Phase 2:

Your website will in part be an information repository – which means you need information. Given the overall topic 'web development', you should gather information about topics like the following (you should refine this list based on what your target persona wants).

This Phase Requires:

- *o The content for your website: you should create this in document form before you convert it to your website and must keep it updated with any changes. Ensure you cite all sources and quote where you have copied text verbatim. Note that a collection of urls is NOT CONTENT.*
- *o How you linked this information to your persona's needs*

Content Gathering:

In the previous section we explained why and how we selected our target persona, summarised into the generalised portrayal of 'John'. In this section we will gather additional information in order to create a satisfactory user experience for 'John', i.e. 18-25 year old males, who are familiar with mobile phone and laptop usage, primarily choosing to use interfaces to learn new things whether that be socially, study related or casual.

Website Features

Keeping the goals, needs and motivations of our persona, 'John', in mind, we believe it would be sensible to base ideas regarding the website and it's features, around the commonly stated goal of enhancing skills and completing tasks.

Our persona represents a specific brand of 'general user'. Our previously mentioned survey provided relevant usability insights for individuals conforming to our 'John' persona. One such insight included quoted webpages, including quora and stackoverflow along with less common sites such as canvas, as being particularly useful to them in achieving their goals. Therefore it can be assumed that our target persona, 'John', would utilize a similar site, particularly when optimised for prime useability.

Obviously if our target persona is using these sites already, there must be some degree of success in terms of usability. We considered the features of these sites that were contributing to positive useability in conjunction with participant's answers to survey questions that matched

our ‘John’, general user persona. Through thorough analysis the team managed to propose the following features to be implemented in our webpage:

1. Available cross platform, i.e. accessible on mobile and laptop and more.
2. Search bars available for navigation
3. menu options
4. navigational context, user can see where they are on the website
5. links to navigate to other parts of the site (hypermedia)
6. Comments and responses to comments (likes etc)
7. text on the page
8. consistent colour palettes
9. fast and responsive

These are the significant features consistent through sites currently used to achieve the goals of our target persona. It is therefore part of the users system image, leading us to assume the user expects the implementation of these features in order to deem the site useable. There are most likely further features for our team to implement and certainly different ways to implement and present these features to the user to optimise usability for our specific target persona, ‘John’.

In order to physically implement these features to our website, the team will require further research on a multitude of programs and concepts used in web development. Our team already knows that the use of HTML coding along with programs such as python’s bottle, javascript and many more are necessary knowledge in order to materialise our ideas. The following research includes descriptions of how some of these programs work and gives an idea of how the team will be able to implement the designs.

TCP/IP Stack

TCP/IP (or, Transmission Control Protocol/Internet Protocol) is a suite of networking protocols. This is used extensively to interconnect network devices across the internet. This can also be used to establish communication within a private network, whether that be intranet or extranet[1]. In order for the group to effectively develop a website, it is imperative that we are aware of these protocols to conform to the formats and rules of websites.

The TCP/IP stack can be broken down further, into both TCP and IP.

- The **TCP** aspect of the protocol acts on the *transport layer*, and is responsible for dividing data into smaller data segments. These are directed to a specific *port* used by an application or program on the respective request machine[2].
- The **IP** protocol operates of the *network layer*, and this protocol is responsible for ensuring that *data packets* arrive at its destination. This address is determined via the *IP*

[1] Rouse, M. (2008). *TCP/IP (Transmission Control Protocol/Internet Protocol)*. SearchNetworking. Available at: https://searchnetworking.techtarget.com/definition/TCP-IP [Accessed 19 Mar. 2019].
[2] Alessio Piergiacomi (2016). *What is the difference between TCP and IP protocols?*. Quora. Available at: https://www.quora.com/What-is-the-difference-between-TCP-and-IP-protocols [Accessed 20 Mar. 2019]

addresses of the host and request machines[3]. This is done by *routing* the packets using a routing table. This table keeps track of all paths to a specific IP address.

When these two protocols work in conjunction, it enables data packets to arrive at the destination to arrive out of order. This allows for packets to take the fasted possible path that exists within the system. Error checking is done on each packet once it arrives at its destination.

