

Założenia i plan działania:

1. Zaprojektować aplikację.
 - a. Założyć repozytorium w githubie i podłączyć do niego folder w xampie o nazwie "Reload".
 - b. Do przemyślenia:
 - i. bazy danych - tabele:
 1. item:
 - a. item_id, weight, arrival_date, status, departure_date
 2. item_haevy:
 - a. item_id, weight, arrival_date, status, departure_date
 - ii. Połączenie z bazą danych
 - iii. interface'y i klasy (lub klasy)
 1. Transportation_Unit
 - a. Truck (virtualnie)
 - b. Truck_Heavy (virtualnie)
 - c. Lorry
 - d. Airplane
 2. Test - opracowanie klasy testowej do załadunku Hangaru
 - iv. psr-4 i konfiguracja
 1. include 'Psr4Autoloader.php' w index.php
 - v. namespace'y stosownie do klas
 - vi. interfejs
 1. Przycisk do testowego przyjazdu tira z towarem, czyli załadowanie bazy losową ilością pudeł o losowej wadze
 2. Przycisk do testowego przyjazdu ciężkiego tira z maszynami rolniczymi, czyli załadowanie bazy maszynami rolniczymi
 3. Stan magazynowy - wyświetlanie w tabeli (?)
 4. Jeden przycisk do załadunku
 - vii. symulator (?)
 - viii. zabezpieczenie
 - c. Zastanowić się nad kolejnością.
 2. Wykonać aplikację zgodnie z planem.
 3. Przeczytać o uruchamianiu z linii komend. (?)
 4. Przetestować.
 5. Usunąć wady.
 6. Odczekać i popatrzeć.
 7. Zrefaktoryzować.
 8. Przesłać do sprawdzenia.
 9. Odczekać.

Utworzyłem dwie tabele: "item" na małe paczki i "item"heavy" na maszyny rolnicze. Wedle specyfikacji nie ma takiej potrzeby. Zrobiłem to dla porządku i większej przejrzystości. Małe paczki z TIRA na ciężarówki mają swoją niezależną ścieżkę od maszyn rolniczych ładowanych na samolot.

Przyjąłem założenie, że przeładunek jest niezależny od ilości dostaw do hangaru. Dostawy mogą się kumulować. Funkcja przeładunku - loadStock() - każdorazowo szykuje listy do załadunku na ciężarówki (lub kolejne kursy tej samej ciężarówki) i na samoloty (lub kolejne kursy jednego samolotu) aż do oczyszczenia hangaru.

Paczki wybierane są w kolejności od najstarszej do najnowszej i grupowane w ilościach adekwatnych do ładowności środka transportu.

Do wyświetlania tabel zastosowałem javascriptową bibliotekę DataTable, która umożliwia łatwą maniulację danymi.