



Akademia  
Humanistyczno  
Ekonomiczna  
w Łodzi

## DOKUMENTACJA PROJEKTOWA SYSTEMU INFORMATYCZNEGO

**Temat:** System Zarządzania Jakością i Śledzenia Partii (Traceability) w Branży Farmaceutycznej

**Autor:** Piotr Kryziński

**Technologia:** Python / Django

### 1. WSTĘP I CEL PROJEKTU

Celem pracy było zaprojektowanie i implementacja systemu informatycznego klasy **Track & Trace**, dedykowanego dla branży farmaceutycznej. Głównym zadaniem aplikacji jest cyfryzacja procesów magazynowych oraz zapewnienie pełnej transparentności cyklu życia produktu leczniczego – od przyjęcia surowców, przez proces produkcji, aż po dystrybucję wyrobu gotowego.

Projekt odpowiada na rygorystyczne wymagania norm **GMP (Good Manufacturing Practice)**, które nakładają na wytwórców leków obowiązek posiadania możliwości natychmiastowego odtworzenia historii każdej serii leku (genealogia partii). System rozwiązuje problem rozproszonych danych i umożliwia wygenerowanie raportu w czasie rzeczywistym, co jest kluczowe w sytuacjach kryzysowych (np. konieczność wycofania leku z rynku).

## ROZDZIAŁ 2. ANALIZA PRZEDMIOTOWA I WYMAGANIA SYSTEMU

### 2.1. Charakterystyka Użytkownika Docelowego (Firma)

Projektowany system dedykowany jest dla przedsiębiorstwa produkcyjnego z branży farmaceutycznej: **"Pol-Pharm Sp. z o.o."** (nazwa przykładowa).

Firma zajmuje się produkcją niesterylnych form leków (tabletek, syropów) i podlega rygorystycznym przepisom Prawa Farmaceutycznego. Obecnie proces ewidencji partii produkcyjnych prowadzony jest w formie papierowej oraz w arkuszach kalkulacyjnych, co generuje ryzyko błędów i wydłuża czas dostępu do informacji.

### **Schemat Organizacyjny Przedsiębiorstwa**

Struktura organizacyjna firmy, w której wdrożony zostanie system, przedstawia się następująco:

- **Zarząd**
  - **Dział IT** (Administratorzy Systemu)
  - **Dział Produkcji**
    - Kierownik Produkcji (Nadzór nad zużyciem)
    - Magazyn Surowców (Przyjęcia materiałowe)
  - **Dział Zapewnienia Jakości (QA/QC)**
    - Kontrolerzy Jakości (Zwalnianie partii, Audyt Traceability)

### **2.2. Definicja Problemu i Kontekst**

Głównym problemem rozwiązywanym przez projekt jest **brak natychmiastowego dostępu do genealogii produktu**. W przypadku wykrycia wady w surowcu (np. zanieczyszczona Laktoza), firma musi ręcznie przeszukiwać setki dokumentów papierowych, aby ustalić, do których serii leków trafił wadliwy składnik.

- **Cel projektu:** Automatyzacja procesu śledzenia przepływu materiałów (Traceability) i skrócenie czasu generowania raportu genealogicznego z kilku godzin do kilku sekund.
- **Procesy objęte systemem:** Przyjęcie materiału, nadanie numeru serii, zmiana statusu jakościowego, rejestracja zużycia produkcyjnego.
- **Dokumenty wynikowe:** Raport Genealogii Partii (PDF/HTML), Raport Stanów Magazynowych.

### **2.3. Charakterystyka Użytkowników i Uprawnienia**

W systemie zidentyfikowano trzy główne grupy użytkowników o zróżnicowanych rolach:

Rola w Systemie	Odpowiednik w firmie	Zakres Uprawnień
<b>Administrator</b>	Dział IT	Pełny dostęp do konfiguracji, zarządzanie użytkownikami, edycja słowników (Katalog Materiałów).
<b>Operator Magazynowy</b>	Magazynier / Produkcja	Tworzenie nowych partii (Batch), rejestracja zużycia (Traceability). Brak możliwości zmiany statusów jakościowych.
<b>Kontroler Jakości (QC)</b>	Dział Jakości	Podgląd raportów, prawo do zmiany statusu partii (Kwarantanna -> Zwolniony), blokowanie wadliwych serii.
<b>Użytkownik (Gość)</b>	Audytory / Zarząd	Dostęp wyłącznie do wyszukiwarki i podglądu raportów (Read-only) na froncie aplikacji.

#### 2.4. Założenia i Zależności (Obiekty Zewnętrzne)

System funkcjonuje w otoczeniu technologicznym i prawnym, które narzuca pewne zależności:

- **Regulacje Prawne:** System musi spełniać wymogi GMP (Good Manufacturing Practice), w tym integralność danych.
- **Infrastruktura:** Aplikacja działa jako system webowy dostępny w sieci wewnętrznej firmy (Intranet).
- **Interakcja z użytkownikiem:** System zakłada obsługę przez przeglądarkę internetową na stacjach roboczych PC oraz tabletach przemysłowych.

