

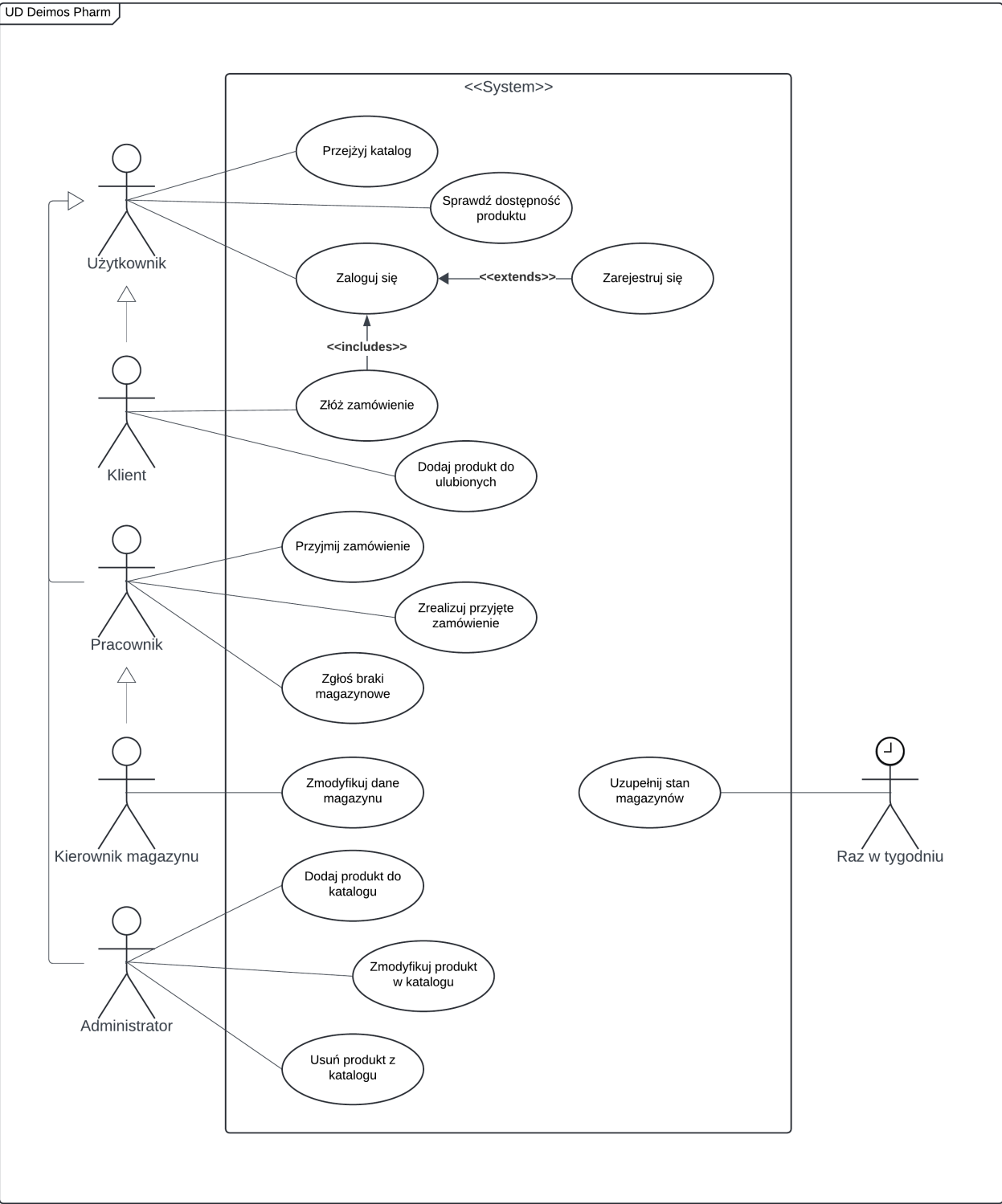
Deimos Pharm - System komputerowy koncernu farmaceutycznego

