Requirements Definition

*Thomas Edison Auction*

1. **Introduction and Context**

The purpose of this project is to build a system that will run an auction at a school fundraiser. The problem that this app aims to solve is the difficulty of running an in-person auction. These difficulties include booking an auctioneer, reserving a space, and gathering volunteers to organize and tally silent auction bids. A silent auction includes paper bids which then must be tallied by volunteers. This is a tedious human driven process which requires much more time and energy than a computerized auction system.

The system we build aims to bring simplicity and convenience to the process of a silent auction. The system we will build will replace a paper voting with an easier access mobile phone system. It will help users know what they’ve spent and help tally results of auction winners. The system will be designed to support both the silent and live auction and will be web friendly and runnable on multiple operating systems.

This application is being built in the context of Thomas Edison’s fundraising auction. Aaron Kay has been helping with this auction at Thomas Edison for a few years and is wanting a fully functional auction app to help streamline the process of a silent and live auction. We as students of CS 3450 have been tasked with creating said application for our semester project.

1. **Users and their Goals**

The human users of this application include administrators, volunteers, and bidders.The main purpose of administrators is to resolve conflicts. They also have the same permissions as a volunteer. Volunteers approve, edit and post donated items. They also have the ability to view the status of an item. Bidders are the standard users of the application. Their purpose is to view and bid on items in the allotted time given for the auction.

All users are required to log-in to the application through a login page. Account flags will be implemented to give administrators and volunteers extra privileges.

1. **Functional Requirements**

* 3.1 The system uses account flags to distinguish users privileges
* 3.2 Bidders can submit bids on desired items
* 3.3 Database keeps track of relationships between users, bids and items
* 3.4 The Admin starts a timer for the auction and locks the application after the time has expired
* 3.5 The system displays user commitment
  + 3.5.1 The system will show the user the total amount of money spent/bidded
* 3.6 The application will display available items in a concise and atheistic design.
* 3.7 The system will keep a list of all winners and produce this list after time is up.
  + 3.7.1 The admin will be able to submit live auction winners which the system will make a list of and present.
* 3.8 Admin can add items to the application and remove items.
* 3.9 The system will be able to handle 300 concurrent users.
* 3.10 The system will assign each user a user number for ease of tracking winners.
* 3.11 The application will be friendly to different OSes

1. **Non-functional Requirements**

* 4.1 A regular user will not be able to add or remove items for the auction.
* 4.2 Users will not be allowed to inject SQL into the application.
* 4.3 All users will be required to sign in.
* 4.4 The application will have fast responses to user requests on bids.
  + 4.4.1 The system will not slow down/freeze up upon multiple user requests on the same item or at the same time.
* 4.5 An admin will not be allowed to bid on an item.
* 4.6 The application will be easy to test
  + 4.6.1 Unit tests will be straightforward to test on the system.
  + 4.6.2 Regression tests will be thorough to cover what unit test can’t.
* 4.7 The system and site will be secure for the user
* 4.8 Availability will be constant as long is the server is running.
  + 4.8.1 In case of a crash of server or application there will be a simple way to bring the application or server backup.

1. **Future Features**

*This section contains a list of ideas or features that are beyond the scope of the project.*

1. **Glossary**