Requirements Definition

*Auction System*

1. **Introduction and Context**

This project aims to build a system for running an auction at a school or church activity.

The auction system will be for managing a live event, not running an online auction like e-bay. It aims to replace paper voting with a mobile phone system so that people can know what they’ve spent so far and will also help tally the results at the end of the night. The system will support both a silent auction and a live auction and be capable of running on a PC (for an administrator) and iOS and Android mobile devices.

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

**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.

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.

**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.

**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.

**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.

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

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.

**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.

**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.

**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.

**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.

1. **Future Features**

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

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

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

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

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

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

1. **Glossary**

**6.1** - *Spring Boot* - Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications. This chapter will give you an introduction to Spring Boot and familiarizes you with its basic concepts. (https://www.tutorialspoint.com/spring\_boot/spring\_boot\_introduction.htm)

**6.2** - *Agile* - Agile software development refers to software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The ultimate value in Agile development is that it enables teams to deliver value faster, with greater quality and predictability, and greater aptitude to respond to change. Scrum and Kanban are two of the most widely used Agile methodologies. (https://www.cprime.com/resources/what-is-agile-what-is-scrum/)

**6.3** *- Scrum* - Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. (https://www.scrum.org/resources/what-is-scrum)