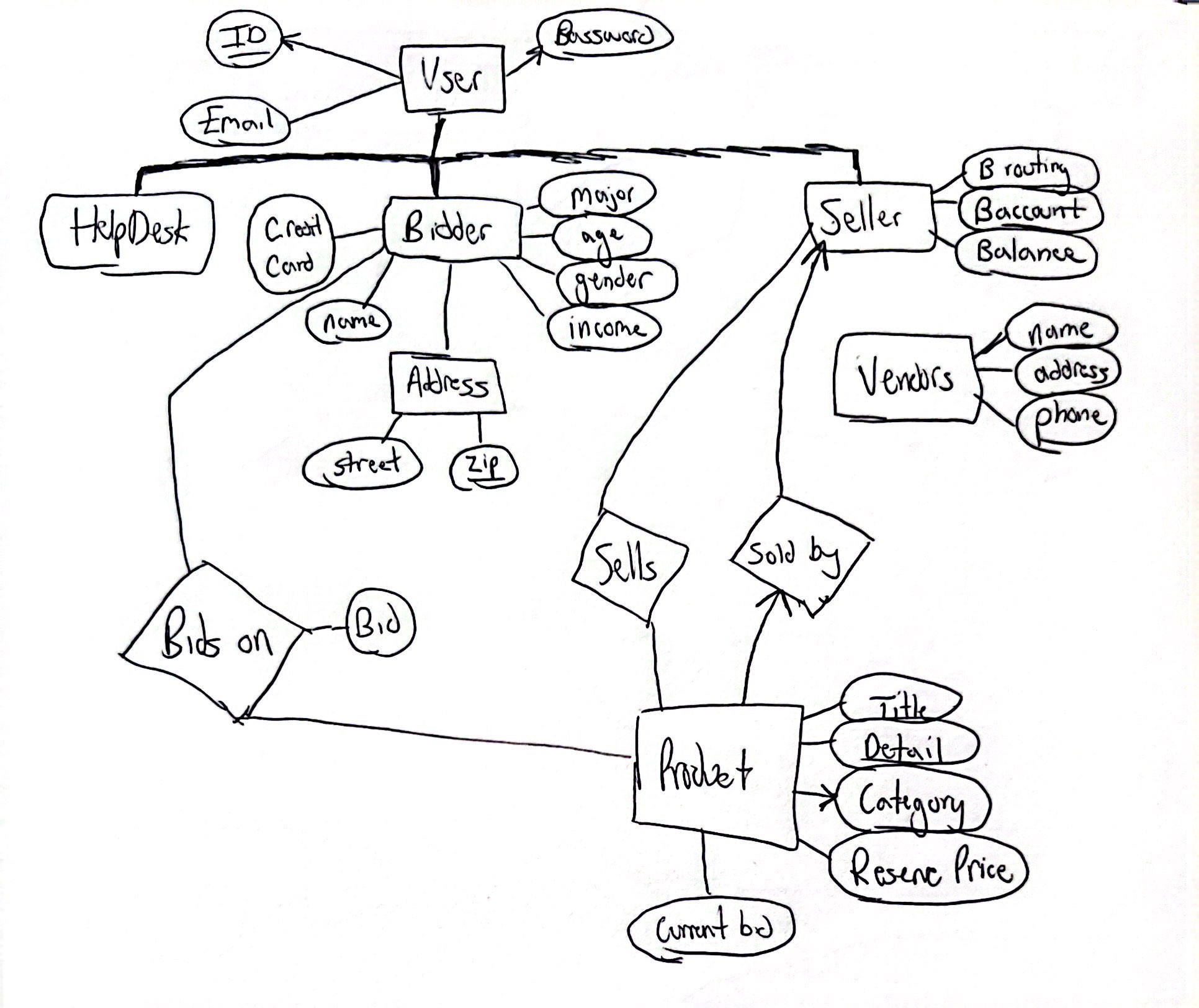
LionAuction is intended to be an online marketplace where individuals and businesses can sell goods in an auction-style format. Mr. Dude has quite a lot of functionality in mind for this project from special local-only products, to specific categories that allow the user to sort products down to exactly what they’re looking for.

Starting with Users, we find User to be the parent of the three different types of users within the website space. There are bidders, sellers, and helpdesk employees, all of which must have an ID (email address) and a password to log into the site. In the case of LionAuction, a bidder can be a seller at the same time, although in order for a bidder to become a seller, they must submit a ticket to the helpdesk. All students are automatically registered as bidders by default, and must provide a host of information, some for their benefit and some for the company’s benefit. They must provide an email, their name, and their address, all of which can be changed in the settings later on. They must also provide their major, their age, their gender, and their annual income, although these pieces of information are solely for marketing analysis. They are assigned an ID within the website which is used to distinguish between users, and this ID cannot be changed unless they submit a ticket through the helpdesk. Bidders must also include payment information to participate in options, which can be changed at any time. Bidders can filter by categories to find the product they want easier, they can send questions about the product to the seller, they can see info about the seller, and related products to the one they are looking at as well. If a bidder wins an auction, they must pay immediately after the bid has ended, and can follow up by leaving a rating out of 5 stars for the seller.

