



JOROCHIAL Auction System Design Specification Document

Authors:

Chiara Maalouf
Alyssa Schilke
Robert Ackerman
Joshua Bennet

Document:

Design Specification Document

The total number of pages:

11

Date:

Sunday November 24, 2019

Location:

Baltimore, MD

cmaalouf@loyola.edu
arschilke@loyola.edu
rbackerman@loyola.edu
jdbennett@loyola.edu

Table of Contents

1.	High Level System Architecture.....	pg. 2
1.1.	System Component & Interaction Identification	
1.2.	Architecture Description and Overview	
2.	Elaborate Design of Modules/Components.....	pg. 3
2.1.	Domain Model	
2.2.	Association Identification	
2.3.	Detailed Textual Narration	
3.	Detailed Implementation Design.....	pg. 4
3.1.	Detailed Class Diagram	
3.2.	Usage Scenario Example	
3.3.	State Identification	
4.	User Interface Design Documentation.....	pg. 6
5.	Backend Database Design.....	pg. 10
6.	Design Pattern Discussion.....	pg. 10

1. A high level architecture of your software system

a. *The Major Components and the Information they share.*

The major components of our system are the Bidder Android Application, a MySQL database, and the Admin Web Application. Bidder Users use the Android Application to share account data, item bids, as well as general auction state information. Admin Users use the Web Application to share item information, account data, auction info, and more. The data-store/database is the implementation of the MySQL server. The database shares all information between all users, both admins and bidders. The web server on Studentvhost integrates the Web Application and the database.

b. *A Description of the Architecture of Our System.*

The system uses a client-server architecture. On the front end, users interact with the system through an application which is connected to a single server, while the administrators connect to the server through a web application. All users rely on the star topology to remain up to date.

Pros: As with most client-server architecture, the Jorchial Auction site has the advantages of separation of responsibilities as seen with separation between client and admin interaction. Additionally, our server fulfills the connection functionality between these two classes of system, allowing for logic and database systems to be separated as well. Secondly, another pro of the client-server architecture is the reusability of server components which although limited for us, can be seen more functionally than hardware-wise. For example, some interactions between the database can be reused/re-coded within the admin system such as getting an item or login verification/sessions within each page. In the future as well, there is a scalability advantage with client-server architecture if we wanted to expand the auction site. Later, we can add resources such as network segments, computers, as well as servers to a client-server network without major interruptions to the network.

Cons: The primary con of the system is that there is a single point of failure. If the server goes down, the system cannot function. However, with smaller systems, this is not as big of an issue. In addition, the system only needs to be fully functional during auctions, reducing the time window in which a server crash would be impactful.

