System Analysis

*Auction System*

1. **An Analysis of our System**

To analyze the functionality of our system, we will compare our original project requirements with the current state of our auction platform.

1. **Users and their Goals**

The users goals differ between the normal users and admins. Below are their separate goals.

**2.1** - *Administrator* - An admin of the system has a primary goal of earning as much money from each auction as possible. To meet this goal the application must have a nice user interface and experience to make the auction enjoyable for users. Administrators need the ability to take control of auctions and ensure they are executed in the best way possible. For examples of implementation see docs/UMLdiagrams.png. These diagrams show how the administrator will use categories and descriptions to promote items and ensure things sell for the highest price possible.

Because we built our system with user interface design in mind, our system satisfies the need for an attractive user interface. Along the way, we updated several parts of our design (profile page, explore page, signup/login pages) to keep with this goal.

**2.2** - *Normal Users* - Normal users have the primary goal of quick and easy access to purchasing items. They desire to get the items for the lowest price possible. As seen in docs/UMLdiagrams.png, the users will be able to look through the categories to find items they desire. This will promote the bidding process and give all items more exposure. This will allow users to quickly and easily access items.

We developed a two-way system for our user to quickly and easily access items up for purchase. The profile page gives the user immediate access to all items the user is bidding on. The explore page shows all auctions, categories, and items the user has to choose between. As such, we satisfy the requirement for the user to easily navigate auctions and to quickly understand the organization structure provided.

1. **Functional Requirements**

**3.1** - *Admin Capabilities* - The system must allow an administrator/s to add, subtract, and audit auctions. This is necessary to facilitate an auction and ensure that that the auction runs smoothly.

We used Django's built-in admin capabilities. This satisfies the need for the admin to add, subtract, and audit auctions. As part of the "auditing" requirement, our system aims to provide the admin a way to hide, edit, and assign items and auctions. As such, the admin has full control over the system.

**3.1.1** - *Add Items to an auction*. This includes, but is not limited to: setting start prices, time limits, adding photos, and setting a minimum bid price. This will support manual data entry through the system provided.

**3.1.2** - *Remove items from an auction*. Removing an item from an auction may be necessary when duplicates are found or other problems arise. This removal process will permanently delete the item from the database.

**3.1.3** - *Hide items from an auction*. There may be times that an item may be in a process of information editing. This will necessitate the ability to show or hide it from the main auction.

**3.1.4** - *Show items*. This effectively starts the auction. People are allowed to bid on the items the moment that it is shown.

**3.1.5** - *Edit items.* If there is a mistake in the original data entry or more information is acquired, a item may need to be edited.

**3.1.5.1** - *Add pictures*. This is the process of adding images to an item.

**3.1.5.2** - *Remove pictures*. The application must give the admin an easy way to remove a picture.

**3.1.5.3** - *Change item description*. The item description may be changed.

**3.1.5.4** - *Change starting bid*. The starting bid may only be changed if the item has not had a bid placed upon it. In the case that there is a previous bid, the administrator must delete the current item and make a new one in order to change a starting price. If there has been no previous bid the admin is allowed to change the start price.

**3.1.6** - *Declare winner.* The system will already declare a winner, but the administrator will have the ability to change the winner to a different bidder if the winner cannot pay or other circumstances arise. This can also happen when an auction item is closed before the advertised time.

**3.1.7** - *See user profiles*. The administrator must have the ability to view user information including but not limited to name, bidding number, items won, total spending, and all bids.

**3.1.8** *End an auction*. There may be times when the time and date entered at starting the auction is incorrect. This may be adjusted or the auctioneer may call the auction at any time and award the highest bidder the item.

**3.1.9** *Remote registration*. The administrator must be able to add user profiles and change them.

**3.2** - *User Capabilities.* The user needs to be able to interact with the application in order for an auction to happen. These necessary actions include being able to bid, see the success of their bids, and retrieve their won items.

The user has the ability to bid on items for any auction they are a part of. As stated earlier, our profile page gives instant access to currently bidding items. The explore page gives the user access to all other items they may want to bid on, as well as the current status of those items.

