1. **What did you do last week?**

In the first week, I started outlining and drawing diagrams to get a general idea of how I’m going to go about this. I created a basic user flow diagram; See below:

A diagram of a diagram

Description automatically generated

Then, I created an initial ERD to get a feel of what type of data I would need for this project. Somewhere in the middle of that, I decided to not use MySQL and use MongoDB instead. The no SQL part was intriguing and I’ve worked with MongoDB before in other personal projects. Along with this decision, I also chose to not use Node.js and Express and just use Go because I really like Go and I’ve been playing with Go lately more than I do with JS/TS. It’s not really the traditional type of OOP language, but it does support the concepts and patterns with structs. It’s basically OOP.

From the diagrams, I figured the Login and Signup parts would be the most fundamental parts of my application, so I decided to start implementing that. But before that, I wanted to decide if I should start with the backend or frontend first. I chose backend because I wanted to do something different than what I usually do. And with that, I started writing the code for the login and signup. I finished those parts. I also wrote tests for the login and signup handlers, which I think are integration tests since the endpoint handlers are interacting with the MongoDB. There’s also some unit tests for those other individual functions.

Then to add with the authentication, I wrote a simple session management package/class/module and made sure to store session ids in the DB. I wrote a test to see if the method to get a session given a sessionID string returned the appropriate session.

I finished the authentication and the session management.

1. **What do you plan to do next week?**

I plan to implement the /list\_furniture endpoint, which handles requests for a new furniture listing. Each new furniture listing has certain information that needs to be provided like a title of the listing, a description, pictures, etc.

I also plan on writing a test for this, hopefully before I actually implement the code for the endpoint handler so I can practice the TDD methodology. Test first, then implement. Right? So, that’s what I’m going to do

1. **What is blocking you from going forward?**

How to send uploaded image files in http request? I did some research and read something about reading images and encoding them into base64 encoded string, and using that string to send. But, I’m not certain on the most reliable method