Luis Santos

Supervisor: **Alexander Fedorec**

Due 18th April 2016

HTTP Remote system management

URL: **cybortech.co.uk/hydra**

Word count: **8455**

Product and Development files:

Uploaded as a zip file

A dissertation submitted in partial fulfilment of the University of Greenwich's

BSc (Hons) Computing

ABSTRACT

A cost effective way to remotely manage various end systems at once, allowing administrator’s full control over the system remotely; allowing tasks to be remotely added to a database and carried out by end bot machines.

ACKNOWLEDGEMENTS

I would like to thank the following people who, in a way or another, have supported me throughout my professional development and academicals carer:

My entire family, my parents especially, for always supporting me throughout my education. A special thank you to my uncle, who is no longer with us, would want you to know that I have followed your footsteps and have made it, one way or the other.

My project supervisors, starting with Dr Asif Maliq, who unfortunately had to leave and Dr. Alexander Fedorec, who stepped up to replace my previous supervisor. Thank you for the prompt feedback on my work and all the ideas and advice given to me.

My friends and folks who I have cut connection to due to concentration on my education.

Contents

INTRODUCTION

1.1 Background

Unsurprisingly, with new technological advancements, hackers have always looked at how these can be exploited and used for their beneficiary intendment, HTTP based system, without a doubt. Post the use of HTTP various other protocols such as IRC, Custom server/client connections using the TCP and the Peer-to-Peer (P2P) Protocol. Each protocol used has its advantages and disadvantages, yet they work to the same basis in mind, remote management of a client.

Thus, the interest of how Hackers exploit this protocol due to almost all firewalls allowing HTTP traffic without blocking the default port 80[1] using such to redirect command and control traffic. Therefore, for our research and development I intend to inquest how the HTTP protocol is vitally exploited and what I can learn from these based on their architecture reliability and performance to develop a reliable system capable of monitoring end user systems securely. Additionally, research into how the system must be modified to ably with set rules, regulations and laws. Botnets, Short for Robot Network, are considered to be an uprising thread to computer networks. thus according to Jae-Seo Lee, these systems can pose a serious threat to the internet society [2], nonetheless, An investigation into how these can be a cost effective way to remotely manage various end systems at once shall be carried out, taking into mind functionality such as allowing administrators full control over the systems remotely; allowing for tasks to be remotely added to a database and carried out by end user machines. Microsoft's operating systems are targeted in this project as they are vastly used worldwide. Additionally the chosen languages are OOP C++, PHP followed by the MySQL database system as I have proficiency with the corresponding.

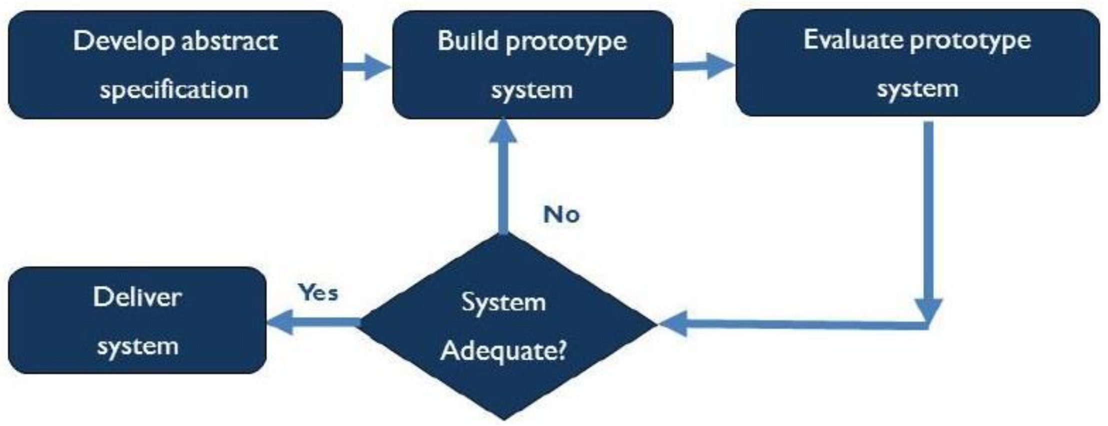
1.2 Aims and Objectives

We aim to investigate into botnets and their architectural sophistication alongside the various protocols used. The base outcome is to develop a system able to monitor a range of end user devices; thus being a modified, for legal purposes, version of a botnet, whose sole scope is to monitor and perform remote tasks, which require large distributed resources of computing power, with the end users acknowledgement. The system shall be aimed at the Microsoft operating system, full compatibility with windows codename ‘Vienna’ and other desktop based releases post Vienna is a fundamental aim of the project. Moreover, the application side of the program shall not interfere with the user and be run as a background process. The Core aim is to develop a system capable of displaying system information a web panel, however, the system is astonishingly versatile and allows for a vast amount of features to be securely added, these are but not limited to, remote key-logging, website traffic redirection, and remote desktop sessions. Therefore, the project shall not be limited to any additional features if these can be

Implemented in future releases, data security shall be taken into count and a basis of encrypting plaintext data shall be implemented.

1.3 Approach

Dynamic approach, thus followed from the Evolutionary prototype model, as a system that can be modelled to various needs and uses, an approach able to communicate, design and prototype the system is essential. Various requirements can be put forward for future released, however, these shall be classified by interest and priority; analysed using a MoSCoW system of prioritisation.



Again, the dynamic approach must matter for additional features from the end user. However, not drifting away, the main core is the crucial base foundation that shall be developed to determine the outcome of the project and its architectural stability. Additionally, it's is of crucial attention that the approach must also allow for rollbacks to occur as we are dealing with a tool that could potentially become and evasive malware.

When following a development stages closely, thus decreases the probability of failure as there are stages to perform, following these stages require you to investigate deeper into the project, I'm not saying it will make your project successful but as said above the probability of failure will decrease. Following pre-set stages provided by the aims, furthermore the baseline for the project are as mention weighted and developed in a dynamic manner with the correct approach.

2 LITERATURE REVIEW

2.1 Introduction

Both governmental and antivirus companies know the impact of a botnet as a method of stealing data and other miscellaneous implications it might bring. However we are concentrating on using such as a sole system to monitor end devices performance. Systems used by fanatical hackers have been developed to a high extend and extreme functionality as these people are passionate about their hacking.

Miscellaneous systems could be transferred and used to improve current methods and drive the current system used by overpowering them with the features used by high end malicious systems. The main basis and approach is to look at high end malicious systems and use that expert code, written fanatically, to develop a legal based system by using the features provided.

Throughout the years, different protocols have been used to develop these systems, always allowing for hackers to be on the top of 'innovative malware'. Thus also as new technologies came out as well as different protocols or the efficiency of current ones upgraded. Some of the protocols used over the years are, but not limited to IRC and Peer-to peer, a detailed explanation of these follows,

P2P- short for peer to peer, does not follow the conventional method of a botnet of which connects to a main command and control system, the peer to peer has a decentralised system, thus meaning that the bots talk to each other in order to keep their connections alive. However the master is in charge of setting up the communication between its peers and assign them command, thus still being used as a peer within the networks to provide itself with a method of masking the origin of the commands.

Moreover the decentralised network makes it much harder for detection thus because the network traffic does not assign late to one single machine [3], it distributed its network hungriness between peers. However, intervening these is easier than a conventional botnet that used a command and control due to encryption, as thus can be modified to 'poison' the entire peer to peer connection.

IRC- abbreviated for Internet Relay Chat, thus a very useful communication protocol available in various operating platforms and still in great use today but servers can be used as a command and control with no modification at all required by the end attackers.

The system allows a user to connect to an online chat room and be joined by other peers for a conversation. However, application can be modified to connect to the one chat room under a valid name, however depending on the output of the text being sent to the server the individual bot would read these and interpreted them as commands.

HTTP – Hypertext transport protocol, architecturally similar to the IRC but uses newer technologies to achieve so. Thus mainly uses PHP as a back-end language but not limited to other bet languages, is used to run its queries and communicate with the bot, is now used widely do to how cheap web hosting has become. It is also very easy and intuitive to set up, thus brings the attention to younger generations who like to learn and set these up. Even though it is a great way to learn about security, starting your own botnet is in some countries illegal, thus does not stop the eager individuals who like to learn about such.

With the scalability of HTTP and the freedom of uploading the bot to any webhost and link it with one domain, it is a sustainable, cost effective way to control and manage various computers worldwide, nonetheless, it is still being exploited by various programs and used for various miscellaneous uses, one of which is famous is the Zeus botnet that had one of its features as a man in the middle attacker [4] to gather information from various banking websites as the victims logged in, thus highly illegal.

Additionally, there is common software that can be used to do just the same of which the main development is not intended for herding however thus take the liberty of using the TCP protocol as a connection point between each other, making the command and control a locally hosted machine, most of these tools are mainly used for remote managements thus similar to products used by system administrators, however these have the sole miscellaneous purpose of invading ones privacy and illegally manipulating data remotely.

Regardless, not every bot is developed to cause harm to other computer, nor is it aimed for evil, this is the case of the egg drop, a considered antique bot still in development aimed at the true purpose of remove management and system monitoring, thus done using IRC it is a slightly better and effective way to monitor computers in comparison to tools such as telnet or ssh, of which are still highly used today.

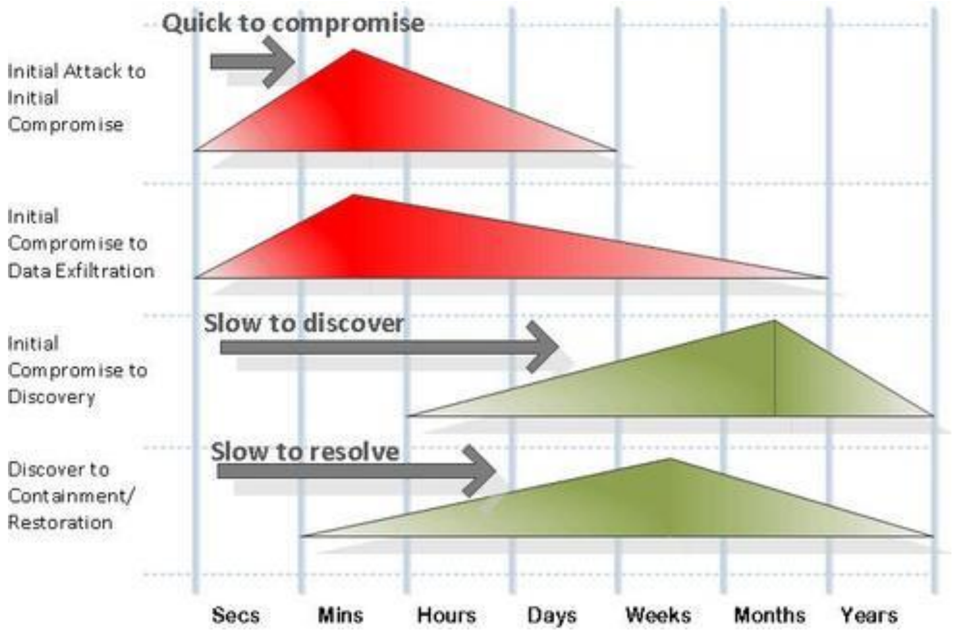
2.2 Key issues to use in the design and implementation

Due to the architecture of the tool, care shall be taken to not compromise any data being gathered and used. Several individuals have over the years been arrested for cybercrime related incidents due to the development of such tools, however these were aimed with the sole purpose of monetary gain. On the other hand, care must be taken to follow laws and legislations and to not create a tool capable of compromising any data.

2.3 Conclusion

Users of these tools are sometimes people who have no programming background or knowledge; younger generations who like challenges; people eager to learn about their usability for profitability. However, it is clear that the system regardless of the threat to privacy and possible misconduct when using so, are highly complex products that can be adapted for legitimate uses.

Furthermore, the current use of encryption witching tools of this scale is allowing for hackers to hide their software, masking these as legitimate products, thus bringing the time of discovery much later than [5], thus after the incident has taken place. As foreseen, these have become slow to discover.



Nonetheless, our product must be visible and clear of any data it shall collect preventing data miss usage by anyone attempting such.

