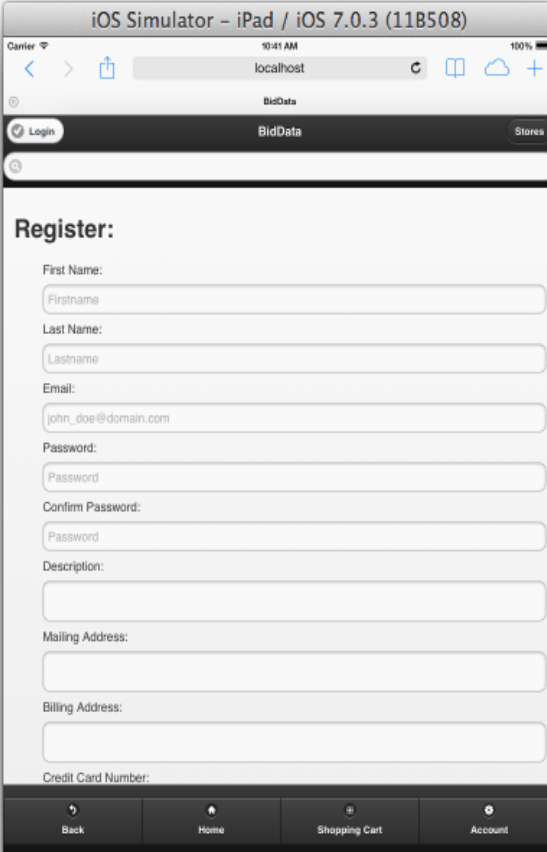


Report Phase II

Walkthrough the user interface: BidData

1. Regular User

- a. ***Register:*** A new user can register in this page by filling the information shown in the figure below.

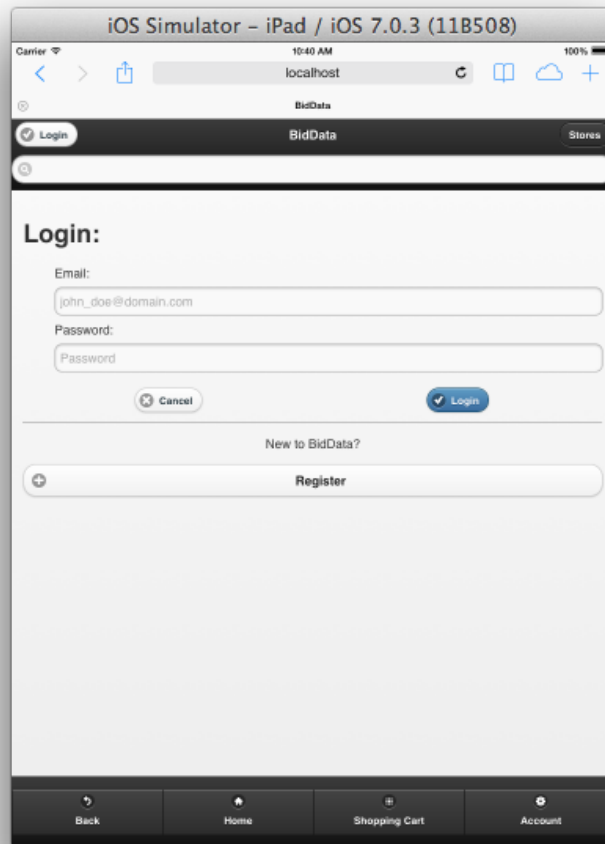


The screenshot shows the 'Register:' screen of the BidData app on an iPad simulator. The status bar at the top indicates 'Carrier', '10:41 AM', and '100%' battery. The browser address bar shows 'localhost'. The app's navigation bar includes 'Login', 'BidData', and 'Stores' tabs. The registration form contains the following fields:

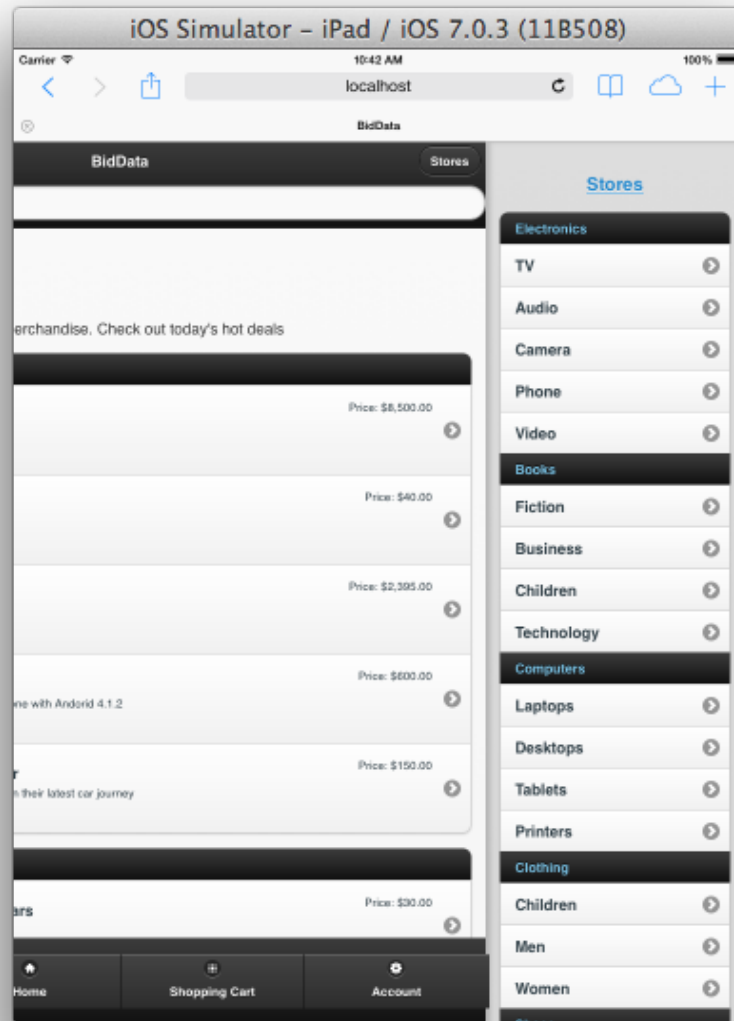
- First Name:
- Last Name:
- Email:
- Password:
- Confirm Password:
- Description:
- Mailing Address:
- Billing Address:
- Credit Card Number:

The bottom of the screen features a dark navigation bar with four icons and labels: 'Back', 'Home', 'Shopping Cart', and 'Account'.

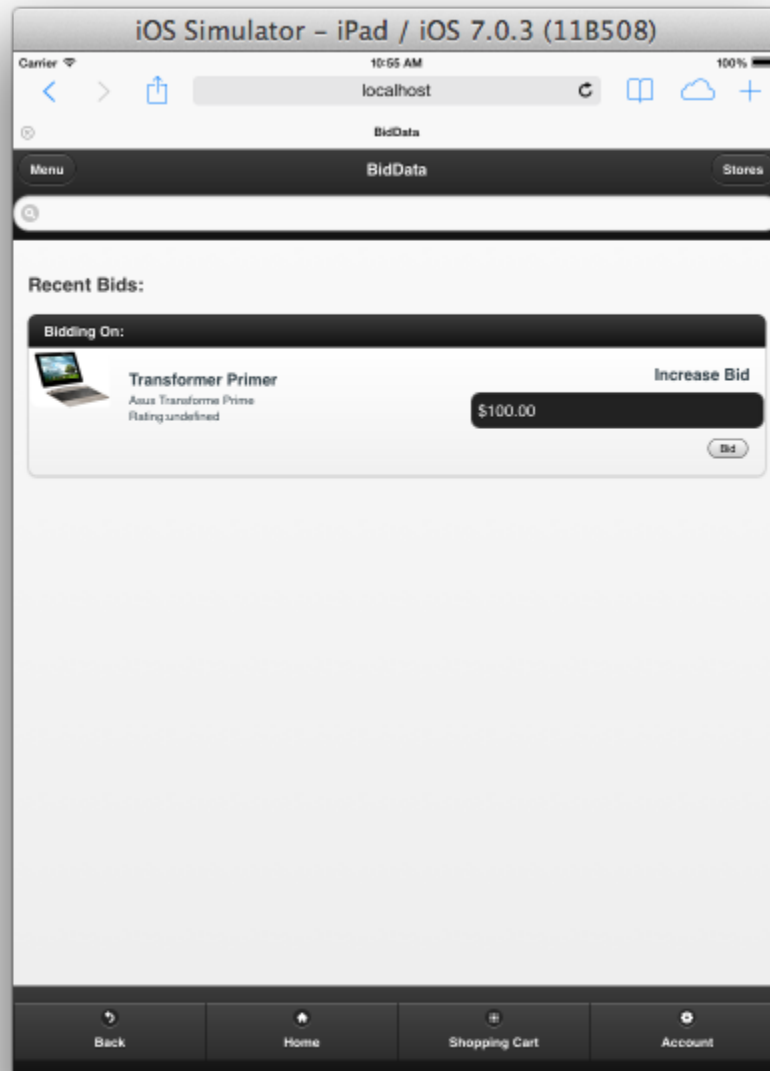
- b. *Log In:* Once the user is registered, he or she can Login by clicking the button in the upper left corner of the header bar.



