Secure Online Auction System

## General project description

This is a web application for an Online Auction System. The application consist of selling some products through price bidding. Bidding have start price and ending time. More users can bid for the same product and the winner is the one who bids the item for highest price.

For buying product online user must have an account and to provide his personal details like email address, name, address, etc. Only the valid user will have authority to bid.

Also, there will be another type of user, the admin. This user can make CRUD operations on the application’s database, and his role is to manage the biddings and the products.

## Project functional requirements

The normal user can create an account and then buy some BidPoints using real money. He can use the BidPoints to bid for the products with an open bid. After the bid closes, the user who bid the most BidPoints will win and will have the product delivered to him.

Also, the user can manage his account. This consist of changing his details if needed, he can buy more BidPoints, he can block or delete his account, etc. In the “history” section the user can see his bids, his losses and his wins, and the transactions he made on the website.

The admin can edit the products list, by adding or deleting products from it. Also he can change product information, like description, specifications, price, etc. He can start a new bid on a product, end he can end one in case of errors. He can block users if needed and unblock them, and he can generate reports on the products to see the stock and more details.

## Project non-functional requirements

The most common metric for gauging overall performance is Page Load. In terms of JavaScript, this means how long until the “window.onload” event handler fires in the browser. Users typically expect a site to load in two seconds or less. Search engines penalize pages with slow load times.

For the non-functional requirements the application will have some security: each user or admin has his personal account. Each user can see only his information and his history and can manage only the operations destined for him.

The eventual errors or exceptions from the application will be handled, so the application can be available in every moment and for every user.

For the implementation of this project I will use as backend technology ORM with Spring REST and for the frontend, AngularJS.

The application will use Layers Architecture and it will have a presentation layer, business layer and a data layer.

The application will also generate reports for the admin, and will have a mail system for password recovery and to announce the bidding winner.

The project will have a various number of tables, like: User, UserDetails, Product, Bid, OnlineBids, etc.

## Main use-cases

There are two types of user for this application: the administrator and the normal user. In function of these privileges the users can do a series of operations.

For the administrator:

* Start new bids;
* Add new products in the bidding system;
* Block users;
* End bids in case of errors;
* Report generating;

For the normal user:

* Account creation;
* Account management;
* Buy BidPoints;
* Bidding;
* View his history;