

NEA PROJECT

COMPUTER SCIENCE

Contents

[Analysis 2](#_Toc54422449)

[Project Problem Definition 2](#_Toc54422450)

[Interview with End User/Client 2](#_Toc54422451)

[Observations on Existing Systems: 3](#_Toc54422452)

[Windows Built-In “Defender”: 3](#_Toc54422453)

[McAfee: 5](#_Toc54422454)

[Prospective Users and Acceptable Limitations 7](#_Toc54422455)

[Design 8](#_Toc54422456)

[Core Elements of the System: 8](#_Toc54422457)

[UML Deployment Diagram 8](#_Toc54422458)

[The API 10](#_Toc54422459)

[The CLIENT 11](#_Toc54422460)

[The GUI 12](#_Toc54422461)

[Package Directory Tree: 12](#_Toc54422462)

# Analysis

## Project Problem Definition

Client/End User: Joanna

Description of the problem:

After using many antiviruses and speaking to a relative who often has to download and open files from clients. It became apparent that modern antiviruses are very intrusive and hardware intensive. Often slowing down the OS for what often equates to a safe file. Ultimately slowing down a user’s work flow, and wasting time.

Additionally, I find antiviruses never show complications of installing the file. Only saying weather, it is dangerous or not based on the algorithm made.

Finally, linking to the first discovered issue. The fact they are slow, creates a very negative user experience for those with less expensive and powerful devices. Especially laptops. For example, I used to run a windows vista desktop. I disabled the McAfee antivirus because it was so slow it made the system unusable for common web browsing and playing simple games.

Later in this analysis I hope to benchmark antiviruses and see how they favour expensive devices. And overall user experience.

## Interview with End User/Client

## Observations on Existing Systems:

### Windows Built-In “Defender”:

|  |  |
| --- | --- |
| Pricing | Insurance |
| Free | None |

|  |  |
| --- | --- |
| PROS | CONS |
| Real-Time Detection | Real-Time Detection to slow for modern viruses |
| Firewall | Non user friendly GUI. Hard to navigate. |
| Parental Controls | No Money back |
| Uses DB for previous viruses | No Adware Prevention |
| Completely free for windows | Doesn’t have profiles for high performance |
| Ransomware Protection | No Support |
| Manual Scanning |  |
| USB Scans |  |
| Can do Registry Scans on Start-up |  |

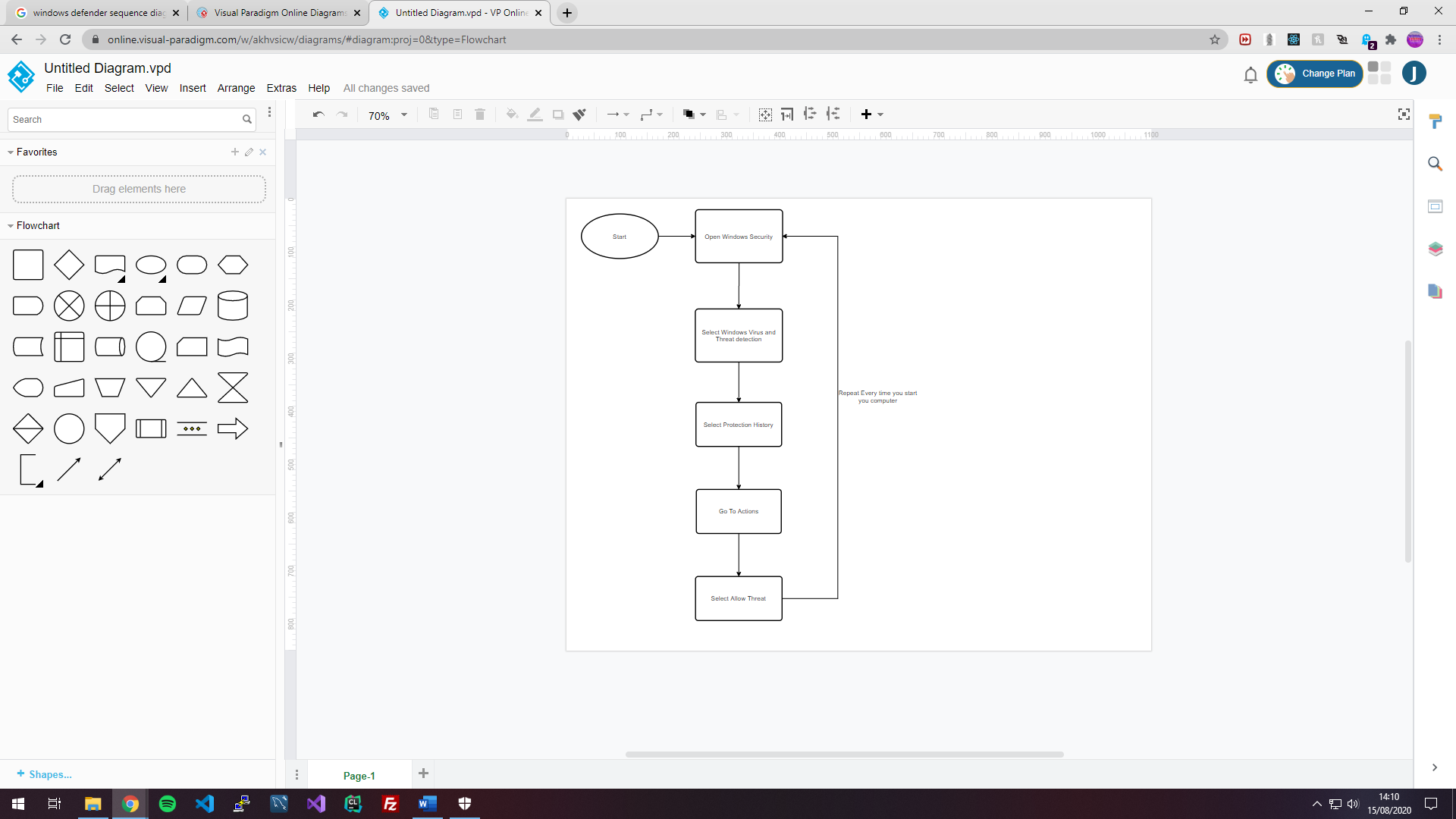
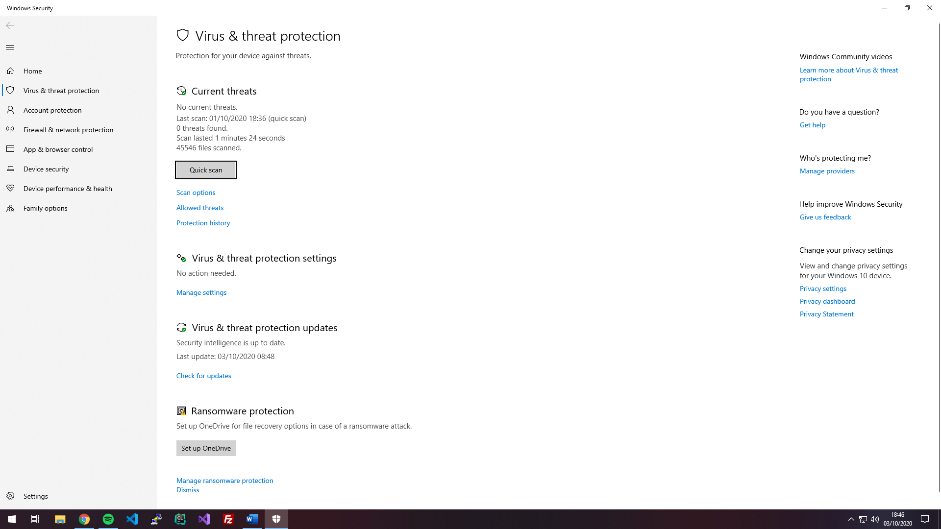
The Pros Elaborated:

It is free and comes with windows 10, for those not bothered by the speed of their device and simply use it for day to day usage, this antivirus is great. It also allows for more advanced features like registry scans, which allows for detection of works and viruses well hidden.

The Cons Elaborated:

Firstly, the GUI is very frustrating to navigate through. When the system detects a virus, it will often remove it and make it hard to recover. And when added to the allowed list of “viruses” it will continue to remove It periodically. While the flow chart seems simple, the reason it is frustrating is because these functions are nested in an application outside of windows defender. Making it hard to locate.

I find that the lack of colours makes it very hard to know what is important, it makes the whole experience less intuitive and more time looing around trying to see what’s what and how to get to a desired button. There is a help section which opens your browser with instructions on how to use it. However, a tutorial on how to use an antivirus isn’t acceptable.



Conclusion of Application:

Windows defender is free. Which means it doesn’t come with guarantees. Additionally, although it comes by default with windows making you device automatically secure and safe, helping those daily internet surfers stay safe. However, it only works for windows devices. Making it non universal, leaving phone users and apple devices an open market.

Also, the GUI is horrible. Following windows design choices in windows 10. Unifying all windows program applications and making them minimalist hasn’t worked. It creates a clear but very cluttered look. Where colours and bold font aren’t used to make some buttons and headers stand out. I don’t know where I’m supposed to look.

I feel windows “Defender” tries to come off as an all in one solution, streamlining into windows products. But its solution is only half done and needs work on.

Improvements I Should Include in My Application:

* An intuitive GUI which neatly organises all the features in a way which is easy to uncover and find.
* A faster and more lightweight solution for Realtime file scanning.
* Strict blacklist for allowed files. If a user says don’t’ delete a file, don’t delete it!

### McAfee:

|  |  |
| --- | --- |
| Pricing | Insurance |
| Costs Money | None |
|  |  |

|  |  |
| --- | --- |
| PROS | CONS |
| Very good anti phishing | Full scans take > 1 hour |
| Nice intuitive GUI showing only what’s needed | Cost’s Money, Tiers of quality |
| Will turn off when playing games |  |
| Can do manual Scans |  |
| Can do automatic Scans |  |
|  |  |
|  |  |

