

## **P2: ForSale.MIT.edu**

### **II. Design Analysis**

#### **A. Overview**

##### **Key design challenges**

###### **User Identification**

Initially the best decision for this application was to use certificates for user identification because the main users of this application will be MIT students, at least to start with. Currently the app uses the standard user login and password method until certificates is set up. However the login username is named Kerberos as a placeholder for where in the future, the Kerberos name will be used.

###### **Shopper & Shopkeeper Model**

This application could either be implemented with separate accounts for buyers and sellers, or as a one-stop-sell-and-buy service. Looking at the target audience, it seems that the latter would be most beneficial.

#### **B. Details**

##### **Data representation**

The Object Model is a very close representation of the data structures used to organize the various models and controllers. The User has both a Bag and a Room that are initialized on creation of the User. Every Item is created from a Room, and may be added to a Bag at a later time. These are the four models included in the design: *User, Room, Bag, Item*. Other than the identification of the User the only information is about each Item. Items have a name, price, date created, date updated, status, and owner. The status starts out as 'active', is changed to 'claimed' when someone adds it to their Bag, and is changed to 'bought' as soon as someone checks it out.

##### **Key design decisions**

###### **Kerberos for identification**

Using Kerberos for identification will prevent possible security issues and organize users in intuitive way.

**Buy and Sell for Every User**

This is the most efficient implementation for this problem because many students have things they want to sell and buy. If these capabilities were separate, students would have to make separate accounts from each.

**Rooms and Bags Contain Items, Not Users**

Because a User has to play the role of both the seller and the buyer in this application, the easiest way to keep these separate is to make them the equivalent of two different Users working as one. Every purchase is made through the User's Bag and every sale is made through the User's Room.

**Item Modes**

In order to keep multiple Users from thinking they have purchased the same item, there had to be some way of distinguishing which items should be displayed in 'The Lounge', which should be displayed only in one User's Bag, and which should only be displayed in a User's sell history. 'Active', 'Claimed', and 'Bought' represent these three modes. Three modes works better than my first idea of using two because it distinguishes between an item that is currently being checked out, and one that is checked out.