

Logos Christian Academy

Website Documentation

Authored by: Emily Clyde, Robert Callahan, Ryan Recinos

Table of Contents

1. Introduction 2-4
2. Requirements Description 4-5
3. System Overview 5
4. System Architecture 5-7
5. Data Design/Diagrams 8-12
6. Human Interface Design 12-19
7. Clearing the Database 19-20

# Introduction

The purpose of this website is to develop a resource for the members of the Logos Christian Academy to use during their annual school auction.

This website will assist the members in helping them to gather, coordinate and report auction information. This includes the organization of items before and after auctions.

The target audience to be defined further in the Scope:

***Web User***

***Admin User/Auction Coordinator***

# Scope

The website will be a single platform offering services that will transform the Logos Christian Academy Yearly Auction Event into a convenient online resource for users. The website will offer the following valuable tools and benefits for Logos Christian Academy:

* The ***Web User*** will be able to view auction items and descriptions before the auction. They will be able to purchase tickets in advance.
* The ***Admin User/Auction Coordinator*** will include school staff and volunteers. They will be able to load auction donor’s names into the system. They will be able to enter auction items into the system before the auction event. They will be able to generate reports with the donor information to send out appropriate tax documentation to donors, as well as thank-you letters. The Logos Auction Coordinator who is physically present at the auction will be able to load bidders and sales during the live auction, silent auction, and dessert auction.

By meeting the specific needs of the expected users this site will be a powerful tool to help facilitate organization and information for the Logos Christian Academy Yearly Auction Event.

# Overview

Below is an overview of what will be include in this document (Sommerville, 2011):

* Requirements Description:
  + This will include the user requirements with the corresponding system requirements. The user requirements will describe what the user will be capable of, and the system requirements will describe what the system must do to satisfy the user requirements.
* System Overview
  + This section will include any product requirements, organizational requirements, and any other external requirements for the system.
* System Architecture
  + This section will analyze the needs of the system requirements. The system at this point in time will have a “transaction processing application” (or TP). The application is database centered. The application will process user request and update the information in the database. The system will have layered information system architecture.
* Data Design
  + This section will include the model of the system. UML diagrams and other graphical models will be used in a detailed description.
* Human Interface Design
  + This section will describe the human interface design of the application. This will include an overall visual design overview system for the user. The application will create a consistent experience for the user. Elements that will be discussed will range from icons, buttons, dialog boxes and web page design.

# Reference Material

Sommerville, I. (2011). *Software Engineering* (9th ed.). Boston, MA: Addison Wesley.

# Definitions/Acronyms

**Logos Christian Academy**: *the client*

**Web User**: *user who can only view the website from a browser*

**Admin User**: *user who can update, create, delete information from the website*

**Auction Coordinator**: *user who can update, create delete, print information from the website*

**Functional Requirements**: *a function that is implemented in a system*

**Non-Functional Requirements**: *a function that refers to a constraint or behavior in a system*

**CRUD:** *Create, Read, Update, Delete*

**Bidder**: *Auction participant who will use their assigned Bidder Number to bid on auction items. This may be an individual, couple or family*.

**Auction Item***: Items or services that will be available to be bid on both the live auction, or the silent auction.*

**Extra Auction Item**: *An item such as a raffle, cash donation or dessert dash that all bidders have an opportunity to give towards.*

**Donor***: Person, family or business that has donated items or services to be Auction off.*

**Load Winning Bids**: *Enter the winning bid amounts after a particular auction has closed.*

**Auction Info**: *Update the auction details shown on the Auction Home page. Update the Contact information shown at the bottom of certain page. Update information pertaining to ticket sales, for use in Auction Reports*.

# Requirements Description

The following are the User Requirements and the corresponding System Requirements descriptions.

***User Requirement Definition 1***

The system shall generate a list of all of the current Live Auction Items for the user to view.  The user will be able to click on the items to view details.

***System Requirements Specification***

* The database shall store the Auction Items with a designated type.
* The system shall return a view with a list that is generated by a query containing each item that has the type of Live Auction.
* Each item name will be a hyperlink to the details view of that item

***User Requirement Definition 2***

The system shall allow authorization for 3 different roles: General User, Admin User, and Auction Coordinator.

***System Requirements Specification***

* The system shall have a secure roles and authentication system.
* The system shall allow for each role to have their own unique permissions.

***User Requirement Definition 3***

The system shall allow the Admin User and Auction Coordinator to input and edit Auction Details, Auction Items, Bidders, and Donors into the database.

***System Requirements Specification***

* The system shall allow for CRUD operations from the User Interface to create, update, or delete records from the entities containing the Auction Details, Auction Items, Bidders and Donors.