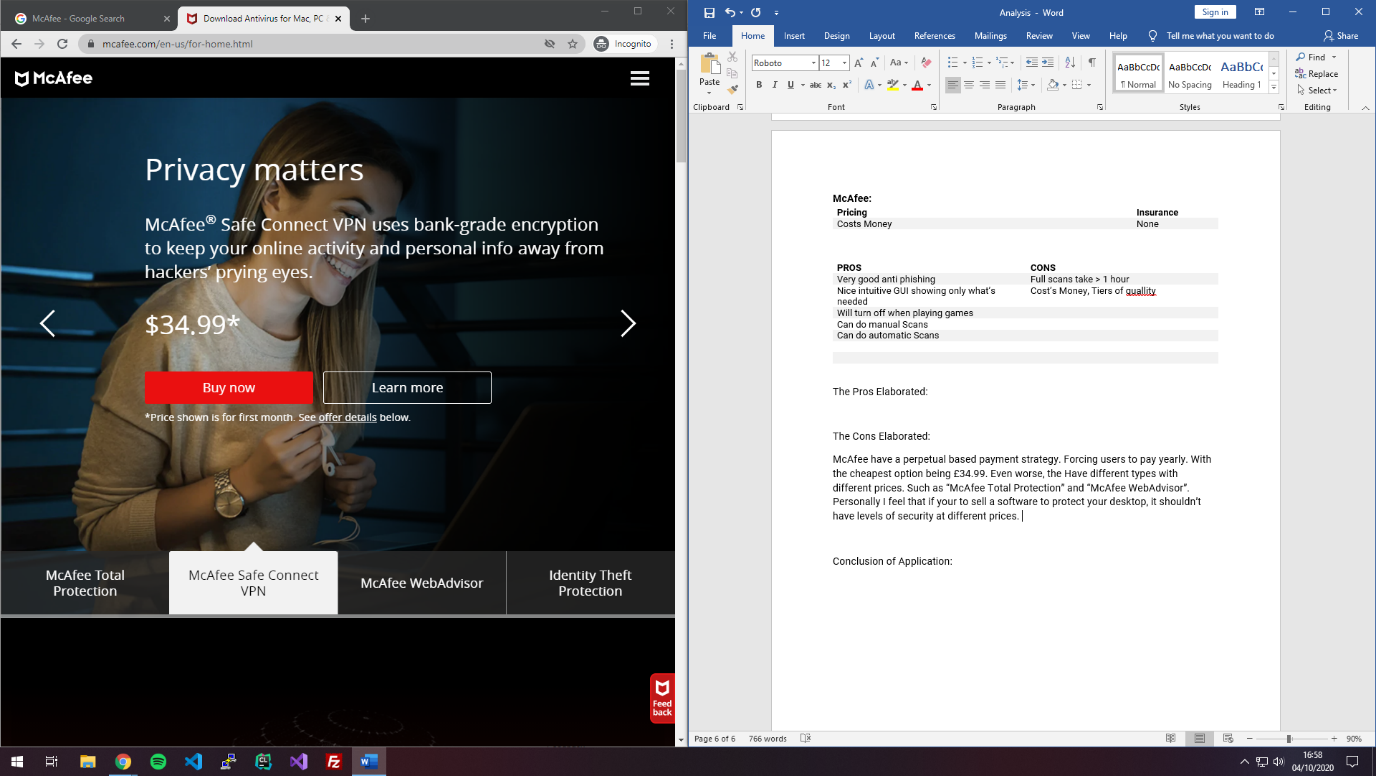
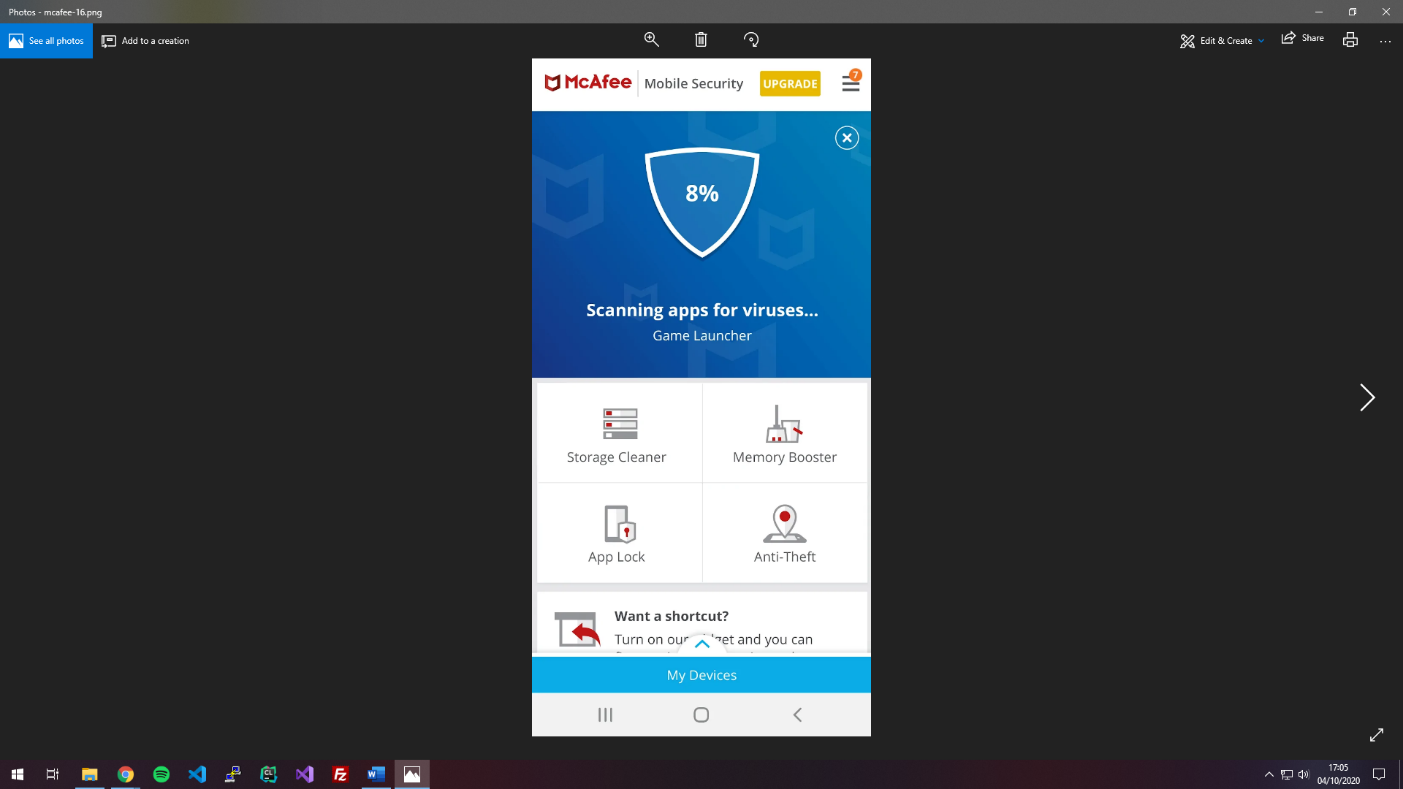
The Pros Elaborated:

It has a strong reputation for stopping zero-day attacks, with a large team constantly updating the software. It has useful utility features like disk cleaner which will free up space on your desktop.

They have a mobile app which makes it the first candidate which complete platform protection. Even more it is included in most of their payed plans, making it a neat addon.

The Cons Elaborated:

McAfee have a perpetual based payment strategy. Forcing users to pay yearly. With the cheapest option being £34.99. Even worse, the Have different types with different prices. Such as “McAfee Total Protection” and “McAfee WebAdvisor”. Personally, I feel that if you’re to sell a software to protect your desktop, it shouldn’t have levels of security at different prices. Additionally, when trying this software and reported reviews, it often scans large files such as CAD drawings, unnecessarily slowing down your device, making it a huge issue for creative artist professionals, such as my client.



Conclusion of Application:

Overall, McAfee offers one of the best antiviruses around and have been in the business for a long time making them a reputable company. However, with the added value, they have increased the pricing and made it subscription-based. Always making income. You could argue that this helps the developers to have funding for better algorithms. Their app has lots of popups constantly getting in the way and I would like to see an application stealthier.

Improvements I Should Include in My Application:

* Make it only scan new files and if a file is modified and large, only scan when user is not using hardware intensive application like AutoCAD or Photoshop…
* Have one tier not loads, having to pay for better security is unacceptable.
* Make it free, many shy away from McAfee because of its egregious pricing.
* Try to run in background and only notify user when it is something urgently wrong.

## Prospective Users and Acceptable Limitations

As illustrated, the focus of this antivirus ranges from creative professionals (End User/Client) to users surfing the internet. Given such a wide range of users some features may be bothersome while not others. Some limitations are to be expected. To partially mitigate this the user of modular features will greatly help.

The limitations of the system (antivirus) are illustrated below:

* It won’t be a very strict antivirus, as this is the main issue discovered with many antiviruses sold these days.
* Time Constraints: the system must be created before easter.
* Each user’s computer has a different computational power, meaning that it mustn’t be to intensive (previously discovered issue).
* My Computer/Development: creating and training an artificial intelligence is very time consuming and as my computer isn’t very powerful it will take a lot of time. (not for end user)

# Design

## Core Elements of the System:

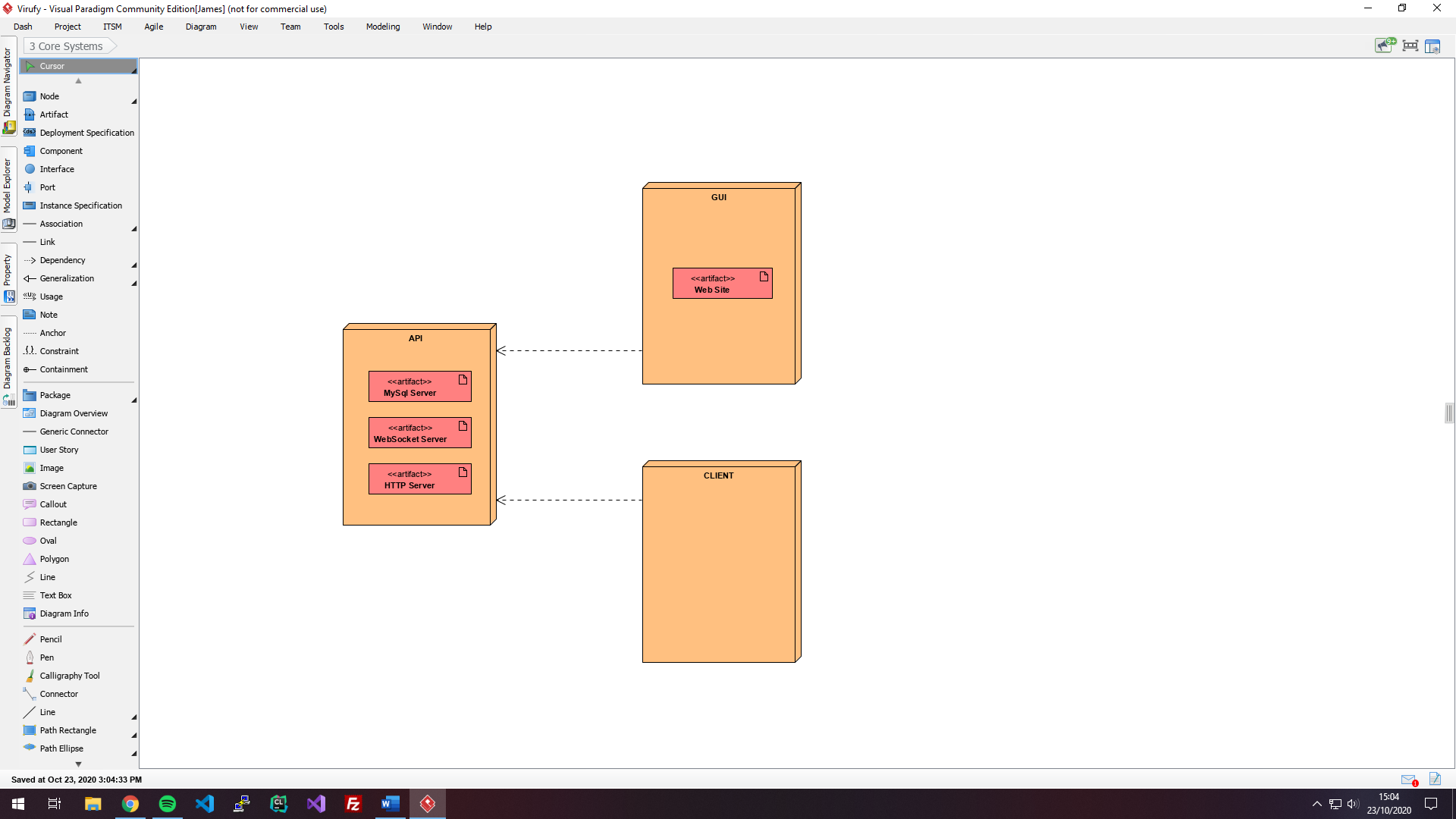
The system will be a **full** finished product and as so will require many different services. The three core elements are as listed.