2. An Elaborate Design of the System Components

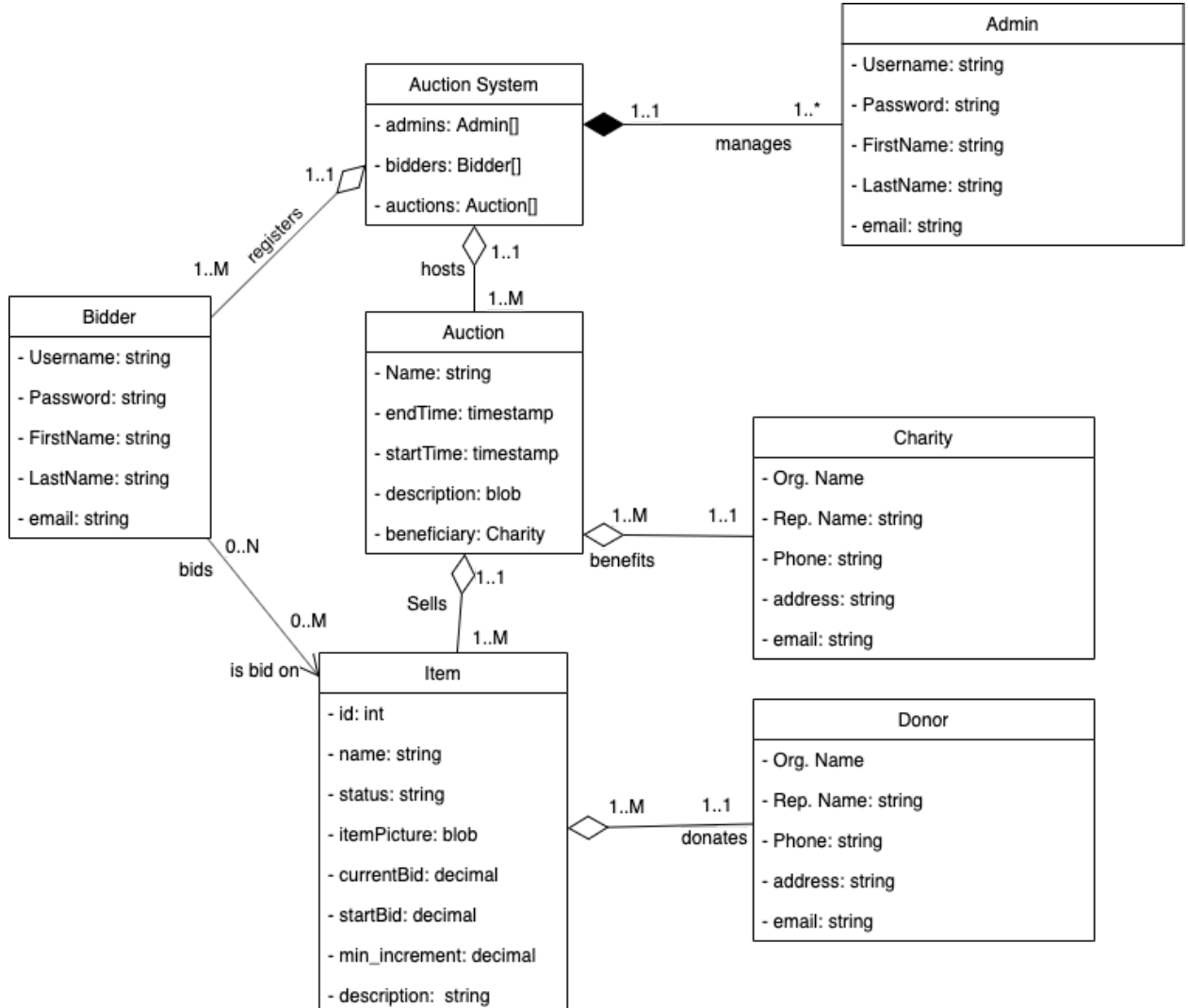


Figure 1: Domain Diagram

The Auction System is composed of admin accounts. These admins manage all the information in the system. The Auction System also has Bidder accounts that have registered. The Auction system can hosts multiple auctions at once. An auction has a beneficiary, a Charity that receives the profits of the auction. Each auction has multiple items for sale, each of these items has a donor. The bidders place bids on items. The admin and bidder accounts have a username, password, email and a first and last name. An auction's attributes include a name, endTime, startTime, as well as a description. Donors and Charities both have an organization name, a representative name, phone number, and email. Lastly, an item has the following attributes, an id number, a name, a

status, a picture, a current winning bid, a starting bid, a minimum increment, and a description.

