POLITECNICO DI MILANO

FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE

Progetto di Ingegneria del Software 2

Prof.ssa Raffaela Mirandola
A.A. 2014\15

Giuseppe Vergori
MATRICOLA: 824298
C.P. 10294634
giuseppe.vergori@mail.polimi.it

Titolo Progetto: GUESSBID
Part I: RASD

Requirements Analysis Specification Document

Sommario

1. INTRODUCTION

1.1 DESCRIPTION OF THE PROBLEM

GUESSBID is a web based application implementing an inverse auction system. In this way, people can offer and bid for anything that wish to sell or buy. An inverse auction works like a regular auction. The difference is that in an inverse auction a user has to propose the lowest unique bid to win the auction.

Each user has to guess the lowest bid. Bids can be placed till the bidding closing time. Then they are evaluated by the system. The winner is the user who has placed the lowest unique bid. If for instance, the situation is the following:

- Two users have bid for 10;
- One user has bid for 8;
- One user has bid for 7;
- Three users have bid for 4.

The lowest unique bid is the one for 7 and therefore the user who has placed this bid will win.

Before the ending of the bidding time each user can provide more than one bid. At each bid the user is informed about is current position with respect to the others. Also, the system provides updates when such current position changes as a result of biddings from other users. The cost to make an offer is of 2 per bid and it is automatically subtracted from the system when a user makes a bid.

1.2 GOALS

The main issue of the GUESSBID project is to develop a web application that will allow users, after registration, to create auctions, to see all auctions present on the application, to participate in all auctions that want and to control the state or the current position in the related auction.

The main features of GUESSBID will be the follow:

- User registration at the web application.
- The system will assign to a new user a virtual credit of 100.
- Each bid costs two virtual credit.
- A registered user can login into the system.
- A logged user can:
    - Create an auction for each good he/she has, defining an expiration date after which the auction expires.
    - Bidding for an existing auction.
    - Be informed by the system about the current status of his/her biddings for opened auctions.

- Be notified about the outcome of the auction when this is closed.

1.3 DOMAIN PROPERTIES

Domain means the real world in which the application work, that is the web.
In particular, there are five main constraints that must will be respected:

- A user can use the application only if it is signed up;
- Every logged user can browse the existing auctions and he/she can bid for the goods of these auctions.
- A logged user cannot to bid for an auction created by himself/herself
- The cost of bidding is 2 and it is withdrawn from the user's credit.
- A user can participate in more of an auction until he has available the virtual credit.

1.4 GLOSSARY

In this section will be define some words that will be used in this document.

- **Auction**: the definition is a public sale in which goods or property are sold to the highest bidder. In this case, auction is intended as an inverse auction.
- **Inverse auction**: is a public sale in which goods or property are sold to the lowest unique bidder.
- **New auction**: when a user create an auction before sharing it with the other users
- **Managing auction**: a section where there are the auctions that the user creates or participates;
- **Notification**: is a message that receives a user when he is informed by the system about the current status of his/her biddings for opened auctions. Also, the system will notified the outcome of the auction when this is closed.
- **Position**: indicates the current or the final ranking that the user has in the related auction.
- **Expired date**: indicates the deadline after which the users who participates to the auction cannot bid.
- **Guest**: a person who is not registered to the web application;

- **User**: a person who is registered to the web application and can use its services;
- **Bid**: the operations that can do an user when he bids for a good in an auction
- **Win/Lose/Parity**: the result of an auction. Only one user can win.
- **Virtual credit**: the money used by users for bid.
- **Balance**: the remaining virtual credit of the users.

1.5 ASSUMPTIONS

The application will be developed assuming that:

- Only registered users can access to the web application.
- When a user is signed up, he can use the services that GUESSBID offer. In particular, he can create and share the auctions with the other users that are registered to the web application.
- When a user create an auction, all the other users, which must be registered to the application, can show the page which contains the information about the good.
- The only operation that a user can do when he shows the page, if the balance of his/her virtual credit is more than zero, is to bid.
- The system sends a notification or message to inform a user about the current status of his/her biddings for opened auctions if his/her position in the ranking is changed, or about the outcome of the auction when this is closed.
- When a user bids, if he confirm his offer, he cannot withdraw.
- When a user create an auction, this cannot be changed.
- After the expired date, the only winner is the user who has proposed the lowest unique bid.
- In case of draw, the system does not assign winning or looser. The user, who had created the auction, must create a new one.

1.6 PROPOSED SYSTEM

The application wants to offer an easy service that helps users to create auctions and to share goods following the characteristics discussed above.
Every users can access to the service when they want e with any device, provided that they possess an internet connection.

In this way, they will always be updated in real time about the status and the current position in the auctions in a simple and fast way, allowing the user who bid to revise its strategy and, if his balance greater than or equal to two virtual credit, to do a new bid.

1.7 STAKEHOLDERS

The application is designed for all people who:

- Want to create auctions sharing goods with other users in a simple and fast way. In fact, the system automatically alert the users about the status of their current position in the auction so that the users can make new offers to win the good.

- Want to try to bid on a good hard to find or try to buy a good at a price lower than the reality.

But the "real" stakeholder is the teacher who needs to have by the end of the semester a working product, and respect the specification assigned to him.

2. ACTORS

There are only two main actors:

- **Guest**: an user who is not logged to the application; he can only see the home page where there is a presentation of the system;

- **User**: an user who is logged to the application; he can use the services that the system provides, like to create and share auctions, to bid for the goods and to see the notification that the system send them, described in the previous sections.

3. REQUIREMENTS

For the goals mentioned in the previous section, the system have the following requirements:

- **User's registration**: the system has to provide a sign up functionality;

- **Creation of an auction**: the system must allow users to create their own auction by displaying a specific page; here, the user can enter the information, like description of good and expired date of the auction. After the creation, no one can change the features of the auction.

- **Sharing**: when the user create an action, the other registered users can show the relative page with the information about the good so they can think to bid.

- **Bid for goods**: the user that create an auction cannot bid. Instead, the other registered users can bid for a good as many times as they want, within the expiration date and if they have virtual credit, at a cost of two virtual coin.