* The **API** which will need to be run on the **cloud**. API
* The **client software** which will run on the **computer**. CLIENT
* A **GUI** which will be ran on the **cloud** as **a progressive web application** GUI

Given such a large project each area will be addressed in different packages to streamline development.

The three elements communicate at a high level like this:

### UML Deployment Diagram

The deployment diagram below is following the **MVC** (model-view-controller) design pattern. As it forces modulatory which will help future changes and feature implementations if my end client needs more features.

The CLIENT and GUI never connect to each other. This makes sure that other users cannot connect to someone else’s computer. This increases security.

* **Model**: Structures your data in a reliable form and prepares it based on controller’s instructions. - API
* **View**: Displays data to user in easy-to-understand format, based on the user’s actions. - GUI
* **Controller**: Takes in user commands, sends commands to the model for data updates, sends instructions to view to update interface. - CLIENT

### The API

cloud\_app:

Routing Algorithm Required for possible commands sent to websocket:

~~Create two vectors, one with the command names and one with the pointers to the actual function. For v1(i) = “command name” where v stands for vector. v2(i) = \*function.~~ Doesn’t work because a vector cannot hold a pointer but must copy the actual function. Which will lead to huge vectors with a single point of error.

### The CLIENT

### The GUI

Requirement’s from research:

* Must have an easy to read and intuitive GUI
* User colours to illustrate what buttons are more important
* Doesn’t constantly popup in the users face as it is very distracting and useless

Based on the simple requirements, I’ve decided to run the GUI and user interface on the cloud as a website; more precisely a progressive web application. This approach will make the application very light and use little to no system resources. It also means a less intrusive update system and it can be updated centrally instead of updating each user’s software on their computer.

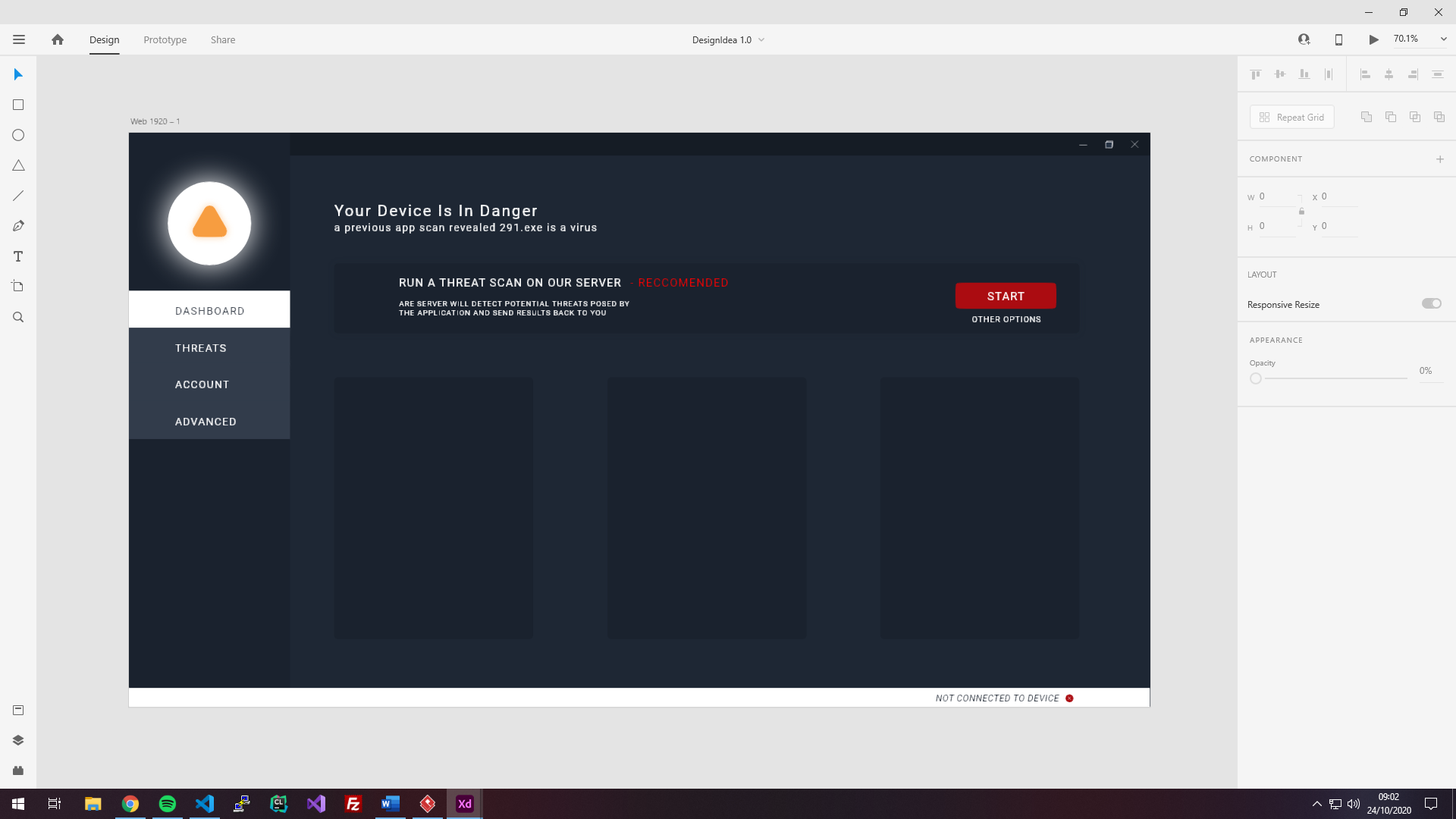
As it is on the cloud it also means that it will only appear when they go online so the intrusive popups won’t exist.