3. Detailed Implementation Design

a. UML Class Diagram of the System

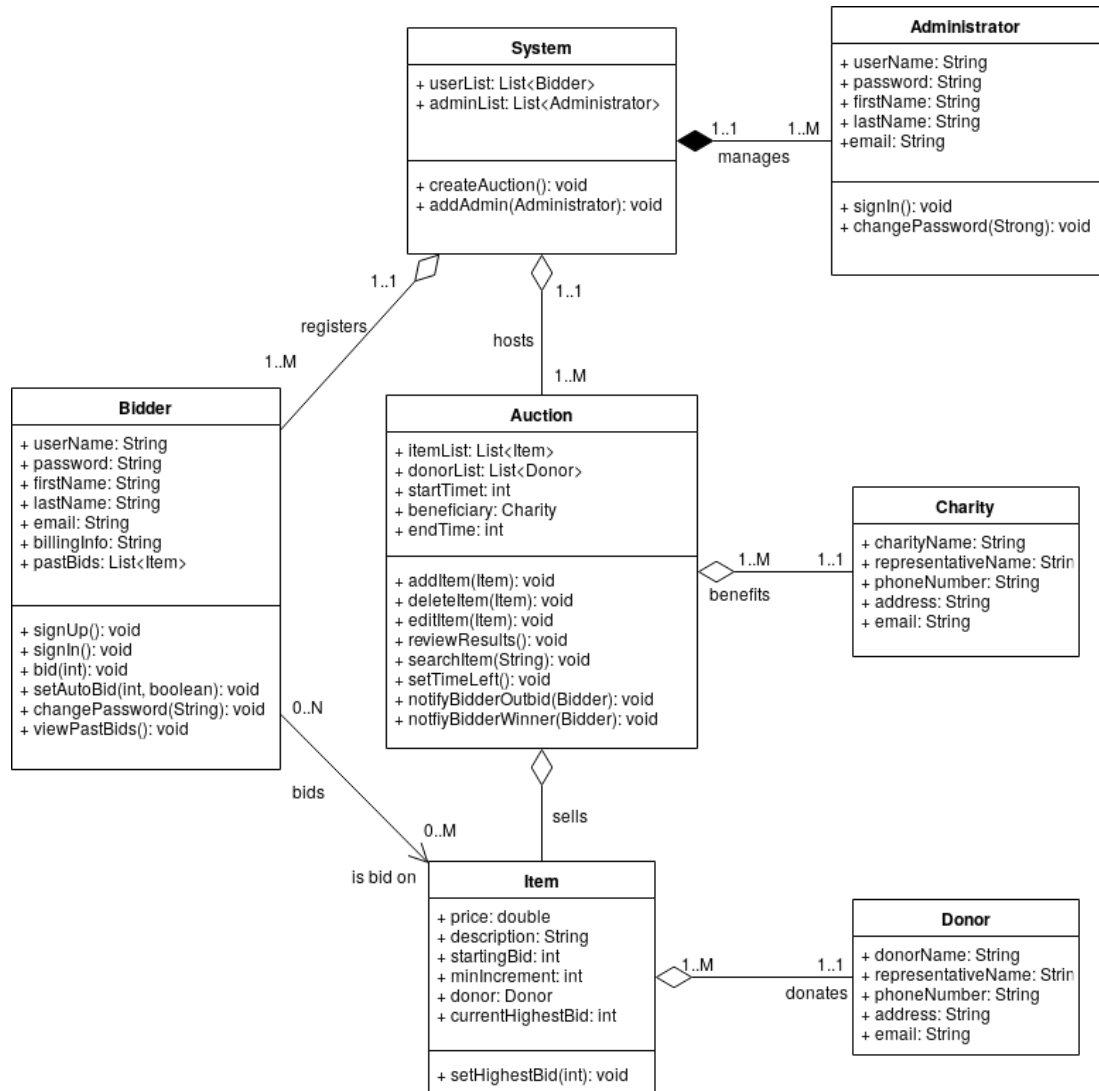


Figure 2: Class Diagram

b. UML Sequence Diagram of the Bid on Item Use Case

