* Odbieranie od pielęgniarek zamówień (przekazanie ilości dań kucharzom).
* Wydawanie posiłków (zarejestrowanie zużycia składników, sprawdzenie poprawności ilości wydania ze zleconym zamówieniem).
* Odczytania zamówienia na posiłki (ilość dań do wykonania na dany dzień).
* Przeglądania menu oraz przepisów na każde z dań wymienionych w bazie.
* Pobór zasobów z kuchni by wykonać posiłek (wiąże się z koniecznością rejestracji zmniejszenia ilości składników w zaopatrzeniu kuchni).
* Przegląd ustalonego menu.
* Wybór posiłku na każde z 3 dań dla każdego pacjenta (ustalany na podstawie karty pacjenta i zawartych w niej przeciwwskazań oraz jego aktualnego stanu).
* Zarejestrowanie wyborów (przekazanie szefowi kuchni) dokładnych ilości każdego z posiłków.

**Niefunkcjonalne:**

* Interfejs użytkownika powinien być prosty, funkcjonalny i intuicyjny. Nazwy elementów użytych w programie powinny być krótkie i zrozumiałe.
* Moduł powinien być zabezpieczony przed utratą danych w czasie awarii. Ponownie uruchomienie systemu powinno trwać najkrócej jak to możliwe.
* Moduł sprawdza poprawność wykonywanych czynności. Prawdopodobieństwo błędnej realizacji transakcji będzie jak najmniejsze.
* Ilość osób korzystających z modułu jednocześnie nie powinna wpływać na jego wydajność.

**Profil użytkownika aplikacji**

Aplikacja przeznaczona jest dla pracowników szpitali, przychodni czy prywatnych klinik. Ze względu na pełniona w placówce funkcji dostęp do funkcjonalności może być różny. System w prosty sposób organizuje pracę szpitalnej kuchni, a dzięki komunikacji z innymi modułami usprawnia działania całego szpitala.

**Ogólnie wyrażone zadania modułu**

Głównym zadaniem modułu jest dostarczenie części systemu wspomagającego prace w szpitalnej kuchni (generowanie menu, składanie zamówień na brakujące produkty, komunikacja między pielęgniarkami a kuchnią w celu przekazania ustalanych menu, odczytywanie, edycja i dodawanie posiłków do bazy, sprawdzanie, zmniejszanie oraz powiększanie zasobów kuchennych). Naszym zadaniem jest także opracowanie interfejsu granicznego dotyczącego naszego modułu.

**Wymagania odnośnie niezawodności działania**

Aplikacja przeznaczona jest dla państwowych i prywatnych placówek współpracujących z bardzo dużą ilością użytkowników, przez co wymagana jest niezawodność jej działania, wysoki poziom bezpieczeństwa, szybkość wykonywanych operacji oraz brak problemów z korzystaniem przez wielu użytkowników jednocześnie. Poprawne funkcjonowanie jest w dużym stopniu uzależnione od bazy danych, dlatego musi ona być zorganizowana i zoptymalizowana by operacje były jak najszybsze. Błędne działanie bądź podpięcie bazy danych będzie miało duży wpływ na niezawodność systemu.

**Poziom ochrony danych**

Dostęp do modułu otrzymywany jest po uprzednim zalogowaniu się do całego systemu. Użytkownik posiada swoje konto z loginem i hasłem oraz przypisanym zestawem uprawnień wyznaczanym przez administratora. Funkcjonalności te zapewnia moduł Użytkowników i Uprawnień.

**Standard komunikacji z użytkownikiem**

Komunikacja z użytkownikiem odbywa się przy wykorzystaniu myszy i klawiatury. Dokładny opis sterowania znajduje się w dokumentacji użytkownika.