**3.2.1** - *Bidding* - A user must be able to bid on an item.

**3.2.2** - *See list of items.*A user needs to be able to see the list of items they are currently bidding on. This may include sub-lists of winning items and won items

**3.2.2.1** - *Current bids* - Shows a list of bids a user is currently in the lead on.

**3.2.2.2** - *Won Items* - This will show a list of the items a user has won and the auction is closed. It will also contain information on how to pay for and retrieve their items.

**3.2.3** - *Edit Profile* - A user needs to be able to edit their profile info in the case that something was entered incorrectly in the registration process.

**3.2.4** - *Find items* - Provide a search feature to locate a desired item

**3.3 *- Registration***. All users need to be able to register and set up their profiles to start out.

We implemented a full registration system for our users. Along with this, we have the infrastructure in place to verify user emails during the registration process.

**3.4 - *Login.*** All users must be able to login to get through security. No actions will be able to happen without being logged in.

Our login system is completely set up. Not only is the user able to login, they can also reset their password as needed. The user can also change their profile's settings.

**3.5 - *Stability.*** The application must be able to support at least 300 concurrent users.

We have tested our system with 3 concurrent users. We do not currently have the ability to test with 300 concurrent users, but we are looking to find a way to effectively stress-test our application. Additionally, our unit tests cover much of the main logic of the application and is therefore a huge testament to its stability.

1. **Non-functional Requirements**

**4.1** - *User Friendly -* The application must be easy to use. It should not be assumed that users will have much technical experience. Because of this, the user experience must be excellent.

Much like the UI design section, we built the application with the user in mind. The interface is kept fairly simply, enabling a user to understand our application with little effort. For example, there are only four main pages for the user to access: profile, explore, item, and profile settings. With this simplistic approach, the user is given a simple and inviting interface to utilize.

**4.2** - *Stable -* The application must be stable. The app will be used at events and must not crash as an auction is going on.

We have tested our website with a variety of auctions, items, bids, and number of users. We have not encountered a stability vulnerability. Additionally, as stated above, our unit tests already cover a majority of a program and show the system to be extremely stable.

**4.3** - *Fast* - The application must load and execute commands swiftly. It must also refresh constantly to keep the user updated on the other users bids.

Because our application is so lightweight, navigating it is very fast. It only takes a fraction of a second to load up any given webpage. In our product backlog, we have the possibility of a user's interface auto-refreshing based on updating bids, etc.

**4.4** - *Secure* - User data will be stored in the application database. This means that the application must be secure and keep that user data safe.

Because we are using Django as our ORM, the users' information is kept secure.

**4.5** - *Robust* - The application must be robust so that everything the user needs is provided. This will make the user experience much better and give people all the tools they need to have a good auction.

We believe we have implemented anything a user would want out of an auction platform. They have the ability to join an auction, view previous auctions, view categories/items, view current bids, bid on an item, win an item, and see the status of an item. Additionally, the user can customize their profile.

1. **Future Features**

**5.1** - *Import items* - Import items from another database such as ebay. This would be useful for mass import

This is a feature in our product backlog. If time allows, we will implement a way to implement external databases.

**5.2** - *QR code* - Give the user the ability to scan a QR code to bid on an item.

This was created during Sprint 2. Scanning in a QR code brings the user to an item's page, allowing them to bid on it.

**5.3** - *Payment system* - Make the payment easier by linking paypal and venmo.

We are unsure if we still want to implement this. Because the auctions performed on the platform are live and silent auctions, we feel payment is something best handled at the point of sale.

**5.4** - *Similar items* - Link items together that the user may also like.

This was implemented in Spring 3. An item's page shows items that the user may also like.

**5.5** - *Bid notifications* - Notifications to text and email when the user is out-bid.

This is a part of our product backlog. If time allows, we will implement a notification system.

**5.6** *- Analysis -* Give the admins the ability to analyze which items sell the most and at what price.

This is a part of our product backlog. If time allows, we will implement a way for an admin to analyze auction results.