Appendix E – User Guide

Below is a screenshot of each page of the website in turn, and a description of the functionality of each page. Due to the nature of the application being fairly simple and only involving two main functionalities these screenshots and descriptions should give users enough information to understand how the application functions.

1. Home Screen (chat_main.html)

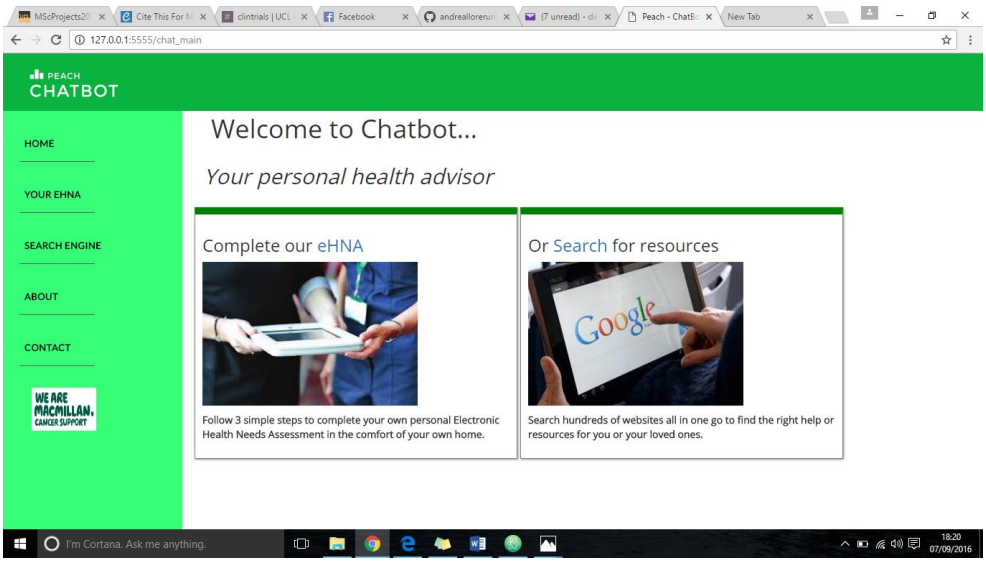

Figure 1 Home Screen

This will the first page accessed when opening the application. On the home page the user will be presented with a list of options along the side-navbar for the different areas of the website, or in a drop down menu for smaller devices. From the main part of the screen there is a smaller link to each of the main elements of the application with a small description of what their function is.

2. eHNA log in page (chatBot.html)

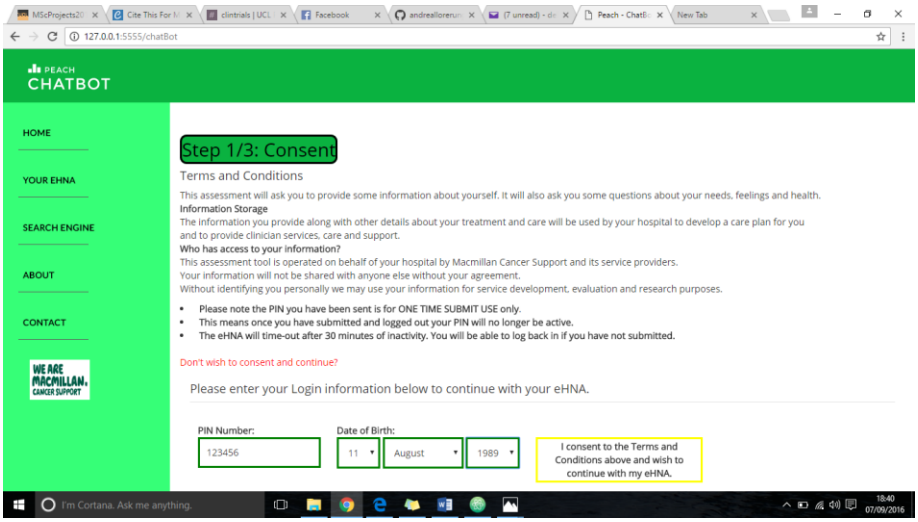

Figure 2 Login Page

This is the first page Users reach when accessing the eHNA. The pre-received Pin number and users' DOB must be filled in correctly, or the user will be presented with an error message. There is also red text link to draw the attention of users who do not want to consent to the terms and conditions.