1. Wymagania:

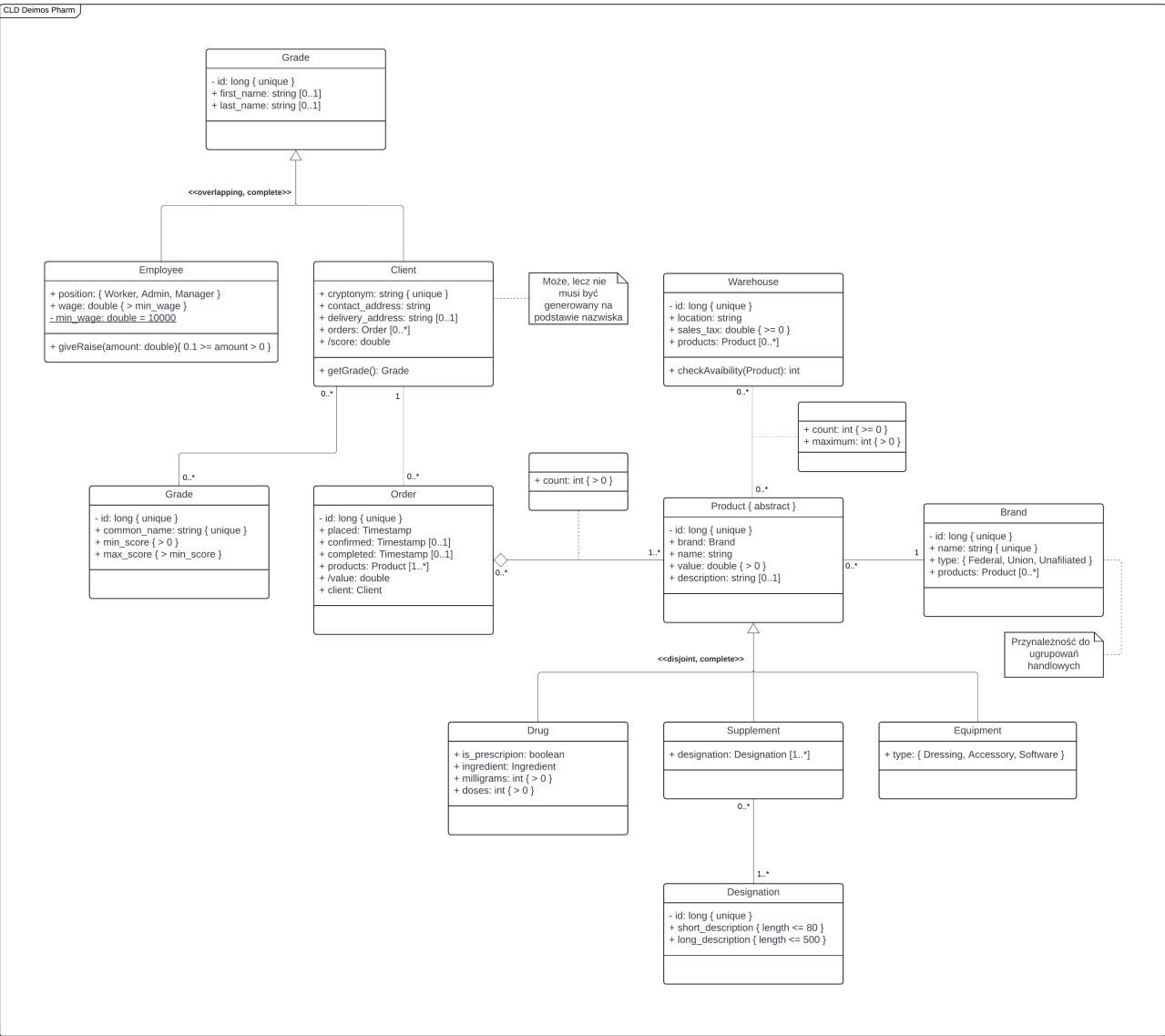
Deimos Pharm to międzyplanetarna megakorporacja zajmująca się importem i dystrybucją produktów medycznych wszelkiej maści. Z uwagi na wysoki popyt, firma rozrosła się ponad skalę którą wydajnie może administrować stary, generyczny program. Zarząd zdecydował się zlecić stworzenie nowego systemu, który będzie w stanie sprostać wymaganiom administracyjnym prężnie rozwijającego się biznesu.

