Auction Report

March 2016

Group 16
Adrian Kamulegeya (ak13g15)
Adam Kantorik(ak2g15)
Anish Katariya(ak7n14)
Jonathan Elliot Keable(jek1g15)

Contents

1 Abstract

Abstract

Our group Consisting of Adrian Kamulegeya (ak13g15), Adam Kantorik(ak2g15), Anish Katariya(ak7n14) and Jonathan Elliot Keable(jek1g15) were given the task of creating a software model for an online auction service.

Our group was given guidelines and specifications to follow while making the software for the online auction service.To make the software model for this service or group made the following

1. An abstract containing introduction and tasks comments Made by Anish Katariya
2. Two full scenarios covering all aspects of the auction including but not limited to successful, failed or cancelled auctions Made by Adrian Kamulegeya
3. Two use cases covering 2 aspects
   (a) Covering the holistic image of the online auction service Made by Jonathan Elliot Keable
   (b) Covering the login service that the software will use to sign in registered users Made by Anish Katariya
4. One Use case diagram covering the holistic image of the full software Made by Jonathan Elliot Keable
5. One class diagram showing all functions and classes that will be used to make the Software. Made by Adrian and Adam Kantorik
6. Two sequence diagrams corresponding to the scenarios described above. Made by Adrian Kamulegeya
7. A state chart diagram for the Auction Object. Made by Adam Kantorik

This software model gives all information and sufficient specifications for making the software for the online e-auction service.

2 Scenario

Scenario 1 - Successful Auction

- Both Steve and Billy sign up to the auction website
- Steve submits a phone up for auction
- Steve submits the name of the item to be One Plus One; the start and end times to be 12th March 2016 19th March 2016 ; and a reserve price of 200
- The item is posted on the website; the auction has begun
- Another user submits a bid of 200
- Billy submits his bid of 220
- Billys bid is successful and he is the current highest bidder
- The auction end time has been reached
- Billy is the highest bidder
- Billy is pleased with the service and gives a star rating of 4 to Steve
- Steve sends off the item and Billy receives the item

Scenario 2 - Unsuccessful Auction

- Bob and Sam sign up to the auction website
- Sam submits a laptop up for auction
- Sam submits name of the item to be Macbook Pro; the start and end times to be 13th March 2016 19th March 2016; and a reserve price of 400
- The item is posted on the website; the auction has begun
- Bob tries to make a bid of 300
- The website notifies him he is below the reserve price
- Bob waits a few days and submits a second bid of 400
- The auction has ended, his bid didnt go through
- Sam takes the item off of the website, the auction has failed

3 Use Case

Use case name	`AuctionItem`
Participating actors	Initiated by `Seller` Communicates with `Bidder`(s)
Flow of events	1. The `Seller` puts an item up for auction 2. A `Bidder` places a bid on the item 3. Another `Bidder` places a higher bid on the item 4. Repeat until an exit condition is met 5. IF The auction was successful then the `Bidder` pays the `Seller` who then dispatches the item 6. The `Seller` may rate the `Bidder` 7. Regardless of auction outcome, any participating `Bidders` may provide the `Seller` with a rating
Entry condition	• The `Seller` is logged into their account • The `Seller` has less than three zero star ratings
Exit condition	• The auction time has passed and the highest bid is higher than the reserve price, OR • The auction time has passed and the highest bid is lower than the reserve price, OR • The auction is cancelled by the `Seller` and the highest bid is below the reserve price, OR • The auction is cancelled by the `Seller` and the highest bid is above the reserve price
Quality requirements	On average bids should take no more than one minute to be registered by the system

Use case name	`Validate User`
Participating actors	Invoked by `User` Communicates with `Server and Database`
Entry condition	• The `User` Enters his user name and password • The `User` Presses the submit button
Flow of events	1. The `user` enters his user name and password and pressed the submit button. 2. The `system` sends the information to the `server` to check credentials 3. The `server` checks the credentials 4. If credentials are right `server` sends a request to the `system` allow the user to log in If credentials are wrong `server` sends request to the `system` to show an error message 5. `System` checks the request sent by the `server` 6. `System` shows user homepage or error message according to the request sent
Exit condition	The `user`is successfully logged in and shown the user homepage, OR The `User` has entered wrong user name or password and shown an error message
Special requirements	Each user has to be pre-registered in the system The users credentials should be stores in the server

4 Use Case Diagram

Figure 1: Use Case Diagram

5 Class Diagram

The Class diagram shows the associations between different classes in the system.

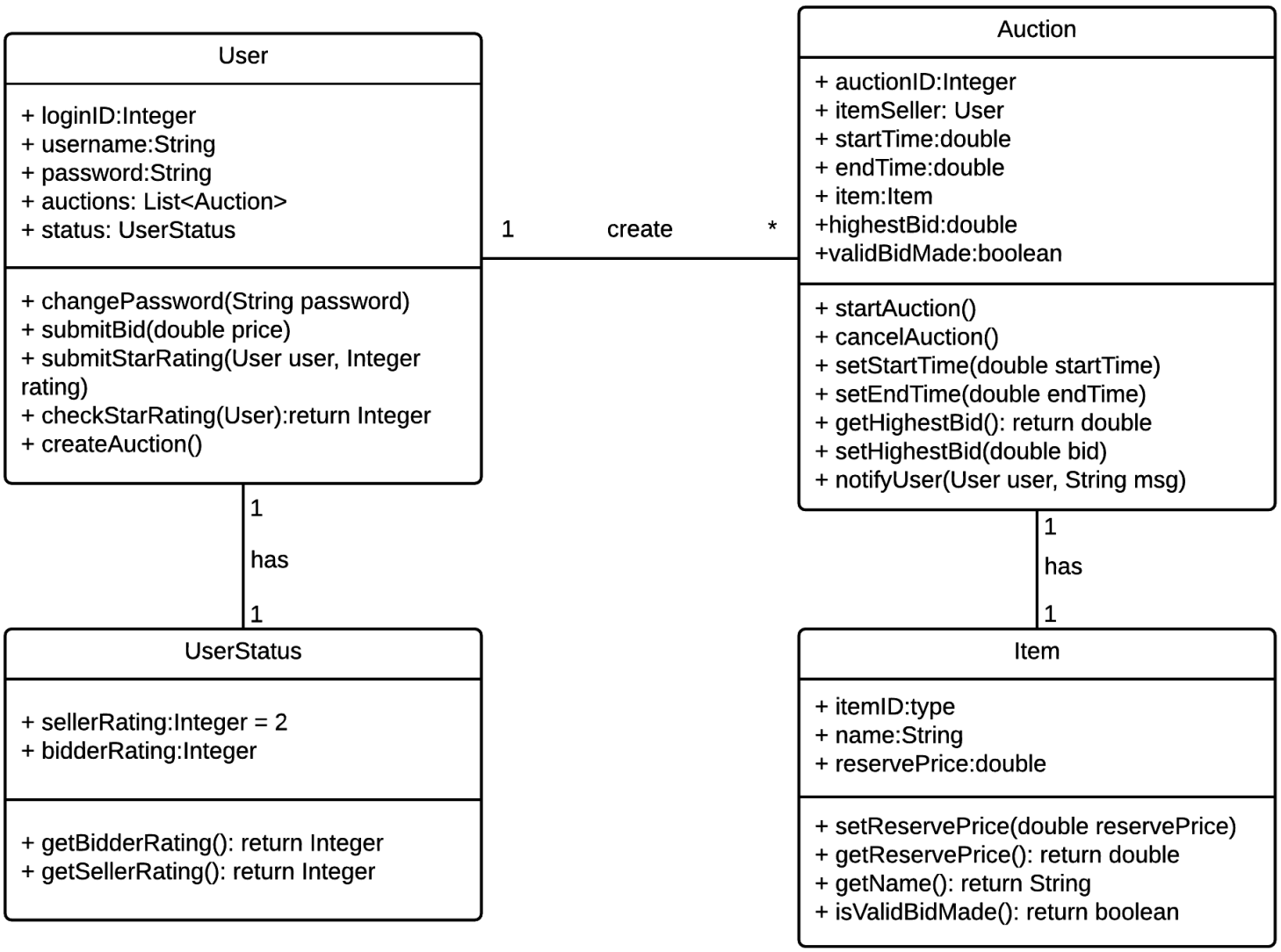

Figure 2: Class Diagram

6 Sequence Chart Diagram

For the auction process we created earlier two scenarios that documented a successful auction process and an unsuccessful auction process. To illustrate these scenarios further we created two sequence diagrams showing the execution of these auctions:

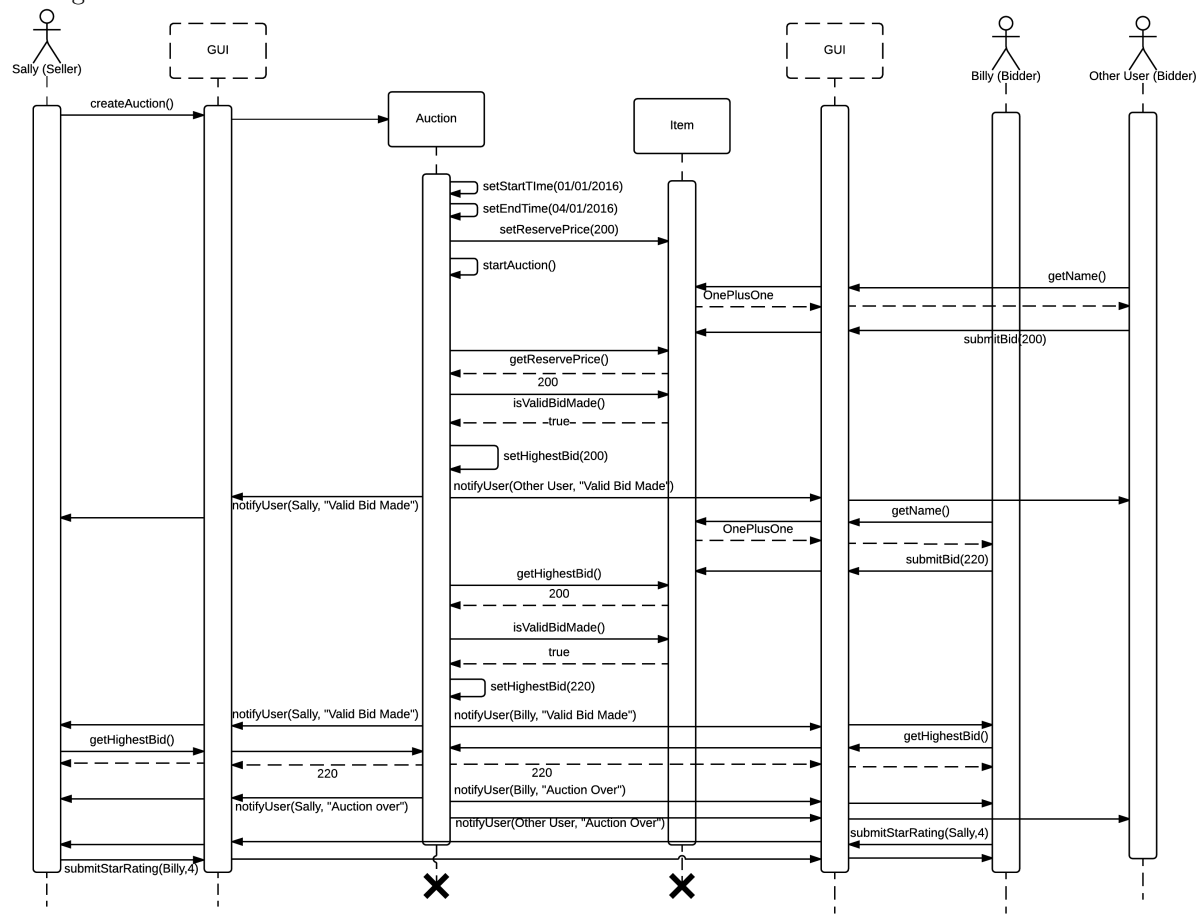

Figure 3: Successful Auction Process

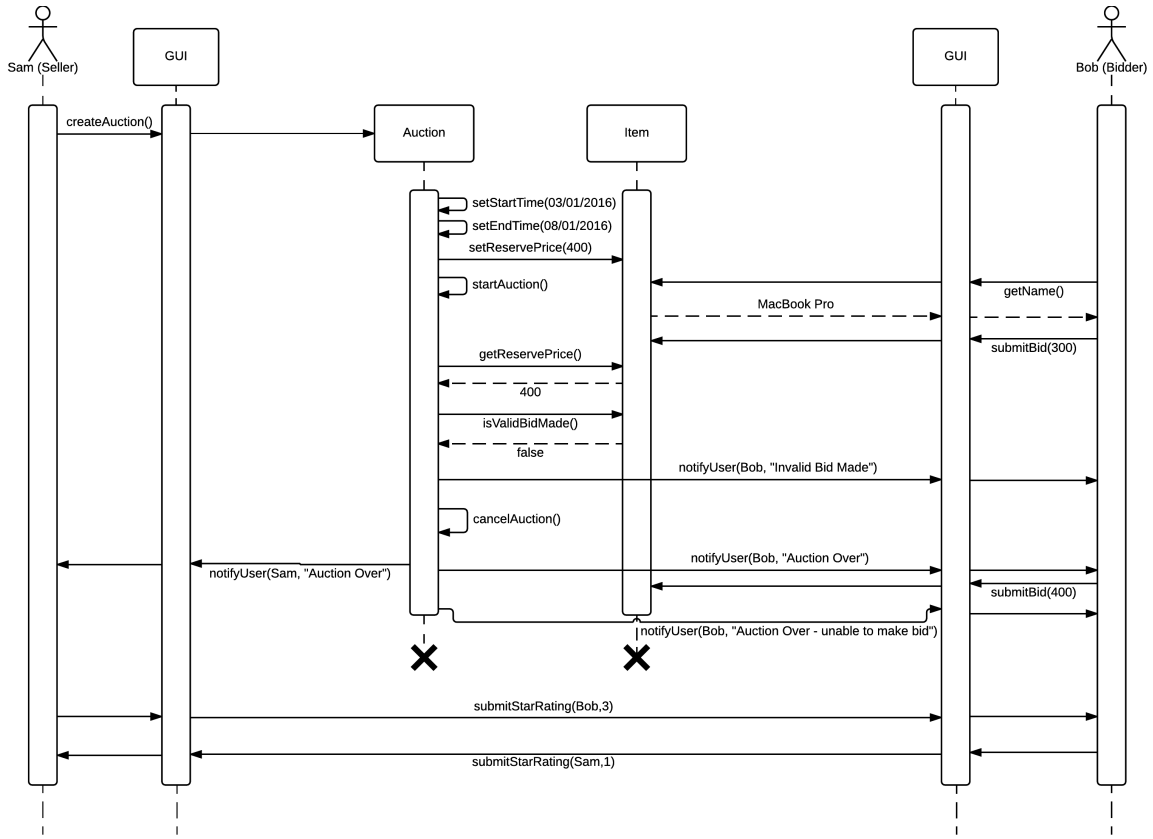

Figure 4: Unsuccessful Auction Process

7 Statechart Diagram

The Statechart diagram shows lifecycle of auction object- its states from creation till end or cancelation.

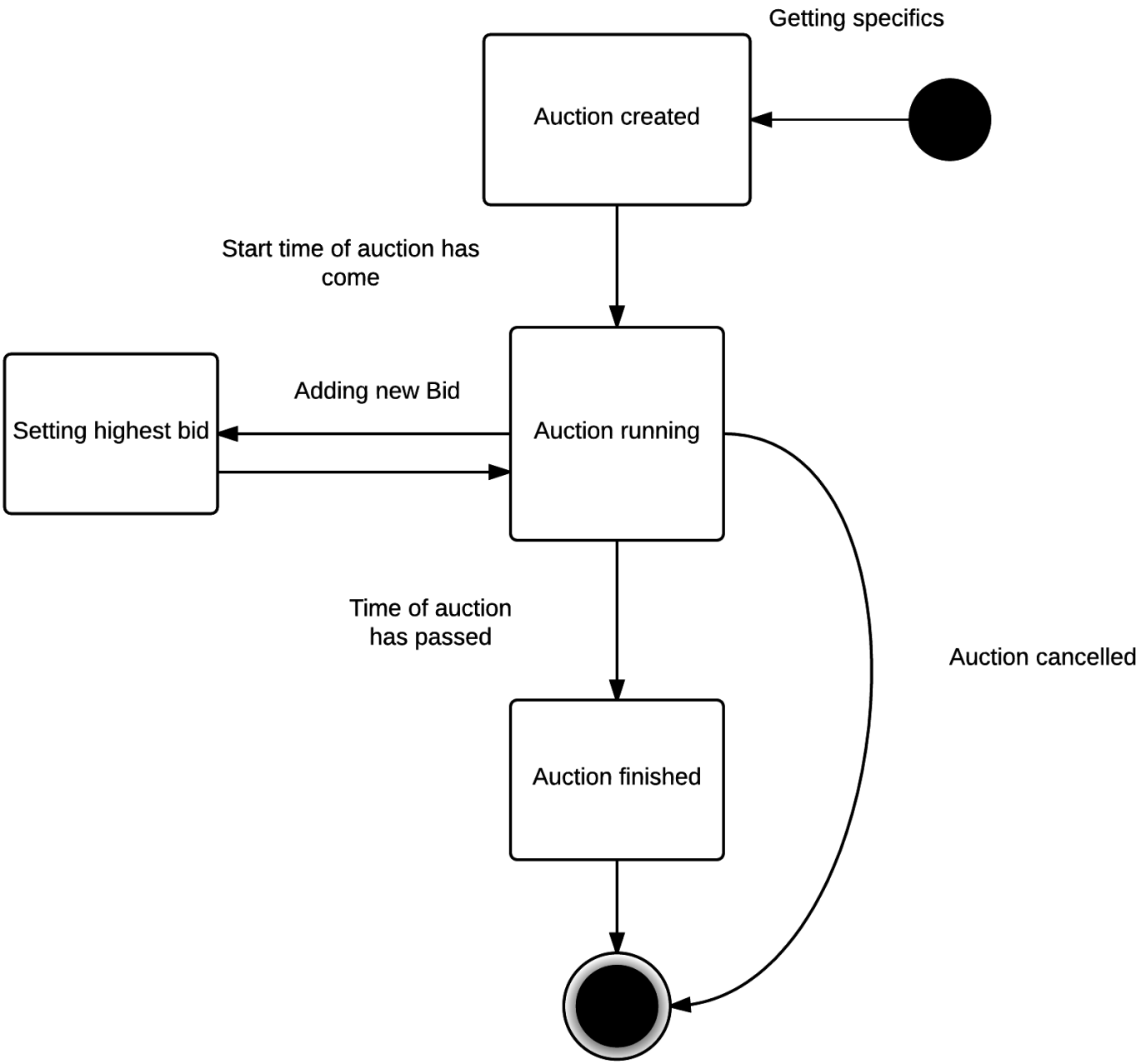

Figure 5: Statechart Diagram