**Progress:**

# Introduction

## Background

The introduction of chatbots into a community has brought us to the time of the conversational interface. It’s an interface that soon won’t demand a screen or a mouse. The interface will be entirely conversational, and those communications will be indistinguishable from the conversations that we have with our friends and relatives. (Onlim, 2017)

Chatbots offer the opportunity of having your own personalised assistant to enhance user experience and .

**Initial Workings**

The beginning of this project saw the creation of an initial work plan to allow scope and structure within the project. As the knowledge on chatbots was rather limited as well as python skills the ultimate goal for the exploration stage was set out for research into chatbots and teachings for the basics of python.

After reading copious amounts of literature and watching several tutorials the tasks began more clear on how the chatbot infrastructure was going to be built.

Research into which NLP API to use required some digging around until I landed on Google Clouds Natural Language API. This came to be the NLP of choice as it offers insight into features of interest including sentiment analysis, entity analysis, entity sentiment analysis, content classification, and syntax analysis.

**Problems encountered**

**Knowledge base**

The chosen database location was AWS (Amazon Web Server) using the Amazon Relational Database Service, this was due to being able to use the cloud for running the database rather than from my own machine. The most prominent cloud computing provider today is Amazon with its Amazon Web Services (AWS). Amazon not only offers the most complete stack of services, but makes it especially easy to integrate different services. [5] The aim of the relational database is for when the application requires a relational database but users want to reduce the time they spent on database management, Amazon RDS automates common administrative tasks to reduce users complexity and total cost of ownership. Amazon RDS allows users to manage their database compute and storage resources with a simple API call, and only pay for the infrastructure resources they actually consume. [6]

**Infrastructure:**

**Future work plan/progression:**