Please note the database at the moment only allows users to log in using pin 123456 and DOB 11th August 1989.

3. Select Concerns Screen (ehna.html)

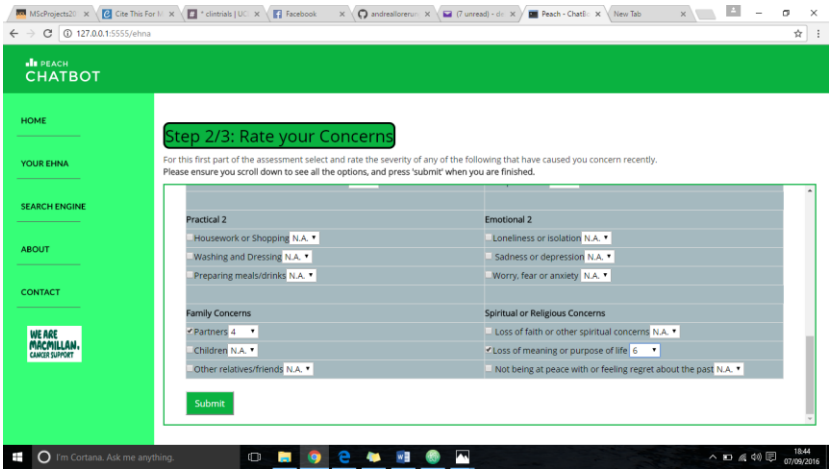

Figure 3 Initial Concerns Screen

On this page user select the concerns and give them a distress rating out of 10. Once submit is pressed the user is asked to confirm they are happy with their choice, and they have the option to return and ask to or edit their choices. If they are happy with their selection, they continue to the actual conversational UI page.

4. Chat Screen (msgchat.html)

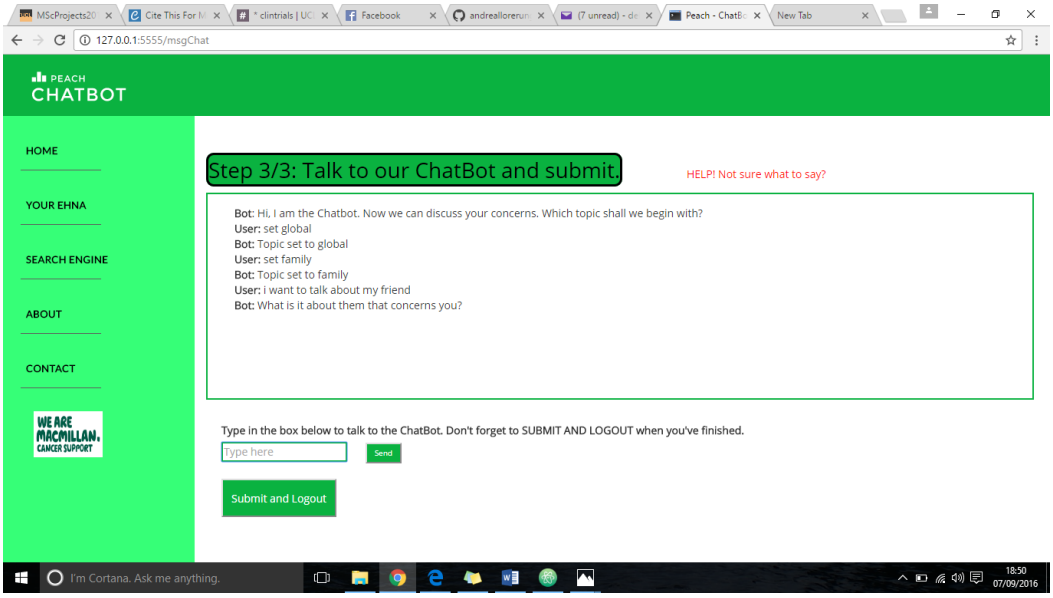

Figure 4 Chat Screen

This is the actual chat screen, where a user is invited to talk through their concerns with the bot. There is a help option to give suggestions on how to start the conversation. When

users are finished they are prompted to submit their assessment and logout, and are redirected to the home screen.