REVIEW OF SIMILAR PRODUCTS

2.4 Introduction

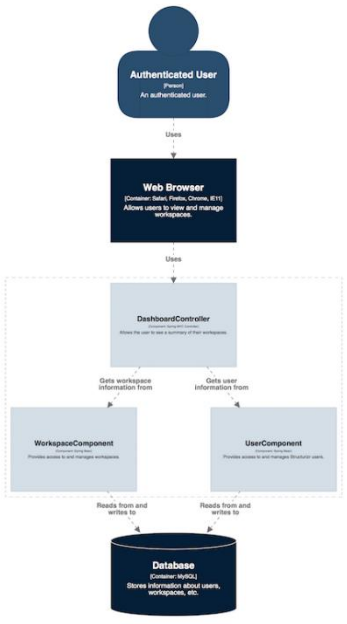
In this section, we compare features used by other similar products and ours. However, it is rare to find some of the related documentation for some of the products being reviewed as the developers may not release the documentation, thus possible of the criminal market that it is intended for. Never the less, we shall also include similar enterprise grade products that are being used worldwide. Regardless of the origin and its intended market, the products shall be reviewed and on basis of the following criteria:

- Overview: determines the main use and innovative architecture in use.
- Utility: details if the product is useful and meaningful for end users, thus touching on the main basis of their use.
- Scalability: investigate on possible updates and the ease of integration of new features.

2.5 Join.me by LogmeIn

LogMeIn is a provider of software and cloud- based remote management as a services. Hence the origin of join.me. This is an online meeting tool aimed at bringing people together by solving the online collaboration issues between users and overcome any problems that originate from such [6]. Mainly aimed at presentation and group meeting over the internet. Additionally, it is written in C# and JavaScript.

2.5.1 Overview


Authenticated User
[Person]
An authenticated user.
Uses
Web Browser
[Container: Safari, Firefox, Chrome, IE11]
Allows users to view and manage workspaces.
Uses
DashboardController
Gets workspace information from
Gets user information from
WorkspaceComponent
Provides access to and manages workspaces.
UserComponent
Reads from and writes to
Reads from and writes to
Database
[Container: MySQL]
Stores information about users, workspaces, etc.

Join.me uses cloud-based technology to provide users with a vast range of functionality, this varies from the amount of video participants that are allowed to view at once to features such internet calling or file transfer. However this varies on the type of package bought by the end user.

The Diagram Presents current architecture being used by LogMeIn, and is analysed by '*Simon Brown*' a member of 'The Coding Architecture' who concluded that the used the web based components could possibly present a risk to the URL routing and its privilege levels, thus could also bring security issues by having users accessing links directly. On a positive note, join.me uses machine readable metadata to specify its user types. However, it is also mentioned that this could possibly be overcome by tweaking the components and used such within a namespace. [7]

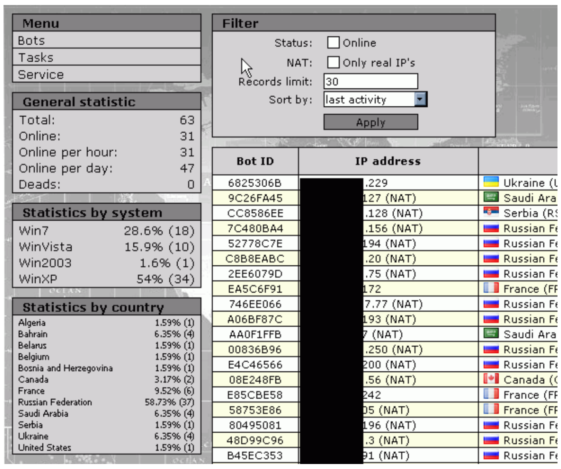
2.5.2 Utility

As previously mentioned, features increase depending on the weighted package bought. He most limited package that is given free of charge allows limited meeting participants(10) where the highest graded package allows for 250, thus being the enterprise package. However these are limited as the max number of connected users.

2.5.3 Scalability

The use of cloud based technology allows various featured to be used and of course we cannot ignore the fact that it is cloud based. Nevertheless, this product only offers a limited amount of storage and features with one of its most valuable being the limited 5TB of managed storage and its user / feature management. Regardless, integration of additional features is limited due to its web based dashboard that restricts only web based application to run.

2.6 Andromeda Botnet



Menu
Bots
Tasks
Service

General statistic	
Total:	63
Online:	31
Online per hour:	31
Online per day:	47
Deads:	0

Statistics by system	
Win7	28.6% (18)
WinVista	15.9% (10)
Win2003	1.6% (1)
WinXP	54% (34)

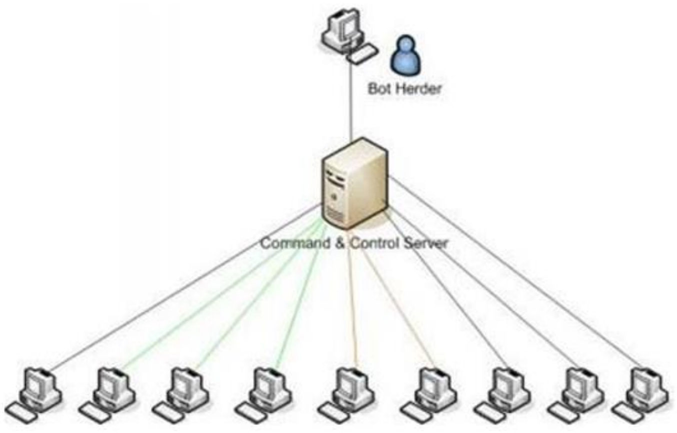
Statistics by country	
Algeria	1.59% (1)
Bahrain	6.35% (4)
Belarus	1.59% (1)
Belgium	1.59% (1)
Bosnia and Herzegovina	1.59% (1)
Canada	3.17% (2)
France	9.52% (6)
Russian Federation	58.73% (37)
Saudi Arabia	6.35% (4)
Serbia	1.59% (1)
Ukraine	6.35% (4)
United States	1.59% (1)

Filter
Status: ☐ Online
NAT: ☐ Only real IP's
Records limit: 30
Sort by: last activity
Apply

Bot ID	IP address	
6825306B	[redacted].229	Ukraine (
9C26FA45	[redacted]127 (NAT)	Saudi Ara
CC8586EE	[redacted].128 (NAT)	Serbia (R
7C480BA4	[redacted].156 (NAT)	Russian Fe
52778C7E	[redacted]194 (NAT)	Russian Fe
C8B8EABC	[redacted].20 (NAT)	Russian Fe
2EE6079D	[redacted].75 (NAT)	Russian Fe
EA5C6F91	[redacted]172	France (F
746EE066	[redacted]7.77 (NAT)	Russian Fe
A06BF87C	[redacted]193 (NAT)	Russian Fe
AA0F1FFB	[redacted]7 (NAT)	Saudi Ara
00836B96	[redacted].250 (NAT)	Russian Fe
E4C46566	[redacted]200 (NAT)	Russian Fe
08E248FB	[redacted].56 (NAT)	Canada (
E85CBE58	[redacted]242	France (F
58753E86	[redacted]05 (NAT)	France (F
80495081	[redacted]196 (NAT)	Russian Fe
48D99C96	[redacted].3 (NAT)	Russian Fe
B45EC353	[redacted]91 (NAT)	Russian Fe

Andromeda, also known as Win32/Gamarue, is an HTTP based botnet. Described as the infamous modular botnet, the program was first spotted in 2011 and became popular on the underground criminal market due to its RC4 encryption and decryption of its network packets[x]. Being used for the wrong purposes such as a 'dropper' for similar malware such as Zeus botnet.

2.6.1 Overview



Andromeda is an HTTP based system that uses a Command and Control system written in PHP, thus being encrypted and with base64 so that access to the source code is restricted to the developer. What we do know is; the system is able to run on various web-servers such as NginX but famously used alongside Apache and its database written for MySQL [8]; its architectural design/structure as follows by the image above, allows for one person to control various systems simultaneously. However no documentation can be found for the project, except for testing and usage testimonies from users of the system [9].

Furthermore, it uses the clients MAC address as its unique identifying number as this is a real unique number within a computer system. However this brings security issues and misconduct due to displaying raw remote MAC addresses which bring serious miscellaneous outcomes.

2.6.2 Utility

The use of a command and control centre for the connecting clients allows Andromeda to specify tasks of which the end client shall download and execute and do so on connection. However the main sell points of the program are:

- Keyloggers
- Form grabbers
- SOCKS4 proxy module
- Rootkits

These, which are aimed for criminal intends, are still high end snippets of code that are tailored to work with every system with functionality in mind. Nevertheless, Andromeda allows for a vast amount of clients to connects, thus the reason of still being used today as a 'herding' botnet.

2.6.3 Scalability

It is clear that the feature that stands out on this program is its stability and ability to handle high volumes of clients. However scaling this an adding future features can possibly affect its stability as mentioned by Thomas Rutter, stable software may run effectively without crashing however adding potential features may alter this by having a higher risk of where the application might fail [10]. Regardless, Andromeda is still a modular botnet which could use the system to allow more features but compromising on stability.

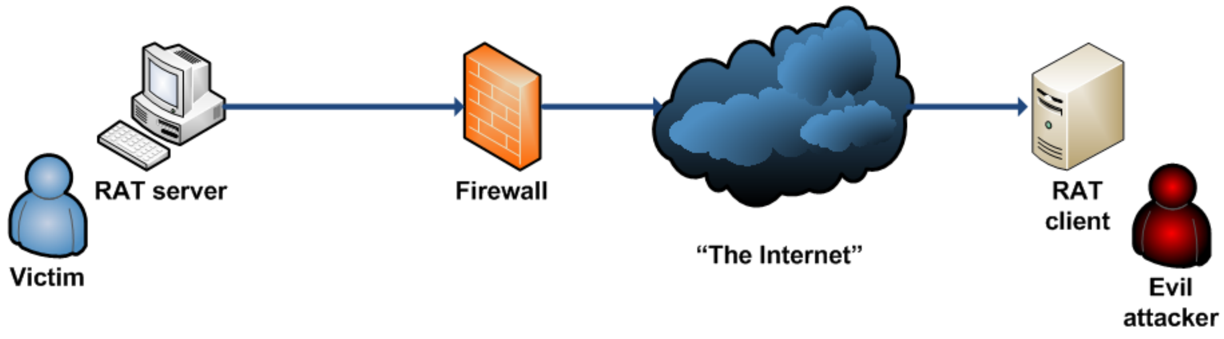
2.7 BlackShades RAT

The FBI has a very particular way of describing this product, [Blackshades RAT] allows criminals to steal passwords and banking credentials; hack into social media accounts; access documents, photos, and other computer files; record all keystrokes; activate webcams; hold a computer for ransom; and use the computer in distributed denial of service (DDoS) attacks [11].

2.7.1 Overview

Blackshades uses TCP as its protocol for communication between its server and client, thus would be connected on a specified port. Furthermore, a secondary port would be used for other recourse hungry features this allowing for a better connectivity between multiple hosts.

The implications that come with this is the limited amount of users that the server is able to handle due to the connection speed. Additionally the user must port-forward to approve connection between both ends. The following diagram shows the architecture used by Blackshades on a One-to-One connection. However the client is able to handle various ‘RAT servers’ connecting to the main ‘RAT Client’.



2.7.2 Utility

As previously mentioned, BlackShades provided an attacker with various tools of which he could take advantage of alongside a plugin system that allowed for custom features. However, it’s one of its most valuable attributes was the amount of data it would gather from a victims computer along with the functions it had to manipulate such.

2.7.3 Scalability

This product is truly illegal due to the modules used for data hijacking, scaling this would add to these miscellaneous features which are exploited by criminals. However, using such a base system such as TCP socket streaming, would allow for the development of a broad software that could handle various features and by following the example of multiple ports, this could additionally allow for greater stability but multiple features can always bring various issues with performance due to bugs.

2.8 Key issues and architectural controversy

2.8.1 Client identification

Using a specific component/part number as a mean of identifying a remote systems is a good approach, however using sensitive data from a computer such as the MAC address, like Andromeda uses, this is a great method of identifying a remote system because a mac address is indeed a unique number [x], which is managed by iEEEE [12] and only specific numbers are given out and all are recorded for ease of identification. However, it brings security issues with such, this due to the ease of a breach by using a MAC address [13]. Regardless, taking a more secure approach to this would be to encrypt the unique identifying number, thus being the MAC, allowing for a legal, more secure way of identifying the system.

2.8.2 Miscellaneous uses

The approach of various developers to include a plugin system is rather risky and would open the core software to various other uses. Not knowing what others may develop as a plugin for your system could have a significant negative impact not just on the way it is used but on the project overall and its reputation. Form-grabbers, key-loggers or any other data gathering tools that do not comply with the computing standards could be possibly implemented as a plugin and used for cybercrime. Therefore, the simple approach to take is to not develop a plugin system, furthermore, blocking down the amount of functions and making sure that none of these can be used as a means of hijacking data. Additionally, inspecting and thoroughly testing the implementation making sure that no workaround of current functions can be exploited to gain such miscellaneous access.

