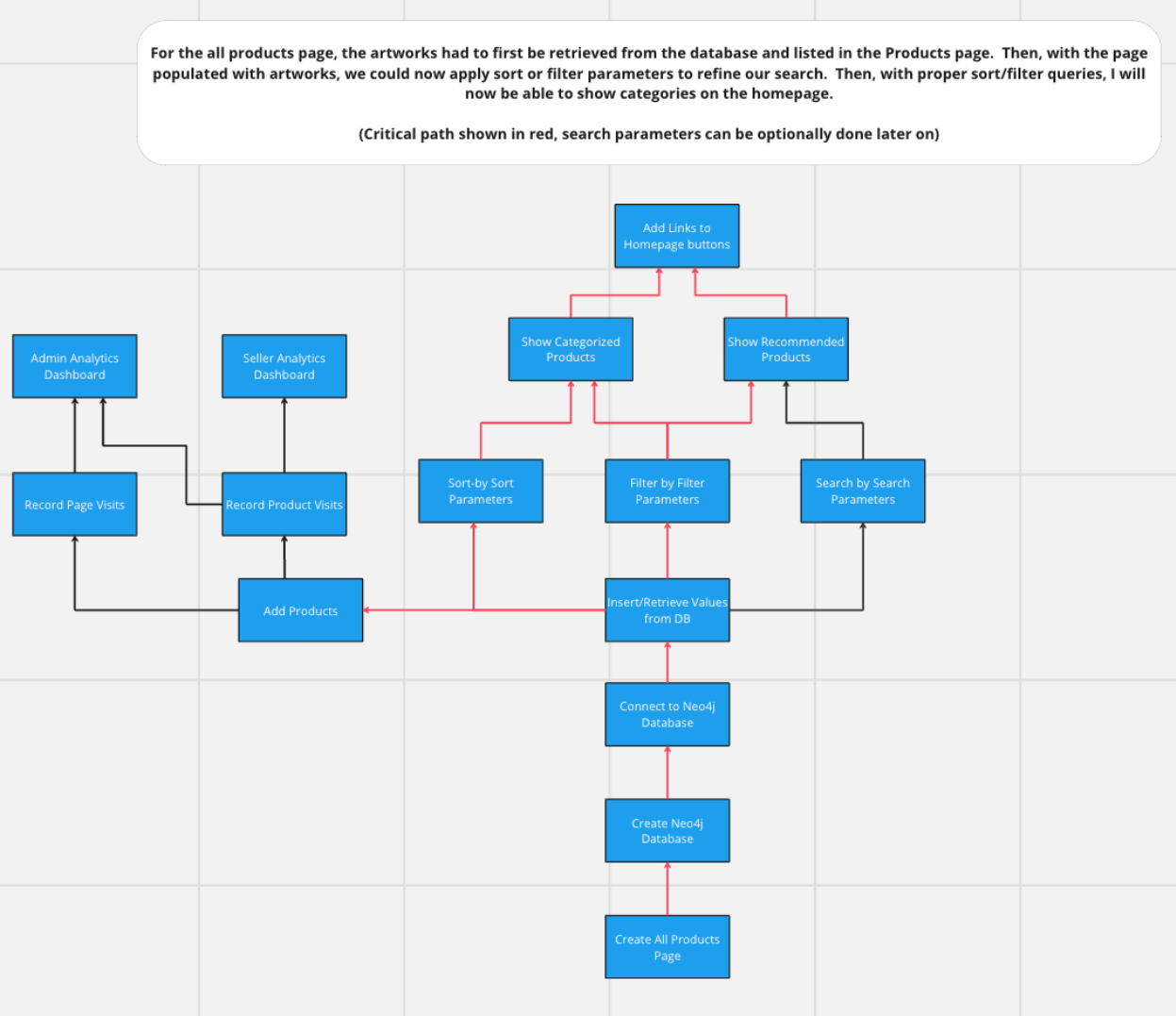
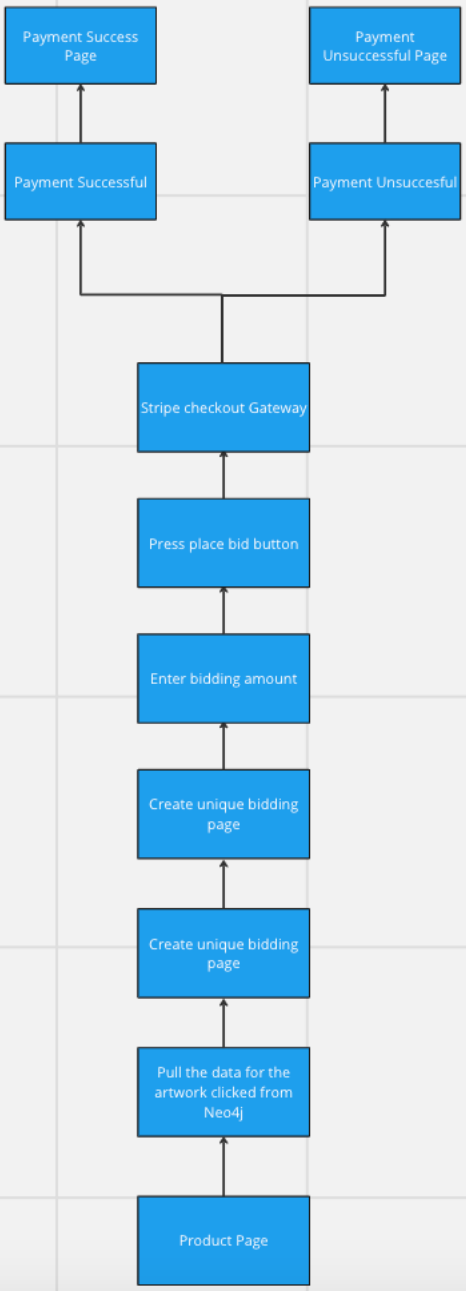