***User Requirement Definition 4***

* During the live auction, multiple Auction Coordinators will be simultaneously inputting bidders and updating items with the final sale amount.

***System Requirements Specification***

* The system will allow for multiple users with the role of Auction Coordinator to be logged in at the same time.
* A form will be displayed on the UI that will allow for CRUD operations to be performed on the database to update Bidder and Item information simultaneously.

***User Requirement Definition 5***

* The system shall generate reports as needed by the User for the following: Donor List, Bidder List, Complete Auction List, Live Auction Items, Silent Auction Items, Raise the Paddle.

***System Requirements Specification***

* The system shall return a query based on the User’s input from the UI.
* The query data will be displayed in a view and styled by HTML and CSS to become the report.

# System Overview

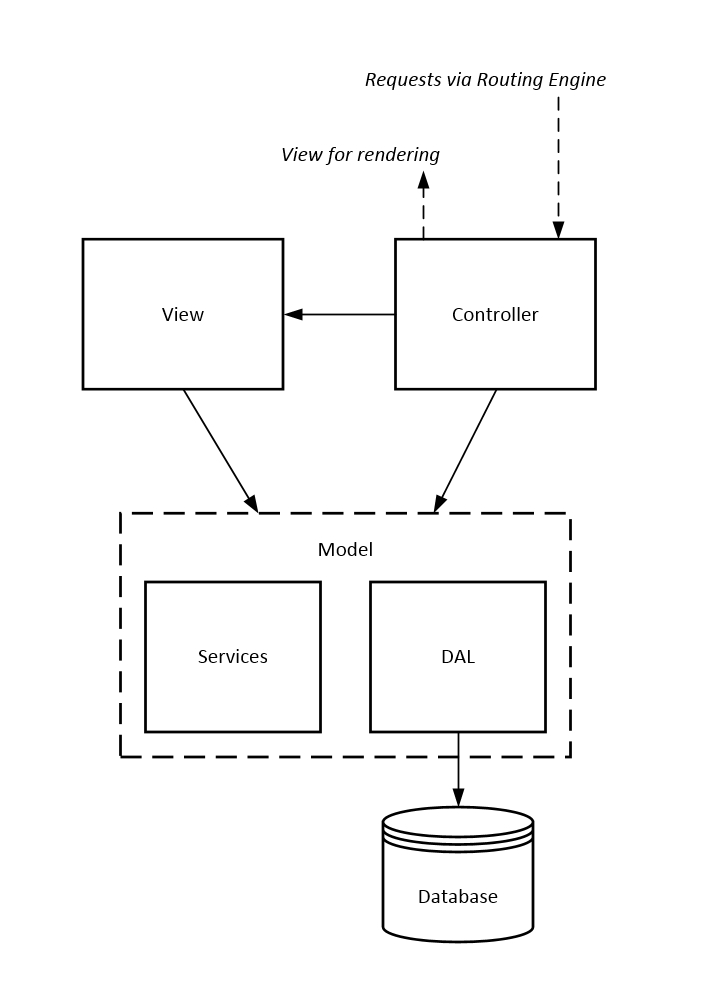
***System Functionality***

* The (system) Auction Website will allow functionality for the Auction Coordinator and Admin User to enter auction data into the database and generate forms and reports as needed.
* The primary functionality will be for the Auction Coordinator to load Auction Bidders into the system during the auction, and update the Auction Items with a final sale amount.
* The Admin User and Auction Coordinator will be able to input Auction Items, Bidders, and Donors into the database.
* The General Users will be able to view the system website to view a full list of Live Auction Items.
* The system will generate reports and forms as needed by the user. The reports will include Donor List, Bidder List, Complete Auction List, Live Auction Items, Silent Auction Items, Raise the Paddle. The reports will be generated by querying the database and styled in a view with HTML and CSS.

***System Design***

* The system will be web-based using ASP.NET with an MVC architecture. The system will be heavily data driven. The database will be created using code-first with Entity-Framework.

# System Architecture

The Auction System will be built using the ASP.NET MVC Framework. Program functionality will be divided into three distinct subsystems: The Model, View, and Controller. The model represents the system domain, the view handles the rendering of the user interface, and the controller controls the flow of the application. 

***Controller***

Controllers handle user interaction as well as dictating the application flow. Automatically invoked by the ASP.NET Routing Engine, a controller interacts with both the model and view layers. The controller requests the model to perform business logic and CRUD operations against the database. It also will select the view which will be responsible for rendering the UI. Controllers usually retrieve data sent from the model so that they may pass it along to the view.

***View***

Views are the templates responsible for generating the UI. Controllers send data to the view, which determines how the data should be displayed. Views interact only with data models so that they may anticipate what data they will receive. Views do not perform any application action themselves. Instead, they rely on controllers to invoke and use them.

