A Report on the Database Design of e-Auction

Project Phase III - System Prototype

BY

Team Scheme

Sela Jung
Ju Hong Min
Yoonsung Son
Alexander Hershman

Table of Contents

1 Introduction……..4

2 System Functionality…….……..…4

3 Conceptual Design…….13

3.1 Design Description…...……...…….……...13

3.2 E-R Diagram……..……...…….18

3.3 Schema Creation……..19

3.4 Normalization……...……20

4 Implementation of System Prototype……23

4.1 Desired Functionality ……..23

4.2 External Component……24

4.3 Internal Component…….26

4.4 Improved Specifications……..26

4.5 Prototype Success……27

5 System Issues…….…28

6 Project Plans……...…….……..29

6.1 Milestones……...29

6.2 Deliverables……29

7 Conclusion…….30

7.1 Lesson Learned……30

7.2 Decision Making……..30

8 Appendices…….……32

8.1 Progress Report…….……..32

8.1.1 Group Report……32

8.1.2 Individual Report……..34

8.2 Changes from Last Report……...……34

8.3 SQL……..35

8.3.1 Table Creation……..35

8.3.2 Implementation……35

8.4 Internal Programs…….35

1. Introduction

The purpose of this project is to create a website capable of allowing users to search for, sell, and bid on various items. Other features offered by this website would exist for the facilitation and improvement of customer-seller interaction, ease of finding desirable items to bid on, and simplicity in regard to the auctions themselves. These three considerations form the framework of our requirements. This report fulfills this framework by stating and analyzing the website requirements, creating a conceptual design to reflect these requirements and the constraints set by them, translating this conceptual design to logical schema, refining the proposed schema design, and addressing any relevant system issues. In addition to this, this report further details an overview of a system prototype based on this design and its specific implementation.

2. System Functionality

Basic Requirement Overview:

- New users can register either as companies or personal users
- Users can create auctions which last for two weeks
- Users can search for items by keyword and by category
- Users can organize search results using specific conditions (i.e., lowest price, highest price, posted soonest, posted latest, etc).
- Users can bid on items
- Users can buy an item immediately using BuyItNow
- After an auction is complete, buyers can rate sellers and sellers can rate buyers
- Regular Auction statistics are taken which can be reviewed by administrators

- e-Auction will deliver items only after the item has been delivered and the charge has gone through, eventually sending a check to the seller. (See "Delivery" section below)
- Popular items will be listed on the front page
- Users can view a user-specific watch-list that contains items they are watching
- Users can view a user-specific bid-list that contains items they have bid upon
- Users can view a user-specific sales-list that containing open auctions they have created

NOTE: The four last requirement listed above are new features that have been added to the original specifications.

<u>*Requirement Specifications*</u>:

Front Page:

- There will be a front page which shall contain the following:
  - A search text field
  - A list of categories which will expand to a list of subcategories if clicked
  - A 'login' link, a 'register' link, a 'FAQ' link, and a 'Contact us' link
  - A list of popular items

Popular Items Listing:

- On the front page, several popular items will be available for view. See specification sections below for information regarding what is displayed by each item entry in a list.

Search Functionality:

- The search text field will have a drop-down menu specifying category and options listed

below will specify how to organize search results

    - The default outcome of searching for an item will output a list of items which match the keyword the most, in specific category of choice
        - If no category specified, all categories will be searched
    - The types of organization are as follows: by highest price, by lowest price, auction ending soonest, auction ending latest, and most relevant (which is the default search - see above). All organizations will be based on results from keyword search.
    - Searching only takes place when the 'search' button next to the search text field has been pressed or if 'enter' on the keyboard has been pressed when the cursor is located in the text field
- For the list of categories, selection of a low level subcategory will yield a search result which displays all items found in that category, organized by popularity
- A search page will display a list of items, each item on the list displaying the following information: a picture (if available), item name, item description, auction posting time and date, end of auction time and date, current highest bid, and BuyItNow price.

Logging In and Personal User Page:

- When the 'login' link is clicked, a web page containing a field for the username and a field for the password will be loaded. A user can enter in the username and password provided he has registered, and thus access his own particular personal user page.
- A personal user page will contain a watch list containing items that are being watched but that he hasn't bid on, and also a list of items containing the history of items that he

has bought.

- The personal user page additionally contains a list of auctions created by the user which are still open.
- The personal user page is distinct from the User Information Page (see below) in that the Personal User Page can only be seen by the user himself, whereas the User Information Page is shown to other users.

Bid List, Watch List, and Sales List:

- The bid list will be a list available on every Personal User Page that lists all open auctions a user has bid upon
- The watch list will be a list available on every Personal User Page that lists all items that user is watching but has not bid upon
- The sales list will be a list available on every Personal User Page and every User Information Page that lists all auctions a user has created. For distinctions regarding the Personal User Page and User Information Page, either see "Logging in and Personal User Page" section above or "User Information Page" below

Registration:

- When the 'register' link is clicked, a web page containing two radio buttons appears which allows the user to select between being a 'company' or 'personal user'
- After the completion of the selection above, if 'personal user' is selected the following text fields appears: username, password, e-mail address, name, address (which consists of street, city, state, and zip), phone number, credit card information (including type of credit card, credit card number, expiration date), age, gender, annual income, and etc.

    - All fields will be required with the exception of date of birth, gender, and annual income, which will be optional. This data will be sent to telemarketers (unbeknownst to the user).
    - A user has the option to enter several different credit cards
- If 'company' was selected, the following information will be asked for in a way similar to how information was obtained for the personal user account: company name, address, point of contact (person's name), phone number, company category, and revenue.

Creating Auctions:

- Each user, whether a company or a personal user, will be capable of creating an auction. Creation of an auction will be available from the user page, by clicking a 'create auction' link.
- The following data will be specified for the creation of an auction on a separate page with text-field for each: An item name, an item description (no more than 400 characters), item location, BuyItNow price, reserve price, a URL, category, and starting price.
    - Of the above fields, the reserve price, BuyItNow price, and URL will be optional.
        - The URL will be a link to the company's site, the BuyItNow price will be a price displayed to the user, which allows him to buy the item immediately if bidding that exact price or higher, and the reserve price is the minimum price the seller is willing to sell an item for (hidden from the

users).

    - The specified category will place the item in that category for users that search categorically

- When an auction is created, the auction item information becomes available in auction searches. Each item will display the information as described in the Search Functionality section above.

Auction Item Page:

- The auction item page will contain further information regarding the item. In addition to the information each item displays in the search query (see Search Functionality section above), the following information will be displayed: seller rating, a link to the seller page, item location, and original price.
    - Additionally, text fields and related buttons will allow the user to place a new bid upon the item, or to BuyItNow.
    - The user also has the ability to tag an item so that it appears on his watch list
    - If the user buys the item immediately, a page will appear alerting the customer as to the fact that he has bought the item. See the "Finalizing a Sale" section for further details,
- The auction item page is accessed by clicking on the item name found in a search query.

User Information Page:

- The user page is available from an item page created by the user. The user page contains all reviews of the seller as well as a list of all items being sold by the seller currently in open auctions. The overall rating of the user will also be displayed.

- The user information page is distinct from the personal user page as the user information page is viewed by other users, and the personal user page is viewed by the user himself.

Bidding:

- This website employs simple bidding in that each bid must be 5% higher than the last bid.
- When a user bids, if he has multiple credit cards, a page will come up asking user to specify which credit card to use.
- If the user bids upon an item, it will then be present on his bid list

Finalizing a Sale (Ending an Auction):

- After a two week period from posting an auction, an auction can end in three ways:
    - 1: The auction failed to produce a bid high enough to exceed the reserve price, or no bids were placed. Thus, the data from the auction is removed from the system (i.e., it cannot be found through search queries or on a seller's sale list or a user's wachlist/bidlist).
    - 2: A bid has been made as of the exact time the auction ends which exceeds the reserve price. If the seller did not specify a reserve price, the reserve price should be treated as "$0".
    - 3: A user buys the item immediately using the BuyItNow option
- When an auction ends, contact information for the buyer is sent to the seller's e-mail address and contact information for the seller is sent to the buyer's e-mail address. This e-mail also alerts the buyer that he won the auction.
- When an auction ends, e-mails are sent to every person who bid in an auction

containing information about the highest bid and the bidder who placed it.

- See "Delivery" section below for information on how items are delivered.
- When auction is successful, information about successful and voided delivery must be stored for at least six months after e-mails are sent to all parties notifying them of the outcome.
    - Information regarding when information was sent must be maintained

Rating and Review System:

- After an auction ends, until the sale is entirely finalized, an option to rate and review the seller/buyer becomes available on both the buyer's and seller's user information pages.
    - Only the seller can review the buyer and only the buyer can review the seller during this period of time
    - Ratings are between 1 and 5, with 5 being the best and 1 being the worst.
        - The overall seller rating will be the average of all ratings he has received
    - Reviews cannot exceed 250 characters.
- Only one review and rating can be submitted for each transaction

Delivery:

- The e-Auction company will serve as a third party to ensure that all transaction are properly accomplished. Thus, the seller will ship the item to e-Auction, and e-Auction will charge the buyer's credit card. If the charge is successful, the item is forwarded to the buyer and a check is mailed to the seller. If the item does not arrive in two weeks, or the charge does not go through, e-Auction declares the auction void and, if necessary, returns the item.

FAQ and Contact Us:

- The FAQ page will contain frequently asked questions.
- The Contact us page will contain relevant e-mail addresses and phone numbers for customer support.

Auction Statistics:

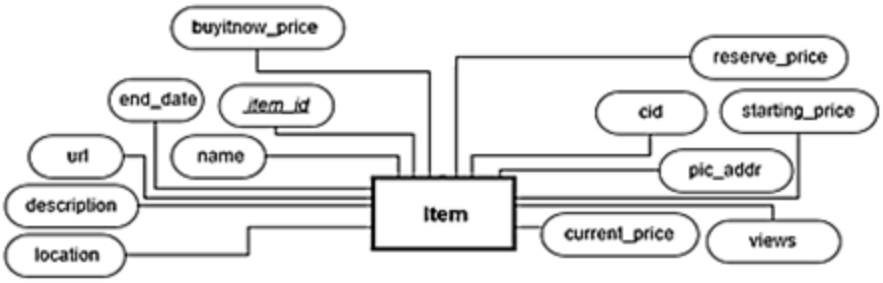
- Auction statistics will be kept for each category, with the following metrics measured: number of auctions, percent successful, percent not sold, and percent cancelled.
- This report will be generated at the end of every week.

3. Conceptual Design

3.1 Design Description

The e-Auction database has 8 entities: Item, Address, Card_info, User, Company, Individual, Statistics and Category_Map. It also has 8 relationships: Sells, Bids, BuyNow, Watches, Lives, Has-a, and Views.

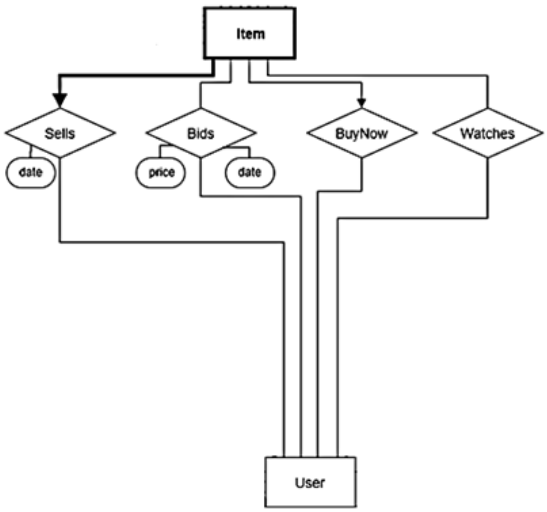
Item Entity:



The item entity contains the following attributes: item_id, name, end_date, url, description, location, current_price, views, pic_addr, starting_price, reserve_price and cid. The primary key for this entity is

the item_id, and cid stands for ‘category id’ which will be discussed at length in the ‘categorization’ section below. Views represent how many views an item has received and pic_addr refers to the address of a picture stored on the system. All the other attributes are clearly represented by their names, further discussion about can be found in the requirement analysis section.

User-Item Relationship:

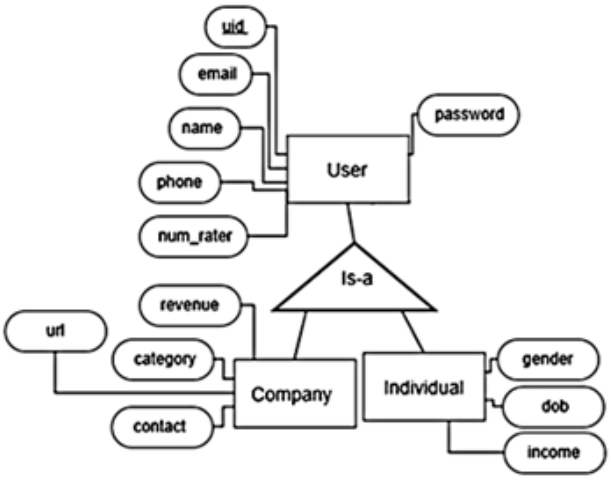


Users are capable of performing four separate item-related transactions on e-Auction: selling items, bidding on items, buying items immediately, and adding items to his or her watch list.

Since every item is being sold, there is total participation enforced between the Item entity and the Sells relationships. Since many items can be sold by multiple users, but each item can only be associated with one user, a one-to-many relationship is enforced through a key-constraint between Item and Sells. The

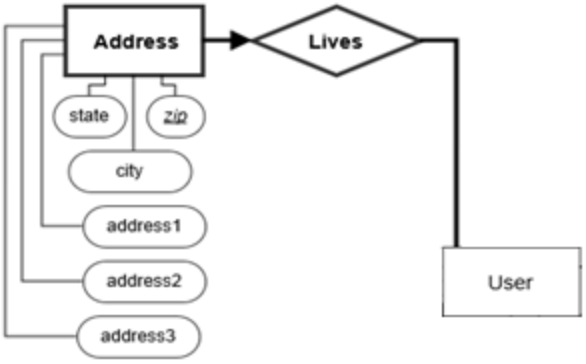
relationship between Item and users is represented above as a many-to-many relationship, as a user can bid on many items and an item can be bid on by many users. Total participation is not enforced as users have no obligation to bid on items. The BuyNow relationship allows a user to buy an item immediately. Since multiple users can buy multiple items, and only one item can be bought by one user, a one-to-many relationship is enforced through a key constraint between Item and BuyNow. The Watches relationship allows a user to watch items. Since many users may watch many items, and many items may be watched by many users, a many-to-many relationship is enforced without any participation.

User, Company and Individual Entities:



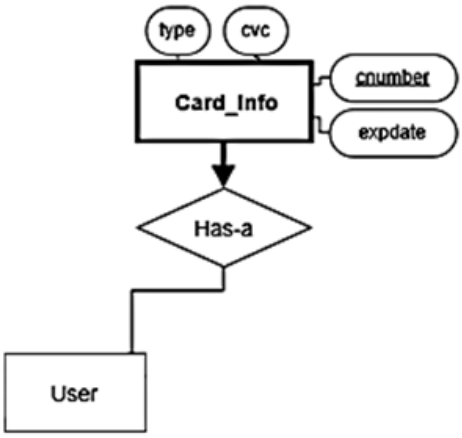
The User entity contains the following attributes: uid, e-mail, name, phone, num_rater, and password. The uid represents the primary key in this case, and num_rater represents the number of raters. The Company entity contains the following attributes: the url, revenue, category, and contact. It has a “IS-A” relationship with User. The Individual entity contains the following attributes: gender, dob (date-of-birth) and income. It has a “IS-A” relationship with User.

User-Address Relationship:

As the Address entity 

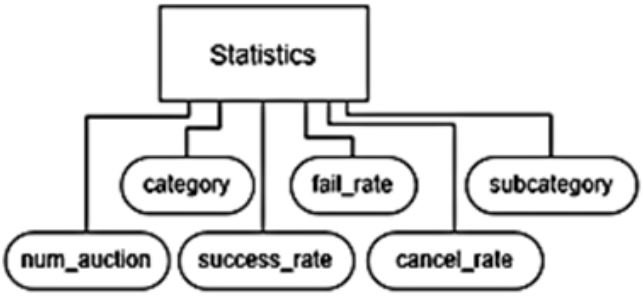
has no primary key, it is a weak entity. Also, as multiple addresses can be held by a user, it has a key constraint between lives and address.

User-Card_Info Relationship:



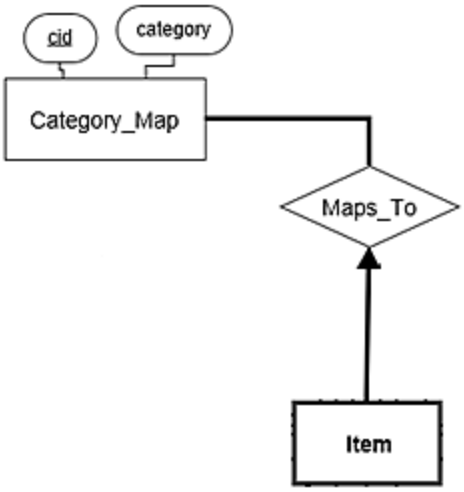
As the User entity can have multiple credit cards, there exists a key constraint between the Has-a relationship and the Card_Info entity.

Statistics Entity:



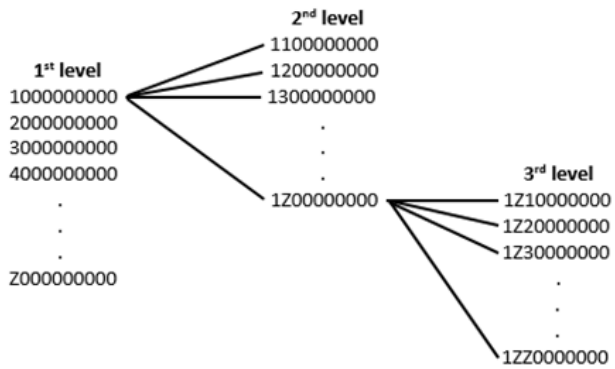
Every time users succeed or fail to buy items, the events are recorded in an independent entity called Statistics. The data in Statistics Entity is refreshed every seven days.

