# Explanation

## Back-End

I have created three models, i.e. Load, LoadWithRotation and Vehicle.

## Class Information

#### Load

This is the model which holds the basic information (length, width and height) about the load.

#### LoadWithRotation

This is the model which maps each load with the number of loads that the User selects.

#### Vehicle

This is the model which holds the basic information about the vehicle.

#### LoadForFormDto

This is the class which acts as an DTO object to map the value submitted by User against the Model.

#### VehicleSelectController

This is the Controller which handles the request to and from client.

#### Core

This is the component which has all the business logic.

#### ApiTest

This is a NUnit test class, that I created following the test-driven development.

## Approach

I first, broke the User story into multiple pieces and for each piece I created a Unit test which was failing. I then worked out each unit test and made them pass and move to another test. Once, all the unit tests were passed, which means all the business logic was ready I developed the front end code in Angular.

### Business Logic

1st A request is received from the Client in the form of LoadForFormDto.

This request has the list of length,height,width and number of loads.

2nd LoadForFormDto is converted to the list of Load model and if the number of loads is greater than 1 we treat them as a separate load.

3rd We check if the load fits in the PessengerCar

* For all Loads we get the possible combination of length,height and width after Rotation using Permutation.
* Then, we ignore those combination which does not fit in the car.
* Then, we run a recursive method to check if for any combination we get Sum of (allLoad’s Length+ allLoad’s Width + allLoad’s Height) is less than the threshold value of Pessenger car.
* If we get any Sum that is less than the threshold value then we return back from the recursive function which means that Load can be shipped from Pessenger car.
* Return the “Pessenger Car” string as Json if the recursive function returns the load.

4th If the above recursive function does not return anything then we check if the load fits in the Delivery Van using the same logic as above, but comparing the Sum of( length,width and height) against the threshold value of Delivery van.

* Return the “Delivery Van” string as Json if the recursive function returns the load.

5th If the above recursive function does not return anything then we check if the load fits in the Truck using the same logic as above, but comparing the Sum of (length,width and height) against the threshold value of Truck.

* Return the “Truck” string as Json if the recursive function returns the load.