Abreham Temesgen

22 April 2022

Auctioneer Report

Overview

I selected this project after thinking over several ideas from health apps to social media aggregators. I realized that I was interested in the fad around NFTs and found that it made sense to create something that was easier to participate in than blockchain technology and was also interactive because all the art is made on the app. I also liked the idea of making an app that had a social aspect to the trading of art pieces and went with an auction-like approach since that would bring multiple people together vying for the same pieces of art.

My original idea had only a few artists but I decided that everyone should be able to be an artist and also buy/sell art. I wanted to implement various mediums for which art can exist but I decided that for my app, art in the most common sense of a stylus and canvas was the most relevant.

Minimum Goals

My first goal was to include a profile page with avatar creation. This page would allow a new user to create an account with an avatar image creation tool and some user information inputs. This would mean that the user would be able to select from a few different features to put together their desired avatar. The idea came from many MMO games I had played which allowed the user to customize, in great detail, their avatar from their facial structure to outfits. I wanted to go in a similar direction but with a more simplified granularity of control. In the end, I opted for choosing an avatar from a preselected group of male and female face/hairstyles and a gradient background. The most challenging part of this process was finding the images I needed as most were copyrighted or of too poor quality to be used.

The next goal was to create a digital drawing tool. A page to allow an ”artist” to draw a simple image with a few interesting tools. This would not be something extremely expansive in terms of tooling (like photoshop), but more like a stripped-down version of the paint program from Windows 7. This part of the project took me quite a lot of time because I didn’t know about PencilKit which had most of what I was looking for in terms of being able to choose from a variety of different pen styles, widths, opacity, color, area selection, etc.

The third goal was to make an auction page. The page displays the auction item, current bid, and options to bid on the item, as well as the amount of time left, the current lead bidder. This page would be one of the more important pages as it's more or less the central concept of the app. I had wanted to add a comments section for the page but I didn’t really think as to why it would be necessary since auctions are mostly conducted in silence with the auctioneer being the one to call for new bids so I didn’t add that to the goals list. This page was mostly visualizing the data in the auction page but I had to be a little more creative with getting the timer to update and how the auction gallery which contains the link to the auction page would deal with a finished auction.

The next goal was a users page. This would be a list of users and their public information. This list would include everyone with an account and a snippet of the original profile page with only the details someone who was visiting would need to see. This would include the name, bio, avatar of the user, and a “montage” of their works. The montage is a gird of the art that doesn’t have any functionality but serves as a snapshot of what the user has been up to. I considered making it only for art that the user themselves created but I decided to include all the art a user either made or owned as showing off a good buy is also part of the experience.

The final goal was to allow users to use augmented reality (AR) to put art they own in their own spaces. This was inspired by a few AR art galleries I had seen because of the covid lockdown. I thought it would make the app stand out from other doodle sharing apps. The AR part was a bit in the wind though since I didn't have reliable access to an iPhone and that was necessary as I use a MAC mini connected to a monitor that doesn't have an in-built camera. I thought it was a little more than a stretch goal however since having a simple list gallery would only be interesting for apps with more extravagant tooling for art creation.

Stretch Goals

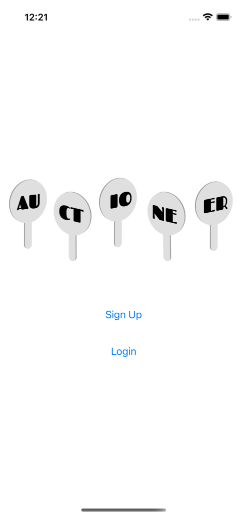
While I implemented all but the last of the minimum goals the stretch goals were more f a hit or miss. The first stretch goal was to implement an auction chat. This was something I considered doing when first thinking up ideas for the app and thought it would make sense. However, the difficulty in setting up a live auction chat with multiple users and the fact that in real auctions typically only the presenter would be the one to talk made it seem like a waste of effort. I didn't implement this stretch goal.