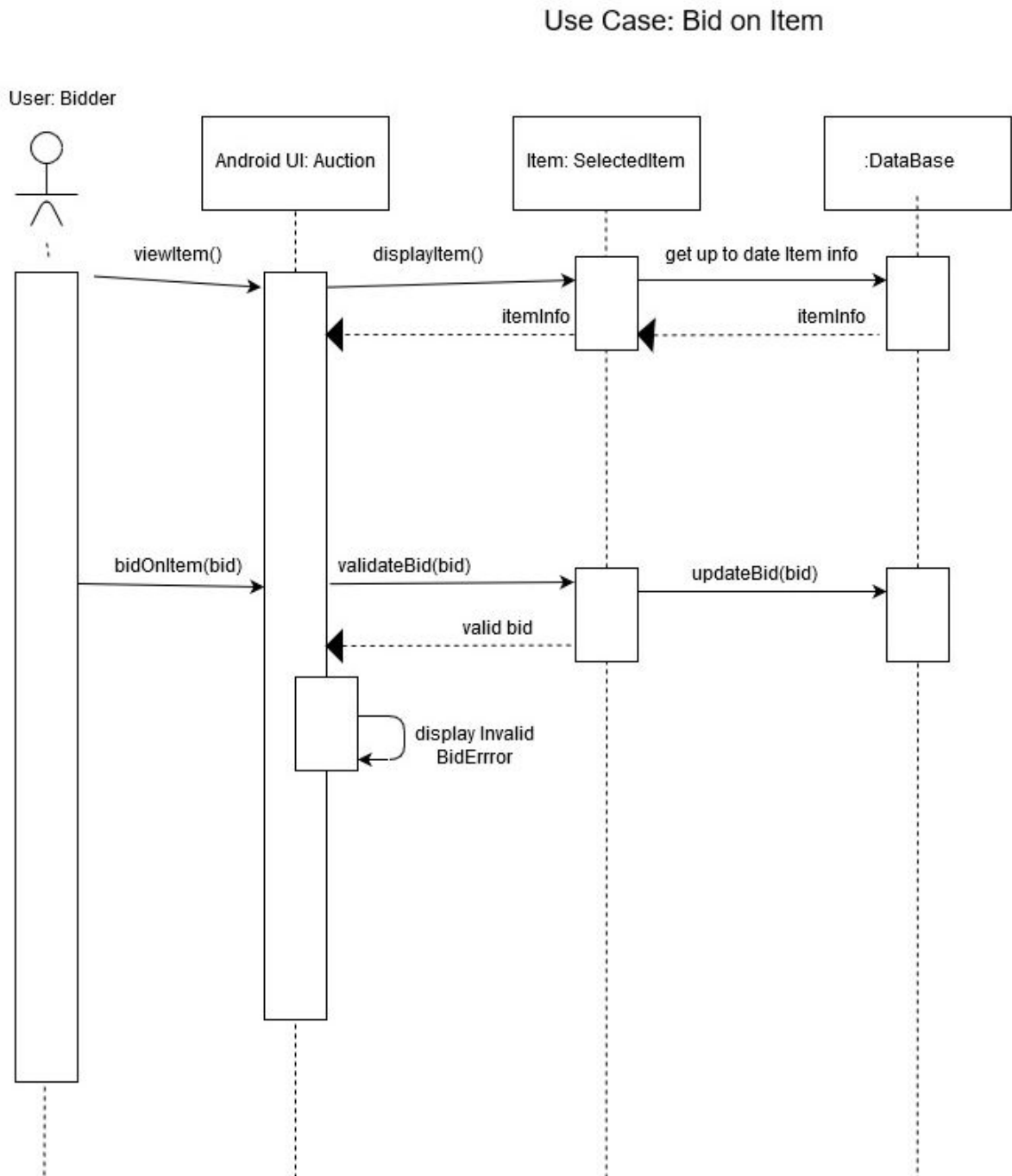


Figure 3: UML Sequence Diagram of Bid on Item

c. *State Diagram of an Item in the Add Item Use Case*

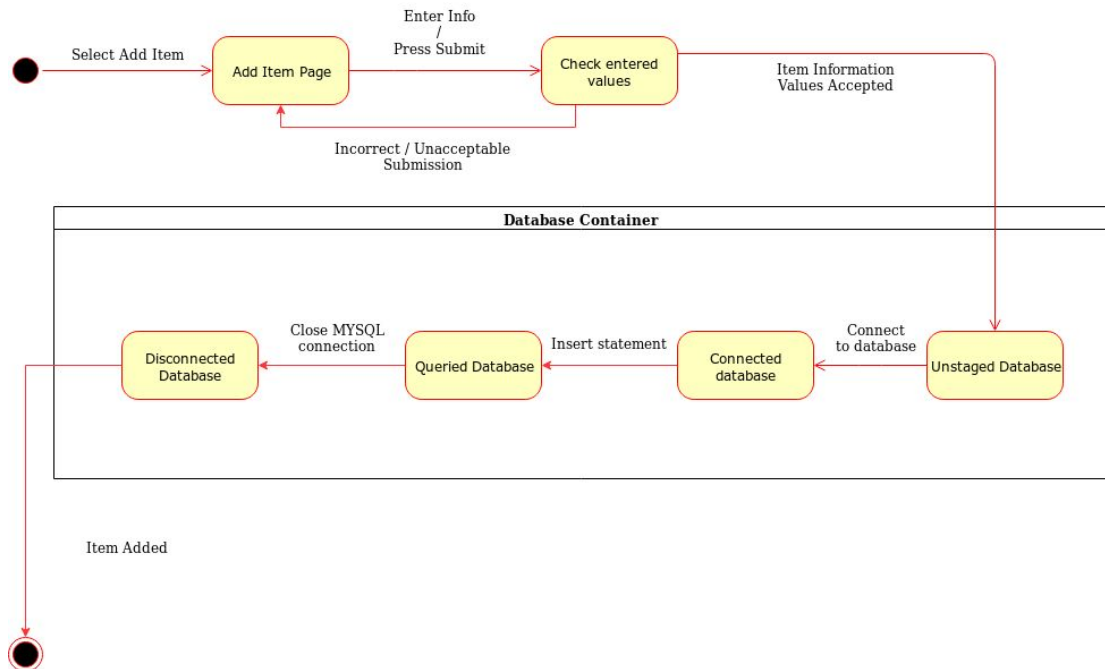


Figure 4: State Diagram of adding an Item.

