

NEA PROJECT

COMPUTER SCIENCE

Contents

[Analysis 2](#_Toc58064157)

[Project Problem Definition 2](#_Toc58064158)

[Interview with End-User/Client 2](#_Toc58064159)

[Observations on Existing Systems: 3](#_Toc58064160)

[Windows Built-In “Defender”: 3](#_Toc58064161)

[McAfee: 5](#_Toc58064162)

[Prospective Users and Acceptable Limitations 8](#_Toc58064163)

[Project Level Objectives: 8](#_Toc58064164)

[Design 9](#_Toc58064165)

[Core Elements of the System: 9](#_Toc58064166)

[UML Deployment Diagram 9](#_Toc58064167)

[The API 11](#_Toc58064168)

[The CLIENT 12](#_Toc58064169)

[The GUI 13](#_Toc58064170)

[Package Directory Tree: 13](#_Toc58064171)

[Design Concept #1: 14](#_Toc58064172)

[Design Concept #2: 14](#_Toc58064173)

# 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 workflow, 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.

## Interview with End-User/Client

Me: *‘so what’s the current issue with your use of antiviruses?’*

Joanna: *’as a creative art professional I find when my clients send mock-ups and other related files my antivirus will often remove them as potential threats. ‘*

Me: *‘have you considered turning it off.’*

Joanna: *‘I have had malware and specifically adware manifest into my computer before, so the use of an antivirus is crucial as malware significantly disrupts my workflow.’*

Me: *‘How is your experience with lessening the antivirus to be less aggressive?’*

Joanna: *’ it isn’t possible, the antivirus has a very convoluted interface, making it hard to even allow certain files. And where I can allow them, they are often deleted again a few days later by the antivirus.’*

Feedback taken from dialogue:

Her current antivirus is too **aggressive** in stopping malware, to the extent of removing safe files. The lack of a clean and intuitive **GUI** has proven difficult to moderate and allow threats.

## 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 |  |

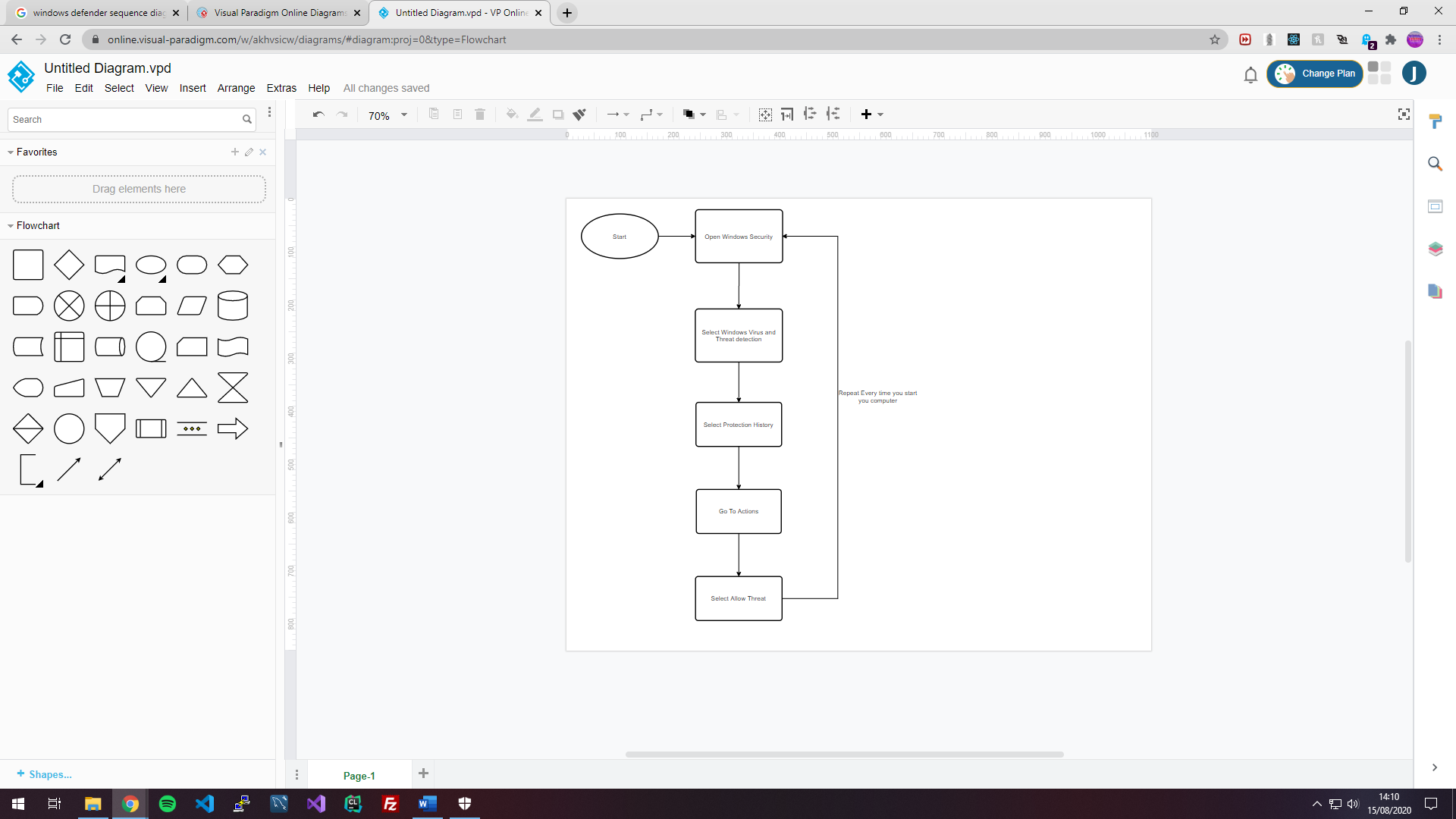
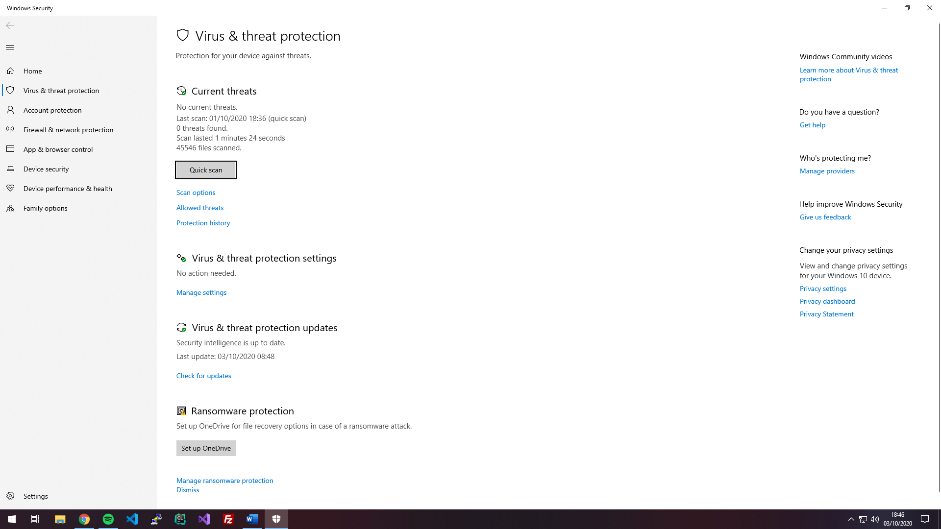
The Pros Elaborated:

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 |  |
|  |  |
|  |  |

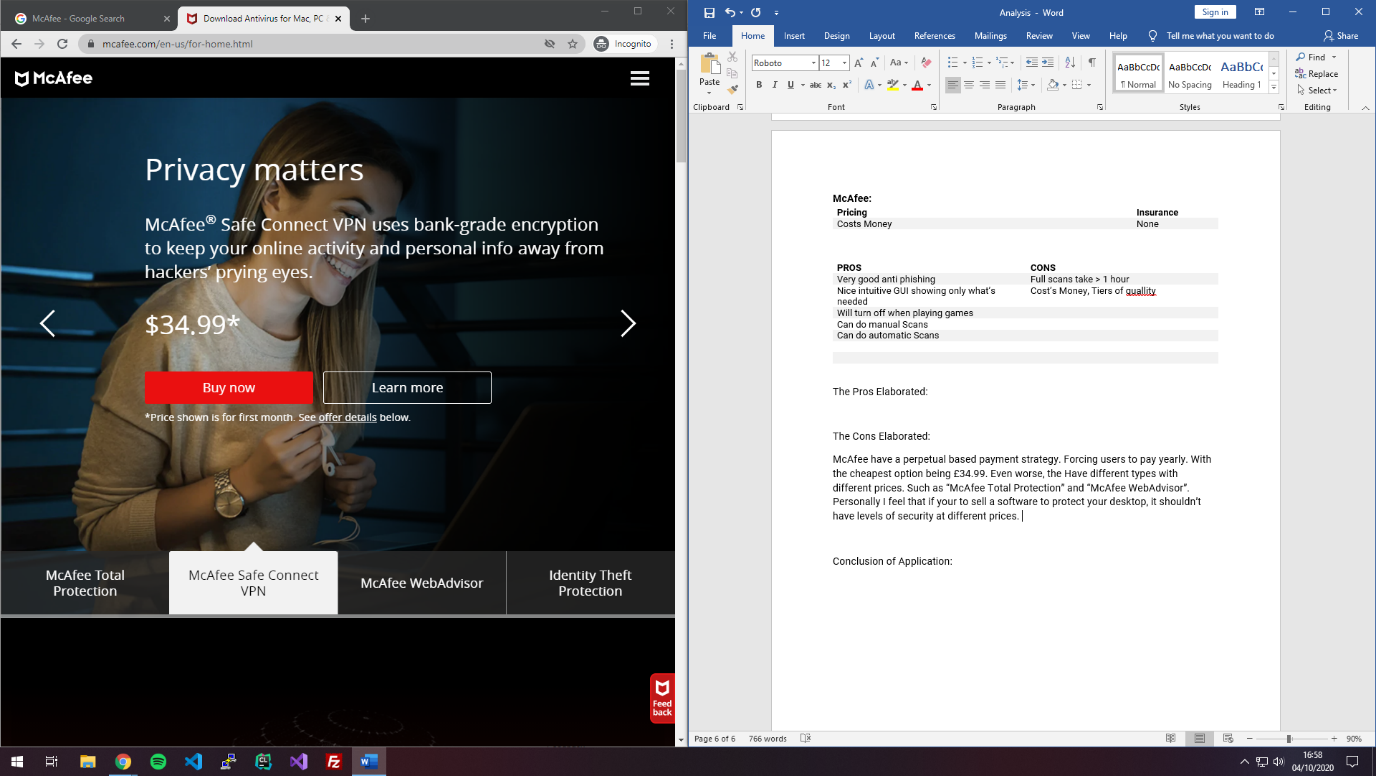
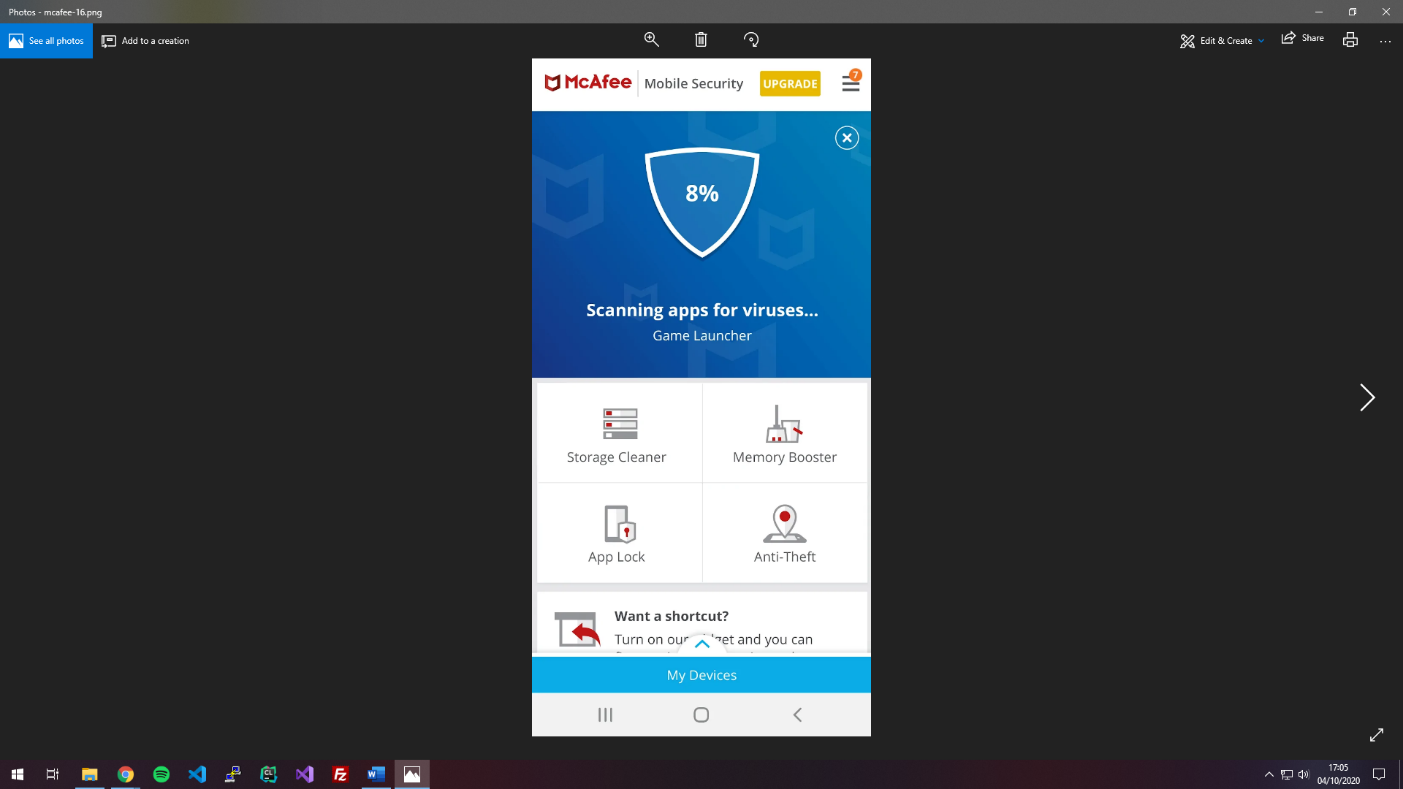
The Pros Elaborated:

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.