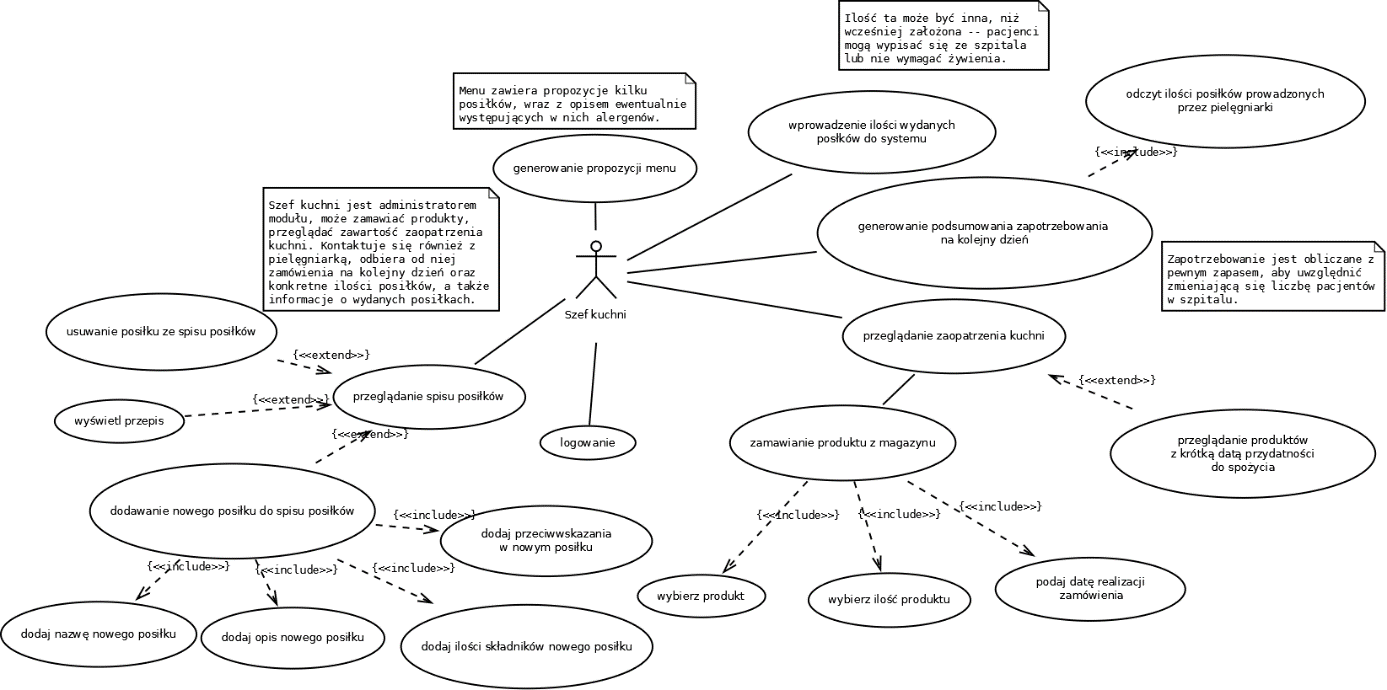
**Przewidywane kierunki rozwoju modułu**

Modyfikacją rozwijającą nasz moduł jest danie możliwości wyboru dań również przez pacjentów. W tym momencie to pielęgniarka na podstawie stanu pacjenta i przeciwwskazań w diecie wybiera posiłki. W przyszłości natomiast funkcjonalność tak mogłaby być udostępniona pacjentom by uwzględnić również ich osobiste preferencje i upodobania.

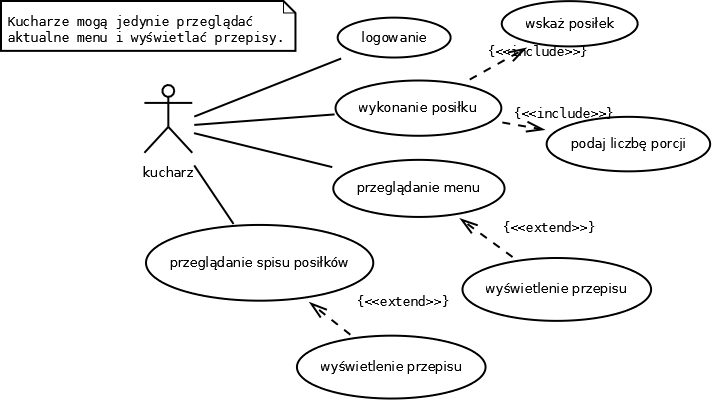
**Harmonogram prac, podział prac w zespole**

Prace przy tworzeniu modułu przebiegały zgodnie z harmonogramem zaproponowanym przez architektów projektu. Pomimo drobnych nieporozumień i opóźnień udało się nam zrealizować postawione przed naszym modułem wymagania.

Model konceptualny obiektów aplikacji

Diagramy przypadków użycia

Diagramy przypadków użycia podzieliłyśmy ze względu na pełnione funkcje w szpitalu.  
Pierwszy diagram przedstawia funkcjonalności, z których skorzystać może szef kuchni. Należy do nich generowanie menu, wprowadzenie zebranych zamówień od pielęgniarek, kontrolowanie magazynu, zamawianie produktów jak i przegląd spisu dostępnych posiłków. Jedynie szef kuchni może dodać nowy posiłek do bazy oraz edytować istniejące.



Kucharz po zalogowaniu może przede wszystkim wykonać posiłek pomniejszając zasoby oraz przeglądać przepisy i menu.