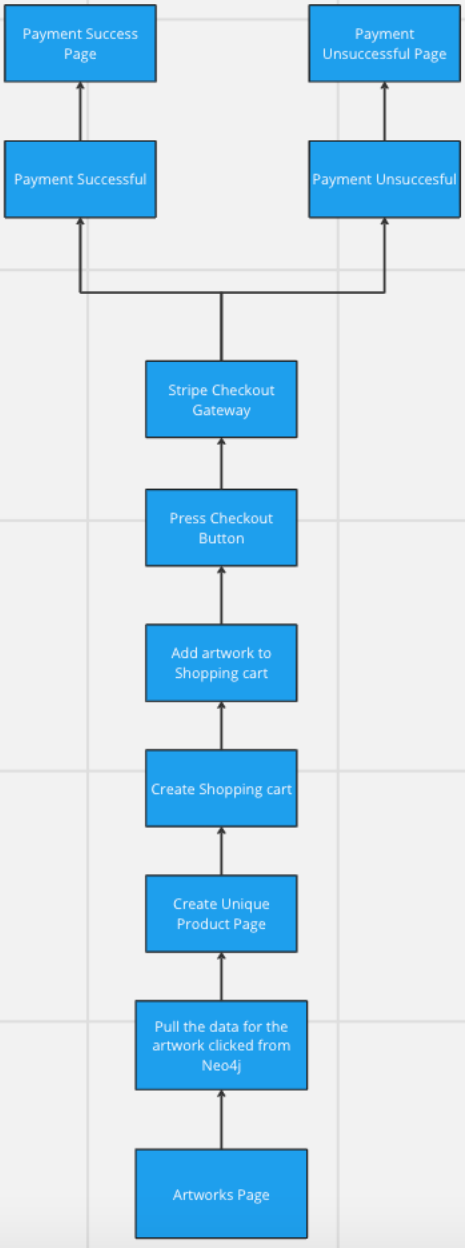
SCHEDULE