4. User Interface Design Documentation

Our user interface is separated between the bidder and admin experiences. The Web App is a series of forms for inputting auction information and dashboards for reviewing input and summarizing output. The Android app also displays view screens as well as taking input in order to bid on items and creating bidder accounts. See below:

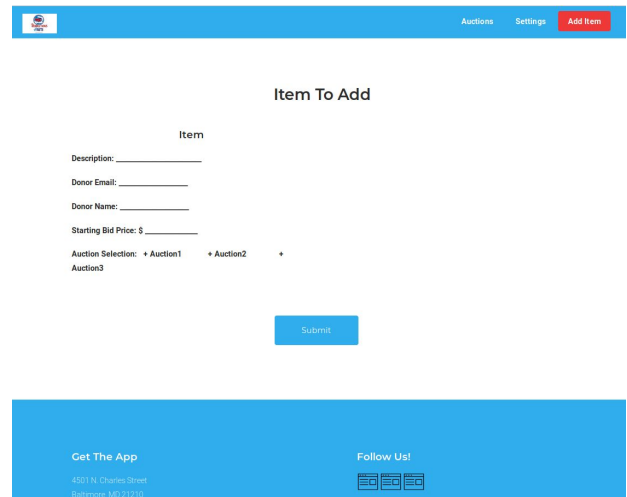
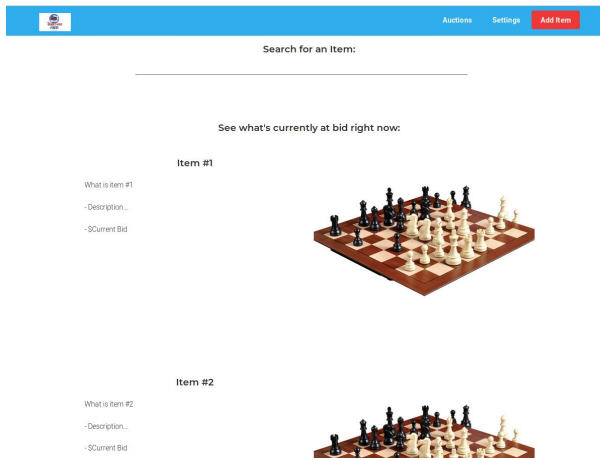


Figure 5a-b: Examples of Web Mockups

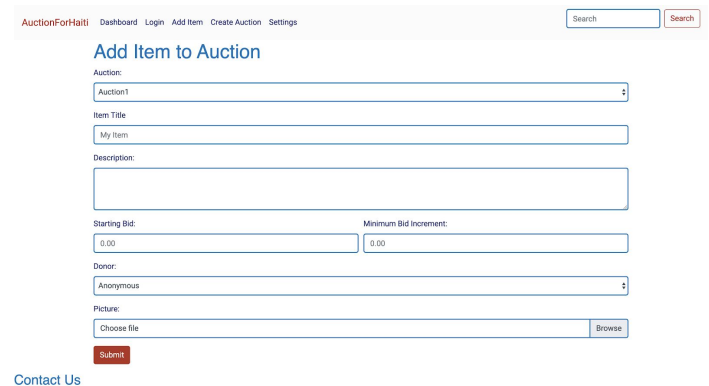
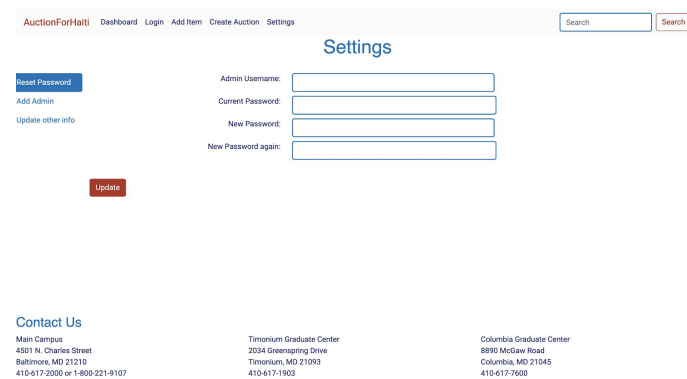
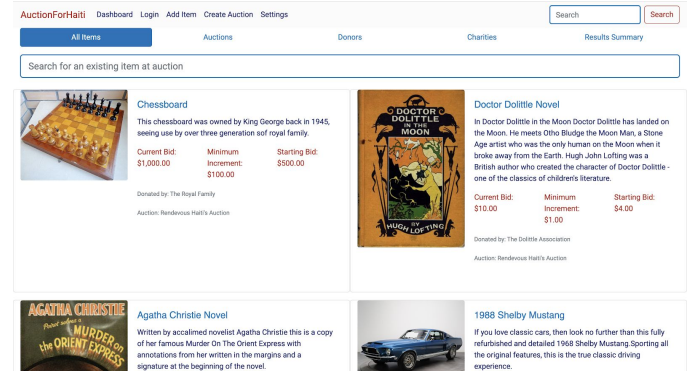
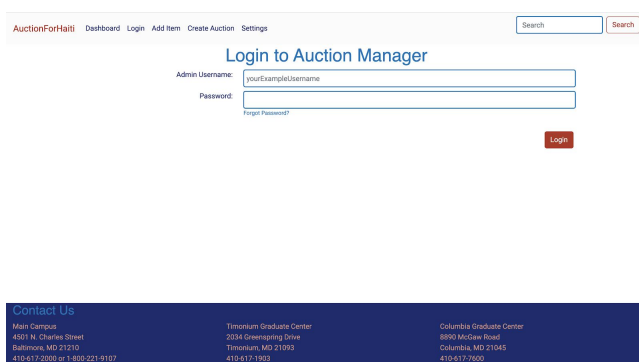