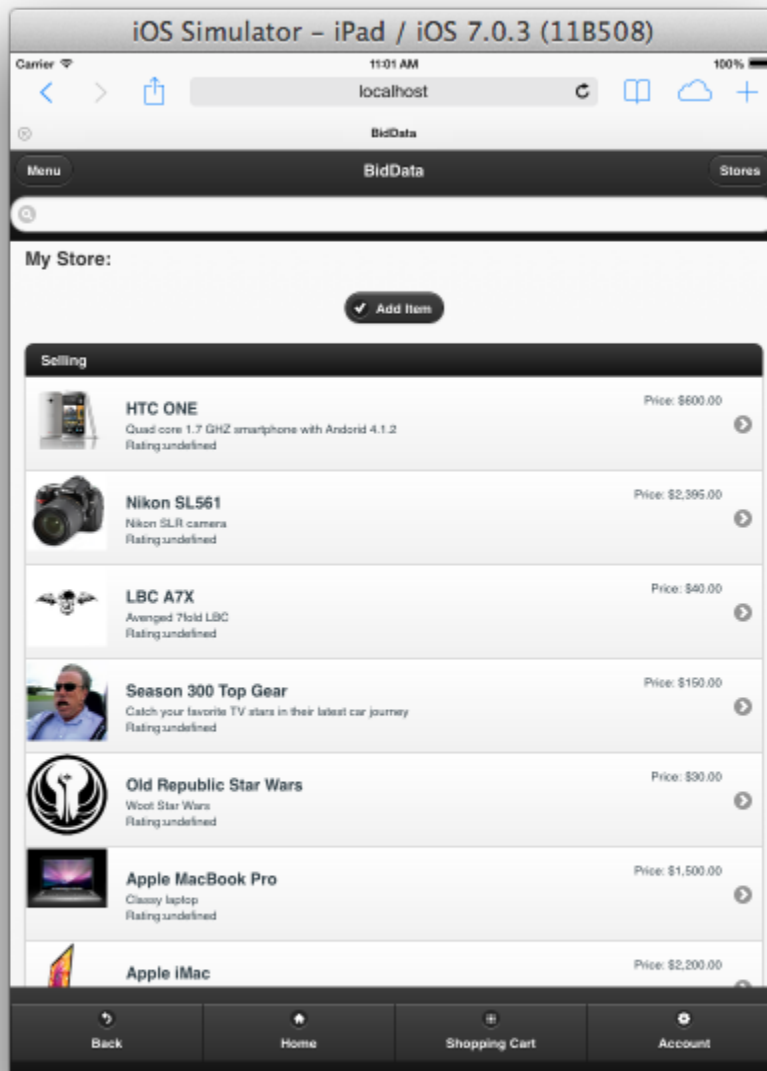
- c. *Browse for categories:* The user may be able to browse the categories by selecting the Store button located the right top corner of the window. This will open a menu located at the right of the window which will display each of the available stores/categories.



