|  |
| --- |
| ICOM5016 Project |
| Phase II Report |
| LHL Mobile App |

|  |
| --- |
| Heidi Negrón Arroyo, Lexter Seda, Luis Tavarez  11/11/2013 |

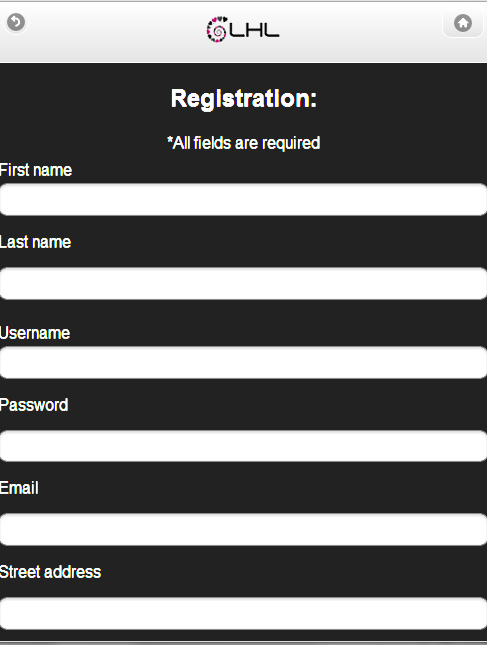
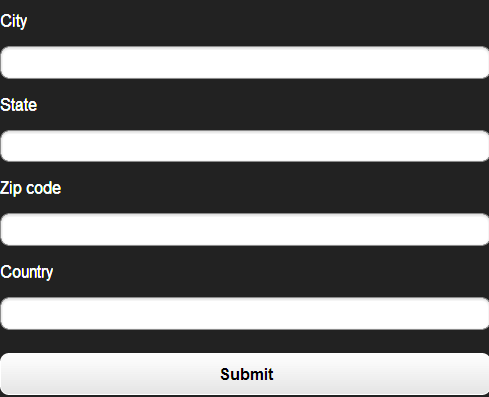
**Introduction**

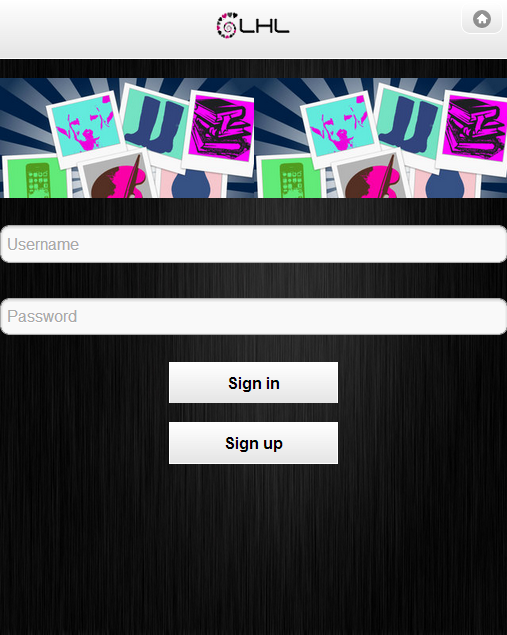
In this mobile application we design, implement and test a mobile app styled as eBay. It has been called LHL. This site is implemented using database and other technologies. The client side is implemented in jQuery Mobile. The web/application server is implemented with Node.js and the database server is PostgreSQL. The communication between client and server occurs through a REST API. In this report we present a task analysis of each of the operations users and administrators are able to perform. Additionally, screenshots are provided to demonstrate how the user can complete the described task. Finally, a description of the ER diagram prepared for this mobile application is included in the report. We describe the diagram, the ER mapping and conclude with a description of each of the tables and attributes that we consider essential for the purpose of the project.

