COSC345 Assignment 3 OhDamnGitUp Levi Faid Duncan Cowie Caleb Mitchell Zac Gardner

In the first stage of showing off our prototype we were met with some well deserved critique. Our prototype was a bit ugly, had some major errors, and generally wasn't that flash. Our code was janky, our ability to run without an active home style wifi network limited and our local site average looking. We have taken these into account and given our top priorities for completion of the production of our device. Our home testing was limited due to us only having one “working” prototype but we did trial it as much as we could. The results were not astounding but as they say if at first you fail try and try again.

Below I will discuss in short the basics of what we are going to do to fix most of our issues.

Offline mode

Currently the clock essentially requires to be always online. Until I started actually using the clock I didn't realise this might be an issue, which it turns out is a pretty big one. It's not so much that it gets messed up if the wireless goes dark, but more that it's frustrating to have to think about it at all, a clock really shouldn't need the internet even if it's going to work better and tell time more accurately when it's online. Switching the functionality to run in offline mode should be a simple change to the way the device operates and an update to the site it hosts.

easier to connect

In its current state the clock can be accessed on most devices through a web browser by typing "gitup.local" into the address bar, the problem being that this works for only most devices. While this did work for my devices in trials and proved to be no issue, it would still be much more beneficial to make this system work for all devices rather than having to resort to typing the devices IP in the address bar. This may come in the form of a request to install Bonjor services which contains the required drivers for local DNS redirecting.

Prettier

Our initial prototype was ugly, but we can fix that. With an entirely new system working together like never before the aesthetics of the device will be unparalleled in the world of IoT standalone alarm clocks. Our big changes will be having the device fully contained in a suitable manor

Wokerier

Our alarm clock will now work better than ever with a reupholstering of the algorithms that makes it 'click'. I'm unsure if I will modify it to run with cron or to continue being it's own standalone thing. Both have their benefits. I'm siding more on not using cron and just rewriting the entire program to be faster more efficient and work less in a less hacked together sense. My reasoning for not using cron is simple, this isn't a hardware project and if we choose to use cron that takes away a fairly large amount of the actual writing of software. In the real world this would almost certainly not be the way I would solve the problem but in this instance I'm sure you can understand why I would rather write something myself.

Multistage prototyping

We will be making two entire set ups this time around the block, with one using a more custom set-up having the clock face, the amplifier and the casing all as hand made as reasonable and another with a pre-built all in one mini mono amp, a premade I2C LCD screen and an old school style box speaker as a casing.

As well as these upgrades we will be reassessing our testing procedures. Currently most of our testing was done under a controlled environment. Testing on the fly as we modified the code and the hardware. We didn't do anywhere near enough on real world tests, and will be fixing that. Caleb has agreed to run the clock 24/7 for the second term, with weekly meeting to work on issues that may have developed as we go. Levi has also volunteered using the secondary prototype. This will allow us to have a far better level of quality assurance in our final exemplary product.