

**ONLINE AUCTION SYSTEM**

High Level Design & Low Level Design

|  |  |
| --- | --- |
| **Index** |  |
| 1. Introduction | ------------------------------------------------ 3 |
| 1.1 Intended audience | ------------------------------------------------ 3 |
| 1.2 Project purpose | ------------------------------------------------ 3 |
| 1.3 Key project objective | ------------------------------------------------ 3 |
| 1.4 Project scope and limitation | ------------------------------------------------ 3 |
| 1.5 Functional overview | ------------------------------------------------ 3 |
| 1.5.1Header files | 3 |
| 1.5.2 Functions | 4 |
| 2. Design overview | ------------------------------------------------ 5 |
| 2.1 Design objective | ------------------------------------------------ 5 |
| 2.2 Design alternative | ------------------------------------------------ 5 |
| 2.3 User interface paradigms | ------------------------------------------------ 5 |
| 2.4 Error detection/ Exceptional Handling  2.5 Memory Management | ------------------------------------------------ 5  -------------------------------------------------5 |
| 2.6 Performance | ------------------------------------------------ 6 |
| 2.7 Maintenance  2.8 Risks | ------------------------------------------------ 6  -------------------------------------------------6 |
| 3. System architecture | ------------------------------------------------ 6 |
| 3.1 Structure  3.2 Flow Chart | -------------------------------------------------6  --------------------------------------------------7 |
| 4. Environment description | -------------------------------------------------8 |
| 4.1 Time zone support | -------------------------------------------------8 |
| 4.2 Language support | -------------------------------------------------8 |
| 4.3 User desktop requirement | -------------------------------------------------8 |
| 4.4 Server-side requirement | -------------------------------------------------8 |
| 4.4.1Deployment consideration | 8 |
| 4.4.2 Application server disk space | 8 |
| 4.4.3 Database server disk space | 8 |
| 4.4.4 Integration requirements | 8 |
| 4.4.5 Network | 8 |
| 4.5 Configuration | 8 |
| 4.5.1 Operating system | 8 |
| 5. Reference | ------------------------------------------------9 |

1. **Introduction: -**

An online auction project that holds online auctions of various products on a website and serves sellers and bidders accordingly. The system is designed to allow users to set up their products for auctions and bidders to register and bid for various products available for bidding.

# Intended Audience: -

The audience set for this project would include admin, buyers, sellers who want to get the services of online auction system for buying products and selling products in online.

# Project Purpose: -

To build a use friendly auctioning website, where user will be able to auction any product which is available nearby or anywhere in the world. By using Online Auction management

system it will be easy for auctioneer to make an auction and time saving also. By making auction through this application will help to reach maximum of buyers bidding in local area. To develop an online auction system which will provide a forum for sellers to meet and interact with buyers, and sell items to interested bidders.

# Key Project Objectives: -

* Secure registration of all users including a personal profile.
* To generate reports for each completed bid in the auction system.
* To computes the sellers shill scores for each seller that sells products on the online auction system.
* The objective is to develop a user-friendly auctioning site where product can be auctioned and provide value-added services to the bidders and the sellers.

# Project scope and limitation: -

* online bids take place at any time, 24/7. In a word, a market that never sleeps.
* There are literally no geographical boundaries with online auctions.
* Items are listed and allowing buyers to research and decide properly before bidding.

# Functional Overview: -

* + 1. Following header files are included in the program:
* #include <stdio.h>
* #include <string.h>
* #include <stdlib.h>
  + 1. Following Functions are included in the program:
* Register:

1.5.2.1 Buyer registration(): If buyer is new user to online Auction system, then he registered himself as a buyer.

1.5.2.2 Seller registration(): If seller is new to online auction system, then he registered himself as a seller.

* Login:

1.5.2.3 Admin login(): Admin logins with Admin username and password.

1.5.2.4 Buyer login(): Buyer logins by entering Buyer username and password.

1.5.2.5 Seller login(): Buyer logins by entering Seller’s username and password.

* Admin:

1.5.2.6 Manage Sellers(): Admin manage Sellers by blocking the particular sellers.

1.5.2.7 Manage Buyers(): Admin manages Buyers by blocking the particular

Buyers.

* Seller:

1.5.2.8 Add products(): Seller register the products and add the products

1.5.2.9 Delete products(): Seller delete the products

1.5.2.10 Modify products(): Seller update or edit the product details

1.5.2.11 Display products(): Seller display the products list.

* Buyer:

1.5.2.12 Auction for product (): Buyer Started Bidding on the Selected product, by increasing the amount of base price of product.

1.5.2.13 Bidding history(): It display the bidding history for seller and buyer.

* Auction():

This function is interface between the Buyer and Seller, Buyer can view the bidding history of his products and seller view the history of what the products are sold.

# Design Overview: -

Online Auction System comprises of the following modules:

# Design Objectives: -

* Admin have enables users to login and registration for the selling or buying any product. Authentication is compulsory whether a buyer or seller.
* The Buyer can start the bidding whenever they want.