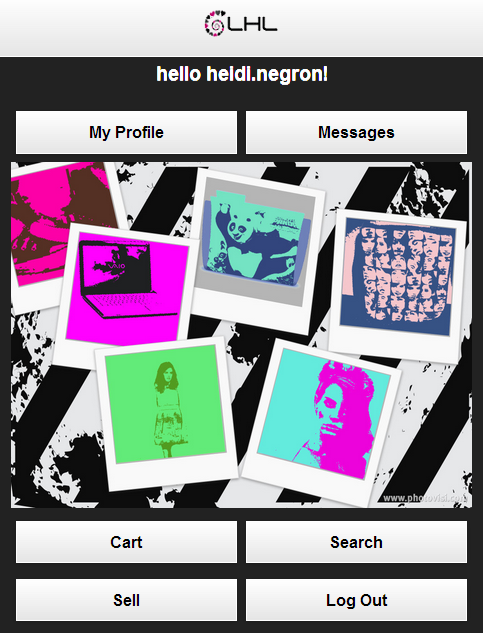
**Task Analysis**

The tasks of the mobile application LHL can classified in three main components. These components are: Account – related tasks, Sales/Auctions-related tasks and Administrator Tasks. A regular user is able to perform the following tasks:

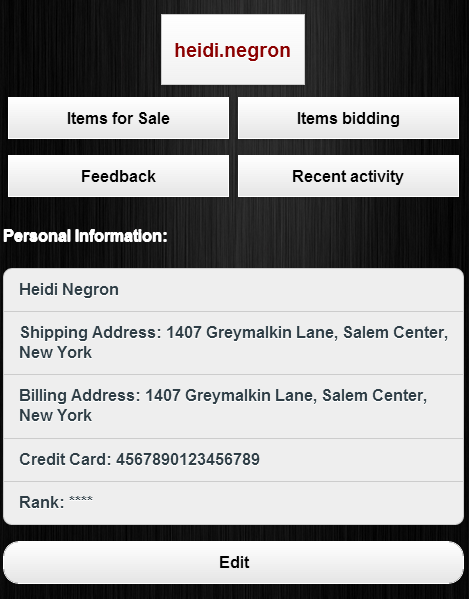
1. Account:
2. Create an account: A user creates an account. He/ She is required to provide name, last name, address, password an email.



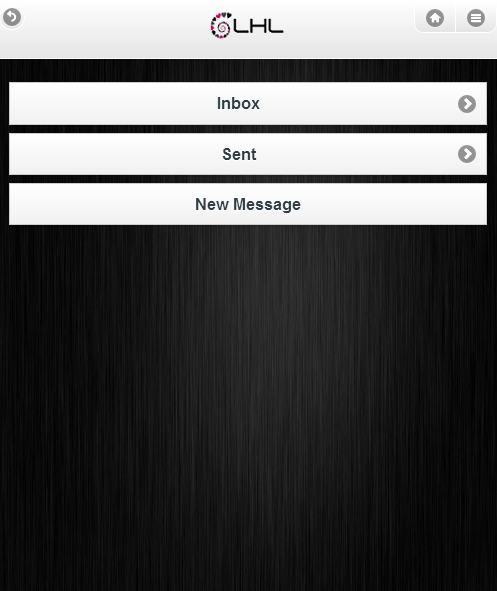
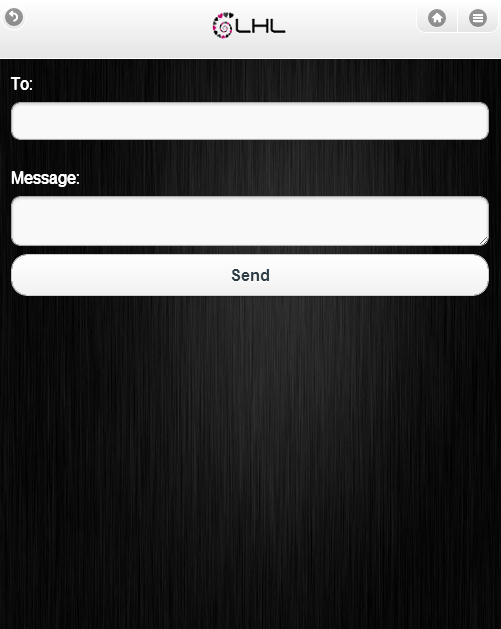
1. Login: User needs to provide a valid username/password combination.
2. Homepage: the main page of a user lets them access the main features of the mobile app: Profile, sales, cart and messages.