- **Ranking notification**: the system when a user bib for a good, updates the current position of the users in the auction. If there is a new lowest unique bid, the system alert the involved users for the auction, i.e. the registered users who have made a bid and their position in the ranking has changed.

- **Final notification**: when the auction is closed, the system sends a message to the involved users to win the good with the final result, so that every user will know if he won or lost. Also, the user who create the auction receive a notification and he can display the name of the winner. In case of equity, all involved users receive a draw notification and lose. So the user must create a new auction.

- **Index auction**: in the main page of the application after the login, all users display a search button that is a link for an index page where they can see all auctions present on the application. If the user clicks on an auction link, he displays the description of the good, the expired date, the number of offers and, if he has made a bid, his current position.

- **My auctions**: in the main page of the application after the login, all users display a "my auction" button that is a link for a page that contain a list of auctions that the user created or he has made a bid. If the user click on an auction link, he displays the description of the good, the expired date, the number of offers and, if he has made a bid, his current position. In case the user is the creator of the auction, when he click on the link he displays only the page of the auction with the number of offers.

- **Balance**: at the top of the main page of the application, after the login, all users display a section that keep track of their virtual credit. Each bid has a cost of two virtual coin and the users can bid if and only if their balance is more than or equal of two.

3.1 FUNCTIONALS REQUIREMENTS

Now, we define the functional requirements related to each actor:

- Guest: an user who is not logged to the application can:
    - display the presentation page of GUESSBID;
    - Access and display the registration page, and fill in the appropriate space his data.
- User: an user who is logged to the application can:
    - display the presentation page of GUESSBID and Login;

- after the Login, he displays the personal page where he can:
  - create e share auctions with other registered user;
  - Bids for an opened auctions;
  - Bids how many times want provided that his balance is more than two virtual coins.
  - Only displays the information about the auction with the number of offers (if the user is the creator of the event);
  - Displays the auctions that he has made bids.
  - Displays all the auctions present on the web application.
  - Displays the pages of the auctions that has made a bid with the current status of the auctions and the number of offers.
  - Displays the notifications received.

3.2 NON FUNCTIONAL REQUIREMENTS

3.2.1 USER INTERFACES

The application uses graphical interfaces designed for the web, easy to use, allowing even inexperienced users to benefit immediately and intuitively of all the functionality of the system. GUESSBID is presented to the users with an easy, complete and essential graphical structure which immediately gives the possibility to identify the key features and to display only the necessary information, for example, for a specific auction. These interfaces use a graphical structure where there are buttons, links and spaces of data entry.

Now, let's see two main page of the web application.

First, let's see the home page of GUESSBID, a page that can display any user with their browser. This page is only a presentation page, where a user can read a short description of the service and login or go to the registration page if a user click on the specific link.

Second, let's see the main user's profile page where any user can accesses only after the login.

In this page, the user can use all the features of the system described in the previous sections, like when he clicks on “auctions” or “management auctions”, the application redirects him in two different page where the user can create, bid or only controls the status of the auction.

Let's see in particular that in this page the user displays the balance with his virtual credit.

The "Notification" link will be implemented according to the expired time of the project. The basic idea is that when a user click on the notice link, the application shows him the notification in a new page with a link of the auction which refers. Remember that the system sends a notification if there is a change in the ranking or the auction is closed.

Finally, let's see an auction page. Here the user displays the information about the good, the number of offers and the expired date. If he bid, the system update the ranking and send notifications to the others bidders, if their current position id changed.

3.2.2 DOCUMENTATION

In order to illustrate the development of the project will be made the following documents:

- **Project Plan**: to define tasks and to show the organization and the timings of this process.
- **RASD**: Requirement Analysis and Specification Document, to well-understand the given problem and to analyze in a detailed way which are the goals and how to reach them defining requirements and specification.
- **DD**: Design Document, to define the real structure of the web application and its tiers.
- **Javadoc comments in the source code**: to make anyone that wants to develop the platform or do maintenance on it understand the code.
- **Installation Manual and User Manual**
- **Testing Document**: a report of a test of another GUESSBID project.

3.2.3 ARCHITECTURAL CONSIDERATIONS

We will use J2EE platform with a database in which all system's information will be stored.
What is this information will be clearer watching the Class Diagram given in the chapter 6, because it could be considered as a basis for our database, but the precise ER Diagram will be drawn in the DD.
An Internet connection is needed to use GUESSBID and also a web browser.

3.2.4 ERROR HANDLING

The system must handle a possible loss of connection ensuring consistency and persistence of the stored data and alert the user to the fact. This fact will be implemented according to the expired time of the project.

4 SPECIFICATION

4.1 DEFINITION OF SPECIFICATION

Let's see some specification that satisfy the requirements:

- The creator of an auction is only one user;
- the creator cannot bid in his auction;
- it does not exist an auction not associated with any creator;
- each auction has an expired date;
- each user can bid, if he has virtual credit;
- for each auction corresponds one expired date;
- For each auction corresponds one winner;
- all closed auction generate a notification to all participants;
- a change of ranking generates a notification to all users all users who have bid;
- each user can bid in the same auction how many times want, if he has virtual credit;
- the user can create more than one auction;

4.2 VERIFICATION OF THE SPECIFICATIONS

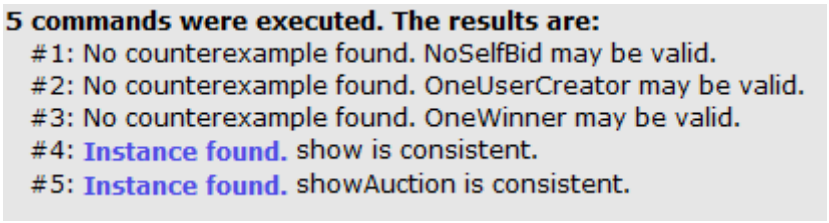
To verify the system model defined by the specification we used a model Alloy, tested by Alloy Analyzer. In the next paragraph there is the code of the model, some instances simulated by the tool and also some assertions verified by Alloy Analyzer.

