For milestone 2, we’ve decided to throw away the existing prototype of the application that we built for milestone 1, and create a new prototype. There were several reasons we decided to make this change to our application. Primarily we made the change because our code wasn’t structured to operate in an asynchronous manner, the language and tools we used are not well suited to an app that utilizes a GUI, and our chosen language doesn’t handle various programming concepts very well; such as interfaces and classes.

The first reason we decided this was a necessary change was that we built the initial prototype to be a console application, therefore the code was written in such a way that blocks of code would run more or less sequentially without allowing for other behaviors of the system. This code structure could not fit with a program that presented a basic GUI to the user, as the GUI basically needs to wait for actions from the user before doing anything, rather than immediately going through the basic code loop. In order to add a GUI to the application we had to make the code operate in a more asynchronous manner, as well as restructure how parts of the data would be accessed in order to display to the user.

Another reason we decided to throw away the existing prototype and create a new one, was the language that we chose to make the initial prototype with. We created the initial prototype using NodeJs and Javascript, and while these are good languages to code in, they lend themselves much better to a server application without a GUI than they do an application with a GUI. Therefore we are going to convert the project to a Typescript project using the Angular framework instead, as these tools make it incredibly easy to create a functional GUI for the user to utilize.

Finally, we chose to throw away the old prototype and create a new prototype because the language we initially chose for the first milestone, Javascript, is not well suited for many of the programming concepts required in this milestone that make the code more flexible and extensible. For example, Javascript can handle concepts such as interfaces and classes, but it doesn’t handle them well, and trying to work those concepts in can be difficult. As a result we decided to transition the project over to a Typescript project instead. Typescript is a superset of the Javascript language, meaning that it works and runs just like Javascript, but it allows the programmer to write code in a strongly typed environment and utilize classes, interfaces, and inheritance much easier than if we were using plain Javascript.

After carefully considering our options, we decided that the best course of action to deliver on the requirements set out by milestone 2, was to throw away our initial prototype and create an entirely new prototype. Our new prototype utilizes a different language and framework tools in order to run the project and display a user-friendly interface in a browser window.