## Further Identification of User’s Needs

To enable me to chose and understand the most important requirement for the proposed new solution I conducted another interview with Joanna to gather what the ideal antivirus would be.

Me*: If you were to have the ideal antivirus, what would it do which sets it apart from other current ones?*

Joanna: *“Ideally, it would run in the background and only mess with my files after prompting for my permission. I dislike how so many will remove files immediately. Additionally, it would use little system usage, although I use a workstation for my work. McAfee still finds a way to spin up the fans.”*

Analysis of response: Joanna has easily addressed the major issues which I also share. 1. Too intrusive, leaving the user feeling out of control of the application. 2. Too intensive on the computer leaving you with a 100% usage on your hard drive. Leaving it feeling sluggish

Me: *Do you use an antivirus on your phone or other devices other than your workstation*.

Joanna: “*I don’t, my phone is an iPhone and I’ve never had any issues there. The only time where I would be exposed to viruses would be on my laptop. Where I don’t use an antivirus as it is just too much for my i3.”*

Me: As a professional creative art produces, what’s your opinion of antivirus GUIS?

Joanna: *“I greatly dislike them; you can tell it was designed by a programmer and not a ux/ui designer. They follow dark patterns, which are ones where finding common buttons and features such as quit membership or restore file are deliberately hidden.”*

Analysis of response: Joanna has evaluated that modern antiviruses are too confusing with a deliberately convoluted design.

After reviewing the dialogue, I have considered the following key objectives which would result in a superior antivirus:

1. The user must have full control of the application, where it is easy to navigate the gui
2. Must run in the background where it will use minimal resources to not ruin the users experience.
3. Have a more modern gui were it is intuitive for all and any users. Perhaps some gui questionaries and research should be conducted.

## 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 my 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 too 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)
* To train the antivirus will require large datasets of viruses, terabytes for a high efficient system

## Proposed Solution:

C++

Reason: For this I have decided on the core language being C++. This is because it is a middle level language, meaning I can optimally utilise the memory and processor better that counterparts such as java script while still having standard classes for data structures such as std::string unlike c or assembler. Because I am making an antivirus artificial intelligence speed is the most important element and C++ will easily satisfy this.

C++ additionally has a large community with libraries for JSON, MySQL, WebSocket/HTTP3 connectors. Which will greatly cut down on production times as they are MIT distribution licences allowing anyone to use them.

Disadvantages: The downside to using a lower-level language is that they offer so much freedom, you can easily write unsafe code which may create memory leaks or worse: vulnerabilities. I intent on additionally using C++ for an api, which is a potentially risky move but rewarding move, as the internet is very exploited. For example, the OpenSSL 1.0.1 heartbleed exploit. Where a malicious TLS heartbeat request could reveal some of the processors heap content.

Safety Considerations:

Pointers and register vulnerabilities:

- Given the requirement of fast applications. Instead of passing initiations as members of a class, direct pointers have been used. The gives the vulnerability

of possible DLL injection attacks. All function variables on the stack > 0xffffff layer must be signed to stop injections of malicious code.

- Additionally, templates are used which may expose the BBS runtime memory layer. Because Data and BBS layer segments by default of the CPU have read/write

permissions, instructions could possibly inject into these segments and executed

- Intel’s IA32 processor registers must forcefully point as follows: EIP->Text Layer, EBP, ESP->Stack Layer. Given these guidelines, manipulating the

runtime memory layout would be harder during the application initialisation.

- Function Pointers->SEH pointer (structured exception handler) can be overwritten as it points to accessible memory runtime layers.