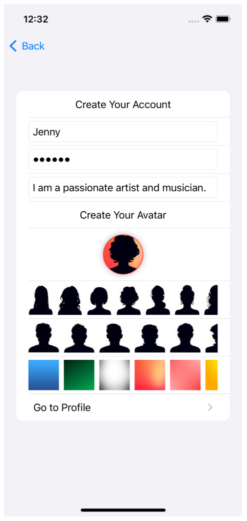
The next stretch goal was implementing in app audio recording or uploading audio. This was another angle for differentiating my app from the rest of the doodle sharing apps on the app store. I thought this would make sense because music is a popular art form and many people enjoy singing even small clips of a song and sharing it on apps like tik-tok. This was another stretch goal I didn't implement but I think It would make a good addition if I continue building on this app.

The final stretch goal was for allowing multiple users to be logged in at the same time. Although I can switch between accounts, to test the auctioning of art I don't currently have any way of allowing multiple users to be logged in. I would obviously need to implement this if I were to try and launch this on the app store but I included it as a stretch goal because I didn't know about firebase and thought it would require some sort of subscription to serve the app on a server I didn't own.

Walkthrough

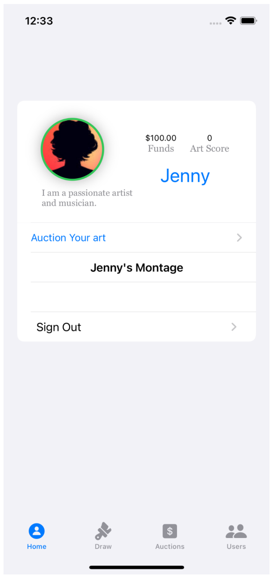
The first page that a user will see is the landing page which has a Sign up and Login button. On clicking sign up the user is taken to the sign up view which includes prompts for entering the users name, password, and their bio information. There is also an avatar creation tool below the text fields for setting up their avatar. The avatar creation tool has options for selecting between different silhouettes and a gradient background.



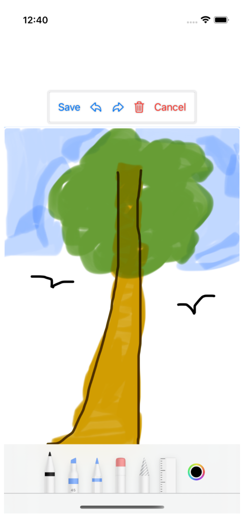


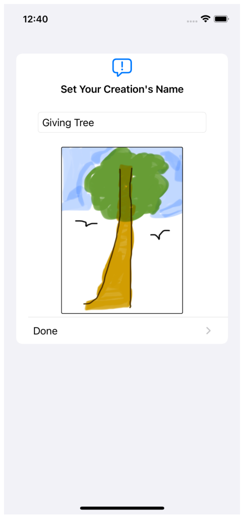
Jenny
••••••
I am a passionate artist and musician.

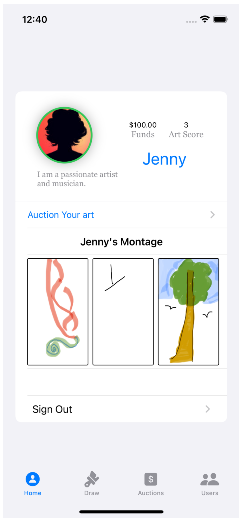
After the user has created their avatar and user information they are taken to their profile page. This includes all of the information they included and links to an empty gallery page, their auctions gallery, and the users gallery. Below the profile header, there is a “Montage” section that shows off all the art the user owns in a grid. The drawing page, the active auctions gallery, and the users gallery is on a tabbed view while the personal art gallery is a separate view accessed via a navigation link. Above the name of the user, there is a “funds” and an “art score”. The funds are the amount of money they have and the art score is the number of art pieces they own.


$100.00 Funds
0 Art Score
Jenny
I am a passionate artist and musician.