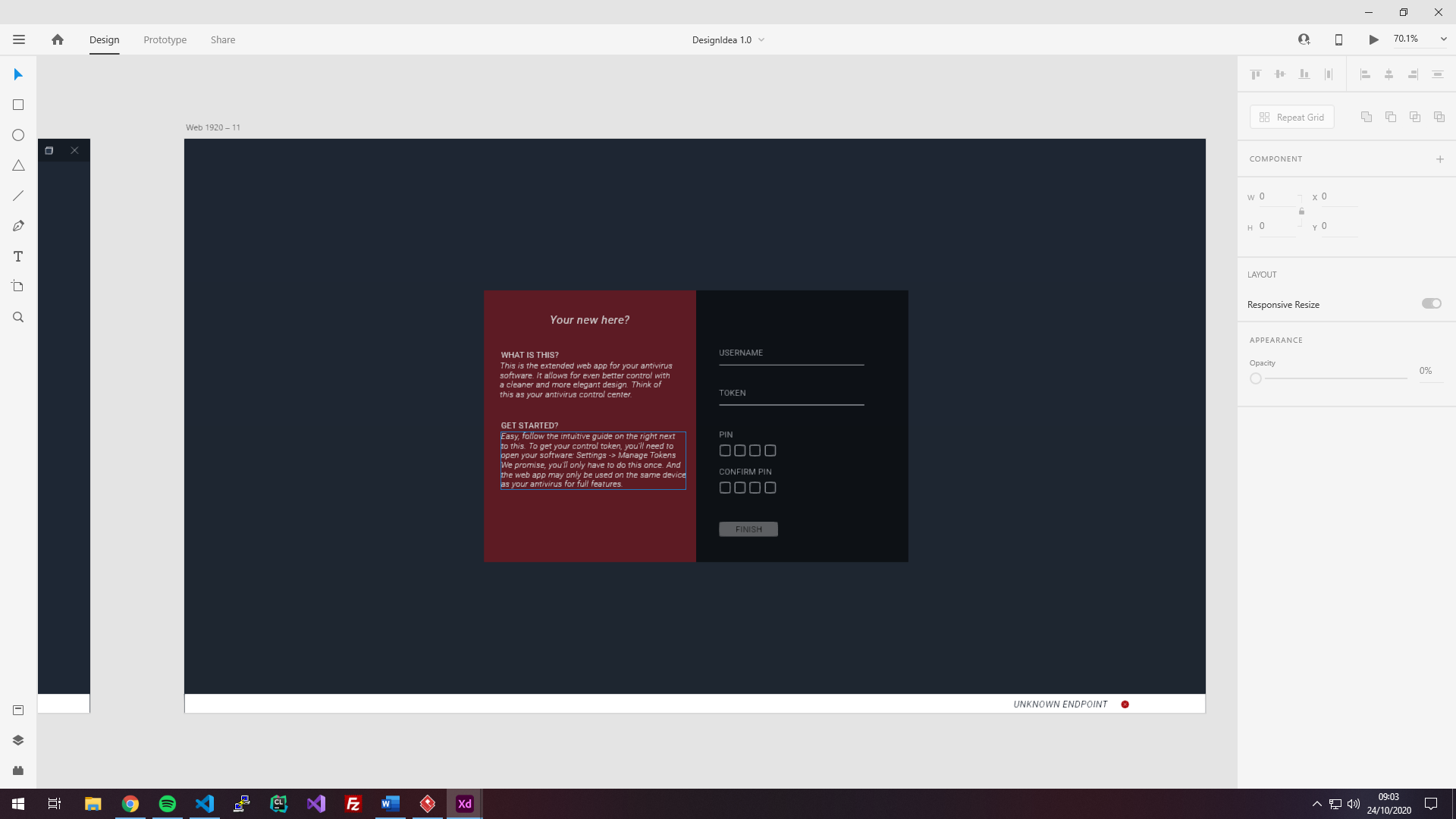
Programming language:

I have decided to use Node.js running on chromes v8 engine. It is fast and allows you to deploy web applications quickly which lots of flexibility. In Addition, I will user React.js framework for direct DOM manipulation.