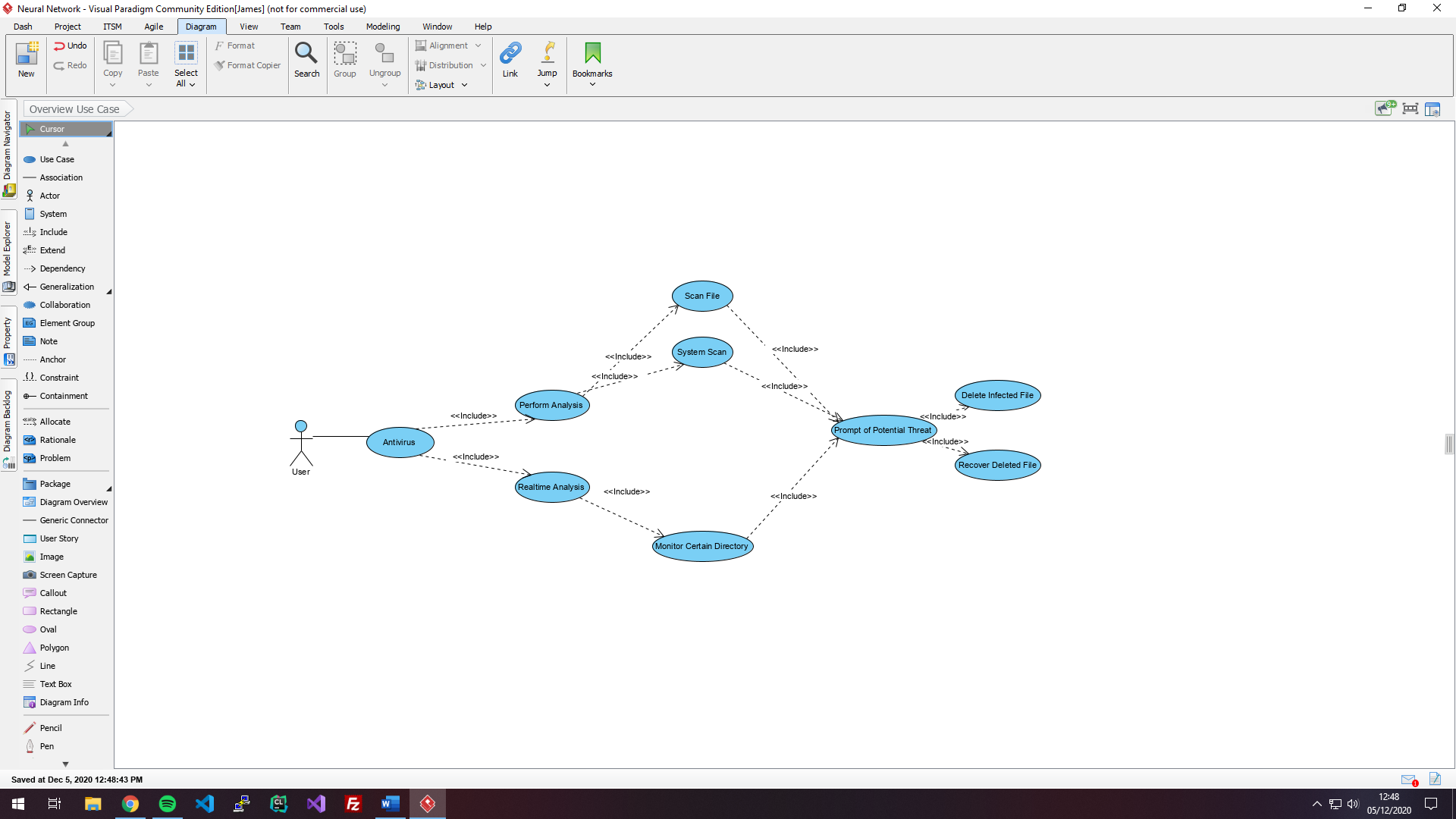
- Don't store data in global pointers due to easy manipulation

Securing the sockets:

- OpenSSL TLS heartbeat extension info leak. Possible heap over-read, which can result in server memory revealed by client as a malformed heartbeat is sent.

1.0.1->1.0.1f

## Project Level Objectives:

* Intuitive GUI that’s clean and present on relevant data
* Not overly aggressive in detective and removing threats
* Should be free so everyone can use it
* I will make an AI antivirus.