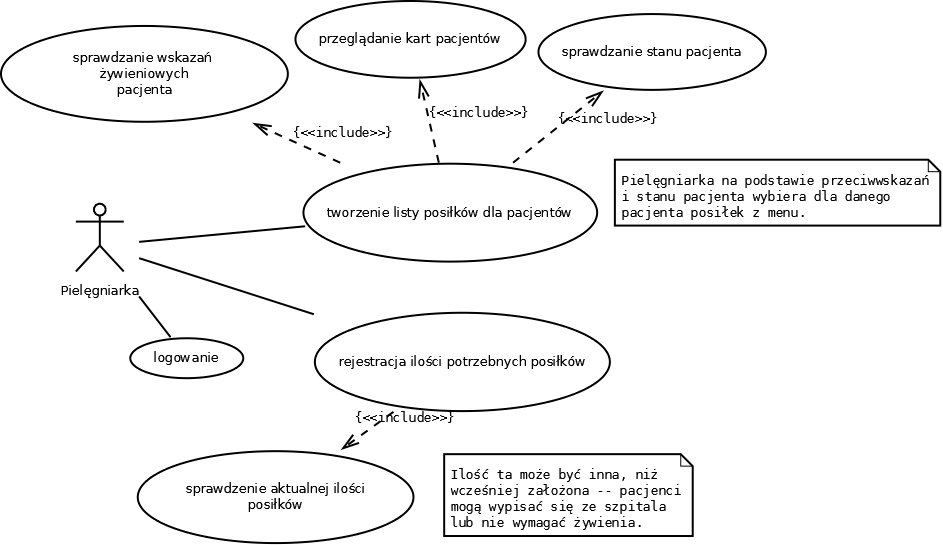
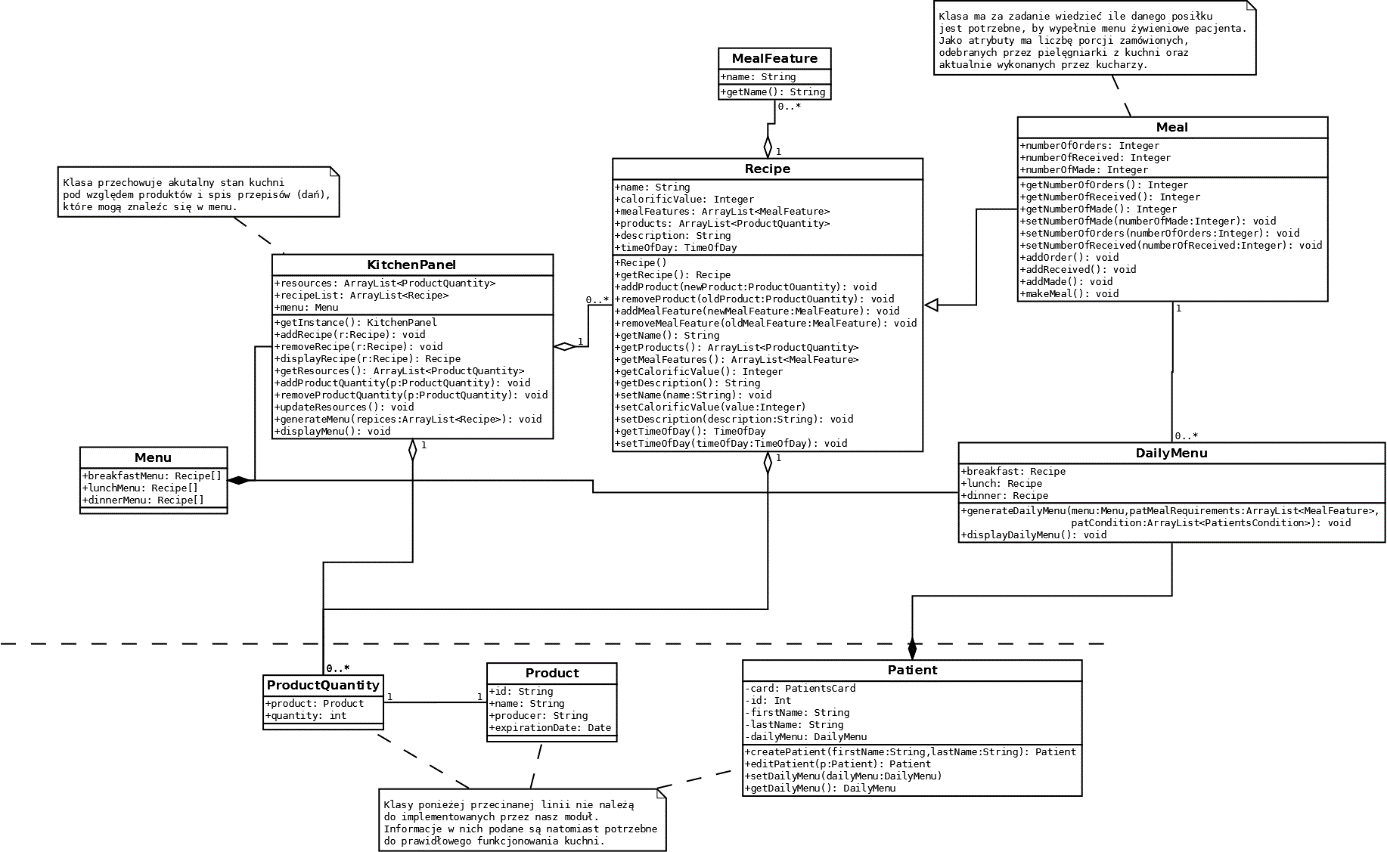