# Design Alternative: -

We have used file handling and linked list instead of stack & queue as Insertion and Deletions operations are fast and easier in linked list. Memory allocation is done during run-time. (i.e., no need to allocate any fixed memory.

# User Interface Paradigms: -

The Online Auction System gives an option to Seller, is given an interface to create a new registration, an option to add his products for selling. Buyer is given an interface to create a new registration, an option to buy the products by bidding at what cost he wants. A specific set of functions for buyers and sellers are given to edit details of the products and delete the products, view the products.

# 2.4 Error Detection / Exceptional Handling: -

* If the Buyer doesn’t have any pre-existing account, the Buyer has create one else it won’t perform any functions and would give “not found” or “Invalid entry”.
* While registering into online Auction system as a buyer, should first enter the name phone number, email, password, are should be validated. Phone number contains only 10 digits not more than 10, else it will display and “Invalid length” error for the respective cases. Passwords having upper and lowercase letters and special characters and one number otherwise it gives “Invalid Password” should not be validated.

**2.5 Memory Management:**

The dynamic memory allocation malloc() function is used in the application which allocates single block of requested memory during runtime.The variables which are assigned in the function occupies the size according to their datatype. While we used linked list to manage the notes so, according to the notes, the size is occupied dynamically.

# 2.6 Performance: -

The system will work on the user’s terminal. The performance shall depend upon hardware components of the online Auction System Application and the internet connection. The responsiveness of the application shall be high and the application shall behave as per the user action. The user shall be acknowledged in the form of visual changes or feedback on the application to enhance the interaction. The response time and throughput time on the application shall be minimal.

# 2.7 Maintenance: -

Very little maintenance should be required for this setup. An initial configuration will be the only system required interaction after system is put together. The only other user maintenance would be any changes to settings after setup, and any specified special cases where user settings or history need to be changed. Physical maintenance on the system’s parts may be required and would result in temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project but may result in downtime.

**2.8 Risks:**

The registration and login of sellers and buyers itself provides authentication & security. No one can see the user's credentials that are stored in a separate file. There are also risks associated with the hackers & malicious insiders & careless employees.

Becoming a victim of fraud including the misrepresentation of an item, the item not being sent to you or not being paid if you're the seller. Doing business on sites that aren't encrypted which can leave your information open to anyone. Scams by international sites that aren't secure or don't have reputable sellers.

# 3.SYSTEM ARCHITECTURE: -

**3.1 Structure Details:**

The system consists of Four structures:

* Buyer

This structure contains all the declaration of all the variables that are present in the Buyer sub menu.

The Buyer id is integer datatype, email, name, pan number and password are character datatype

* Seller

This structure contains all the declaration of all the variables that are present in the Seller

sub menu.

The seller id is integer datatype, email, name, pan number and password are character datatype

|  |
| --- |
|  |

* Product:

This structure contains all the declaration of all the variables that are present in the Seller

sub menu.

The product id, product price is integer datatype, product name, product details are character datatype.

* Admin:

This Structure contains all the declaration of all the variables that are present in the Admin submenu. Admin username and password both are character datatype.

Diagram

Description automatically generated

**Entity relationship diagram of online Auction System**

* 1. **Flow Chart:**

Diagram

Description automatically generated

**4.Environment Description:**

* 1. **Time Zone Support: -** It will support time zones as per Indian standard time(IST) in (GMT +5:30) and UST standard.
  2. **Language Support: -** English

# User Desktop Requirements: -

* + 64-bit processor, 1 GHz or faster
  + At least 10 GB free hard drive space
  + At least 1 GB RAM **Server**

# Server-Side Requirements: -

* + 64-bit processor, 1 GHz or faster
  + At least 2 GB free hard drive space
  + At least 1GB RAM
    1. **Deployment Considerations**: -
  + Local storage is used
  + No network latency to consider
  + To scale buy a bigger CPU, more memory, larger hard drive, or additional hardware
    1. **Application Server Disk Space:** -

Disk Space - Less space is required. Local Operating System is required and three txt files to store the records of processes.

* + 1. **Database Server Disk Space: -**

No such disk space is required as the program is fully functional on online IDE(s) as well. Local Operating System is required and two txt file to store the records of processes.

* + 1. **Integration Requirements**: -
* Language: - C
* Tools: -Valgrind, Splint, gcov, gprof, cunit, Makefile
* Complier: - gcc
* Linux Environment -ubuntu
* Git hub

# Configuration: -

**4.5.1**. Operating System: - Linux environment

# 5.Reference: -

The references are:

* https://[www.programiz.com/dsa/linked-list](http://www.programiz.com/dsa/linked-list)
* https://[www.javatpoint.com/file-handling-in-c](http://www.javatpoint.com/file-handling-in-c)
* [Online Auction System - GeeksforGeeks](https://www.geeksforgeeks.org/traceroute-in-network-layer/)
* <https://www.geeksforgeeks.org/bidding-process-in-e-commerce/>
* Git hub