Component: Abstraction Models

Component: Abstraction Models continued

```
my_test_framework
   |- my_test_framework
      |- apps
         |- website
             - base.py
             - home.py
             - cart.py
         |- api
             - base.py
             - letters.py
      |- data
           - shirts.py
      |- framework
          - selenium_utils.py
          - utils.py
          - exceptions.py
      |- models
          - products.py
          - carts.py
          - pricing.py
      |- tests
          - conftest.py
          - pytest.ini
          - test_simple.py
    - requirements.txt
```

Component: Abstraction Models

An abstraction model is a way of rolling up individual interactions, actions, or touch-points into a form that can support the simplification of end-to-end testing.

In our example, we have the following business processes that need to be abstracted:

- The definition of products. The set of products is likely to change over time, so we need a robust way of extending our tests.
- The definition of orders, or the combination of products.
- The pricing model. The data model for products can cover individual pricing, but the pricing of combinations of products, discounting, etc. is definitely something we want to do algorithmically and outside of test case logic.