In de laravel applicatie zijn een aantal feature testen opgenomen. Ze lijken que opzet erg op unit testen. Een unit test gaat meestal om een klein stukje code (hooguit één functie). Een feature test gaat wat verder.

Kijk naar de testen in:

* [HomeTest.php](https://github.com/summacollege/project4/blob/master/tests/Feature/IsRoleAanwezigTest.php)

Bijvoorbeeld een nieuwe testcase voor een eenheid van een pizza ingrediënt (gram, stuk etc):

*use RefreshDatabase;*

*public function test\_store\_method\_unit(){*

*// maak een unit aan*

*$unitController = new UnitController();*

*$unitController->store(new Request([*

*'name' => '10 gram'*

*]));*

*$this->assertDatabaseHas('units', [*

*'name' => '10 gram'*

*]);*

*}*

Of bijvoorbeeld een pizza ingrediënt (item):

*use RefreshDatabase;*

*public function test\_store\_method\_item(){*

*// maak een eenheid aan*

*$itemController = new itemController();*

*$itemController->store(new Request([*

*'name' => 'tomatensaus',  
 ‘unit\_id’=>1,  
 ‘price’=>35*

*]));*

*$this->assertDatabaseHas('items', [*

*'name' => 'tomatensaus'*

*]);*

*}*

In de bovenstaande testen beschrijven we dus dat we een unit of een item willen bewaren in de database. Om dit voor elkaar te krijgen gebruiken we een controller (die dus moet worden aangemaakt). In de controller gebruiken we een “store” methode (die dus ook moet worden aangemaakt). Uiteindelijk testen we of het bewaren van een nieuw record is gelukt.

Maak zelf minimaal twee testen (je mag bovenstaande als voorbeeld gebruiken). Dit mag zowel in de laravel applicatie als in de c# applicatie. Beschrijf hoe je vanuit de test de daadwerkelijke code hebt ontwikkeld. De beschrijving mag op papier. Je mag er ook een korte video van maken.