Item

|  |  |  |
| --- | --- | --- |
| **delete** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in and upgraded as seller and an Item is selected* | | |
| Item not found on the database |  | an ItemNotFoundException is thrown |
| valid |  | it deletes the object Item |

|  |  |  |
| --- | --- | --- |
| **modify** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in and upgraded as seller and an Item is selected* | | |
| the title is null, composed only by numbers, longer than 30 characters or contains a special character |  | the Item is not modified and a TitleInvalidException is thrown |
| the description is null, composed only by number or longer than 500 characters |  | the Item is not modified and a DescriptionInvalidException is thrown |
| category not selected |  | the Item is not modified and a CategoryInvalidException is thrown |
| all valid |  | it modifies the data of the selected instance of the object Item |
| Item not found on the database |  | an ItemNotFoundException is thrown |

|  |  |  |
| --- | --- | --- |
| **addPicture** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in and upgraded as seller and an Item is selected* | | |
| there is less than one image or more than ten | *precondition: the file uploaded is an image type* | the Item is not created and an ImageInvalidException is thrown |
| all valid |  | It creates an instance of ItemPicture linked with the selected item |

|  |  |  |
| --- | --- | --- |
| **getAuctionHistory** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in and upgraded as seller and an Item is selected* | | |
| The item has never been put in an auction |  | An NoHistoryException is returned |
| valid |  | It returns a list of all the old auctions and, if present, the current one |

Auction

|  |  |  |
| --- | --- | --- |
| **subscribe** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in and has not already subscribed* | | |
| User is not allowed to subscribe due to seller’s restrictions |  | A NotAllowedExecption is thrown |
| User can subscribe |  | it adds the user to the subscribedUsers ArrayList and allows the user to bid |

|  |  |  |
| --- | --- | --- |
| **checkTime** | | |
| **Input** | **Description** | **Output** |
| *precondition: the auctions starting time has been reached* | | |
| currentTime < endTime |  | No output |
| currentTime = endTime |  | it closes the auction, calls assignToUser( ), stops checking the time and stops the possibility of place new bids |
| currentTime > endTime |  | Throws an InvalidTimeException |

|  |  |  |
| --- | --- | --- |
| **assignToUser** | | |
| **Input** | **Description** | **Output** |
| *precondition: auction has ended* | | |
| No user has placed a bid  (currentWinner = null) |  | Throws a ItemNotSoldException |
| currentWinner = buyer that has the bid with the last timestamp |  | The item is assigned to that user  (winner = currentWinner) |

Auction

|  |  |  |
| --- | --- | --- |
| **placeBid** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in and subscribed to the auction* | | |
| New bid with the currentValue + fixed value chosen by the seller |  | the user is set as a currentWinner and computePrice( ) is called |
| Two bids with the same timestamp value |  | Throws BidNotValidException to randomly one of them |
| New bid with timestamp > endTime of the auction |  | Throws EndedAuctionException |

|  |  |  |
| --- | --- | --- |
| **computePrice** | | |
| **Input** | **Description** | **Output** |
| null |  | Throws nullPointerException |
| valid |  | Computes new current price and updates currentValue |

Seller

|  |  |  |
| --- | --- | --- |
| **createAuction** | | |
| **Input** | **Description** | **Output** |
| *precondition: seller logged in* | | |
| the Item selected does not exist |  | the Auction is not created and an ItemNonExistentException is thrown |
| the start or end time are not in the right format (dd/mm/yyyy hh:mm) or the end date is prior the start date |  | the Auction is not created and a DateInvalidException is thrown |
| the price and/or the price incrementation are not in the right format (currency) or below zero |  | the Auction is not created and a PriceInvalidException is thrown |
| all valid |  | it creates an instance of the object Auction and return true |

|  |  |  |
| --- | --- | --- |
| **addItem** | | |
| **Input** | **Description** | **Output** |
| *precondition: seller logged in* | | |
| the title is null, composed only by numbers, longer than 30 characters or contains a special character |  | the Item is not created and a TitleInvalidException is thrown |
| the description is null, composed only by number or longer than 500 characters |  | the Item is not created and a DescriptionInvalidException is thrown |
| there is less than one image or more than ten | *precondition: the file uploaded is an image type* | the Item is not created and an ImageInvalidException is thrown |
| category not selected |  | the Item is not created and a CategoryInvalidException is thrown |
| all valid |  | it creates an instance of the object Item and returns true |

|  |  |  |
| --- | --- | --- |
| **displayMyAuctions** | | |
| **Input** | **Description** | **Output** |
| *precondition: seller logged in* | | |
| ArrayList myauctions empty |  | print “no Auctions already created” |
| ArrayList myauctions at least one some Auction instances |  | return ArrayList<Auction> |

|  |  |  |
| --- | --- | --- |
| **displayMyItems** | | |
| **Input** | **Description** | **Output** |
| *precondition: seller logged in* | | |
| ArrayList myItems empty |  | print “no Items already inserted” |
| ArrayList myItems at least one some Auction instances |  | return ArrayList<Item> |

User

|  |  |  |
| --- | --- | --- |
| **userLogin** | | |
| **Input** | **Description** | **Output** |
| the username field is empty |  | the login will not be done and a UsernameInvalidException is thrown |
| the mail field is empty, or not in the right format (xxx@xxx.xxx) |  | the login will not be done and a MailInvalidException is thrown |
| the password field is empty |  | the login will not be done and a PasswordInvalidException is thrown |
| the username or email are not bonded with the provided password |  | the login will not be done and a LoginInvalidException is thrown |
| all valid |  | true |

|  |  |  |
| --- | --- | --- |
| **retrieveSubscribedAuctions** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in* | | |
| ArrayList subscribedAuctions empty |  | prints “no Auctions already subscribed” |
| ArrayList subscribedAuctions at least one some Auction instances |  | returns ArrayList<Auction> |

|  |  |  |
| --- | --- | --- |
| **becomeSeller** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in* | | |
| SSN empty or not in the right format (XXX-XX-XXXX) |  | the Seller is not created and a SSNInvalidException is thrown |
| all valid |  | creates a Seller instance and returns true |

Dashboard

|  |  |  |
| --- | --- | --- |
| **searchItem** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in* | | |
| the title is null, composed only by numbers, longer than 30 characters or contains a special character |  | a TitleInvalidException is thrown |
| category not selected |  | a CategoryInvalidException is thrown |
| all valid |  | return ArrayList<Item> |

|  |  |  |
| --- | --- | --- |
| **searchAuction** | | |
| **Input** | **Description** | **Output** |
| *precondition: user logged in* | | |
| the start or end time are not in the right format (dd/mm/yyyy hh:mm) or the end date is prior the start date |  | a DateInvalidException is thrown |
| the price and/or the price incrementation are not in the right format (currency) or below zero |  | a PriceInvalidException is thrown |
| the title is null, composed only by numbers, longer than 30 characters or contains a special character |  | a TitleInvalidException is thrown |
| category not selected |  | a CategoryInvalidException is thrown |
| all valid |  | returns ArrayList<Auction> |

ItemPicture

|  |  |  |
| --- | --- | --- |
| **remove** | | |
| **Input** | **Description** | **Output** |
| *precondition: seller logged in* | | |
| the Item selected does not exist |  | the Auction is not created and an ItemNonExistentException is thrown |
| all valid |  | the picture is removed, and the method returns true |