#### 2.5. Wymagania Funkcjonalne

System musi realizować następujące funkcje (oznaczone jako **WF** - Wymaganie Funkcjonalne):

- **WF-01 Zarządzanie Katalogiem:** Możliwość dodawania i edycji definicji materiałów (Surowce, Produkty).
- **WF-02 Ewidencja Partii:** Rejestracja nowej partii z polami: Numer LOT, Data Produkcji, Data Ważności.
- **WF-03 Traceability:** Możliwość powiązania partii "Rodzica" z partią "Dziecka" wraz z określeniem ilości.

- **WF-04 Statusy Jakościowe:** Obsługa cyklu życia statusu (Kwarantanna -> Zwolniony -> Odrzucony).
- **WF-05 Wyszukiwanie:** Wyszukiwarka partii po numerze LOT dostępna dla użytkownika końcowego.
- **WF-06 Raportowanie:** Automatyczne generowanie drzewa użycia (wstecz i w przód).

## 2.6. Wymagania Niefunkcjonalne

Ograniczenia i parametry jakościowe systemu (oznaczone jako **WN** - Wymaganie Niefunkcjonalne):

- **WN-01 Język:** Interfejs użytkownika w języku polskim.
- **WN-02 Czas reakcji:** Wygenerowanie raportu genealogicznego dla 1000 rekordów nie może przekraczać 2 sekund.
- **WN-03 Bezpieczeństwo:** Ochrona przed atakami typu SQL Injection (zapewniana przez Django ORM).
- **WN-04 Dostępność:** System dostępny 24/7 poprzez przeglądarkę WWW, bez konieczności instalacji klienta.
- **WN-05 Spójność danych:** Zastosowanie transakcyjności bazy danych (ACID).

## 2.7. Projekt Koncepcyjny (Przepływ Logiczny)

System opiera się na centralnej bazie danych. Logika przepływu informacji wygląda następująco:

1. **Input:** Użytkownik wprowadza dane o przyjęciu surowca (powstaje obiekt Batch).
2. **Process:** Podczas produkcji operator łączy surowiec z produktem (powstaje obiekt Movement).
3. **Logic:** System waliduje statusy (tylko zwolniony surowiec powinien być użyty - *funkcja w rozwoju*).
4. **Output:** Na żądanie audytora system rekurencyjnie odpytuje tabelę powiązań i zwraca wizualizację graficzną genealogii.

## 3. ANALIZA FUNKCJONALNA I ZASADA DZIAŁANIA

System został zaprojektowany w architekturze klient-serwer i składa się z dwóch głównych modułów interfejsu: Panelu Administracyjnego (Back-office) oraz Panelu

Raportowego (Frontend). Poniżej przedstawiono szczegółowy scenariusz działania systemu.

### 3.1. Etap I: Definicja Katalogowa (Słowniki)

Fundamentem systemu jest baza danych materiałowych. Administrator systemu definiuje obiekty w katalogu, przypisując im unikalne kody SKU, nazwy oraz typy (Surowiec, Opakowanie, Wyrób Gotowy). Zapobiega to błędom polegającym na przyjęciu do magazynu towaru niezdefiniowanego.

### 3.2. Etap II: Ewidencja Partii i Zarządzanie Jakością

System rozróżnia definicję materiału od jego fizycznej partii (**Batch**). Podczas przyjęcia towaru lub zakończenia produkcji tworzona jest nowa partia, która otrzymuje:

- Unikalny numer serii (LOT).
- Datę produkcji i datę ważności.
- Status jakościowy.

Domyślnie każda nowa partia otrzymuje status „**Kwarantanna**”. System umożliwia osobie uprawnionej zmianę statusu na „Zwolniony” lub „Odrzucony”, co odwzorowuje proces decyzyjny Działu Kontroli Jakości.

### 3.3. Etap III: Rejestracja Procesu (Traceability)

Kluczową funkcjonalnością jest rejestracja zużycia materiałów. Użytkownik tworzy w systemie powiązanie, wskazując:

1. Partię źródłową (Składnik).
2. Partię docelową (Produkt).
3. Dokładną ilość zużytego surowca.

Dzięki temu tworzona jest cyfrowa „nić” łącząca surowce z wyrobem gotowym, umożliwiającą późniejsze bilansowanie produkcji.

### 3.4. Etap IV: Raportowanie i Wyszukiwanie

Dla użytkownika końcowego przygotowano dedykowany interfejs wyszukiwania. Po wprowadzeniu numeru serii, system generuje raport zawierający dwa kluczowe zestawienia:

- **Genealogia Wstecz (Backward):** Lista składników użytych do produkcji danej serii.
- **Genealogia W Przód (Forward):** Informacja o dalszym wykorzystaniu danej partii.