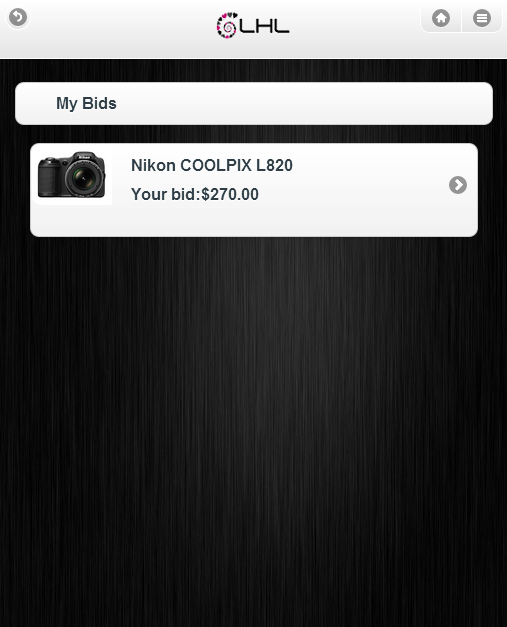
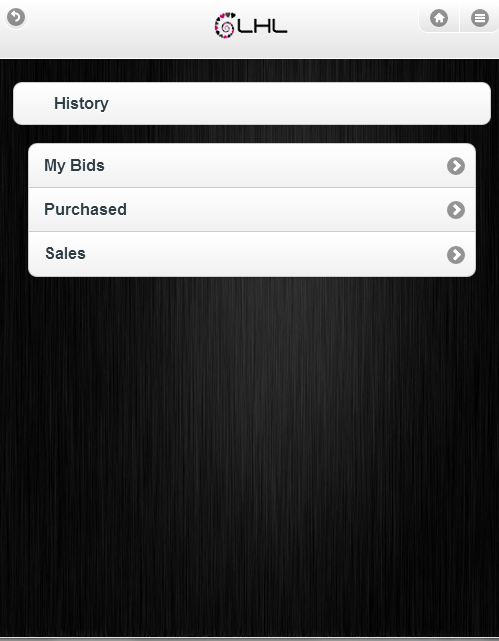
1. Profile: The user can see his profile at all times and edit its personal information.

