GuessBid

1 Problem description

GuessBid is an application implementing an inverse auction system. 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 that one bid. At each bid the user is informed about its 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. Issuing bids has a cost of 2 per bid.

More information on the inverse auctions can be found on the web. "Bid and go" is an example of system implementing this kind of auction.

GuessBid comprises the following features:

- A new user can register. The system will assign to it a virtual credit of 100. The cost of bidding (2) is withdrawn from this credit.

- A registered user can login into the system

- A logged user can:

    o Create an auction for each good he/she has, defining an expiration date after which the auction expires.

    o Bidding for an existing auction. Every user can browse the existing auctions and he/she can bid for the goods of these auctions (Notice that it is forbidden for a user to bid for an auction created by himself/herself).

    o Be informed by the system about the current status of his/her biddings for opened auctions.

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

2 Your task

You must develop the system using the Java EE platform. In particular, you will use EJBs to develop the business logic. The user interface can either be a web application or a normal Java application. In both cases the user interface has to interact with the business logic. During the development you should fulfill the deadlines indicated in Section 3 and should abide by the rules described in Section 4.

3 Steps and deadlines

The steps you must proceed through and the milestones for your development process are the following:

Deadline 10 April 2015, 23.59: *Group registration*. You should form your group and register it by going through the following steps:

1. Create a repository for your project on github (https://github.com). Note that in previous projects we have been using GoogleCode, but, unfortunately, Google has decided to dismiss it. Make sure that you give a meaningful name and description for your project. Make sure that all group members have a github account and have access to the repository. We would like to see commits performed by all group members.
2. Register your group by filling in the following form https://docs.google.com/forms/d/1PaYzfX4Osntqjq4bd3t3XOjl4XdrUM7wQpEnYpaf420/viewform?usp=send_form.

From that point on **you will start keeping track of the number of hours each group member works toward the fulfillment of each deadline**.

Deadline 27 April 2015, 23.59: *Requirements analysis and specification document (RASD)*. The RASD contains the description of the scenarios, the use cases that describe them, and the models describing requirements and specification. You are to use a suitable mix of natural language, UML and Alloy. UML and Alloy MUST be part of the documentation. You must also show that you used the Alloy tool for analysis, by reporting the models you obtained by using it. Of course, the initial written problem statement provided above suffers from the typical drawbacks of natural language descriptions: it is informal, incomplete, uses different terms for the same concepts, etc. You may choose to solve the incompleteness and ambiguity as you wish, but be careful to clearly document the choices you make. You will also include in the document information on the number of hours each group member has worked towards the fulfillment of this deadline.

Deadline 15 May 2015, 23.59: *Design document (DD)*. DD must contain a functional description of the system, and any other view you find useful to provide. You should use all the UML diagrams you need to provide a full description of the system. Alloy may also be useful. You will also include in the document information on the number of hours each group member has worked towards the fulfillment of this deadline.

Deadline 22 June 2015, 23.59. *Implementation*. You should provide an implementation for the requirements you specified in the RASD, following the design you specified in the DD. You are requested to release source code and executable, installation and use manuals, system test cases. You will also include in the release a document containing information on the number of hours each group member has worked towards the fulfillment of this deadline.

Deadline 28 June 2015, 23.59. *Validation and acceptance testing*. We will assign you to validate the

project produced by another group. You will act as a quality assurance group. You will review the documentation produced by your colleagues and highlight any issue you may find, define acceptance test cases and report on the execution of tests for the system. You will also include in the document information on the number of hours each group member has worked towards the fulfillment of this deadline.

Deadline 30 June 2015, 23.59. *Project reporting*. You will apply the Function Point approach to your project and check if the results you get are similar to the actual size of your project. Moreover, you will use the project's actual size to apply the formulas from COCOMO and compare the resulting effort with the one you have actually spent. The project reporting document will contain the results of this analysis.

Date to be defined. *Final presentation*. You will have to present your project providing an overview on your requirement and design decisions, a demo of the system, describe your code and test cases and present the results of the acceptance testing you have performed. Overall, groups of two students will have 30 minutes, groups of three students will have 40 minutes, while single students will have 20 minutes.

4 Rules

The project should be developed by groups of 1, 2 or 3 persons (two persons is the suggested size).
- **Groups composed of one person** will develop the entire set of functions described above.
- **Groups composed of two persons** will extend the system to keep track of the story of previous auctions.
- **Groups composed of three persons** will extend the system by adding <u>all requirements for groups composed of two persons</u>, and the following additional requirement: they will develop a mechanism to enable users to see each other presence in the auction and to chat with the others.
- **Telecom students**: Students enrolled in "Telecommunication Engineering/ Ingegneria delle Telecomunicazioni" and following the Internet studies will obey to the rules valid for the students enrolled in "Computer Science and Engineering/Ingegneria Informatica". The others following the traditional telecommunication studies will either develop a more extensive testing activity instead of implementing the solution using JEE or they will join computer science students in their groups. A group featuring a telecom non Internet student should do the implementation part as if it was composed of a number of students not comprising the telecom student. For instance a group of three people featuring a telecom non Internet student should do the implementation part mandated for a group of two.
- **Each group MUST provide the requested artifacts within the stated deadlines. A delay of a few days will be tolerated but it will result in a penalty in the final score. It is mandatory to provide these artifacts and to present them to the reference professor in a final meeting that will be scheduled on the web.**
- **The material presented in one artifact is not fixed in stone. You can provide updates as part of the following deliverables. For instance, if you realize that you need to modify the RASD document and you are in the implementation phase, you can include in your implementation deliverable a short explanation of why you need to change the RASD, and update this last document accordingly by including, at the end of the new version, a short summary of changes.**
- For any question related to the project that could be interesting also for the other groups please use the forum available on the course website. We will answer as promptly as possible.