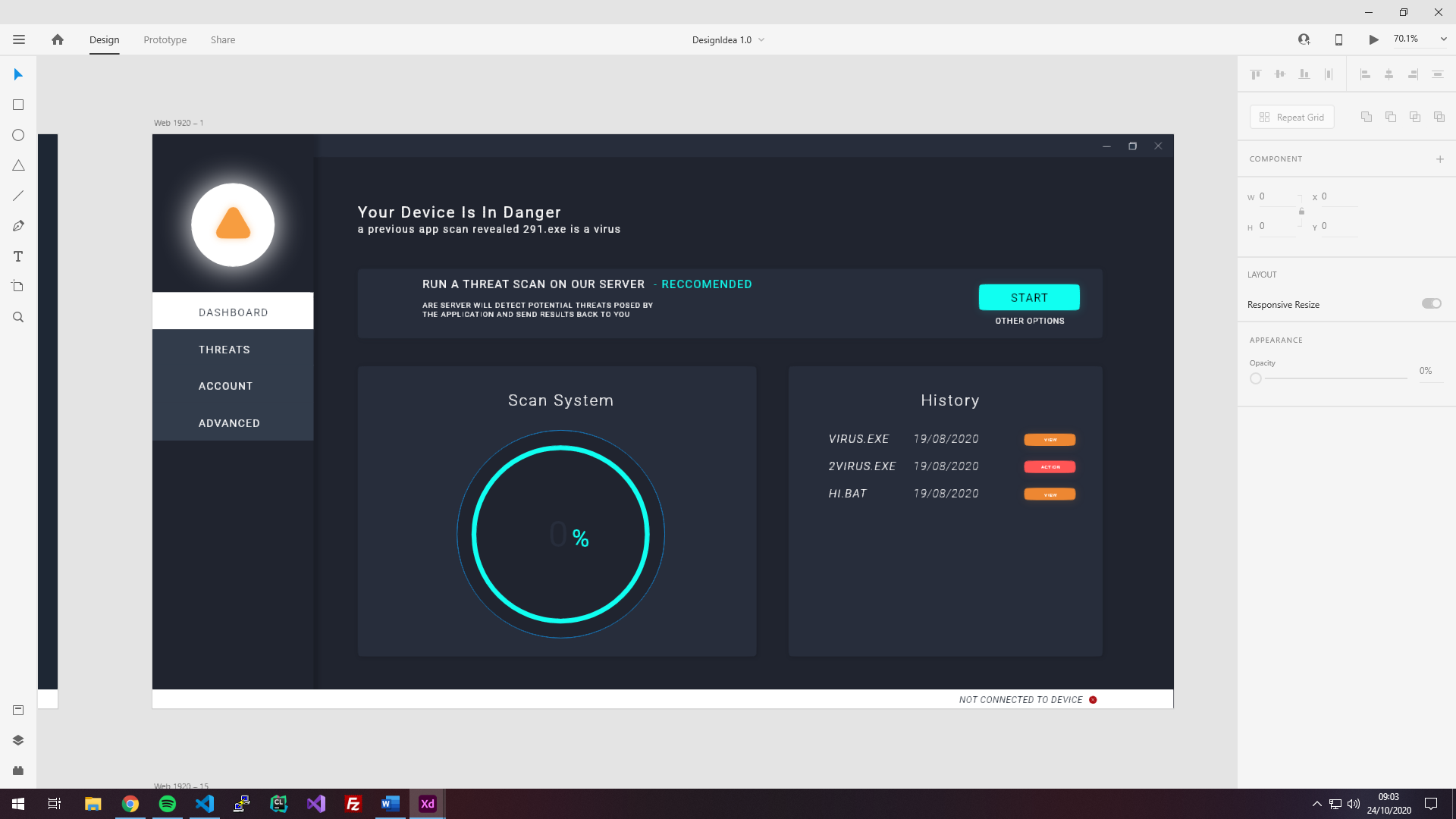
### Package Directory Tree:

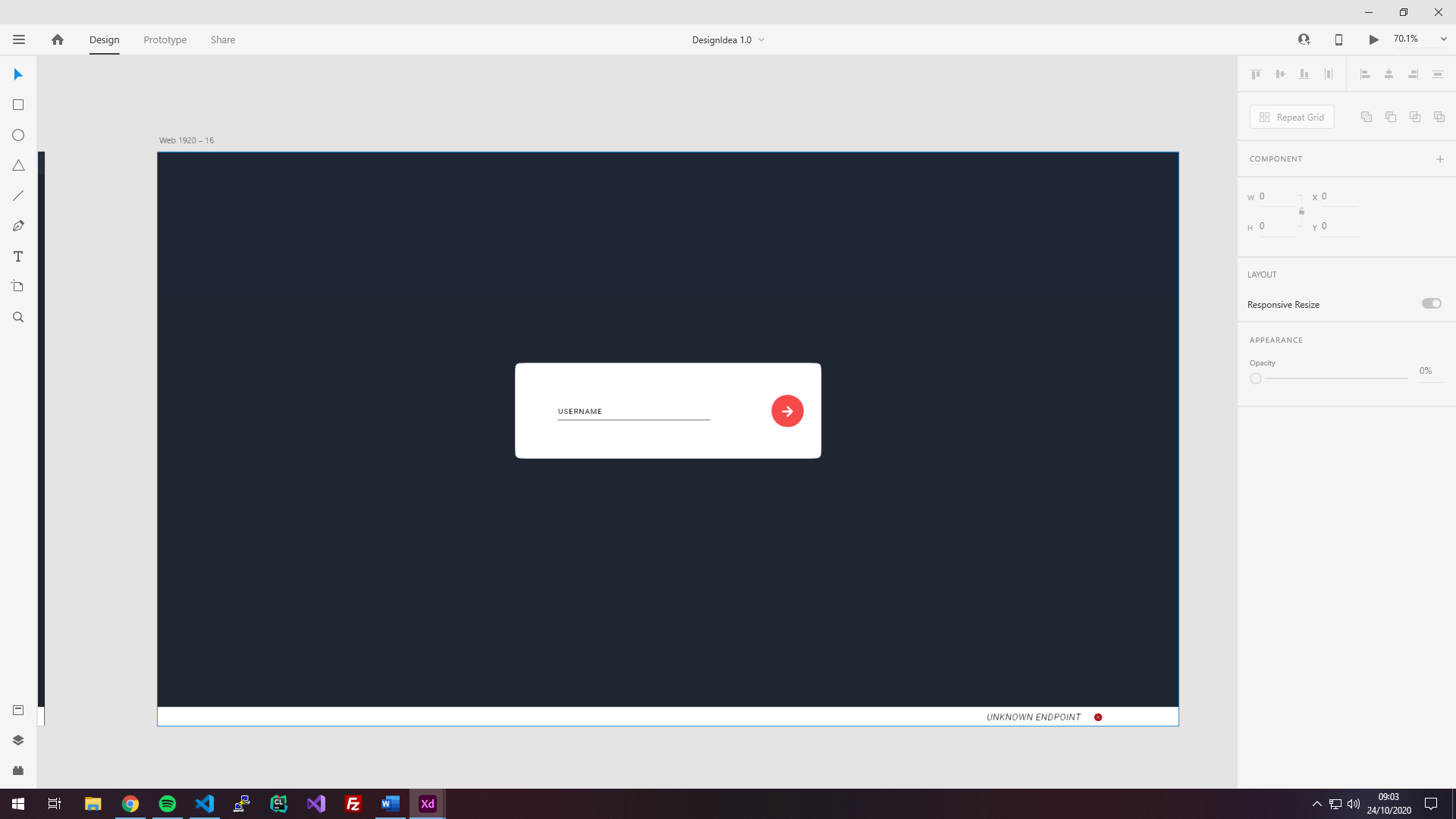
### Design Concept #1:





### Design Concept #2:





PLEASE IGNORE: this data is not currently formatted

AI Data Preparation (how and what will the input neurons receive):

Number ai-> takes std::vector<int> for input and target.

Ai will run through the neural network and use multiplane differentiation to achieve a value high to the target.

Debugging the AI:

Biases aren’t working or assigned wrong:

Weights aren’t between default parameters 0 =< x =< 1 following normal distribution:

The weights are between 0 and 1 so WORKING

Feedfoward function not working: WORKING

setError\_MSE function not working: WORKING ONLY FOR {x, x, 1}

back\_propagation function not working:

The neurons attempt forward propagate may be incorrect.

FOUND: sigmoid function wasn’t written properly (1/1-e^-x). FIXED

Forward propagation is functioning properly with {1,1,1} topology. More complex topology’s have not been tested as manually doing the equation is very time consuming.

Biases have been excluded for now (set to 0)

step by step how system works:

1. Main.cpp calls for initiation of NeuralNetwork.cpp (\*NN) in pointer form
2. NN make num of layers then the weights/matrices
3. NN then makes a vector for the error function later

Issues detected:

Random biases generated in the Matrix.cpp users Misc::math:generate\_val() has no limits so it will generate a number outside of the normal distribution. (Slows down training and even breaks NN)

Notes:

When defining the topology for NN {1, 3, 2} the first “1” is the input layer and the “2” is the output layer.

Stages to back propagation:

Firstly the error margin must be calculated: for a basic neural network it is defined as: *target – calculated =output sum margin of error.*

*(target – calculated) ^2= output sum margin of error*;

Secondly, the delta output sum must be calculated: dsum/dresult \* output sum margin of error = change in sum.

dsum/dresult = f’(x) where f’(x) = *x* \* (1 - x);

Problems with Implementing an antivirus:

Firstly, virus files could be 1gb or 1mb. How many input layers and what will the input layers contain? Scrambled lines of code. ***Research into neural networking on genetic sequences as it is very similar***.

A better problem definition: how can I take a large string of words (aka code) and feed it into a neural network as a feature vector: <https://datascience.stackexchange.com/questions/869/neural-network-parse-string-data>

Common technique is called Topic Modelling, using the LDA neural network design

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.1770&rep=rep1&type=pdf>

Reducing noise and other pre-processing processes are required.

Issue: there is a bias for the whole neural network. A bias should exist for each neuron. Making a vector of biases for each layer and a matrix of biases corresponding to each neuron. FIXED. A vector of vectors has been created for the biases

The cost function has now been created to allow multiple output neurons. It takes a vector if targets in the data type double.

Removed a random function in NeuralNetwork.hpp called this->error which was the data type double.

The back propagation will be semi modular, running the last hidden layer to output neurons through a fixed function not using for loops as much.

