Good afternoon. I’m Stephen Wang. I’m a software engineering intern here at smiths detection, and today I will be talking about the projects that I have worked on during my internship.

First, I’m going to talk a little bit about my background. I’m a computer science and engineering major at UC Davis, and this fall I will be returning to Davis for the senior year of my undergrad education. In college I became fluent in many programming languages, like C, C++, and Java, but I never programmed in JavaScript before, and that’s the language I had to learn and use during my internship. Learning JavaScript was a bit challenging for me at first because, unlike the languages I already know, JavaScript is an asynchronous programming language.

In traditional synchronous programming languages like C++, code is executed synchronously, or in order. For example, in this C++ code snippet, we have a program that starts at the main function, and prints out One, Two, and three in order, from top to bottom. This is synchronous programming. In this next example we have a snippet of asynchronous JavaScript code, which also prints out one, two, and three. However, in this example the code isn’t executed from top to bottom like in C++. Instead, the code is executed based off of user input. For example, the functions one, two, and three can be mapped to keys on a keyboard, and the order that one, two, and three are printed, depends on the order that the user pushes the keys on the keyboard.

This user/event driven property of JavaScript was very useful for the type of projects that I worked on during my internship. For example, in my first project I created a Software Update Application using JavaScript and HTML. This application listens for a connection to a URL, and when a connection is made, it posts Software to the URL so that the user who made that connection can download it. As you can see, JavaScript is perfect for this type of application because it is driven by user input, where the user input is someone connecting to the URL.

Here is my finished application. When you open the application, you are asked to set a path to the directory that contains your installation files. In my case, I have a folder called releases in my current folder that contains my files. After that, you click on set up server. As you can see, the app is exhibiting asynchronous programming properties right now by listening to a URL and waiting for a user input. Now that our app is waiting for a response, we can test to see if it works by providing user input by visiting that URL. On a second computer, I wrote a python script that tests to see if the app works. As you can see, the script asks for an ip address, so I used the ip address that the app is listening to. After that, the script calls wget with the ip address to download a file from the releases folder of the first computer, and as you can see, it successfully does that here.

So after I finished this project, I worked on another JavaScript application, which is a publishing filter. In the software engineering department here at smiths, whenever there is a software release, engineers copy software package managers, or RPMS, to a server. My job was to create an application that blocks certain rpms from being copied to the server.

What it does is, it takes a name and version from a user, enters that information in to a database of banned names and versions, and when a user tries to publish a file, if the files name and version matches ones in the database, that file is filtered out.

Here is the application in action.