***Model***

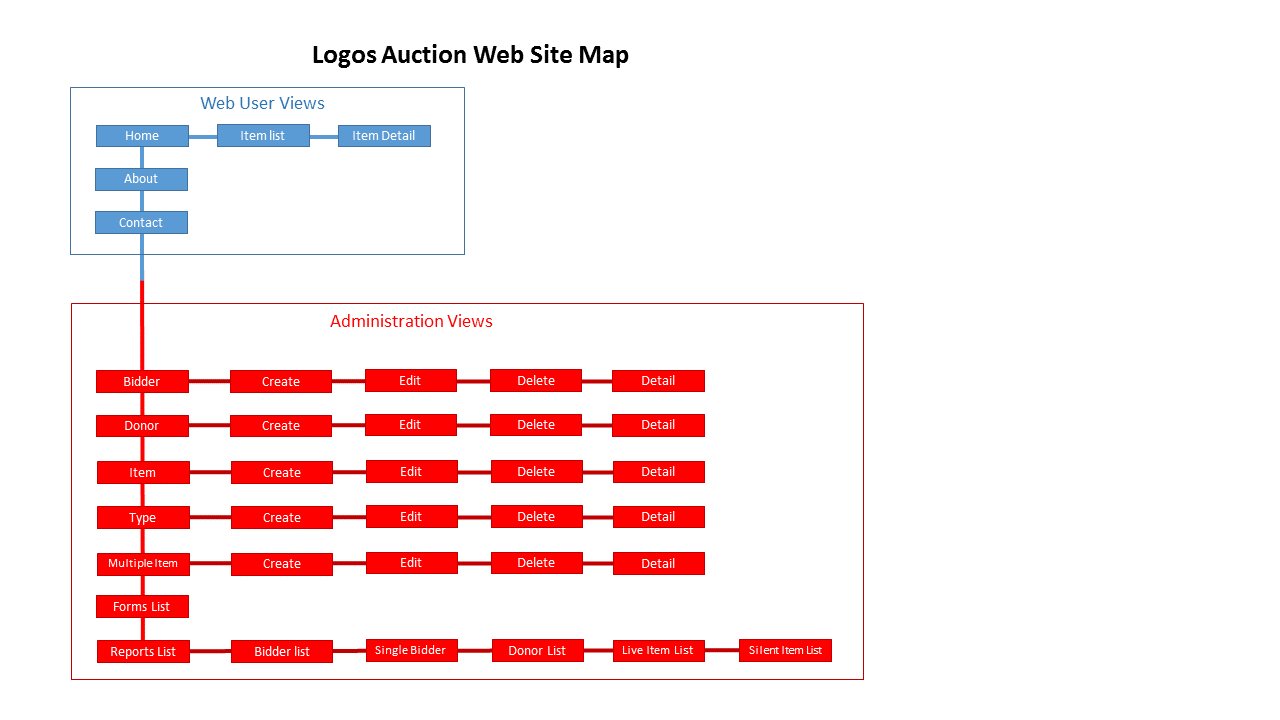
The model contains all data persistence and domain logic. As the Auction system is mostly concerned with data persistence, there will be little business logic in the model. Input validation logic will exist within the model and utilized by the controller wherever necessary. The data access layer (DAL) will be split into a few layers including a unit of work and repositories. These extra layers will be loosely coupled and interchangeable to allow for system flexibility and testing. Object Relational Mapping via Entity Framework (EF) will allow the DAL to remain relatively thin. The EF automatically converts and stores data from domain data objects to a relational database.

***Framework***

The Auction system will use the ASP.NET MVC Framework to take advantage of its inherent and implied separation of concerns (SOC). While ASP.NET Web Forms were an option that would provide quick implementation of basic features, Web Forms lacks qualities such as an implied SOC and the ability to easily test various parts of the application. The Auction application is data driven with little need for state preservation outside of data persistence. MVC embraces the statelessness of the web but Web Forms attempts to preserve it. The extra state preservation is unnecessary for the Auction app and can cause confusion amongst developers. Also, given the inherent SOC of the different components in MVC, it is easier for development duties to be split between team members so that they may work on different components of the system at the same time.