# Design

## Core Elements of the System:

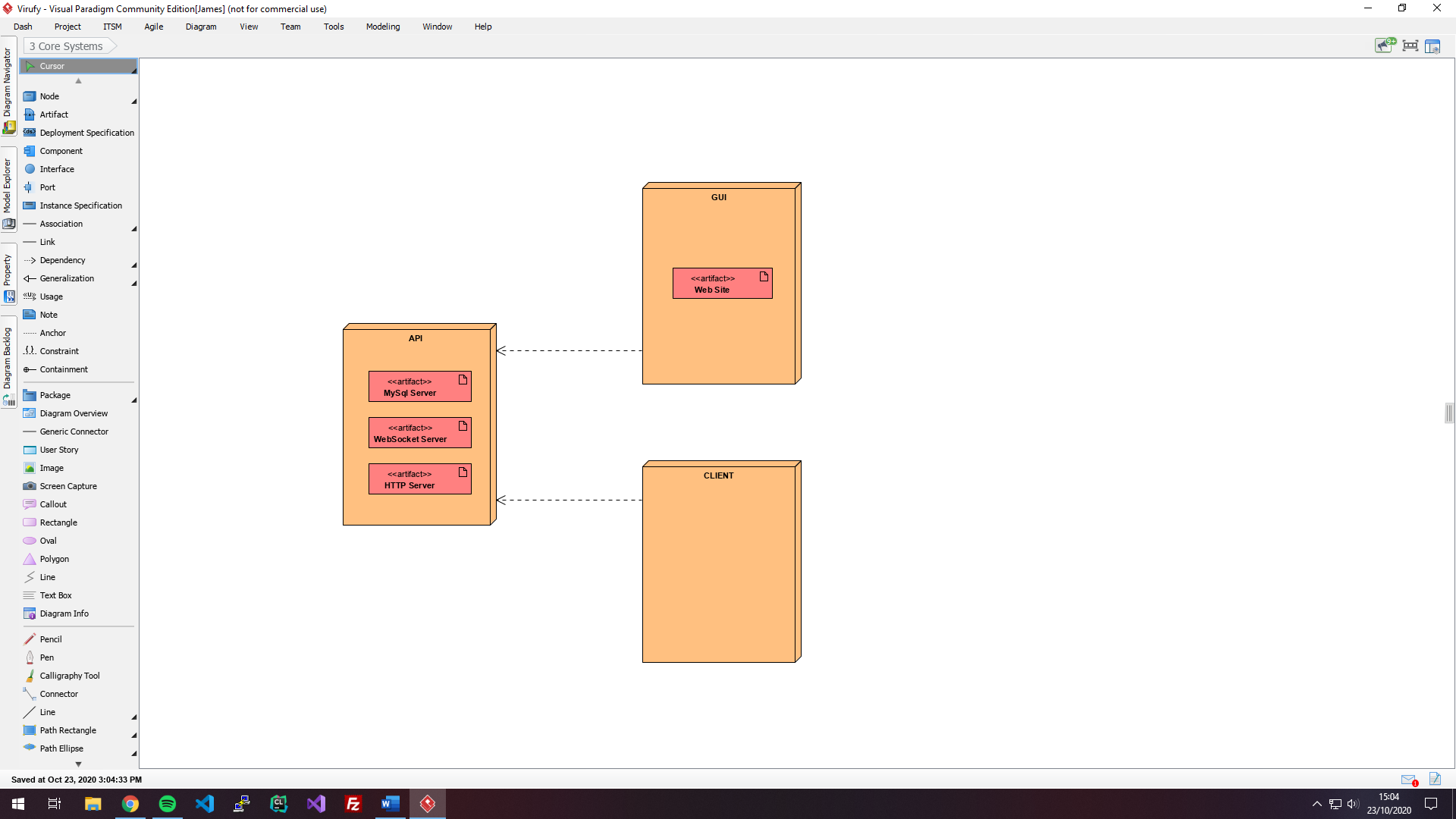
The system will be a **full** finished product and as so will require many different services. The three core elements are as listed.

* 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