Figure 5c-5h: Web User Interface Examples

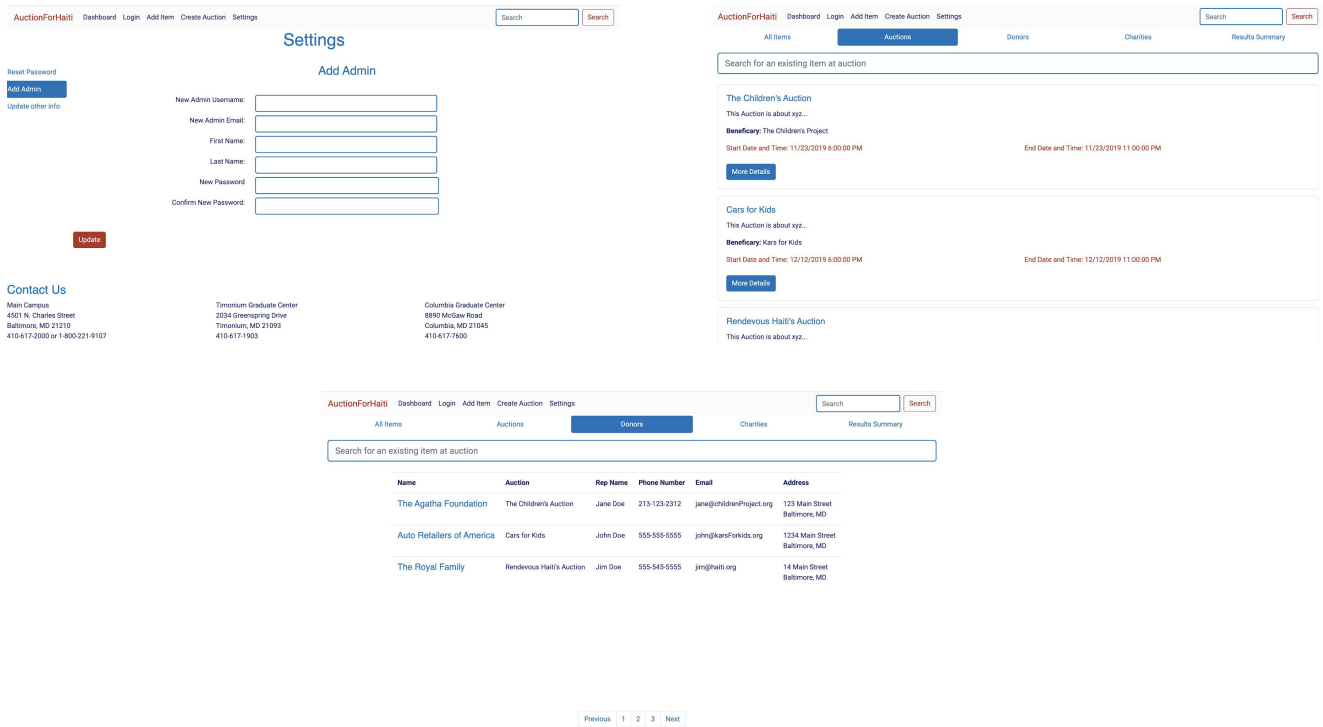


Figure 5i-5k: Web User Interface Examples

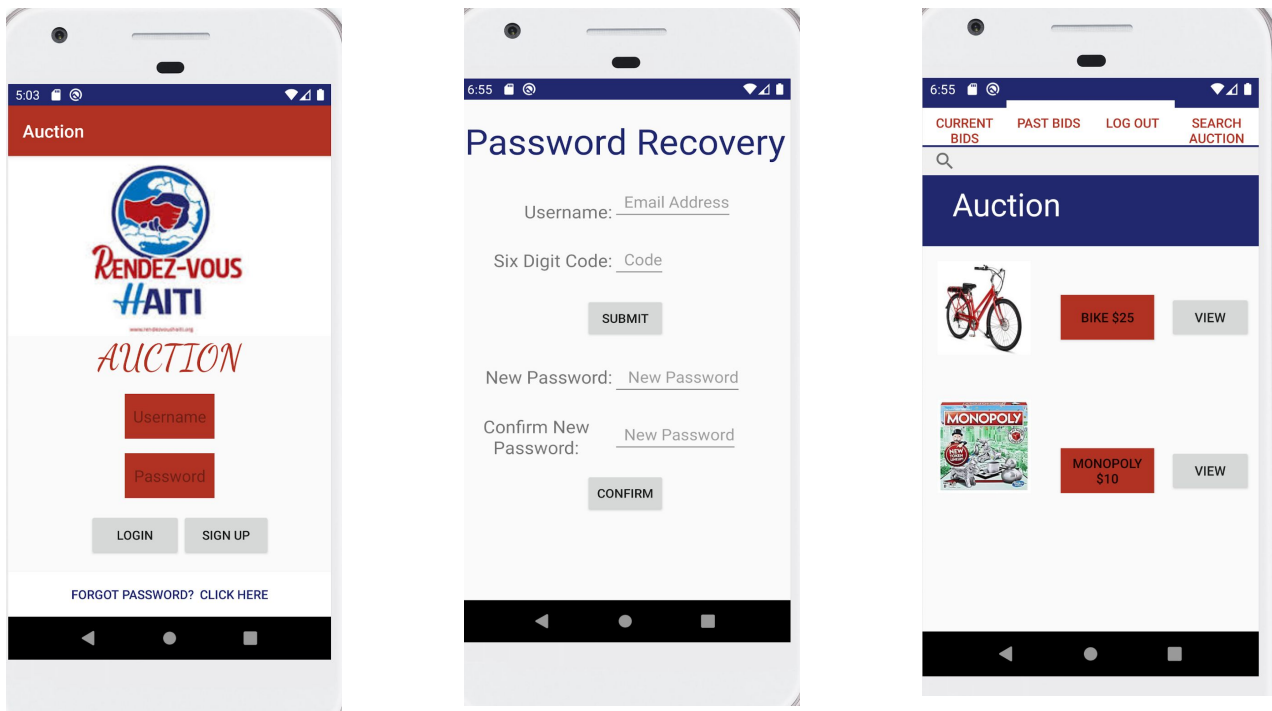