1. System przechowuje informacje o produktach oferowanych przez firmę. Dla każdego z nich należy przechować informację o producencie, nazwie produktu, cenie jednostkowej netto, oraz opcjonalnie opisu (maksymalnie 500 znaków). Nazwy producentów muszą być unikalne na poziomie systemu, a nazwy leków na poziomie portfolio danego producenta.
2. Produkty dzielą się na leki, suplementy diety, oraz elementy wyposażenia. Dla leków należy pamiętać dodatkowo substancję aktywną i jej gramaturę, ilość porcji w opakowaniu, a także to czy do jego zakupu wymagana jest recepta. Dla suplementów należy przechowywać informację o przeznaczeniu suplementu (co najmniej jedno, choć może być więcej). Dla elementów wyposażenia natomiast chcemy znać ich typ (opatrunek, narzędzie, akcesorium, produkt elektroniczny).
3. System powinien umożliwiać administrację magazynów poszczególnych placówek na przestrzeni układu gwiazdowego. Dla każdego z magazynów chcemy znać listę dostępnych produktów, ilość na stanie każdego z nich, oraz podatek lokalny, który należy doliczyć do ceny netto każdego sprzedawanego tam produktu.
4. Do magazynów raz w tygodniu dostarczana jest dostawa uzupełniająca wyprzedane produkty. Dla każdego produktu którego stan jest niższy niż 20%, dostarczana jest partia w wysokości 50% maksymalnego stanu który jest w stanie pomieścić magazyn.
5. W systemie należy także zapewnić miejsce na dane zarejestrowanych klientów: unikalny kryptonim, który może ale nie musi zawierać imienia i nazwiska klienta, adres kontaktowy, opcjonalny adres zamieszkania używany przy dostawach zamówień, a także tzw. "Grade", będący pozycją na skali lojalności klienta, wyliczaną na podstawie wartości przeszłych zamówień.
6. System powinien zapewniać użytkownikom dostęp do następujących funkcjonalności:
 - 6.1. Przeglądanie katalogu produktów (każdy użytkownik)
 - 6.2. Sprawdzenie dostępności danego produktu w magazynach (każdy użytkownik)
 - 6.3. Dodanie produktu do listy ulubionych (zarejestrowany klient)
 - 6.4. Złożenie zamówienia (zarejestrowany klient)
 - 6.5. Przyjęcie zamówienia (pracownik, kierownik magazynu)
 - 6.6. Realizacja zamówienia (pracownik, kierownik magazynu)
 - 6.7. Zgłoszenie braków magazynowych (pracownik)
 - 6.8. Dodanie nowego produktu (administrator)
 - 6.9. Modyfikacja produktu (administrator)
 - 6.10. Usunięcie produktu (administrator)
 - 6.11. Modyfikacja danych magazynu (kierownik magazynu)

Use Case Diagram



Class Diagram



Przypadek użycia: Złożenie zamówienia

Opis przypadku użycia

- **Uczestnicy:** Klient sklepu
- **Początek:** Klient rozpoczyna sesję w serwisie sklepu internetowego
- **Koniec:** Klient składa poprawne zamówienie

Plan zdarzeń:

1. System sprawdza czy klient rzeczywiście jest zalogowany
 - 1.1. Jeżeli nie jest, system przekierowuje go na stronę logowania. Przypadek użycia kończy się
2. System oczekuje na dodanie produktu do koszyka
3. Klient dodaje produkt do koszyka
4. System prosi o wybranie następnej akcji
 - 4.1. Jeżeli klient wybierze kontynuację zakupów, powrót do **2**.
5. Klient wybiera przejście do ekranu koszyka
6. System podlicza wartość brutto koszyka i prezentuje ją klientowi
7. System prosi o wybór dalszej akcji
 - 7.1. Jeżeli klient wybierze powrót do sklepu, powrót do **2**.
8. Klient wybiera złożenie zamówienia
9. System prosi o wybranie sposobu dostawy
 - 9.1. Jeżeli klient wybierze dostawę na adres, system sprawdza czy takowy znajduje się w systemie
 - 9.1.1. W systemie brak adresu, system prosi o jego podanie.
 - 9.1.2. Klient podaje adres, przejdź do **11**.
 - 9.2. Adres jest w systemie, przejdź do **11**.
10. Klient wybiera odbiór osobisty
11. System przekierowuje klienta do platformy płatności. Przypadek użycia kończy się.

