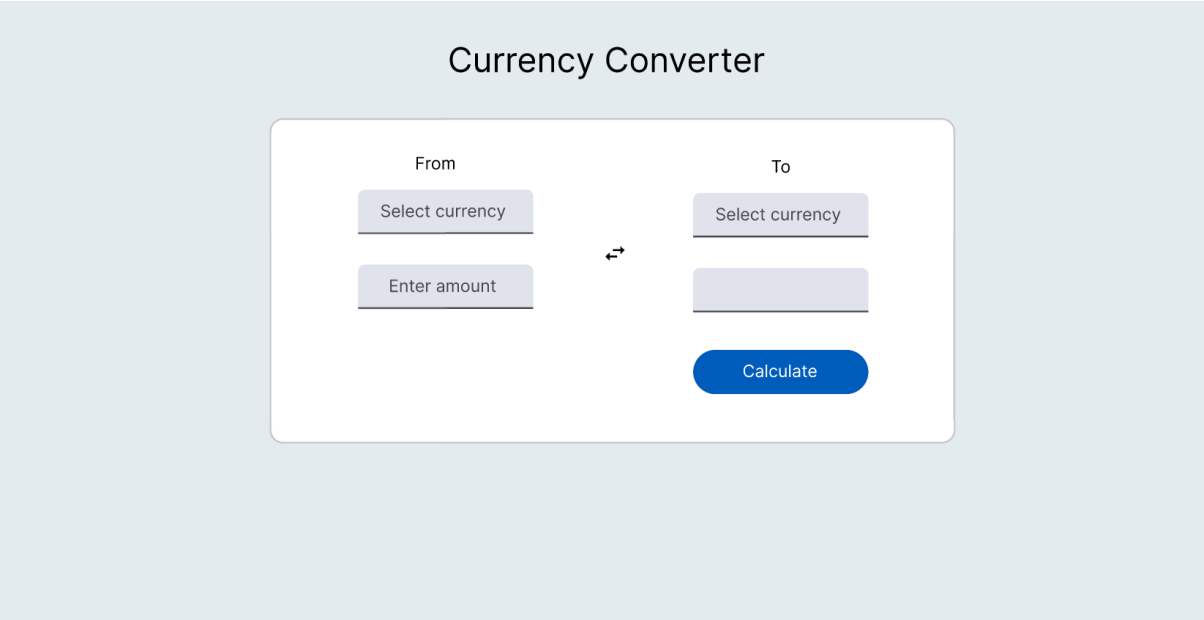
Currency Calculator Thought Process

I made this PDF so that I could explain my thoughts clearly and take you through the steps I took to make this app.

I decided to make the app in Angular as this is what I have most experience in and to use the most recent version of Angular 20.