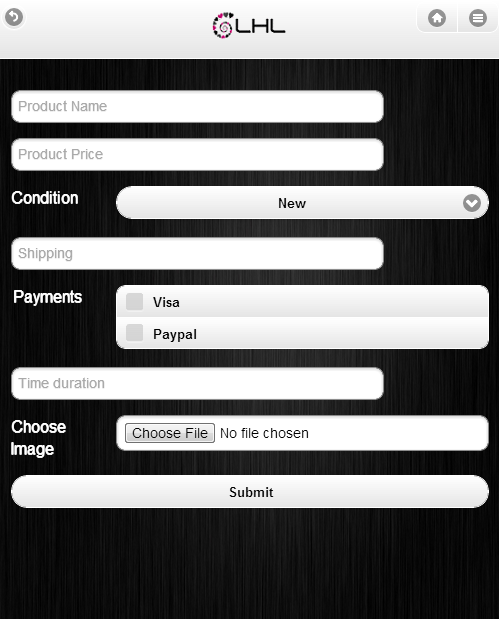


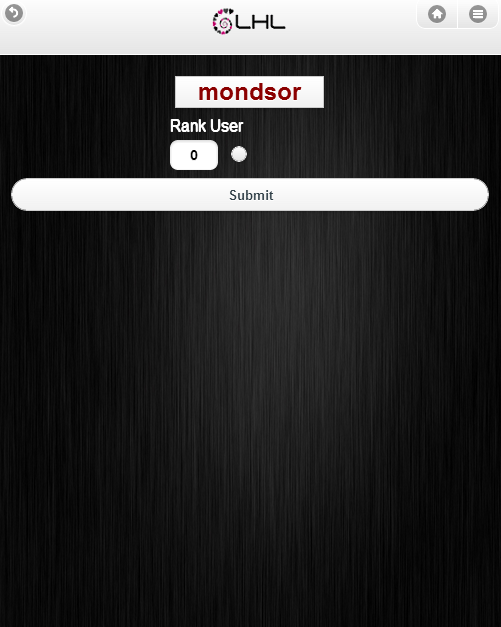
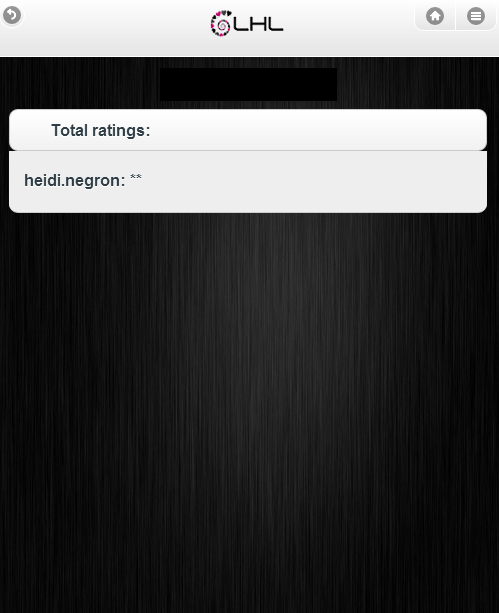
1. Messages: The user can send messages and see the messages he has sent/ received.



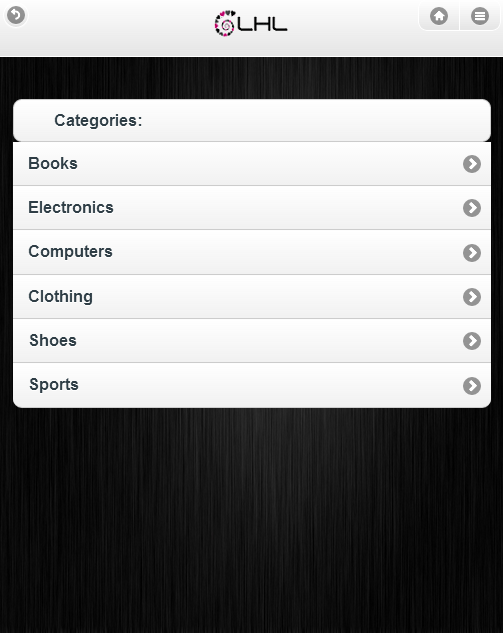
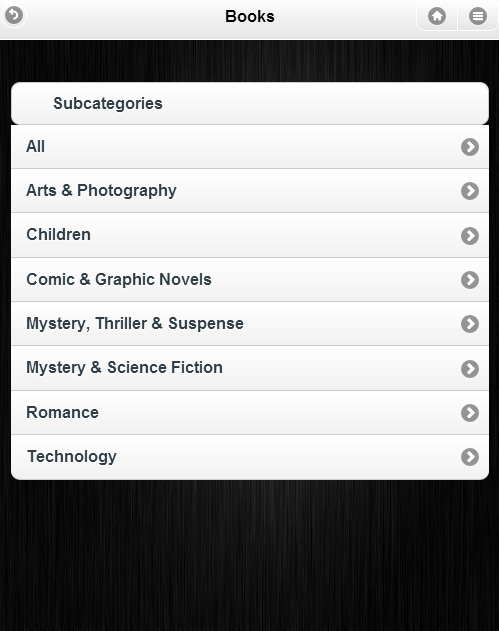
1. History: A user can see his recent activity including the bids he has placed, the current sales and the products he has purchased.

