The specification for this project was a fairly open framework style application, which allowed for a quite a bit of experimentation when it came to designing the final product. There were three high level goals that needed to be met to consider the project successful, and this analysis section will be a critical evaluation of the project versus its specification.

The first objective for the project was to review literature related to chat bots, natural language processing, cyber security and visualisation methods. This research showed the natural progression of chat bots and natural language, including uncovering some services that offered easy connections between the two technologies to have reactive and understanding conversations with bots. The research into cyber security showed how quickly things can go wrong and how people need to be informed of issues, which led onto the idea of large quantities of log data in different ways to make it easier to consume for a system administrator. Furthering on the cyber security research, it was shown that a lot of security auditing and live alert tools require a user to be on-site and have direct access to the work network, or required a laptop or desktop to access the information. It was at this point that it was decided that a remote network administration tool using chat bots and natural language processing would be a worthy area of research.

The second objective was the one identified in the first – Address the inability for administrators to be able to access commands from within a network from a remote location. Specifically, we were interested in providing a framework that allowed an administrator to set up a remote dashboard away from the environment they are looking at, as a tool for accessing commands and the response to those commands.

The third objective was to address the connection between notification systems, such as our chat bot, and data generation systems, such as logs and security alerts. This means having a very open framework that allows various different methods for gathering logs and other data. This objective acted as an extension to the second one – Where the second one allowed the idea of troubleshooting being done remotely by the use of commands, this objective allowed the gathering and analysis of data and the results of which to be delivered outside the network.

We believe that we did well at achieving our objectives, as well as the stated aims in our specification. We successfully reviewed literature to do with chat bots, natural language processing, cyber security and visualisation methods, and came to a conclusion based upon that research. However, more research could have been done relating to remote administration, and perhaps the research was slightly too focussed on the idea of building a natural language processing tool rather than using a service that already existed.

We also created a framework capable of creating chat bots and easily expanding upon its functions in a modular fashion, and while there were some small deviations from the original design, the implementation worked well and we were able to use the bot to do some basic commands like pinging and tracerouting remotely. To address the third objective, we also created a log gathering function along with an alerts function that was able to inform us of an attack on a production server as it was happening, which was not even intended. This proves that the tool, even in its current example form is still capable of being a useful asset.

We were able to connect the framework to an Instant Messaging Service using Microsoft’s Bot Builder, which allows us to connect to many different types of IM service from Telegram to Skype to Slack. However, we were not able to prove that the framework is as modular as it was designed to be, as because training takes a while, we could not set up the framework to another NLP service to see if it could be easily swapped out. This is unfortunate, but we are confident that the fairly service-ambiguous design of the framework would allow for easy replacement of modules, as long as the person writing them followed the design and existing data structures correctly.

There were several smaller requirements in the problem specification that the example application and framework did not meet, such as supporting API reads and Windows Management Instrumentation tools. However, we believe that this type of application will gain traction as a useful tool in the system administration world, and that this is a very good start in that area. The example functions are not incredibly important, but the framework on which it works is, as it allows for easy expansion of an already powerful toolset.