CAS Digital Auction Website

Dive Into Denver’s Largest Exotic Fish Auction
Problem

Hybrid-Model Approach is Not Efficient for Conducting the Auction

CAS conducts in-person routine auctions on a monthly basis and 2 large bi-annual exotic fish live auctions.

They experience difficulty conducting the auction in real-time and are forced to run part digitally and part-in person.
# Initial Documentation

<table>
<thead>
<tr>
<th>Requirements Doc</th>
<th>Technical Specification</th>
<th>Design Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Full understanding of scope, needed functionality, deliverables</td>
<td>● Analysis of considered technology paths and respective pros/cons</td>
<td>● Implementation design structure for the website. Requires sitemap to interconnect different components of site.</td>
</tr>
<tr>
<td>● Extensive documentation regarding requirements from client</td>
<td>● Narrowed down to 2 choices</td>
<td>● Intuitive layout for both staff/users</td>
</tr>
</tbody>
</table>
2.2 Product Functions

1. Sign-in: Users and Staff members can sign-in confidently with our authentication
2. Registration: New users can register to become member
3. Members can register for upcoming auctions and upload their items for sale
4. Store centralized database for all auctions, items within auctions, and participants.
5. Staff can create new auctions, update items after auction is finished, and make them dynamically assign to user dashboard for checkout
6. 3rd Party Payment API
7. Staff will have modifiable access to all DB touchpoints (Auctions, Users, Items, etc)

- Iteratively concluded on necessary functionality and baseline requirements
- Real-Time assignment of item to person in for auction
  - Unique dashboards for User/Staff
  - Checkout
- Confirmation from client-end on requirements so we are held accountable
Technology Specification

- Considered JSP + Java Servlet and tried initial implementation.
  - Chose to switch because of testing issues

- Decided that Angular, Mongo, Express, Node.js stack is best option for implementation
  - Caught up on equal progress within one week of switching tech

- **Pros of Mean**: Dynamic, Modern, Maintainable, and Functionally the correct fit for the scope of the project
1st Semester Timeline

**Documentation, Initial Iterations**
  - Started with JSP/Java Servlet

**MEAN Stack Implementation**
- Made significant progress with MEAN. We all spent a lot of time learning the tech and worked hard over break to build homepage, dashboards, functionality for each and routing/linking components in angular.
  - We caught up within 1 week of switching.

**First Build Attempt**
- Began development in JSP/Java Servlet, but had issues with testing. Tried to troubleshoot and solve problem. Decided to switch to MEAN stack.

**Adding more functionality, Enhancing design**
- Spent time on building more key features and iterating of client feedback. Cleaning CSS, preparing all documentation and presentation. Completed partially functional prototype.
Implementation
Implementation

- Building out initial pages and respective components, features, and assets on the page
- Sign Up/Sign In
  - Staff/User Dashboards
  - Item Object Creation
- MVC design pattern
Database Design

- We are using MongoDB
- 5 collections
  - Auctions
  - Items
  - Users
  - Image.chunks
  - images.item
- MongoDB connected with Node.js (thru mongoose)
- We use postman for testing
Design: SiteMap UI Navigation

- **HomePage**
  - Authenticated Registration
  - Sign In/Sign Up
  - Auction Schedule
  - Education on Website
  - Sign In/Up will identify users and staff (different DB)
  - UserID & StaffID go to unique portals
  - User/Staff Dashboards are separate and custom

- **Staff Dashboard**
  - Create/Manage Auctions
  - Quick Sale Functionality
  - View All Items/Participants
  - Quick Sale: Enter Name & Price, Assign to UserID,
  - Staff updates the queue in real-time to reflect winner,
  - final price, and seller

- **User Dashboard**
  - User can register for upcoming auctions and add item(s) pre-auction for sale
  - My Cart: User is shown list of items they won
  - Checkout Functionality
  - Items are assigned by Staff using UserID during
  - Payment Handled using Paypal API integration
  - Manage Profile, View Previous Won Items

- **Auction Queue**
  - Stores items that sellers upload before auction
  - Updated by Staff during Auction
  - Staff can modify auctions, participants, and items
  - UserID (seller & winner) & Item will be assigned item
  - This queue is structured within each Auction instance of the DB
  - User Dashboard will update to reflect the auction results after staff updates data.

- **Auction DB Structure**
  - Auctions store UserID for participants and ItemIDs
  - based on the user who uploaded the item
  - Staff has modifiable access

- **Updated Auction Queue**
  - During Auction, Staff will update the auction items in real-time
  - New parameters: Item/ItemID + userID, final bid $, sellerID
  - Staff will publish updates and dynamically after auction completes, users can checkout
  - User Dashboard should pull this in when assigning for checkout
  - Adds functionality to assign winning bid to userID
Challenges deep-dive

Technology Choice

Java + JSP vs. MEAN Stack
Java/JSP was our initial choice however presented several issues when debugging and troubleshooting. We realized that MEAN stack offered a better end-solution for the client and switched technology.