Diagram przypadków użycia dla pielęgniarki pokazuje, że może ona na podstawie kart pacjentów i ich stanu wybierać dla nich posiłki na podstawie wygenerowanego menu. Przekazuje również zapotrzebowanie na dany posiłek.

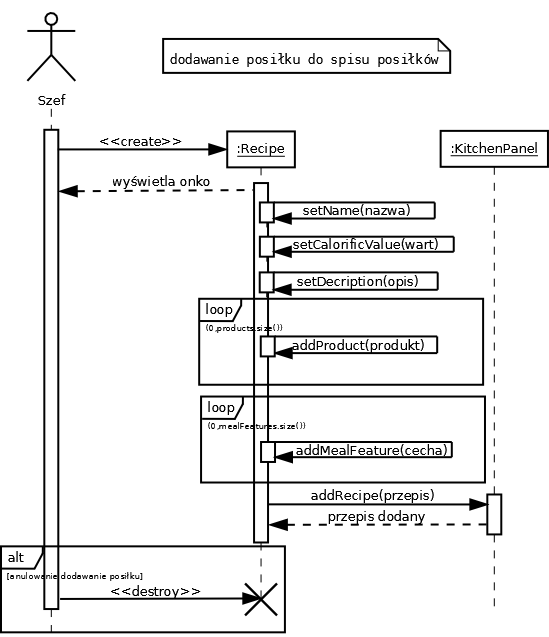
Diagram klas



* Klasa *KitchenPanel* przechowuje aktualny stan zapasów kuchni, dzienne menu i listę wszystkich dostępnych w bazie posiłków oraz posiada metody potrzebne do zarządzania tymi danymi.
* Klasa *Menu* to 3 tablice przechowujące po parę przepisów na każdy z posiłków w ciągu dnia.
* Klasa *Recipe* jest odzwierciedleniem przepisu. Obiekt tej klasy ma m.in. nazwę, wartość kaloryczną czy listę produktów potrzebnych do wykonania danego dania.
* Klasa *Meal* dziedziczy po klasie *Recipe* i jest to zestawienie, zapotrzebowanie na dany dzień poszczególnych posiłków.
* Klasa *DailyMenu* ma jedynie odpowiednie pola dla posiłku w ciągu dnia, w które wpisuje wybrane pola z tablicy klasy *Menu*.
* Klasy *Patient*, *Product* i *ProductQuantity* nie należą bezpośrednio do naszego modułu. Korzystamy jedynie z pewnych ich funkcjonalności, dlatego zdecydowałyśmy się umieścić je na diagramie oddzielając przerywaną linią.

Diagramy interakcji

Diagram sekwencji dla przypadku użycia "dodawanie posiłku do spisu posiłków"



Szef kuchni uruchamiając tę funkcjonalność tworzy obiekt klasy Recipe. W oknie ustawia jego wartości używając klasowych setterów. Po wypełnieniu wszystkich pól obiekt dodawany jest do listy znajdującej się w klasie KitchenPanel. Anulowanie działania oznacza usunięcie utworzonego obiektu.