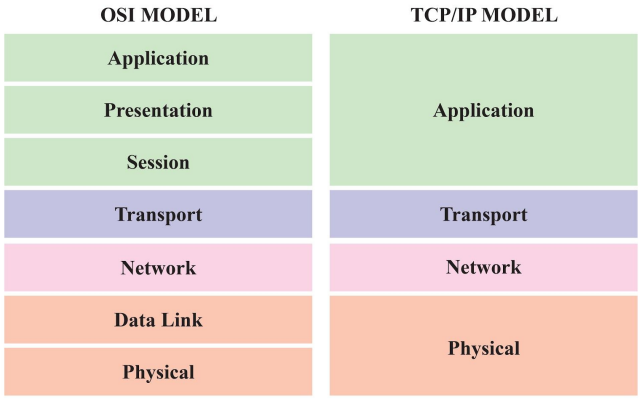
The TCP/IP model runs on a *client/server model*. This definition means that once the client *requests* data from the webpage, the host server will then work to send that requested information to the client.

This model definition can be expanded using routing. TCP/IP is considered '*stateless'*.[4] This means that all data packets are sent independently of each other; they're not dependent on any other variable. This is what enables for packets to travel down different (but always the fastest) communication paths.

This is a 'simplified' version of the internationally recognised *OSI Model*, which outlines each separate layer required to complete a full communication link. The TCP/IP model completes these layers in leaps and bounds, enabling this simplification to be covered in many less layers. This in turn allows for even quicker distribution of information over the internet.

These layers all include their own layer protocols, which acts as a means of communication between successive protocols.

- **Application Layer**: This layer is responsible for the node-to-node communication, as well as controlling user-interface specifications[5]. Examples include;
    - HTTP - Hypertext Transfer Protocol
    - FTP - File Transfer Protocol
    - SMTP - Simple Mail Transfer Protocol
- **Transport Layer:** This layer is responsible for end-to-end communication of data, and its error-free delivery[6]. The two main examples include;
    - Transmission Control Protocol (TCP)
    - User Datagram Protocol (UDP)
- **Network Layer:** This layer handles the logical transmission of data over the entirety of the network. Examples include;
    - IP - Internet Protocol
    - ARP - Address Resolution Protocol



[3] Bradley Mitchell (2018) *Understanding Transmission Control* Available at: https://www.lifewire.com/transmission-control- [Accessed 20 Mar. 2019]
[4] Refer to footnote 1 (Rouse, M. (2008). *TCP/IP)*.
[5] Achiv C, Palak J (2018). *Computer Network | TCP/IP Model*. GeeksForGeeks. Available at: https://www.geeksforgeeks.org/computer-network-tcpip-model/ [Accessed 20 Mar. 2019]
[6] Refer to above.

    - ICMP - Internet Control Message Protocol
- **Physical Layer:** This layer is responsible for specifying the hardware aspects of the communication of data used within the network. This layer holds standards, such as the IEEE 802.3, the specification for Ethernet network media[7].

I'll need to add reference to the picture on the right because it looks nice and I spent 10 mins making it so I'm going to force people to look at it.

Relation to Persona

This topic within our information repository website is absolutely vital in understanding the foundational elements of how communication is executed over the internet. By providing this information to users about this process and its associated protocols, users can develop a deeper understanding and appreciation of the transmission of data. This was especially required for our target persona as very little programming was identified amongst this targeted niche, and even less would be known about the 'behind the scene' protocols that enable this world-transversing communication link.

HTML/CSS, Frontend Display

Hyper**T**ext **M**arkup **L**anguage, or **HTML**, is the code and language used to develop the *structure* and *form* of a website. This works in conjunction with **C**ascading **S**tyle **S**heets, or **CSS**, which gives this structure *style*, *aesthetic* appeal and an eye-catching *appearance*[8].

More on HTML

The foundational characteristic of HTML is its use of ***Markup***. This enables encapsulation of certain data to inherit certain aspects of that markup[9]. These are also commonly known as *tags*, which are constructed using *elements*.

Elements

These designate the definitions of the structure and the content of objects within the webpage[10]. Common elements include heading allocation elements (<h1>, <h2> … <h6> elements) and the paragraph element (<p>).