2.8.3 Desktop Streaming

One of the main aims of the project is to add a desktop streaming of some sort in a potential later release, thus this feature is currently too complex to achieve in the time gap given for the projects report, and it still is one of the main outcomes that the project shall work towards.

The use of already premade tools such as VNC or any other JavaScript video streamer could be implemented potentially, however thus shall mean that a higher grade of research into the matter shall be completed and the validation of the feature assessed.

2.9 Conclusions

The use of a web based transport layer would allow for various systems to connect without any other additional requirements, such as port forwarding. Additionally, the portability of the system would also increase as well as compatibility. However, this would not allow for the integration of various features such as remote file management or desktop sharing, thus limiting the amount of web features available, regardless only with cloud-based technologies would the web-desktop streaming possibly work and this seems extremely complex.

3 Requirements Analysis

3.1 Choice of technologies

3.1.1 Protocol

Using the HTTP protocol will allow for world wide access to the dashboard, taking into consideration that its set up accordingly. Furthermore, by using the GET method, we shall be able to hide information witching the header of the request instead of the bod, making our data slightly better secured.

3.1.2 Web-Server

Using Apache as a web server is simple a preference as this is a greatly developed and designed webserver which has been around for some time, troubleshooting such is therefore easier, by following common knowledge posts.

3.1.3 Database

Using an open source database system such as MySQL, allows us to have access to the code and make any changes we require, nonetheless, MySQL has no data limit allowing us to store as much data as we possibly could desire on such, the database will not store much data as it stands, however for future development, a lot of more data will be added to the database along with new version of the database itself.

3.1.4 Dashboard languages

Various developers provide users with already made dashboards, but these sometimes do not meet everyone’s need, as it is my case at the moment. Therefore the panel shall be developed all from scratch using HTML5 and CSS, additionally new responsive technologies shall also be added for cross platform use, therefore a custom panel which was built for this projects sole purpose is therefore preferred.

Additionally, using new technologies will possibly allow for integration of desktop sharing on a web-based platform, however this shall be aimed at in future development cycles.

3.1.5 Use of Third party libraries

As a method to gather the location information based on the bots IP we must loop up the IP provided, however we have no means to do so unless we use an already built API that provides us with such MaxMind does just that, allowing developers to download their pre-built database of location based on their range, this is a free to use.

Furthermore, another API free to use was DynaTable, thus allowed us to display our bots on th dashboard, of which already manages the table data on its own.

3.2 System Requirements

3.2.1 Technological Requirements

The current requirements for the server dashboard panel are:

- Internet Connection
- Web server running,*
    - PHP5 or greater
    - MySQL database system
- Valid Domain name.
- Certified that ports are open, default port is 80

The application side requires:

- Internet connection
- Windows XP or later
- .NET Framework 2.0

3.2.2 Feature requirements

As gathered form the analysis of various other products these are some of the effective objectives or so called requirements that we will work towards achieving, some

- Reliable transport protocol
- Multi user system
- User management
- Intuitive interface
- Allow administrator to distinguish between bots
- Auto connection to panel as per set time
- Web based
- Cross platform
- Perform remote tasks

Any other further requirements that may be brought up for possible future integration shall be weighted using the MoSCoW prioritisation system, furthermore these shall also be analysed to make sure they comply with regulatory laws.

3.2.3 Functionality

The main application must first connect to the gate and thus must identify if the computer is a returning computer or a new installation, after querying the database and doing such check , the gate must identify that the bot is required to do next and set thus as a task by sending it back thought te application connection.

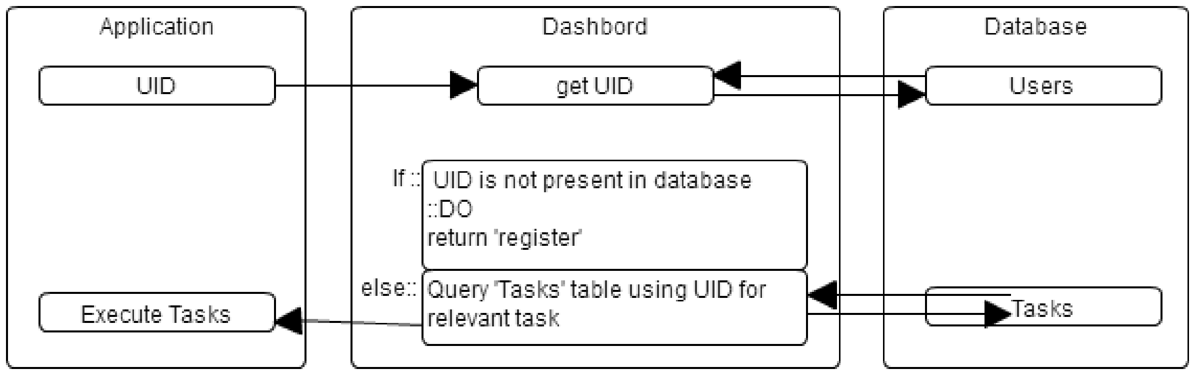
When received the application must device the command and interpreted the current task. When it is a new registration the application must gather information from the computer and re connect to the REST API which will proceed to insert the data into the database and assign its first that to be completed. On the other hand, if the computer is identified as a returning bot the task string shall be interpreted and the task executed, when completed it shall alert the panel of so.

Lastly, the when accessing the dashboard the user must be prompted by a password before he can proceed to add tasks or any other manage any of the users, when validated, the user will be presented with a table of bots followed by the two options of user management and add task. Moreover, the tasks that have been previously added are displayed on the dashboard with the option to add more to the dashboard, thus also applies to the users.

4 SYSTEM DESIGN

4.1 Execution flow

As per the image show, the application and web server must work side by side to achieve the valid outcome of the desired product however to achieve so a set of steps must be followed.



4.1.1 Application steps

Step 1: Gather Dashboard address and port

Step 2: Set the API to connect to gate

Step 3: Connect to set API

Step 4: Start infinite loop for 3 minutes

Step 5: Retrieve given data

Step 6: Split string data

Step 7: Identify task and execute parameter

Step 8: Set task as complete API

Step 9: Connect to API

Step 10: Wait for 3 minutes loop

4.1.2 API Steps

Step 1: Get UID

Step 2: Query database for uid

Step 3: Retrieve the corresponding task (if no record, task will be to register)

Step 4: Send task

Step 5: Get confirmation of execution

Step 6: Update task number in database

Step 7: Await next connection

A more detailed diagram of the connection structure is set in APENDIX C under the name of execution flow, thus represents an easy way to analyse the architecture present.

4.2 Dashboard design

The dashboard was designed for a cross platform usability, thus being allowed to run on various web browsers of which including mobile phones. The design is presented in Appendix D.

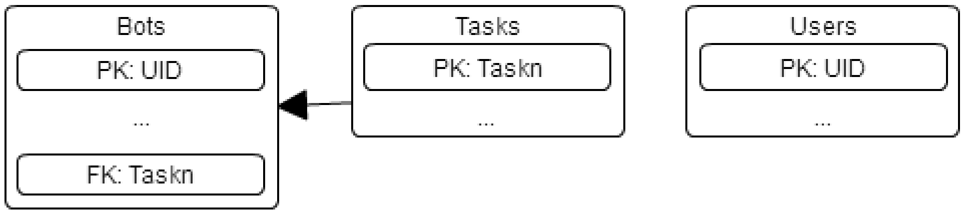
The colour scheme has been set for tree shades of monochromatic followed by a light blue, these having great contrast between each other aiding people who may suffer from visual complications as impairment or colour blindness.

The page layout and its menu has been set on the top right as mist common websites do, these being a set of simple links to pages. Nothing fancy, it was made to just work, as functionality and bug free dashboard was my main objective.

On this note, the login system has been design to prohibit any user to navigate to the page directly without logging in, if do so the individual shall be redirected to the main login page.

4.3 Database design

The only fields necessary to hold the data were limited to the most essential ones, removing any unnecessary fields, therefore the design ended with tree tables of which have a connection of one to many, tasks and bots, and a standalone database for user. The database of course aimed at MySQL allowing for the future development and possible expansion of the current product.

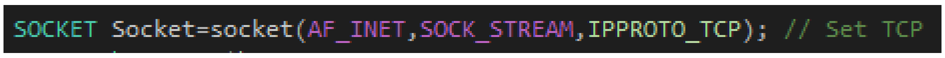


IMPLEMENTATION

As per planned on the design, the product would consist of three programs, these have been implemented successfully however various drawbacks have called for some changes however these do not affect the various features that can be implemented in future development cycles.

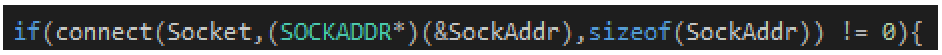
4.4 Socket assembly

The use of an http request would only be possible by assembling a TCP socket possible of sending and receiving the data from the server, as shown below.



```
SOCKET Socket=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP); // Set TCP
```

Additionally the socket must also gather an address has been hard coded into the application, the address is in its URL/Domain format and thus must be translated to its IP by a hostname lookup. Following all the socket pre-requisites, the connection is attempted as per the figure bellow



```
if(connect(Socket,(SOCKADDR*)(&SockAddr),sizeof(SockAddr)) != 0){
```

Additionally, the connect function allows us to check if it has truly connected, at this stage the header must be constructed and sent to the web server using the socket just created.

After the header data is sent and completed, the data from the server must be retrieved, thus is done by using the same socket, as it is streamed into a buffer on the application, when completed the buffer is closed with the ‘\0’ char.

The full example code can be found in Appendix E followed by an example of the header construction.

4.5 Header construction

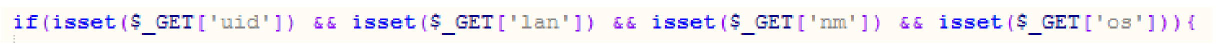
When the socket is completed the HTTP Header must be constructed, this is where the data sent to the API is held, and this is sent to the API as a complete formatted string as shown bellow

?0=[uid]&1=[ip]&2=[hostname]&3=[winos]&4=[lanip]

As presented above, any data being send is gathered form the local machine and inserted into its corresponding filed located in the header parameters, thus these parameter not only take the local computer information it also sends other parameters such as the time to maintain the connection open to the type of data being sent but most importantly the type of method that it is using, in this case we are using the GET method.

4.6 API Management

The current API takes the uid as a method of authentication, thus also using it as a string to query the database due to being a set unique attribute. The data is send as per the following figure



```
if(isset($_GET['uid']) && isset($_GET['lan']) && isset($_GET['nm']) && isset($_GET['os'])){
```

It then uses a webserver method to gather any other data required such as the ip and calculate its location by querying the database.

4.7 Database connection

The current database is the link between the dashboard and the API, thus also being the pain heart of the data as it is storing all the crucial information, the approach to a the database was that it would not become redundant nor would it provide redundant data. Additionally it could not register various users under the same name, therefore user number were auto incremented and some given unique attribute names as shown bellow:

```
Create table if not exists bots(
       uid varchar(13), […]
       PRIMARY KEY (uid));
```

The connection to the database has been created on a single file located on the ‘configs’ folder, thus allowing for various calls to be made by different files. The connection was between the database and dashboard was also set so that these could both be put on separate servers and still be able to connect and manage itself, could this be the API or the standalone dashboard.

4.8 Technical issues

4.8.1 API

The original design would state a gate file and register file on the API, however the addition of tasks to the API will beneficiate the API for ease of use. Thus because the application was connecting several times to the API and registering itself every time it would like to perform a task.

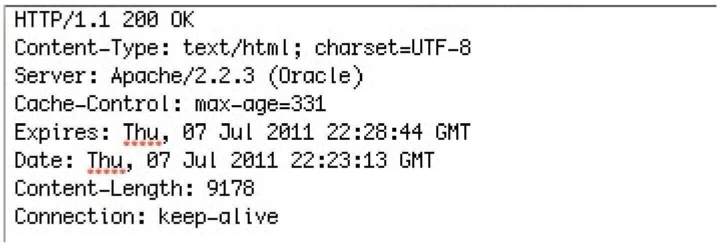
The task manipulation issue has been overcome by adding this extra section to the API. Additionally this also improved the overall performance of task handling, allowing various users to perform task at the same time, something similar to a distributed task system.

