Use case

|  |  |
| --- | --- |
| Use Case Name | Accept Offer On My Thing |
| Participating Actors | Owner (Primary) |
| Goal | Owner accepts an existing bid on an item |
| Trigger | Owner clicks on the Accept Bid button |
| Precondition | Owner must be a registered user and must own an item with a bid on it |
| Postcondition | The item’s bid is accepted, the item’s status becomes Borrowed (to borrower), and all other bids are declined |

Basic Flow

|  |  |
| --- | --- |
| 1 | Owner clicks on the Accept Bid button |
| 2 | System prompts Owner if they wish to accept the bid |
| 3 | Owner clicks yes |
| 4 | System changes item’s status to Borrowed |
| 5 | System changes item’s borrower to Borrower and rate to Rate from Bid |
| 6 | System declines all other outstanding Bids on item |
| 7 | System notifies Owner the bid has been successfully accepted |
| 8 | System returns to Owner’s List of Things With Bids |

Exceptions

|  |  |
| --- | --- |
| 3 | Owner clicks no |
| 3.1 | System notifies Owner the bid was not accepted |
| 3.2 | Exit use case |

Open Issues

|  |  |
| --- | --- |
| 1 | Other Borrowers not notified of decline |

Unit Test

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

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

String username2 = “Joe.Stevens”; //some other user

String username3 = “Mike.Gimli”; //third user

double amount = 1.46;

double amount2 = 1.12;

Item item = new Item( name );

User user = new User( username );

User borrower = new User( username2 );

User borrower2 = new User( username3 );

Bid bid = new Bid( borrower, amount );

Bid bid2 = new Bid( borrower2, amount2 );

item.addBid( bid );

item.addBid( bid2 );

user.addOwnedItem( item );

assertEquals( user.getItem( item ).getStatus(), “Bidded” );

user.getItem( item ).acceptBid( bid );

assertEquals( user.getItem( item ).getStatus(), “Borrowed” );

assertFalse( user.getCurrentBids( item ).contains( bid ); //ensure the item’s accepted bid is no longer in current bid list

assertFalse( user.getCurrentBids( item ).contains( bid2 ); //ensure the item’s current bids do not include the second, auto declined bid.

assertEquals( user.getItem( item ).getBorrower(), borrower );

assertEquals( user.getItem( item ).getRate(), amount );