Categorization System:



Each item is given a ten character category id (cid) which contains information regarding all categories and subcategories implicitly. Each character position in this cid string is capable of being in that ranges

0-9, a-z, or A-Z. A zero represents an empty space, whereas any non-zero character specifies a certain category. A non-zero character in the first position specifies a first-level category, a non-zero character in the second position specifies a second-level category, and so on. Since there are ten characters in the category id, a category depth of ten is possible. A category mapping table keeps track of all cid-category associations.



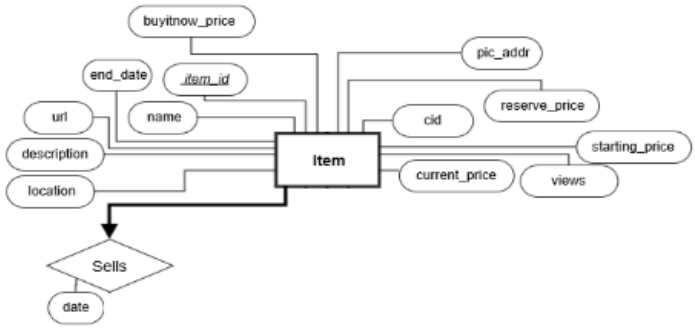
3.3 Schema Creation

The exact SQL code and discussion regarding it for the following schema definitions is found in the Appendices (7.3). High-level logical schema are listed below.

For *Category_Map* entity:

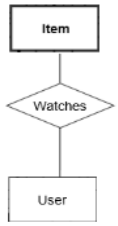
In SQL: *Category_Map*(cid:string, category:string)

For *Item_Sell_by* entity:



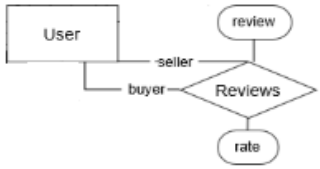
In SQL: *Item_Sell_By*(item_id:integer, name:string, cid:string not null, end_date:date, reserve_price:real, starting_price:real, current_price:real, buyitnow_price:real, views:integer, url:string, location:string, description:string, uid:string, date:date)

For *Watches* relation:



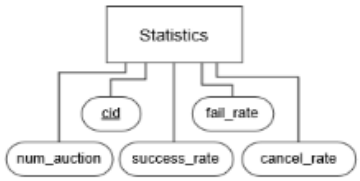
In SQL: *Watches*(uid:string, item_id:integer)

For *Reviews* relation:



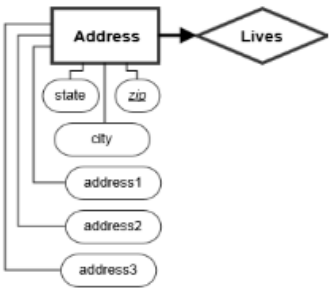
In SQL: *Reviews*(rater_uid:string, ratee_uid:string, review:string, rate:integer)

For *Statistics* entity:



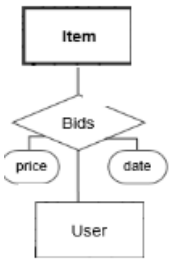
In SQL: *Statistics*(cid:string, num_auction:integer, fail_rate:real, success_rate:real, cancel_rate:real)

For *Lives_Address* entity:



In SQL: *Lives_Address*(state:string, zip:string, city:string, address1:string, address2:string, address3:string, uid:string)

For *Bids* relation:

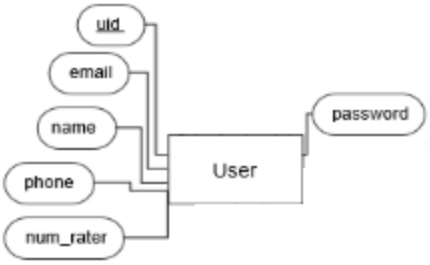


In SQL: *Bids*(price:real, date:date, uid:string, item_id:integer)

For *Buynow* relation:

In SQL: *Buynow*(item_id:integer, uid:string)

For *User* entity:



In SQL: *User*(uid:string, email:string, name:string, phone:string, password:string)

For *Card_info* entity:

In SQL: *Card_info*(type:string, cyc:string, cnumber:string, expdae:string, uid:string)

For *Individual* entity:

In SQL: *Individual*(gender:string, dob:date, income:integer, uid:string)

For *Company* entity:

In SQL: *Company*(url:string, revenue:integer, category:string, contact:string, uid:string)

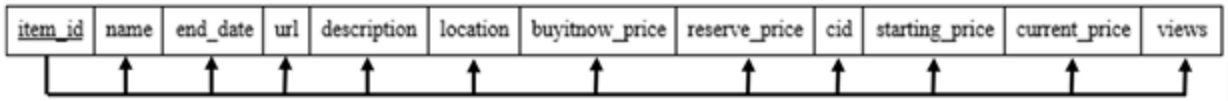
3.4 Normalization

Overall, few actions *had* to be taken to refine the schema, as most schema taken from the ER diagram were already in third normal form. All changes made and redundancies eliminated are described at length below.

Category Map (Combined Entity and Relationship)

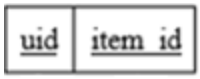
Category is in Third Normal Form as all the attributes depend on the key attribute, *cid*. This schema represents the combination of the "Maps_To" entity and the "Category_Map" entity.

Item_Sell_by (Combined Entity and Relationship)



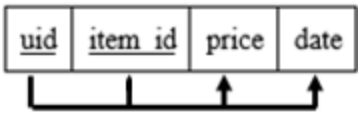
Item is in Third Normal Form as all the attributes depend on the key attribute, *item_id*. This schema was created from combining the "Item" entity with the "Sells" relationship. This enforces total participation between "Item" and "Sells".

Watches (Relationship)



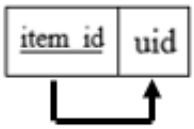
The *Watches* relationship is comprised of two primary keys, thus lacks any functional dependencies whatsoever.

Bids (Relationship)



The *Bids* relationship keeps track of the bids between various users and items. All non-key attributes are dependent on key-attributes, thus putting *Bids* into Third Normal Form.

BuyNow (Relationship)

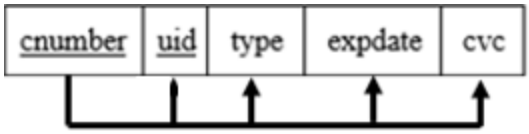


The *BuyNow* relationship is also in Third Normal Form as all non-key attributes are dependent only upon one key.

User (Entity)

User is in 3rd Normal Form as all the attributes depend on the key attribute, *uid*. As can be seen, the attribute "num_rater" has been removed to eliminate a redundancy. This was leftover from the old rating system which was employed in our last design, and during the process of schema refinement, was determined to be holding unnecessary data. Thus, eliminating this attribute conserves space.

Card_info (Entity)

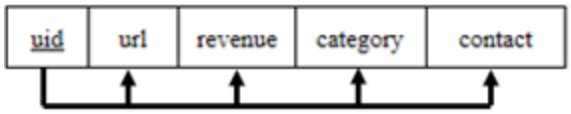


Card_info is in Third Normal Form as all the attributes depend on the key attribute, *cnumber*.

Individual (Weak Entity)

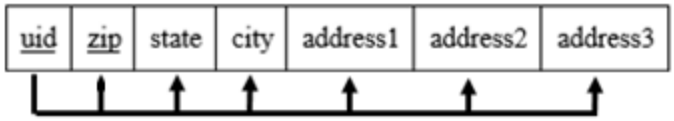
Since *Individual* is a weak entity, it uses a foreign key attribute *uid* from the *User* entity. All the attributes depend on the foreign key *uid*.

Company (Weak Entity)



Since *company* is a weak entity, it uses a foreign key attribute *uid* from the *User* entity. All the attributes depend on the foreign key *uid*.

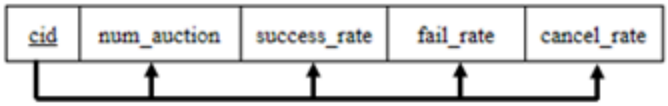
Lives_Address (Weak Entity)



Address is a weak entity that uses a foreign key attribute *uid* from the *User* entity. *zip* attribute is assigned as a partial key. There is an option of making *state* and *city* attributes to depend only on *zip* attribute by creating a weak entity with *zip* as the foreign key. This can reduce redundancy and save storage. However, it was decided to keep it as it is since *Address* is already a weak entity, and there can be multiple state names for one zipcode, for example, a zip code '16801' is used for 'State

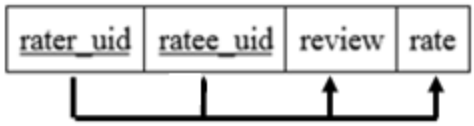
College' as well as 'University Park'.

Statistics (Entity)



Statistics is in Third Normal Form as all the attributes depend on the key attribute, *cid*.

Reviews (Relationship)



The *Reviews* relationship records all data regarding reviews and ratings. Additionally, *Reviews* fulfills the requirements for Third Normal Form.

4. Implementation of System Prototype

4.1 Desired Functionality

This phase of the project represents the culmination of the design phases, as it implements a prototype which demonstrates specified system functionality. The functionality of this system prototype can be separated into external and internal functionality, with external functionality representing the implementation of the user front-end and internal functionality representing the implementation of programs used for business functions. Specifically, the external component of this prototype contains:

- A simplistic front-end web interface

- The ability for users to register and log in
- Auction creation functionality for logged-in users
- The ability for users to browse items based on keyword or conditions
- A means of users to bid on auction items
- A means for users to watch items
- A user-specific page detailing items contained in watch and bid lists

NOTE: The last two specifications listed above represent the prototype implementation of two additional features not part of the original requirements.

The internal component of this prototype contains:

- A program used to terminate expired auctions
- A program used to draw up a report to be sent to telemarketers

Most importantly, several important changes were done in the creation of this prototype which either represent modifications or improvements of the original specifications. These changes are of two types: functionality to be implemented in the final product which was not necessary for the prototype, and functionality which has fundamentally been improved. The former type of change simply represents the fact that this prototype only covers basic functionality in terms of the desired functionality listed above. Thus, this type of change is mentioned briefly in the below two sections. The latter type of change, being more significant, is discussed in the *Improved Specifications(4.4)* section.

4.2 External Component

For the front-end, a web interface was designed in HTML and CSS. In order to implement the other described external specifications relating to this web interface, all these HTML files were converted to JSP. Using Java written into these web pages, front-end functionality was able to be connected with database access.

In terms of the web page design, a simple yet sleek approach was chosen over the more cluttered appearance of competitor sites. Five tabs sit atop of the page next to one another: Home, Account, Sell, Buy and Activity. The Home tab is the first page to be shown to potential customers that use the site. It gives a brief overview of EAuction's mission and why EAuction is preferable to competitors. The Login and Register buttons at the bottom, when clicked, will cause a pop-up box to show, either asking for information regarding logging in or registration. In terms of the original requirement specifications, this page corresponds to the "Front Page" as discussed earlier in this report. A link to a FAQ page, a link to a Contact Us page, and a listing of popular items were not included on this page as they are not relevant in terms of our basic prototype and do not demonstrate any important functionality. In later development phases of this project, these features will be added in appropriately.

The Account tab gives an overview of account information to logged in users, and is capable of being edited if the user clicks the edit button at the bottom of the page. The Sell tab leads to page filled with text fields relevant to the creation of an item to be auctioned. An item uploaded through this page will be inserted into the item database accordingly, and will be available in item searches or category browsing.

The Buy tab leads to a page which lists items by category. Subcategory buttons on the right of the page allow users to traverse subcategories, thus altering search results. Each item is denoted by a picture and associated information. A button entitled "See Description" will show an item description of

clicked upon. The Bid, Watch, and BuyNow buttons allow the user to respectively bid upon, watch or immediately buy an item. A search bar which allows for conditional category-driven searches also was implemented and is displayed on this page.

The Activity tab leads to a page which displays the user's selling list, bidding list, watch list, as well as activity history. This corresponds to the "Personal User Page" as described in the original project requirements.

4.3 Internal Component

For business functions not meant to be accessed by customers, two specific programs were created: one which terminated all expired auctions and another which creates a report to be sent to telemarketers. Both of these functions were written in Java and are completely distinct from the web interface described in length above.

The rating and review system, the delivery system, and the payment system, were not implemented in this prototype as they fall outside the scope of the prototype requirements.

4.4 Improved Specifications

All fundamental revisions to the original specifications were matters of style rather than substance. That is, after experimenting with designing the web interface in HTML and CSS, some specifications were slightly changed to be more aesthetic. Thus, no functionality has been removed from this project whatsoever, however, some small details were changed in order for the web interface to look more appealing. The changes to the original specifications are detailed below:

Logging in and Registration:

Instead of linking to a separate page, these functions are instead accomplished through pop-up boxes. This contributes to the overall simplistic design and Zen-like atmosphere of EAuction.

Item Display:

To contribute to this philosophy of clarity and simplicity, separate item pages were not created and instead all relevant item data is displayed on the search page. By eliminating unnecessary links to other pages, users will be able to enjoy a sleek, efficient, aesthetic and clear interface.

4.5 Prototype Success

The prototype fully and successfully embodies the functionality specified in the prototype requirements, and has a front-end which is simple and intuitive. Overall, the success of this prototype indicates that EAuction's basic system is efficient and functional, allowing for further research, implementation and development.

5. System Issues

The biggest concern in creating the e-Auction database system is how to deal with an enormous amount of data and transactions in a limited number of servers. A competitor product, eBay, claims to have over 212 million registered users with 26 billion SQL executions per day [1]. To deal with this issue, eBay and many other large shops use the scale-out approach. Usually a scale-in approach is utilized prior to the scale-out approach. The scale-in approach with SMP servers allows on demand resource allocation by sharing CPU and RAM between many resources, thus optimizing utilization of RAM and CPU resources. In contrast, the scale-out approach with multiple SMP servers yields optimal utilization of servers by allowing quicker implementation and easier maintenance with fewer servers [2].

Additionally, it would be a good practice to move processor-intensive operations out of the database for a better use of limited servers. By implementing a proper approach, we can solve system issues that may arise when dealing with large amount of data and transactions.

References:

1. Donald Burleson. "EBay's Massive Oracle Database." *Oracle News*. Burleson Consulting, 2014. Web. 05 Mar. 2014.

2. Donald Burleson. "Expert Remote DBA." *Oracle Scalability and Grid Technology*. Burleson Consulting, 2013. Web. 05 Mar. 2014.

6. Project Plan

6.1 Milestones

A SQL database was created in Oracle based on the design from phase I and phase II, and it was populated with testing data we generated. Several database transactions were implemented to test the external user operations of e-Auction. A website was created using JSP to provide a front-end user interface.

6.2 Deliverables

A project report containing the codes to generate and populate the databases have been prepared in addition to the requirement analysis, conceptual design, SQL relations and an explanation of design normalization from phase I and II. There will also be a short presentation on how we created our database and created a website that connects to it along with explaining several decisions we have made throughout this project. A slideshow for the presentation has been prepared as well. In addition, a project demonstration using a front-end website will be made.

7. Conclusion

The progress of this project is divided to three phases, and this report describes the progress of phase III which is the final report on this project.

7.1 Lessons Learned

During this project, we have learned database design from the back-end to front-end. In phase I of the project, we learned how to design a database and to create an ER model. In phase II, we produced a refined schema to reduce redundancy to an acceptable level by using normalization. We realized how a well-made conceptual design produces already normalized schema. We also learned to make decisions on what and what not to normalize under certain conditions. In phase III of the project, we learned how to put our design into life by implementing a database using MySQL and connecting it with a front-end website using JSP.

7.2 Decision Making

We tried to make the front-end eAuction website using PHP, but we failed to correct a connecting issue possibly due to the firewall. So instead, we chose to use JSP in NetBeans, because JSP uses Java programming language which all our team members are fluent in. JSP was a good substitute for PHP since they both create dynamic content by accepting input from a user, interacting with a database and then returning custom content to the user's Web browser. [1]

For the interface, we chose to keep the interface same for both individuals and companies since there is not much difference in their roles; only the account information produced different sections such as an additional section for individuals' gender information. We also did not differentiate the interface

between seller and buyers, because there can be users both selling and buying items on eAuction. Thus, any user who logs in can buy or sell depending on whether he or she chooses the buy or sell tab.

References:

1. Tommy Charles. "The Percentage Use of PHP Vs. JSP." *eHow*. Burleson Consulting, 2014. Web. 05 Mar. 2014.

8. Appendices

8.1 Progress Report

8.1.1 Group Report:

Date	Hours	Members Participated
1/17/2014 (Phase I)	1	All members
1/28/2014	3	All members
2/09/2014	6	All members
2/25/2014 (Phase II)	3	All members
3/02/2014	7	All members
3/05/2014	3	All members
4/01/2014 (Phase III)	5	All members
4/04/2014	2	All members
4/17/2014	6	All members
4/20/2014	5	All members
4/29/2014	4	All members

Summary:

(Phase I)

1/17/2014: We shared each other's names and contact numbers. Our group name, "Scheme" was chosen.

1/28/2014: We spent time to read through the project descriptions in detail. Each member was assigned four requirements to analyze. After the requirement analysis, we finished working on our first progress report.

2/09/2014: We first went over the requirements and made detailed adjustments. Then, we distributed work for the project report, and created the slides for the presentation on Monday.

(Phase II)

2/25/2014: We made corrections on the ER diagram according to the project feedback.

3/02/2014: We analyzed what we can normalize for a better design, and decided just to keep the original design for the most part. SQL was produced based on the finalized design. Finished project report for Phase II.

3/05/2014: We prepared for a short presentation by creating a slideshow and went over our report to check if there are any errors.

(Phase III)

4/01/2014: We connected to mySQL at the Lab218. Database was created according to our design.

4/04/2014: Test data was populated. We created a website using PHP, but could not correct the connection issue.

4/17/2014: We have decided to use JSP instead of PHP. So we had to change our PHP pages to JSP. Embedded Java codes to connect to the database within JSP.

4/20/2014: Finished working on the website. Started writing the project report.

4/29/2014: Finished writing the project report. Created a slideshow for the final presentation.

8.1.2 Individual Report:

Individually spent hours working on this phase of the project.