Moving on to sellers, once again they have an ID and login with their email and password, same as all Users. Sellers must include a bank routing number in order to make and receive payments, as well as a bank account number, and must have a running balance within the site to keep track of money they’ve made. For a business or vendor, they must also include their business’s name, the business’s address, and a customer service number, should a bidder need to make contact. If a seller ever decides to leave the site for good and delete his account, all his auctions must leave with him, a seller’s item cannot be sold if the seller no longer exists. When listing a product, the seller must include the category of the product, a description of the product, a reserve price which is hidden from the bidders, and the end time for the auction. They can see the questions asked about the item, as well as the reviews left on their account as a seller. If the product they are selling does not fit into one of the predetermined categories, they can contact helpdesk in order to create a new category. Once an auction ends, the product is delisted regardless of whether or not it was sold, and the information surrounding the auction is archived.

The helpdesk once again carries the same login information as all users, and serves as the support for the site. The helpdesk will deal with a number of requests from both bidders and sellers within the site. If at any time any user wants an ID change, they must contact the helpdesk in order to get it changed, and must provide a valid reason for doing so. In order for a bidder to upgrade his or her account to a seller, they must submit a ticket to the helpdesk. In order for a seller to make a new category for their product, they must submit a ticket to the help desk. The helpdesk is also in charge of using the data from bidders and sellers to perform marketing analysis to continuously improve the site and its functionality.

Moving on from the various users, there are certain requirements and specifications that come with the other aspects of the site, such as the products, the auctions themselves, and the different categories used to sort products. Starting with products, each product can only be sold by exactly one seller, and belongs to exactly one category. If multiple of the same product are listed, they are auctioned off together. As said before, each product must have a title, description, category, and reserve price. Additional information can also be included if the seller deems it necessary, and products will disappear if their seller disappears. Bidders can place multiple bids on different products at a time, so long as they are not the seller of any of the products. All bids must be at least $1 higher than the previous bid, and must be placed before the auction is over. Once over, all bidders are notified of what the highest bid was, and the owner of that bid must then pay the seller to complete the transaction. As for categories, each category may have its own smaller subcategories, until you reach the leaves, which is the lowest category form. Buyers can select these categories as filters in order to narrow down their search.



Pictured above is a rough ER diagram representing Mr. Dude’s idea for LionAuction. We can see that User is the parent entity that a helpdesk employee, a bidder, and/or a seller all stem from, all inheriting the three main attributes of a user, their ID, their email, and their password. A helpdesk employee has no separate attributes outside of what is provided by the User hierarchy, however we can see that both bidder and seller have their own separate sets of attributes unique to them. We see that bidders have a credit card for payments, as well as a name attribute and a separate address entity that is based off of a street and a zip code, the zip code being used to determine the city and the state. We can also see that a bidder has a set of four attributes on the side that are used solely for market analysis. The seller has its own child entity, a vendor that is a specific type of seller. An individual seller must include some financial information in order to be able to sell their goods, but a vendor must include more information, such as their business name and location, and a customer service number, should a bidder need to get in touch for any reason. We then see the product entity, which holds relationships to bidders and to sellers. A product must have a title, a detailed description, a category, and a reserve price. Each product may only have one category which we can see in the diagram represented by the arrow, and each product can only be sold by one seller, which is also represented by an arrow. Each product also has a current bid price attribute, which will be matched against the bid attribute in the bidding relationship between product and bidder, so that we can confirm each bid is at least $1 more than the current highest bid. This ER diagram roughly represents how Mr. Dude specified LionAuction should work.

When creating a project such as LionAuction, there are many options when it comes to technology. We’ve learned in class primarily about SQL and Python, but those aren’t the sole options. One option is PHP, which is a popular platform that is used to develop complex websites. It has been used many times, and some would consider it a go-to when deciding what to use to create an interactive website such as LionAuction. The pros are that it is free to use, open-source, and can scale very well to the size of your project. The cons however, is that it is not as versatile as some of the other options out there, and it doesn’t debug very well. Both of these are areas that other options may improve on, so PHP is likely not the best option for us. Another option is R. R like PHP is open source, and easy to pick up. However, R is mostly for use on data, and doesn’t have as much utility when developing a website. This means that for handling data, R could be fine, but since this project entails more than just the data side of things, R is not the best option for us. A third option is C#, which is very similar to R in the sense that it has great utility when creating and managing datasets. However, once again creating and managing the dataset is not the only important part of this project, so we will want to stick with an option that is best for the entire project, not just one aspect of it. Despite these options not being the most attractive, that doesn’t mean they are entirely useless. PHP, R, and C# have all had enormous impacts on the world of programming and data science. They all have a niche that they fit, and are very well suited for the purpose they were designed for. It just so happens that our project includes both a dataset and a website aspect, which means we should be using something that can help with both those aspects, as PHP is best with websites and not so much data, which is the opposite for R and C#. This is why I believe using Python for this project is the best option, as the Pandas package handles data very well, arguably better than some of the other options I have outlined, as well as having Flask in order to help organize a working site.