4.8.2 Application

Application wise, thus has become a challenge to control and locate various of the function used to identify and gather data from the bot machine, therefore the use of classes and structuring the code accordingly has improved productivity and core usability overall. Never the less the code itself burn challenges.

4.8.3 Request data format

Different web servers along with various other versions of the same, may sometimes act differently depending on the request being sent, one of the most challengeable tasks when developing has come down to the design of the HTTP headers, thus used to transport information between the application and the API. As seen on the figure, the web request originally used was spamming out irrelevant data, this seriously breaking our code when we tried to interpreted such.



```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Server: Apache/2.2.3 (Oracle)
Cache-Control: max-age=331
Expires: Thu, 07 Jul 2011 22:28:44 GMT
Date: Thu, 07 Jul 2011 22:23:13 GMT
Content-Length: 9178
Connection: keep-alive
```

The use of the code '\r\n' at the end of each header parameter, seen as:

```
…"Accept: text/html\r\n Accept-Charset: utf-8\r\n Keep-Alive: 300\r\n\r\n\r\n\r\n"
```

Thus, has allowed me to only send and receive header information essential to the project, ruling out any other data that was being spammed by the web server and return to the application. This allowed me to retrieve a clean string of data that allowed to split such using a delimiter and perform the commands according.

4.9 Windows OS Version

Gathering this via C++ as a challenge for myself as thus is not written nowhere nor could I find a guide or had done it before, however by using the operating systems version number I can roughly figure out what the name of the current operating system by using the format of the table below, the full table can be found on appendix F.

Operating system	Version number	dwMajorVersion	dwMinorVersion
Windows 8.1	6.3*	6	3
Windows 8	6.2	6	2
Windows Server 2012	6.2	6	2
Windows 7	6.1	6	1

4.10 Integration

Each of the features were testes one by one before integration, these were designed and tested separately, then added to the main project under its corresponding class, the development met the design and no significant changes were made apart from an additional part on the API to handle the task on its sole. Thus providing a better platform for task handling.

5 LEGAL, SOCIAL, ETHICAL AND PROFESSIONAL ISSUES

5.1 Legal

When sending information around via a local or global internet link it must be encrypted or it may fall into the wrong hands, this including sensitive data such as the one we are sending.

The Data Protection Act 1998, implies that the company must have a secure system when sending personal information and keep it secure from outside sources. On top of the Data Protection Act 1998, the Computer Misuse Act 1990 requires that by using such a system, that you do not use it for miscellaneous outcomes and prevent anyone trying to achieve such not to do so.

Security and encryption, because we are sending data between two devices, a secure approach must be taken to data by providing encryption, this does not just protect the application against out sources trying to affect the system or steal computer or user information but it keeps the entire infrastructure secure and pointless to hack into.

The project identified the mac address as its unique identifying number, however the use of such must be done appropriately as thus can me miscellaneous, to overcome such the use of encryption or encoding must be done to make sure that the current mac address is not the raw output being sent to the dashboard.

Additionally, encrypting dashboard passwords with the latest encryption techniques, thus of MD5, pepper and salt, making the web dashboard safe from password leaks, but teaching administrators not to provide this password to other users comes in another note.

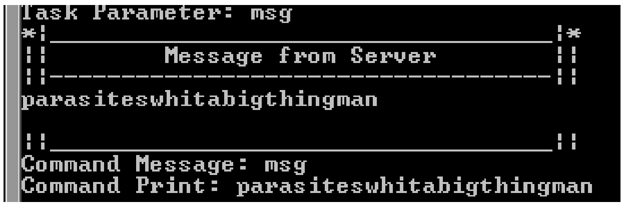
5.2 Maintenance

Maintain the base core of this software would be easy however will cost effective, thus brings the need of an individual with knowledge of various technologies, thus who can possibly teach those who struggle with the system.

Getting the system up and running is one thing and it can be done by the specialist that understand what they are doing and are familiar with the systems, however you must take into consideration if your current personnel(organization employees) will be able to adapt to the system and use it.
If your current staff are not able to use the new system you must take one of two option, educate them on how to use it or get new staff that understands and can use the new system.

6 EVALUATION

6.1 Application



```
Task Parameter: msg
*|________________________________|*
||        Message from Server       ||
||----------------------------------||
parasiteswhitabigthingman

||__________________________________||
Command Message: msg
Command Print: parasiteswhitabigthingman
```

The debug mode has been set on the application and as testing it all the data was printed to the screen thus following the steps it should take, as seen on the present image, the task has been set as an 'msg' thus stand for message, and this will print the message that comes from the server. We can see that it did just so.

Therefore, thus being the ultimatum of the application and its final outcome, we can assure that the application has run successsfully with no errors.

6.2 Dashboard

The dashboard has been tested against 'link hopping' and is fully locked down and secure, thus means that a user cannot enter the direct path to a page without having the correct authentication. If done so he shall be relayed back to the login page.

6.2.1 Task and User management

Both these features have been testes, information required to be displays as so and so does the addition of new data to the database using these set form. Overall success with no bugs to report.

Any further testing is available in appendix G, thus with picture evidence and a brief description of what should happen and what did. Moreover, the end product was fully bug free as tested on various operating systems thus from tree different machines and two different locations.

7 CONCLUSION

The overall development of the project has been a challenge yet a joyful one, on the other hand, writing report to an academically standard has always been my main struggle, though this project has helped me on both I believe I could have done better on presenting my findings,. Regardless the information given in the report has stated the main use of similar products and the online they take, thus the way we have managed to do modify the view and uses of such products, especially the ones used by fanatical cybercriminals.

During the development cycle, implementing the products was a great joy and I took great pride in using the tool, watching it work, overviewing its full potential.

7.1 Personal evaluation

The project has been a personal challenge I have set myself, nonetheless I consider the first development cycle a complete success, one from which I shall keep on developing and learning from.

As a single individual I was able to evaluate and create a functional system from pre-set aims that did not only challenge me intellectually, it challenged my everyday life as part of my personal development and time management, thus as I am working full time alongside university.

The main conclusion taken by myself is that the project has been a valuable experience as it developed my skill along with broadening my experience in development; challenging yet extremely enjoyed carrying out this project.

7.2 Project journey

Since a young age I have been interested in information gathering, and so was my previous project mentor, Trojans horses to be precise, have always been my main focus and their use for remote management, evolving these to botnets as a better way to perform task remotely with ease of access via a web dashboard, however features that I would love to use have never been implemented due to the lack of technology, at this day and age, it is possible now and I shall try to develop these myself.

A web based remote administration tool that provides a system administrator with a desktop streaming is my main objective from the project, any additional features come afterwards, however I must prioritise them and include them not just for a good foundation to achieve the ultimatum but to expand my skill as I do such.

Moreover, when the project is fully functional and secured to a vast majority I shall make sure to release it to the public, not doing so if it shall imply the illegal use, therefore research into the legality of the product shall be carried out further.

7.3 Future development

The integration of a desktop streaming feature would increase the products reputation, thus as there is no other software used for remote monitoring using this technology or possibly I haven’t come across it. Regardless, it is a complex part of the end product and it shall be implemented in future development.

The products as stands can be implemented into various system, used as a core connection, this can be sold or adapted to various customer’s needs. It is currently on its simplest form of a register and simple task execution. However, adapting and adding features shall be intuitive as the code is structured in a developer friendly manner.

This does not make it any legal as seen throughout the report, products as such can be used for cyber-crime and given the opportunity someone would do so, therefore one of the main objectives is to make sure that only specific tasks can be completed and encrypting or obfuscating them to a level that no other possible individual could alter such.

7.4 Project conclusion

Though it can be used for illegal purposes, botnets have been looked down by many individuals as a tool of evil due, however using its technologies for good such as mass management of system can shine some light on the authentic use of these tools.

Implementing legal, valid, meaningful features into a product can mean that various other products will be rendered obsolete and a tool as the one being researched and developed can and will bring a vast amount of interest due to its aim and potential.

Lastly, the projects is balanced in a very thin line of legality and usability, it shall be great when finished and able to manage various end computers, however this must not become a product of miscellaneous use nor can any of its implemented features be able to be used for such.

REFERENCES

[1] Kritika Govind; S. Selvakumar; Auto-Pattern Programmable Kernel Filter (Auto-PPKF) for Suppression of Bot Generated Traffic”, I.J. Computer Network and Information Security, 2014, 1, IJCNIS Vol. 6, No. 1,pp. 48-54, November 2013

[2] Jae-Seo Lee; Hyuncheol Jeong; Jun-Hyung Park; Minsoo Kim; Bong-Nam Noh, "The Activity Analysis of Malicious HTTP-Based Botnets Using Degree of Periodic Repeatability," in Security Technology, 2008. SECTECH '08. International Conference on , vol., no., pp.83-86, 13-15 Dec. 2008

[3] [online] http://www.malwaretech.com/2013/12/peer-to-peer-botnets-for-beginners.html

[4] [online] http://www.symantec.com/connect/blogs/international-takedown-wounds-gameover-zeus-cybercrime-network

[5] [online] http://www.darkreading.com/attacks-and-breaches/data-breach-persistence-gives-hackers-the-upper-hand-/d/d-id/1114124

[6] [online] https://developer.join.me/docs

[7] [online] http://www.codingthearchitecture.com/ Posted by Simon Brown on 29 March 2016 09:23:00 BST #

[8] [online] http://www.hackmantra.com/2013/06/how-to-setup-botnet-video-tutorial.html

[9] [online] http://resources.infosecinstitute.com/andromeda-bot-analysis/

[10] [online] https://bitdepth.thomasrutter.com/2010/04/02/stable-vs-stable-what-stable-means-in-software/

[11] [online]
https://www.fbi.gov/news/stories/2014/may/international-blackshades-malware-takedown/international-blackshades-malware-takedown

[12] [online] http://www.networkworld.com/article/2340364/lan-wan/are-mac-addresses-really-unique-.html

[13] [online] http://www.howtogeek.com/204458/why-you-shouldn%E2%80%99t-use-mac-address-filtering-on-your-wi-fi-router/

APPENDIX A - ORIGINAL PROJECT SCHEDULE

Original schedule

Revised schedule

Have several appendices (A,B,C…). These should include detailed and technical documentation such as table of results, diagrams, program source code, etc, which are essential parts of the project but not directly a part of the main discussion in the report. All contents of appendices should be exclusively, products of the student's own work.

Other materials used during the project work (such as information from user manuals, interview notes, etc), which it is necessary to include, should if possible be summarized to only a few pages before entering into the appendix. Original copies of such material should be kept by the student and may be required to be produced as supporting evidence of their work.

Examples of key coding files may be provided in an Appendix but generally it should be uploaded via teachmat or on the P drive with its associated software.