5. Search Engine

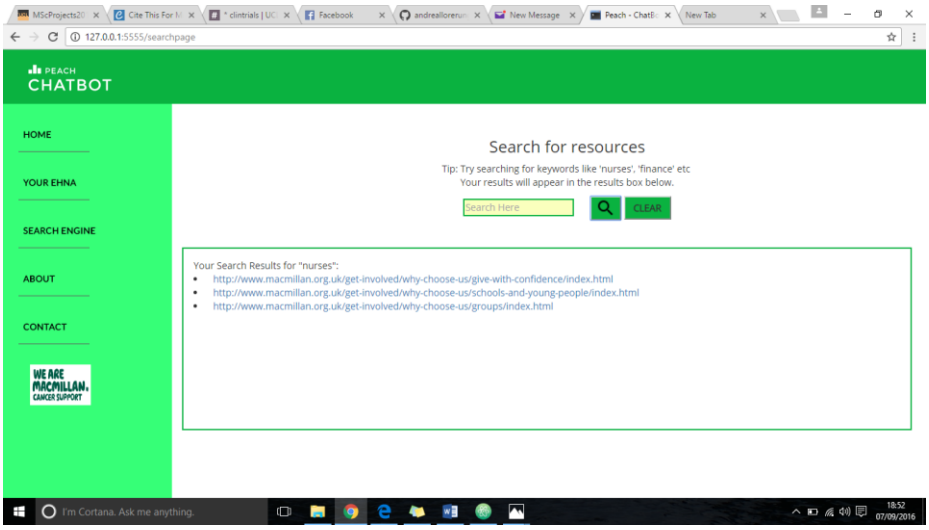

Figure 5 Search Screen

This is the search screen. It is a simple format, users fill in a search field and press the magnifying glass icon. Users are either presented with a list of result URLs they can select, or are told there were no results available for their desired search field. Users can also press the 'clear' button to clear the results box and search again.

6. About Screen

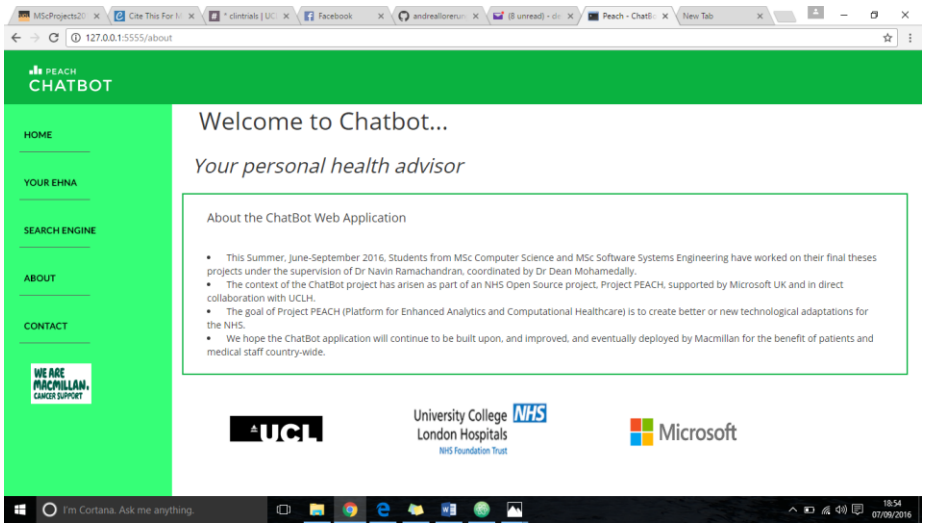

Figure 6 About Screen

Users can select the 'About' Screen from the side bar or the dropdown nav menu (on smaller device screen). Here there is a simple description of how the application came about and links to the major organisations involved in the project.