Name	**Hours (Phase III)**	**Cumulative Hours (Total)**
Sela Jung	22	45
Alexander Hershman	22	45
Ju Hong Min	22	45
Yoonsung Son	22	45

8.2 Changes from Last Report

In response to constructive criticism offered from the instructor since the second portion of this report was submitted, the following changes have been made in response to the following concerns:

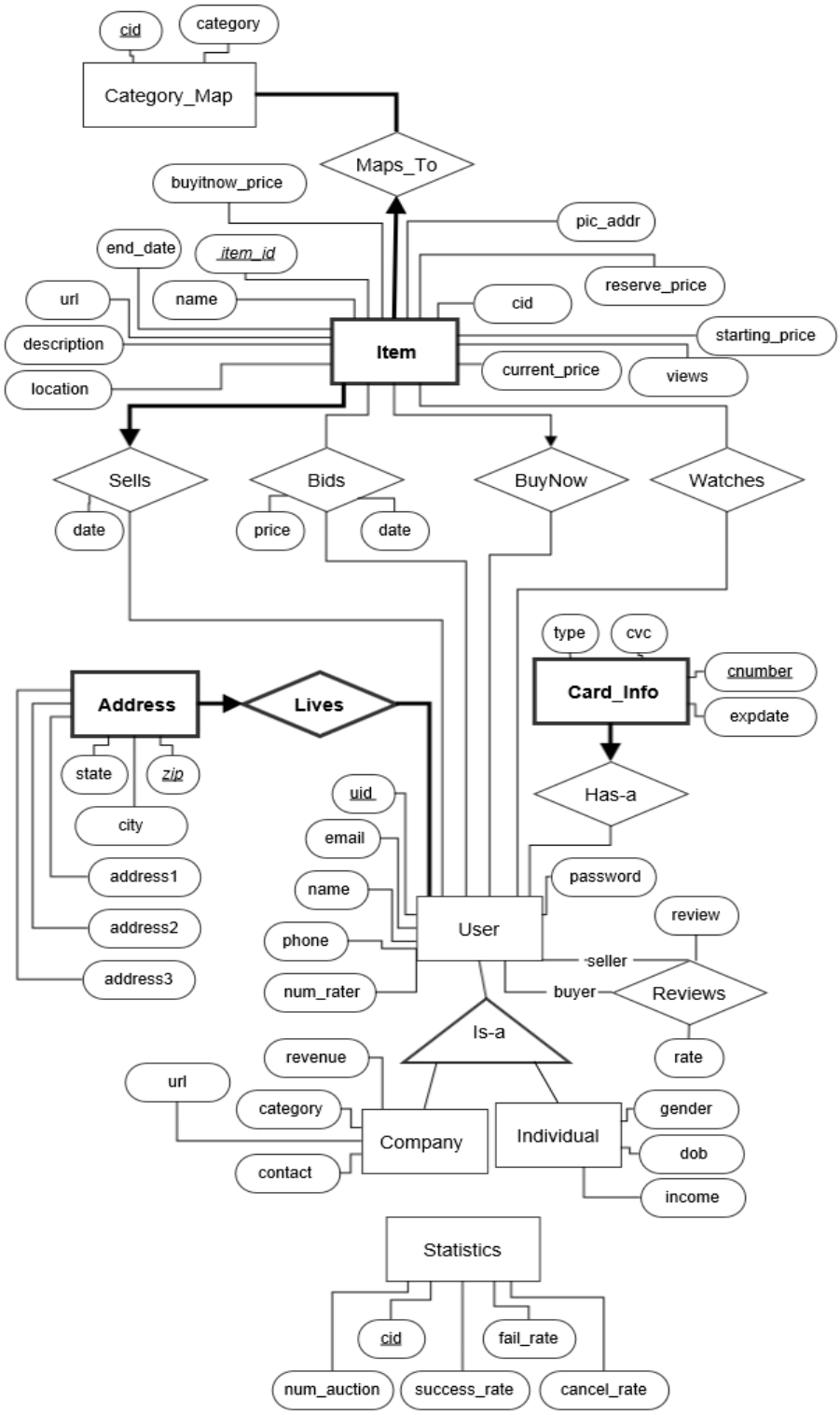
1. *To improve, it would be better to put the full E-R diagram into the appendix since there is no description surrounding it.*

We moved the diagram into the appendix accordingly.

2. *The scheme creation section is a bit too brief which only lists the high-level logical schema. To improve readability, we encourage you to map every created table to the corresponding ER component.*

We added pictorial representations of the logical schema in terms of their original, corresponding ER entities.

8.3 Entity-Relationship Diagram for e-Auction


cid
category
Category_Map
Maps_To
buyitnow_price
pic_addr
end_date
item_id
reserve_price
url
name
cid
description
Item
starting_price
location
current_price
views
Sells
Bids
BuyNow
Watches
date
price
date
type
cvc
Address
Lives
Card_Info
cnumber
expdate
state
zip
Has-a
uid
city
email
password
address1
name
review
address2
User
phone
seller
address3
num_rater
buyer
Reviews
Is-a
revenue
rate
url
category
gender
Company
Individual
dob
contact
income
Statistics
cid
fail_rate
num_auction
success_rate
cancel_rate

8.4 SQL

8.4.1 Table Creation

```
CREATE TABLE CATEGORY_MAP (
       cid                 CHAR(10),
       category            VARCHAR(40),
       PRIMARY KEY (cid) )
```

- The Category entity and Maps_To relationship are combined.

```
CREATE TABLE ITEM_SELL_BY (
       item_id             INTEGER,
       name                VARCHAR(60),
       cid                 CHAR(10) NOT NULL,
       end_date            DATE,
       reserve_price       REAL,
       starting_price      REAL,
       current_price       REAL,
       buyitnow_price      REAL,
       views               INTEGER,
       url                 VARCHAR(50),
       location            VARCHAR(100),
       description         VARCHAR(1000),
       uid                 VARCHAR(15) NOT NULL,
       date                DATE,
       PRIMARY KEY (item_id),
       FOREIGN KEY (uid) REFERENCES USER
              ON DELETE CASCADE
              ON UPDATE CASCADE,
       FOREIGN KEY (cid) REFERENCES CATEGORY_MAP
              ON DELETE SET NULL
              ON UPDATE CASCADE )
```

- As every item must have exactly one seller, the Sells relationship and Item entity are combined. When either uid or item_id is deleted or updated, the selling information must be similarly updated or deleted.

```
CREATE TABLE WATCHES (
       uid                 VARCHAR(15),
       item_id             INTEGER,
       PRIMARY KEY (uid, item_id),
       FOREIGN KEY (uid) REFERENCES USER
              ON DELETE CASCADE
```

```
            ON UPDATE CASCADE,
      FOREIGN KEY (item_id) REFERENCES ITEM_SELL_BY
            ON DELETE CASCADE
            ON UPDATE CASCADE )
```

- The Watches relationship has a many-to-many relationship, and deletion and updating are specified as ‘cascade’.

```
CREATE TABLE BIDS (
      price                REAL,
      date                 DATE,
      uid                  VARCHAR(15),
      item_id              INTEGER,
      PRIMARY KEY (uid, item_id),
      FOREIGN KEY (uid) REFERENCES USER
            ON DELETE CASCADE
            ON UPDATE CASCADE,
      FOREIGN KEY (item_id) REFERENCES ITEM_SELL_BY
            ON DELETE CASCADE
            ON UPDATE CASCADE )
```

- The Bids relationship has a many-to-many relationship. When either uid or item_id is deleted or updated, the bidding information must be also be deleted or updated.

```
CREATE TABLE BUYNOW (
      item_id              INTEGER,
      uid                  VARCHAR(15),
      PRIMARY KEY (item_id),
      FOREIGN KEY (item_id) REFERENCES ITEM_SELL_BY
            ON DELETE CASCADE
            ON UPDATE CASCADE,
      FOREIGN KEY (uid) REFERENCES USER
            ON DELETE CASCADE
            ON UPDATE CASCADE )
```

- The Buynow relationship has a one-to-many relationship. Both deletion and updating are cascaded.

```
CREATE TABLE USER (
      uid                  VARCHAR(15),
      email                VARCHAR(50),
      name                 VARCHAR(40),
      phone                VARCHAR(15),
      password             CHAR(32),
      PRIMARY KEY (uid) )
```

- As every user must be either a company or an individual, covering constraint exist and overlapping constraint are disallowed.

```
CREATE TABLE CARD_INFO (
       type                CHAR(1),
       cvc                 CHAR(3),
       cnumber             CHAR(16),
       expdate             CHAR(4),
       uid                 VARCHAR(15),
       PRIMARY KEY (cnumber, uid),
       FOREIGN KEY (uid) REFERENCES USER
              ON DELETE CASCADE
              ON UPDATE SET NULL )
```

- Card_info is a weak entity so it must be updated when there is a change in user information.

```
CREATE TABLE INDIVIDUAL (
       gender              CHAR(1),
       dob                 DATE,
       income              INTEGER,
       uid                 VARCHAR(15),
       PRIMARY KEY (uid),
       FOREIGN KEY (uid) REFERENCES USER
              ON DELETE CASCADE )
```

- The Individual entity has a IS-A relationship with User entity. It must be deleted when a corresponding User tuple is deleted.

```
CREATE TABLE COMPANY (
       url                 VARCHAR(50),
       revenue             INTEGER,
       category            VARCHAR(30),
       contact             VARCHAR(30),
       uid                 VARCHAR(15),
       PRIMARY KEY (uid),
       FOREIGN KEY (uid) REFERENCES USER
              ON DELETE CASCADE )
```

- The Company entity also has a IS-A relationship with User entity. The references of users has been set to cascade upon deletion.

```
CREATE TABLE LIVES_ADDRESS (
       state               CHAR(2),
       zip                 VARCHAR(10),
       city                VARCHAR(25),
       address1            VARCHAR(40),
       address2            VARCHAR(40),
       address3            VARCHAR(40),
       uid                 VARCHAR(15),
       PRIMARY KEY (uid, zip),
       FOREIGN KEY (uid) REFERENCES USER
              ON DELETE CASCADE )
```

- Similar to the above two weak entities, the Address entity must be deleted when a corresponding user tuple is deleted.

```
CREATE TABLE STATISTICS (
      cid                 CHAR(10),
      num_auction         INTEGER,
      fail_rate           REAL,
      success_rate        REAL,
      cancel_rate         REAL,
      PRIMARY KEY (cid))
```

- Statistics is an independant entity.

```
CREATE TABLE REVIEWS (
      rater_uid           VARCHAR(15),
      ratee_uid           VARCHAR(15),
      review              VARCHAR(300),
      rate                INTEGER,
      PRIMARY KEY (rater_uid, ratee_uid),
      FOREIGN KEY (rater_uid) REFERENCES USER(uid),
      FOREIGN KEY (ratee_uid) REFERENCES USER(uid) )
```

- User can rate or review to another user. This is a relationship with the same entity set. Since neither 'review' nor 'rate' are defined as "NOT NULL", a user can rate another user without reviewing him and vice-versa.

8.3.2 Implementation

Account.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%
  String uidCookie = "";
  String cookieName = "uidCKscheme";
  Cookie cookie = null;
  Cookie[] cookies = null;
  // Get an array of Cookies associated with this domain
  cookies = request.getCookies();
  if( cookies != null ){
        for (int i = 0; i < cookies.length; i++){
        cookie = cookies[i];
        if (cookies[i].getName().equals(cookieName))
              {
```

```
                uidCookie = cookies[i].getValue();
                }

        }
  }
%>
<%
        Connection con=null;
        Statement stat=null;
        ResultSet rs=null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
        System.out.println("Connection established.");
        }
        catch(SQLException sqle)
        {
        System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
        }
        catch(ClassNotFoundException cnfe)
        {
        System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
        }
        catch(Exception e)
        {
        System.out.println("General Exception:"+e.getMessage());
        con=null;
        }

        String id = "";
        String name = "";
        String phone = "";
        String gender = "";
        String dob = "";
        String url = "";
        int revenue = 0;
        String category = "";
```

```
String contact = "";
int income = 0;
String state = "";
String zip = "";
String city = "";
String addr1 = "";
String addr2 = "";
String addr3 = "";
String address = "";
char IorC = ' ';
String setVariable = "";

int MAXCARD = 3;
String cardtype[] = {"","",""};
String cardtypestr[] = {"","",""};
String cardnum[] = {"","",""};
String cardcvc[] = {"","",""};
String cardexpdate[] = {"","",""};
String pwd = "";
int num_cards = 0;

if(uidCookie != "") {

String sql = "SELECT email,name,phone,password FROM USERS WHERE uid = '" +
uidCookie + "'";
Statement s = null;

try{
s = con.createStatement();
rs = s.executeQuery(sql);

while (rs.next()) {
        id = rs.getString(1);
        name = rs.getString(2);
        phone = rs.getString(3);
        pwd = rs.getString(4);
}
}
catch(Exception e){e.printStackTrace();}

sql = "SELECT type,cvc,cnumber,expdate FROM CARD_INFO WHERE uid = '" +
uidCookie + "'";
Statement s0 = null;
int i = 0;
```

```
try{
s0 = con.createStatement();
rs = s0.executeQuery(sql);

while (rs.next()) {
        cardtype[i] = rs.getString(1);
        cardcvc[i] = rs.getString(2);
        cardnum[i] = rs.getString(3);
        cardexpdate[i] = rs.getString(4);
        i++;
        num_cards++;
}
}
catch(Exception e){e.printStackTrace();}

for (i = 0; i < num_cards; i++) {
if (cardtype[i].equals("0"))
        cardtypestr[i] = "VISA";
else if (cardtype[i].equals("1"))
        cardtypestr[i] = "MasterCard";
else if (cardtype[i].equals("2"))
        cardtypestr[i] = "American Express";
}


sql = "SELECT state,zip,city,address1,address2,address3 FROM LIVES_ADDRESS
WHERE uid = '" + uidCookie + "'";
Statement s1 = null;

try{
s1 = con.createStatement();
rs = s1.executeQuery(sql);

while (rs.next()) {
        state = rs.getString(1);
        zip = rs.getString(2);
        city = rs.getString(3);
        addr1 = rs.getString(4);
        addr2 = rs.getString(5);
        addr3 = rs.getString(6);

}
}
catch(Exception e){e.printStackTrace();}
```

```
address = addr1 + " " + addr2 + " " + addr3 + ", " + city + ", " + state + ", " + zip;


int count = 0;
sql = "SELECT count(*) FROM INDIVIDUAL WHERE uid = '" + uidCookie + "'";
Statement s2 = null;

try{
s2 = con.createStatement();
rs = s2.executeQuery(sql);

while (rs.next()) {

        count = rs.getInt(1);

}
}
catch(Exception e){e.printStackTrace();}

if (count == 1)
IorC = 'I';
else IorC = 'C';

if (IorC == 'I') {
sql = "SELECT gender,dob,income FROM INDIVIDUAL WHERE uid = '" + uidCookie +
'";
Statement s3 = null;

try{
        s3 = con.createStatement();
        rs = s3.executeQuery(sql);

        while (rs.next()) {
        gender = rs.getString(1);
        dob = rs.getString(2);
        income = rs.getInt(3);
        }
}
catch(Exception e){e.printStackTrace();}

if (gender == "M")
        gender = "Male";
else gender = "Female";
}
else {
```

```
	sql = "SELECT url,revenue,category,contact FROM COMPANY WHERE uid = '" +
uidCookie + "'";
	Statement s3 = null;

	try{
		s3 = con.createStatement();
		rs = s3.executeQuery(sql);

		while (rs.next()) {
		url = rs.getString(1);
		revenue = rs.getInt(2);
		category = rs.getString(3);
		contact = rs.getString(4);
		}
	}
	catch(Exception e){e.printStackTrace();}
	}

	}
%>
<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <link rel="stylesheet" type="text/css" href="css/about.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <script src="js/about.js"></script>
   <script src="js/aboutjquery.js"></script>
   <title>eAuction</title>
</head>
<body>
   <div id="background"></div>
   <form action="userEdit.jsp" method="post">
   <table id="registerTable">
		<tr>
			<td colspan="2" style="text-align: center;"><h1>Edit Information</h1></td>
		</tr>

		 <tr>
			<td><p>New Password: </p></td>
			<td><input name="pwd" type="password" value="<%=pwd%>"/></td>
		 </tr>
```

```
<tr>
        <td><p>Name: </p></td>
        <td><input name="name" type="text" value="<%=name%>"/></td>
</tr>

<tr>
        <td><p>State: </p></td>
        <td><select name="state">
                <option value="<%=state%>"><%=state%></option>
                 <option value="AL">Alabama</option>
                 <option value="AK">Alaska</option>
                 <option value="AZ">Arizona</option>
                 <option value="AR">Arkansas</option>
                 <option value="CA">California</option>
                 <option value="CO">Colorado</option>
                 <option value="CT">Connecticut</option>
                 <option value="DE">Delaware</option>
                 <option value="DC">District Of Columbia</option>
                 <option value="FL">Florida</option>
                 <option value="GA">Georgia</option>
                 <option value="HI">Hawaii</option>
                 <option value="ID">Idaho</option>
                 <option value="IL">Illinois</option>
                 <option value="IN">Indiana</option>
                 <option value="IA">Iowa</option>
                 <option value="KS">Kansas</option>
                 <option value="KY">Kentucky</option>
                 <option value="LA">Louisiana</option>
                 <option value="ME">Maine</option>
                 <option value="MD">Maryland</option>
                 <option value="MA">Massachusetts</option>
                 <option value="MI">Michigan</option>
                 <option value="MN">Minnesota</option>
                 <option value="MS">Mississippi</option>
                 <option value="MO">Missouri</option>
                 <option value="MT">Montana</option>
                 <option value="NE">Nebraska</option>
                 <option value="NV">Nevada</option>
                 <option value="NH">New Hampshire</option>
                 <option value="NJ">New Jersey</option>
                 <option value="NM">New Mexico</option>
                 <option value="NY">New York</option>
                 <option value="NC">North Carolina</option>
                 <option value="ND">North Dakota</option>
```