APPENDIX B - USER DOCUMENTATION

Setup requirements

The current requirements for the server side panel are:

- Internet Connection
- Web server running,*
  - PHP5 or greater
  - MySQL database system
- Valid Domain name.
- Certified that ports are open, default port is 80

The application side requires:

- Internet connection
- Windows XP or later
- .NET Framework 2.0**

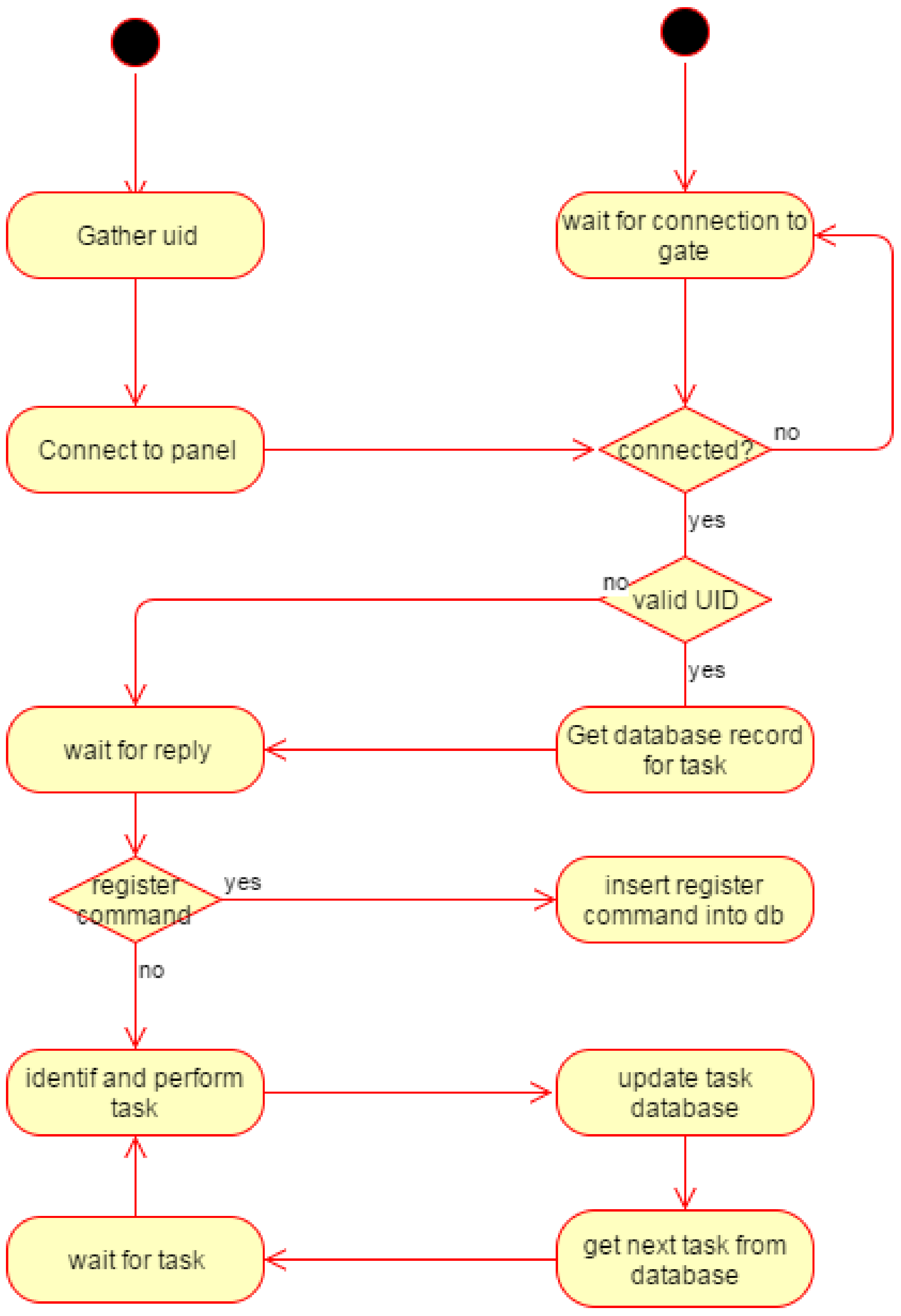
*At least 100MB of disk space is required, the database is surely to need much more space but as always this depends on how much user data you will be hosting, 50MB minimum is recommended. Furthermore, this tool also requires a minimal amount of bandwidth keeping a track record of so is crucial to prevent unexpected charges.
**The malware is currently coded in a native language however it still uses some parts of dependencies on the .NET framework) this will be removed in future updates according to development cycles.

Setup

Server side:

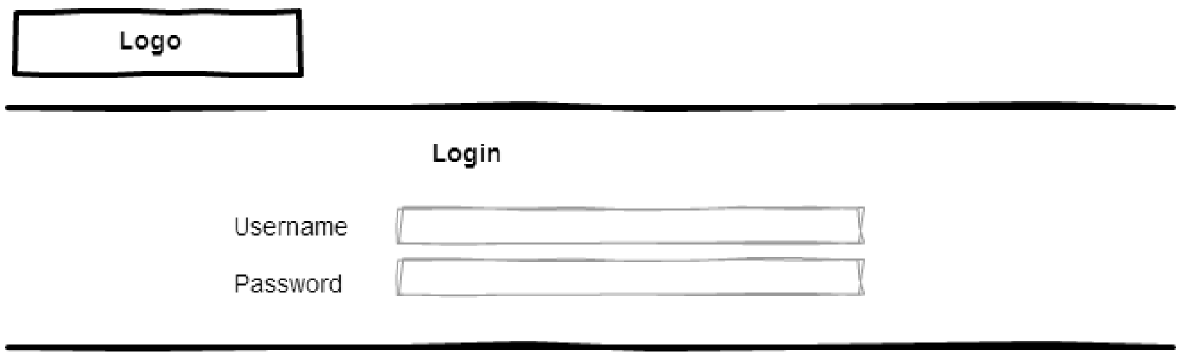
- Make sure that the required software is running correctly, if so, connection between the domain and web host must be established(if required) therefore adding the name servers of the web host is a crucial task as all of the connection will be using the domain to connect.
- After making sure that the domain and server side are fully functional proceed to use a FTP client of choice to upload the contents of the panel the web server.
- Navigate to the dashboard, this being:
  - 'HTTP://YourDomain/location_uploaded/'
- To login the default credentials are:
  - Username: admin
  - Password: admin

APPENDIX C - EXECUTION FLOW

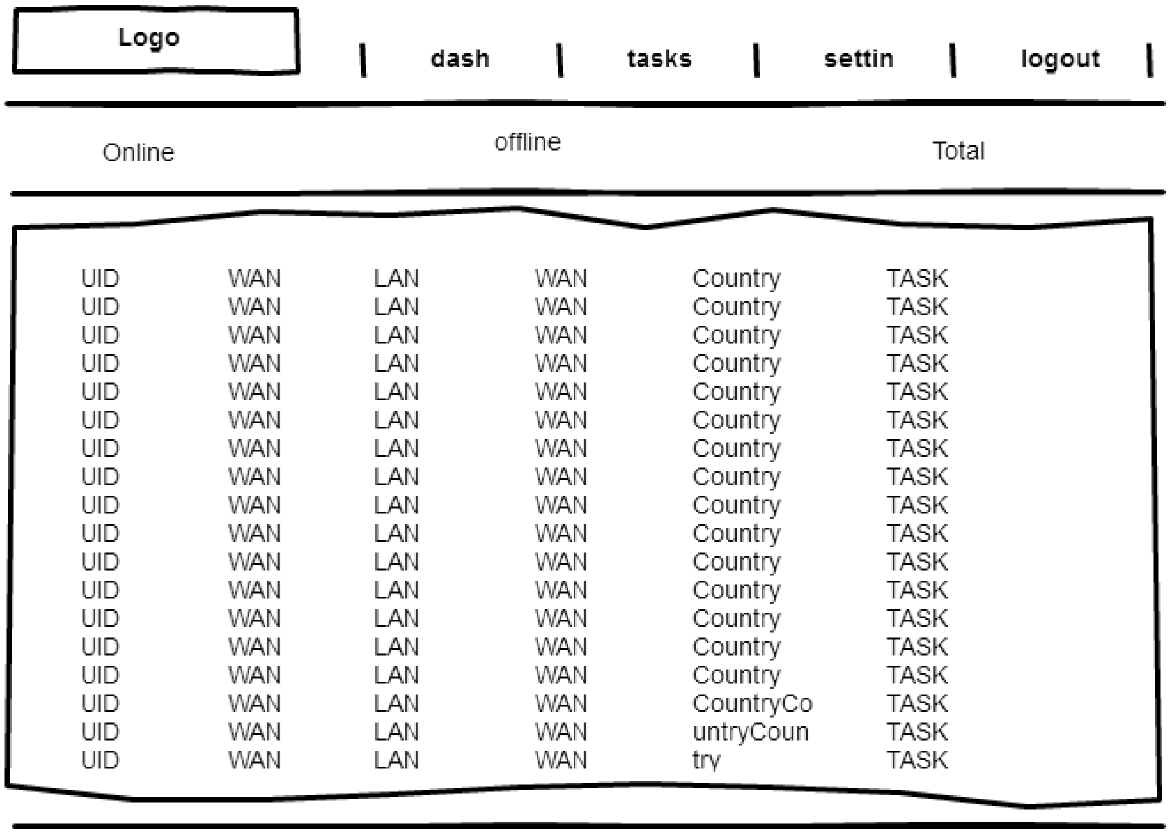


APPENDIX D – DASHBOARD DESIGN

Login

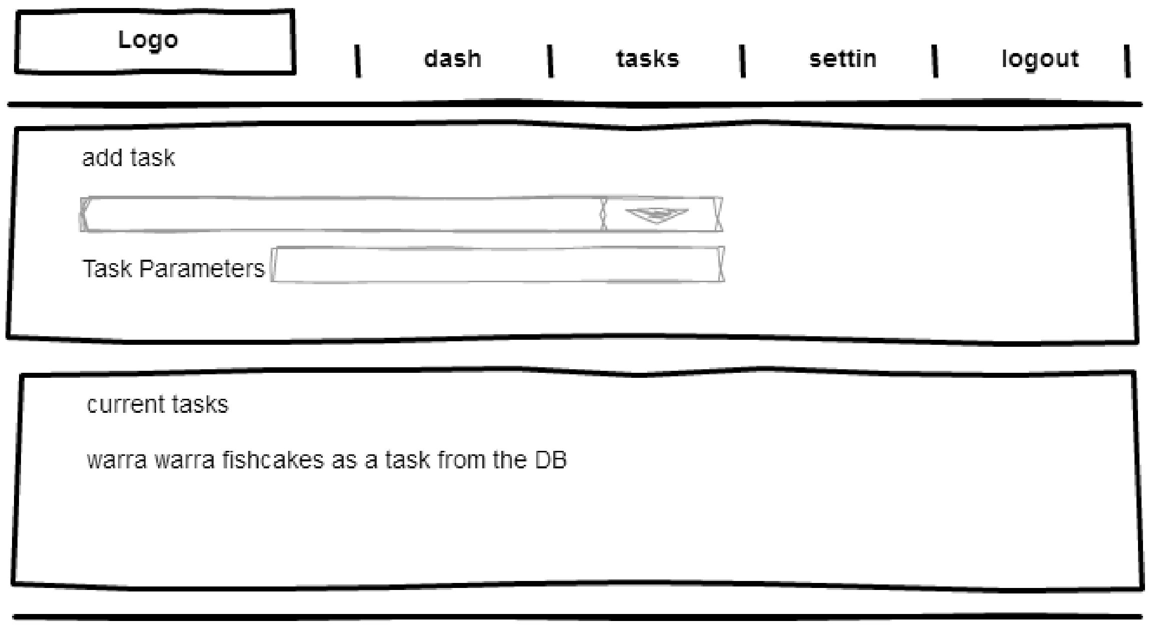


Dashboard

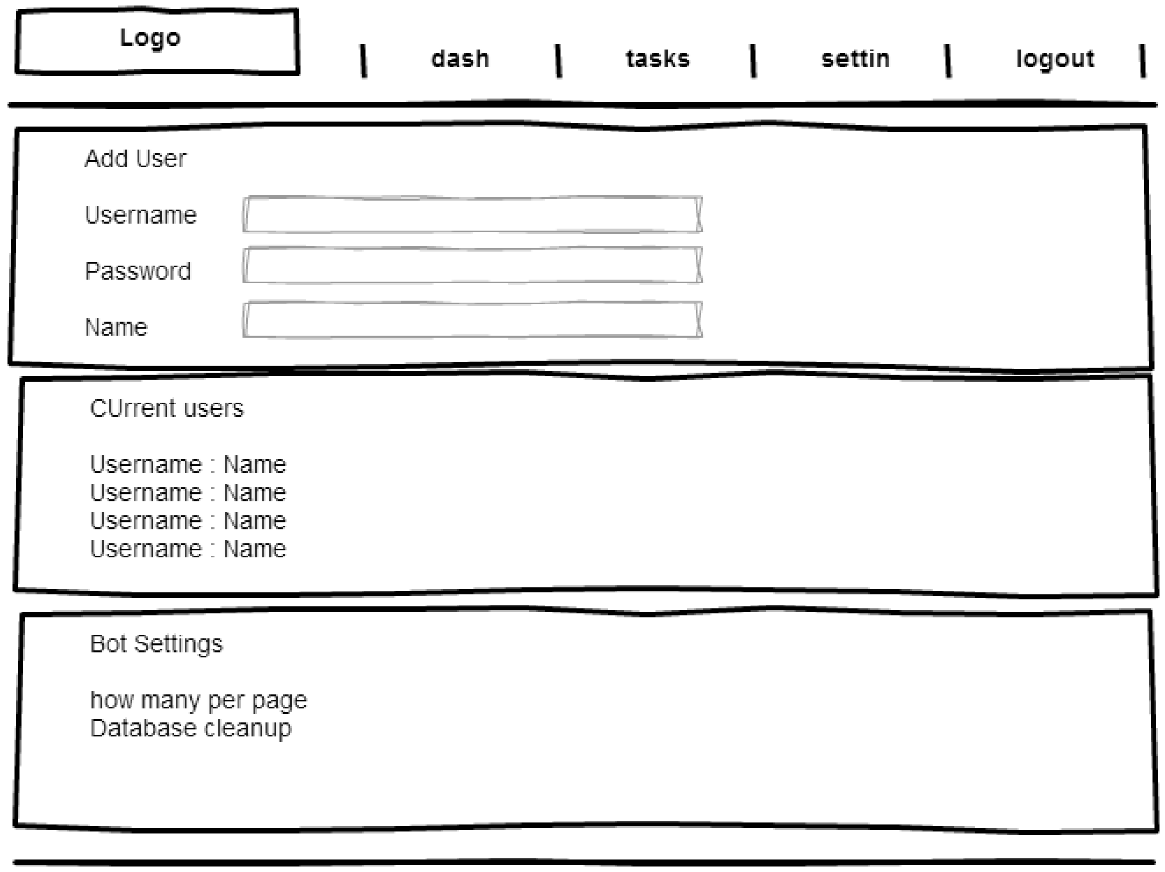


UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	Country	TASK
UID	WAN	LAN	WAN	CountryCo	TASK
UID	WAN	LAN	WAN	untryCoun	TASK
UID	WAN	LAN	WAN	try	TASK