Diagram przypadku użycia

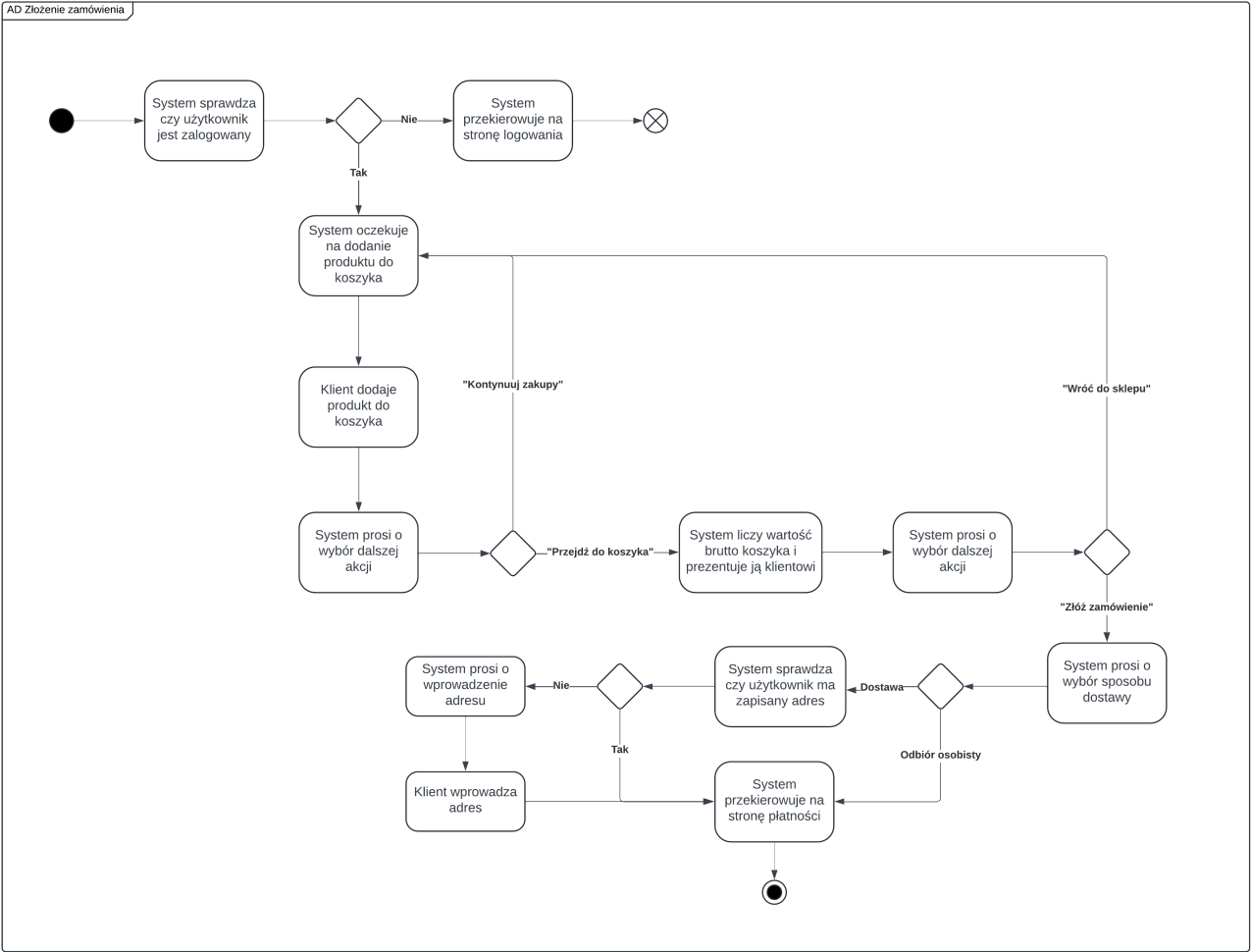
