

# Functional Specification of DCUBuddy

---

**Student 1:** Mark Kevin Queypo

**Student Number:** 19404214

**Student 2:** Conor Patrick Marsh

**Student Number:** 19728351

**Supervisor:** Dr. Brian Davis

**Completion Date:** 09/12/2021

---

## Table of contents

<b>1. Introduction</b>	<b>2</b>
1.1 Overview	2
1.2 Business Context	2
1.3 Glossary	3
<b>2. General Description</b>	<b>3</b>
2.1 Product / System Functions	3
2.2 User Characteristics and Objectives	4
2.3 Operational Scenarios	5
2.4 Constraints	9
<b>3. Functional Requirements</b>	<b>9</b>
<b>4. System Architecture</b>	<b>12</b>
<b>5. High-Level Design</b>	<b>13</b>
<b>6. Preliminary Schedule</b>	<b>14</b>
<b>7. Appendices</b>	<b>17</b>

# **1. Introduction**

## **1.1 Overview**

Our project is a web application which is a virtual assistant or chatterbot called DCUBuddy which is targeted towards newer students and for current students attending Dublin City University. We believe our application will help students during their time in university by offering helpful features that will help newer students get accustomed to the college and get any additional help. The virtual assistant shall help them from any queries they have. Features that will help include: answering queries like getting information for courses, finding the current timetable for their course for the day and a map of the college. For existing students, we hope this application can help them from day to day work. Features that will help include the ability to add assignments with a due date or a command to find past exam papers of a module.

The current system in DCU involves scrolling through the Loop website to find such features as assignment details and course information. There is a need to visit several pages through different links. There is a search function on the homepage to handle search terms. However there is no direct means to provide a student with a natural-language response to a single recognisable query. Additionally timetable and past exam paper information is primarily locatable through separate websites with various input requirements from the student. The possibility exists to represent this information through a single assistant which can retrieve and present the required information at a single location.

## **1.2 Business Context**

This system could be deployed, and will be designed with the intention to be deployed, at Dublin City University. It is intended that the application could be provided to first-year students as they would be targeted by the designers as having the greatest need for help in campus orientation. However, many of the intended features of this application will be of benefit to any student of the university irrespective of year or course of study and it is at the discretion of DCU as to precisely how many students the application could be provided to. It is the intention that the application could be incorporated into the design of the university's current Loop website which is currently the primary location for all campus and course-related information. Alternatively, the intention is for the application to have a standalone graphical user interface which could be incorporated into a website of its own.

## 1.3 Glossary

**Chatterbot:** An artificial intelligence software which can accept a typed or voice-based question/enquiry and engage in a simulated conversation on the topic in natural human language.

**HTML:** Hypertext Markup Language, which is the language used for creating documents for the World Wide Web. The structure of the web document is defined with tags and attributes.

**CSS:** Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML

**API:** Application Programming Interface, which is a connection between computers or between computer programs that allows the two applications to talk to each other.

**Natural Language Processing:** the branch of artificial intelligence concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

**Machine Learning:** a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

## 2. General Description

### 2.1 Product / System Functions

The virtual assistant will answer typical queries from a new student would have. Queries include typical questions a student may have such as information about a module, a guide around the campus, frequently asked questions on registration etc. Answers for these queries can be retrieved from DCU's official website. Student's will be able to get their timetable for the current day. The app should take the information from [opentimetables.dcu.ie](http://opentimetables.dcu.ie) in order to get the information on modules the student has for the day. Another function the application has is the ability to add tasks (such as assignments) with details and the ability to delete those tasks as well. The virtual assistant includes speech recognition using state of the art APIs that should detect a student's voice and enter the command. We have identified a set of open source tools and services that will be used for the application. The intended Python libraries to be used for speech recognition have been identified and it is the objective to provide a very high standard of accurate recognition without requiring advanced microphone technology to perform this. Speech synthesis libraries such as *pyttsx3 2.90* will provide a means to translate text to speech in order to provide optional text or voice-based feedback mirroring that of the end user. GUI interface will be created to function as part of the current DCU Loop website system or as an element on a separate website. GUI requirements aim to be fulfilled using

software such as Flask and python libraries such as *Tkinter 8.5* to create a respectable and recognisable look which fits the overall brand identity of DCU as an institution. We aim to structure and model the data received by the DCUBuddy in such a way as to recycle it to produce new knowledge. The introduction of data from external sources will boost the operation of the application enormously. The chatbot will be able to capture, process and reproduce large amounts of data in the shortest time frame possible based on available resources.

## **2.2 User Characteristics and Objectives**

The User Interface(UI) for the virtual assistant should replicate the UI for messaging apps like Whatsapp, Messenger, Discord etc. Users should be more familiar with the standard UI for chatting apps for easy use of the application. UI should be simple to understand and user-friendly for users unfamiliar with the UI. The speech recognition will enable faster use instead of just typing. The user must have a microphone for their computer to be able to access this feature. How the application recognises the user's voice and the accuracy of the words detected is based on the quality of the microphone the user has.