---

[7] Oracle Corporation (2010). *TCP/IP Protocol Architecture Model*. Oracle. Available at: https://docs.oracle.com/cd/E19253-01/816-4554/ipov-10/index.html [Accessed 20 Mar. 2019]

[8] Unknown. (2016). *HTML and CSS*. Available from: https://www.w3.org/standards/webdesign/htmlcss#w3c_content_body [Accessed 20 Mar. 2019]

[9] José Luis Quintana (2016). *What is markup in HTML?*. Quora. Available at: https://www.quora.com/What-is-markup-in-HTML [Accessed 20 Mar. 2019]

[10] Shay Howe (Unknown). *Building Your First Web Page*. Available from: https://learn.shayhowe.com/html-css/building-your-first-web-page/#common-html-terms [Accessed 20 Mar. 2019]

This can be identified and established with an opening ‘less than’ sign (<), and ending in the ‘greater than’ sign (>). Both of these should appear on either side of the element name. An example can be seen below.

```
1: <h1>
```

Tags

A HTML element usually consists of a **start** and **end** tag[11], encapsulating the content to be affected.

The opening tag is created as the element, beginning with a less than sign, elements name, then the greater than sign.

```
1: <p>
```

Then to close this, the same thing is applied. However, there is a forward slash placed before the element name.

```
1: </p>
```

When this is put together, anything in between the opening and closing tags will have the element name applied to it. For example;

```
1: <p> This is a paragraph. </p>
```

This example will apply the <p>, or paragraph, attributes onto the content in between the tags. This simple yet effective means for marking content with a specific tag allows for the organisation and separation of data.

Attributes

Attributes are placed within the opening tag, after the elements name. This adds additional properties to the element. An attribute generally includes an attribute name, and an attribute value[12]. An example of this can be seen below;

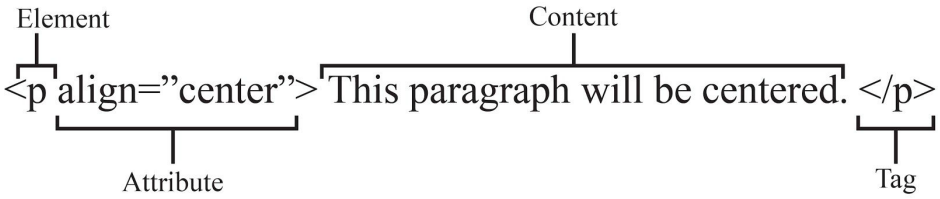
```
1: <p align=”center”> This paragraph will be centered. </p>
```[13]

---

[11] w3Schools (Unknown). *HTML Elements*. Available from: https://www.w3schools.com/html/html_elements.asp [Accessed 20 Mar. 2019]

[12] Same as foot 10, https://learn.shayhowe.com/html-css/building-your-first-web-page/#common-html-terms

[13] BitDegree (Unknown). *Mostly Used p Tag Attributes*. Available from: https://www.bitdegree.org/learn/html-paragraph [Accessed 20 Mar. 2019]



Building a Website

HTML must be initially set up in a certain way. An example can be seen below.

```
1:  <!DOCTYPE html>
2:  <html>
3:       <head>
4:            <title> Hello World! </title>
5:       </head>
6:       <body>
7:            <p> Hello Again! </p>
8:       </body>
9:  </html>[14]
```

- <!DOCTYPE html> - Line 1 is a document declaration type, which informs the browser which version of HTML is going to used from thereon.
- <html> - Line 2 informs the beginning of the html document
- <head> - Establishes special parameters of the website
    - <title> - Changes the tab name to what you allocate it to be
- <body> - Establishes the body of the document
- <p> - Specifies the area to be in paragraph form

More on CSS

CSS dictates the appearance of a website. CSS can be implemented in three main ways:

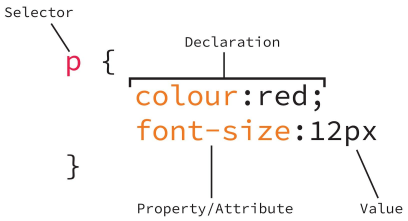
- External style sheet
- Internal style sheet
- Inline style

[14] w3Schools. (Unknown). *HTML Editors*. Available from: https://www.w3schools.com/html/html_editors.asp [Accessed 20 Mar. 2019]

The easiest way to manage a website is to use an external style sheet. From this single style, it single handley holds all the power to change its looks. To do so, the HTML page must make a direct link to the style sheet to be used. This can be done in a single line, as seen below:

```
1:  <head>
2:  <link rel="stylesheet" type="text/css" href="style.css">
3:  </head>[15]
```

CSS provides the user a means of giving particular tags certain global attributes. Let's take the following example.