# Data Design/Diagrams

**SITE MAP**

****



# **Data Dictionary**

# 

# **UML DIAGRAM**

# 

# **USE CASE DIAGRAM**

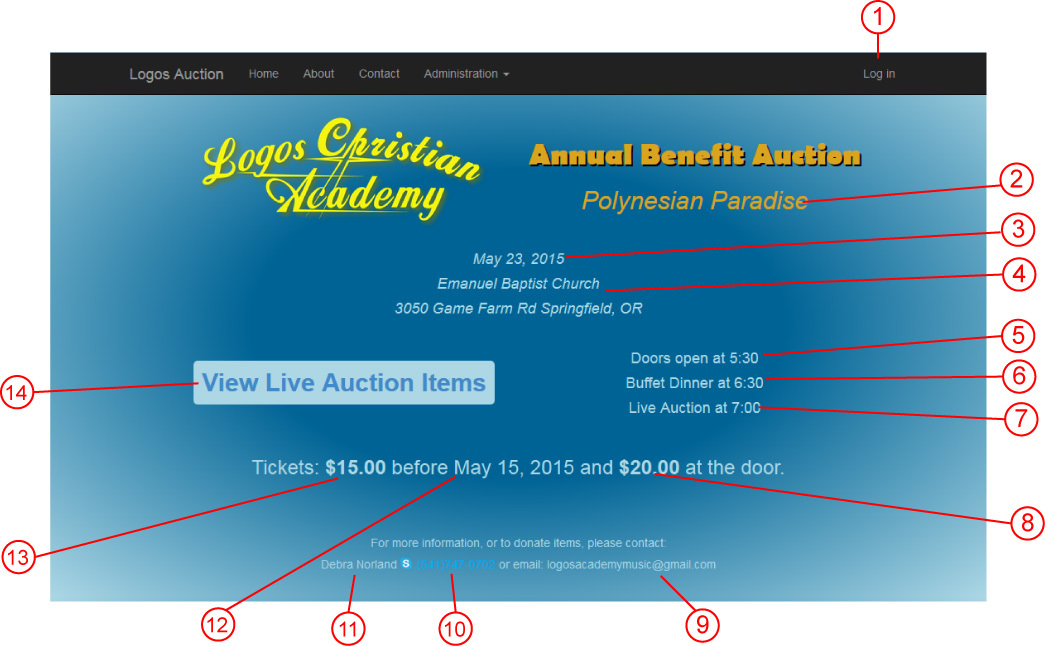
# 

# Human Interface Design

***Web User*:** The Web User will click on an auction link on the Logos Christian Academy web site which will redirect them to the Logos Auction Website. The Web User will be able to view the auction homepage. The Auction Home page will display details about the auction. These details will include: *Date, Time, Location, Cost, Theme, and Contact Information*. The Home page will have a button that will display a current list of the Live-auction items. The Web User can select an item from the list to display a view that show details about the selected item including a picture when appropriate.

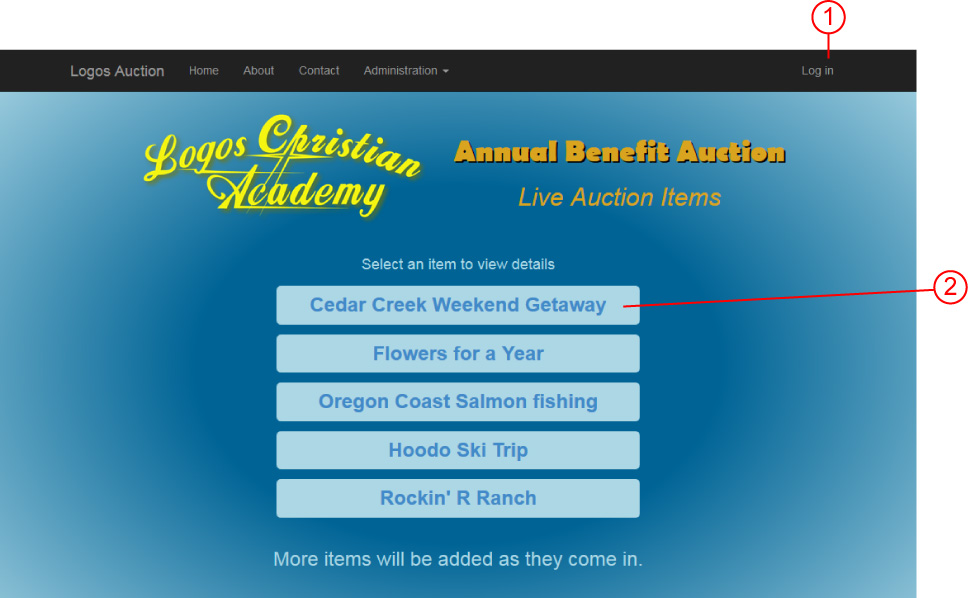
***Admin User/Auction Coordinator*:** The Admin User will access the Auction Web site through the Home page view. The Admin User will be verified through a standard login view. Once logged in the Admin User will have access to a menu where they will be able to perform CRUD procedures on, *Auction details, Auction Items, Donors, and Bidders*, through appropriate views. They will also be able to select from a list of reports. The view page for each report will give the Admin User the ability to print the report.