## 4. REALIZACJA TECHNICZNA I IMPLEMENTACJA

System został zrealizowany w oparciu o nowoczesny stos technologiczny, zapewniający bezpieczeństwo danych i skalowalność.

### 4.1. Środowisko i Technologie

- **Język programowania:** Python 3.11 – wybrany ze względu na czytelność kodu i wsparcie bibliotek.
- **Framework:** Django 5.2.9 – wykorzystany do budowy szkieletu aplikacji, obsługi zapytań HTTP oraz zabezpieczenia przed atakami (SQL Injection, XSS).
- **Baza danych:** SQLite – relacyjna baza danych zapewniająca integralność transakcyjną.
- **Frontend:** HTML5, CSS3, Bootstrap 5.

### 4.2. Architektura Danych (Backend)

W pliku models.py zaimplementowano trzy kluczowe modele danych odwzorowujące procesy biznesowe:

1. **Model Material:** Reprezentuje słownik materiałowy. Wykorzystuje pole wyboru (choices) do typowania materiałów.
2. **Model Batch:** Reprezentuje fizyczną partię. Posiada klucz obcy (ForeignKey) do modelu Material, co wymusza spójność danych.
3. **Model Movement:** Tabela asocjacyjna realizująca relację "wiele do wielu" z atrybutami. Przechowuje informację o przepływie (Źródło -> Cel) oraz ilości (quantity).

### 4.3. Algorytm Śledzenia (Views)

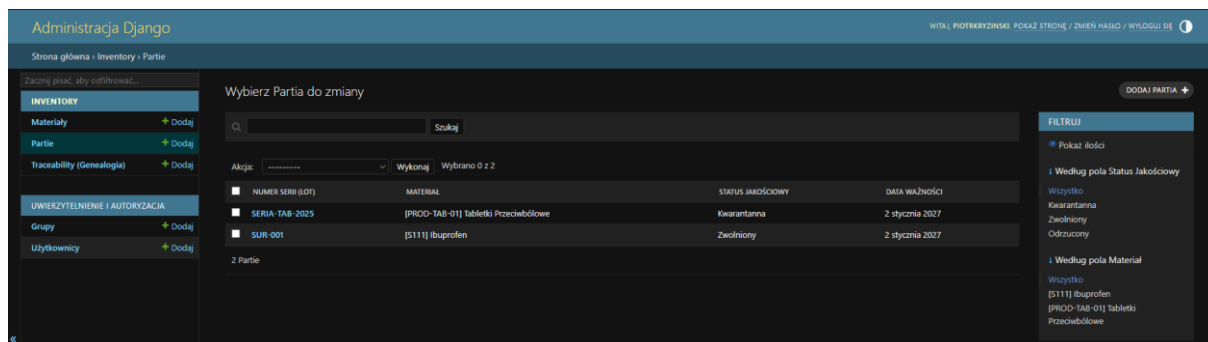
Logika biznesowa została zaimplementowana w pliku views.py. Funkcja traceability\_search realizuje rekurencyjne przeszukiwanie bazy danych w dwóch kierunkach, identyfikując powiązania w tabeli Movement na podstawie numeru serii przekazanego w zapytaniu GET.

## 5. PREZENTACJA DZIAŁANIA APLIKACJI (Zrzuty Ekranowe)

W niniejszym rozdziale przedstawiono widok rzeczywistego działania systemu na przykładzie ścieżki produkcyjnej leku "Tabletki Przeciwbólowe".