The *selector* selects the tag to be edited. In this case, it's the paragraph ("p") tag. Each tag has specific attributes associated with them. This *property* can be added within the curly brackets. Once this is done, edits can be made to the *value* of this property. As seen about, the paragraphs tag will have font colour of red and be of font-size 12.

As mentioned before, this would change ***all*** <p> tags within the HTML document, making style changes easy and global.

JavaScript

JavaScript (JS) is a dynamic scripting language that allows interactivity to exist within a static, HTML website. This essentially means that if it moves, refreshes or interacts with the webpage, it is run via JS. As mentioned before, it is a scripting language. But what is a script? A scripting language is a language that can be used to automate many of the process that a user would otherwise have to execute manually[16]. This means that the user can focus more on what *is* happening, than *how* to make it happen.

Frameworks & Libraries

What is this? A software framework or library provides a layer of abstraction for the programmer. This means that predefined functions are readily available for the user to use. This makes the use of JavaScript to be a much more enjoyable, easier and smoother experience; enabling the programmer to focus on the major task at hand, and not the minimal, smaller problems.

---

[15] w3Schools. (Unknown). *CSS How To…* . Available from: https://www.w3schools.com/css/css_howto.asp [Accessed 20 Mar. 2019]

[16] https://skillcrush.com/2012/04/05/javascript/

How does JavaScript work within a Web Browser?

1. The browser reads in, or parses, the script
2. Converts, or compiles, the script into machine language, or computer language
3. User machine runs it, very fast[17]

The search engine will attempt to optimise each step of this process. In-browser JavaScript can edit anything relating to the web page, manipulating data, interacting with the user, the webs server. Newer versions of JavaScript cannot directly access files or commands from the OS of the system. This improves the security of the overall system.

Examples

```
1: var x, y, z;         //Declares variables to be used
2: x = 6; y = 7;        //Assigns values to the variables
3: z = x * y;           //Computes a variable using variables[18]
```

A script can be imported in the following way.

```
1: <script src="script/file.js"></script>
```

Relation to Persona

The above front end development tools are extremely important for newcomers to grasp. This does not necessarily mean technically, but more conceptually. By developing a mental model of the difference between HTML, how it is the structure of the website, as opposed to CSS, which is made for the styling of the website. These small and minute details, which would be completely new concepts for most, is vital to understand early. Simple language and non-specific jargon has been used were possible to allow readers to relate to the content. Simple examples to explain the code structure and its elements have been used for demonstrations. Graphics with annotations of code lines have been created to further develop their understanding, as well as another means for explaining these concepts.

Web Backends

If the frontend development (HTML/CSS, JavaScript) is the facade of the website, backend is the structural integrity of the website. The backend is responsible for things such as the calculations, business logic, database management and interactions, handling user performance[19]. This technology is for managing and processing incoming user requests,

[17] https://javascript.info/intro
[18] https://www.w3schools.com/js/js_syntax.asp
[19] https://www.upwork.com/hiring/for-clients/frontend-vs-backend-web-development/

generating the required responses, and sending this response to the user client. This generally occurs in three major steps;

- **Server:** Computer that receives the incoming request
- **Application:** Program that awaits for requests, performs its relative calculations, and sends the response out
- **Database:** Table that holds persistent data that the application can access, read, write, manipulate and use to calculate

Bottle

Bottle is a Python Web Framework. This fast, simple and lightweight Web Server Gateway Interface allows for the easy implementation of web servers. The qualities described previously are thanks to this module having no dependencies other than the Python standard library (the default python library). Below is an example of a dynamic URL, which says “Hello <name>”, depending on the user input in the URL. This has many various applications in larger projects.

```
1:  from bottle import route, run, template
2:  @route('/hello/<name>')
3:  def index(name):
4:       return template('<b>Hello {{name}}</b>!', name=name)
5:  run(host='localhost', port=8080)
```

Flask

Flask, like Bottle, is also a Python Web Framework. However, it includes many more features that allows the user to ‘fine-tune’ and edit more aspects of the website functionality and usability. It supports a wide range of database structures to be implemented into the roots of the website server. This enables developers to have more range with their choices in deciding which algorithms and datastructures are required to be developed for future features of the current (or in development) website. Flask also has no external dependencies.