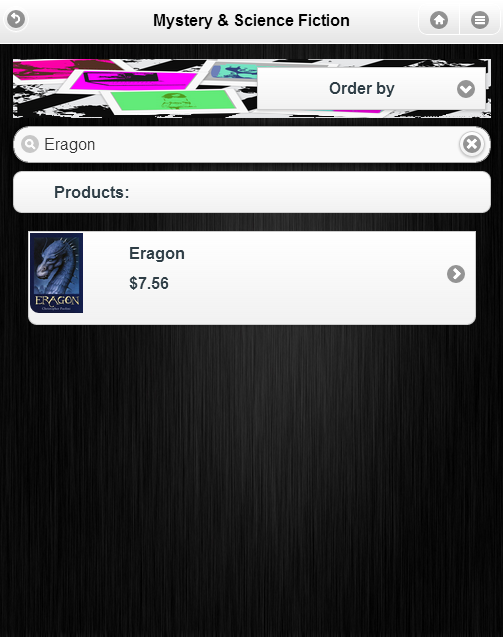


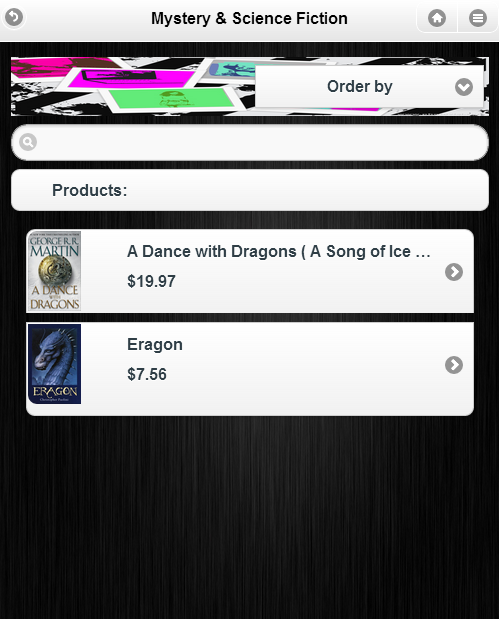
1. Create sales/auctions. It is easy for the user to create an auction/ buy-now product. He needs to specify the name, price, condition and payment methods of the product he desires to sell.
2. Rank other users: User can rank other user based on the experience of buying from the user.



1. Sales/ Auctions task:
2. Browse categories of products available.

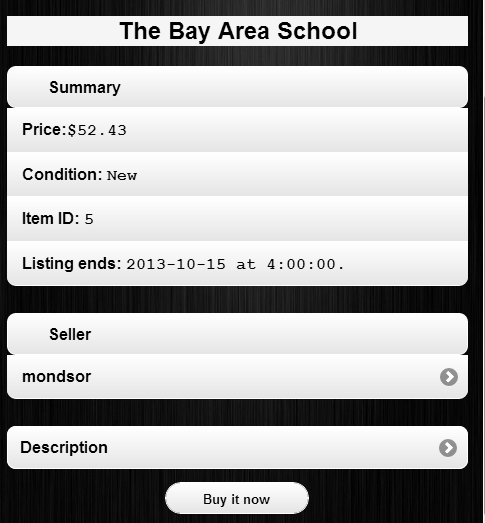


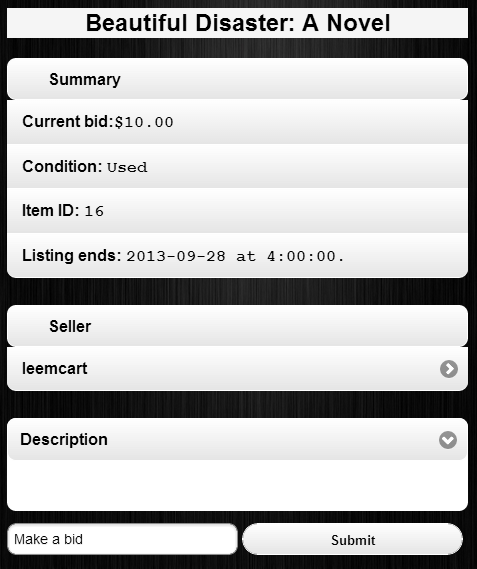
1. Search based on product name to filter results.