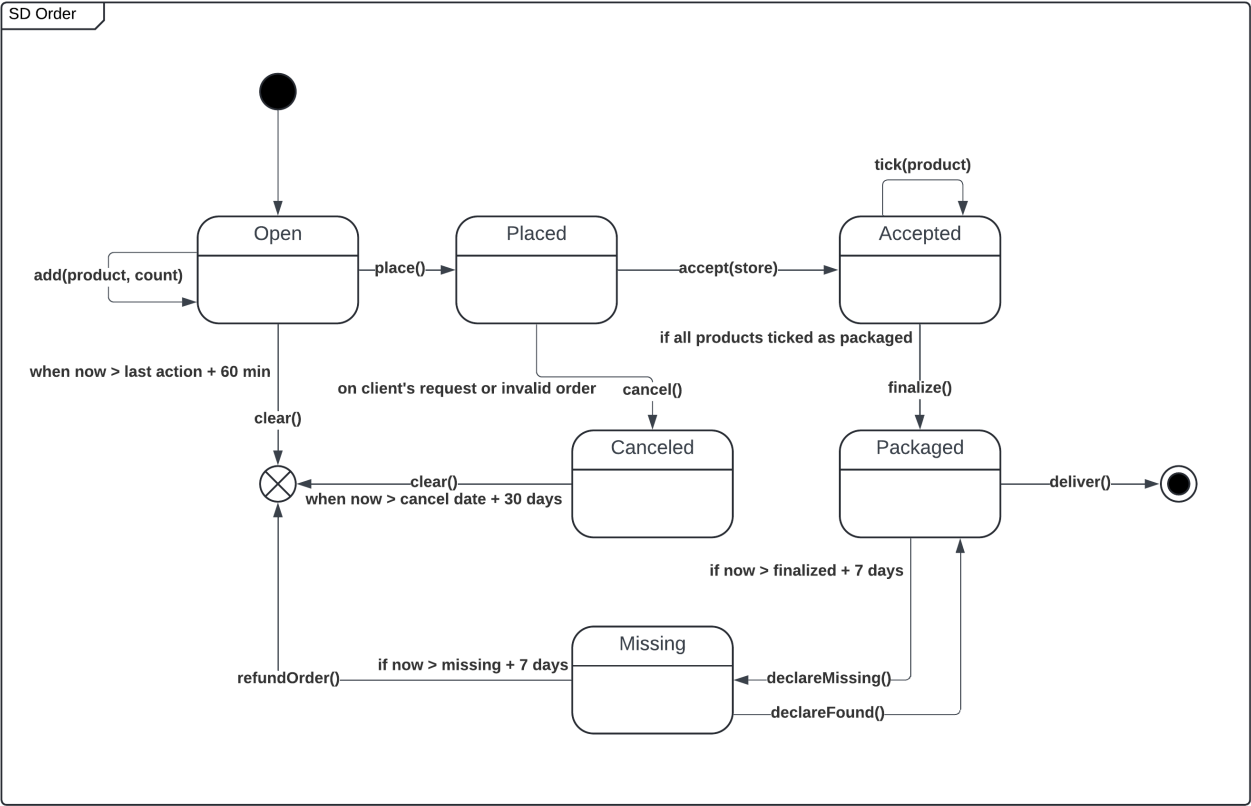
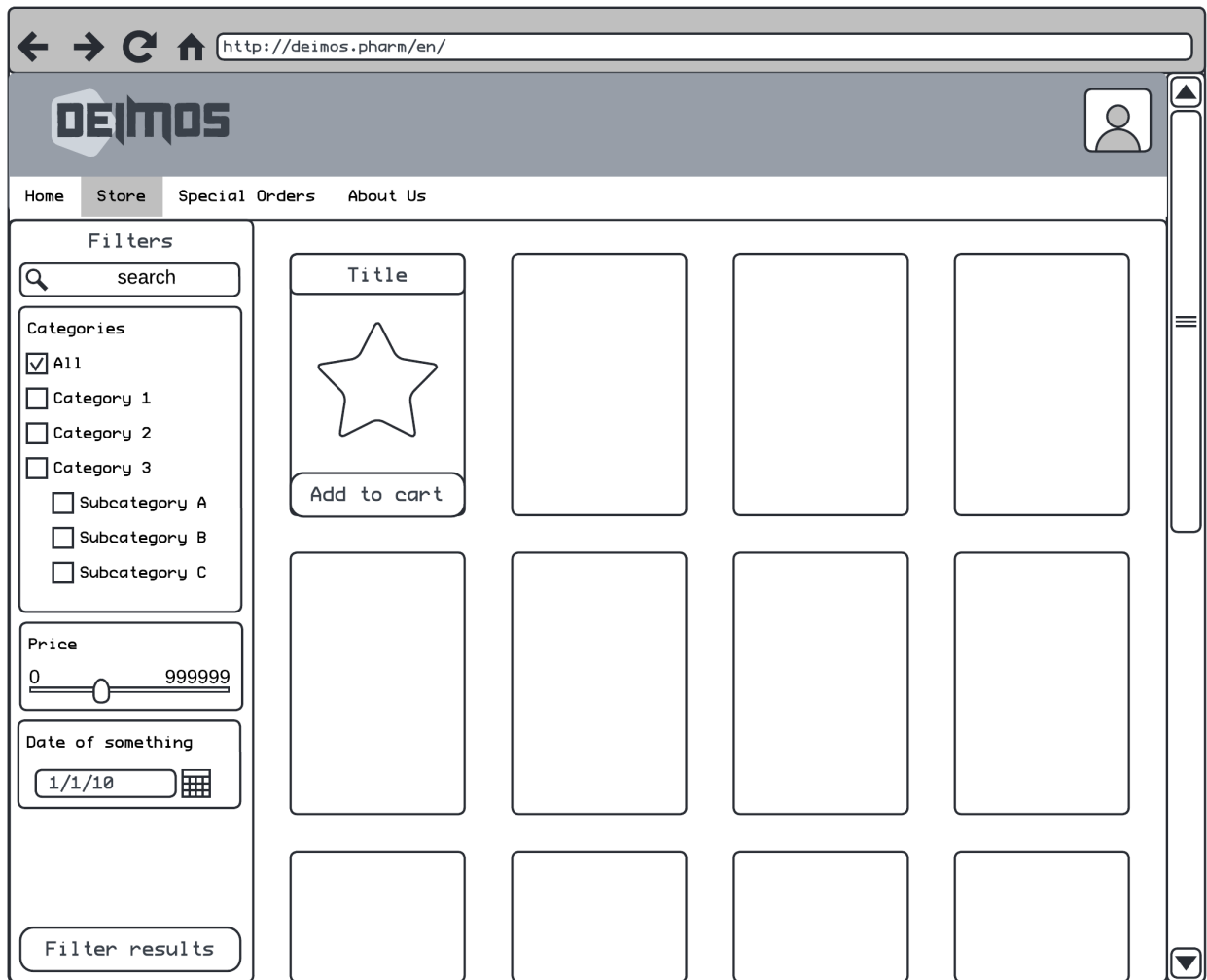