```
                <option value="OH">Ohio</option>
                <option value="OK">Oklahoma</option>
                <option value="OR">Oregon</option>
                <option value="PA">Pennsylvania</option>
                <option value="RI">Rhode Island</option>
                <option value="SC">South Carolina</option>
                <option value="SD">South Dakota</option>
                <option value="TN">Tennessee</option>
                <option value="TX">Texas</option>
                <option value="UT">Utah</option>
                <option value="VT">Vermont</option>
                <option value="VA">Virginia</option>
                <option value="WA">Washington</option>
                <option value="WV">West Virginia</option>
                <option value="WI">Wisconsin</option>
                <option value="WY">Wyoming</option>
        </select></td>
</tr>

<tr>
        <td><p>City: </p></td>
        <td><input name="city" type="text" value="<%=city%>"/></td>
</tr>

<tr>
        <td><p>ZIP: </p></td>
        <td><input name="zip" type="text" value="<%=zip%>"/></td>
</tr>

<tr>
        <td><p>Address1: </p></td>
        <td><input name="addr1" type="text" value="<%=addr1%>"/></td>
</tr>

<tr>
        <td><p>Address2: </p></td>
        <td><input name="addr2" type="text" value="<%=addr2%>"/></td>
</tr>

<tr>
        <td><p>Address3: </p></td>
        <td><input name="addr3" type="text" value="<%=addr3%>"/></td>
</tr>

<tr>
```

```
                <td><p>Phone: </p></td>
                <td><input name="phone" type="text" value="<%=phone%>"/></td>
            </tr>

            <%
                String gendTF = "false";
                if (gender == "Male")
                gendTF = "true";
            %>

            <% if (IorC == 'I') { %>
            <tr>
                <td>Income ($):</td>
                <td><input name="income" type="text" value="<%=income%>"/></td>
            </tr>
            <% } else if (IorC == 'C') { %>
            <tr>
                <td>Revenue ($):</td>
                <td><input name="revenue" type="text" value="<%=revenue%>"/></td>
            </tr>
            <tr>
                <td>Category:</td>
                <td><input name="category" type="text" value=""/></td>
            </tr>
            <tr>
                <td>URL:</td>
                <td><input name="url" type="text" value="<%=url%>"/></td>
            </tr>
            <tr>
                <td>Contact:</td>
                <td><input name="contact" type="text" value="<%=contact%>"/></td>
            </tr>

            <% } %>

            <tr>
                <td style="font-weight: bold;">Card1 type:</td>
                <td><input name="cardtype1" type="text"
value="<%=cardtypestr[0]%>"/></td>
            </tr>
            <tr>
                <td>Number</td>
                <td><input name="cardnum1" type="text"
value="<%=cardnum[0]%>"/></td>
            </tr>
```

```
                <tr>
                        <td>Exp. date</td>
                        <td><input name="cardexp1" type="text"
value="<%=cardexpdate[0]%>"/></td>
                 </tr>
                <tr>
                        <td>CVC</td>
                        <td><input name="cardcvc1" type="text" value="<%=cardcvc[0]%>"/></td>
                 </tr>


                <tr>
                        <td style="font-weight: bold;">Card2 type:</td>
                        <td><input name="cardtype2" type="text"
value="<%=cardtypestr[1]%>"/></td>
                 </tr>
                <tr>
                        <td>Number</td>
                        <td><input name="cardnum2" type="text"
value="<%=cardnum[1]%>"/></td>
                 </tr>
                <tr>
                        <td>Exp. date</td>
                        <td><input name="cardexp2" type="text"
value="<%=cardexpdate[1]%>"/></td>
                 </tr>
                <tr>
                        <td>CVC</td>
                        <td><input name="cardcvc2" type="text" value="<%=cardcvc[1]%>"/></td>
                 </tr>

                <tr>
                        <td style="font-weight: bold;">Card3 type:</td>
                        <td><input name="cardtype3" type="text"
value="<%=cardtypestr[2]%>"/></td>
                 </tr>
                <tr>
                        <td>Number</td>
                        <td><input name="cardnum3" type="text"
value="<%=cardnum[2]%>"/></td>
                 </tr>
                <tr>
                        <td>Exp. date</td>
                        <td><input name="cardexp3" type="text"
value="<%=cardexpdate[2]%>"/></td>
```

```
                </tr>
                <tr>
                        <td>CVC</td>
                        <td><input name="cardcvc3" type="text" value="<%=cardcvc[2]%>"/></td>
                </tr>

                <tr>
                        <td colspan="2"><button id="regbutton" class="button1">Edit</button></td>
                </tr>
                <input name="IorC" type="hidden" value="<%=IorC%>"></input>
</table>
</form>

<div id="header">
        <div id="logobox">
                <a href="index.jsp">eAuction-Scheme.com</a>
        </div>
</div>

<div id="navbar">
        <div id="menubox">
                <ul>
                        <li><a href="index.jsp">HOME</a></li>
                        <li><a href="account.jsp" class="activeIndex">ACCOUNT</a></li>
                        <li><a href="sell.jsp">SELL</a></li>
                        <li><a href="stack.jsp">BUY</a></li>
                        <li><a href="activity.jsp" >ACTIVITY</a></li>
                </ul>
        </div>
</div>

<div id="container">
                <% if (uidCookie != "") {%>
        <div id="aboutbox">
                <div class="aboutlibox">
                        <h2 style="margin-bottom: 15px;">My Account</h2>
                        <p class="aboutli"><a class="editLink">ID:</a></p>
                        <p class="aboutsubli"><%=id%></p>

                        <p class="aboutli"><a class="editLink">Password:</a></p>
                        <p class="aboutsubli">Hidden for security</p>

                        <p class="aboutli"><a class="editLink">Name:</a></p>
                        <p class="aboutsubli"><%=name%></p>
```

```
                    <p class="aboutli"><a class="editLink">Address:</a></p>
                    <p class="aboutsubli"><%=address%></p>

                    <p class="aboutli"><a class="editLink">Phone:</a></p>
                    <p class="aboutsubli"><%=phone%></p>

                   <% if (IorC == 'I') { %>
                   <p class="aboutli"><a class="editLink">Gender:</a></p>
                   <p class="aboutsubli"><%=gender%></p>

                   <p class="aboutli"><a class="editLink">Date of Birth:</a></p>
                   <p class="aboutsubli"><%=dob%></p>

                   <p class="aboutli"><a class="editLink">Income ($):</a></p>
                   <p class="aboutsubli"><%=income%></p>
                   <% } %>

                   <% if (IorC == 'C') { %>
                   <p class="aboutli"><a class="editLink">URL:</a></p>
                   <p class="aboutsubli"><%=url%></p>

                   <p class="aboutli"><a class="editLink">Revenue ($):</a></p>
                   <p class="aboutsubli"><%=revenue%></p>

                   <p class="aboutli"><a class="editLink">Category:</a></p>
                   <p class="aboutsubli"><%=category%></p>

                   <p class="aboutli"><a class="editLink">Contact:</a></p>
                   <p class="aboutsubli"><%=contact%></p>
                   <% } %>

                    <p class="aboutli"><a class="editLink">Card Info:</a></p>
                   <% for (int i = 0; i < num_cards; i++) { %>
                    <p class="aboutsubli"><%=cardtypestr[i]%>. <%=cardnum[i]%></p>
                    <p class="aboutsubli"> - Exp: <%=cardexpdate[i]%></p>
                    <p class="aboutsubli"> - CVC: <%=cardcvc[i]%></p>
                   <% } %>
              </div>
                   <button id="editLinkUser" style="margin-left: 20px; margin-top: 20px; width:
250px;" class="button2">Edit Information</button>
              </div>

         <div id="skillsbox">
```

```
        </div>
                <%} else { %>

                <h2 style="margin-bottom: 15px; text-align: center;">Please login first.</h2>

                <% } %>
    </div>

    <div id="footer">
        <hr/>
        <p id="copyright">Copyright &copy; 2014. All Rights Reserved.</p>
    </div>
</body>
</html>

<%
        if (con != null) {
        con.close();
        con = null;
        System.out.println("DB connection closed successfully.");
        }
%>
```

Activity.jsp

```
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.text.DateFormat"%>
<%@page import="java.lang.*"%>
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<%@page import="java.util.*"%>
<%    Connection con=null;
    Statement stat=null;
    ResultSet rs=null;
        try
    {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
                 con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme",
"scheme");
```

```
        System.out.println("Connection established.");
    }
        catch(SQLException sqle)
        {
                System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
    }
        catch(ClassNotFoundException cnfe)
        {
                System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
    }
        catch(Exception e)
        {
                System.out.println("General Exception:"+e.getMessage());
        con=null;
    }

        java.util.Date date = new java.util.Date();
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        String currentDate = dateFormat.format(date);
%>

<%
   String uidCookie = "";
   String cookieName = "uidCKscheme";
   Cookie cookie = null;
   Cookie[] cookies = null;
   // Get an array of Cookies associated with this domain
   cookies = request.getCookies();
   if( cookies != null ){
        for (int i = 0; i < cookies.length; i++){
        cookie = cookies[i];
        if (cookies[i].getName().equals(cookieName))
                {
                uidCookie = cookies[i].getValue();
                break;
                }

        }
  }
%>
<!DOCTYPE html>
<html>
<head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
    <link rel="stylesheet" type="text/css" href="css/projects.css">
    <script src="js/jquery.js"></script>
    <script src="js/stdjquery.js"></script>
    <title>eAuction</title>
</head>
<body>
    <div id="background"></div>
    <div id="header">
        <div id="logobox">
            <a href="index.jsp">eAuction-Scheme.com</a>
        </div>
    </div>

    <div id="navbar">
        <div id="menubox">
            <ul>
                <li><a href="index.jsp">HOME</a></li>
                <li><a href="account.jsp">ACCOUNT</a></li>
                <li><a href="sell.jsp">SELL</a></li>
                <li><a href="stack.jsp">BUY</a></li>
                <li><a href="activity.jsp"  class="activeIndex">ACTIVITY</a></li>
            </ul>
        </div>
    </div>

    <div id="container">
            <% if (uidCookie != "") {%>
        <div class="genprojectbox">
            <h1>Selling List</h1>
            <div class="projectbox">
                <%
                stat = con.createStatement();

                String query = "SELECT * FROM ITEM_SELL_BY "
                        + "WHERE uid = '" + uidCookie + "' and end_date > '" + currentDate
+ "'";
                rs = stat.executeQuery(query);

                try
                {
                        if(!rs.next())
                        {%>  <p>There is no item currently selling.</p><%;}
                        else{
```

```
                                do
                                {
                                %>
                                <h3><%=rs.getString("name")%> (Item ID:
<%=rs.getInt("item_id")%>)</h3>
                                <p style="text-indent:2em">Current Price:
$<%=rs.getFloat("current_price")%>
                                (Started at $<%=rs.getFloat("starting_price")%>)</p>
                                <p style="text-indent:2em">Current Winner ID:
<%=rs.getInt("uid")%></p>
                                <p style="text-indent:2em">Ends on
<%=rs.getDate("end_date")%></p>
                                <br><%
                                }while(rs.next());}
                        }
                        catch(SQLException ex)
                        {
                                System.out.println(ex.getMessage()+ex.getSQLState()+
                                ex.getErrorCode());
                        }
                        %>
                </div>
        </div>

        <div class="genprojectbox">
                <h1>Bidding List</h1>
                <div class="projectbox">
                        <%
                        stat = con.createStatement();

                        query = "SELECT * FROM BIDS, ITEM_SELL_BY "
                        + "WHERE BIDS.uid = '" + uidCookie + "' and end_date > '" + currentDate +
"'";
                        rs = stat.executeQuery(query);

                        try
                        {
                        if(!rs.next())
                        {%>  <p>There is no item bidding on.</p><%;}
                        else{
                        do
                        {%>
                                <h3><%=rs.getString("name")%> (Item ID:
<%=rs.getInt("item_id")%>)</h3>
                                <p style="text-indent:2em">Your bid price:
```

```
$<%=rs.getFloat("price")%>
                                (Current price: $<%=rs.getFloat("current_price")%>)
                                <% if(rs.getFloat("current_price") > rs.getFloat("price"))
                                %> <strong>You have been outbid!</strong></p>%>

                                <p style="text-indent:2em">Ends on
<%=rs.getDate("end_date")%></p>
                                <br>
                                <%
                        }while(rs.next());}
                        }
                        catch(SQLException ex)
                        {
                        System.out.println(ex.getMessage()+ex.getSQLState()+
                                ex.getErrorCode());
                        }
                        %>
                </div>
        </div>

        <div class="genprojectbox">
                <h1>Watch List</h1>
                <div class="projectbox">
                        <%
                        stat = con.createStatement();

                        query = "SELECT * FROM WATCHES W, ITEM_SELL_BY I "
                                + "WHERE W.uid = '" + uidCookie + "' and W.item_id = I.item_id "
                                + "and end_date > '" + currentDate + "'";
                        rs = stat.executeQuery(query);

                        try
                        {
                        if(!rs.next())
                        {%>  <p>There is no item watching.</p><%;}
                        else{
                        do
                        {
                                %>
                                <h3><%=rs.getString("name")%> (Item ID:
<%=rs.getInt("item_id")%>)</h3>
                                <p style="text-indent:2em">Current Price:
$<%=rs.getFloat("current_price")%>
                                (Started at $<%=rs.getFloat("starting_price")%>)</p>
                                <p style="text-indent:2em">Current Winner ID:
```

```
<%=rs.getInt("uid")%></p>
                                <p style="text-indent:2em">Ends on
<%=rs.getDate("end_date")%></p>
                                <br>
                                <%
                        }while(rs.next());}
                        }
                        catch(SQLException ex)
                        {
                        System.out.println(ex.getMessage()+ex.getSQLState()+
                                ex.getErrorCode());
                        }
                        %>
                </div>
        </div>

        <div class="genprojectbox">
                <h1>Activity History</h1>
                <div class="projectbox">
                        <%
                        stat = con.createStatement();

                        query = "SELECT * FROM ITEM_SELL_BY "
                        + "WHERE uid = '" + uidCookie + "' and end_date < '" + currentDate + "'";
                        rs = stat.executeQuery(query);

                        try
                        {
                        if(!rs.next())
                        {%>  <p>There is no activity history.</p><%;}
                        // get selling history
                        while(rs.next())
                        {%>
                                <h3><%=rs.getString("name")%> (Item ID:
<%=rs.getInt("item_id")%>)</h3>
                                <% if(rs.getFloat("current_price")>=rs.getFloat("reserve_price") ||
                                rs.getFloat("current_price")>=rs.getFloat("buyitnow_price"))
                                {%> <p style="text-indent:2em"><strong>Item
Sold!</strong></p><%;}
                                else
                                { %><p><strong>Item NOT Sold</strong></p><%;}%>
                                <p style="text-indent:2em">Current Price:
$<%=rs.getFloat("current_price")%>
                                (Started at $<%=rs.getFloat("starting_price")%>)</p>
                                <p style="text-indent:2em">Ended on
```

```
<%=rs.getDate("end_date")%></p>
                                <br>
                        <%}
                        }
                        catch(SQLException ex)
                        {
                        System.out.println(ex.getMessage()+ex.getSQLState()+
                                ex.getErrorCode());
                        }

                        // get bidding history
                        stat = con.createStatement();

                        query = "SELECT * FROM BIDS, ITEM_SELL_BY "
                        + "WHERE BIDS.uid = '" + uidCookie + "' and end_date < '" + currentDate +
"'";
                        rs = stat.executeQuery(query);


                        try
                        {
                        while(rs.next())
                        {%>
                                <h3><%=rs.getString("name")%> (Item ID:
<%=rs.getInt("item_id")%>)</h3>
                                <p style="text-indent:2em">Your bid price:
$<%=rs.getFloat("price")%>
                                (Current price: $<%=rs.getFloat("current_price")%>)
                                <p style="text-indent:2em">Ends on
<%=rs.getDate("end_date")%></p>
                                <br>
                                <%
                        }
                        }
                        catch(SQLException ex)
                        {
                        System.out.println(ex.getMessage()+ex.getSQLState()+
                                ex.getErrorCode());
                        }

                        // get watching history
                        stat = con.createStatement();

                        query = "SELECT * FROM WATCHES W, ITEM_SELL_BY I "
                                + "WHERE W.uid = '" + uidCookie + "' and W.item_id = I.item_id "
```

```
                        + "and end_date < '" + currentDate + "'";
                    rs = stat.executeQuery(query);

                    try
                    {
                    while(rs.next())
                    {%>
                        <h3><%=rs.getString("name")%> (Item ID:
<%=rs.getInt("item_id")%>)</h3>
                        <p style="text-indent:2em">You have been watching this item.</p>
                        <p style="text-indent:2em">End Price:
$<%=rs.getFloat("current_price")%>
                        (Started at $<%=rs.getFloat("starting_price")%>)</p>
                        <p style="text-indent:2em">Ends on
<%=rs.getDate("end_date")%></p>
                        <br>
                        <%
                    }
                    }
                    catch(SQLException ex)
                    {
                    System.out.println(ex.getMessage()+ex.getSQLState()+
                        ex.getErrorCode());
                    }
                    %>
                </div>
            </div>
                <%} else { %>

