**Please be aware that by submitting your work through 'Turnitin' you are agreeing to the following statement:**

I declare that the work submitted for assessment contains no section copied in whole or in part from any other source unless it is explicitly identified by means of quotation marks and that I have acknowledged such quotations by providing detailed references in the approved format. I understand that either or both unidentified and unreferenced copying constitutes plagiarism which is considered Academic Misconduct and could result in investigation by the Academic Misconduct Panel. Further details Academic Misconduct can be found in HE11 Academic Misconduct on our website.

**Student and assessment details *(Fill every box)***

|  |  |  |  |
| --- | --- | --- | --- |
| **STUDENT ID NO. AND INSTITUTE EMAIL ADDRESS:** *e.g. S1234567@grimsby.ac.uk* | | |  |
| **PROGRAMME TITLE:** |  | | |
| **MODULE TITLE:** |  | | |
| **ASSESSMENT TYPE (e.g. report, essay, etc):** | |  | |
| **ASSESSMENT TITLE** | | <tutor to insert assessment title> | |
| **ASSESSMENT WORD COUNT** | |  | |
| **STUDENT DECLARATION OF ANY LEARNING DIFFICULTY OR DISABILITY:** | |  | |
| **EXTENSION GRANTED:** | | Yes/No | |
| **EXTENSION DATE HAND IN:** | |  | |
| **NAME OF TUTOR:** | | <tutor to insert> | |

|  |  |
| --- | --- |
| **LEARNING OUTCOMES TESTED BY ASSESSMENT** | |
| **LO1:** | <tutor to insert LO's being tested> |
| **LO2:** |  |
| **LO3:** |  |
| **LO4:** |  |

Ownership of Intellectual Property Rights for assessment is normally vested in the student. Institute staff are not permitted to pass on student work to unauthorised third parties. See policy in HE15 Intellectual Property Rights on the website.

|  |  |
| --- | --- |
| **ASSIGNMENT CHECKLIST** | |
| **I have made sure any data used is anonymised/follows ethical guidelines. Where required, ethical approval has been sought and approved (dissertations/projects).** | Yes/No |
| **I have used Arial font size 12 (or equivalent) with 1.5 line spacing.** | Yes/No |
| **I have checked that every author and document I mention in my text appears in my reference list.** | Yes/No |
| **I have put all in-text quotation in single or double inverted commas, stated their origin and included page references.** | Yes/No |
| **All information included is correctly cited in line references and in my own words unless using a direct quote.** | Yes/No |
| **My reference list is set out according to awarding body referencing guideline and any other tutor guidance I have received.** | Yes/No |

**Major Development Project**

**Assignment 2 – Product Demonstration and Documentation**

Student number:570421

Submission Date:

Word count:

Contents

[Project Introduction 4](#_Toc165379587)

[Overview of Project Functionality 4](#_Toc165379588)

[Machine learning model 4](#_Toc165379589)

[Machine learning Model pseudo code 4](#_Toc165379590)

[Machine learning Model flowchart 6](#_Toc165379591)

[Text processing 6](#_Toc165379592)

[Text processing pseudo code 7](#_Toc165379593)

[Text processing flow chart 8](#_Toc165379594)

[Training script 8](#_Toc165379595)

[Training script pseudo code 9](#_Toc165379596)

[Training script flow chart 11](#_Toc165379597)

[Chatbot script 12](#_Toc165379598)

[Chatbot script pseudocode 13](#_Toc165379599)

[Chatbot script flow chart 14](#_Toc165379600)

[Flask app (app.py) 14](#_Toc165379601)

[Flask app pseudocode 15](#_Toc165379602)

[Installs script 15](#_Toc165379603)

[Installs script pseudocode 16](#_Toc165379604)

[Installs script flowchart 17](#_Toc165379605)

[Supporting Functions and Files 17](#_Toc165379606)

[Spell check 17](#_Toc165379607)