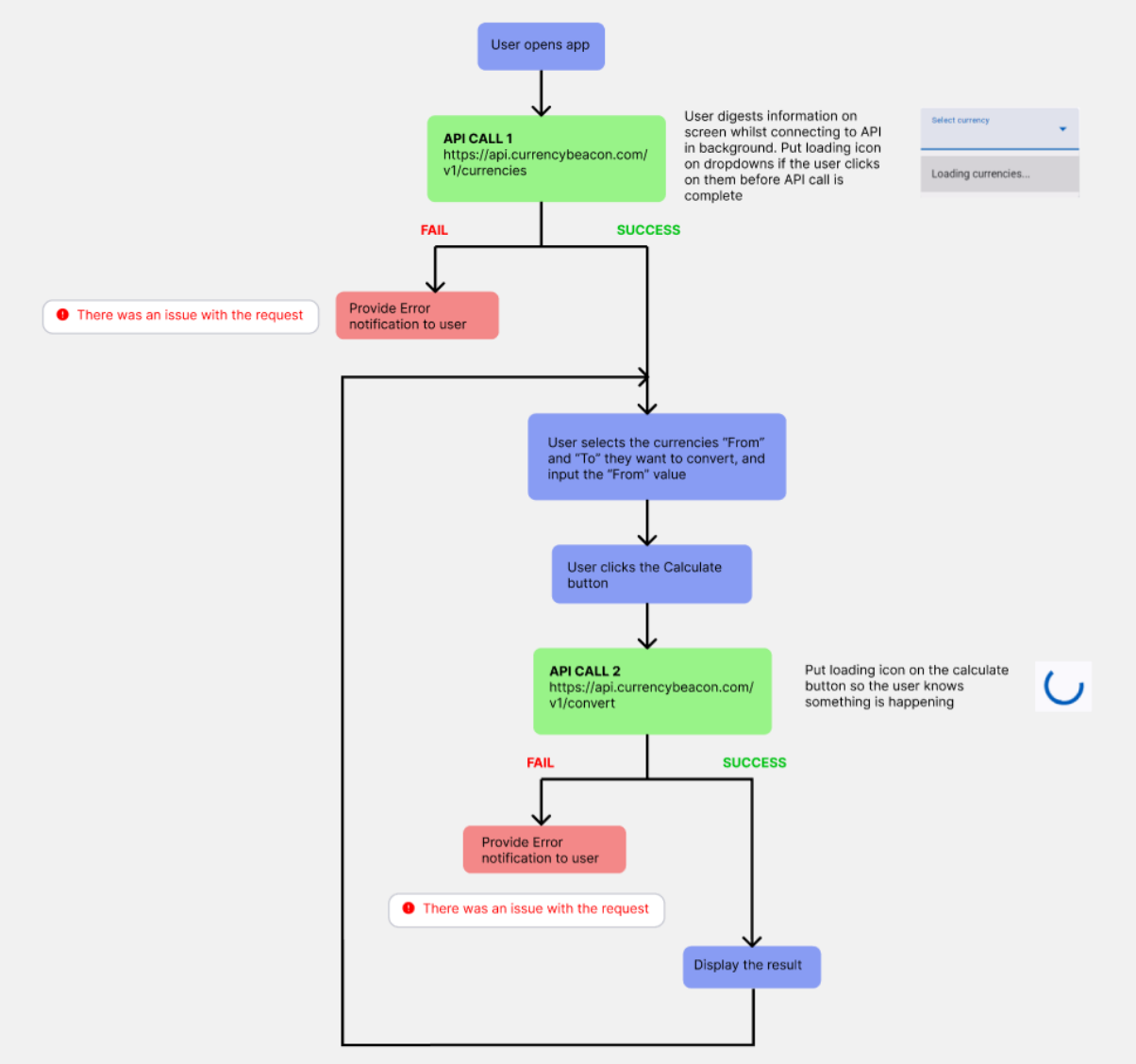
I used the library Angular Material as this has plenty of common components and had what I needed to quickly build what was required of me. I found that there was a slight peer dependency mismatch with one of the packages `@angular/animations` (as they hadn't caught up yet), but for the purpose of this app I decided to force install it and skip the npm checks and it worked fine.

After setting up the local repository and connecting it to my github account, I decided that before I actually start coding I need to properly understand the problem before I get stuck in so I carefully re-read the description and created a quick design to make sure I was making what was required of me. Here is a Figma design I sketched up:



The requirements were to keep it simple and have a basic design. As no colour scheme was mentioned, I decided to just use the standard colour scheme of Angular Material to make things easy. The components in the image were based off the designs of the components on the Angular Material website so I knew what they would look like once I implemented them.

After this I made a quick user flow just so I could understand the order in which the API calls needed to be made, and how the user was to interact with this design. I drafted up the following user flow on Figma:



Although the instructions did not specify anything about user handling, such as things like loading wheels or feedback about the API requests, I could see this was critically needed in my user flow. Again, to keep things simple and fast I decided I could provide all the user feedback I needed with one or two loading wheels and a notification popup. I did this due to ease of implementation (and there was no requirement), but if more time was permitted a proper use of form control would be better (where the inputs on the screen highlight red when the user makes a mistake ect).

Once I well and truly understood the task and the app I was to make, I began coding.

I wanted to demonstrate my understanding of re-usable components so I basically made everything on the screen a re-usable component, even though they were very basic. Using Angular Material significantly saved me time as the components did exactly what I wanted, so I created my individual components and created the “page”, which I just called “home” and there placed all my components together to build my app like the design.

Once I had constructed my HTML and CSS how I wanted, I started implementing the APIs in which I decided to create as a service (to show my knowledge of this). The APIs could be called

and “subscribed” to anywhere in my app, and made it easy to fit it into my “home” page. There was no need to un-subscribe from this service because the user never leaves this page.

Based upon my user flow, I knew I had to load the first API as soon as the page was rendered so that the dropdowns could be populated, and then the second API when the “Calculate” button was clicked and the user had to wait for the response. It was fairly easy to implement this and then it was just a case of plugging up all my components and tidying up a bit. I wrote a type / interface for everything to keep it robust and tried to make my code as readable as possible with plenty of comments where appropriate.

Angular has a very useful built-in unit test functionality “Karma” and “Jasmine” which make it very easy to write quick unit tests. Every generated component in Angular comes with a spec.ts file and it comes with a pre-generated unit test to check if the component is rendering correctly, so it’s very easy to piggy-back off that and add a couple of tests to check that the HTML and functions are working as they should.

If I had some extra time I could have made the CSS more responsive and create a mobile view, but as per the instructions I decided not to spend too much time on this app and keep it simple.

Thank you for taking the time to read this, I sincerely hope we get the opportunity to meet over an interview.