4.2.1 CODE OF THE MODEL

```
module GuessBid

//SIGNATURES

sig User{}

sig Date{}

sig Auction{

ExpiredDate: one Date,

creator: one User,

bidder: some User,

winner: one User

}

sig Notification{

alert: some User

}

// FACTS

fact UserProperties{

//A creator cannot bid in his auction

no u: User | all a: Auction | u = a.creator && u = a.bidder

//The creator of an auction is only one user

no u1,u2: User | all a: Auction | u1 =  a.creator && u2 =  a.creator

}

fact AuctionProperties{

//It does not exist an auction not associated with any creator

all a: Auction| some u: User | u= a.creator

//For each auction can correspond more users

all a: Auction | some u: User | u= a.bidder

//For each auction corresponds one  expired date

no d1,d2: Date | all a: Auction |  d1 =a.ExpiredDate && d2 = a.ExpiredDate

//For each auction corresponds one winner

no u1,u2: User | all a: Auction | u1= a.winner && u2= a.winner

// all auction have an expired data

all d: Date | all a:Auction | d= a.ExpiredDate
```

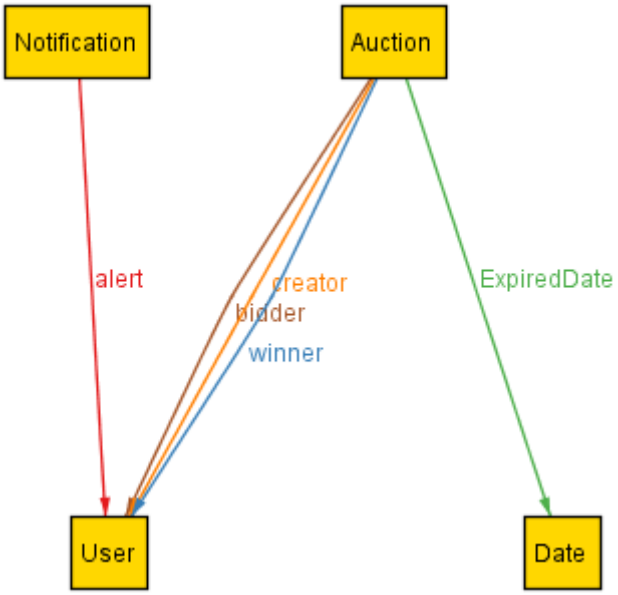
```
//all  closed auction generate a  notification to all participans

all u: User | one n: Notification | u= n.alert

}

//ASSERTIONS

assert NoSelfBid {

no u: User | all a: Auction | u= a.creator && u= a.bidder

}check NoSelfBid

assert OneUserCreator{

no u1,u2: User | one a: Auction | u1 = a.creator && u2 = a.creator

}check OneUserCreator

assert OneWinner{

no u1,u2: User | all a: Auction | u1 = a.winner && u2 = a.winner

} check OneWinner


//PREDICATES

pred show(){}

run show for 3

pred showAuction(){

#Date=#Auction}

run showAuction for 3
```

And here the report of Alloy Analyzer:



4.2.3 EXAMPLE WORLD

Let's see, for example, the simplest world generated by the model. In the figure below we can see the all operations that a user can do with the constraints.



5. SCENARIOS IDENTIFYING

Here are some possible scenarios of GUESSBID:

- Giuseppe wants to find an application that can implements a mechanism of auction for to buy goods at a price lower than their real value. His friends told him about GUESSBID, a new web application that can allow him to create auctions and to bid for goods keeping track of his current position in the auction. So, he decides to find this application through one of the most popular online search engines. After this, he clicks on the specific link and he displays the home page of the web application. He reads the short description of GUESSBID and clicks on "Register", where he completes the form choosing a Username, a password and a system confirm the registration. Now, Giuseppe can login and he is redirect to his home profile page. Here, he sees in the top of the page his balance that contains 100 virtual coin. In the page, there are also the links to create and share auctions with other users that have an account on the application, and to see the auctions that there are on the system. For the moment, he clicks on the auctions link; the system shows him a list of the open auctions and he sees the goods that other users offer. He does not see anything interesting, so he clicks on logout button, in the top left of the page, and he exits the application.

- Giuseppe works too many hours every day and he never has time to do shopping. But recently he discovered GUESSBID and he decides to go on the web site. So, he open the browser, writes the address and go in the home page of the web application. He enters his Username and password and go in his profile page. Now, he clicks on the link "auctions". In this page, the system shows him an index of the auction present on the platform and he clicks on the link to see the goods that the other registered users offer. So, in any relative page, he can see the description of good with the expired date for bid and the number of offers. He is more interested about a new model phone and he decide to bid. So he enter the value of his offers and he clicks the “bid” button. The system update the page and update his balance, that the user displays in top right of the page. He is the first user who bids, so he decides to attend the final moment of the auction to make another bid, if his position is changed; so he clicks on the “logout” button and exits the application.

- Giuseppe, making "zapping" on his browser, decides to check his personal page in GUESSBID to control the status of the auction for which he has bid. So, he enters the address of the application and login. In his personal page he displays a notification: his position in the auction is changed. He clicks on the notification and displays the page of the auction.
    - Unfortunately, now he has the third position in the auction.
      So:
        - No problem: there is still time, and everything can change.

        - The expired date is near and he decides to make a new bid. The system update the page, show him his current position in the auction and update his balance. Now:

        - Perfect: he have the first position in the auction. Eventually, he can attend other notifications about the status of his current position.
        - Bad: he not has the first position and he not have virtual credit. The only hope is that his position change before the auction closed.
    - Very happy, Giuseppe have the first position in the auction. He can attend that the auction closed but sometimes he has to check the system notification because his position can change before the expired date.

  Finally, he clicks on the "logout" button and exits the application.