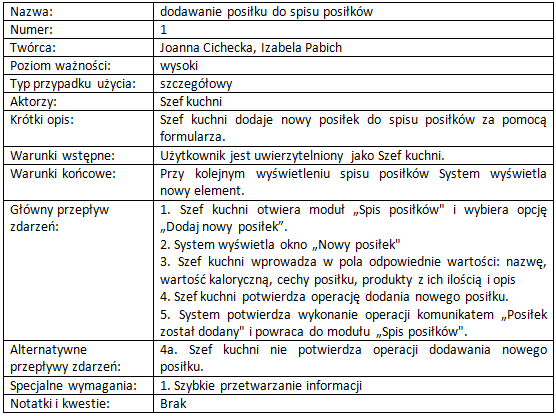
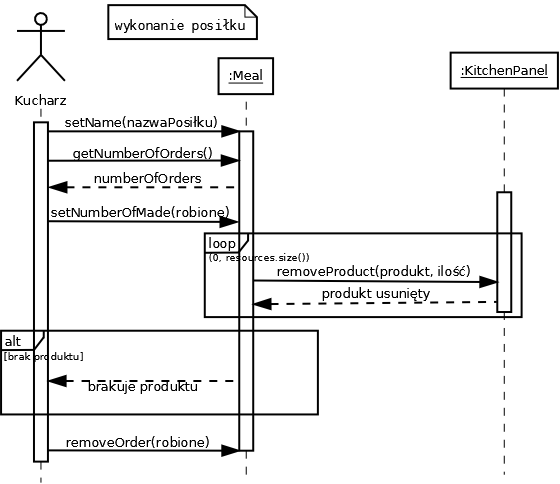
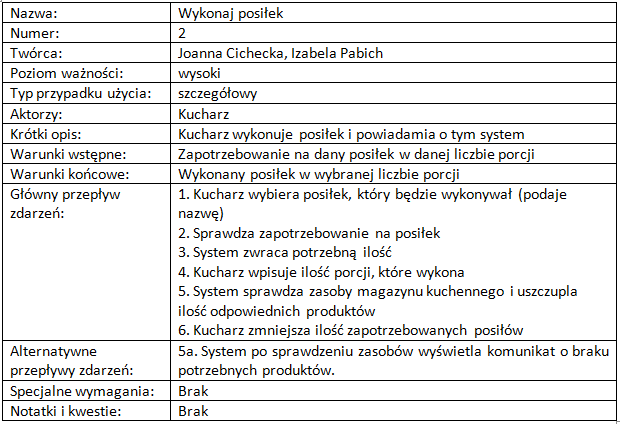


Diagram sekwencji dla przypadku użycia "wykonanie posiłku"