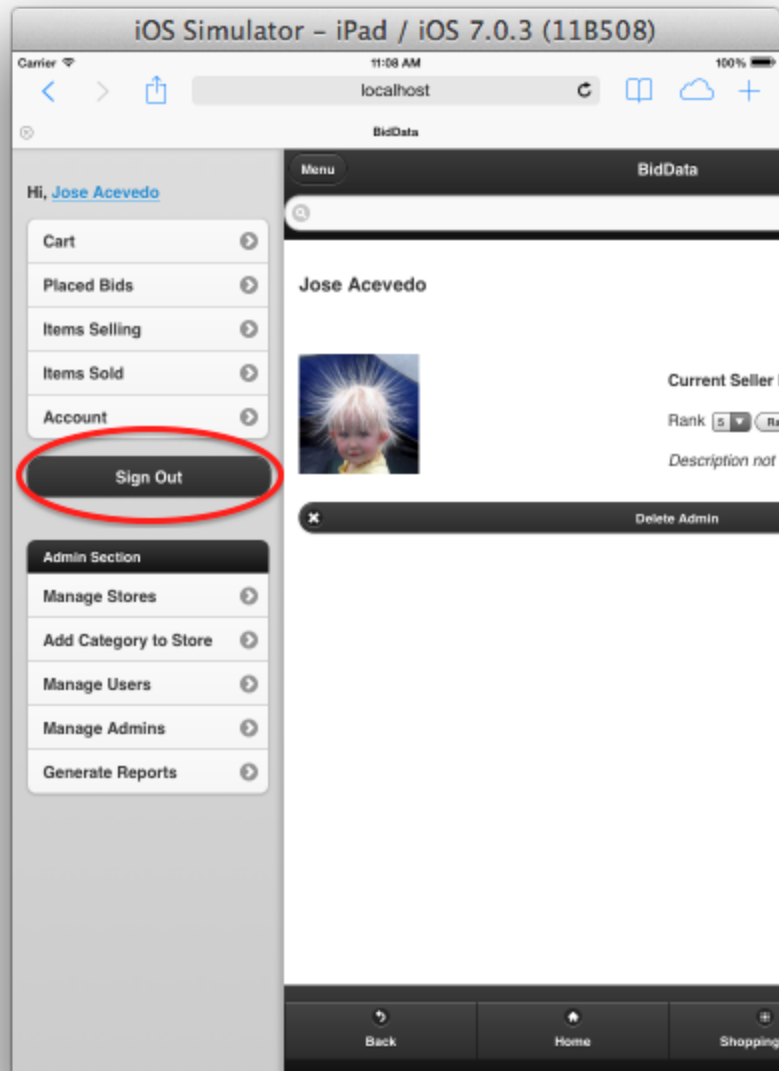
- d. *Bid on products:* To place a bid on a product, click on the item's bid box and type in the amount you want to bid. To track the bids you currently have placed you can go to the left panel menu and in there look for the *Placed Bids* button. In there the user will see a list of the products he or she is bidding on.



- e. **Sell Product:** To sell a product just click on the Menu button which it's located at the top left corner of the window. Afterwards, navigate to the Items Selling button. This will display all the items that the user is selling. if you want to add a new item for sale, simply click on the Add Item button. This will present a form that will enable the user to add a new product.