For the all products page, the artworks had to first be retrieved from the database and listed in the Products page. Then, with the page populated with artworks, we could now apply sort or filter parameters to refine our search. Then, with proper sort/filter queries, I will now be able to show categories on the homepage.
(Critical path shown in red, search parameters can be optionally done later on)
Add Links to Homepage buttons
Show Categorized Products
Show Recommended Products
Admin Analytics Dashboard
Seller Analytics Dashboard
Sort-by Sort Parameters
Filter by Filter Parameters
Search by Search Parameters
Record Page Visits
Record Product Visits
Add Products
Insert/Retrieve Values from DB
Connect to Neo4j Database
Create Neo4j Database
Create All Products Page

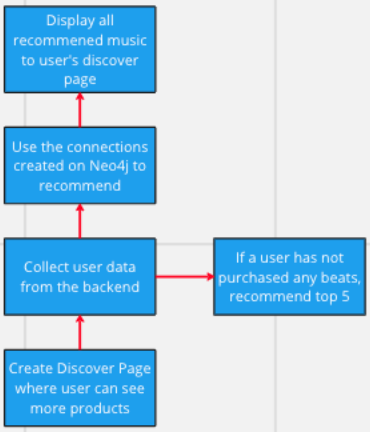
In order for users to be bid on a product, they must be a logged in user. On a specific product page, a user can click on a "Place a Bid" button to take them to a bidding page, as long as the specific product is set up for bid and not for direct purchase. Once on this page, the user can decide to place a bid or to return to the original product page they were at. When the user selects to place a bid, they will be navigated to a page that allows the user to enter a bid value and place a bid, as long as it is greater than the starting bid and as long as the specified end date is after the current date. The user with the highest bid by the time the end date passes will be awarded the product.



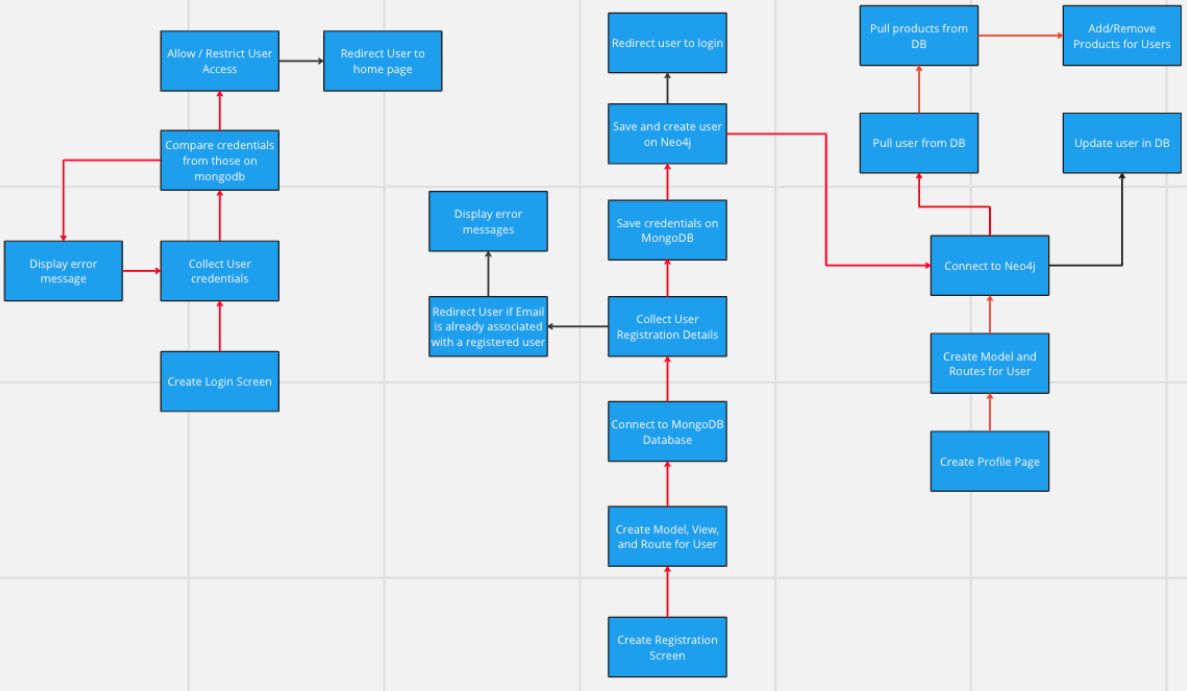
In order for the user to purchase an artwork, they need to be logged in. Then when they click on an artwork it takes them to the product page for that artwork where they can add the artwork to the shopping cart and then press checkout which will take them to the stripe secure checkout page. Over there they can enter their payment details and complete the transactions.

