



The notification system implements the Observer design pattern, it holds a data structure containing all the notifications that are needed to be displayed in the UI, once the UI initiates a checkNotifications request (every 500 msec for example), the UserService returns all the notifications back to the UI, simulating a real time response of the server.

Each registered user (subscriber) has a unique username

System Administrator (admin) is a kind of subscriber with managing permissions, there has to be at least 1 at all times

Each Owner/Manager object represents an ownership/management of a single store by a single user, therefore a single user can be an owner in some stores and a manager in other stores with no problem

Each shopping basket contains the requested products from a single store

Represents the order history of the user, each order is created from the current shopping cart, copies all information from the baskets once the order system successfully processed it.

Checks the availability using the external services before being marked as complete. If one of the external services failed the order is marked as failed but could still be saved in the order history