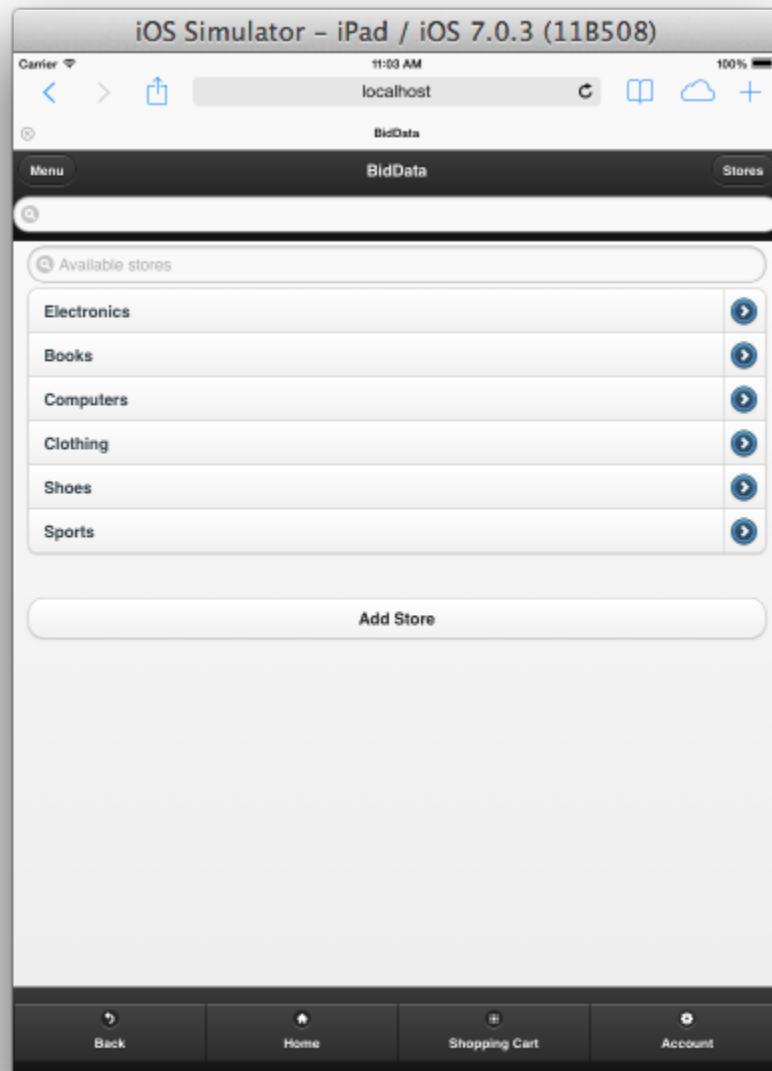


- f. *Log out*: When the user is logged, the left panel will harbor the *Sign Out* button. If the user wishes to log out just click the button and the server will clear your session.

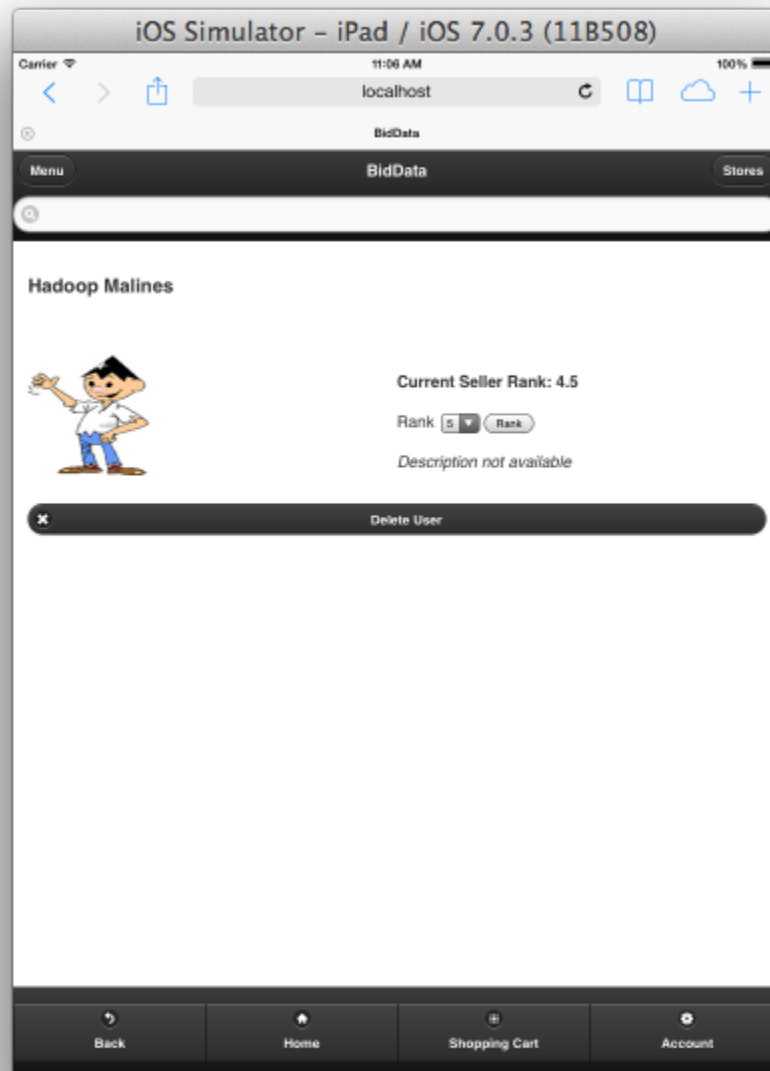


- g. Add product to shopping cart:* To add a product to the shopping cart, click on an item that is for sale and you'll see a button to add to your cart. Click it and an item will be added to your cart.