                <h2 style="margin-bottom: 15px; text-align: center;">Please login first.</h2>

                <% } %>
            </div>

    <div id="footer">
        <hr/>
        <p id="copyright">Copyright &copy; 2014. All Rights Reserved.</p>
    </div>
</body>
</html>
```

Bid.jsp

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.text.DateFormat"%>
<%@page import="java.util.*"%>
<%
 Connection con=null;
        ResultSet rs=null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
        System.out.println("Connection established.");
        }
        catch(SQLException sqle)
        {
        System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
        }
        catch(ClassNotFoundException cnfe)
        {
        System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
        }
        catch(Exception e)
        {
        System.out.println("General Exception:"+e.getMessage());
        con=null;
        }

        String itemid = request.getParameter("itemidb");
        String uid = request.getParameter("userid");
        String bidprice = request.getParameter("bidprice");
        int bidcount = -1;
        float bidpriceReal = (Float.valueOf(bidprice)).floatValue();

        java.util.Date date = new java.util.Date();
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        String currentDate = dateFormat.format(date);
```

```
String sql = "SELECT count(*) FROM BIDS WHERE uid = '" + uid + "' and item_id = " +
itemid + "";
Statement s = null;

try{
s = con.createStatement();
rs = s.executeQuery(sql);

while (rs.next()) {
bidcount = rs.getInt(1);
}
}
catch(Exception e){System.out.println("SQL error");}

sql = "SELECT reserve_price, current_price FROM ITEM_SELL_BY WHERE item_id = " +
itemid;
Statement s1 = null;
float reservepriceReal = 0;
float currentpriceReal = 0;

try{
s1 = con.createStatement();
rs = s1.executeQuery(sql);

while (rs.next()) {
reservepriceReal = rs.getFloat(1);
currentpriceReal = rs.getFloat(2);
}
}
catch(Exception e){System.out.println("SQL error");}



System.out.println("cur price: " + currentpriceReal);
System.out.println("res price: " + reservepriceReal);
System.out.println("bid price: " + bidpriceReal);
System.out.println("bidcount: " + bidcount);
System.out.println("date:  " + currentDate);
System.out.println("uid: " + uid);
System.out.println("itemid: " + itemid);

String redirectURL;
if (reservepriceReal <= bidpriceReal) {
```

```
        // automatically sold to this uid
        // 1. delete item in bid list
        PreparedStatement ps11=con.prepareStatement("DELETE FROM BIDS WHERE uid='" +
uid + "' and item_id=" + itemid);
        ps11.execute();

        // 2. alter table
        PreparedStatement ps4=con.prepareStatement("UPDATE ITEM_SELL_BY SET
end_date='"+currentDate+"', current_price=" + bidpriceReal
                + " WHERE item_id = " + itemid);
        ps4.execute();

        redirectURL = "exceedReserve.jsp";
        response.sendRedirect(redirectURL);
        }
        else if ((currentpriceReal*1.05 > bidpriceReal)) {
        // Error: already bidded OR bid price lower than current price
        redirectURL = "biderror.jsp";
        response.sendRedirect(redirectURL);
        } else if ((bidcount > 0) && (bidpriceReal >= currentpriceReal*1.05)) {
        // Update price
        PreparedStatement ps6=con.prepareStatement("UPDATE ITEM_SELL_BY SET
current_price="+bidpriceReal
                + "WHERE item_id = " + itemid);
        ps6.execute();

        redirectURL = "stack.jsp";
        response.sendRedirect(redirectURL);
        }
        else {
        if (currentpriceReal*1.05 <= bidpriceReal) {
        System.out.println("error3");

        PreparedStatement ps5=con.prepareStatement("UPDATE ITEM_SELL_BY SET
current_price="+bidpriceReal
                + "WHERE item_id = " + itemid);
        ps5.execute();

        PreparedStatement ps=con.prepareStatement("INSERT INTO BIDS VALUES (?,?,?,?)");
        ps.setFloat(1, bidpriceReal);
        ps.setString(2, currentDate);
        ps.setString(3, uid);
        ps.setString(4, itemid);
        ps.execute();
```

```
        redirectURL = "stack.jsp";
        response.sendRedirect(redirectURL);
        }

        redirectURL = "stack.jsp";
        response.sendRedirect(redirectURL);
        }


%>


<%
        if (con != null) {
        con.close();
        con = null;
        System.out.println("DB connection closed successfully.");
        }
%>
```

Biderror.jsp

```
<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <link rel="stylesheet" type="text/css" href="css/about.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <script src="js/aboutjquery.js"></script>
   <title>eAuction</title>
</head>
<body>
   <div id="header">
         <div id="logobox">
                <a href="#">eAuction-Scheme.com</a>
         </div>
   </div>

        <div id="navbar">
         <div id="menubox">
                <ul>
                        <li><a href="index.jsp">HOME</a></li>
                        <li><a href="account.jsp">ACCOUNT</a></li>
```

```
                        <li><a href="sell.jsp">SELL</a></li>
                        <li><a href="stack.jsp">BUY</a></li>
                        <li><a href="activity.jsp" >ACTIVITY</a></li>
                </ul>
        </div>
    </div>

    <div id="container">
        <h2 style="margin-top: 15px; text-align: center; font-weight: normal;">
                Bid price is lower than current price!</br>
                Bid price must be higher than 5% of current price!</br></br>
        </h2>
        </div>

</body>
</html>
```

Buyitnow.jsp

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.text.DateFormat"%>
<%@page import="java.util.*"%>
<%
 Connection con=null;
        ResultSet rs=null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
        System.out.println("Connection established.");
        }
        catch(SQLException sqle)
        {
        System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
        }
        catch(ClassNotFoundException cnfe)
```

```
{
System.out.println("Class Exception:"+cnfe.getMessage());
con=null;
}
catch(Exception e)
{
System.out.println("General Exception:"+e.getMessage());
con=null;
}

String itemid = request.getParameter("itemidn");
String uid = request.getParameter("userid");
String redirectURL;
int bidcount = -1;

String sql = "SELECT count(*) FROM BIDS WHERE uid = '" + uid + "' and item_id = " +
itemid + "";
Statement s = null;

try{
s = con.createStatement();
rs = s.executeQuery(sql);

while (rs.next()) {
        bidcount = rs.getInt(1);
}
}
catch(Exception e){System.out.println("SQL error");}

System.out.println("itemid: " + itemid);
System.out.println("uid: " + uid);
System.out.println("bidcount: " + bidcount);

java.util.Date date = new java.util.Date();
DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
String currentDate = dateFormat.format(date);


String sql1 = "SELECT buyitnow_price FROM ITEM_SELL_BY WHERE item_id = " +
itemid + "";
Statement s1 = null;
float price = 0;

try{
s1 = con.createStatement();
```

```
	rs = s1.executeQuery(sql1);

	while (rs.next()) {
		price = rs.getInt(1);
	}
	}
	catch(Exception e){System.out.println("SQL error");}

	if (bidcount > 0) {
	PreparedStatement ps11=con.prepareStatement("DELETE FROM BIDS WHERE uid='" +
uid + "' and item_id=" + itemid);
	ps11.execute();

	redirectURL = "stack.jsp";
	response.sendRedirect(redirectURL);
	}
	PreparedStatement ps4=con.prepareStatement("UPDATE ITEM_SELL_BY SET
end_date='"+currentDate+"', current_price=" + price
		+ " WHERE item_id = " + itemid);
	ps4.execute();

	redirectURL = "exceedReserve.jsp";
	response.sendRedirect(redirectURL);
%>


<%
	if (con != null) {
	con.close();
	con = null;
	System.out.println("DB connection closed successfully.");
	}
%>
```

compReg.jsp

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%	Connection con=null;
	Statement stat=null;
	ResultSet rs=null;
	try
	{
	Class.forName("org.apache.derby.jdbc.ClientDriver");
```

```
if(con!=null)
{
        con.close();
con=null;
}
con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
System.out.println("Connection established.");
}
catch(SQLException sqle)
{
System.out.println("SQL Exception:"+sqle.getMessage());
con=null;
}
catch(ClassNotFoundException cnfe)
{
System.out.println("Class Exception:"+cnfe.getMessage());
con=null;
}
catch(Exception e)
{
System.out.println("General Exception:"+e.getMessage());
con=null;
}

Integer uid = -1;
String id = request.getParameter("id");
String pwd = request.getParameter("pwd");
String name = request.getParameter("name");
String state = request.getParameter("state");
String city = request.getParameter("city");
String zip = request.getParameter("zip");
String addr1 = request.getParameter("addr1");
String addr2 = request.getParameter("addr2");
String addr3 = request.getParameter("addr3");
String phone = request.getParameter("phone");
String revenue = request.getParameter("revenue");
String category = request.getParameter("category");
String url = request.getParameter("url");
String contact = request.getParameter("contact");

String sql = "SELECT count(*) FROM USERS";
Statement s = null;

try{
s = con.createStatement();
```

```
rs = s.executeQuery(sql);

while (rs.next()) {

uid = rs.getInt(1);

}
}
catch(Exception e){e.printStackTrace();}

String uidstr = Integer.toString(uid);

PreparedStatement ps=con.prepareStatement("INSERT INTO USERS VALUES
(?,?,?,?,?)");
ps.setString(1, uidstr);
ps.setString(2, id);
ps.setString(3, name);
ps.setString(4, phone);
ps.setString(5, pwd);
ps.execute();


int revenueint;
if ( revenue == "" ) {
revenueint = 0;
} else {
revenueint = Integer.parseInt(revenue);
}

PreparedStatement ps2=con.prepareStatement("INSERT INTO COMPANY VALUES
(?,?,?,?,?)");
ps2.setString(1, url);
ps2.setInt(2, revenueint);
ps2.setString(3, category);
ps2.setString(4, contact);
ps2.setString(5, uidstr);
ps2.execute();


PreparedStatement ps3=con.prepareStatement("INSERT INTO LIVES_ADDRESS
VALUES (?,?,?,?,?,?,?)");
ps3.setString(1, state);
ps3.setString(2, zip);
ps3.setString(3, city);
ps3.setString(4, addr1);
```

```
        ps3.setString(5, addr2);
        ps3.setString(6, addr3);
        ps3.setString(7, uidstr);
        ps3.execute();

        Cookie uidCK = new Cookie("uidCKscheme", uidstr);
        response.addCookie( uidCK );

        Cookie nameCK = new Cookie("nameCKscheme", name);
        response.addCookie( nameCK );

%>


<%
        if (con != null) {
        con.close();
        con = null;
        System.out.println("DB connection closed successfully.");
        }


        String redirectURL = "index.jsp";
        response.sendRedirect(redirectURL);
%>
```

exceedReserve.jsp

```
<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <link rel="stylesheet" type="text/css" href="css/about.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <script src="js/aboutjquery.js"></script>
   <title>eAuction</title>
</head>
<body>
   <div id="header">
         <div id="logobox">
                <a href="index.jsp">eAuction-Scheme.com</a>
         </div>
   </div>
```

```
        <div id="navbar">
         <div id="menubox">
                <ul>
                        <li><a href="index.jsp">HOME</a></li>
                        <li><a href="account.jsp">ACCOUNT</a></li>
                        <li><a href="sell.jsp">SELL</a></li>
                        <li><a href="stack.jsp">BUY</a></li>
                        <li><a href="activity.jsp" >ACTIVITY</a></li>
                </ul>
         </div>
   </div>

   <div id="container">
        <h2 style="margin-top: 15px; text-align: center; font-weight: normal;">
                Congratulations!</br>
                You just bought this item!
        </h2>
        <form method="post" action="stack.jsp">
        <button class="button2" style="width: 200px; margin: 0 auto; margin-top: 10px;">Go
back</button>
        </form>
        </div>

</body>
</html>
```

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%      Connection con=null;
        Statement stat=null;
        ResultSet rs=null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
```

```
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
        System.out.println("Connection established.");
        }
        catch(SQLException sqle)
        {
        System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
        }
        catch(ClassNotFoundException cnfe)
        {
        System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
        }
        catch(Exception e)
        {
        System.out.println("General Exception:"+e.getMessage());
        con=null;
        }

        if(con != null)
        {
        con.close();
        con = null;
        System.out.println("Connection Terminated");
        }
%>
<%
  String uidCookie = "";
  String cookieName = "nameCKscheme";
  Cookie cookie = null;
  Cookie[] cookies = null;
  // Get an array of Cookies associated with this domain
  cookies = request.getCookies();
  if( cookies != null ){
        for (int i = 0; i < cookies.length; i++){
        cookie = cookies[i];
        if (cookies[i].getName().equals(cookieName))
                {
                uidCookie = cookies[i].getValue();
                break;
                }

        }
  }
```

```
%>

<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <link rel="stylesheet" type="text/css" href="css/index.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <script src="js/index.js"></script>
   <title>eAuction</title>
</head>
<body>
   <div id="background"></div>

   <form action="login.jsp" method="post">
   <table id="loginTable">
                <tr>
                        <td colspan="2" style="text-align: center;"><h1>Login</h1></td>
                </tr>
                 <tr>
                         <td><p>ID: </p></td>
                         <td><input style="height:80%;" name="id" type="text" value=""/></td>
                 </tr>

                 <tr>
                         <td><p>Password: </p></td>
                         <td><input style="height:80%;" name="pwd" type="password"
value=""/></td>
                 </tr>
                 <tr>
                         <td colspan="2"><button class="button1">Login</button></td>
                 </tr>
   </table>
   </form>

         <form action="userReg.jsp" method="post">
   <table id="registerTable">
                <tr>
                        <td colspan="2" style="text-align: center;"><h1>Register</h1></td>
                </tr>
                 <tr>
                         <td colspan='2' style="text-align: center;"><a id="registerCompanyLink"
class="button1">Register as Company</a></td>
```

```
</tr>

<tr>
        <td><p>Email (ID): </p></td>
        <td><input name="id" type="text" value=""/></td>
</tr>

<tr>
        <td><p>Password: </p></td>
        <td><input name="pwd" type="password" value=""/></td>
</tr>

<tr>
        <td><p>Confirm: </p></td>
        <td><input name="confpwd" type="password" value=""/></td>
</tr>

<tr>
        <td><p>Name: </p></td>
        <td><input name="name" type="text" value=""/></td>
</tr>

<tr>
        <td><p>State: </p></td>
        <td><select name="state">
                <option value="AL">Alabama</option>
                <option value="AK">Alaska</option>
                <option value="AZ">Arizona</option>
                <option value="AR">Arkansas</option>
                <option value="CA">California</option>
                <option value="CO">Colorado</option>
                <option value="CT">Connecticut</option>
                <option value="DE">Delaware</option>
                <option value="DC">District Of Columbia</option>
                <option value="FL">Florida</option>
                <option value="GA">Georgia</option>
                <option value="HI">Hawaii</option>
                <option value="ID">Idaho</option>
                <option value="IL">Illinois</option>
                <option value="IN">Indiana</option>
                <option value="IA">Iowa</option>
                <option value="KS">Kansas</option>
                <option value="KY">Kentucky</option>
                <option value="LA">Louisiana</option>
                <option value="ME">Maine</option>
```

```
                <option value="MD">Maryland</option>
                <option value="MA">Massachusetts</option>
                <option value="MI">Michigan</option>
                <option value="MN">Minnesota</option>
                <option value="MS">Mississippi</option>
                <option value="MO">Missouri</option>
                <option value="MT">Montana</option>
                <option value="NE">Nebraska</option>
                <option value="NV">Nevada</option>
                <option value="NH">New Hampshire</option>
                <option value="NJ">New Jersey</option>
                <option value="NM">New Mexico</option>
                <option value="NY">New York</option>
                <option value="NC">North Carolina</option>
                <option value="ND">North Dakota</option>
                <option value="OH">Ohio</option>
                <option value="OK">Oklahoma</option>
                <option value="OR">Oregon</option>
                <option value="PA">Pennsylvania</option>
                <option value="RI">Rhode Island</option>
                <option value="SC">South Carolina</option>
                <option value="SD">South Dakota</option>
                <option value="TN">Tennessee</option>
                <option value="TX">Texas</option>
                <option value="UT">Utah</option>
                <option value="VT">Vermont</option>
                <option value="VA">Virginia</option>
                <option value="WA">Washington</option>
                <option value="WV">West Virginia</option>
                <option value="WI">Wisconsin</option>
                <option value="WY">Wyoming</option>
        </select></td>
</tr>

<tr>
        <td><p>City: </p></td>
        <td><input name="city" type="text" value=""/></td>
</tr>

<tr>
        <td><p>ZIP: </p></td>
        <td><input name="zip" type="text" value=""/></td>
</tr>

<tr>
```

```
                <td><p>Address1: </p></td>
                <td><input name="addr1" type="text" value=""/></td>
            </tr>

            <tr>
                <td><p>Address2: </p></td>
                <td><input name="addr2" type="text" value=""/></td>
            </tr>

            <tr>
                <td><p>Address3: </p></td>
                <td><input name="addr3" type="text" value=""/></td>
            </tr>

            <tr>
                <td><p>Phone: </p></td>
                <td><input name="phone" type="text" value=""/></td>
            </tr>

            <tr>
                <td><p>Gender: </p></td>
                <td>  <input name="gender" type="radio"
value="Male"/>   Male</td>
            </tr>
            <tr><td></td>

                        <td>  <input name="gender" type="radio"
value="Female"/>   Female</td>
            </tr>

            <tr>
                <td><p>Date of Birth: </p></td>
                <td><select name="dobmonth">
                <option value="default">Month</option>
                <option value="01">Jan</option>
                <option value="02">Feb</option>
                <option value="03">Mar</option>
                <option value="04">Apr</option>
                <option value="05">May</option>
                <option value="06">Jun</option>
                <option value="07">Jul</option>
                <option value="08">Aug</option>
                <option value="09">Sep</option>
                <option value="10">Oct</option>
                <option value="11">Nov</option>
```

```
        <option value="12">Dec</option>
        </select></td></tr>

<tr>
        <td></td>
        <td><input name="dobday" type="text" value=""/></td></tr>
<tr>
        <td></td>
        <td><select name="dobyear">
        <option value="default">Year</option>
        <option value="2003">2003</option>
        <option value="2002">2002</option>
        <option value="2001">2001</option>
        <option value="2000">2000</option>
        <option value="1999">1999</option>
        <option value="1998">1998</option>
        <option value="1997">1997</option>
        <option value="1996">1996</option>
        <option value="1995">1995</option>
        <option value="1994">1994</option>
        <option value="1993">1993</option>
        <option value="1992">1992</option>
        <option value="1991">1991</option>
        <option value="1990">1990</option>
        <option value="1989">1989</option>
        <option value="1988">1988</option>
        <option value="1987">1987</option>
        <option value="1986">1986</option>
        <option value="1985">1985</option>
        <option value="1984">1984</option>
        <option value="1983">1983</option>
        <option value="1982">1982</option>
        <option value="1981">1981</option>
        <option value="1980">1980</option>
        <option value="1979">1979</option>
        <option value="1978">1978</option>
        <option value="1977">1977</option>
        <option value="1976">1976</option>
        </select>
        </td>
</tr>
<tr>
        <td>Income ($):</td>
        <td><input name="income" type="text" value=""/></td>
</tr>
```

```
                <tr>
                        <td colspan="2"><button id="regbutton"
class="button1">Register</button></td>
                </tr>
    </table>
    </form>