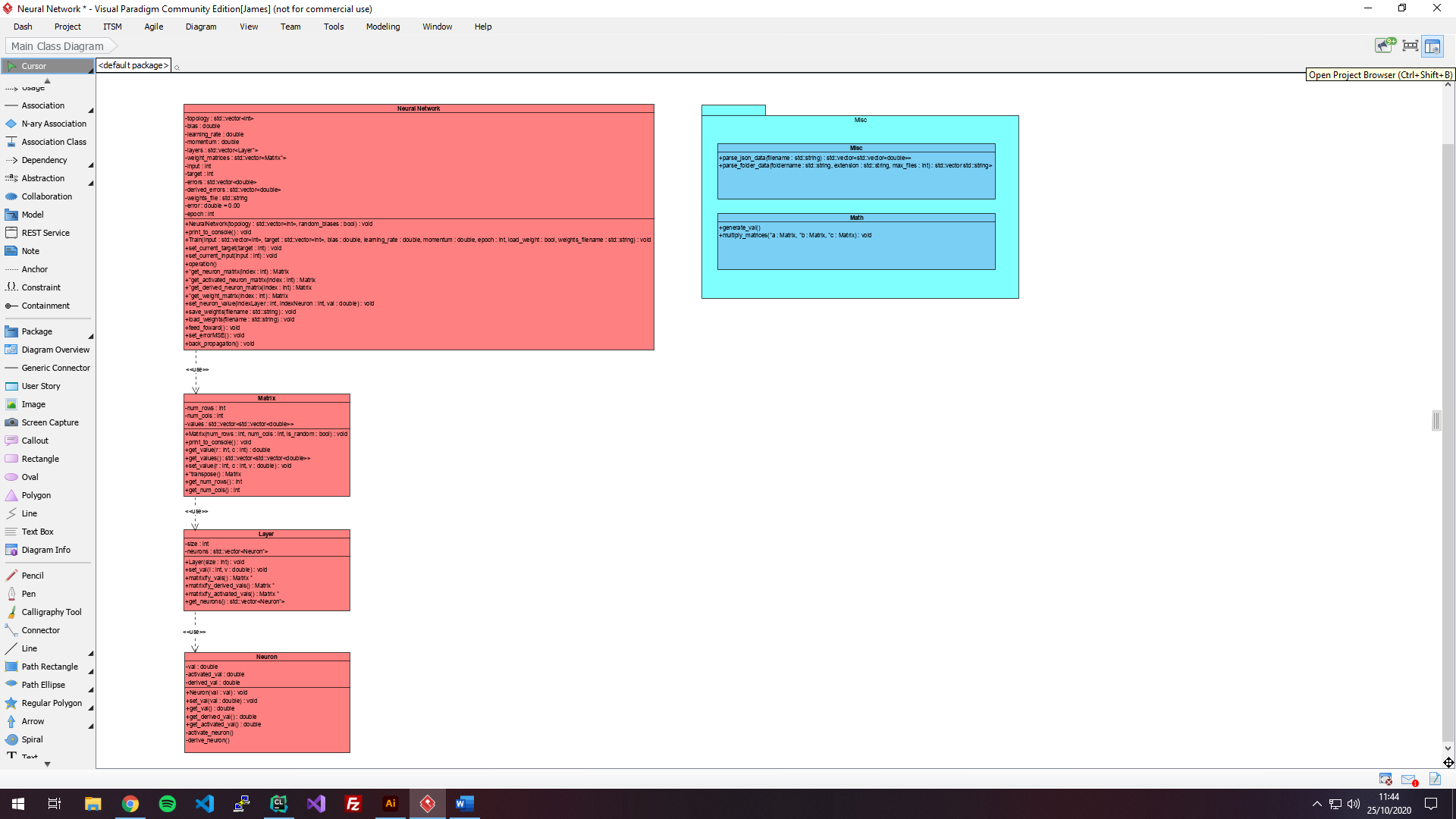
The deployment diagram below is following the **MVC** (model-view-controller) design pattern. As it forces modulatory which will help future changes and feature implementations if my end client needs more features.