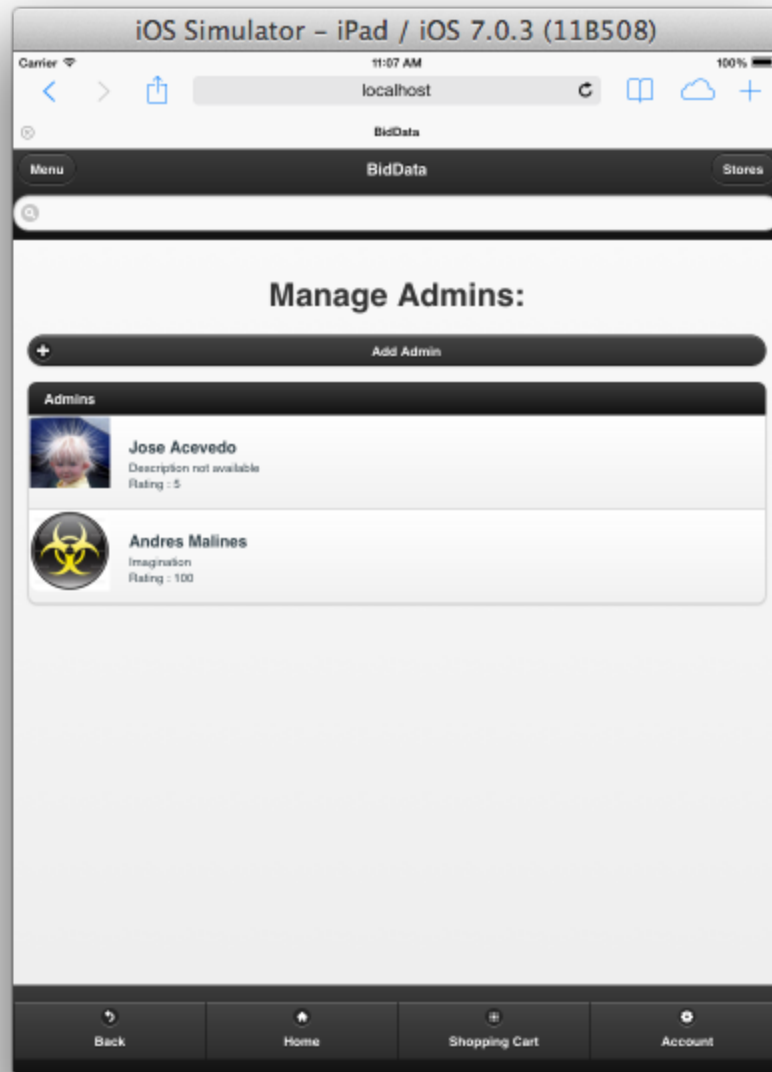
2. *Administrator* When a user is also an administrator there are new extended operations. These operations are listed below.
- Add Stores*: The admin has the ability to see all the stores currently in the database. This gives him power to add more stores to the website.



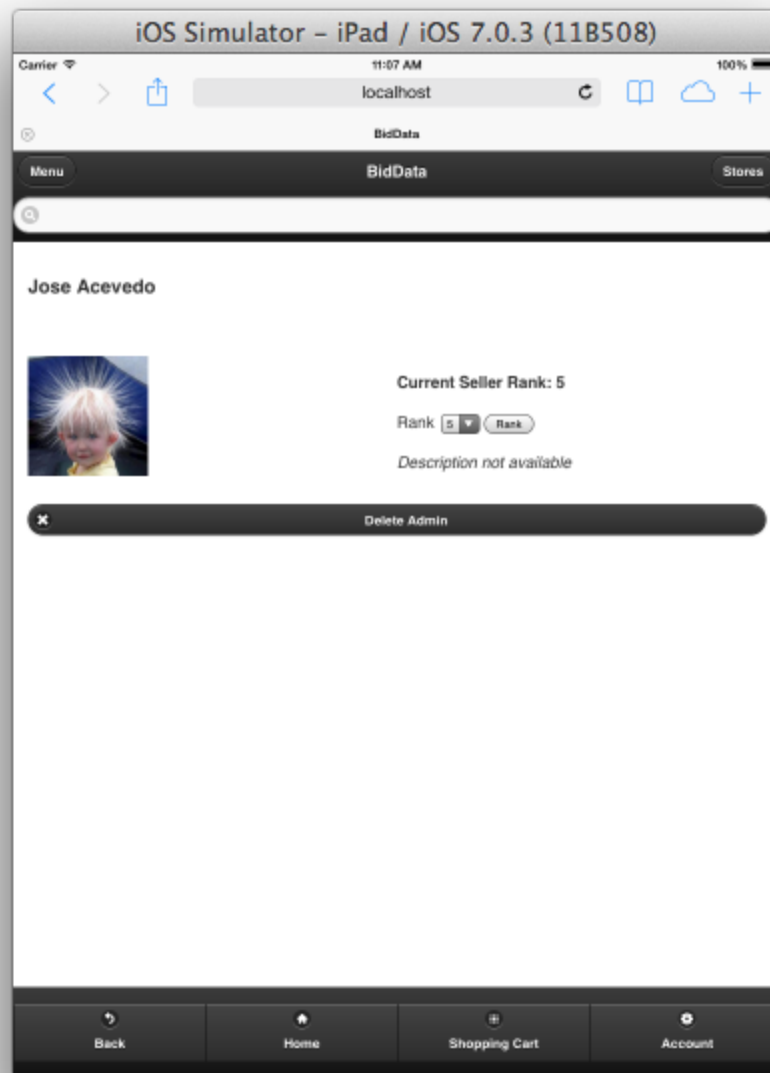
- b. *Remove Users:* Users in the database can be removed only by an admin. For example if a user is flagged for misconduct, cheating or any other wrong behavior. To remove a user, go to the *Manage User* section on the left panel. Here, a list of users currently registered is shown. To delete one, simply click on the user and there will be a button to delete that user.