    <form action="compReg.jsp" method="post">
    <table id="registerCompanyTable">
                <tr>
                        <td colspan="2"><h1>Register as Company</h1></td>
                </tr>
                <tr>
                        <td><p>Email (ID): </p></td>
                        <td><input name="id" type="text" value=""/></td>
                </tr>

                <tr>
                        <td><p>Password: </p></td>
                        <td><input name="pwd" type="password" value=""/></td>
                </tr>

                <tr>
                        <td><p>Confirm: </p></td>
                        <td><input name="confpwd" type="password" value=""/></td>
                </tr>

                <tr>
                        <td><p>Name: </p></td>
                        <td><input name="name" type="text" value=""/></td>
                </tr>

                <tr>
                        <td><p>State: </p></td>
                        <td><select name="state">
                                <option value="AL">Alabama</option>
                                <option value="AK">Alaska</option>
                                <option value="AZ">Arizona</option>
                                <option value="AR">Arkansas</option>
                                <option value="CA">California</option>
                                <option value="CO">Colorado</option>
                                <option value="CT">Connecticut</option>
                                <option value="DE">Delaware</option>
                                <option value="DC">District Of Columbia</option>
                                <option value="FL">Florida</option>
```

```
                <option value="GA">Georgia</option>
                <option value="HI">Hawaii</option>
                <option value="ID">Idaho</option>
                <option value="IL">Illinois</option>
                <option value="IN">Indiana</option>
                <option value="IA">Iowa</option>
                <option value="KS">Kansas</option>
                <option value="KY">Kentucky</option>
                <option value="LA">Louisiana</option>
                <option value="ME">Maine</option>
                <option value="MD">Maryland</option>
                <option value="MA">Massachusetts</option>
                <option value="MI">Michigan</option>
                <option value="MN">Minnesota</option>
                <option value="MS">Mississippi</option>
                <option value="MO">Missouri</option>
                <option value="MT">Montana</option>
                <option value="NE">Nebraska</option>
                <option value="NV">Nevada</option>
                <option value="NH">New Hampshire</option>
                <option value="NJ">New Jersey</option>
                <option value="NM">New Mexico</option>
                <option value="NY">New York</option>
                <option value="NC">North Carolina</option>
                <option value="ND">North Dakota</option>
                <option value="OH">Ohio</option>
                <option value="OK">Oklahoma</option>
                <option value="OR">Oregon</option>
                <option value="PA">Pennsylvania</option>
                <option value="RI">Rhode Island</option>
                <option value="SC">South Carolina</option>
                <option value="SD">South Dakota</option>
                <option value="TN">Tennessee</option>
                <option value="TX">Texas</option>
                <option value="UT">Utah</option>
                <option value="VT">Vermont</option>
                <option value="VA">Virginia</option>
                <option value="WA">Washington</option>
                <option value="WV">West Virginia</option>
                <option value="WI">Wisconsin</option>
                <option value="WY">Wyoming</option>
        </select></td>
</tr>

<tr>
```

```
        <td><p>City: </p></td>
        <td><input name="city" type="text" value=""/></td>
</tr>

<tr>
        <td><p>ZIP: </p></td>
        <td><input name="zip" type="text" value=""/></td>
</tr>

<tr>
        <td><p>Address1: </p></td>
        <td><input name="addr1" type="text" value=""/></td>
</tr>

<tr>
        <td><p>Address2: </p></td>
        <td><input name="addr2" type="text" value=""/></td>
</tr>

<tr>
        <td><p>Address3: </p></td>
        <td><input name="addr3" type="text" value=""/></td>
</tr>

<tr>
        <td><p>Phone: </p></td>
        <td><input name="phone" type="text" value=""/></td>
</tr>

<tr>
        <td><p>Revenue: </p></td>
        <td><input name="revenue" type="text" value=""/></td>
</tr>

<tr>
        <td><p>Category: </p></td>
        <td><select name="category">
        <option value="">Choose...</option>
        <option value="">Category1</option>
        <option value="">Category2</option>
        <option value="">Category3</option>
        </select></td>
</tr>

<tr>
```

```
                <td>URL:</td>
                <td><input name="url" type="text" value=""/></td>
            </tr>

            <tr>
                <td>Contact:</td>
                <td><input name="contact" type="text" value=""/></td>
            </tr>

            <tr>
                <td colspan="2"><button style="margin-top: 15px;"
class="button1">Register</button></td>
            </tr>
    </table>
    </form>

    <div id="header">
        <div id="logobox">
            <a href="index.jsp">eAuction-Scheme.com</a>
        </div>
    </div>

    <div id="navbar">
        <div id="menubox">
            <ul>
                <li><a href="index.jsp" class="activeIndex">HOME</a></li>
                <li><a href="account.jsp">ACCOUNT</a></li>
                <li><a href="sell.jsp">SELL</a></li>
                <li><a href="stack.jsp">BUY</a></li>
                <li><a href="activity.jsp" >ACTIVITY</a></li>
            </ul>
        </div>
    </div>

    <div id="container">
        <div id="homeleftbox">
                <hr/><p id="welcomearticle">Welcome! <span style="color:
#cc7a29;"><%=uidCookie%></span></p><br/><hr/>

            <p class="article">EAuction is a world-renowned online marketplace that makes
buying and
                selling easy. Unlike many competitor marketplaces, which require you to ship
                merchandise to your customers, EAuction buyers and sellers only interface with
                EAuction itself. This allows buyers and sellers to both protect their privacy
                and avoid messy shipping details. </p><br/><hr/>
```

```
                <p class="article">EAuction's intuitive design is easy to use and free from the
                        clutter found on other competitor sites. Our devoted staff work twenty-four
hours,
                        seven days a week to ensure that your needs are met. Start bidding on
EAuction today!</p><br/><hr/>

                <noscript style="color: red;">Please enable Javascript in order to navigate this
website!</noscript>
                <%
                if (uidCookie == "") {
                %>
                        <a id="loginLink">Login</a>

                        <a id="registerLink">Register</a>
                <%
                }
                else {
                %>
                        <a id="logoutLink" href="logout.jsp">Logout</a>
                <%
                }
                %>
        </div>

        <div id="homerightbox">
                <img id="homeimg" src="image/PennStateLivesHere.png"/>

        </div>
    </div>

    <div id="footer">
        <hr/>
        <p id="copyright">Copyright &copy; 2014. All Rights Reserved.</p>



    </div>
</body>
</html>

<%
        if (con != null) {
        con.close();
        con = null;
```

```
        System.out.println("DB connection closed successfully.");
        }
%>
```

infoSucc.jsp

```
<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <link rel="stylesheet" type="text/css" href="css/about.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <script src="js/aboutjquery.js"></script>
   <title>eAuction</title>
</head>
<body>
        <div id="background"></div>
   <div id="header">
        <div id="logobox">
                <a href="index.jsp">eAuction-Scheme.com</a>
        </div>
   </div>

   <div id="navbar">
        <div id="menubox">
                <ul>
                        <li><a href="index.jsp">HOME</a></li>
                        <li><a href="account.jsp">ACCOUNT</a></li>
                        <li><a href="sell.jsp" class="activeIndex">SELL</a></li>
                        <li><a href="stack.jsp">BUY</a></li>
                        <li><a href="activity.jsp">ACTIVITY</a></li>
                </ul>
        </div>
   </div>

   <div id="container">
        <div id="blogHeader"><h2 style="line-height: 170%; margin-left: 10px;">SUCCESSFULLY
COMPLETE</h2></div>
        <div id="blogContent">
        <form id="uploadItemForm">
                <table id="uploadTable">
                        <tr></tr>
                        <tr></tr>
```

```
                        <tr></tr>
                         <tr>
                        <td> REGISTRATION SUCCESSFUL!! </td>
                        </tr>
                        <tr>
                        <td> <a href='index.jsp'>Go back to main</a></td>
                        </tr>
                    </table>
            </form>
                    </form>
            </div>
            <div id="blogSidebar"></div>
    </div>
    </div>

    <div id="footer">
            <hr/>
            <p id="copyright">Copyright &copy; 2013. All Rights Reserved.</p>
    </div>
</body>
</html>
```

login.jsp

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%
 Connection con=null;
        Statement stat=null;
        ResultSet rs=null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
        System.out.println("Connection established.");
        }
        catch(SQLException sqle)
```

```
{
System.out.println("SQL Exception:"+sqle.getMessage());
con=null;
}
catch(ClassNotFoundException cnfe)
{
System.out.println("Class Exception:"+cnfe.getMessage());
con=null;
}
catch(Exception e)
{
System.out.println("General Exception:"+e.getMessage());
con=null;
}


String id = request.getParameter("id");
String pwd = request.getParameter("pwd");
String uid = "";
String name = "";

String sql = "SELECT uid,name FROM USERS WHERE email = '" + id + "' and password =
'" + pwd + "'";
Statement s = null;

try{
s = con.createStatement();
rs = s.executeQuery(sql);

while (rs.next()) {

uid = rs.getString(1);
name = rs.getString(2);

}
}
catch(Exception e){e.printStackTrace();}

if ( uid != "" && name != "" ) {

Cookie uidCK = new Cookie("uidCKscheme", uid);
response.addCookie( uidCK );

Cookie nameCK = new Cookie("nameCKscheme", name);
response.addCookie( nameCK );
```

```
        String redirectURL = "index.jsp";
        response.sendRedirect(redirectURL);
        }

        else {
        String redirectURL = "loginFail.jsp";
        response.sendRedirect(redirectURL);
        }

%>
```

loginFail.jsp

```
<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <script src="js/aboutjquery.js"></script>
   <title>eAuction</title>
</head>
<body>
   <div id="header">
         <div id="logobox">
                <a href="index.jsp">eAuction-Scheme.com</a>
         </div>
   </div>

   <div id="container">
        <br/><br/><br/><br/>
        <center><h2>No such user exists!</h2></center>
        <center><h2>Or</h2></center>
        <center><h2>Password does not match!</h2></center>
        <br/><br/>
        <center><a href="index.jsp">Back</a></center>
   </div>

   <div id="footer">
         <hr/>
         <p id="copyright">Copyright &copy; 2013. All Rights Reserved.</p>
   </div>
```

```
</body>
</html>
```

logout.jsp

```
<%
   Cookie cookie = null;
   Cookie[] cookies = null;
   String cookieName = "uidCKscheme";
   String cookieName2 = "nameCKscheme";
   // Get an array of Cookies associated with this domain
   cookies = request.getCookies();
   if( cookies != null ){
          for (int i = 0; i < cookies.length; i++){
          cookie = cookies[i];
          if (cookies[i].getName().equals(cookieName)) {
          cookie.setMaxAge(0);
          response.addCookie(cookie);
          }
          if (cookies[i].getName().equals(cookieName2)) {
          cookie.setMaxAge(0);
          response.addCookie(cookie);
          }
          }
  }


          String redirectURL = "index.jsp";
          response.sendRedirect(redirectURL);
%>
```

sell.jsp

```
<%@page import="java.sql.*"%>
<%@page import="java.awt.*"%>
<%@page import="java.awt.event.*"%>
<%@page import="javax.swing.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%
   String uidCookie = "";
```

```
  String cookieName = "uidCKscheme";
  Cookie cookie = null;
  Cookie[] cookies = null;
  // Get an array of Cookies associated with this domain
  cookies = request.getCookies();
  if( cookies != null ){
        for (int i = 0; i < cookies.length; i++){
        cookie = cookies[i];
        if (cookies[i].getName().equals(cookieName))
              {
              uidCookie = cookies[i].getValue();
              }

        }
 }
%>

<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <link rel="stylesheet" type="text/css" href="css/blog.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <title>eAuction</title>
</head>
<body>
        <div id="background"></div>
   <div id="header">
         <div id="logobox">
               <a href="index.jsp">eAuction-Scheme.com</a>
         </div>
   </div>

   <div id="navbar">
         <div id="menubox">
               <ul>
                     <li><a href="index.jsp">HOME</a></li>
                     <li><a href="account.jsp">ACCOUNT</a></li>
                     <li><a href="sell.jsp" class="activeIndex">SELL</a></li>
                     <li><a href="stack.jsp">BUY</a></li>
                     <li><a href="activity.jsp">ACTIVITY</a></li>
               </ul>
         </div>
```

```
    </div>

    <div id="container">
        <% if (uidCookie != "") {%>
         <div id="blogHeader"><h2 style="line-height: 170%; margin-left: 10px;">Fill out information of
an item to be sold.(* is required)</h2></div>
         <div id="blogContent">
                <form method ="post" action="textInput.jsp">
                <table id="uploadTable">
                        <tr>
                                <td><font color="red">Item name*:</td>
                                <td colspan="3"><input name="itemName" type="text" value=""
/></td>
                        </tr>
                        <tr>
                                <td><font color="red">Starting price*:</td>
                                <td colspan="3"><input name="startPrice" type="text"
value=""/></td>
                        </tr>
                        <tr>
                                <td>Reserve price:</td>
                                <td colspan="3"><input name="reservePrice" type="text"
value=""/></td>
                        </tr>
                        <tr>
                                <td>BuyItNow price:</td>
                                <td colspan="3"><input name="buyItNowPrice" type="text"
value=""/></td>
                        </tr>
                        <tr>
                                <td>URL:</td>
                                <td colspan="3"><input name="url" type="text" value=""/></td>
                        </tr>
                        <tr>
                                <td><font color="red">Location*:</td>
                                <td colspan="3"><input name="location" type="text" value=""/></td>
                        </tr>
                        <tr>
                                <td><font color="red">Category*:</td>
                                <td colspan="3"><input name="category" type="text" value=""/></td>
                        </tr>
                        <tr>
                                <td>Picture:</td>
                               <td colspan="3"><input type="file" name="datafile">
                               <form action="uploadTable" method="post"
```

```
enctype="multipart/form-data"></td></form>
                        </tr>
                        <tr>
                                <td>Description:</td>
                                <td colspan="3"><textarea name ="txtArea"></textarea></td>
                        </tr>
                        <tr>
                                <td colspan="4" style="border: none;"><input type="submit"
value="Upload Item" /> </td>
                        </tr>
                </table>
                </form>
        </div>

        </div>
         <div id="blogSidebar"></div>


                <%} else { %>
                <h2 style="margin-bottom: 15px; text-align: center;">Please login first.</h2>
                <% } %>

   <div id="footer">
         <hr/>
         <p id="copyright">Copyright &copy; 2014. All Rights Reserved.</p>
   </div>
</body>
</html>
```

stack.jsp

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%String DEFAULT_PICTURE = "image/default.png";%>

<!DOCTYPE html>

   <%
         Connection con=null;
   Statement stat=null;
   ResultSet rs=null;
```

```
        try
   {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
                 con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme",
"scheme");
        System.out.println("Connection established.");
   }
        catch(SQLException sqle)
        {
                 System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
   }
        catch(ClassNotFoundException cnfe)
        {
                 System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
   }
        catch(Exception e)
        {
                 System.out.println("General Exception:"+e.getMessage());
        con=null;
   }
%>

<%
   String uidCookie = "";
   String cookieName = "uidCKscheme";
   Cookie cookie = null;
   Cookie[] cookies = null;
   // Get an array of Cookies associated with this domain
   cookies = request.getCookies();
   if( cookies != null ){
        for (int i = 0; i < cookies.length; i++){
        cookie = cookies[i];
        if (cookies[i].getName().equals(cookieName))
                {
                uidCookie = cookies[i].getValue();
                break;
                }
```

```
        }
  }
%>
<html>
<head>
<script type="text/javascript">
        function showDiv(count)
        {
        document.getElementById('description_' + count).style.display = "table-row";
        document.getElementById('description_button_' + count).style.display = "none";
        }
        function switchPages(page, specific_cid)
        {
        var myForm = document.createElement("form");
        myForm.method = "GET";
        myForm.action = "stack.jsp";

        var input = document.createElement("input");
        input.setAttribute("name", "pagenum");
        input.setAttribute("value", page);
        myForm.appendChild(input);

        if(specific_cid !== null)
        {
                var input = document.createElement("input");
                input.setAttribute("name", "CID");
                input.setAttribute("value", specific_cid);
                myForm.appendChild(input);
        }

        document.body.appendChild(myForm);
        myForm.submit();
        document.body.removeChild(myForm);
        }

        /* function checkSelect()
        {
        var select_search = document.getElementByName("select_search")[0].value;
        return select_search;
        }

        function checkText()
        {
        var keyword = document.getElementByName("keyword")[0].value;
        return keyword;
```

```
        }

        function submitSelectAndData(select_cid, search_data)
        {
        var myForm = document.createElement("form");
        myForm.method = "GET";
        myForm.action = "stack.jsp";

        var input = document.createElement("input");
        input.setAttribute("name", "select_cid");
        input.setAttribute("value", select_cid);
        myForm.appendChild(input);

        var input = document.createElement("input");
        input.setAttribute("name", "search_data");
        input.setAttribute("value", search_data);
        myForm.appendChild(input);

        document.body.appendChild(myForm);
        myForm.submit();
        document.body.removeChild(myForm);
        }*/

</script>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <link rel="stylesheet" type="text/css" href="css/stdstyle.css">
   <link rel="stylesheet" type="text/css" href="css/stack.css">
   <script src="js/jquery.js"></script>
   <script src="js/stdjquery.js"></script>
   <title>eAuction</title>

   <style>

   </style>
</head>
<body>
   <div id="background"></div>


   <div id="header">
         <div id="logobox">
                <a href="index.jsp">eAuction-Scheme.com</a>
         </div>
   </div>
```

```
<div id="navbar">
	<div id="menubox">
		<ul>
			<li><a href="index.jsp">HOME</a></li>
			<li><a href="account.jsp">ACCOUNT</a></li>
			<li><a href="sell.jsp">SELL</a></li>
			<li><a href="stack.jsp" class="activeIndex">BUY</a></li>
			<li><a href="activity.jsp" >ACTIVITY</a></li>
		</ul>
	</div>
</div>