The AI of the DCUBuddy will lean heavily on using the strongest Python libraries possible to fully optimize an exceptional user experience. There must be a deep understanding of the context of the conversation in terms of NLP (Natural Language Processing). Emphasis will be on identifying the context of a question to provide an accurate answer, make suggestions to resolve the issue, and actively search for information where necessary.

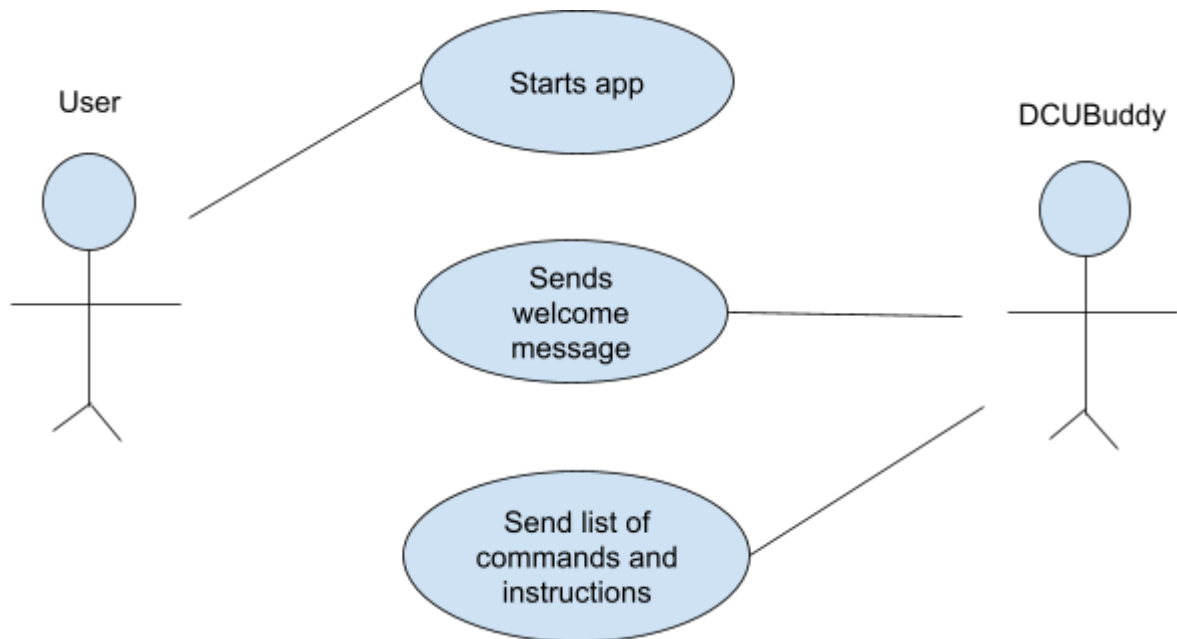
The aim of the designers is to create DCUBuddy in a manner where it is continuously learning and gaining more access to different data over time. In this way, its function will improve the more frequently it is used. Building up a repository or corpus of information will be a feature used to train the chatbot to use information in increasingly accurate and useful manner for the user.

A well-designed user interface will allow for adjustable visual elements in order to make it possible to incorporate the app with different educational facilities. The planned design will centre on the Dublin City University brand and provide a clear overview of conversations in a manner befitting an esteemed educational facility such as this. It is the intention of the designers to align the use of data fully with GDPR requirements in order to maximise the use of different data sources in compliance with legal rules.

It is the intention of the DCUBuddy designers to make use of technically complex technologies while nonetheless presenting an end product which can be used quickly, easily and immediately by the end user. It is expected that the amount of data pre-programmed into the DCUBuddy will be more than adequate to provide smooth and straightforward functionality from the start with improved performance over time.