The Auction Coordinator will access the website and have the same permissions as the Admin User. They will access to a menu where they will have the ability to perform C.R.U.D. procedures on, *Auction details, Auction Items, Donors, and Bidders*. They will primarily using the Add Bidder page to load in bidders as they arrive at the auction, the Update Item page to enter final bid amounts when each portion of the auction closes, and print individual bidder reports showing all Items purchased with the dollar amounts and total cost amount. The Auction Coordinator will also have access to the list of reports.

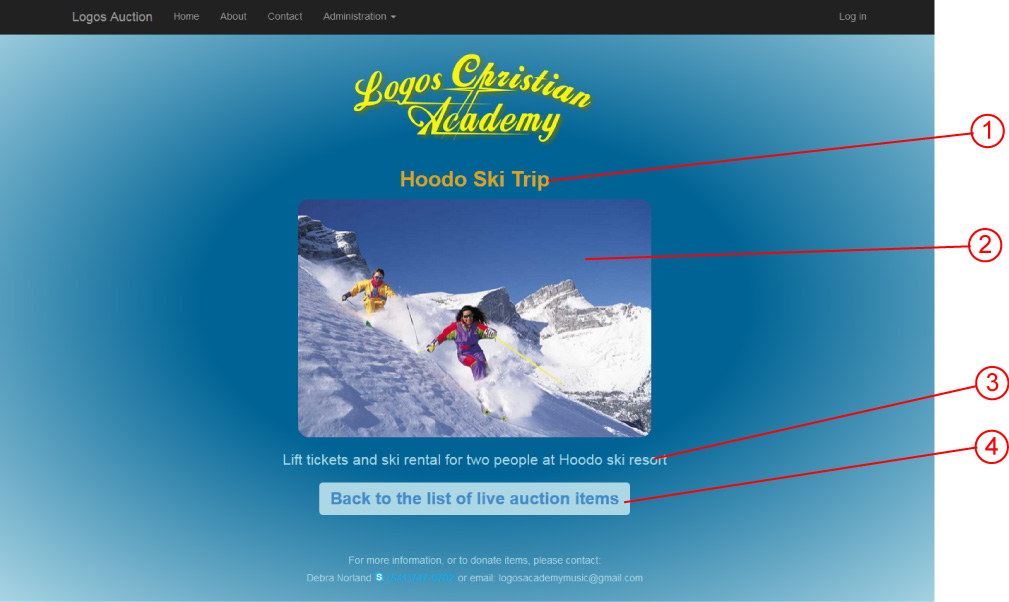


***Home Page View*:**

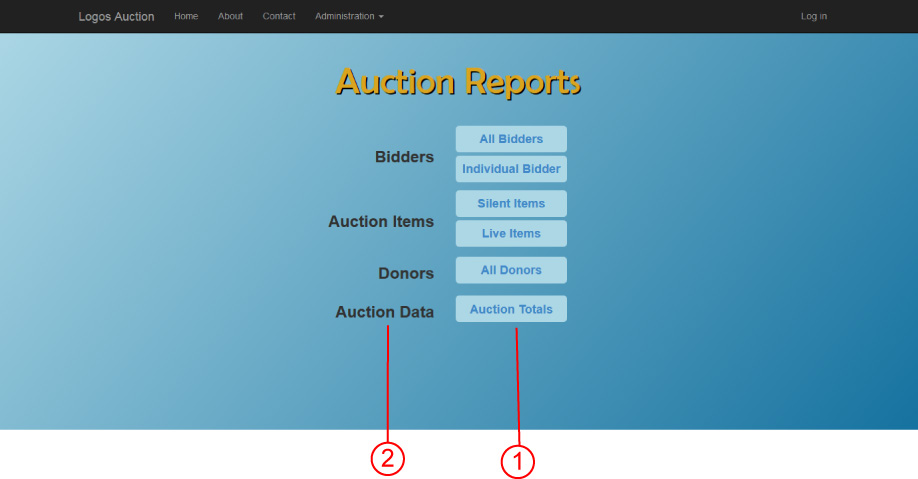
|  |  |
| --- | --- |
| 1 | Navbar link to Login view for Admin Users |
| 2 | Theme text from *Description* entity model |
| 3 | Date text from *Description* entity model |
| 4 | Location text from *Description* entity model |
| 5 | Time1 text from *Description* entity model |
| 6 | Time2 text from *Description* entity model |
| 7 | Time3 text from *Description* entity model |
| 8 | DoorCost from *Description* entity model |
| 9 | Contact email from *Contact* entity model |
| 10 | Contact phone number from *Contact* entity model |
| 11 | Contact Name from *Contact* entity model |
| 12 | EarlyDate from *Description* entity model |
| 13 | EarlyCost from *Description* entity model |
| 14 | Button link to View Index list of Live Auction Items |

