**Requirements Constraints Responsibilities:**

1. A user will have A single identifiable name in system.

Enforcing class – UserController.

1. System must always have a minimum of one SystemManager.

Enforcing class – Market.

1. SystemManager must be a subscribed user.

Enforcing class – SystemManager Constructor.

1. ShopOwner and ShopManager must be a subscribed user.

Enforcing class – Shop.

1. Only market users can perform actions in market – Permission
2. An active shop must have at least one ShopOwner.

Enforcing class – Shop.

1. A store must have purchase types and discount types. There can be defaults for these types.

Enforcing class – ShopPolicy.

1. A store must have purchase policy and discount policy. There can be defaults for these policies.

Enforcing class – ShopPolicy.

1. A User has a single cart which is composed from all their ShoppingBaskets.

Enforcing class – User.

1. A User may a have a max of one ShopppingBasket in each shop.

Enforcing class – User.

1. A User’s cart is in their Sole possession. No Users can Alter another User’s cart.

Enforcing class – User.

1. Inventory of a product in shop may not be negative.

Enforcing class – Inventory.

1. The Number of products purchasable in a shop does not exceed the Inventory of the product in shop.

Enforcing class – Shop.

1. Users can be only debited for purchases that were completed by them.

Enforcing class – Shop.

1. Users can be only debited for the amount due of the purchase.

Enforcing class – Shop.

1. A checkout process is successful only if payment went through and only if supply was confirmed.

Enforcing class – Shop.

1. A Seller may receive payment only for Successful purchases.

Enforcing class – MethodsOfPayment (External Service).

1. System must always be connected to a PaymentService at all times.

Enforcing class – ExternalConnector.

1. System must always be connected to a SupplierService at all times.

Enforcing class – ExternalConnector.

* Responsibility for external services requests: ExternalConnector is responsible for making the connections, disconnecting, sending requests to services.
* Responsibility for Consistency: In each class we have a set of methods which are responsible to enforce the relevant constraint. The Market is consistent because we ensure that each component of the system is consistent.
* Responsibility for user privacy: UserController and User Classes are responsible for obscuring unique identity data.