1. **Cel zadania**

Celem zadania jest przekształcenie samodzielnej aplikacji sklepu na infrastrukturę sklepową typu serwer-klienci.

1. **Zadania do wykonania**

Przed rozpoczęciem wykonywania zadań należy przeczytać całą instrukcję a następnie skonsultować się z innymi członkami grupy czy proponowane zadania pozwolą osiągnąć zamierzony cel.

Należy zwiększyć liczbę projektów do dwóch. Następnie każdemu z projektów nadać nazwy jasno określające przeznaczenie: CarAndHorseStoreClient i CarAndHorseStoreServer.

1. **CarAndHorseStoreClient**

W pierwszej kolejności należy usunąć z projektu wszelkie powiązania bazodanowe: modele bazy danych, samą bazę danych, repozytorium itp.

Kolejnym krokiem jest przekształcenie logiki odpowiedzialnej za zadania wymagające połączenia z bazą danych np. logowanie, dodanie produktu do koszyka. Wyświetlenie pomocy czy wyczyszczenie ekranu zostaje w aplikacji klienckiej chyba, że koszty ich pozostania będą większe niż zysk otrzymany z ograniczenia ilości zapytań do serwera. W efekcie powinny powstać metody, które wyświetlają dane na ekranie w niezmieniony sposób. Różnica polega na uzyskaniu tych danych; są one otrzymywane z serwera za pomocą nowych funkcjonalności, nie jak dotychczas przez wykonywanie operacji na bazie danych.

Należy dodać funkcjonalność odpowiedzialną za komunikację z serwerem (google „c# komunikacja serwer klient”. Jest to jedynie pomoc na początek. Proszę nie traktować tej ani kolejnych jako umywanie rąk obok problemu przez autora).

Należy dodać funkcjonalność odpowiedzialną za pakowanie komend do struktury, przesyłaną później za pomocą powyższej funkcjonalności.

Komendy odczytanie CommandParserem należy przekierować do powyższej funkcjonalności tak jak obecnie delegowane są do SystemStore.cs (Proszę wybaczyć autorowi niekonsekwentną i niefachową nomenklaturę gdyż, na co dzień zajmuje się innymi zadaniami).

Należy dodać funkcjonalność odpowiedzialną za wypakowywanie informacji z otrzymanych struktur.

Należy dodać funkcjonalność, która będzie delegowała otrzymane komendy do klasy je obsługujących.

1. **CarAndHorseStoreServer**

Należy usunąć elementy uprzyjemniające korzystanie ze sklepu użytkownikom tj. komunikaty głosowe, fantastyczne logo i pomoc.

Należy utworzyć funkcjonalność odpowiedzialną za komunikację z wieloma klientami.

Należy utworzyć funkcjonalność odpowiedzialną za pakowanie informacji do struktur przesyłanych później do klientów.

Należy utworzyć funkcjonalność odpowiedzialną za wypakowywanie komend otrzymanych od klientów.

Należy przekształcić CommandParser tak by oczekiwał komend rozpakowanych przez powyższą funkcjonalność. Nie jest już konieczne sprawdzanie poprawności składni gdyż walidacja odbywa się po stronie klienta.

Należy utworzyć funkcjonalność, która będzie odbierała rozpakowane komendy i przekazywała je do CommandParsera.

Należy przekształcić logikę zawartą w SystemStore.cs. Wynik operacji nie powinien być wyświetlany na ekranie a przekazywany funkcjonalności od pakowania informacji.

Należy zastanowić się czy funkcjonalności nie da się połączyć.

1. **Ogólne**

Należy stworzyć funkcjonalność odpowiedzialną za zapewnianie synchronizacji informacji w przesyłanych strukturach (autor nie wie o co chodzi, możliwe, że trochę drzemał na zajęciach).

1. **Uwagi**

Powyższe zadania mogą wydać się bardzo ogólne. Nie jest to jednak ich ostateczna wersja. Zdaniem autora w sposób wyczerpujący przedstawiają podstawowe działania, niezbędne by osiągnąć cel. Szczegóły zostaną omówione podczas meetingu z pozostałymi członkami grupy.