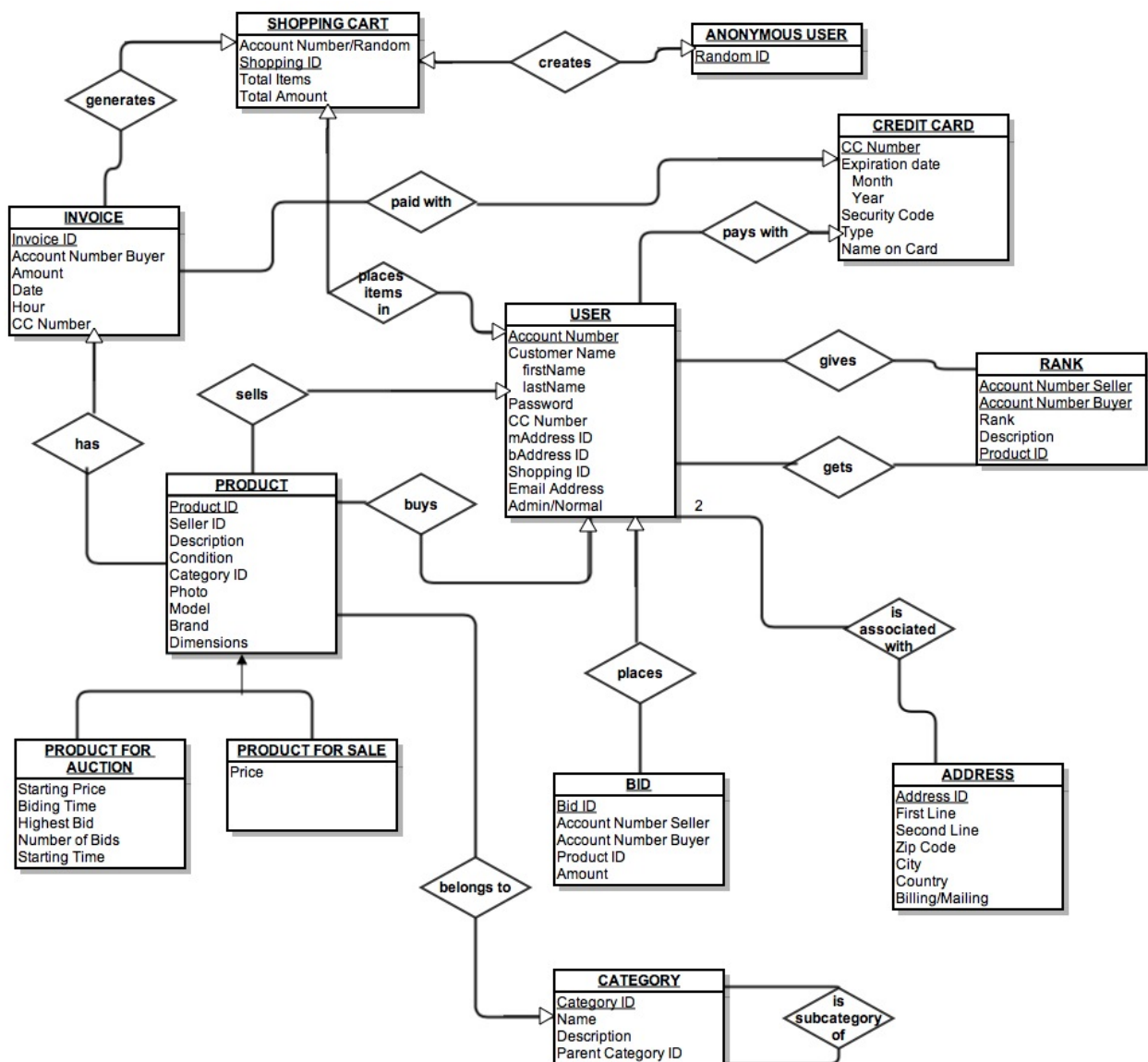
- c. *Add More Administrators:* In order to add a new administrator, first the user needs to be logged in as an administrator. Then, he can access the Manage Admin button located in the Menu which will display a list of all of the Administrators of the page. Here, the user can select the Add Admin button. This button will display a form that will grant the user the ability of adding another user.