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:

### 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.

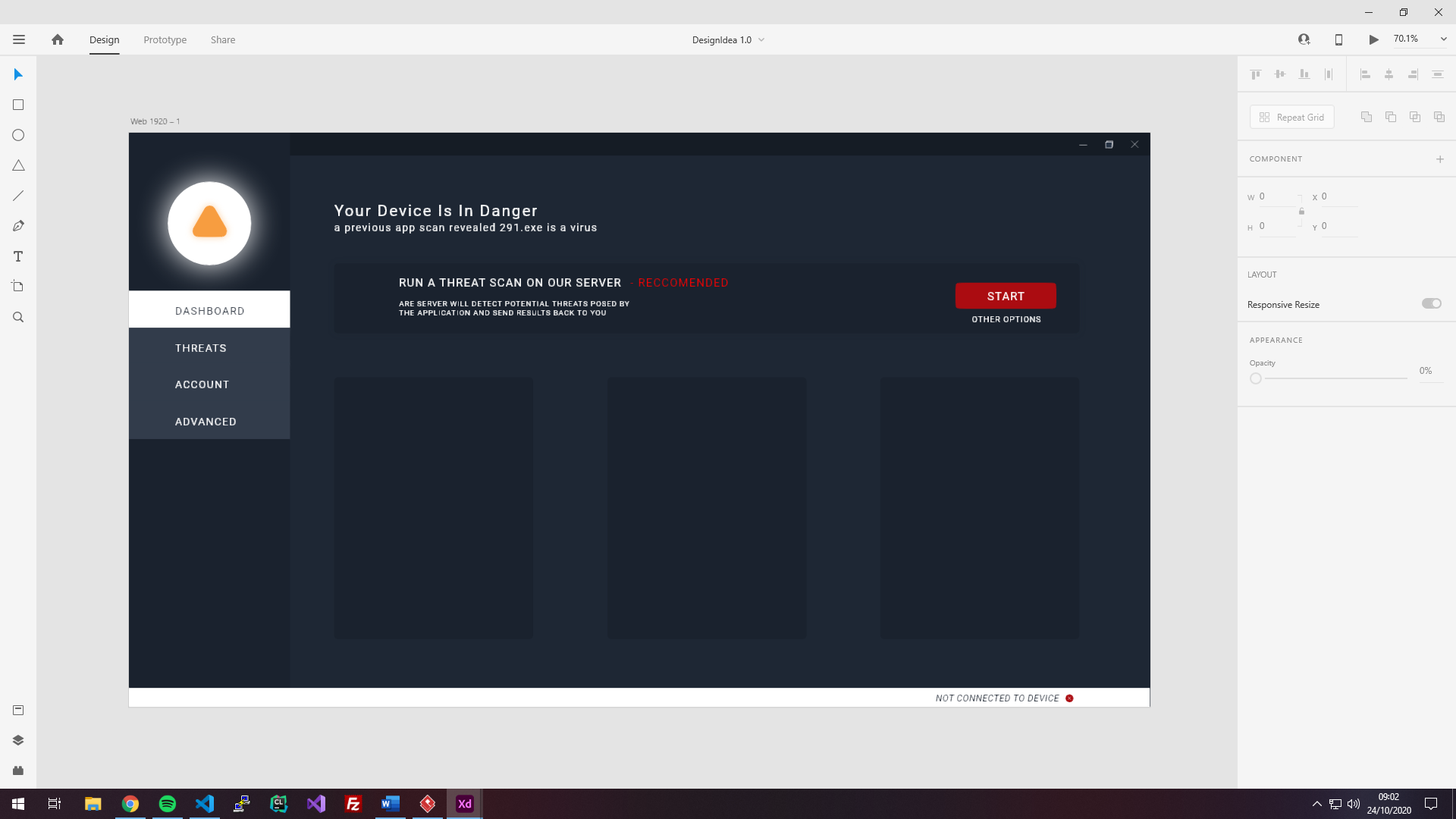
As it is on the cloud it also means that it will only appear when they go online so the intrusive popups won’t exist.