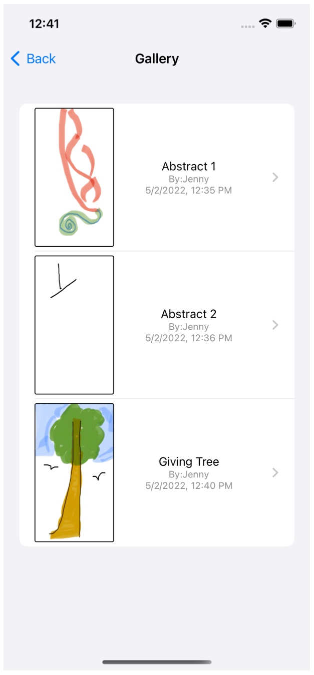
Once the user selects the draw tab, they are taken to the canvas page which has several art tools. There is a standard pen, a highlighter, and a spray like tool. There is also a straight edge, an area selector, and a color selection tool with options to change the opacity, RGB values, hue, etc. For each of the drawing tools, there is a width and opacity option. I included an undo/redo and delete option for the canvas. Once the user finishes creating their art piece and presses “save” they are moved onto the new drawing view. Here they select the name of their art piece and add it to their list of artworks. After they finish they are redirected to the profile page where their new art piece has been added to their montage.

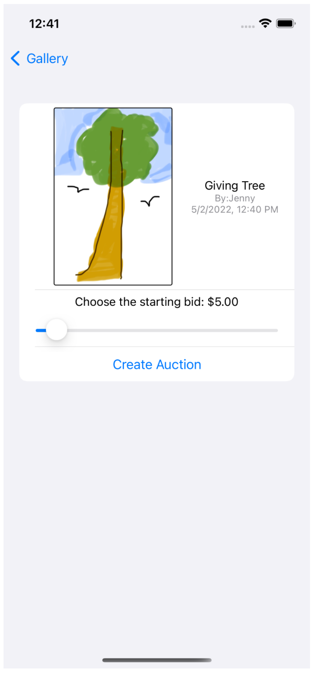






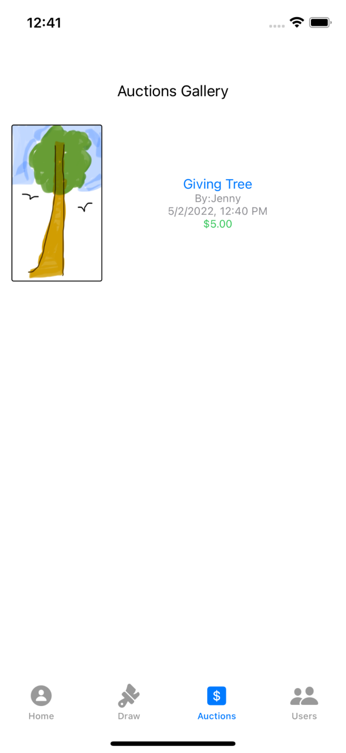
Once the art has been made the artist has an option to put it up for auction. To do so they press the "Auction your art" navigation link and are taken to the gallery view for their art pieces. Here they are presented with a list of all the art pieces they have made and once they select an art piece they are prompted to choose the "Starting bid" and finally to create the auction.

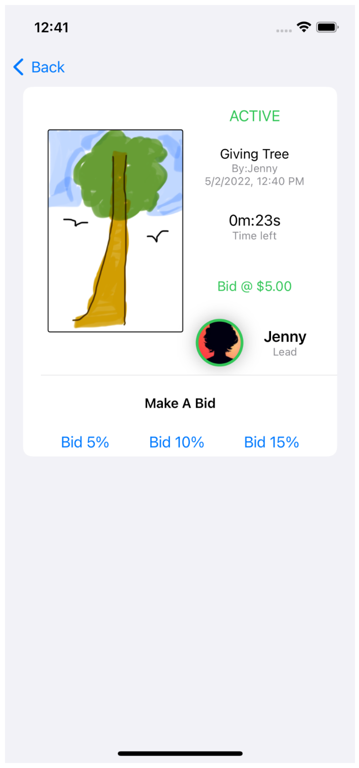




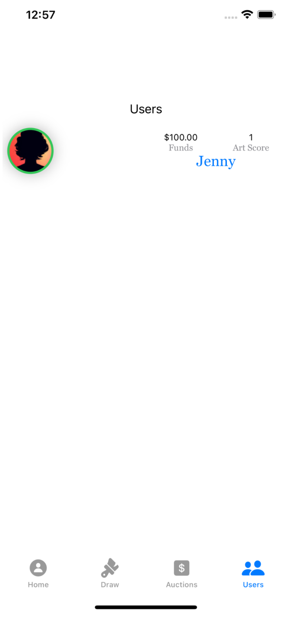
If a user wants to see their created auction they return to the profile page and go to "Auctions". Once they go to the auctions page they are shown a list of all auctions that have been made and are still active. Here they can choose to go watch an auction as it takes place.