	<% if (uidCookie != "") { %>
	<%
	try
	{
	String category_name = "CATEGORY";
	String general_cid; //used to search for subcategories
	String specific_cid = null; // represents specific category
	String implicit_cid;
	String pagenum;
	String select_cid, search_data, general_select_cid = "";
	int super_categories = 0, curpage, rowcount = -1;



	select_cid = request.getParameter("select_search");
	search_data = request.getParameter("keyword");


	specific_cid = request.getParameter("CID");
	if(specific_cid == null)
		specific_cid = "0000000000";

	pagenum = request.getParameter("pagenum");

	if(pagenum == null)
		curpage = 1;
	else
		curpage = Integer.parseInt(pagenum);

	if(search_data != null && select_cid != null)
	{
		if(select_cid.equals("0000000000") || select_cid.equals("0") )
		{
```

```
                category_name = "KEYWORD > " + search_data;
                select_cid = "0000000000";
                }
                else
                {
                stat=con.createStatement();
                rs=stat.executeQuery("SELECT CID, CATEGORY FROM CATEGORY_MAP C
WHERE C.CID LIKE '" + select_cid + "'");
                rs.next();

                category_name = "KEYWORD > " + search_data + " [" +
rs.getString("CATEGORY") + "]";
                }

                general_cid = "%%%%%%%%%%";
                specific_cid = "0000000000";
                general_select_cid = "";

                if(select_cid != null)
                {
                for(int i = 0; i < select_cid.length(); i++)
                {
                if(select_cid.charAt(i) == '0')
                {
                        general_select_cid+="%";
                }
                else
                {
                        general_select_cid+=select_cid.charAt(i);
                }
                }
                }

        }
        else if(specific_cid.equals("0000000000") || specific_cid.equals("0"))
        {
                category_name += " > ALL";
                general_cid = "%%%%%%%%%%";
        }
        else
        {
                // general_cid equals specific_cid with % instead of 0
                general_cid = "";

                // Count number of categories in category name string
```

```
		for(int i = 0; i < specific_cid.length(); i++)
		{
		if(specific_cid.charAt(i) != '0')
		{
		general_cid+=specific_cid.charAt(i);
		super_categories++;
		}
		else
		{
		general_cid+='%';
		}

		}

		// Get implicit categories from general_cid
		for(int i = 0; i < super_categories; i++)
		{
		category_name+=" > ";
		implicit_cid = "";
		for(int j = 0; j < general_cid.length(); j++)
		{
		if(j > i)
			implicit_cid+='%';
		else
		{
			implicit_cid+=general_cid.charAt(j);
		}
		}

		stat=con.createStatement();
		rs=stat.executeQuery("SELECT CID, CATEGORY FROM CATEGORY_MAP C
WHERE C.CID LIKE '" + implicit_cid + "'");
		rs.next();
		category_name+=rs.getString("CATEGORY");
		}
	}
	%>


  <div id="container">
	<div id="stackheader">
		<form method="" action="">
		 <h1><%=category_name%></h1>
		<div id="searchbox">
			<select name="select_search">
```

```
                        <option value="0000000000">Select category</option>
                        <%
                                stat=con.createStatement();
                                rs=stat.executeQuery("SELECT CID, CATEGORY FROM
CATEGORY_MAP C WHERE C.CID LIKE '%000000000'");

                                while(rs.next()){

                        %>

                        <option
value="<%=rs.getString("CID")%>"><%=rs.getString("CATEGORY")%></option>
                        <%} %>
                        </select>
                        <input type="text" name="keyword"/>
                        <div id="searchBbox">
                        <button class="button2" type="submit"
onclick="submitSelectAndData(checkSelect(), checkText());">Search</button>
                        </div>
                </div>
                </form>
        </div>


        <div id="stackcontainer">
                <div id="stackContent">
                        <div id="contentMargin" style="min-height: 1000px;">

                        <%
                        if(search_data != null)
                        {
                                stat=con.createStatement();
                                rs=stat.executeQuery("SELECT COUNT(*) as count FROM
ITEM_SELL_BY WHERE CID LIKE '" + general_select_cid + "' AND NAME='" + search_data +
"'");
                                rs.next();


                                if(rs.getString("count") != null)
                                rowcount = Integer.parseInt(rs.getString("count"));
                                else
                                rowcount = 0;


                                if(rowcount == 0)
```

```
{
%> <div class="itemBox">

No results found.

</div> <%
}


stat=con.createStatement();
rs=stat.executeQuery("SELECT
ITEM_ID,DESCRIPTION,CID,NAME,STARTING_PRICE,CURRENT_PRICE,
BUYITNOW_PRICE, LOCATION, UID, URL FROM ITEM_SELL_BY WHERE CID LIKE '" +
general_select_cid + "' AND NAME = '" + search_data + "'");

}
else
{
stat=con.createStatement();
rs=stat.executeQuery("SELECT COUNT(*) as count FROM
ITEM_SELL_BY WHERE CID LIKE '" + general_cid + "'");
rs.next();

if(rs.getString("count") != null)
rowcount = Integer.parseInt(rs.getString("count"));
else
rowcount = 0;


if(rowcount == 0)
{
%> <div class="itemBox">

No results found.

</div> <%
}


stat=con.createStatement();
rs=stat.executeQuery("SELECT
ITEM_ID,DESCRIPTION,CID,NAME,STARTING_PRICE,CURRENT_PRICE,
BUYITNOW_PRICE, LOCATION, UID, URL FROM ITEM_SELL_BY WHERE CID LIKE '" +
general_cid + "'");
```

```
        int seen = (curpage - 1)*20;
        for(int i = 0; i < seen; i++)
        {
        rs.next();
        }
}


int count = 0;

while(rs.next())
{
        if(count == 20  && search_data != null)
        {
        count = 0;
        break;
        }
        else
        count++;

        int itemid = rs.getInt("ITEM_ID");
        String itemidstr = Integer.toString(itemid);
        String name=rs.getString("NAME");
        String starting_price=rs.getString("STARTING_PRICE");
        String current_price=rs.getString("CURRENT_PRICE");
        String buyitnow_price=rs.getString("BUYITNOW_PRICE");
        String location=rs.getString("LOCATION");
        String uid=rs.getString("UID");
        String url=rs.getString("URL");
        String description=rs.getString("DESCRIPTION");

        if(name == null)
        name = "N/A";
        if(starting_price == null)
        starting_price = "N/A";
        else
        starting_price="$" + starting_price;
        if(current_price == null)
        current_price = "N/A";
        else
        current_price = "$" + current_price;
        if(buyitnow_price == null)
        buyitnow_price = "N/A";
        else
        buyitnow_price = "$" + buyitnow_price;
```

```
                if(location == null)
                location = "N/A";
                if(uid == null)
                uid = "N/A";
                if(url == null)
                url = DEFAULT_PICTURE;
                if(description == null)
                description = "No description.";


        %>



                <div class="itemBox">
                <h2><%=name%></h2>
                        <div class="itemPicBox"><img src="<%=url%>"/></div>
                        <div class="itemDesBox">
                                <table class="itemDesTable">
                                        <tr>
                                                <td>Starting price:<td>
                                                <td><%=starting_price%></td>
                                        </tr>
                                        <tr>
                                                <td>Current price:<td>
                                                <td><%=current_price%></td>
                                        </tr>
                                        <tr>
                                                <td>BuyItNow price:<td>
                                                <td><%=buyitnow_price%></td>
                                        </tr>
                                        <tr>
                                                <td>Location:<td>
                                                <td><%=location%></td>
                                        </tr>
                                        <tr>
                                                <td>Seller:<td>
                                                <td><%=uid%></td>
                                        </tr>
                                        <tr id=description_button_<%=count%> >

                                       <td colspan="3" style="border: none;">
                                       <button class="button1" style="width: 200px;
margin: auto; font-size: 12px;" onclick="showDiv(<%=count%>);">See description</button>
                                       <td>
```

```
                                                    </tr>
                                                    <tr id=description_<%=count%> style="display:
none;"><td colspan="3" style="min-height: 20px; width: 100px; overflow: auto;">
                                                    <p style="text-align: left;"><%=description%></p>
                                                    </td>

                                                    </tr>

                                                     </table>
                                            </div>
                                            <div class="itemButBox">
                                           <form method="post" action="bid.jsp">
                                           <input type="hidden" name="itemidb"
value=<%=itemidstr%>></input>
                                           <input type="hidden" name="userid"
value=<%=uidCookie%>></input>
                                           <input type="text" name="bidprice" class="bidinput"
value="Type bid amount"></input>
                                           <button type='submit' class="button2">Bid</button>
                                           </form>

                                           <form method="post" action="watch.jsp">
                                           <input type="hidden" name="itemidw"
value=<%=itemid%>></input>
                                           <input type="hidden" name="userid"
value=<%=uidCookie%>></input>
                                           <button type='submit' class="button2">Watch</button>
                                           </form>

                                           <form method="post" action="buyitnow.jsp">
                                           <input type="hidden" name="itemidn"
value=<%=itemid%>></input>
                                           <input type="hidden" name="userid"
value=<%=uidCookie%>></input>
                                           <button type='submit' class="button2">BuyItNow</button>
                                           </form>
                                            </div>
                                    </div>
                            <%}%>



                            </div>
```

```
</div>

<div id="stackSidebar">
	<h2 style="text-align: center;">Subcategory</h2>
	<%
		String specific_previous_cid,immediate_subcategories;

		int flag = 0;

		// If not in 'ALL' category
		if(!general_cid.equals("%%%%%%%%%%"))
		{
		specific_previous_cid = "";
		int idx = general_cid.indexOf('%');

		// Get
		for(int i = 0; i < 10; i++)
		{
		if(i == (idx - 1))
		{
			flag = 1;
			specific_previous_cid+='0';
		}
		else if((i == general_cid.length() - 1) && (flag == 0))
		{
			// if at last character and '%' not found
			specific_previous_cid+='0';
		}
		else if(general_cid.charAt(i) == '%')
		{
			specific_previous_cid+='0';
		}
		else
		{
		specific_previous_cid+=general_cid.charAt(i);
		}
		}
		%>

		<form action="stack.jsp" method="GET">
		<button name="CID" class="button3"
value="<%=specific_previous_cid%>">PREVIOUS CATEGORY</button>
		</form>

		<%}
```

```
immediate_subcategories = "";
flag = 0;

// Get a string representing immediate subcategories cid
for(int i = 0; i < general_cid.length(); i++)
{
if(general_cid.charAt(i) == '%' && flag == 0)
{
immediate_subcategories+='%';
flag = 1;
}
else if(general_cid.charAt(i) == '%')
{
immediate_subcategories+='0';
}
else
{
immediate_subcategories+=general_cid.charAt(i);
}

}



String subcategory_cid;
String subcategory_item_counter;

stat=con.createStatement();
rs=stat.executeQuery("SELECT CID,CATEGORY FROM
CATEGORY_MAP C WHERE C.CID LIKE '" + immediate_subcategories + "'");
ResultSet rs2 = null;

while(rs.next())
{
subcategory_cid = rs.getString("CID");

subcategory_item_counter= "";
for(int i = 0; i < subcategory_cid.length(); i++)
{
if(subcategory_cid.charAt(i) == '0')
        subcategory_item_counter+='%';
else
        subcategory_item_counter+=subcategory_cid.charAt(i);
}
```

```
                                stat=con.createStatement();
                                rs2=stat.executeQuery("SELECT COUNT(*) as count FROM
ITEM_SELL_BY WHERE CID LIKE '" + subcategory_item_counter + "%'");
                                rs2.next();

                                if(!subcategory_cid.equals(specific_cid)){
                                %>
                                <form action="stack.jsp" method="GET">
                                <button name="CID" class="button3"
value="<%=rs.getString("CID")%>"><%=rs.getString("CATEGORY")%>(<%=rs2.getString("count")
%>)</button>
                                </form>
                                <%}
                                }

                                %>
                 </div>


                 <div id="stackFooter">
                <%

                int pages = (int)((rowcount + 19)/20);

                for(int i = 1; i <= pages; i++)
                {
                        if(curpage == i)
                        { %>
                        <p style="display: inline; margin-right: 8px;"
id="curPage"><%=curpage%></p>
                        <% }
                        else
                        { %>


                <a class="button1" onclick="switchPages(<%=i%>,
<%=specific_cid%>);"><%=i%></a>


                        <% }
                } %>
```

```
                </div>
        </div>


    </div>



        <% }

        catch(SQLException sqle)
        {
                System.out.println("SQL Exception:"+sqle.getMessage());
                sqle.printStackTrace();
        }
        catch(Exception e)
        {
                System.out.println("General Exception:"+e.getMessage());
        }
        finally
        {
                stat=null;
        }
        } else {
        %>
                <h2 style="margin-top: 15px; text-align: center; font-weight: normal;">Please login
first.</h2>

        <% } %>
</body>
</html>

<%
        if (con != null) {
        con.close();
        con = null;
        System.out.println("DB connection closed successfully.");
        }
%>
```

textInput.jsp

```
<%@ page import="java.sql.*" %>
<%@ page import="java.text.SimpleDateFormat" %>
```

```
<%@ page import="java.text.DateFormat" %>
<%@ page import="java.util.Calendar" %>
<%@ page import="java.util.Date" %>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%@ page contentType="text/html" pageEncoding="UTF-8"%>

<%
  String uidCookie = "";
  String cookieName = "uidCKscheme";
  Cookie cookie = null;
  Cookie[] cookies = null;
  // Get an array of Cookies associated with this domain
  cookies = request.getCookies();
  if( cookies != null ){
        for (int i = 0; i < cookies.length; i++){
        cookie = cookies[i];
        out.println("Cookie Found!");
        out.println(cookie.getName());
        out.println(cookie.getValue());
        if (cookies[i].getName().equals(cookieName))
                {
                uidCookie = cookies[i].getValue();
                }

        }
  }
%>

<!DOCTYPE html>
<html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>eAuction</title>
        </head>
        <body>

        <% if (uidCookie != "") {%>
        <%
        String iName = request.getParameter("itemName");
        String sPrice = request.getParameter("startPrice");
        String rPrice = request.getParameter("reservePrice");
        String bPrice = request.getParameter("buyItNowPrice");
        String urlA = request.getParameter("url");
        String loc = request.getParameter("location");
```

```
String cate = request.getParameter("category");
String file = request.getParameter("datafile");
String txt = request.getParameter("txtArea");
double intSPRICE = Double.parseDouble(sPrice);
double intBPRICE = Double.parseDouble(bPrice);
double intRPRICE = Double.parseDouble(rPrice);
/*
if(sPrice != null && rPrice == null && bPrice == null) {
        intSPRICE = Double.parseDouble(sPrice);
        intBPRICE = 0;
        intRPRICE = 0;
}
else if (rPrice != null && bPrice == null && sPrice == null) {
        intRPRICE = Double.parseDouble(rPrice);
        intBPRICE = 0;
        intSPRICE = 0;
}
else if (bPrice != null && rPrice == null && sPrice == null) {
        intBPRICE = Double.parseDouble(bPrice);
        intSPRICE = 0;
        intRPRICE = 0;
}
else {
        intSPRICE = 0;
        intBPRICE = 0;
        intRPRICE = 0;
} */

// Initialize
String userID = "00";
String cateID = "1000000000";
int itemNUM = 0;

// Current Date
java.util.Date today = new java.util.Date();
java.sql.Date sqlToday = new java.sql.Date(today.getTime());

// End Date ( Current Date + 14)
java.sql.Date endDate = new java.sql.Date(today.getTime()+ 1 * 24 * 60 * 60 * 1000 * 14);

Connection con=null;
Statement stat=null;    // get CID
Statement stat1=null;  // get ITEM_NUM
Statement stat2=null;  // get UID
ResultSet rs=null;      // get CID
```

```
ResultSet rs1=null;     // get ITEM_NUM
ResultSet rs2=null;     // get UID

try {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null) {
                con.close();
                con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme",
"scheme");
        System.out.println("Connection established.");
}
catch(SQLException sqle) {
        System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
}
catch(ClassNotFoundException cnfe) {
        System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
}
catch(Exception e) {
        System.out.println("General Exception:"+e.getMessage());
        con=null;
}
// Get the CID
try {
        stat = con.createStatement();
        rs = stat.executeQuery("select CID from CATEGORY_MAP where CATEGORY = '"
+ cate + "'");
        while(rs.next()) {
        cateID = rs.getString("CID");
        }
}
catch(Exception e){e.printStackTrace();}

// Get the Item number
try {
        stat1 = con.createStatement();
        rs1 = stat1.executeQuery("select COUNT(*) AS COUNT from ITEM_SELL_BY");
        while(rs1.next()) {
        itemNUM = rs1.getInt("COUNT") + 1;
        }
}
catch(Exception e){e.printStackTrace();}
```

```
	// Get the User ID
	try {
		stat2 = con.createStatement();
		rs2 = stat2.executeQuery("select UID from USERS where UID = '" + uidCookie + "'");
		while(rs2.next()) {
		userID = rs2.getString("UID");
		}
	}
	catch(Exception e){e.printStackTrace();}

	int i = stat.executeUpdate("insert into ITEM_SELL_BY (ITEM_ID, NAME, CID,
END_DATE, RESERVE_PRICE, STARTING_PRICE, CURRENT_PRICE, BUYITNOW_PRICE,
URL, LOCATION, DESCRIPTION, UID, DATE, PIC_ADDR) values ("+ itemNUM + ",'" + iName
+ "','" + cateID + "','" + endDate + "'," + intRPRICE + "," + intSPRICE + "," + intSPRICE + "," +
intBPRICE + ",'" + urlA + "','" + loc + "','" + txt + "','" + userID + "','" + sqlToday + "','" + file + "')");
	if (i > 0) {
		response.sendRedirect("infoSucc.jsp");
	}