[Intents file (intents.json) 17](#_Toc165379608)

[Training GUI 17](#_Toc165379609)

[Class diagram 18](#_Toc165379610)

[Structure diagrams 18](#_Toc165379611)

[UI Structure 18](#_Toc165379612)

[Training model Structure 19](#_Toc165379613)

[Visual elements 20](#_Toc165379614)

[Design considerations 20](#_Toc165379615)

[Minimalism 20](#_Toc165379616)

[Error handling 20](#_Toc165379617)

[Accessibility 20](#_Toc165379618)

[Colour theory 20](#_Toc165379619)

[Web Wireframes 21](#_Toc165379620)

[Low-fi 22](#_Toc165379621)

[Hi-fi 26](#_Toc165379622)

[Use Case Diagram 29](#_Toc165379623)

[Project structure diagrams 30](#_Toc165379624)

[Testing log 30](#_Toc165379625)

[Bibliography 31](#_Toc165379626)

# Project Introduction

The document presented is the design document for Hull College chatbot. The document shows the considerations taken when developing and designing the whole project.

The project will be a chatbot that will be able to answer questions about the client (Hull College). The project will have the capabilities to be added to their existing systems with ease.

# Programming language of choice

The project will use python for the AI/machine learning model

# Overview of Project Functionality

This section of the document details each function and part of the project what it will do and how it will aid the project. The section will show diagrams appreciate to the function/part of the project.

## Machine learning model

The machine model will be used to process data. The model is a neural network that uses three layers to process data, the input layer, hidden layer, and the output layer. Layers pass data through each other to further analyse the data. Two processes will use the model (The training model and the chatbot). Other process will use the model, this will be shown predetermined processes, for the corresponding scripts and functions. The model is viral the project as it will not be able to process data given without it, then the project will not be able to pick an accurate response.

Machine learning Model pseudo code

A computer screen shot of a program

Description automatically generated

### Machine learning Model flowchart

A diagram of a process

Description automatically generated

## Text processing

The project uses a text processing script to determine a vocabulary for the chatbot.

The text processing allows the program to look at words that are stored in a data file. These words are stored from past questions, then each word is assigned a number 0 – 1, depending on if the word could be found. The text processing also uses stemming so a word it processes will be striped down to its route (eg. “programming,” “programmer,” and “programs” can all be stemmed down to program). Stemming is used withing the program in increase proficiency. The project will the use the text processing to aid its answer to the user, as it will be able to view if they words are in the predefined patterns later.

The text processing is used throughout the system, if this section of the system needs to be down again it will be shown within a predefined system in other diagrams.

### Text processing pseudo code

A computer screen shot of white text

Description automatically generated

### Text processing flow chart

A diagram of a work flow

Description automatically generated

## Training script

The training script is used to train and aid the chatbot in answering user questions. The training script uses epochs to test how well the AI can deal with user questions. The training script then get data from the epochs, optimises it and then sends it into the data file so the chatbot part of the program can use it.

An epoch is one pass through of training data, that then pass through data so the program can use data to become more accurate. The training program will use 1000 epochs each time the program runs, any more than this can cause the chatbot to be “overly intelligent”. If the chatbot were to have any more epochs, then it could lead in the chatbot trying to answer questions it won’t be designed to answer. Epochs determine how accurate the AI is based on a loss, the script will try to aim for close to no loss, however loss is not bad in this scenario. As the interaction with the user cannot always be predictable or prepared for as users could ask questions that the developer and client did not think to give the project the ability to answer. Users could also ask questions that are not related to the current system or the client. Allowing for losses in the training script then allows the chatbot to account for questions it may not know how to answer, then allow it to use a fallback response instead of tyring to answer it (more to be discussed in the chatbot script section).