***Auction Items List View*:**

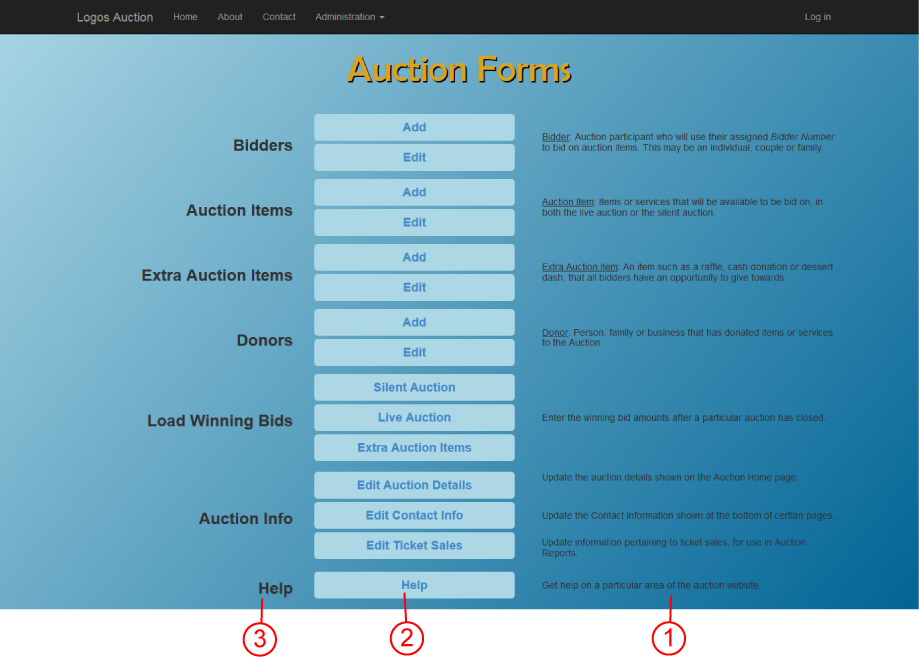
|  |  |
| --- | --- |
| 1 | Navbar link to Login view for Admin Users and Auction Coordinator Users |
| 2 | List of Live Auction Items from *Items* entity model |

***Auction Item Detail View*:**

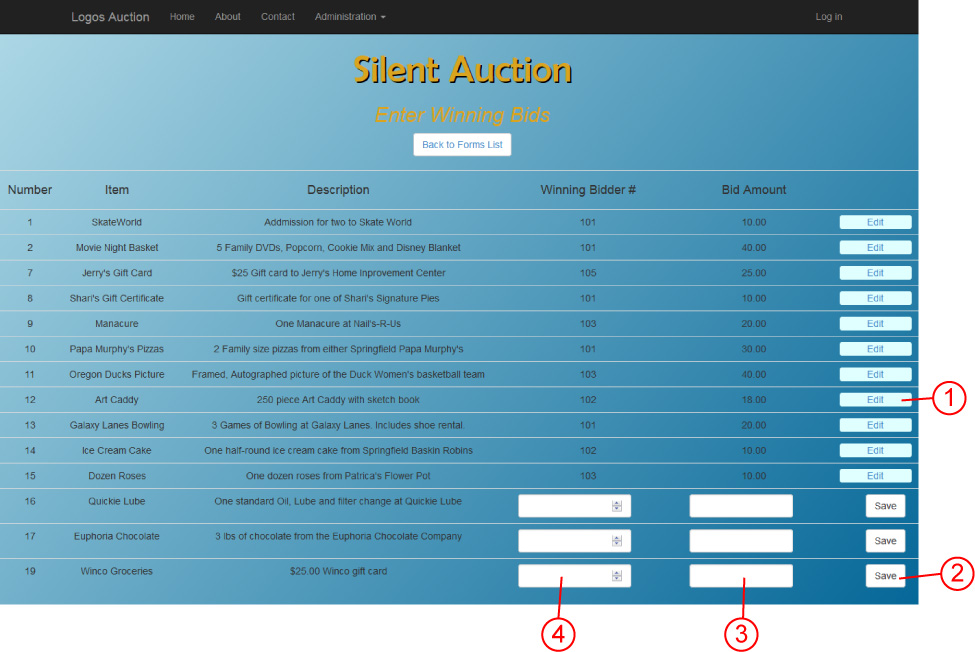
|  |  |
| --- | --- |
| 1 | Item Title text from *Items* entity model |
| 2 | Optional Item Image. ImageURL from *Items* entity model |
| 3 | Item Description from *Items* entity model |
| 4 | Button link back to Items List View |