7. Contact Us Screen

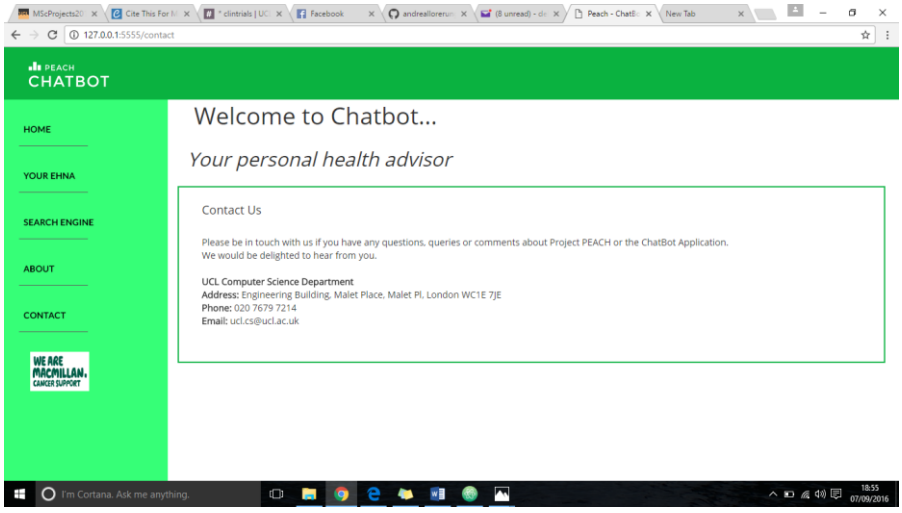

Figure 7 Contact Screen

Similar to the 'About' Screen, the 'Contact' Screen simply lists some information for those interested in contacting someone about the Chatbot application. It can be accessed from the side bar or the top navbar (on smaller devices).

Appendix F – System Manual

Installing Python and Pip

You will need Python version 2.7.12 in order to edit/add to the code of the Chatbot Application. If you have python installed, you can check your version using the command 'python -V' or 'python --version'.

Python can be installed from https://www.python.org/downloads/ for different operating systems.

There are ways in which to port python version 2 code to version 3 if that is the version you are currently running. Please refer to the following documentation:

https://docs.python.org/2/howto/pyporting.html.

You will need to ensure pip is installed in order to be able to install any packages for Python. pip is a package management system used to install and manage software packages written in Python.

If you are not sure if it is already installed type the following command in the command line:

pip list

If pip is not recognized as a command you will need to install Pip with get-pip.py and then run with command line:

python get-pip.py

You may need to add to Python and Python/scripts in the System Variables PATH (within Environment Variables for Windows) for Pip to be recognised.

Setting up your Python Application

Once Pip is installed you will probably want to install virtualenv. This is a tool to create isolated python environments. You can install virtualenv with the following command line:

pip install virtualenv

Your environment is now ready for a python application.

Setting up Flask application

To set up your Flask application simply use the command line:

pip install Flask

See the Flask documentation for full details on how to set up a Flask project:

Flask - http://flask.pocoo.org/

Setting up the Chatbot Application

A full list of dependencies for the Chatbot project are listed in requirements.txt file.

These dependencies will either need to be imported and installed via pip or are Python internal libraries and just need to be imported.

You can view Python's standard library at https://docs.python.org/3/library/. These functions can be simply imported without installing with pip.

https://pypi.python.org/pypi contains the Python Package Index, these packages need to be installed using Pip.

Git

The Git repository where all the Chatbot code configured for an Azure webpage can be found on:

https://github.com/andreallorerung/peach-chatbot-alpha

The repository can be cloned and downloaded to open on your preferred IDE.

Atom

The Atom editor can be downloaded from: https://atom.io/

Additional packages for Atom can be seen at https://atom.io/packages.

Launching Web Server Locally

Using Atom, the Web Server can be begun with the command 'Shift-Control-B'.

Enter browser and go to Localhost:5555 (the application is currently configured to 5555 but this can be altered within runserver.py.

Launching the Azure Site

The Azure website can be accessed at: http://peach-chatbot-alpha.azurewebsites.net/

SQL Database

The application contains the code to connect to the SQL database contained on Azure. However, the code is commented out as it will not work with the UCL Eduroam Wifi network. It is recommended to leave this code commented out unless developing outside of the UCL zone.

In order to connect to the database, you will need to you have downloaded the correct ODBC Driver and have it stored within System DSN within Administrative Tools (for Windows).

The required driver is Devart ODBC Driver for SQL Azure. 32 bit and 64 bit versions are available and you will need to ensure you download the correct one for your operating system.

This is available on https://www.devart.com/odbc/sqlazure/.

In order to actually alter the database, which has not needed to be fully formed for this project, you will need to install Microsoft SQL Server Management Studio.

Please note in order to use in the current test log in details the Pin number is required to be 123456 and the DOB is 11th August 1989.