### Training script pseudo code

A screen shot of a computer

Description automatically generated

### Training script flow chartA diagram of a flowchart Description automatically generated

## Chatbot script

The chatbot script is the main part of the project, it is used to chouse an response based on a user’s question. The script uses data gathered from using the training script to aid its responses and its ability to understand what the user is telling it. The chatbot uses the text processing in conjunction with a spell check function. The script stores a log of questions that it could not answer, to aid its ability to the future, if the client wanted to update its responses based on what users are asking.

The script uses an idiom with an infinite while loop inside. The idiom is used so a user can ask the chatbot as many questions as they would like, the function needs to be like this so multiple user will be able to ask questions at the same time through the web interface.

### A screen shot of a computer screen Description automatically generatedChatbot script pseudocode

### A diagram of a flowchart Description automatically generatedChatbot script flow chart

## Flask app (app.py)

The flask app is what the project will use to host the chatbot to the used through the web interface of the project. The flask app will act as a controller, as users will send HTTP requests to the script then it will manage the request and send back the appropriate response/ information to the web interface. The app uses chatbot script to determine responses. The flask app also has necessary validation to stop users sending data that shouldn’t be sent to the chatbot, this incudes messages that could be too long and blank messages. The app also has 2 methods, one to validate that the server is running and one to ask the Chabot a question as well as receive a response. Do to the nature of a script that primarily uses flask it would not be beneficial to use a flowchart.