	}
	%>
	</body>
</html>
```

<u>userEdit.jsp</u>

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%
  String uidCookie = "";
  String cookieName = "uidCKscheme";
  Cookie cookie = null;
  Cookie[] cookies = null;
  // Get an array of Cookies associated with this domain
  cookies = request.getCookies();
  if( cookies != null ){
	for (int i = 0; i < cookies.length; i++){
	cookie = cookies[i];
	out.println("Cookie Found!");
	out.println(cookie.getName());
	out.println(cookie.getValue());
	if (cookies[i].getName().equals(cookieName))
		{
```

```
                uidCookie = cookies[i].getValue();
                }

        }
  }
%>
<%      Connection con=null;
        ResultSet rs = null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
        System.out.println("Connection established.");
        }
        catch(SQLException sqle)
        {
        System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
        }
        catch(ClassNotFoundException cnfe)
        {
        System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
        }
        catch(Exception e)
        {
        System.out.println("General Exception:"+e.getMessage());
        con=null;
        }

        String pwd = request.getParameter("pwd");
        String name = request.getParameter("name");
        String state = request.getParameter("state");
        String city = request.getParameter("city");
        String zip = request.getParameter("zip");
        String addr1 = request.getParameter("addr1");
        String addr2 = request.getParameter("addr2");
        String addr3 = request.getParameter("addr3");
        String phone = request.getParameter("phone");
        String income = request.getParameter("income");
```

```
String revenue = request.getParameter("revenue");
String category = request.getParameter("category");
String url = request.getParameter("url");
String contact = request.getParameter("contact");

String cardtypeC[] = {"9","9","9"};
String cardtype[] = {"","",""};
String cardnum[] = {"","",""};
String cardexp[] = {"","",""};
String cardcvc[] = {"","",""};

cardtype[0] = request.getParameter("cardtype1");
cardnum[0] = request.getParameter("cardnum1");
cardexp[0] = request.getParameter("cardexp1");
cardcvc[0] = request.getParameter("cardcvc1");

cardtype[1] = request.getParameter("cardtype2");
cardnum[1] = request.getParameter("cardnum2");
cardexp[1] = request.getParameter("cardexp2");
cardcvc[1] = request.getParameter("cardcvc2");

cardtype[2] = request.getParameter("cardtype3");
cardnum[2] = request.getParameter("cardnum3");
cardexp[2] = request.getParameter("cardexp3");
cardcvc[2] = request.getParameter("cardcvc3");

String IorC = request.getParameter("IorC");



PreparedStatement ps=con.prepareStatement("UPDATE USERS SET name='" + name + "',
phone='" +
phone + "', password='"+pwd+"' WHERE uid = '" +
uidCookie + "'");
ps.execute();

PreparedStatement ps3=con.prepareStatement("UPDATE LIVES_ADDRESS SET
state='"+state+"', zip='"+zip+"', "
+ "city='"+city+"', address1='"+addr1+"', address2='"+addr2+"', address3='"+addr3+"'
WHERE uid = '" +
uidCookie + "'");
ps3.execute();
```

```
if (IorC.equals("I")) {
int incomeint;
if ( income == "" ) {
incomeint = 0;
} else {
incomeint = Integer.parseInt(income);
}
PreparedStatement ps2=con.prepareStatement("UPDATE INDIVIDUAL SET
income="+incomeint+" "
        + "WHERE uid = '" + uidCookie + "'");
ps2.execute();
}
else {
int revenueint;
if ( revenue == "" ) {
revenueint = 0;
} else {
revenueint = Integer.parseInt(revenue);
}

System.out.println("revenue: " + revenue + ", " + revenueint);

PreparedStatement ps2=con.prepareStatement("UPDATE COMPANY SET url='" + url + "',
revenue=" +
        revenueint + ", category='"+category+"', contact = '" + contact + "' WHERE uid = '" +
        uidCookie + "'");
ps2.execute();

}

String sql = "SELECT cnumber FROM CARD_INFO WHERE uid = '" + uidCookie + "'";
String prevCardNum[] = {"","",""};
Statement s1 = null;

try{
s1 = con.createStatement();
rs = s1.executeQuery(sql);
int i = 0;

while (rs.next()) {
prevCardNum[i] = rs.getString(1);
i++;
}
}
catch(Exception e){e.printStackTrace();}
```

```
sql = "SELECT count(*) FROM CARD_INFO WHERE uid = '" + uidCookie + "'";
Statement s = null;
int cardCount = 0;

try{
s = con.createStatement();
rs = s.executeQuery(sql);

while (rs.next()) {
cardCount = rs.getInt(1);
}
}
catch(Exception e){e.printStackTrace();}

int i;
for (i = 0; i < 3; i++) {
if (cardtype[i].equals(""))
break;
if (cardtype[i].equals("VISA"))
cardtypeC[i] = "0";
else if (cardtype[i].equals("MasterCard"))
cardtypeC[i] = "1";
else if (cardtype[i].equals("American Express"))
cardtypeC[i] = "2";
}

int cardFilled = i;

if (cardCount < cardFilled) {

for (i = 0; i < cardCount; i++) {
PreparedStatement ps4=con.prepareStatement("UPDATE CARD_INFO SET
type='"+cardtypeC[i]+"', cvc='"+cardcvc[i]+"', "
        + "cnumber='"+cardnum[i]+"', expdate='"+cardexp[i]+"' WHERE uid = '" +
        uidCookie + "' and cnumber = '" + prevCardNum[i] + "'");
ps4.execute();
}

for (i = cardCount; i < cardFilled; i++) {
PreparedStatement ps5=con.prepareStatement("INSERT INTO CARD_INFO VALUES
(?,?,?,?,?)");
ps5.setString(1, cardtypeC[i]);
ps5.setString(2, cardcvc[i]);
ps5.setString(3, cardnum[i]);
```

```
        ps5.setString(4, cardexp[i]);
        ps5.setString(5, uidCookie);
        ps5.execute();
        }
        }
        else {
        for (i = 0; i < cardCount; i++) {
        PreparedStatement ps4=con.prepareStatement("UPDATE CARD_INFO SET
type='"+cardtypeC[i]+"', cvc='"+cardcvc[i]+"', "
                + "cnumber='"+cardnum[i]+"', expdate='"+cardexp[i]+"' WHERE uid = '" +
                uidCookie + "' and cnumber = '" + prevCardNum[i] + "'");
        ps4.execute();
        }
        }
%>


<%
        if (con != null) {
        con.close();
        con = null;
        System.out.println("DB connection closed successfully.");
        }


        String redirectURL = "account.jsp";
        response.sendRedirect(redirectURL);
%>
```

<u>userReg.jsp</u>

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%      Connection con=null;
        Statement stat=null;
        ResultSet rs=null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
        }
```

```
con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
System.out.println("Connection established.");
}
catch(SQLException sqle)
{
System.out.println("SQL Exception:"+sqle.getMessage());
con=null;
}
catch(ClassNotFoundException cnfe)
{
System.out.println("Class Exception:"+cnfe.getMessage());
con=null;
}
catch(Exception e)
{
System.out.println("General Exception:"+e.getMessage());
con=null;
}

Integer uid = -1;
String id = request.getParameter("id");
String pwd = request.getParameter("pwd");
String name = request.getParameter("name");
String state = request.getParameter("state");
String city = request.getParameter("city");
String zip = request.getParameter("zip");
String addr1 = request.getParameter("addr1");
String addr2 = request.getParameter("addr2");
String addr3 = request.getParameter("addr3");
String phone = request.getParameter("phone");
String gender = request.getParameter("gender");
String dobmonth = request.getParameter("dobmonth");
String dobday = request.getParameter("dobday");
String dobyear = request.getParameter("dobyear");
String income = request.getParameter("income");

String sql = "SELECT count(*) FROM USERS";
Statement s = null;

try{
s = con.createStatement();
rs = s.executeQuery(sql);

while (rs.next()) {
```

```
uid = rs.getInt(1);

}
}
catch(Exception e){e.printStackTrace();}

String uidstr = Integer.toString(uid);

PreparedStatement ps=con.prepareStatement("INSERT INTO USERS VALUES
(?,?,?,?,?)");
ps.setString(1, uidstr);
ps.setString(2, id);
ps.setString(3, name);
ps.setString(4, phone);
ps.setString(5, pwd);
ps.execute();

if (gender == "Male")
gender = "M";
else gender = "F";

String birthParsed = dobyear + "-" + dobmonth + "-" + dobday;

java.sql.Date sqlDate = java.sql.Date.valueOf(birthParsed);
int incomeint;
if ( income == "" ) {
incomeint = 0;
} else {
incomeint = Integer.parseInt(income);
}

PreparedStatement ps2=con.prepareStatement("INSERT INTO INDIVIDUAL VALUES
(?,?,?,?)");
ps2.setString(1, gender);
ps2.setDate(2, sqlDate);
ps2.setInt(3, incomeint);
ps2.setString(4, uidstr);
ps2.execute();

PreparedStatement ps3=con.prepareStatement("INSERT INTO LIVES_ADDRESS
VALUES (?,?,?,?,?,?,?)");
ps3.setString(1, state);
ps3.setString(2, zip);
ps3.setString(3, city);
ps3.setString(4, addr1);
```

```
        ps3.setString(5, addr2);
        ps3.setString(6, addr3);
        ps3.setString(7, uidstr);
        ps3.execute();

        Cookie uidCK = new Cookie("uidCKscheme", uidstr);
        response.addCookie( uidCK );

        Cookie nameCK = new Cookie("nameCKscheme", name);
        response.addCookie( nameCK );

%>


<%
        if (con != null) {
        con.close();
        con = null;
        System.out.println("DB connection closed successfully.");
        }


        String redirectURL = "index.jsp";
        response.sendRedirect(redirectURL);
%>
```

watch.jsp

```
<%@page import="java.sql.*"%>
<%@page import="java.lang.*"%>
<%@page import="java.io.*"%>
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.text.DateFormat"%>
<%@page import="java.util.*"%>
<%
 Connection con=null;
        ResultSet rs=null;
        try
        {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
        con=null;
        }
```

```
con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme","scheme", "scheme");
System.out.println("Connection established.");
}
catch(SQLException sqle)
{
System.out.println("SQL Exception:"+sqle.getMessage());
con=null;
}
catch(ClassNotFoundException cnfe)
{
System.out.println("Class Exception:"+cnfe.getMessage());
con=null;
}
catch(Exception e)
{
System.out.println("General Exception:"+e.getMessage());
con=null;
}

String itemid = request.getParameter("itemidw");
String uid = request.getParameter("userid");
String redirectURL;
int watchcount = -1;


String sql = "SELECT count(*) FROM WATCHES WHERE uid = '" + uid + "' and item_id =
" + itemid + "";
Statement s = null;

try{
s = con.createStatement();
rs = s.executeQuery(sql);

while (rs.next()) {
        watchcount = rs.getInt(1);
}
}
catch(Exception e){System.out.println("SQL error");}

System.out.println("itemid: " + itemid);
System.out.println("uid: " + uid);
System.out.println("watchcount: " + watchcount);

if (watchcount > 0) {
redirectURL = "stack.jsp";
```

```
        response.sendRedirect(redirectURL);
        }
        else {
        System.out.println("aaaa ");
        PreparedStatement ps=con.prepareStatement("INSERT INTO WATCHES VALUES (?,?)");
        ps.setString(1, uid);
        ps.setString(2, itemid);
        ps.execute();

        redirectURL = "stack.jsp";
        response.sendRedirect(redirectURL);
        }
%>


<%
        if (con != null) {
        con.close();
        con = null;
        System.out.println("DB connection closed successfully.");
        }
%>
```

terminateAuction.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */


import java.sql.*;
import java.text.SimpleDateFormat;



/**
 *
 * @author adh5239
 */
public class TerminateAuctions
{

        public static void main(String[] args)
```

```
{
        TerminateAuctions cfa=new TerminateAuctions();
        cfa.connect();
        cfa.clear();
        cfa.disconnect();
}

       private Connection con;
       public TerminateAuctions ()
       {
con=null;
       }

       public void connect()
       {
Statement stat=null;
ResultSet rs=null;

        try
{
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {
                con.close();
                 con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme_root","adh5239",
"root");
        System.out.println("Connection established.");
}
        catch(SQLException sqle)
        {
                System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
}
        catch(ClassNotFoundException cnfe)
        {
                System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
}
        catch(Exception e)
        {
                System.out.println("General Exception:"+e.getMessage());
        con=null;
}
```

```
        }

        public void disconnect()
        {
                try
{
         if(con!=null)
                {
                con.close();
         }
}
catch(SQLException sqle)
{    System.out.println("SQL Exception:"+sqle.getMessage());
}
finally
{    con=null;
         System.out.println("Connection disconnected.");
}
        }

        public  void clear()
        {
        Statement statement=null;
         ResultSet rs=null;
         try
         {
        java.util.Date today = new java.util.Date();

        statement=con.createStatement();
        rs=statement.executeQuery("SELECT ITEM_ID,END_DATE FROM ITEM_SELL_BY");

        String datestr;
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        java.util.Date end_date;

        while(rs.next())
        {

                datestr = rs.getString("END_DATE");
                end_date = sdf.parse(datestr);



                if(today.after(end_date))
                {
```

```
                // delete from database
                //System.out.println(end_date.toString() + " " + today.toString());

                statement=con.createStatement();
                statement.execute("DELETE FROM ITEM_SELL_BY WHERE ITEM_ID = " +
rs.getInt("ITEM_ID"));
                }
        }
        }
        catch(SQLException sqle)
        {
                System.out.println("SQL Exception:"+sqle.getMessage());
                sqle.printStackTrace();
    }
        catch(Exception e)
        {
                System.out.println("General Exception:"+e.getMessage());
    }
    finally
    {
        statement=null;
    }
        }
}
```

reportToTelemarketer.java

```
import java.io.*;
import java.sql.*;
import java.lang.*;
import java.text.SimpleDateFormat;
import java.text.DateFormat;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;

class reportToTelemarketer
{

        public static void main(String[] args) throws FileNotFoundException
    {
        reportToTelemarketer db=new reportToTelemarketer();
        db.connect("scheme","scheme","scheme");
                try {
                db.display();
```

```
                } catch (IOException ex) {
                Logger.getLogger(reportToTelemarketer.class.getName()).log(Level.SEVERE, null,
ex);
                }
        db.disconnect();
    }

        private Connection con;
        public reportToTelemarketer()
        {
    con=null;
        }
        public void connect(String dbname, String userid, String passwd)
        {
    try
    {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        if(con!=null)
        {   con.close();
                con=null;
        }
        con=DriverManager.getConnection("jdbc:derby://localhost:1527/scheme",userid,passwd);
        System.out.println("connection established.");
    }
        catch(SQLException sqle)
        {
                System.out.println("SQL Exception:"+sqle.getMessage());
        con=null;
    }
        catch(ClassNotFoundException cnfe)
        {
                System.out.println("Class Exception:"+cnfe.getMessage());
        con=null;
    }
        catch(Exception e)
        {
                System.out.println("General Exception:"+e.getMessage());
        con=null;
    }
        }

        public void disconnect()
        {
                try
    {
```

```
        if(con!=null)
                {
                con.close();
        }
}
catch(SQLException sqle)
{    System.out.println("SQL Exception:"+sqle.getMessage());
}
finally
{    con=null;
        System.out.println("Connection disconnected.");
}
        }

        public void display() throws FileNotFoundException, IOException
        {    Statement statement=null;
         ResultSet rs=null;
        java.util.Date date = new java.util.Date();
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        String currentDate = dateFormat.format(date);
        int successful = 0;
        int total = 1;

        // Number of auctions
         try
         {
        statement=con.createStatement();

        rs=statement.executeQuery("Select count(*) from ITEM_SELL_BY");

        while(rs.next())
        {
                total = rs.getInt(1);
        }
        }
         catch(SQLException sqle)
         {
                System.out.println("SQL Exception:"+sqle.getMessage());
                sqle.printStackTrace();
}
         catch(Exception e)
         {
                System.out.println("General Exception:"+e.getMessage());
}
finally
```

```
{
        statement=null;
}

        // Percent successful
        try
        {
        statement=con.createStatement();

        rs=statement.executeQuery("Select count(*) from ITEM_SELL_BY WHERE end_date < '" +
currentDate + "'");

        while(rs.next())
        {
                successful = rs.getInt(1);
        }
        }
        catch(SQLException sqle)
        {
                System.out.println("SQL Exception:"+sqle.getMessage());
                sqle.printStackTrace();
}
        catch(Exception e)
        {
                System.out.println("General Exception:"+e.getMessage());
}
finally
{
        statement=null;
}

        int numusers = 0;

        // User info
        try
        {
        statement=con.createStatement();

        rs=statement.executeQuery("Select count(*) from ITEM_SELL_BY i, USERS u WHERE i.uid
= u.uid and end_date < '" + currentDate + "'");

        while(rs.next())
        {
                numusers = rs.getInt(1);
        }
```

```
        }
         catch(SQLException sqle)
         {
                System.out.println("SQL Exception:"+sqle.getMessage());
                sqle.printStackTrace();
  }
         catch(Exception e)
         {
                System.out.println("General Exception:"+e.getMessage());
  }
  finally
  {
         statement=null;
  }

        String itemName[] = new String[numusers];
        String userName[] = new String[numusers];
        String userMail[] = new String[numusers];
        int i = 0;

        // User info
         try
         {
        statement=con.createStatement();

        rs=statement.executeQuery("Select i.name, u.name, u.email from ITEM_SELL_BY i, USERS u
WHERE i.uid = u.uid and end_date < '" + currentDate + "'");

        while(rs.next())
        {
                itemName[i] = rs.getString(1);
                userName[i] = rs.getString(2);
                userMail[i] = rs.getString(3);
                i++;
        }

        }
         catch(SQLException sqle)
         {
                System.out.println("SQL Exception:"+sqle.getMessage());
                sqle.printStackTrace();
  }
         catch(Exception e)
         {
```

```
                System.out.println("General Exception:"+e.getMessage());
    }
    finally
    {
         statement=null;
    }

        float Psuccessful = (float)successful/(float)total;
        float Pfail = 1-Psuccessful;

        Writer writer = null;

        try {
        writer = new BufferedWriter(new OutputStreamWriter(
                new FileOutputStream("statistics.txt"), "utf-8"));
        writer.write("Total number of auctions: " + total);
        writer.write("\r\n");
        writer.write("Successful %: " + Psuccessful);
        writer.write("\r\n");
        writer.write("Not sold %: " + Pfail);
        writer.write("\r\n\r\n");

        for (int j = 0; j < numusers; j++) {
                writer.write(userName[j] + "(" + userMail[j] + ") bought " + itemName[j] + ".\r\n");
        }
        } catch (IOException ex) {
        // report
        } finally {
        try {writer.close();} catch (Exception ex) {}
        }

        }
}
```