Purplebricks Test

# Code review

There are a few places where null reference exceptions could arise:

* OffersOnPropertyViewModelBuilder, method Build (property could be null, following the query to retrieve it, but the following code does not take account of this).
* AcceptOfferCommandHandler, method Handle
* RejectOfferCommandHandler, method Handle
* Several other places

There are some examples of source files which contain more than 1 class (e.g. OffersOnPropertyViewModel.cs, AccountViewModels.cs)

The ApplicationDbContext.cs source file contains a definition of an interface, as well as of the class which implements it.

There is a lot of code duplication within the ViewModel builders, the Command handlers and in the Controllers. For example, all of these implement similar constructors, and a private IOrangeBricksContext instance.

* The Controllers could all have a common parent Controller class to avoid having to implement the same constructor several times.
* The ViewModel builders and Command handlers could also have common parent ViewModelBuilder and CommandHandler classes, respectively, for the same purpose.

A number of source files have redundant ‘using’ declarations at the top of them. These can be removed to reduce clutter.

There is a redundant “case” in the switch statement in AccountController, method Login. This case drops through to the default case, which may or may not be intentional. If it is intentional, then that case could be removed without any impact. If it is not intentional, then there is a missing “break” statement after that case.