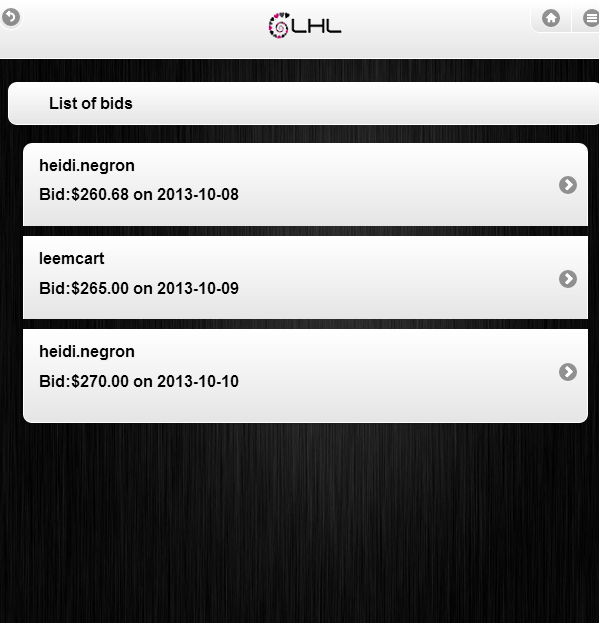
1. See description of a specific item. This description includes the summary of the product: price, condition, listing time and the item id. A description of specific details pertaining to the product such as color / model / brand is provided. The seller of item is listed and users can access his profile from there.