Catch Up Phase

Learning Angular
Though angular was new to us all, we were able to recover our progress in MEAN within one week of switching. We faced more technical challenges ahead but knew this was the right choice.

Recent Challenges

Angular Components/Routing
Our divide and conquer approach allowed us to recover progress, however combining our work together was a brief challenge. Angular routing of components required problem solving in a short window of time.
Completed Functionality

Home Page:
- Authenticated Registration, Sign-In,
  -Educational Homepage Design

User Dashboard:
- Sign Up for Auction, Real-Time
Auction updating, My Cart functionality
with checkout, Manage Profile, View
previous items, Upload Items to Auction,
View other Participants

Staff Dashboard:
- All user functionality, Create Auction
+ Scheduling functionality, Admin access
to all auctions and their participants &
items

Next Steps (Over Break)

Home Page:
- CSS on Side Bar Menu Nav

User Dashboard:
- Clean Up Buttons, Layout, Add Paypal
Link into Checkout Button of “My Cart”, Tie
Mongo to Cart with Items

Staff Dashboard:
- Add “Quick Sale” Form, Clean the CSS on
Create Auction and Manage Auctions, Define
Order Function

Link MongoDB With Items/Cart Component
2nd Semester Focused Objectives

- Polishing Design with Client
- Conducting Trial-Run with Client
- Deploying with Host(s) to Domain
- Testing on Live Site
- Iterating on Feedback from client and Board
- Improving CSS and mobile layout
Visiting the CAS Live Auction in person was a fundamentally necessary step to truly understand the mechanics of the auction meetings.

We used this meeting as direct insight to build out the “Define Order” Functionality to make queueing the auction for staff members simple and easy to use in real-time with Auctions.
Testing phases in Semester 2

Testing Round 1

Feedback and Clean-Up
- Change button from Not Participate → Stop Participating
- Fix Upload Image
- Update Homepage
- Add address field in sign up and in edit Profile
- Add field to edit email in Profile
- Change Edit Button → Save
- Link Cart & Order to redirect
- Add testing for payment and checkout for User (provide staff logins)
- Change Button Alias → “3 letter Seller code” (for user view)
- Ask Alex about Donation (Staff adds Item for user donation)
- Recreate Bootstrap Homepage → New css, no external libs
- Item List & Auction List UI
- Remove Watch Picture in the Checkout Page

Testing Round 2

Functionality Re-testing
Testing essential components including define order, upload image, optimizing page load speed, adding new bootstrapped homepage cleaning UI, Paypal payment gateway, and creating useable auction records

Testing Round 3

Handoff to Client and Training
A major priority is to ensure client has full clarity on using website, creating thorough documentation, videos for easy understanding, training CAS staff on how to use the site in detail, optimizing performance and transferring assets for future use.
2nd Semester Timeline

Successful Deployment

Heroku Deployment was a key piece in bringing the site to life. After completing most requirements we began to prepare for Testing phase.

Testing Phase + ReVisiting Functionality

After visiting CAS auction in person, we realized we needed to add functionality to allow for dynamic item queuing into each auction. “Define Order” was created for this purpose. Began creating testing documentation and forms for our testers.

3 Round of Intensive Testing

March we put our website through many test rounds to perfect and optimize the site and fix bugs iteratively. We learned a lot about user-centric design and also gathered key data to measure load balancing and performance.

Perfecting Design and Functionality

After extensive testing, we addressed our feedback and incorporated new features and design elements for optimal use. Spend time on performance fixes, adding in a better payment system for users, and also details regarding auction records for staff. Also spent time training our client and in the technology transfer process.
User Dashboard

- My cart, Previous Orders, Manage Profile, Secure Checkout
- Register and add items to specific auctions/
  Purchase Items won at auction
- View other participants in auction
- Won Items assigned and updated in real-time

Staff Dashboard

- Custom Staff functionality
- Create Auction, Manage Participants, Manage Auction Items, View Auction Records, Queuing Auction Items, Checkout Users and Real-Time Auction Projection
Real Time Auction Handling

Staff can manage the live auction queue, auction items, auction participants. Custom tailored components were created for CAS staff to easily and intuitively navigate the auction process with minimal downtime.

Live Auction View specifically built to showcase items for CAS auctions with checkout and transaction handling dynamically built-in.
User & Staff Profile Functionality

← Staff can manage all users in the website from a single page and update/edit user info.

Users can directly edit their personal profiles and edit their information with ease and update at anytime.

← Staff can easily add new users and new staff members in one simple form from their Staff Dashboards.
Auction & Quick Sale Secure Transaction Integration

Staff members can checkout users using “Checkout” tab in Admin Panel. This allows for cash transaction and in-person payments.

Members of the Club can easily transact securely from their own accounts after an auction or quick sale is conducted. Using PayPal API, we have offered users the choice to checkout using their paypal or credit/debit cards. This reduces the load on Staff members because most people can now easily pay on their mobile devices.
Video Walk Through