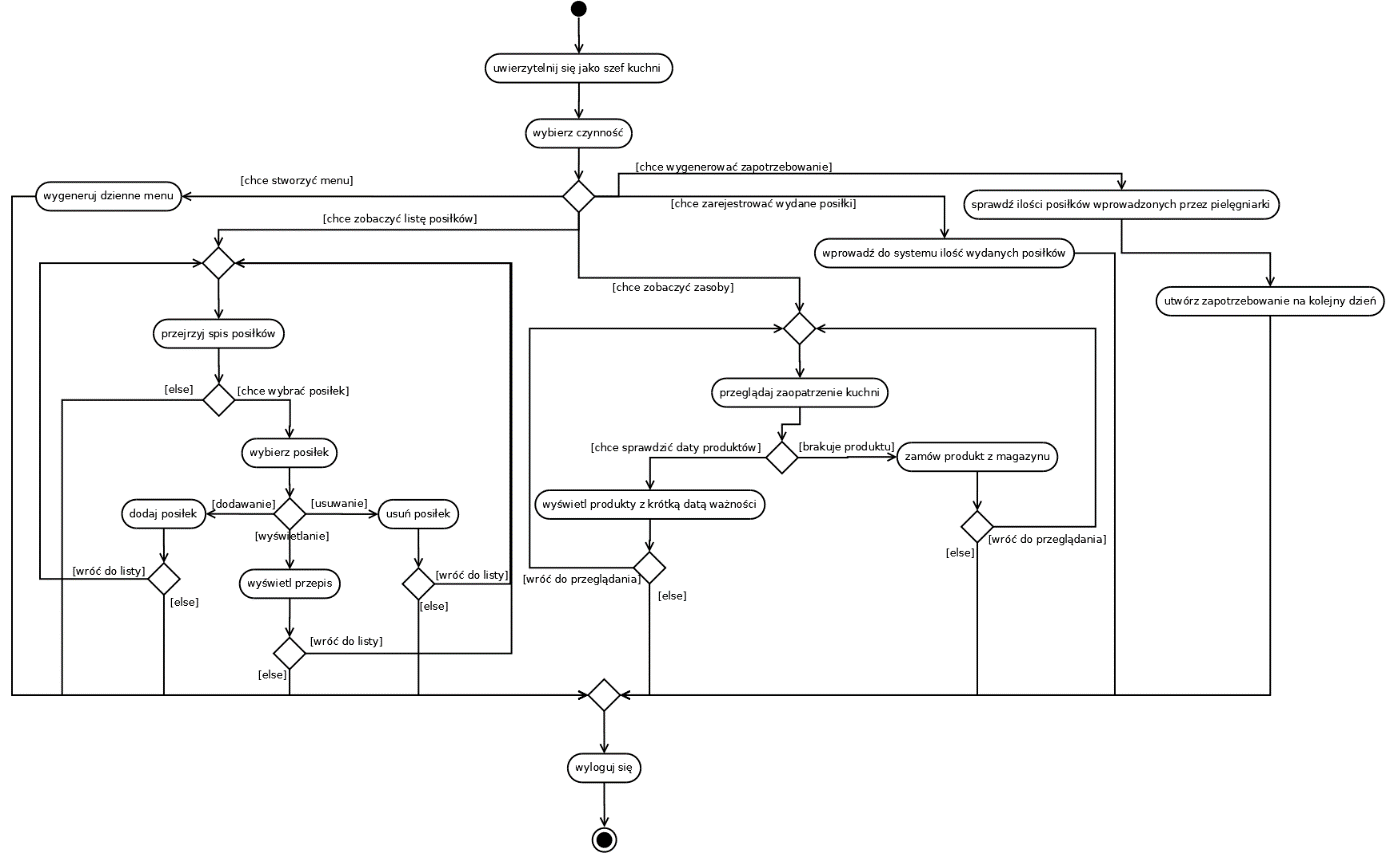


Kucharz na początku sprawdza zapotrzebowanie na dany posiłek podając jego nazwę. Klasa Meal zwraca wartości pola getNumerOfOrders. Następnie kucharz wprowadza uaktualnioną wartość posiłków zrobionych. Operacja ta zmniejsza również zasoby w kuchni.

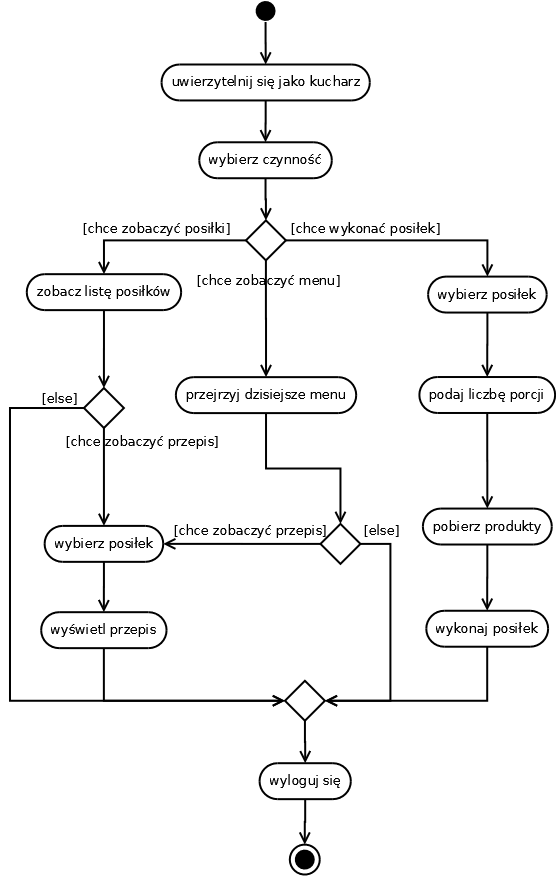


Diagramy czynności

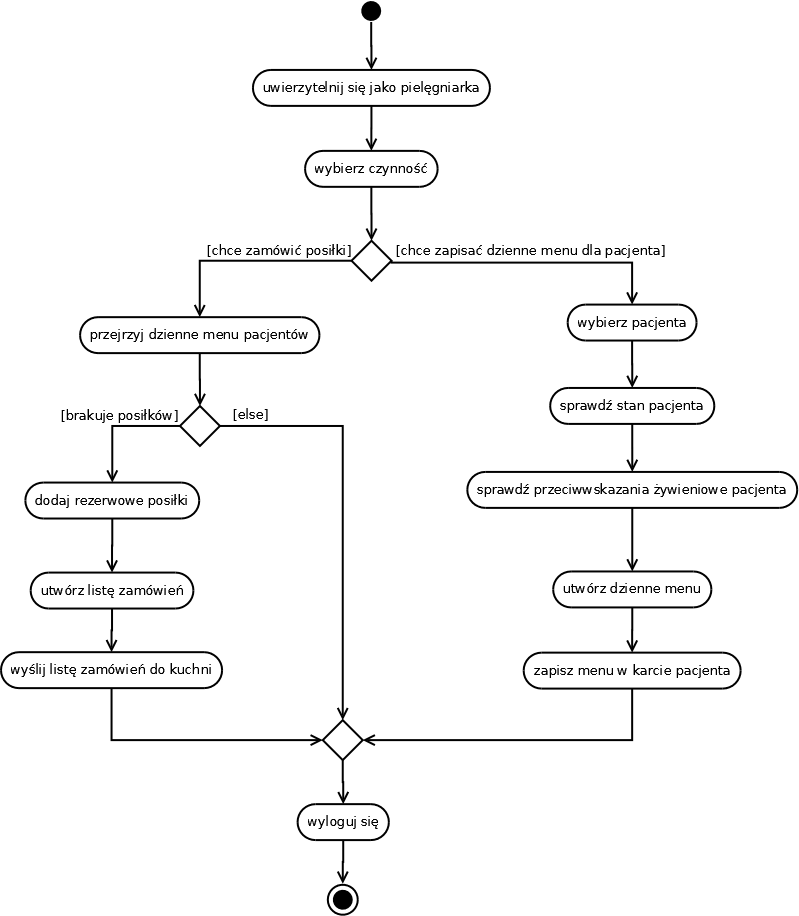
Diagramy czynności podzielone ze względu na użytkowników modułu. Ukazują możliwości jakie mają poszczególni użytkownicy.