### Flask app pseudocode

A screenshot of a computer program

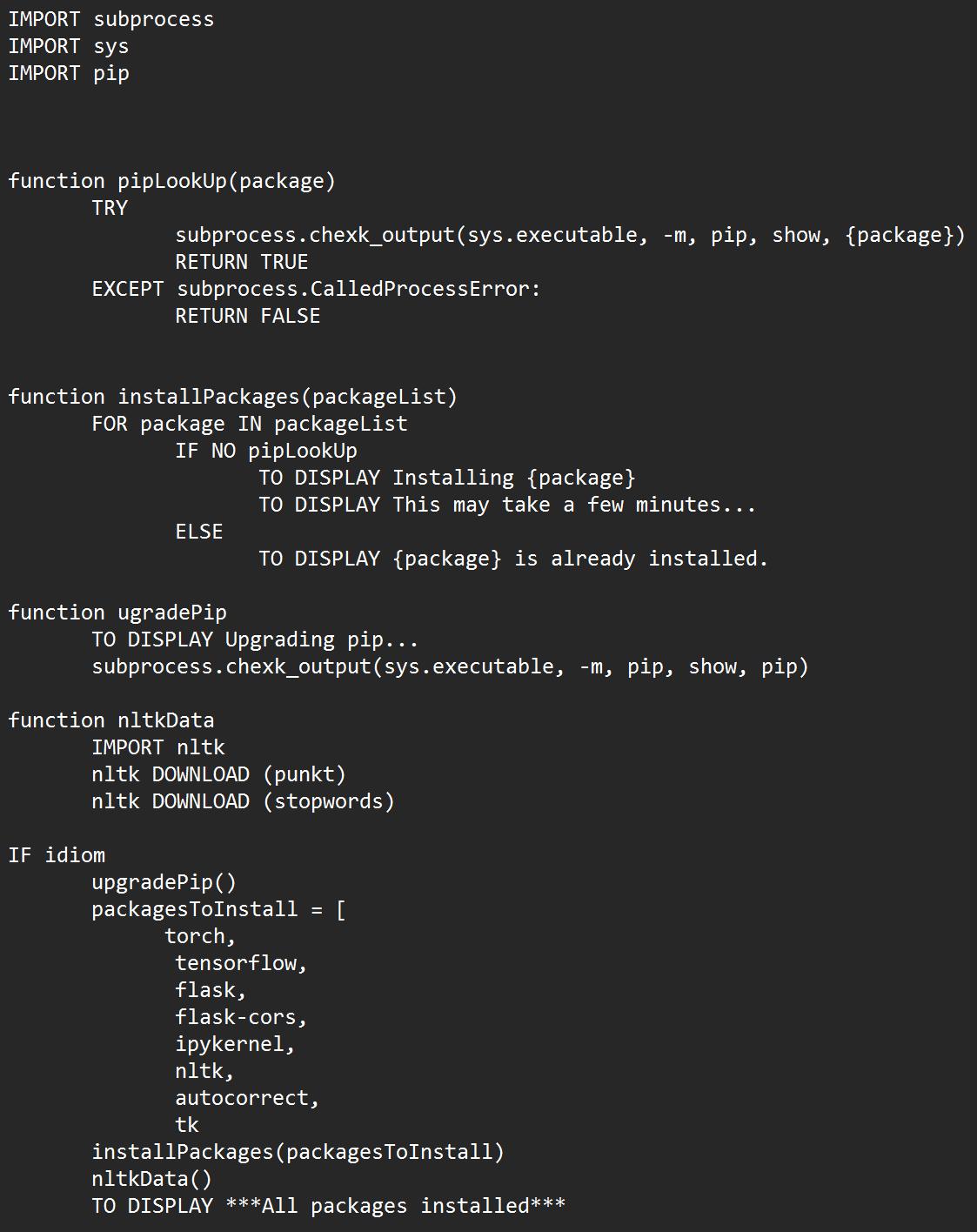
Description automatically generated

## Installs script

The installs script will be used in conjunction with the chatbot script and the chatbot script and the training script. The purpose of the script is ensure that the users computer has the necessary packages installed. The script means that the user will not have install the script externally and will all be done through the project, therefore making the user experience of the client much better.

If this script was not developed then it could have left users with unplanned errors, effecting the projects ability to be edited in the future.

### Installs script pseudocode



### A diagram of a package Description automatically generatedInstalls script flowchart

## Supporting Functions and Files

Below are other functions and files that aid the project, the aspects discussed will not include any diagrams, either because other diagrams show theses function are shown in the scripts above or the files are not in a correct format to create diagrams.

### Spell check

The project uses external packages to spell check the user’s questions before their questions or processed through the chatbot. This help uses that may have learning difficulties. The function is only located in the chatbot script as it will only be useful if users type questions as the training script only uses words in the intents file.

### Intents file (intents.json)

The intents file stores patters that users may use to ask questions, the responses that the chatbot should respond back with. All this information is stored into a tag called something appropriate to the question.

The file is structured in such a way that that the client can use the file to add new questions to the file, then use the training script to allow the project to answer new questions. The easy-to-use structure can allow the project to adaptable in the future.

### Training GUI

The training GUI is a graphical user interface. The training GUI is used to allow the client to train the chatbot to be able to answer new questions that the user may have, making a means it will be much easier for the client to update the chatbot.

# Class diagram

# Structure diagrams

## UI Structure

A diagram of a computer

Description automatically generatedThe Project has two different UI that users can use. They both use the same Java Script files to retrieve the data from the app.py file (server). The smaller UI uses jQuery to access the UI files needed to display the chat. The project uses jQuery so the chatbot can be accessed on any page of the clients existing system. In the diagram below this is shown through index.html however in practise they can be any webpage that the client may have.

## Training model Structure

The project needs a way train the chatbot to recognize new responses as well as ensure that the chatbot will recognise responses over time. To do this the client will need to run the training program, then they will be able to start the chatbot app back up and all training data will apply. Files such as **intents.json , model.py, textprocessing.py** and **data.pth**. Files here are used else whereas the project follows the object ornated programming as functions, classes and data in them files are used elsewhere.

A diagram of a training model

Description automatically generatedThe training model will be only be ran server side and base users such as customer will not be able to see or run the code. The training model does not need to run in conduction with the chatbot and only needs to be ran when updating the chatbot, then the chatbot can be ran on its own

# Visual elements

## Design considerations

The project will follow many users’ interaction (UI) and user experience (UX) considerations to make the project as *user friendly* to the user possible.

