|  |  |
| --- | --- |
| Use Case Number | 03.01.01 |
| Use Case Name | BrowseFriendsInventories |
| Participating Actors | User |
| Goal | Show all (public) items of all friends |
| Trigger | User clicks browse |
| Precondition |  |
| Postcondition | User has options to return from item of propose trade |
| Basic Flow | 1. User clicks Browse 2. System runs CacheFriendsInventories UC 3. System shows all inventories 4. User selects an item 5. System displays item |
| Exceptions | 2.1 no friends or no items  2.1.1 system display informational message where results would be  2.2 No connection  2.2.1 Load from Cache  2.2.2 System Alerts User Inventories may be out of date |
| Qualities | User can propose trade for item or go back  User has options to refine search/browse |
| Constraint |  |
| Includes | CacheFriendsInventories, Search Query, SearchBrowseCategory |
| Extends |  |
| Related Artifacts | US03.01.01, |
| Notes |  |
| Open Issues |  |
|  |  |

Test

User user = new User(“Userman”); //creates user

User friend = new User(“Friendman”); //creates friend x?

User friend2 = new User(“Friendman2”);

user.sendFriendReq(friend); //Creates FRIENDSHIP!!! between user and friends

friend.acceptFriendReq(friendreq);

user.sendFriendReq(friend2);

friend2.acceptFriendReq(friendreq);

Item item = new Item(“Starbucks”,20);

Item item2 = new Item(“Starbucks”,25);

friend.addItem(item); //adds to item to friends inventory x?(public or private)

friend2.addItem(item2);

Item item3 = user.browseAll().get(0);//calls generic browse method stores the 1st return value in item3 Item item4 =user.browseAll().get(1); //calls generic browse method stores the 2nd return value in item4

assertTrue(item2==item); //checks that all public items are returned

Item item4 = new Item(“Starbucks”,20); //adds items to user

User.addItem(item3);

assertTrue(Friend.browseall().get(0)==item3); //calls generic browse method for friend

//checks all public line 7 items are returned