Tasks



Settings



APPENDIX E – SOCKET CONNECTION

API Construction

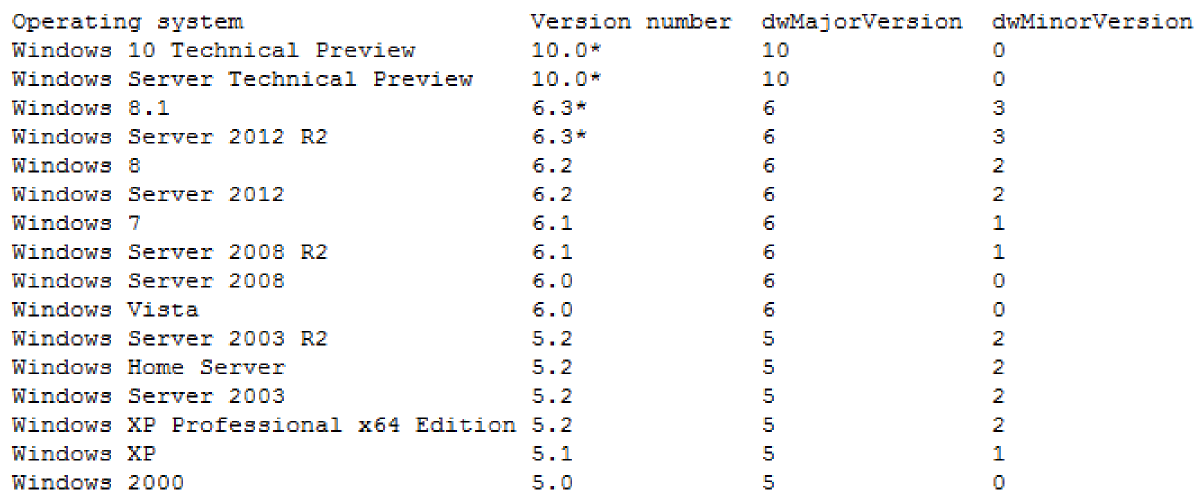
```
char* httprp::post_reg(char* wp){
	std::stringstream ss;
	Machine_inf lcl_inf;
	char* xx= "";
	ss << wp
	<< "register.php?uid="
	<< lcl_inf.getMAC()
	<< "&lan="
	<< lcl_inf.lan_ip()
	<< "&nm="
	<< lcl_inf.host_name()
	<< "&os="
	<< lcl_inf.OS_version();

	std::string request = ss.str();
	xx = strdup(request.c_str());

	return xx;
};
```

HTTP Request

```
char* httprp::getrp(char* Domain_name,int Port_no, char* Head_api){

	WSADATA wsaData;
	if (WSAStartup(MAKEWORD(2,2), &wsaData) != 0) {
		//cout << "WSAStartup failed.\n";
		//system("pause");

		return "WSAStartup failed.";
	};

	SOCKET Socket=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP); // Set as TCP
	struct hostent *host;
	host = gethostbyname(Domain_name);

	SOCKADDR_IN SockAddr;
	SockAddr.sin_port=htons(Port_no);
	SockAddr.sin_family=AF_INET;
	SockAddr.sin_addr.s_addr = *((unsigned long*)host->h_addr);

	if(connect(Socket,(SOCKADDR*)(&SockAddr),sizeof(SockAddr)) != 0){
		return "Could not connect";
	}

	std::stringstream ss;

	ss << "GET "<< Head_api << "\r\n HTTP/1.1\r\n"
```

```
      << "Host: "<< Domain_name << ""
      << "User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5)
Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)\r\n"
      << "Accept: text/html\r\n"
      << "Accept-Charset: utf-8\r\n"
      << "Keep-Alive: 300\r\n"
      << "Connection: close\r\n"
      << "Pragma: no-cache\r\n"
      << "Cache-Control: no-cache"
      << "\r\n\r\n\r\n";

    std::string request = ss.str();

      if (send(Socket, request.c_str(), request.length(), 0) != (int)request.length())
{
return "Error sending request";
      }

      char buffer[10000];
      int nDataLength;
      char ret[10000];
      while ((nDataLength = recv(Socket,buffer,10000,0)) > 0){
            int i = 1;
            int j =0;
            while (buffer[i] >= 32 || buffer[i] == '\n' || buffer[i] == '\r') {

            if(isalnum(buffer[i])){
                  ret[j] = buffer [i];
                  i++;
                  j++;
            }else if(buffer[i] == ';'){
                  ret[j] = buffer [i];
                  i++;
                  j++;
            }else{
                  i++;
            }
            }
            buffer[i] = '\0';
            ret[j] = '\0';
            // finish the string at the end of the transmission, if not
            // used block will be displayed from || char buffer[10000];
      }
      closesocket(Socket); // close socket.
      WSACleanup();//always cleanup!

      return ret;
}
```

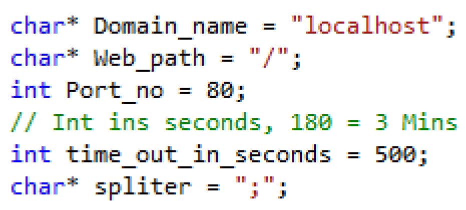
APPENDIX F – WINDOWS VERSION CONVERSION



Operating system	Version number	dwMajorVersion	dwMinorVersion
Windows 10 Technical Preview	10.0*	10	0
Windows Server Technical Preview	10.0*	10	0
Windows 8.1	6.3*	6	3
Windows Server 2012 R2	6.3*	6	3
Windows 8	6.2	6	2
Windows Server 2012	6.2	6	2
Windows 7	6.1	6	1
Windows Server 2008 R2	6.1	6	1
Windows Server 2008	6.0	6	0
Windows Vista	6.0	6	0
Windows Server 2003 R2	5.2	5	2
Windows Home Server	5.2	5	2
Windows Server 2003	5.2	5	2
Windows XP Professional x64 Edition	5.2	5	2
Windows XP	5.1	5	1
Windows 2000	5.0	5	0

APPENDIX G – TESTING

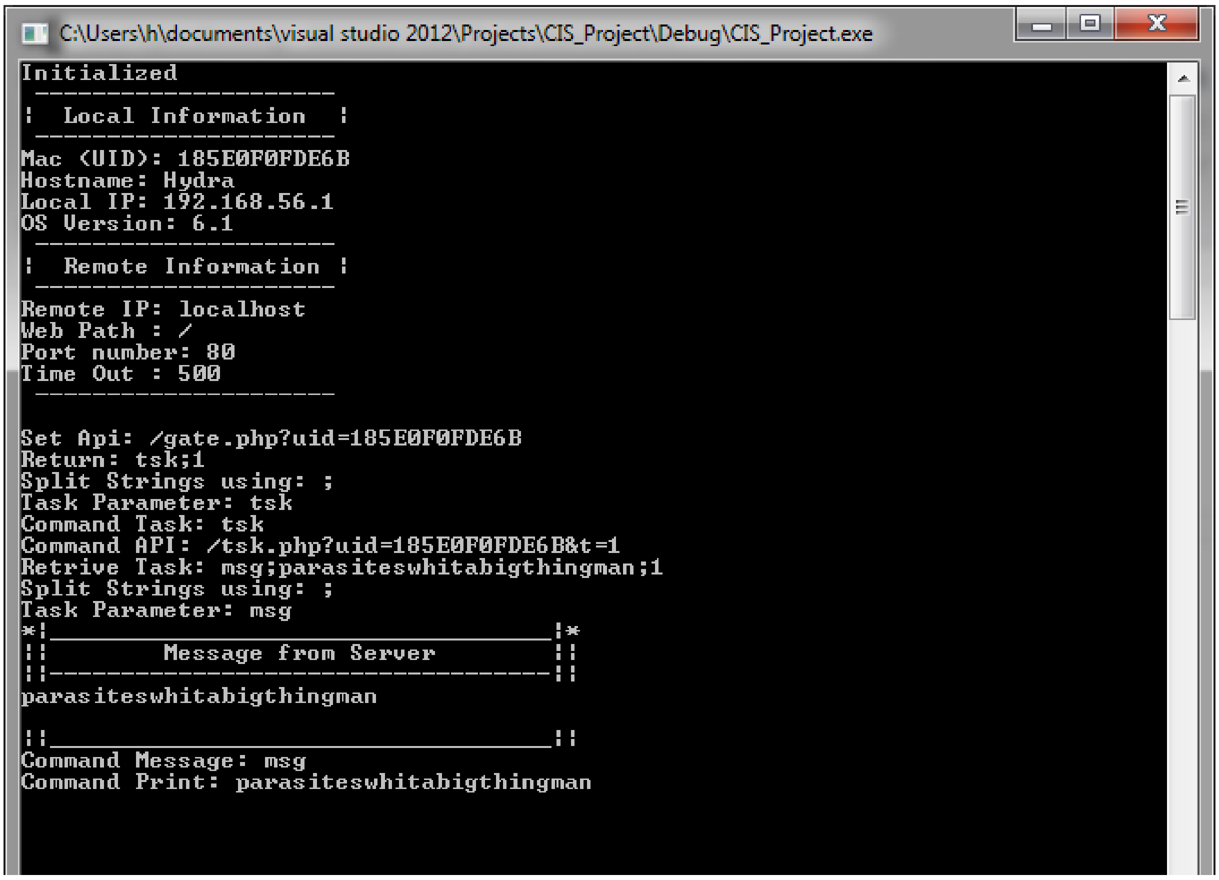
Application Configuration



```
char* Domain_name = "localhost";
char* Web_path = "/";
int Port_no = 80;
// Int ins seconds, 180 = 3 Mins
int time_out_in_seconds = 500;
char* spliter = ";";
```

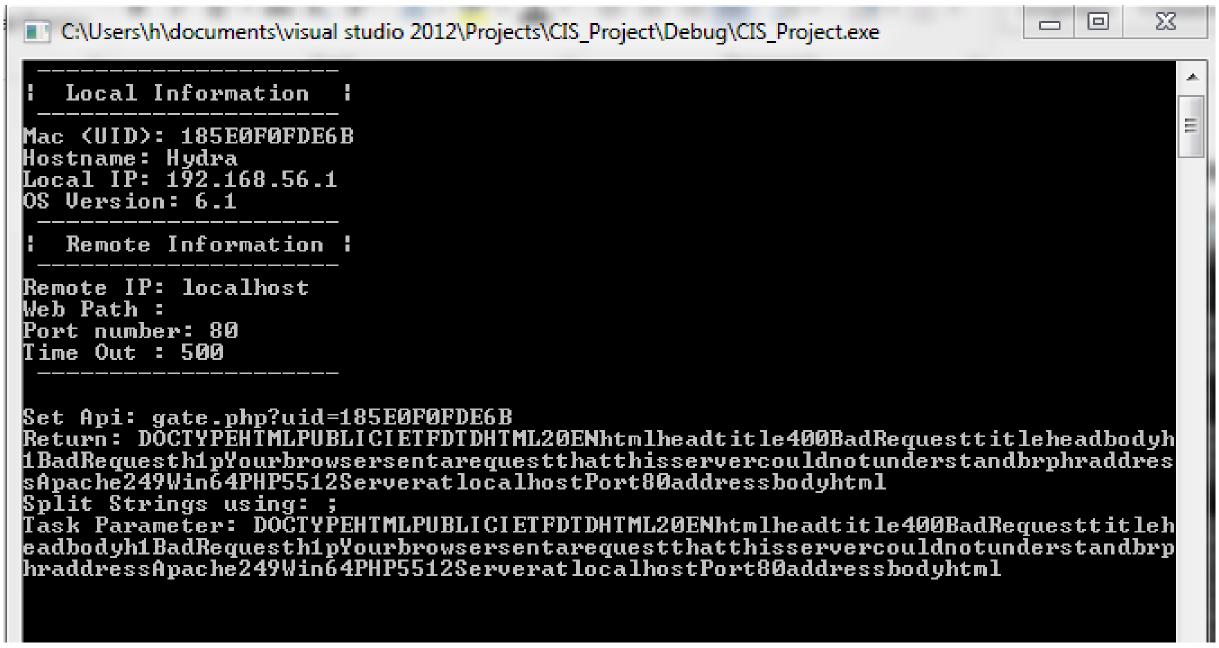
Application Execution

The following is of the application running smooth and correctly, thus being able to connect to the main panel, also proceeding into executing the task.