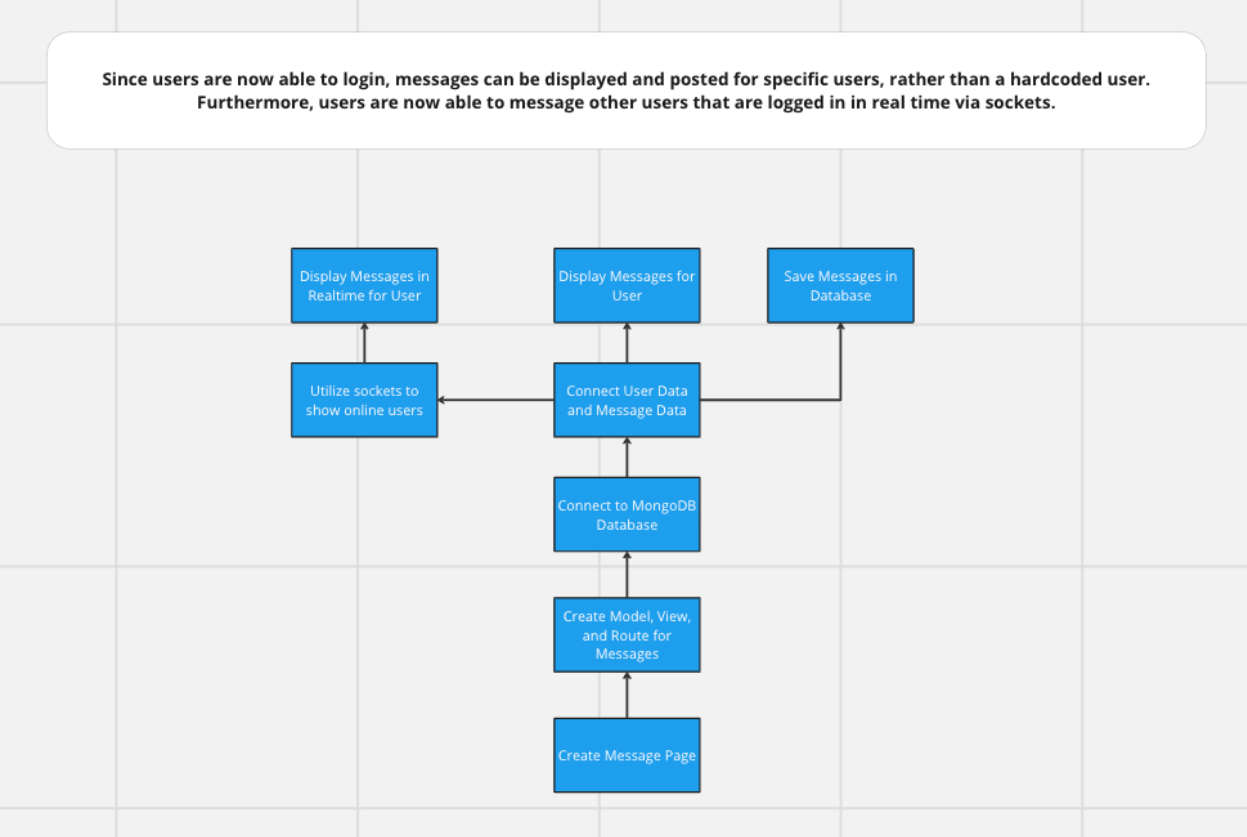


In order for user to be recommended songs, they must be a logged in user. For this, users are recommended music based on the music that they perviously purchased.
The critical path highlighted in red shows what must be carried out in order for a user to be recommended beats



For the registration and login of the user, several steps had to first be taken before this can happen. Firstly, when the user is registered, we have to make sure that emails are left unique. Further, we must now create a new user in both mongoDb (that will store user credentials) and neo4j (that stores users artwork, profile picture and more personal details).
The critical path highlighted in red shows what must be carried out in order for a user to be properly registered and allowed to change / edit their details





The above images show the Network Diagram with various tasks and their dependencies

To keep the sprint on schedule we had very regular stand-ups to track progress. This way we can always adjust our priorities accordingly. Further, we have a discord server that is active most days. This frequent communication with the team also allows us to prioritize and assign tasks very efficiently. This also allows minimum merge conflicts.

Since this was our last sprint, we planned on perfecting most of the features that we already had implemented, as well as focused on making our app more consistent. We were able to get a lot done since we opted to work in pairs, which proved to be very sufficient. We were still unable to complete every user story initially planned for our project, but we were able to complete majority of our sprint goals.