Flask allows for the use of templates to be easy implemented. Have you ever wanted to make a small, but global edit to a website? By editing the template, this can easily and quickly be achieved. This can save a lot of time, which can be put to better use in the future.

Django[20] [21]

Django is a high-level python web framework that supports the clean, practical and quick implementation of web services. Django prides itself on being ridiculously fast. This was

[20] https://www.djangoproject.com/start/overview/
[21] https://www.javatpoint.com/django-features

designed as so as to allow for developers to focus as much as their energy as possible to their project at hand, and not the small and tedious tasks which are only time consuming. Some of these included features are;

- Administration Rights and Monitoring
- User Authentication
- RSS Update feed
- Site Maps
- Encourages human-readable URLs[22]

PHP

PHP (Hypertext PreProcessor) is a programming language that can be embedded into HTML code[23]. The main purpose of this is to allow **server-side** scripting to occur. This essentially means that the code is run on the server, and these results are the sent to the user. This means that the user can not 'see' the code behind the scenes[24].

NodeJS

NodeJS is an asynchronous web framework which allows for websites to have multiple users executing commands concurrently. This is due to the clean NodeJS functions, which has no direct input/output requirements. This means that there won't be a backlog of data to process on the server system when multiple users are completing similar user functions and actions.

Apache[25] [26]

Apache is a **web server**. A webserver is a program that listens and accept requests from clients, and send a response back. In its most basic form, Apache is rather limited. However, its packages is what releases its true power.

In essence, Apache is the road that enables for the communication between the client and the webserver. It maintains this connection as best it could, and will raise an error if one occurs.

Nginx

Nginx is also a web server. It receives requests and sends results. However, what makes Nginx unique is its ability to handle large volumes of users. In fact, Nginx was made by Igor Sysoev to solve this problem (The C10k Problem) in the late 90's[27]. It does so by offering low memory usage and high concurrency. This is enabled by it handling multiple user requests (all of which are similar) in a single thread. This allows for the system to work independently of each other, and not slow eachother down.

---

[22] https://medium.com/swlh/what-is-django-and-why-is-it-so-popular-2b225620cca0
[23] http://php.net/manual/en/intro-whatis.php
[24] http://php.net/manual/en/intro-whatcando.php
[25] https://kinsta.com/knowledgebase/what-is-apache/
[26] https://code.tutsplus.com/tutorials/an-introduction-to-apache--net-25786
[27] https://en.wikipedia.org/wiki/C10k_problem

.NET[28] [29]

.NET was developed by Microsoft. The purpose of this application is to allow for cross language interoperability. This enables for different code segments to be ‘stitched’ together to allow all functions to work. .NET has multiple Application Programming Interfaces (APIs) which enables for an even wider range of possibilities to emerge.

Relation to Persona

Backend web development is a difficult concept to understand, even for intermediate programmers. This is why each succeeding paragraph goes into a more in depth concept of programming application. This gives users an idea of where their own understanding is at. This type of self reflection allows for readers to appreciate the scope of backend development, while at the same time not overestimating their own abilities.

Web Server Hosting Platforms

A Web Server Hosting Platform is an individual or organisation that allows for other entities to ‘rent’ out server space on the platforms own devices. This is done as the web hosting services already have the infrastructure in place to provide world-class, global internet access. Particular web hosting companies have their own interfaces which improves the ease of use for consumers to manage their own websites[30].

Wix[31] [32]

Boasting over 110 million users, Wix is a website builder. They allow you to embed text, images and anything you want on their fully online platform. This means that just about anyone can take advantage of this, as no coding or other technical skills are required. They even handle the web hosting aspect of the website, allowing the creator to fully focus their attention on what matters most.

Google Sites[33] [34] [35]

This completely free website builder and hosting service is perfect for anyone with a tight budget and schedule. It has very simple gadgets and designs, which make it easier to create, bt doesn’t have the advanced features required for a more robust and in depth website. Google Sites also allows for multiple people to edit the same website at the same time. This means that at team of

---

[28] https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet
[29] https://en.wikipedia.org/wiki/.NET_Framework
[30] https://en.wikipedia.org/wiki/Web_hosting_service
[31] https://www.wix.com/about/us
[32] https://www.websitetooltester.com/en/reviews/wix-review/
[33] https://gsuite.google.com.au/intl/en_au/products/sites/
[34] https://chrome.google.com/webstore/detail/google-sites/gmandedkgonhldbnjpikffdnneenijnd
[35] https://www.websiteplanet.com/review/google-sites/

people can work concurrently, with fluidity and purpose, without hindering the work of each other. This services enables users to make fully responsive and dynamic websites that deliver the nest quality service that can be expected for a free website builder.