- d. *Remove Administrators:* In order to remove an administrator, the user must first access the Manage Admins menu, as previously explained, and then select the profile of the Admin he wishes to delete. There, the user can, simply, press the Delete Admin button.



- ### ***Entity-Relationship: Model***



Entity-Relationship: Table Descriptions

1. *address* : represents either the mailing or billing address of an user
2. *anon_user*: If we have an anonymous user, his information is stored in this table.
3. *auction_product*: Products listed in this table have fields specific to auctions like *bid_id* and starting price of the product.
4. *bid*: This table holds information about the amount credited to a product through a bid
5. *buys*: This table records what products have been bought by which user.
6. *category*: Table used to hold information about the categories.
7. *credit_card*: In this table, the information of the credit cards that the user has are stored.
8. *has_invoice*: This table stores the products that are tied to an invoice.
9. *invoice*: Represents a shopping cart or won items once they are bought.
10. *items_in_cart*: This maps all the products that are in a shopping cart to that shopping cart.
11. *product*: This table is generalized to include all the information that a product should have regardless of its type, be it for sale or for auction.
12. *rank*: This table record the rank of each of the users, regardless if it is an administrator or a regular user.
13. *sale_product*: This is a specialized table that holds information specific to an item that is for sale. This includes the starting price.
14. *shopping_cart*: This table holds the information about a shopping cart of one user. This is used to link the items the user has in the shopping cart to that user.
15. *web_user*: The most significant entity is *web_user*. A user can buy and sell multiple products, but a product can only be bought and sold by a single user. A user is associated with 2 addresses (mailing and billing) and with one credit card. A user can rank another user after buying something from that user. This ranking system will give people an idea of how good of a seller is a user.

Relation	Entities Associated	Description
generates	invoice - shopping cart	When a <i>User</i> places an order, its corresponding <i>Shopping Cart</i> generates an <i>Invoice</i> which describes the placed order.
has	invoice - product	Every <i>Invoice</i> has a series of <i>Products</i> associated with it. An <i>Invoice</i> can have one or more <i>Products</i>
sells	user - product	A <i>User</i> can sell <i>Products</i> . It's important to note that we make a distinction between items for auction and items for sale.
buys	user - product	A <i>User</i> can buy <i>Products</i> which are sold by other <i>Users</i> .
places	user -bid	When a <i>User</i> wants to participate in an auction, she/he places a <i>Bid</i> on the <i>Product</i> . At the end of the auction, the <i>User</i> with the highest <i>Bid</i> amount will buy the <i>Product</i> .

belongs to	product - category	A <i>Product</i> has a category field which designates where it belongs. This values comes in the form of categories concatenated with a delimiter to establish the whole inheritance. This is due to the relation between categories since a category can have child categories.
is subcategory of	category - category	A <i>Category</i> can be child of itself. This means that a category can have a subcategory and that subcategory can have its own subcategories.
places items in	user - shopping cart	A registered <i>User</i> can place <i>Products</i> in a <i>Shopping Cart</i> and buy them when he or she is ready.
paid with	invoice - credit card	The <i>Invoice</i> is like a receipt from an order placed by a <i>User</i> . The total amount of money from an <i>Invoice</i> is paid by an <i>User</i> using its <i>Credit Card</i> .
creates	shopping cart - anonymous user	An <i>Anonymous</i> user can create a <i>Shopping Cart</i> by selecting items for sale. In order to make the purchase, the <i>Anonymous</i> user must first create an account to become a <i>User</i> .
pays with	user - credit card	A <i>User</i> is associated with a <i>Credit Card</i> which him or her uses to pay for bought and won items.
gives/gets	user - rank	A <i>User</i> can get or give a <i>Rank</i> when certain events occur. A <i>User</i> that buys a <i>Product</i> from another <i>User</i> can give that user a <i>Rank</i> based on how good was his or her experience with the bought <i>Product</i> .
is associated with	user - address	A <i>User</i> has a an <i>Address</i> . This <i>Address</i> can be either the billing or the shipping address. This distinction is established through a field in the <i>Address</i> entity.