## 5.1. Panel Administracyjny - Lista Partii

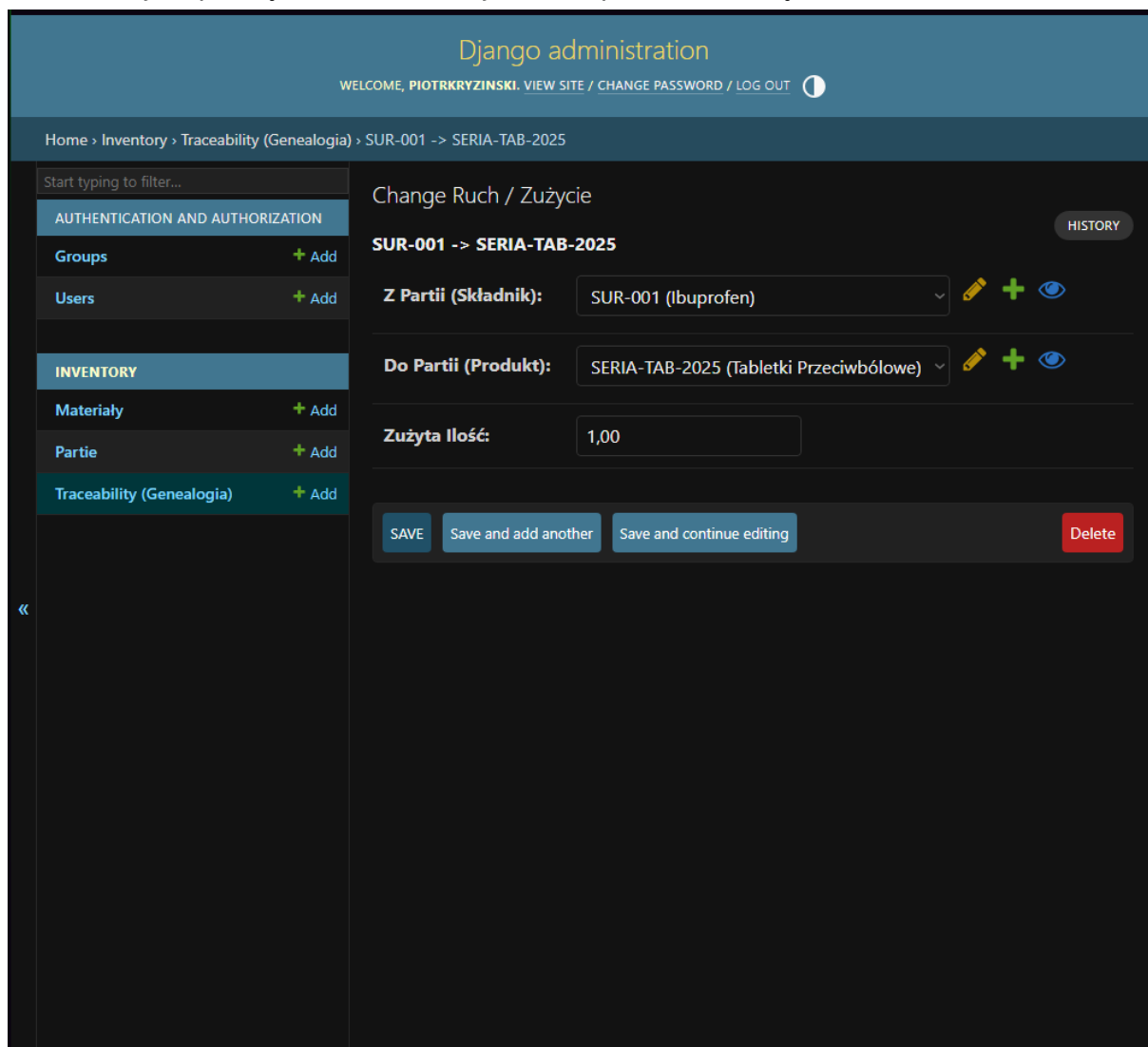
Widok przedstawia zaewidencjonowane partie surowców (Ibuprofen) oraz wyrobów gotowych wraz z ich statusami jakościowymi.



Rysunek 1 Panel zarządzania partiami produkcyjnymi z widocznymi statusami jakościowymi oraz filtrowaniem danych.

## 5.2. Tworzenie Relacji Produkcyjnej

Formularz służący do rejestracji zużycia surowca. Widoczne jest powiązanie konkretnej serii Ibuprofenu z serią Tabletek



Rysunek 2 Formularz rejestracji zużycia surowców (Traceability) – tworzenie powiązania między partią źródłową a docelową.

### 5.3. Raport Końcowy Traceability

Widok interfejsu dla użytkownika końcowego. Po wyszukaniu serii **SERIA-TAB-2025**, system prezentuje jej status, daty ważności oraz pełną genealogię (użyty surowiec Ibuprofen).

Rysunek 3 Widok raportu końcowego dla użytkownika prezentujący pełną genealogię partii oraz status kwarantanny.

### 6. Dalszy Rozwój

W obecnej wersji system pozwala na ręczną zmianę statusów. W ramach dalszego rozwoju planowane jest zaimplementowanie blokady logicznej, która uniemożliwiłaby zużycie surowca o statusie 'Kwarantanna' lub 'Odrzucony'. Dodatkowo system można rozbudować o moduł generowania etykiet z kodami QR, co przyspieszyłoby wprowadzanie danych przy użyciu skanerów magazynowych.

### PODSUMOWANIE

Zrealizowany projekt stanowi w pełni funkcjonalny prototyp systemu klasy Traceability. Aplikacja poprawnie realizuje założenia projektowe, umożliwiając identyfikację i śledzenie przepływu materiałów w łańcuchu dostaw. Zastosowanie architektury opartej na relacyjnej bazie danych gwarantuje spójność informacji, co jest krytycznym wymogiem w systemach klasy farmaceutycznej.