Diagram stanów dla przypadku użycia

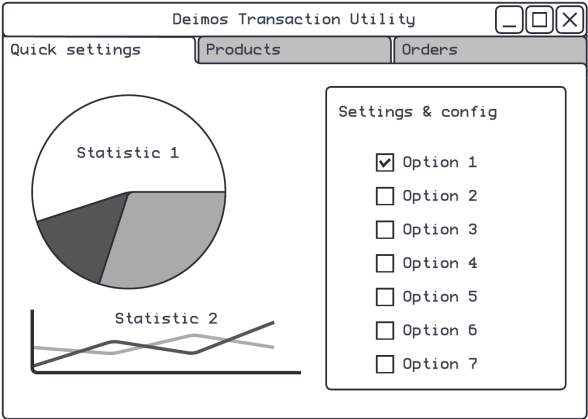


Prototypy interfejsów użytkownika

Serwis internetowy



Panel pracownika



Deimos Transaction Utility


Quick settings | Products | Orders

Name	Amount	Thing one	Thing two
Something	57	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
Something else	27	<input type="checkbox"/>	<input type="radio"/>
Yet another thing	25	<input checked="" type="checkbox"/>	<input type="radio"/>


Deimos Transaction Utility

Quick settings | Products | Orders

Statistic 1




Statistic 2



Settings & config

- ion 1
- ion 2
- ion 3
- ion 4
- ion 5
- ion 6
- ion 7



Decyzje projektowe:

- Podstawową technologią systemu jest język programowania Java 22. Została wybrana ze względu na stabilność, elastyczność, oraz doświadczenie zespołu z tym językiem.
- Główna baza danych do przechowywania istotnych danych biznesowych powstanie z użyciem standardu SQLite. Został on wybrany ze względu na prostotę integracji z systemem ORM i mobilność bazy.
- Do komunikacji z bazą danych wykorzystana zostanie biblioteka ORMLite, ze względu na prostotę integracji z resztą technologii, oraz potencjalnej rozbudowy.
- Jako biblioteka graficznego interfejsu użytkownika wybrana została sprawdzona Java Swing. Powodem wybrania jej ponad bardziej nowoczesne rozwiązania takie jak chociażby Vaadin jest doświadczenie dewelopera z tą platformą. Biorąc pod uwagę charakter projektu i ograniczenia czasowe, rozpoczynanie pracy z nową technologią byłoby czasochłonne i ryzykowne. Oprócz frameworku Swing, do tworzenia prostych interfejsów debugowych i skryptów automatycznych, wykorzystana zostanie biblioteka JLine.

Diagram Projektowy - z uwzględnieniem rozwiązań technicznych