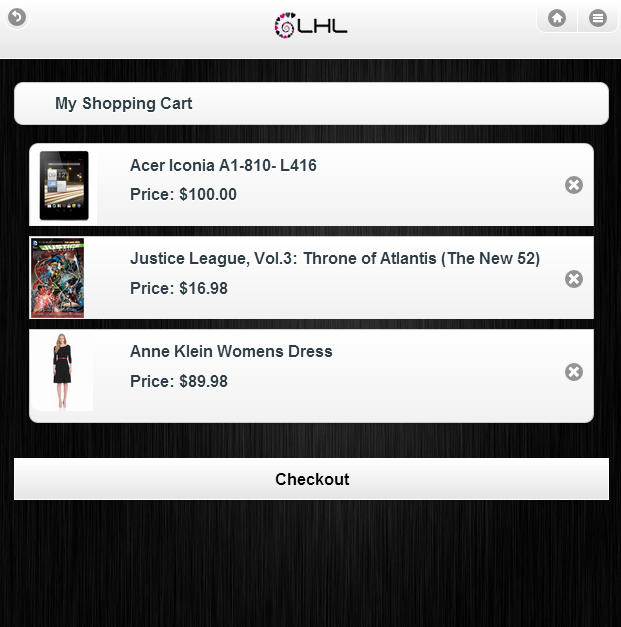
1. Place bid and/or buy a product



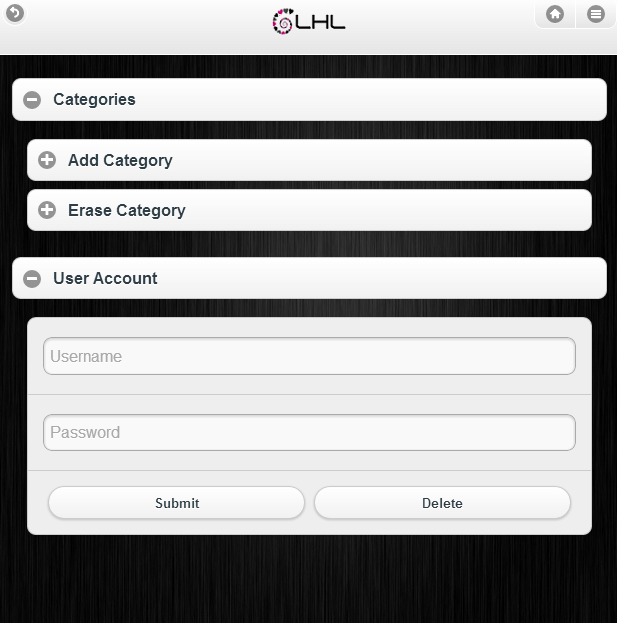
1. If the person is owner of the item, see a list of the bids placed. The username, the bid placed and the date of the bid is displayed.



1. Add item to a shopping cart and checkout.

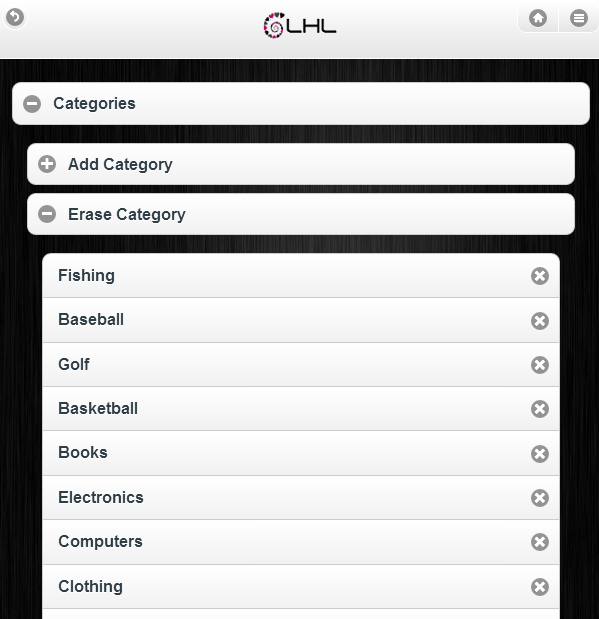


1. Get an invoice of an order placed.
2. The users with administrator rights will also be available to:
3. Create, view and modify accounts.



1. Create and delete categories of products.

The administrator can also delete products.



**ER-Mapping**

**Table 1: Category**

create table category(

catId bigserial primary key,

catName varchar(15) NOT NULL,

parentID bigint)

**Table 2: Address**

create table address(

addressId bigserial primary key,

address varchar(100))

**Table 3: DepositAccount**

create table depositaccount(

depositaccountId bigserial primary key,

bankaccountnumber varchar(17),

bankrouting varchar(9))

**Table 4: Account**

create table account(

accountId bigserial primary key,

username varchar(20) not null,

fname varchar(20) not null,

lname varchar(20) not null,

email varchar(30) not null,

apassword varchar(16) not null,

rank int not null,

shippingId bigint, billingId bigint,

depositId bigint, isAdmin boolean,

foreign key(shippingId) references address (addressId),

foreign key(billingId) references address (addressId),

foreign key(depositId) references depositaccount(depositaccountId))

**Table 5: CreditCard**

create table creditcard(

creditId bigserial primary key,

addressId bigint,

cardtype varchar(20) not null,

cardnumber varchar(20) not null,

securitynumber varchar(4) not null,

expdate char(5) not null,

foreign key(addressID) references Address (addressId))

**Table 6: Product**

create table product(

productId bigserial primary key,

catId bigint,

prodName varchar(80) not null,

condition varchar(4) not null,

description varchar(160) not null,

imagelink varchar(255),

foreign key(catId) references category (catId))