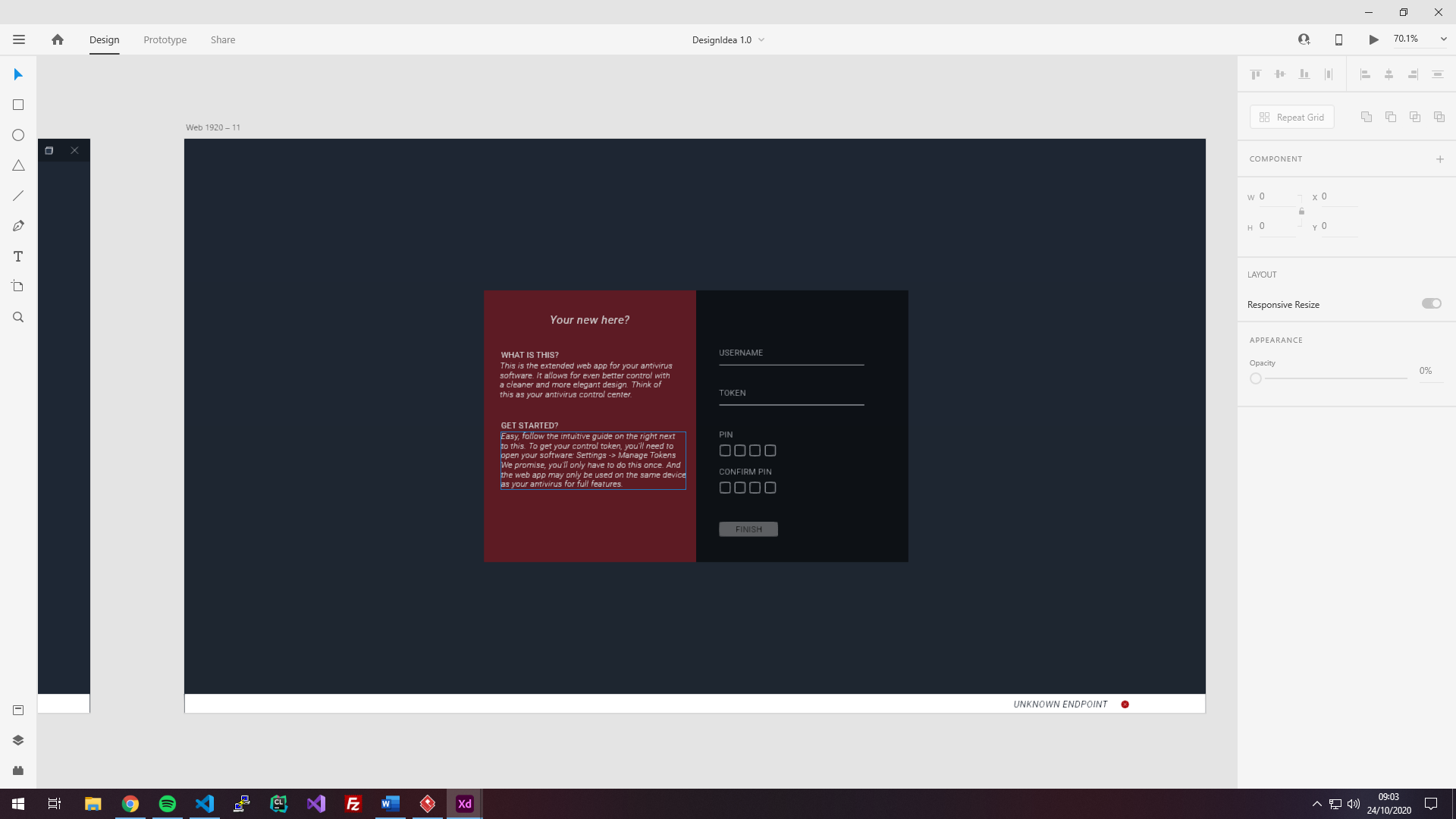
Programming language:

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:

