Authors: Thomas Ansill, Kabo Cheung, Kemoy Campbell

Code Documentation

What we have finished so far:

- Database Initialization system - will initialize the database automatically if there are no database
- Sample data entries - random data to simulate
- Basic Website frontend + backend codes
    - Login using RIT's LDAP server
    - Inserting new user data in the database if the user is new
    - Posting a new book
    - Displaying the newly post book
    - Randomly select books to showcase on the index page
    - Viewing history(My book) which contain all bids and trades by the user
    - Approving or Denying the bid offer from the other user
    - Approving or Denying the trade offer from the other user
    - Composing and sending a message from a user to user
    - Displaying and ordering the messages by the time of receiving these.
    - Ability to search for books using author names, isbn or book title

Basic Login:

The login page uses RIT's LDAP to check the dce and the password supplied by the user. If the user is using the database for the first time, their name and DCE email will be fetched from LDAP and inserted into the student table of TigreNoble's database. If the user supplies an incorrect data, they will get an error message. On the other hand, if the user is a returning user, his or her data is fetched directly from the student table instead of LDAP.

Posting:

Once logged in, the user will be automatically redirected to an index page with all available books on sale. In order to post, the user needs to click on the button "post a book" in the header and supply the isbn number and all required fields of the book you want to sell. Then press "Post" and voila! A book have been added to Post database and will be displayed on the index page.

Bidding & Trading:

In order to bid or trade a book, the user needs to click on the "Bid|Buy" or "Trade" buttons on the index page. Later on, during the phase of development. This can also be done when the user search for a book.

Authors: Thomas Ansill, Kabo Cheung, Kemoy Campbell

Messaging

Inbox contains all the messages you receive from other users who may want to buy, sell, or make a deal with you. If you want to send a message, click “Compose” and write a subject and a message to the desired receiver. You can now search for the user you want to send a message to by typing his or her name.

Approving

Approving contains a list of current bids/trade offers that any user have made to the sellers’ post and the seller gets a choice to accept or reject each bid/trade offers. When a bid or trade is accepted, the post is automatically removed from the listing. If seller rejects the bid/trade, the bid or trade is deleted, and the post still stands.

History

History contains a list of history of bids and trade offers made by the seller so this seller knows which post he/she has made a bid or trade offer.

Authors: Thomas Ansill, Kabo Cheung, Kemoy Campbell

Name of database: tigreNoble

SQL FILES (located in database folder):

updateTiger.sql: this file has the tables and their respectively data.
storedProcedure.sql: this file has the stored procedures.

Primary Keys:

dce in Student
isbn in Book
id in Post
id in Inbox

Foreign Keys:

bookisbn in Post referring to Book isbn

Stored Procedures:

createStudent
adds student in the students table

getBook
retrieves book based on isbn

getHighestBid
get highest bid for that post

insertBook
adds book into the books table

insertPost
inserts post into posts table

makeBid_Money
make a bid on post and record a bid on bid_money table(deprecated)

makeBid_trade
make a trade offer on post and record a trade offer on bid_trade table(Deprecated)

getPost
retrieves a post on id number

searchPost
searches all posts based on search query

acceptMoneyBid
update the bid's status to accept

acceptMoneyBidId
update the bid's status to accept

rejectMoneyBid
update the bid's status to reject

rejectMoneyBidId
update the bid's status to reject

acceptTradeBid
update the bid's status to accept

acceptTradeBidId
update the bid's status to accept

rejectTradeBid
update the bid's status to reject

rejectTradeBidId
update the bid's status to reject

getListOfBidsMoney
gets the list of bids from bid_money on bidder's dce

getListOfBidsTrade
gets the list of trade offers from bid_trade on bidder's dce

getListOfBidsMoneySeller
gets the list of bids from bid_money on seller's dce

getListOfBidsTradeSeller
gets the list of trade offers from bid_trade on seller's dce

makeMoneyBid
make a bid and record it in bid_money table

makeTradeBid

make a trade offer and record it in bid_trade table

getRandomListOfPosts
get a list of random posts to populate the dashboard

getListOfMessages
get a list of inbox messages

getMessage
get a specific message with message inside of it

sendMessage
sends a message and add it to the inbox table

getListofBidsMoneySellerWithMoreInfo
enhanced stored procedure of get list of bids Money

getListOfBidsTradeSellerWithMoreInfo
enhanced stored procedure of get list of bids Trade

Indexes:

Inbox
Post
Bid_money
Bid_trade