### Minimalism

The project will have a minimalist look, to help user be able to navigate and use the project. To achieve such look the project UI will be well spaced out, so users do not accidently click on other elements of the project. Another aspect that will contribute towards the projects manualism is white space. The project will use white space to help other aspects of the project stand out to the users while less important aspects are less focused on. The project will also make use of icons and animations, to demonstrate what the project is doing or what the project is doing.

### Error handling

All parts of the project will have error handling and appropriate error messages, this means running any scripts and the UI the customers will see. Errors will show useful information to users (eg. If the server for the chatbot is unavailable, telling users that its unavailable). All errors the developer is aware have been accounted (on both sides of the project) and the code has been adjusted to make sure the project will not beak and show unplanned errors or messages. Focusing on error handling ensures all users have a pleasant user experience.

### Accessibility

The project will cater towards the select users who may have additional needs. The project will have a function to correct spelling when if a user is to send a question with uncorrected spelling, the program will correct the spelling of the question before processing the users question, this then means the project should be able to process questions even if they are spelt wrong. Other accessibility features are having a font that is rounded and large to make they easier to read and understand.

## Colour theory

The colours that were chosen are mixtures of green and blue. The reason these colours were chosen specifically is that green typically represents new growth and new beginnings. Green can also represent a calming atmosphere, in contrast to this blue also represents calming attributes such as the feeling of responsibility, friendliness and peace. The web app portion of the project will mostly use light blues and greens as they provide the most relaxed and calming environments. Considering the project will be used in an education environment, the developer thought these colours would invite students to use the system.

The system also uses a white that has a tint of blue to use for a background, as they allow for text to stand out. Text colours will widely depend on what colours the text is above, if the colour is a dark colour, then text will be white, then if text is a light colour, then it will be black. Have text colours like this will then help users who may struggle to read from different contrasts. Some elements may also use a dark shadow to make some elements pop up, but will not affect elements like text and buttons, this way users will still be able to take existing elements of the project.

A blue background with black text

Description automatically generatedAll colours and Hex codes are shown in the image below.

## Web Wireframes

Below are web frames that shows how the project will look to user once implantation. All web frames have a low-fi and a high-fi virent. The low-fi diagram shows a simple design that only shows the base design with elements the user may need to use. The hi-fi shows all colours and possible fonts that the colour. Both sets of wireframes show a demo page, the demo page will be used to demo parts of the project and will not be a part of the final system.

### Low-fi

A screenshot of a wireframe

Description automatically generated





A diagram of a picture

Description automatically generated

### Hi-fi



A diagram of a chat

Description automatically generatedA screenshot of a computer

Description automatically generated

# Use Case Diagram

The project has two users that will interact with the system; however, the use case diagram shows three. The third user is the machine learning AI. The reason the AI has been added is it acts like a user as it answers the user’s questions. The other two users are users within the client’s origination and the customer or students. For the purpose of simplicity, The diagram has narrowed users down to user/customer and client.

A diagram of a chatbot system

Description automatically generated

# Project structure diagrams

The project will complete many tasks by using multiple files in different orders. The diagrams below show how different users of the system will interact with it. The system has been split into sections of the program for simplicity.

# Testing log

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Num | Test name | Data | Expected outcome | outcome | Success | Actions |
|  |  |  |  |  |  |  |

# Bibliography

Chapman, C. (2021). *Color Theory for Designers, Part 1: The Meaning of Color* [Online]. Available at:[*https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/#:~:text=Green%20(Secondary%20Color)%20%23&text=It%20can%20represent%20new%20beginnings,of%20the%20energy%20of%20yellow*](https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/#:~:text=Green%20(Secondary%20Color)%20%23&text=It%20can%20represent%20new%20beginnings,of%20the%20energy%20of%20yellow)[Accessed: 19/04/2024]