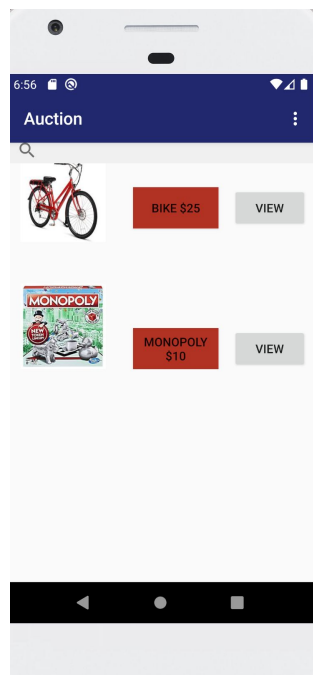
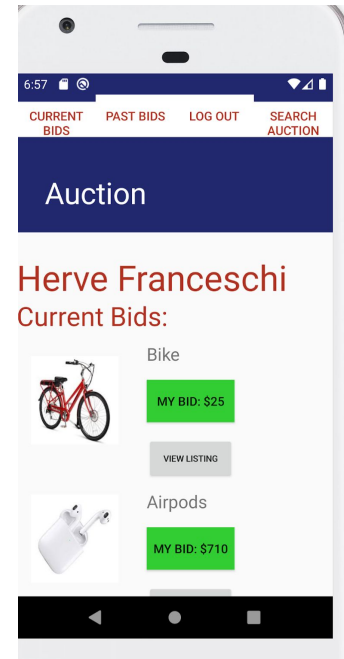
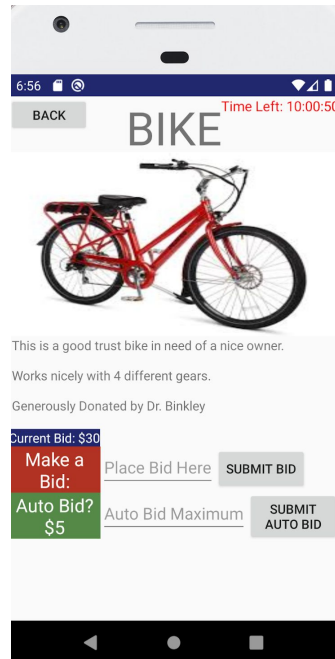
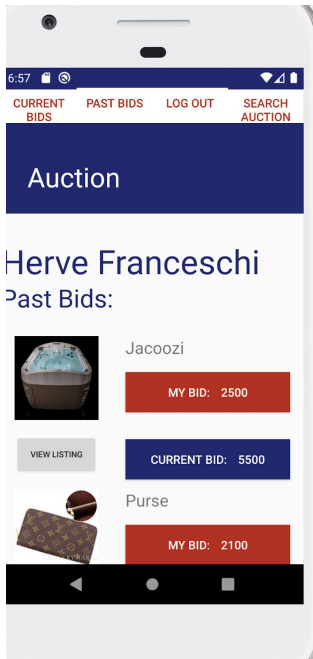