C:\Users\h\documents\visual studio 2012\Projects\CIS_Project\Debug\CIS_Project.exe

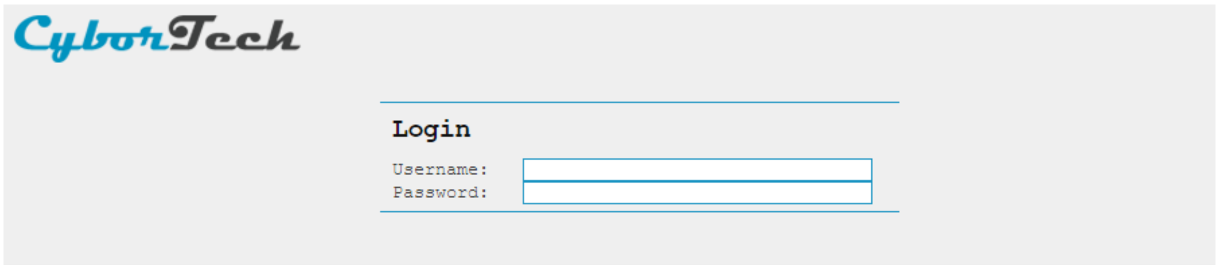
```
Initialized
 ----------------------
|  Local Information  |
 ----------------------
Mac (UID): 185E0F0FDE6B
Hostname: Hydra
Local IP: 192.168.56.1
OS Version: 6.1
 ----------------------
|  Remote Information |
 ----------------------
Remote IP: localhost
Web Path : /
Port number: 80
Time Out : 500
 ----------------------

Set Api: /gate.php?uid=185E0F0FDE6B
Return: tsk;1
Split Strings using: ;
Task Parameter: tsk
Command Task: tsk
Command API: /tsk.php?uid=185E0F0FDE6B&t=1
Retrive Task: msg;parasiteswhitabigthingman;1
Split Strings using: ;
Task Parameter: msg
*|___________________________________|*
||          Message from Server      ||
||-----------------------------------||
parasiteswhitabigthingman

||___________________________________||
Command Message: msg
Command Print: parasiteswhitabigthingman
```

However if the connection to the API had failed the outcome would be as follows:


C:\Users\h\documents\visual studio 2012\Projects\CIS_Project\Debug\CIS_Project.exe

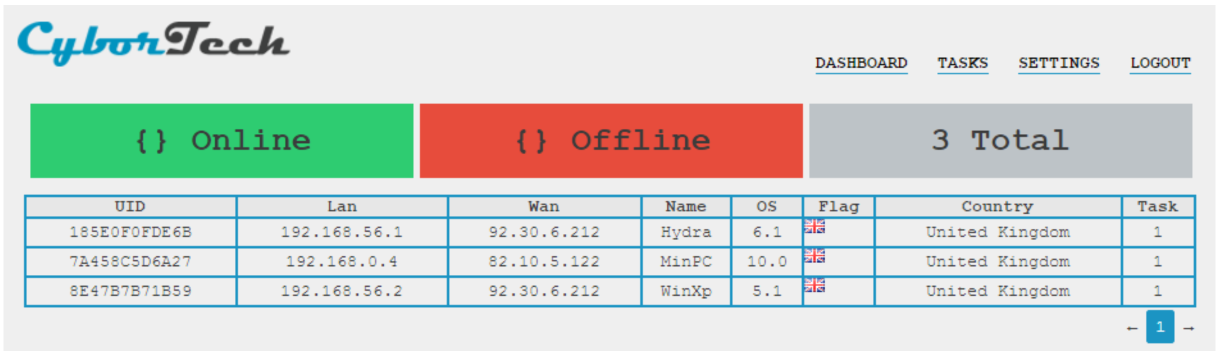
```
 ----------------------
|  Local Information  |
 ----------------------
Mac (UID): 185E0F0FDE6B
Hostname: Hydra
Local IP: 192.168.56.1
OS Version: 6.1
 ----------------------
|  Remote Information |
 ----------------------
Remote IP: localhost
Web Path :
Port number: 80
Time Out : 500
 ----------------------

Set Api: gate.php?uid=185E0F0FDE6B
Return: DOCTYPEHTMLPUBLICIETFDTDHTML20ENhtmlheadtitle400BadRequesttitleheadbodyh
1BadRequesth1pYourbrowsersentarequestthatthisservercouldnotunderstandbrphraddres
sApache249Win64PHP5512ServeratlocalhostPort80addressbodyhtml
Split Strings using: ;
Task Parameter: DOCTYPEHTMLPUBLICIETFDTDHTML20ENhtmlheadtitle400BadRequesttitleh
eadbodyh1BadRequesth1pYourbrowsersentarequestthatthisservercouldnotunderstandbrp
hraddressApache249Win64PHP5512ServeratlocalhostPort80addressbodyhtml
```

Dashboard login



As per mentioned on the user documentation, the use of the username admin and password admin grants access to the panel, any direct link entry is nulled and the user is bright back to this login page, these features have been testes and are working.

Dashboard

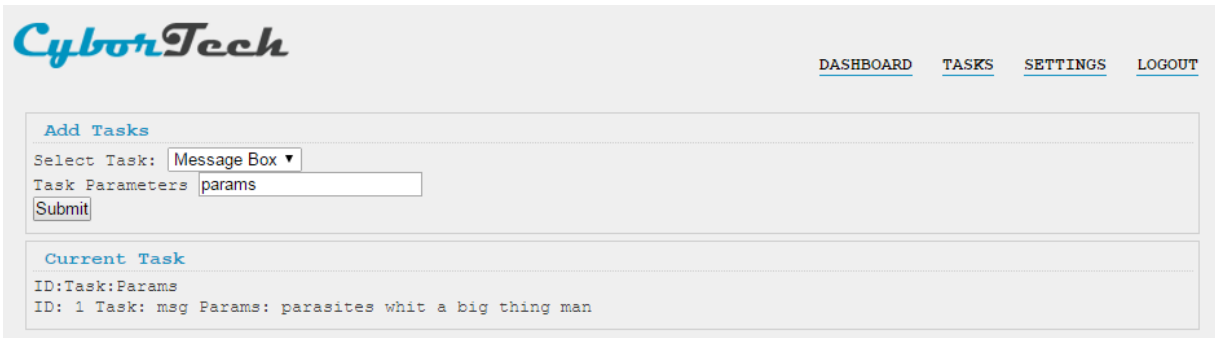


UID	Lan	Wan	Name	OS	Flag	Country	Task
185E0F0FDE6B	192.168.56.1	92.30.6.212	Hydra	6.1		United Kingdom	1
7A458C5D6A27	192.168.0.4	82.10.5.122	MinPC	10.0		United Kingdom	1
8E47B7B71B59	192.168.56.2	92.30.6.212	WinXp	5.1		United Kingdom	1

The dashboard has the task of displaying all of the data from the bots table, thus being the data of the bots themselves, the page does just that with the addition of identifying the country and placing if correct flag by the side of the bot, additionally it also displays the current amount of entries in the database.

By using the third party table controller, Dynatable, it has been possible to arrange the data in the table according to what we would like, thus from alphabetical order or numerical, this is achieved by pressing any of the table header as these act as the controller.

Task page



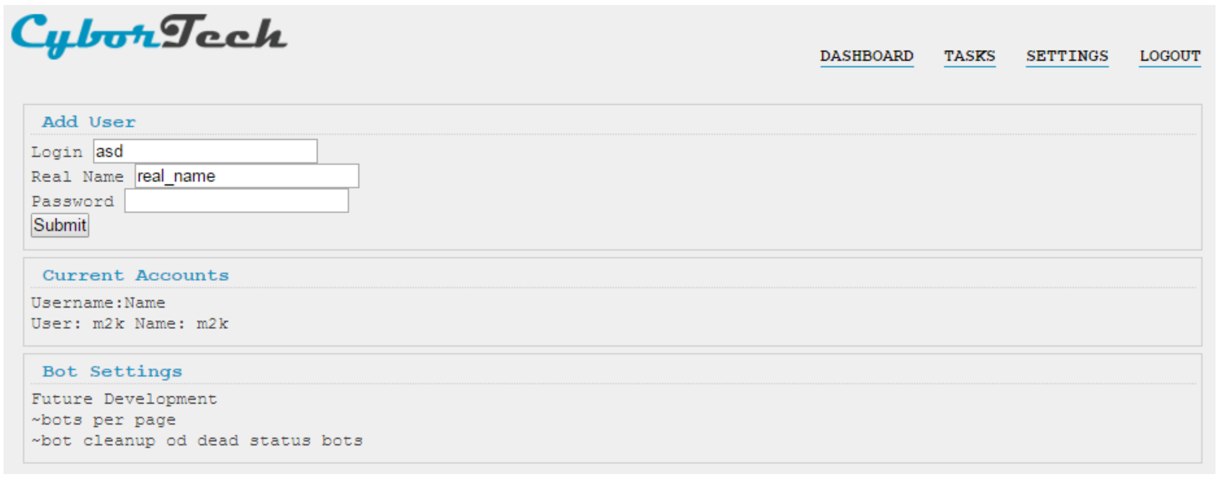
Add Tasks
Select Task: Message Box
Task Parameters params

Current Task

```
ID:Task:Params
ID: 1 Task: msg Params: parasites whit a big thing man
```

Though this page is simple for what it currently displays and set, the queries running behind such to prevent any invalid data from being entered is a great for the moment being, the page as loads should display current tasks within the table and give the option to add a new one to so, both these functions are fully functional.

Settings page



Add User
Login asd
Real Name real_name
Password

Current Accounts

```
Username:Name
User: m2k Name: m2k
```

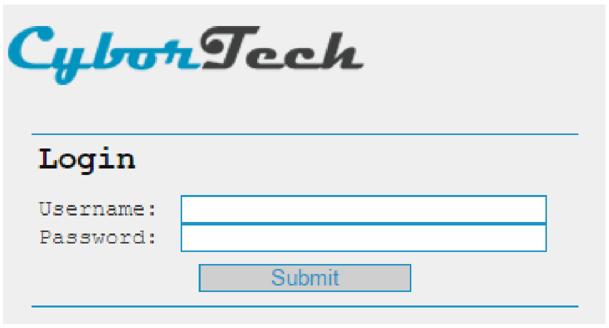
Bot Settings

```
Future Development
~bots per page
~bot cleanup od dead status bots
```

User management was a required feature, here it is presented, the page should load and display any name and username of any user in the database, thus also allowing the addition of another one. Additionally the bot setting has been left for any future development.