***Reports List View:***

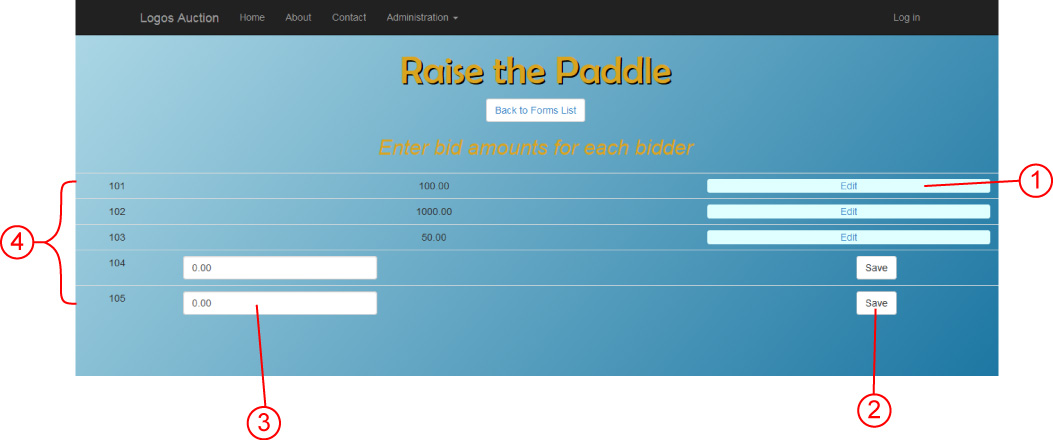
|  |  |
| --- | --- |
| 1 | Button links to individual forms grouped into category |
| 2 | Reports category name |

***Forms List View:***

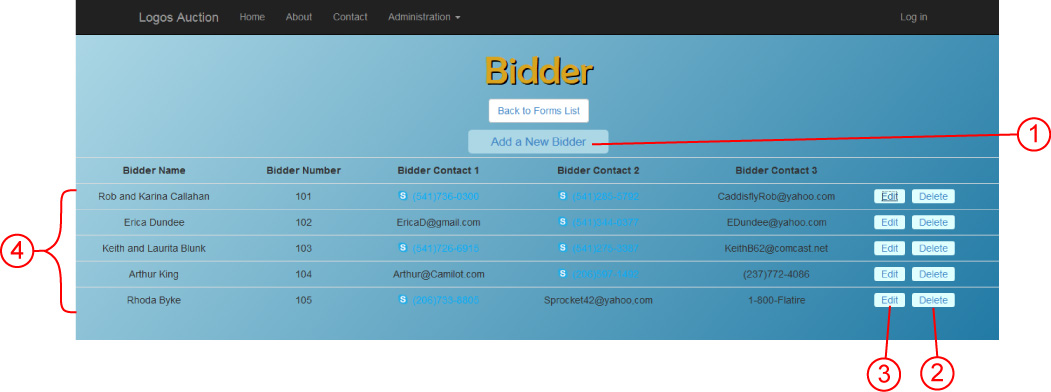
|  |  |
| --- | --- |
| 1 | Quick reference description for each category of forms |
| 2 | Button links to individual forms grouped into category |
| 3 | Forms category names |

***Load Auction Bids View (same for Live Auction)***: 

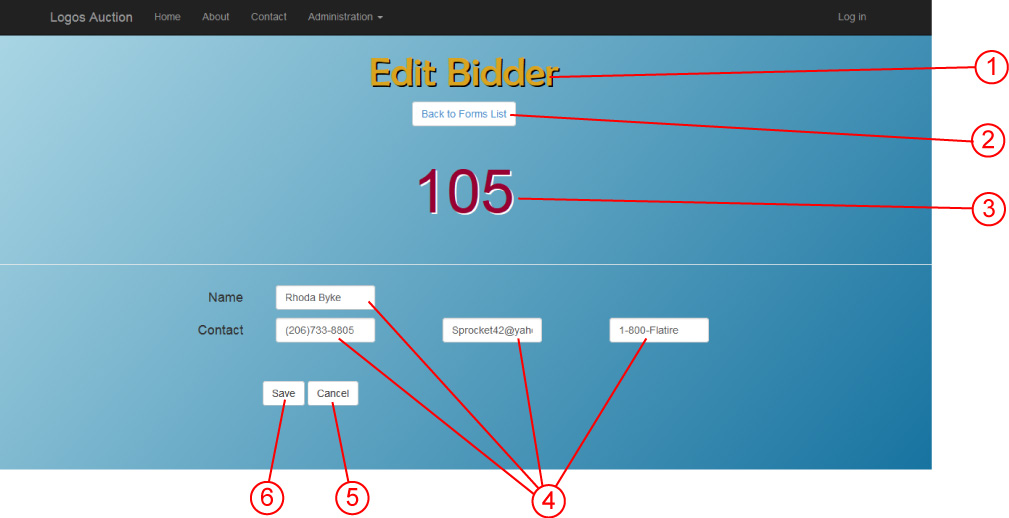
|  |  |
| --- | --- |
| 1 | Edit Button used to change bid information |
| 2 | Save Button saves data loaded into that respective row |
| 3 | Form Text box to enter the amount bid |
| 4 | Form Text box to enter the bidder number |