**Table 7: Auction**

create table auction(

auctionId bigserial primary key,

accountId bigint,

prodId bigint,

startingBid money not null,

startdate timestamp not null,

enddate timestamp not null,

foreign key(accountId) references account (accountId),

foreign key(prodId) references product (productId))

**Table 8: Bid**

create table bid(

bId bigserial primary key,

accountId bigint,

auctionId bigint,

bdate timestamp not null,

bammmount money not null,

foreign key(accountId) references account (accountId),

foreign key(auctionId) references auction (auctionId))

**Table 9: WinningBid**

create table winningbid(

bId bigserial primary key,

accountId bigint,

auctionId bigint,

bidDate timestamp not null,

bidammount money not null,

foreign key(accountId) references account (accountId),

foreign key(auctionId) references auction (auctionId))

**Table 10: Sale**

create table sale(

saleId bigserial primary key,

accountId bigint,

prodId bigint,

starttime timestamp not null,

endtime timestamp not null,

price money not null,

totalquantity int not null,

Ended boolean not null default false,

foreign key(accountId) references account (accountId),

foreign key(prodId) references product (productId))

**Table 11: Checkout**

CREATE TABLE checkout

(

checkoutid bigserial NOT NULL PRIMARY KEY,

creditid bigint NOT NULL, invid bigint NOT NULL,

totalprice money NOT NULL,

saleid bigint NOT NULL,

quantity integer DEFAULT 1,

FOREIGN KEY (creditid) REFERENCES creditcard (creditid)

FOREIGN KEY (invid) REFERENCES invoice (invoiceid)

FOREIGN KEY (saleid) REFERENCES sale (saleid))

**Table 12: Invoice**

CREATE TABLE invoice

(

invoiceid bigserial NOT NULL PRIMARY KEY,

buyerid bigint NOT NULL,

date date NOT NULL,

FOREIGN KEY (buyerid) REFERENCES account (accountid)

)

**Table 13: Message**

create table message(

messageId bigserial primary key,

senderId bigint,

receiverId bigint,

subject charvar(50) default'(No Subject)',

text varchar(150) not null,

date timestamp not null,

foreign key(senderId) references account (accountId),

foreign key(receiverId) references account (accountId))

**Table 14: Rank**

create table rank(

rankId bigserial primary key,

accountId bigint,

buyerId bigint,

stars int not null,

foreign key(accountId) references account (accountId),

foreign key(buyerId) references account (accountId))

**Table Explanation**

**Table 1: Category:**

This table stores each category, of the products to be sold, with its name and

It’s parent category if any.

**Table 2: Address:**

This table stores each address to be used for the accounts and credit cards.

**Table 3: DepositAccount:**

This table contains the bank account number and its bank routing number, for the

deposit account in which payments of sales will be stored.

**Table 4: Account:**

Here we have all the basic information about the accounts and the users.

**Table 5: CreditCard:**

This table has the information about every user's credit card. This table reference an

address which is the billing address of the user.

**Table 6: Product:**

This table stores only the basic information of a product such as category, name, condition,

description and image. It references category to obtain its category name.

**Table 7: Auction:**

This table contains the information on every auction and use the product table to look for the item being sold.

**Table 8: Bid:**

In this table we have the information for all the bids submitted to an auction.

**Table 9: WinningBid:**

Here we can find the information of every bid that has won an auction. This table is used

to connect an auction, that has ended, with its winner.

**Table 10: Sale:**

This table contains the information on every sale (buy it now) and use the product table

to look for the item being sold.

**Table 11: Checkout:**

This table stores the record of products bought at a sale and the creditcard that payed for it.

There is a value InvId which connects each checkout item to the invoice in which it was paid.

It also stores the quantity of each product and the total price paid.

**Table 12: Invoice:**

In here we connect the buyer to an invoice (which is created after he places an order).

**Table 13: Message:**

This table has all the messages sent from user to user with their corresponding subject, text (body)

and the time it was sent.

**Table 14: Rank:**

Here we have the data of all the ranking stars users have given to other users