Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 19 Bratislava

Software modeling

Charity e-auctions

Algebraic specification

Matej Hrnčiar
3. ročník, 5. semester
Utorok, 14:00

Class: Bidder

Types:

Inherited from User:

id: int
first_name: char
last_name: char
username: char
address_line: char
OP: char
ICO: char

Unique for Bidder:

payment_methods: PaymentMethod[ ]
bookmarks: Auction[ ]

Functions:

Inherited from User:

changePersonal(first_name: char, last_name: char): void
changeUsername(username: char): void
changePassword(password: char): void

Unique for Bidder:

new(first_name: char, last_name: char, username: char, address_line: char, op_ico_number: char, op: boolean): Bidder
searchAuctions(filters: char[ ]): Auction[ ]
selectAuction(auction: Auction): Bidding[ ]
selectBidding(bidding_name: char): Bidding
makeBid(new_amount: float): void
addPaymentMethod(payment_method: PaymentMethod): void
removePaymentMethod(payment_method_name: char): void
bookmarkAuction(auction: Auction): void
removeBookmarkedAuction(auction_name: char): void

Axioms:

A1: ∀a ∈ Auction: ∃b ∈ Bidding (For every Auction, there is at least one Bidding)

A2: ∀a ∈ Auction: a.auction_start < a.auction_end (The start of Auction is before its end)

A3: ∀b ∈ Bidding: b.bidding_start < b.bidding_end (The start of Bidding is before its end)

A4: ∀u ∈ User: OP ∪ ICO = OP ∨ OP ∪ ICO = ICO (User either has set OP or ICO, not both at the same time)

A5: ∀u1, u2 ∈ User: u1.username ≠ u2.username ∨ u1.OP ≠ u2.OP ∨ u1.ICO ≠ u2.ICO (Two Users cannot have the same username, OP, or ICO)

Assumptions:

∀ *bidder*: Bidder,
a: Auction,
b: Bidding,
bid: float (Highest bid saved in bidding),
bid_new: float (New bid),
p: PaymentMethod,
a_start: Datetime (Start of Auction),
a_end: Datetime (End of Auction),
b_start: Datetime (Start of Bidding),
b_end: Datetime (End of Bidding),
now: Datetime (Current date and time)

changeUsername(new_username: char) requires ∀*u* ∈ User: *u.username* ≠ *new_username* (New username cannot be taken by other user)

changePassword(new_password: char) requires bidder.*password* ≠ *new_password* (New password cannot be the same as the previous one)

searchAuctions(filters: char[ ]) requires ∃*a* ∈ Auction (To perform search, there must be at least one Auction registered in system)

selectAuction(a: Auction) requires ∃*a* ∈ Auction: *a.a_end* > *now* (Auction cannot be selected after it ended)

selectBidding(b: Bidding) requires ∃*b* ∈ Bidding: *b.b_end* > *now* (Bidding cannot be selected if it's over)

makeBid(bid_new: float) requires
a.*a_start* < *now* < a.*a_end* (Auction started, and is in progress) AND
b.*b_start* < *now* < b.*b_end* (Bidding started, and is in progress) AND
b.*bid* < *bid_new* (New amount is higher than previous) AND
∃*p* ∈ bidder.*paymentMethods* (Bidder has at least one payment method)

removePaymentMethod(payment_method_name: char) requires ∃*p* ∈ *bidder.paymentMethods* (There is at least one PaymentMethod saved in the list of Bidder's payment methods)

removeBookmarkedAuction(payment_method_name: char) requires ∃*a* ∈ *bidder.bookmarks* (There is at least one Auction saved in Bidder's bookmarks)