***Load Extra Auction Bids***:

|  |  |
| --- | --- |
| 1 | Edit Button for bids erroneously entered |
| 2 | Save button for each bid amount entered |
| 3 | Text Box to enter bid amount |
| 4 | List of bidder numbers |

***List Page for C.R.U.D (similar for all Items)***

|  |  |
| --- | --- |
| 1 | Button to add a new Item |
| 2 | Delete button to delete a particular item |
| 3 | Edit Button to edit a particular item |
| 4 | List of items and some details |

***Detail Page for C.R.U.D (similar for all Items, both Edit and Add new)*** 

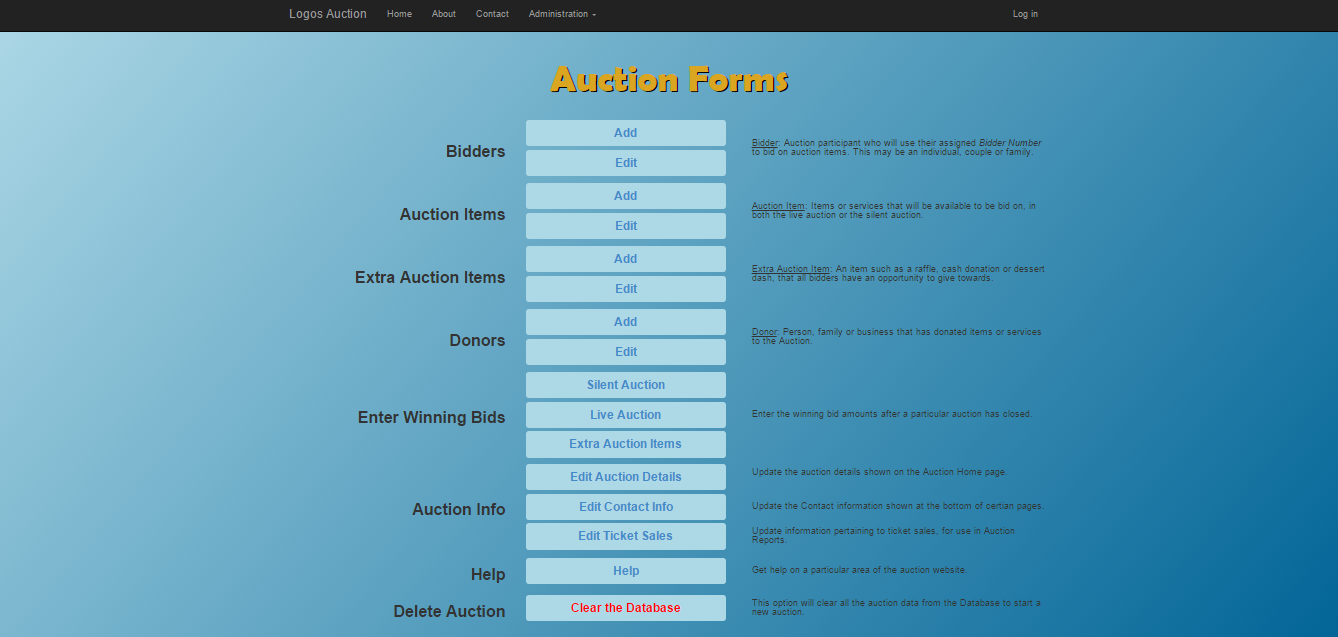
|  |  |
| --- | --- |
| 1 | Title |
| 2 | Back to Forms List nav Button |
| 3 | Name of individual item to edit (will not be present on Add New page) |
| 4 | Text Boxes to add information |
| 5 | Save Button to save changes |
| 6 | Cancel Button returns back to Forms list without saving changes |

# Clearing the Database

Deleting the data is a necessary function to the Auction system, this will clear the database and get it ready for the next Auction Event.

***IMPORTANT: The deletion of data should only be done after the Administrators of the completed auction have printed out ALL reports for their records. Once the database is cleared there is no going back to retrieve any data from the auction.*** ***This is a permanent delete of all Auction Bidders, Donors, and Items!!***

***Clearing the Database View from the Auction Forms Page:***



***Clearing the Database View from the Auction Delete Page:***