Pierwszy diagram czynności przedstawia możliwości użytkownika zalogowanego jako szef kuchni. Kucharz odpowiedzialny jest za generowanie menu, z którego potem korzystają pielęgniarki. Dzięki funkcjonalności wyszukiwania produktów z krótką datą przydatności, może on optymalnie wybrać dania do menu. Dodaje, edytuje i usuwa przepisy z bazy. Również on kontroluje zasoby w kuchni i zamawia brakujące produkty.



Powyższy diagram opisuje czynności, które wykonywać może użytkownik uwierzytelniony jako kucharz, czyli przygotowanie posiłku, zobaczenie dzisiejszego menu czy też spisu wszystkich przepisów.



Ostatni z diagramów opisuję drogi czynności możliwych do zrealizowania przez pielęgniarkę. Może ona zapisać pacjentowi dzienne menu oraz przekazań te wybory do kuchni.

Diagramy maszyny stanowej

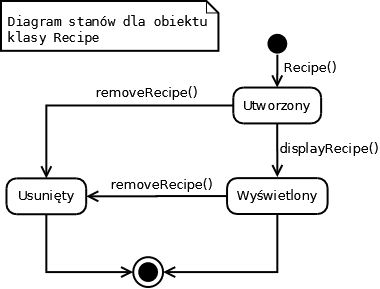
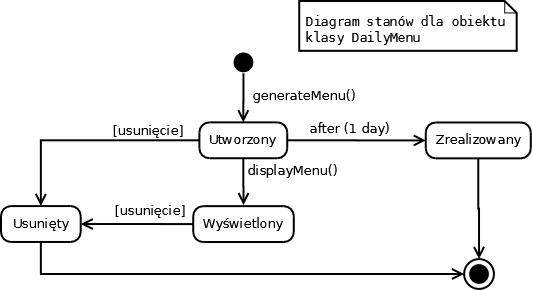


Diagram przedstawia wszystkie stany, w których może się znaleźć obiekt klasy Recipe. Po wywołaniu konstruktora obiekt zostaje utworzony. Można obiekt wyświetlić. Z obu stanów możliwe jest usunięcie obiektu przez co stan zmienia się na usunięty.



Powyższy diagram ukazuje stany, w których znaleźć może się obiekt klasy DailyMenu. Pielęgniarka, która wybierze dania dla pacjenta uruchamia metodę generateMenu() tworząc obiekt tej klasy. Kiedy minie dzień, dany obiekt zostaje "zrealizowany". Utworzone menu można wyświetlić. Obiekt w tym stanie, bądź bezpośrednio po utworzeniu można usunąć, co także zmienia jego stan.

Diagram komponentów

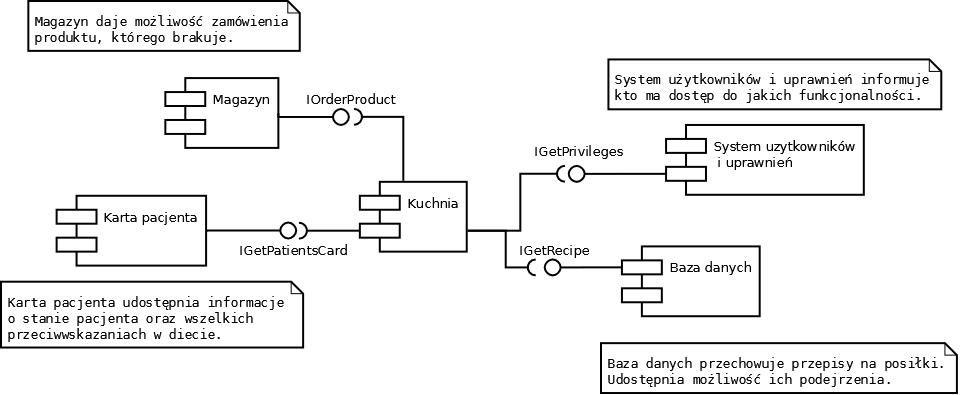


Diagram komponentów przedstawia połączone ze sobą za pomocą interfejsów moduły. Moduł Kuchni współpracuje z czterema innymi modułami w całym systemie:

* Magazyn - udostępniając interfejs IOrderProduct zezwala na wprowadzanie zamówień, które to muszą zostać zrealizowane i dostarczone do Kuchni. Nasz moduł korzysta z tej opcji w sytuacji braku danego produktu w zasobach kuchennych.
* Karta pacjenta - poprzez interfejs IGetPatientsCard pielęgniarki posiadają informację o aktualnym stanie pacjenta oraz jego przeciwwskazaniach w diecie na podstawie, których układają im menu.
* System użytkowników i uprawnień - na bieżąco po zalogowaniu sprawdzane są przywileje jakie dany użytkownik posiada z pośród przypasanych do niego praw. Dzięki rozwiązaniu, które implementuje ten moduł, nie ma prawa zajść sytuacja naruszenia praw przez danego pracownika szpital.
* Baza danych - to ten moduł przechowuje wszelkie informację, które potrzebne są do nawet najmniejszych czynności (np. wyświetlenie przepisu). Dzięki odpowiednim tabelkom w bazie szybko wyszukujemy potrzebne informacje.