Once a user has selected a particular auction, they are shown who created the art that is being auctioned and who is in the lead for the auction (unless someone makes a bid this will just be the person who put it up for sale). There is also a timer for how much time is left for the auction, options to update the bid by 5%, 10%, or 15% of what is started as and the actual art piece itself. Once the auction ends the winner is declared at the top of the page, the winning bid and the art piece are exchanged and the auction is taken off the Auctions gallery.





If the user was interested in the other users of the app and wanted to see their art, they simply would have to select the “Users” tab. This tab has a list of the users, with each user having a link to a stripped down version of their profile page. This simpler page shows the name and bio of the user as well as the amount of funds they have and a montage of all the art they own.



Development process

The technical overview would have models called User, UserList, Auction, AuctionList, and ImageWrap. User defines a user who has a username, a password, a bio, an avatar, a list of artworks they either own or bought from the auction gallery, and money. Most of these were straightforward to implement except for the list of artworks. I had to ditch using the raw Image type because I needed an image to remember who its original creator was, when it was created and other important information for an image to conform to several necessary protocols.

The next model was the UserList model. This would store a list of users so that multiple users can create an account, have artworks, create auctions, etc. It would store a list of users in a list of type [User].

The next model was my Auction model. This model was more complex relative to the other models because it required a counter which needed to fire every second. This was achieved by first creating a. This was because an auction was based on a timer and after the timer was deactivated whoever was the highest bidder would win the art piece. The Auction model also held data on who the original author of the art was the winner of the auction, the current bid, and other relevant information.

The last model was very simple as it only held a single property of type [Auction] but was necessary for passing around a mutable list of information.

When I was developing the graphics for the drawing page I didn't know about Pencil Kit which made the rather challenging part of making a canvas straightforward. Originally I had a model that stores each point that the user went through when they made a drag. This was a "drag" object which stored the sequential positions as a list of CGPoints. There was a Geometry view that had a Path view that took in a series of points and drew them in for me. I would feed it all of the drag objects the user made which each contained a list of CGPoints that the Path understood how to draw. The only issue, which took up a large chunk of my focus before I found Pencil Kit, was in making it draw each drag object with the color selected at the time that object was made. Each time I tried to change the color of the future drag objects it would change *every* drag object I ever made. Pencil Kit handled almost everything out of the box, I just needed to implement my own undo/redo buttons and use the environment object to get access to the undo manager.

A large difficulty with the project was also navigation. I used a combination of Tabs and navigation links and this led to many bugs in the UI because I would treat them the same way but having learned about how to properly remove the navbar from tab view many of these issues were fixed. One important mistake I made was in using multiple Navigation Views, this was the cause of much of my confusion and once I realized I only needed one, I was able to fix many of the navigation bugs.

The timer for the auctions was implemented using a tutorial for making endless counters. The way mine works is it makes the Auction class the delegate for a “BeatEmitter" class. This class has an initialization method which creates a timer running in the background, notifying the delegate every second. The delegate adheres to the protocol for BeatEmitters and so it has a beat method. This method updates an instance of a class “Model” which is a property of “Auction” and contains the counter and updates that counter. This way I know how many beats it's been since the auction began. In the view where I show the auction, I use the beat counter to determine how many minutes it has been since the start of the auction and use a standard onReceive and Timer with some math to display the time left.

Future Direction

The app can contain a lot more functionality in regards to managing past auctions having a personal auctions list, including the AR aspect to displaying a users art, search features for users and more art tools. I think the most important addition, however, is getting the multi-user aspect down using firebase. I wanted to try to implement firebase into the app but I got bogged down by many issues when it came to how views are navigated and lacked some research into how to best implement certain features in my app. The overall look and feel of the app does need quite a bit of polish because I neglected that for after the functionality was finished. Overall I think this will be a good summer project to continue and if it makes sense a release for the app store.