Amazon Web Services[36] [37]

Amazon Web Services (AWS) is a cloud service provider with over 60 data centers across the whole world. AWS provides a multitude of services from database storage, auxiliary processing, analytics, Internet of Things and many more. AWS has the global infrastructure necessary for producing the results so many companies want.

Azure

Azure is a Microsoft run cloud service provider. This particular services runs a ‘pay-as-you-use’ model, meaning you will only have to pay for the storage or running time you actually use. This helps minimise user expenditure on unnecessary elements of packages they never end up using.

DigitalOcean

DigitalOcean is yet another cloud service provider. Their main focus is on scalability, a feature present in all cloud service providers. One major feature that DigitalOcean does have over its competitors is its online marketplace. This allows users to purchase apps that can better suit their personal and business needs. This reduces the costs of developing ‘in-house’ softwares as they won’t have to be developed from scratch, or developed at all since it already comes in an easy to user, already complete package.

//*

I reckon what we wanna do in this is one is yeah obviously the research and shit we know we have to put in but like i reckon we sorta lay it out a bit like this:

[36] https://en.wikipedia.org/wiki/Amazon_Web_Services
[37] https://aws.amazon.com/what-is-aws/

First we'll just basically briefly state the general target persona that we identified in the previous module. We then go over what we found the main things that they were using websites for and what our main idea is for the website, lets just say for example we found that our target was like a mid to late teens/early twenties intellectual with the aim of learning and stuff like Pat put in his survey, then basically all im saying for the first bit is that we say something along the line of; ***In the previous section we explained why and how we selected our target persona, in this section we will gather additional information in order to create a satisfactory user experience for mid-teen to early twenty year old males, who are familiar with mobile phone and laptop usage, primarily choosing to use interfaces to learn new things whether that be socially, study related or casual.***

So then i reckon the best thing to do after defining all of that whatever that turns out to be is to start talking about the general subject of the website we're going to create and why, and after defining that going into some research as to what those sites look like and what are some useable features and stuff. So for example:

Keeping the goals of our target user in mind, we believe it would be sensible to base our idea around the commonly stated goal of being able to learn something. Websites that were quoted included pages like quora and stackoverflow along with less common sites such as Khan Academy and Edstem. In order to optimise a site with the same goal of these websites, i.e. to learn and understand problems, We considered the usability aspects of the similar sites contributing to their success, listing noteworthy features for further research.

Then basically we would list features like the search bar on stack overflow, or the simplicity of the EdStem task bar and give them somebullshit research about how like purple and big text attract attention, then some further research on like why simplicity is good for usability. Ideally oyu'd want some kinda article that mentioned our specific demographic saying like idk, teen boys are more responsive to black and white colour schemes and that.

After we say some research and stuff along those lines we then go into the ways to physically implement these features, so this is where basically a huge data dump of info on shit like python-bottle, HTML and all that would come so we'd essentially write something along the lines of this:

In order to physically implement our justified usability aspects above, the group was required to expand upon current knowledge and begin researching and implementing new skills in web design. Some of these features and tools to materialise our ideas onto the internet are explained below:

JavaScript: Java Script is an object-oriented computer programming language commonly used to create interactive effects within web browsers. JavaScript is a tool essential in optimising our usabiliy ideas as it can _____ helping to make our ideas of _____ possible. Things the team needs to know about javaScript _____. Features of JavaScript applicable to our teams idea includes _____ to implement _____.

Bottle: ……

So yeah then we basicallt jsut go through all the web development tools and shit that we'll probably need to make our website, just remebering to relate it back to the idea and like what it can do for usability.

Then basically in summary when thats all done go through like what the main ideas are and potential features that could enhance the usabilty of our target demographic, and like just say that the team needs all this research and shit to be able to bring it to the screen for the good of the user and all that.

Again just ideas, feel free to come up with better ideas -Darby