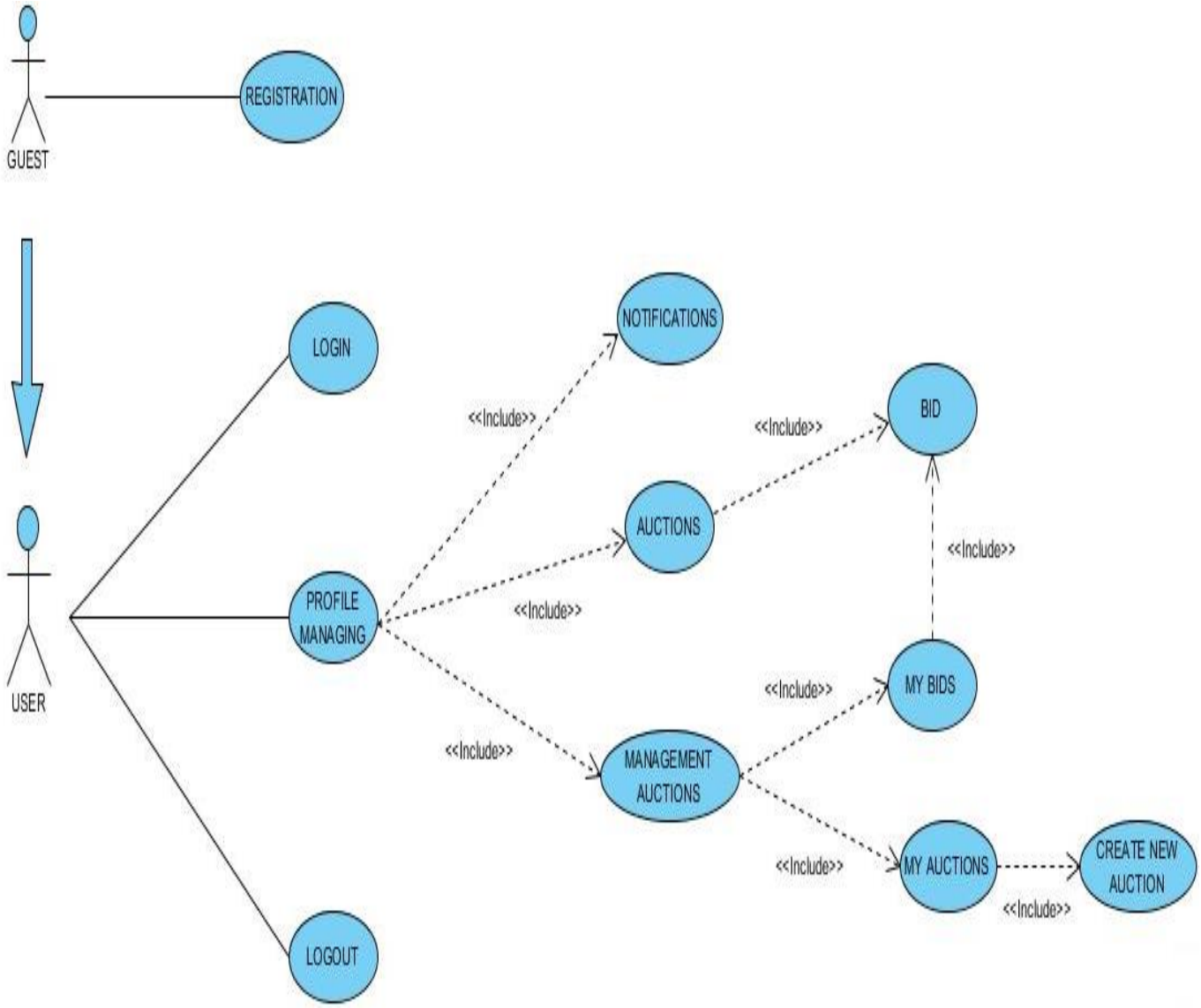
- Yesterday the auction closed but Giuseppe does not control the final result. So, he decides to go on the GUESSBID web page. He opens the browser, enter the address and login on the application. Here, in his personal page, he displays a "notification" and he clicks.
  The system shows him the result of the auction:
  -Unfortunately, he lost. Giuseppe may try again, if he has virtual credit, to bid for another good in another auction.
  -Good: he won. Soon a user, i.e. the creator of the auction, will contact Giuseppe and send him the good.
  - What bad luck! The result of the auction is a draw. He must attend that the creator of the auction create a new auction and, if he wants and has virtual credit, bid for the good.

- Giuseppe has a new phone that what never using and would like to sell. He is registered on GUESSBID, a web application where he can create an auction or bid. So, he opens the browser, enter the address and login on the application. Here, in his personal page, he clicks on "my auctions" and the system opens the relative page. Here, he can see the auctions that he creates and those where he made bid. Also, there is "new auction" button, and he clicks on. Now, he is in the creation page, where Giuseppe add a description of the good and the expired date. After, he clicks on the "save" button, and the system save the auction. From now, all registered users can see the auction and bid. When the auction closed, Giuseppe will receive a notification with the final result. Also, he can control the number of offers in the page of the auction.
  Finally, he returns in his profile page and logout.

6. UML MODEL

6.1 USE CASE

6.1.1 DIAGRAM



6.1.2 USE CASES DESCRIPTION

In this section there are some use cases derived from the scenarios identified in the previous paragraph:

NAME	REGISTRATION
ACTORS	Guest
ENTRY CONDITIONS	The user enter the address on the browser or find the link for the application
FLOW OF EVENTS	- The user displays the home page of GUESSBID; -the application shows him the page; -the user clicks on "registration" button; -the application displays this page; -the user enter in the specific space his personal data, like username and password. - the user clicks on "submit" button.
EXIT CONDITIONS	The application confirm the registration.
EXCEPTIONS	-Username already exists; -password too short or invalid; The user displays an error message.

NAME	LOGIN
ACTORS	User
ENTRY CONDITIONS	The user has successfully signed up to the web application.
FLOW OF EVENTS	-The user opens the home page of the application; -the application shows him the page; -the users enters his Username and password in the input form provided; -the user clicks on the "login" button.
EXIT CONDITIONS	The application checks the authentication of the user and shows him his home profile page.
EXCEPTIONS	The information entered in the form is wrong, an error message is shown.

NAME	CREATE AUCTION
ACTORS	User
ENTRY CONDITIONS	The user is logged and he is in his home profile page. Then he clicks on “managing auctions”.
FLOW OF EVENTS	-the user clicks on a link or a button that says "create auction"; -the application shows him the page for create a new auction; -the user enters in the specific space the information like description of the good and expired date; -the user clicks on the "save button";
EXIT CONDITIONS	The application create the auction. Now all registered users can show the new auction page.
EXCEPTIONS	- an Alert is generated if the auction not contain the expired date. -invalid format date.

NAME	BID FOR AN AUCTION
ACTORS	User
ENTRY CONDITIONS	The user is logged and he is in the auctions page.
FLOW OF EVENTS	-the user clicks on the "auction" link; -the application shows him the page; -the user displays the page, he reads the description of the good, the expired date the number of offers; -the user, interested, enter the value of the offer and clicks on "bid" button. -the application updates the page increasing the number of bid.
EXIT CONDITIONS	The application stores new data and update the status of the auction. Also updates the balance of the user.
EXCEPTIONS	No virtual credit.

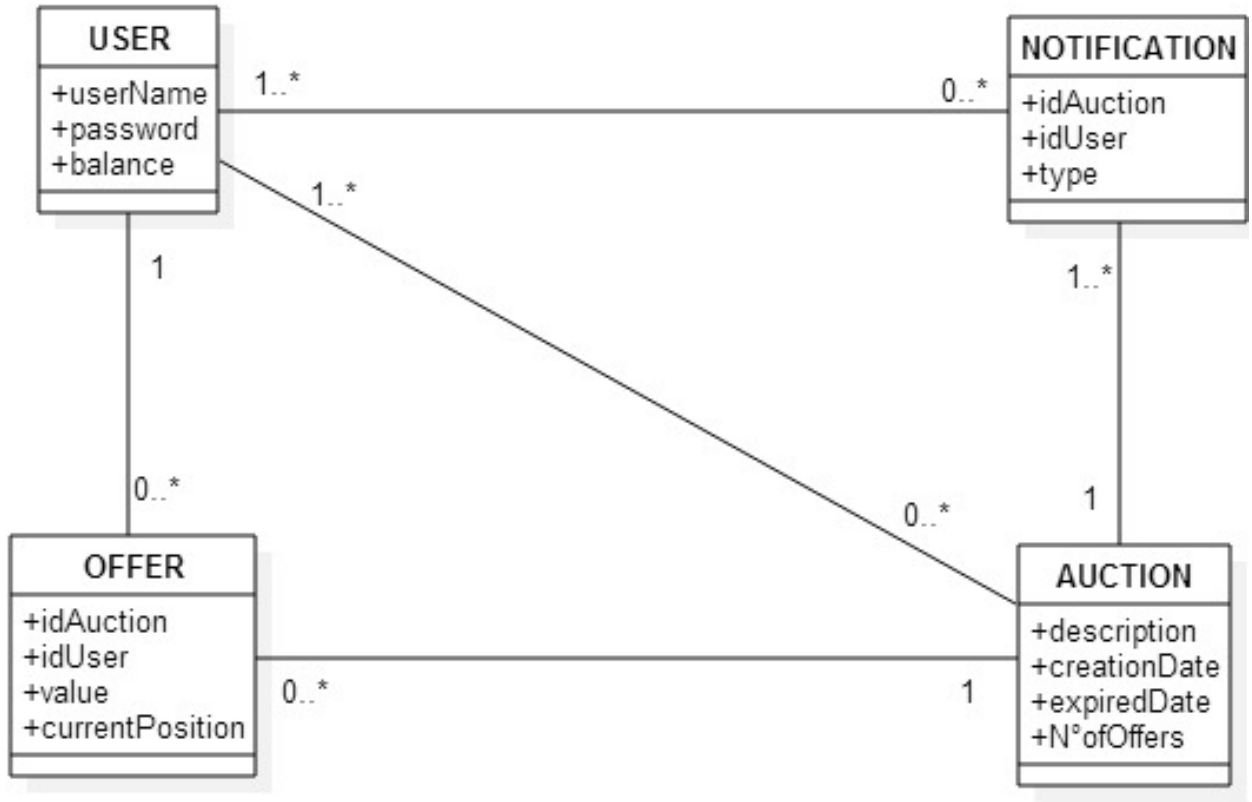
NAME	CHANGES IN THE RANKING
ACTORS	User
ENTRY CONDITIONS	The user is logged and he is in his home profile page.
FLOW OF EVENTS	-the user displays a notification. -he clicks on "notification" button; -the application shows him the notification with his current position in the auction.
EXIT CONDITIONS	The application set the pending status equal to false.
EXCEPTIONS	No connection

NAME	CLOSED AUCTION
ACTORS	User
ENTRY CONDITIONS	The user is logged and he is in his home profile page.
FLOW OF EVENTS	-the user displays a notification. -he clicks on "notification" button; -the application shows him the notification with the final result.
EXIT CONDITIONS	The application set the pending status equal to false.
EXCEPTIONS	No connection

NAME	LOGOUT
ACTORS	User
ENTRY CONDITIONS	The user is logged and he is in his home profile page.
FLOW OF EVENTS	-the user clicks on "logout" button; -the application asks him to confirm the requested operation; -the user clicks on "ok" button and exits.
EXIT CONDITIONS	The application redirects him to the guest home page.
EXCEPTIONS	No connection

6.2 CLASS DIAGRAM

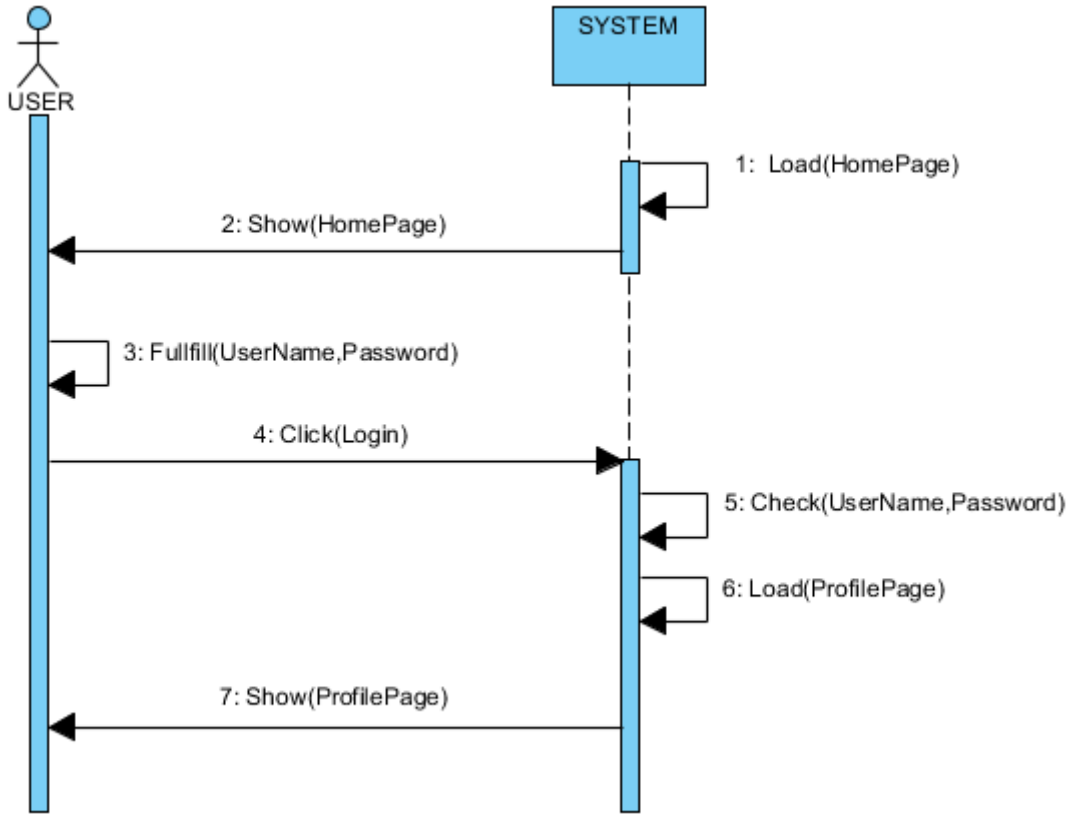
Let's see in this section a basically class diagram of the application. More details will be added in the Design Document.



6.3 SEQUENCE DIAGRAMS

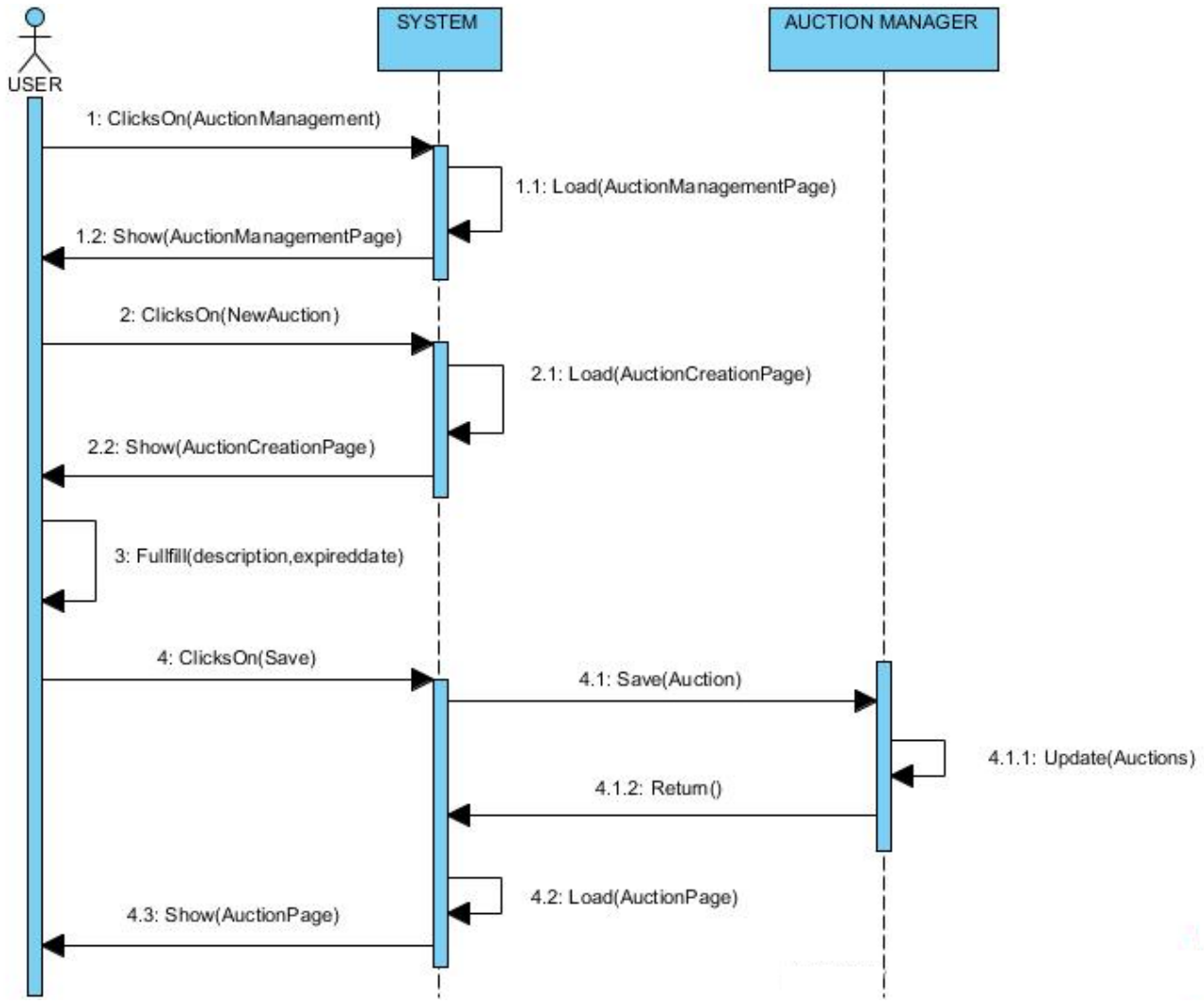
6.3.1 LOGIN

First, let's see the flow of operations that the user and the system must perform during the login phase. The diagram start assuming that the user go in the guest home page of the application.



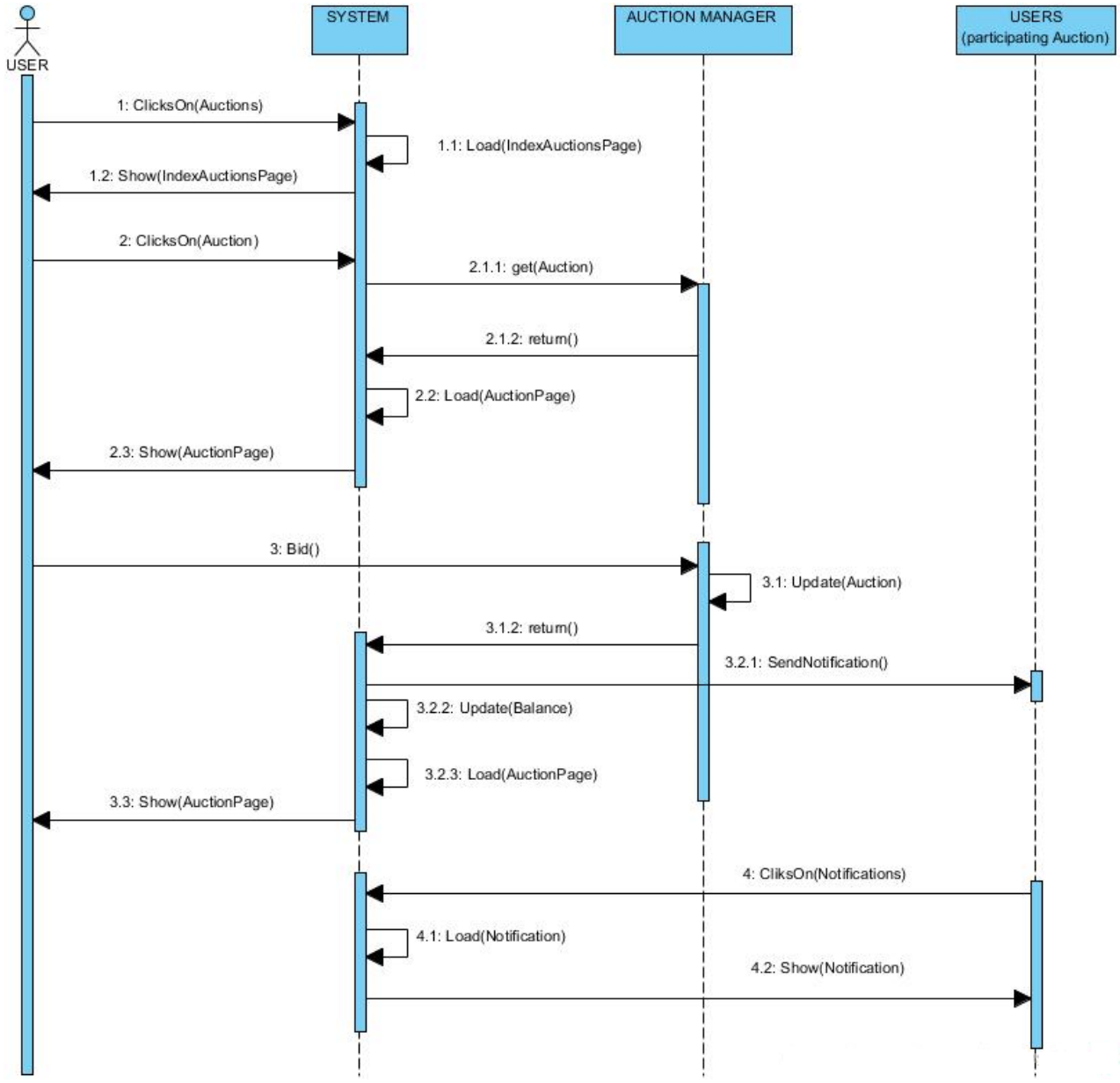
6.3.2 AUCTION CREATION

Second, let's see the flow of operations that the user and the system must perform during the creation phase. The diagram start assuming that the user has been logged and is in his home profile page. At the end of the creation phase, the system add the new auction and all registered users can display the auction and bid.



6.3.3 BID FOR A GOOD

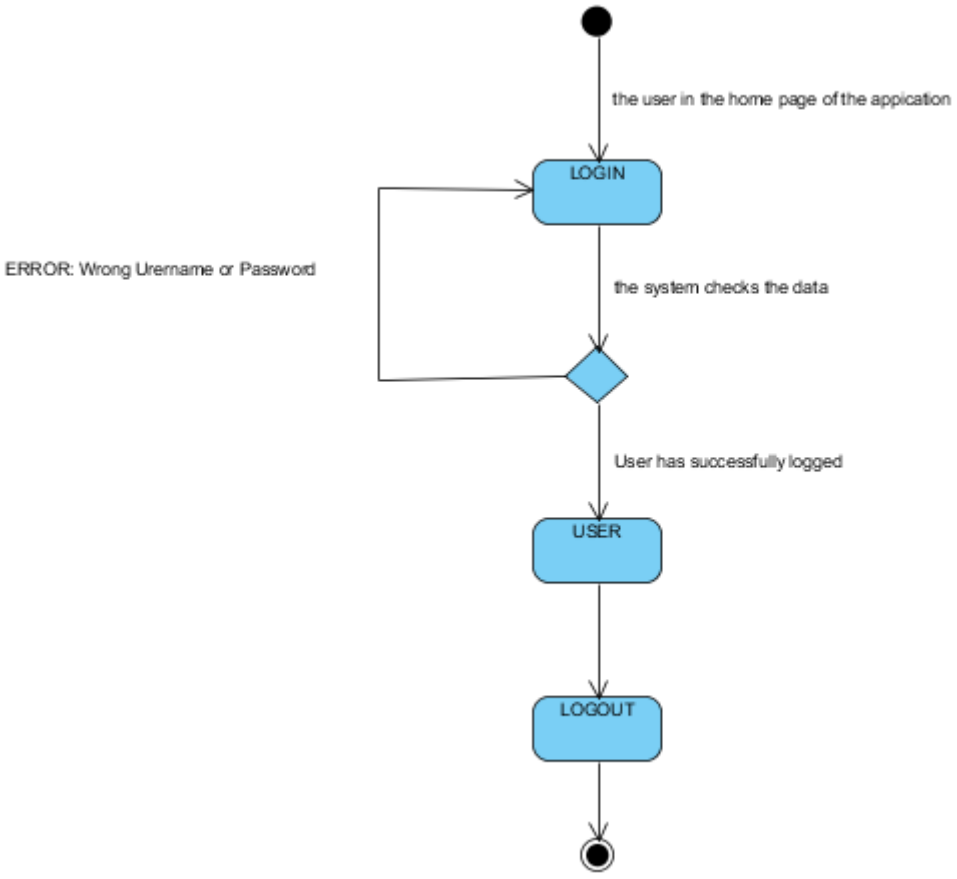
Finally, let's see the flow of operations that the user and the system must perform during the bidding phase. The diagram start assuming that the user has been logged and is in his home profile page. At the end of the bidding phase, the system updates the data of the auction and, if there are changes in ranking, sends a notification to all users involved. Also, the system use the same mechanism when the auction is closed, so a notification with the final result of the auction is sent to all participants.



6.4 STATE CHART DIAGRAMS

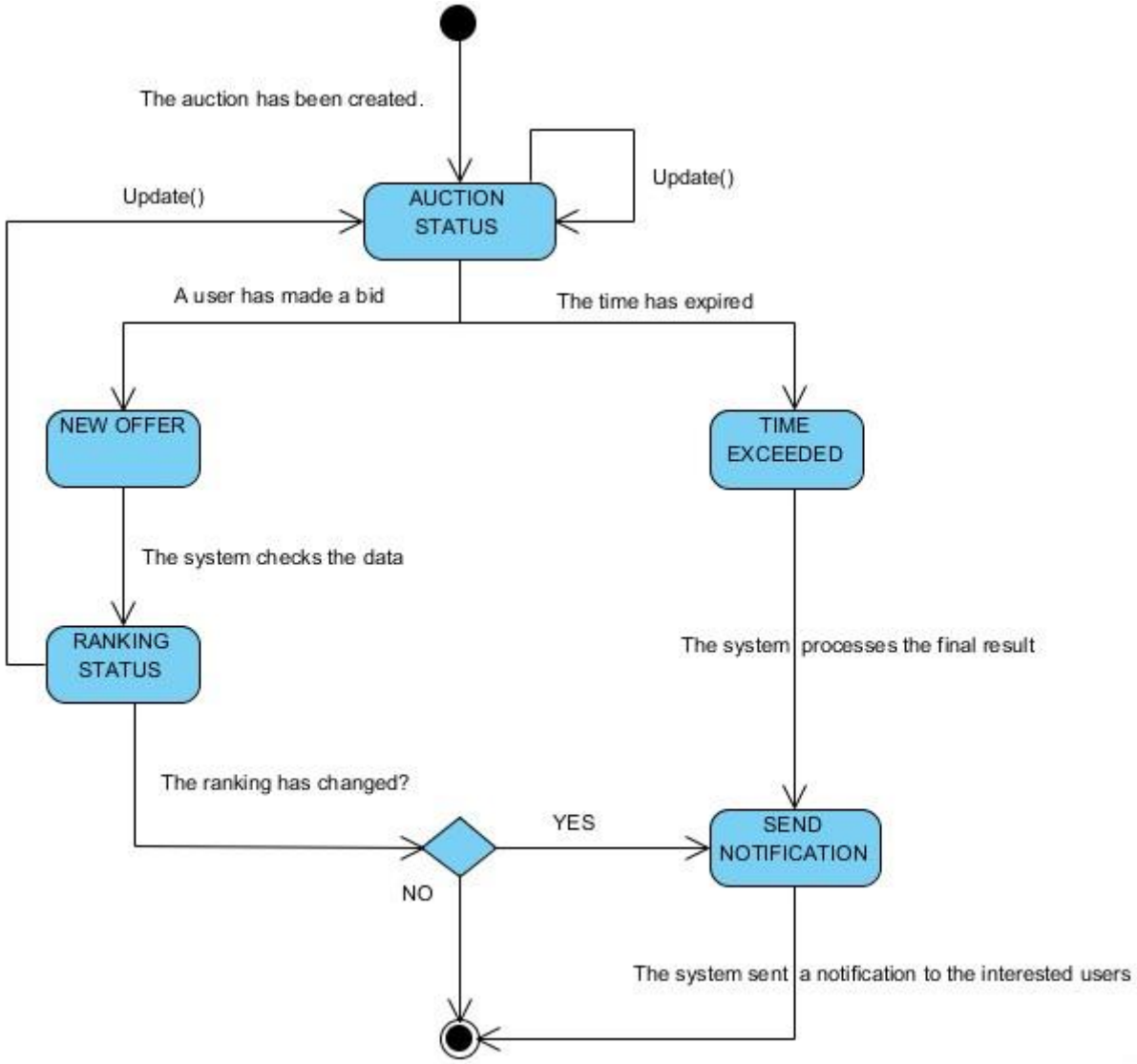
6.4.1 LOGIN

First, let’s see the behaviour of the system during the login phase.



6.4.2 AUCTION STATUS

Finally, let's see the behaviour of the system when the auction is open. The system attends a new offer or the expired date and, when it is necessary, sends a notification to the users who are involved in the auction.



7. USED TOOLS

The tools that I used to create this RASD document are:

- **Microsoft Office Word 365**: to redact and to format this document; also to create the UI sketches;
- **Star UML**: to create Class Diagrams;
- **Visual Paradigm**: to create Use Case, Sequence and State Charts diagrams;
- **Alloy Analyser 4.2**: to prove the consistency of the model and to generate the world.