Use case

|  |  |
| --- | --- |
| Use Case Name | Mark My Thing Returned And Available |
| Participating Actors | Owner (Primary) |
| Goal | Owner marks a Borrowed item that they own as Available upon its return |
| Trigger | Owner clicks on the Mark Returned button |
| Precondition | Owner must be a registered user and must own an item that currently has Borrowed status |
| Postcondition | The item’s status is returned to Available |

Basic Flow

|  |  |
| --- | --- |
| 1 | Owner clicks on the Mark Returned button |
| 2 | System prompts user to agree that the item is returned |
| 3 | Owner clicks yes |
| 4 | System notifies Owner the item is now Available |

Exceptions

|  |  |
| --- | --- |
| 3 | Owner clicks no |
| 3.1 | Exit use case |

Unit Test

String name = “Monopoly”; // some meaningful item name

String username = “Steve.Smith”; //some meaningful username

Item item = new Item( name );

item.setStatus( “Borrowed” ); //item is now borrowed

User user = new User( username );

user.addOwnedItem( item ); //user owns borrowed item

assertTrue( user.getOwnedBorrowedItems().contains( item ) );

assertFalse( user.getOwnedAvailableItems().contains( item ) );

assertEquals( item.getStatus(), “Borrowed” ); //redundant check

user.markItemReturned( item );

assertTrue( user.getOwnedAvailableItems().contains( item ) );

assertFalse( user.getOwnedBorrowedItems().contains( item ) );

assertEquals( item.getStatus(), “Available” ); //redundant check