Diagram przeglądu interakcji

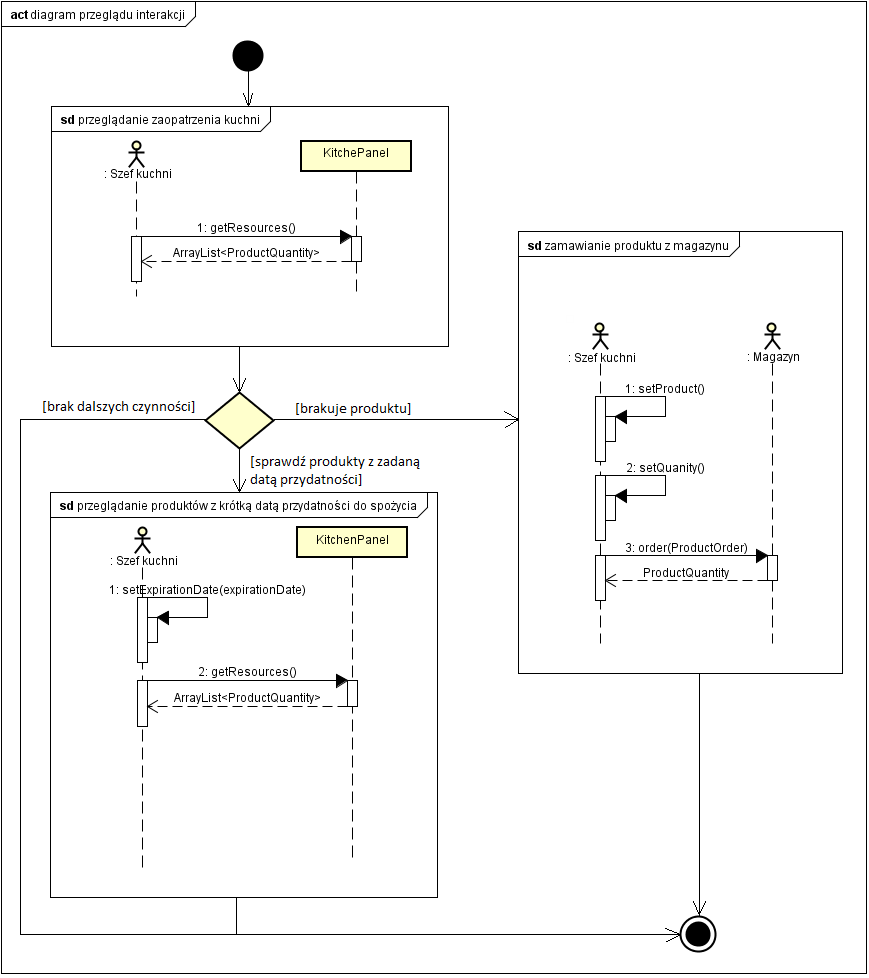
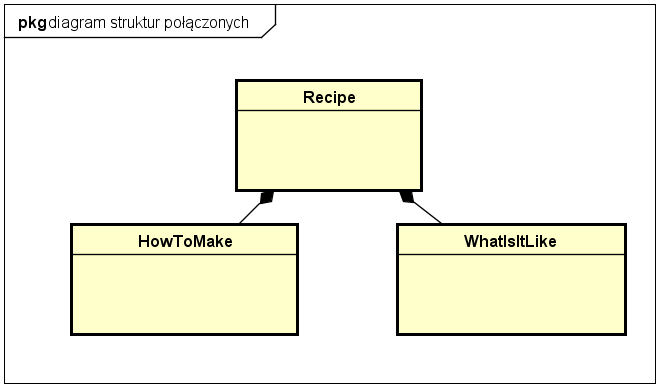


Diagram przeglądu interakcji obrazuje przepływ sterowania pokazuje współpracę między diagramami sekwencji dla wybranego przypadku użycia.

Diagram strukturalny



Powyższy diagram pokazuje z jakiś części składa się klasa Recipe. Nie było potrzeby tworzenia osobnych klas, ale są one warte wyróżnienia. Część HowToMake składa się z opisu dania, składników, a druga część WhatIsItLike przechowuje informacje o wartościach odżywczych i cechach posiłku.