Figure 5l-5n: Bidder Phone App Examples



Figures 5o-5r: Android User Interface Examples

5. Back-end Database Design

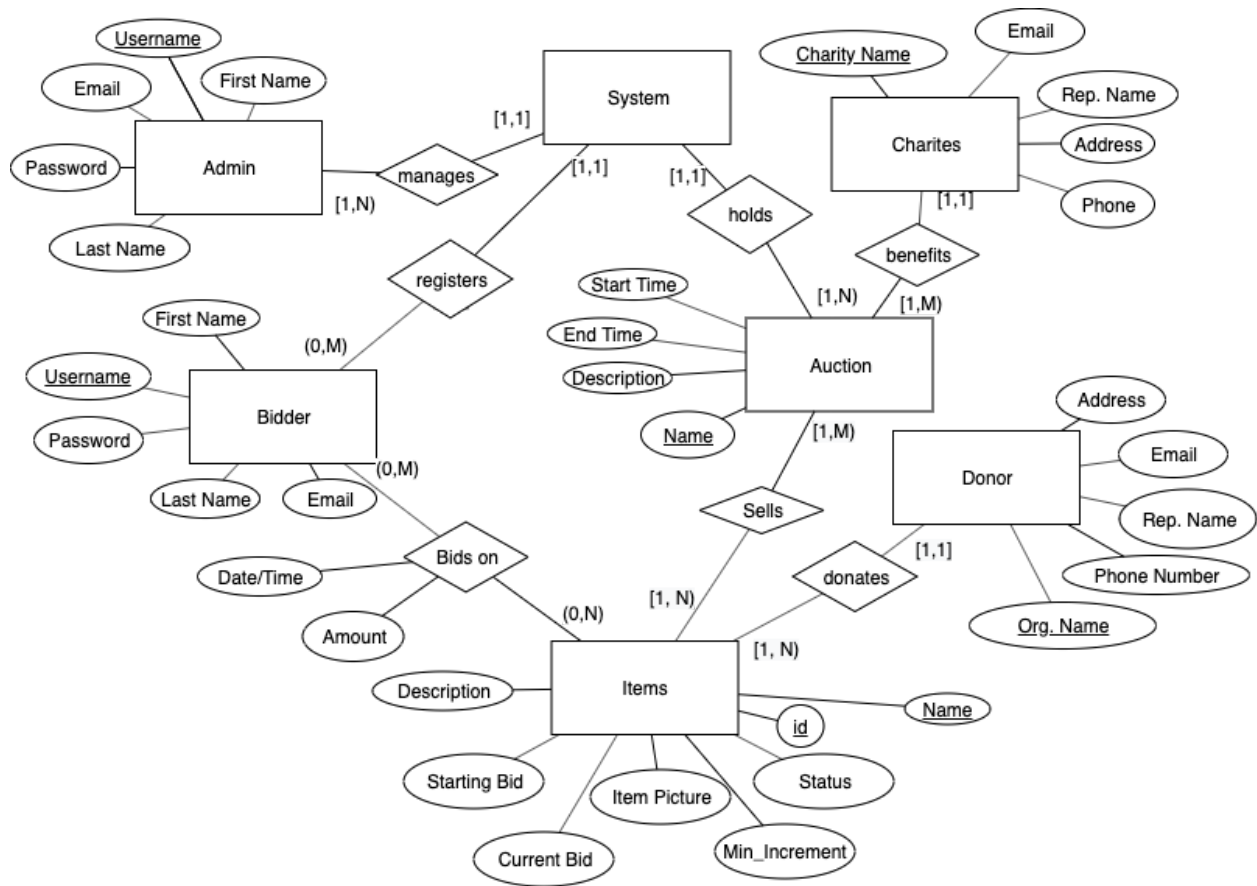


Figure 6: ER Diagram

6. Design Patterns

To be completed in Final Report