Case Study: EAuction – Auction System

Group 3

Team member Emp ID	Team member Name	Team Member Email Id
46283118	**DEEP SHIKHA**	**deepshikha.deepshikha@capgemini.com**
46292703	Harsh Prasad	harsh.prasad@capgemini.com
46283148	Raixi Singh	raixi.singh@capgemini.com
46292705	Manish Saha	manish.a.saha@capgemini.com
46283153	A. Karishma Rani Patro	a-karishma-rani.patro@capgemini.com
46283106	manjulika sahoo	manjulika.sahoo@capgemini.com

PROBLEM S TATEMENT

1.2 O BJECTIVE

To create an online auction application EAuction where users can live bid. The application is to be develope as Executable file compiled on Linux. There are 2 entities Admin and User.

1.3 A BSTRACT OF THE PROJECT

1. Admin and user should be able to login into the application.
2. Admin should be able to create auction events.
3. All the auction items should be able to added/updated by the admin.
4. Admin should publish auction catalog containing items that are going to be auctioned on particular date with starting bid price. There can be only one auction per date.
5. User should be able to bid in open auction.
6. They should be able to know the current highest bidder.
7. They should be able to add their own bidding price.
8. At end of day, admin will check all bids and sell the item highest bidder.
9. Once an item sold, repeat the process with next item until all items are sold.
10. Winning bidder should get message about his win and ask for payment.
11. Admin should be able to view the report containing items sold to whom and what price.
12. Handle data and errors properly. Show appropriate messages to user.
13. Display good input, output messages and reports in proper format.
14. Security features should be implemented whereever possible. For example user passwords can be stored in encrytped format.

1.4 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

1. Data files

Create initial data in these files as comma separated fields. Each line stores one record.

- Maintain information about items in “items.txt” file. Item is identified by Item ID which is auto increment number. Purchase price of item is stored in this file along with other information.

- Maintain information about auctions in “auctions.txt” file. Item is identified by Auction ID which is auto increment number.
- Maintain information about which items are in a auction in “auctionItems.txt” file. Item in auction is identified by Auction ID + Item ID. Minimum bid price is available in this file along with other information.
- User information is stored in “users.txt” file. User is identified by username which is given by user at the time of registeration.

In addition to above master files following transaction files should be maintained by EAuction.

- Bidding information of users is stored in “bids.txt” file. Each bid is uniquely identified by Auction ID + Item ID + username. User’s bid price is stored in this file along with other information.
- Items sold information is stored in “sales.txt” file. Once item is sold to highest bidder it is stored in this file. Each record in this file is uniquely identified by Auction ID + Item ID. Sell price is stored in this file along with other information.

2. When EAuction starts it displays Following Screen -

----------Login Screen----------
1. Login as admin
2. Login as user
3. Register new user
0. Quit

Enter your option = <option>

option = 1 (Login as admin)
By default admin/admin is username/password of administrator. When this option is selected “Auction menu” will be displayed.

option = 2 (Login as user)
username and password will be asked and verified with “users.txt” file. When this option is selected “Auction Bid Menu” will be displayed.

option = 3 (Register new user)
All information about user will be asked and stored in “users.txt” file.

3. When admin logs in EAuction displays “ Auction Menu”

---------- Auction Menu----------
1. Add items
2. Add auctions
3. Display auctions
4. Sell Items
5. Send messages
0. Quit
Enter your option : <option>

option = 1 (Add items)

Items to be sold in auctions will be added using this option.
Item details will be asked and stored in file “items.txt”. Item is identified by Item ID which is auto increment number.

option = 2 (Add auctions)

Admin can create auction by giving date on which auction will be conducted. He will then select items to be sold by adding minimum sell price. This information will be stored in “auctions.txt” and “auctionItems.txt” files.
Multiple auctions can be created using this option but only one auction per date. Eauction should not allow creating more than one auction on same date.

Option = 3 (Display auctions)

Admin will enter date. All auctions scheduled on that date will be displayed in tabular format-

Auction ID__________ Date _____________

Item ID Description purchase price minimum sell price
- -

option = 4 (Sell Items)

At the end of the day, Eauction will display Itemwise bids for items in today’s bid. Username of maximum price bidder with bid price is displayed to admin. After admin approval item is sold to that user. Corresponding entry is added to “sales.txt” file.
At the end Auction, items, usernames and sell price is listed in tabular format.

option = 5 (Send messages)

After Items are sold in today’s auction Admin will send messages to users to whom items were sold. Admin will enter Auction ID then sales information for that auction will retrived from “sales.txt” file. Messages will be formed containing Aution ID, Item ID and sell price for every user to whom item is sold. These message will be stored in “message.txt” file along with username

3. When user logs in EAuction displays “ Auction Bid Menu”

------------------------- Auction Bid Menu---------------------------

1. Display today’s auction
2. Bid for items
3. Check messages
0. Quit

Enter your option : <option>

option = 1 (Display today's auction)

Today's auction will be displayed in tabular format as above.

option = 2 (Bid for items)

User will enter Item ID from today auction item list which he/she wants to bid for. After user enters <BidPrice> it will be checked against minimum bid price. All norml validations regrading item price should be done.
An entry will be added to "bids.txt" along with Auction ID, Item ID etc..

option = 3 (Check messages)

Messages regarding whether user got the item with bid price can be checked with this option. All messages for logged in user from "messages.txt" file will be displayed.

Assumptions: <Write assumptions made>

Technical Requirements -

1) C programming language
2) Use file input/output operations to read and write data.
3) Use multiple Linked Lists to read data from files at the beginning and write updated data to files before applicaton ends.
4) Use dynamic memory allocation.

Non Functional Requirements

1) Multi-file multi-directory solution is expected. Modular and maintainable code (comments) and all coding standards should be followed.
2) makefile to build application. Two-step compilation process - .o and then executable should be generated.
3) Use valgrind tool on application executable to detect memory leak. Final valgrind report to be submitted in "reports" directory.
4) Level 0 DFD (context diagram), Level 1 DFD, Flow diagram and pseudocode for 2 complex functions logic.
5) SRS in pdf format, RTM, Plan, Presentation. MOMs
6) HLD_LLD Document (optional)
7) Design review, Code review, Inspection Logs of design and code reviews
8) Unit test cases and Integration test cases in UT_IT document. Both types of test cases i.e. sunny and rainy should be present in this document

Set Up Checklist for Project

Software Requirement:

Vi Editor, ctags, splint, valgrind, gcc, make, git account

Minimum System / Hardware Requirements:

Laptop with access to internet and Linux OS