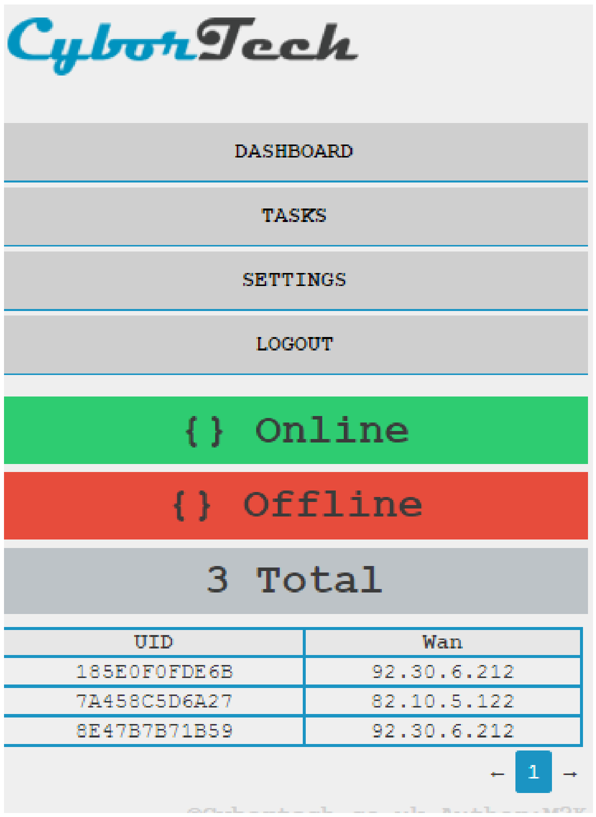
Mobile

Login



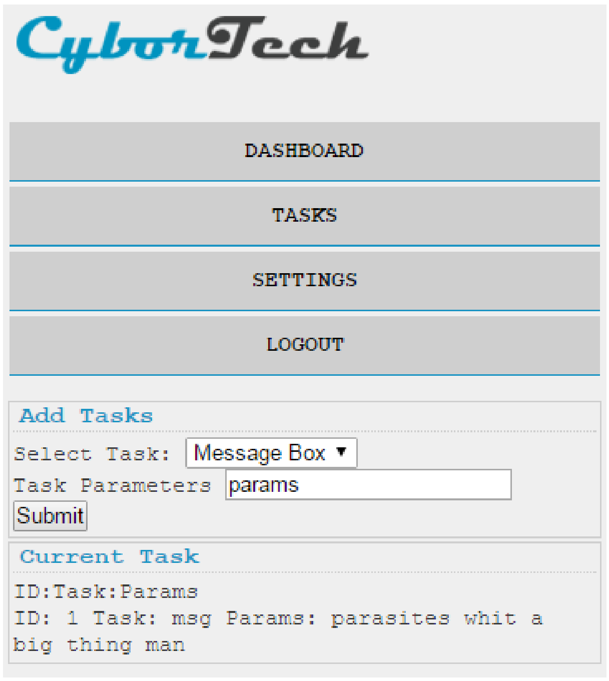
Username:	
Password:	

Dashboard



UID	Wan
185E0F0FDE6B	92.30.6.212
7A458C5D6A27	82.10.5.122
8E47B7B71B59	92.30.6.212

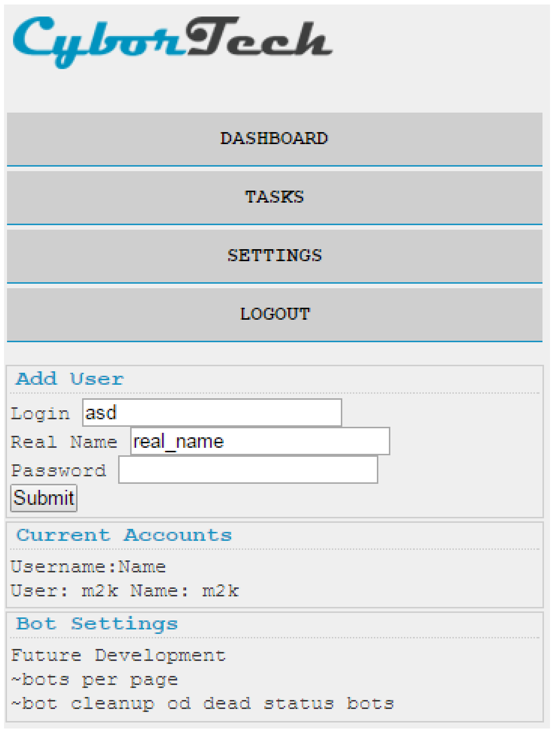
Tasks



Select Task:	Message Box
Task Parameters	params

```
ID:Task:Params
ID: 1 Task: msg Params: parasites whit a
big thing man
```

Settings



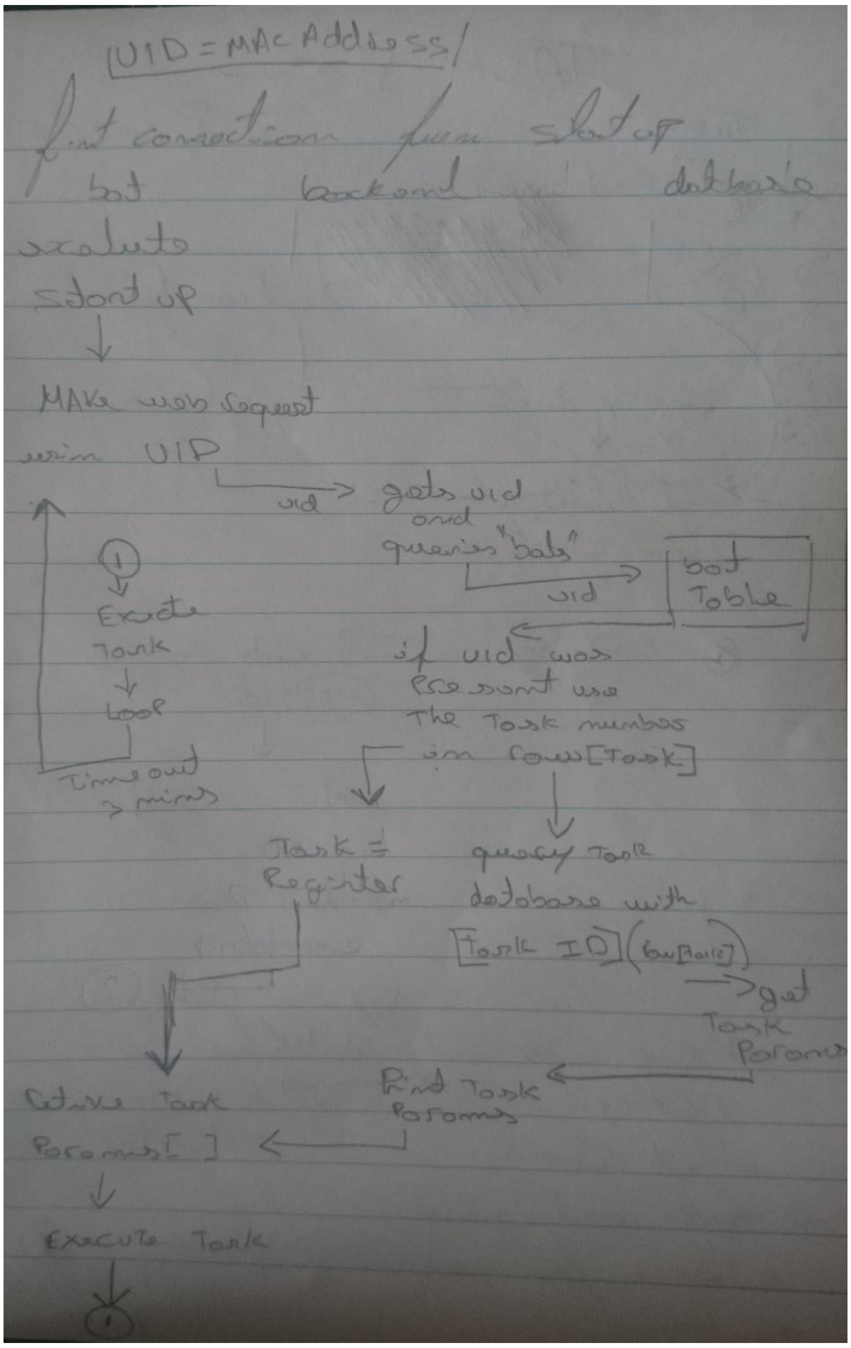
Login	asd
Real Name	real_name
Password	

```
Username:Name
User: m2k Name: m2k
```

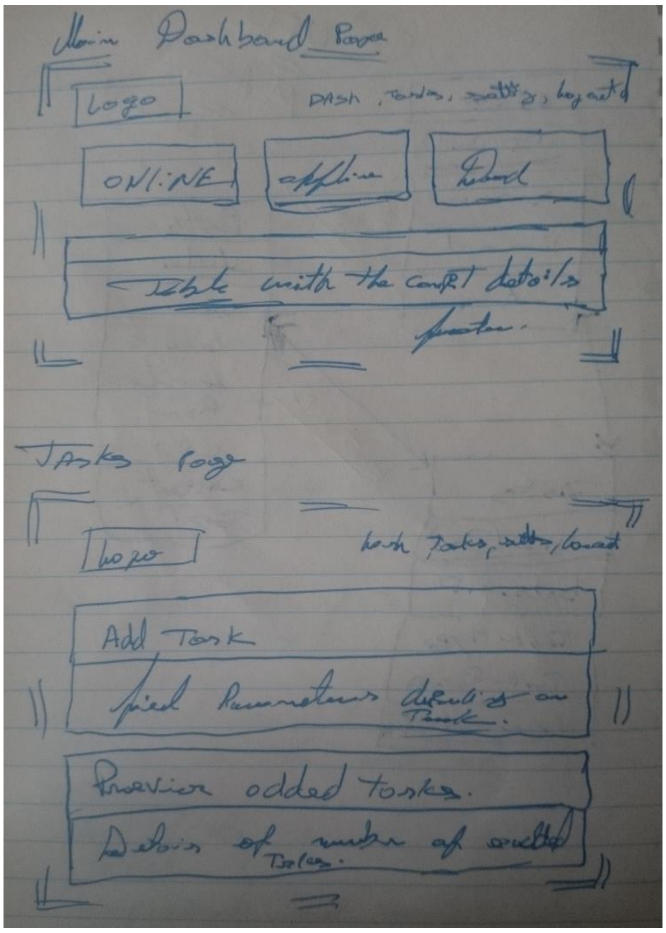
```
Future Development
~bots per page
~bot cleanup od dead status bots
```

APPENDIX H – INITIAL PRODUCT DESIGN

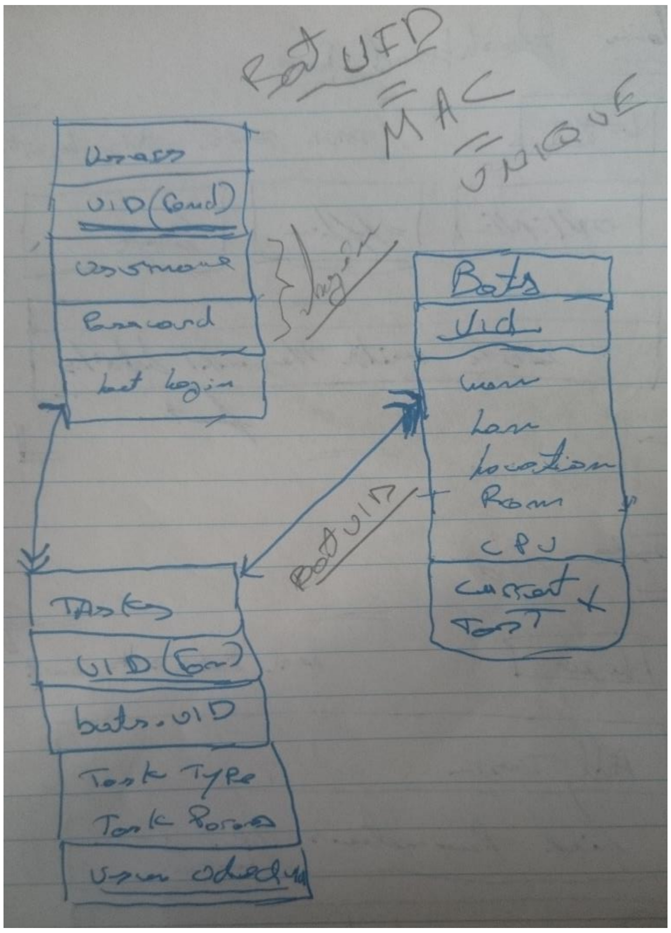
Architecture



Dashboard


Main Dashboard Page
Logo
DASH, Tasks, setting, logout
ONLINE
offline
Table with the correct details
feature.
Tasks page
logo
Add Task
Preview added tasks.

Database


Bot UID = MAC = UNIQUE
Users
UID (rand)
Username
Password
last login
login
Bots
Uid
name
Ram
location
Rom
CPU
current Task
Bot UID
Tasks
UID (rand)
bots.UID
Task Type
Task Params
User ordered