**1. Wzorzec Fabryka Abstrakcyjna (Abstract Factory)**

**1.1 Tytuł:** Fabryka Abstrakcyjna (Abstract Factory)

**1.2 Motywacja:** W projekcie istniała potrzeba tworzenia powiązanych ze sobą obiektów reprezentujących różne typy użytkowników (np. Seller, Buyer) oraz potencjalnie powiązanych z nimi zasobów. Bezpośrednie tworzenie tych obiektów w kodzie klienckim prowadziło do silnego powiązania i utrudniało wprowadzanie zmian oraz dodawanie nowych typów użytkowników lub zasobów. Zastosowanie wzorca Fabryka Abstrakcyjna pozwoliło na:

* + **Oddzielenie tworzenia obiektów od ich użycia:** Klient operuje na abstrakcyjnych interfejsach fabryk i produktów, nie znając konkretnych implementacji.
  + **Łatwe zarządzanie rodzinami obiektów:** Umożliwia tworzenie spójnych zestawów obiektów (np. wszystko związane z administratorem lub zwykłym użytkownikiem).
  + **Zwiększenie elastyczności:** Ułatwia dodawanie nowych typów produktów lub całych rodzin produktów bez modyfikacji kodu klienta.
  + **Konsekwencje:** Wprowadzenie dodatkowych klas i interfejsów (fabryki abstrakcyjne, konkretne fabryki, abstrakcyjne produkty), co może nieznacznie zwiększyć złożoność struktury, ale znacząco poprawia organizację i elastyczność kodu.

**1.3 Fragment projektu przed zastosowaniem wzorca:**

A diagram of a computer program

AI-generated content may be incorrect.

**1.4 Struktura i elementy po zastosowaniu wzorca:** A diagram of a computer

AI-generated content may be incorrect.**1.5 Współdziałanie po wdrożeniu wzorca:**

**2. Wzorzec Budowniczy (Builder)**

**2.1 Tytuł:** Budowniczy (Builder)

**2.2 Motywacja:** W projekcie istniała potrzeba tworzenia złożonych obiektów Dispute, tworzenie których składać się z wielu etapów i których proces konstrukcji mógł być skomplikowany lub wymagać wielu kroków. Dodatkowo, mogła istnieć potrzeba tworzenia różnych reprezentacji tego samego Dispute (np. zaakceptowany czy nie) Zastosowanie wzorca Budowniczy pozwoliło na:

* + **Oddzielenie konstrukcji obiektu od jego reprezentacji:** Ten sam proces konstrukcji może tworzyć różne reprezentacje obiektu (Dispute).
  + **Kontrolę nad procesem tworzenia krok po kroku:** Budowniczy udostępnia metody do ustawiania poszczególnych części obiektu, a dopiero na końcu zwraca gotowy produkt.
  + **Uproszczenie tworzenia złożonych obiektów:** Ukrywa skomplikowaną logikę konstrukcji przed klientem.
  + **Konsekwencje:** Wprowadzenie dodatkowych klas (interfejs budowniczego, konkretni budowniczowie), co zwiększa liczbę klas, ale poprawia czytelność i elastyczność procesu tworzenia obiektów.

**2.3 Fragment projektu przed zastosowaniem wzorca:**

A screenshot of a computer

AI-generated content may be incorrect.

**2.4 Struktura i elementy po zastosowaniu wzorca:**

A screenshot of a computer

AI-generated content may be incorrect.

**2.5 Współdziałanie po wdrożeniu wzorca:**