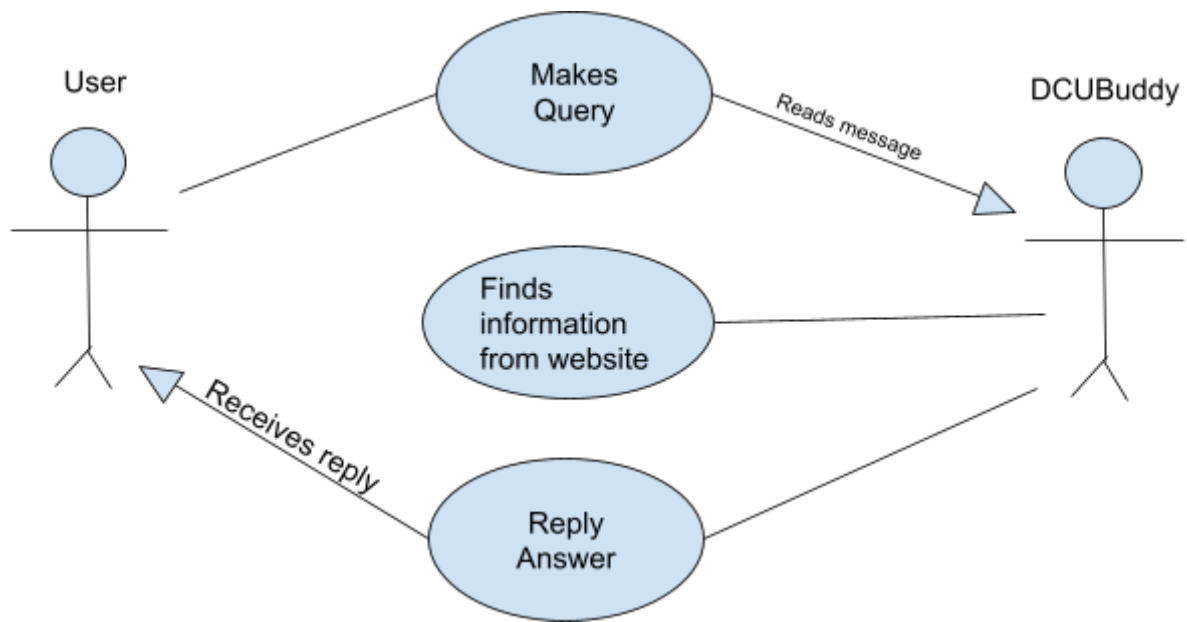
## 2.3 Operational Scenarios

Use Case 1: User starts application:



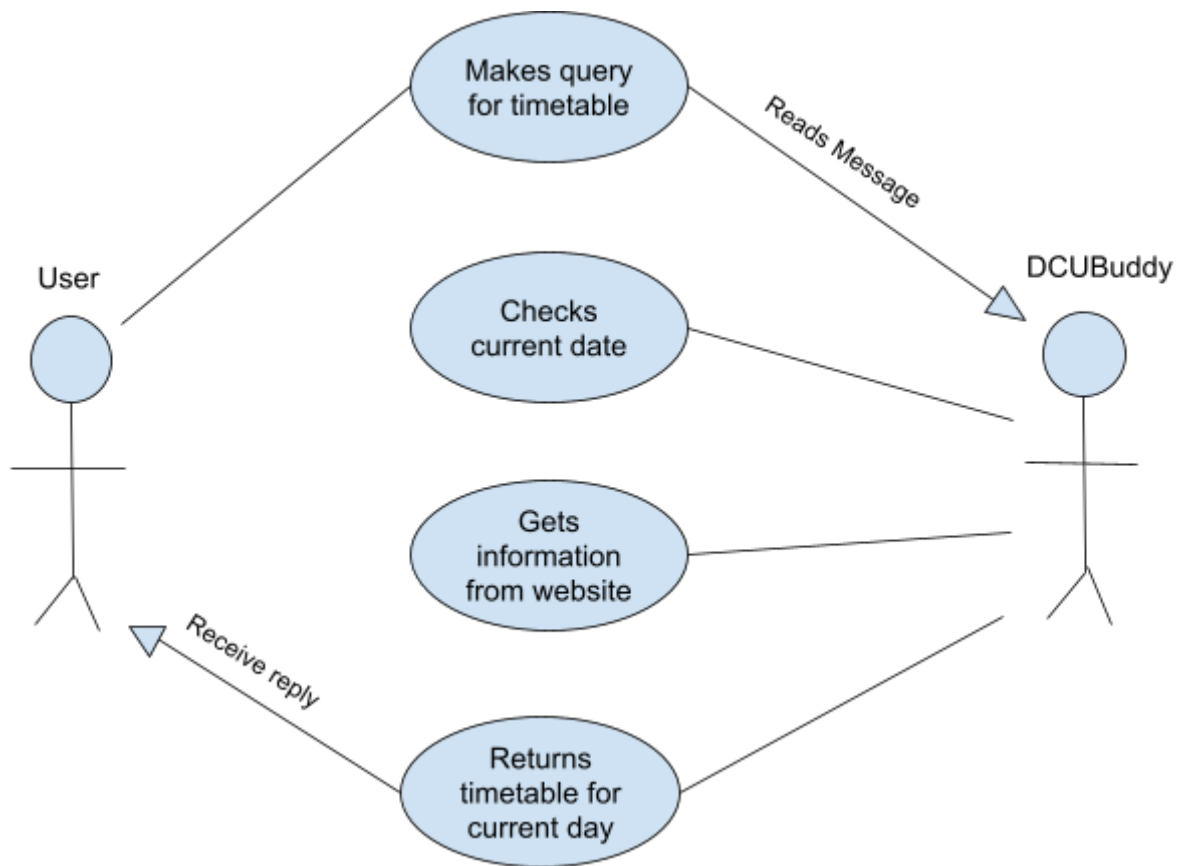
- When the user opens the web application, the bot will welcome the user and give the user information on how to use the application.

Use Case 2: User inputs a query



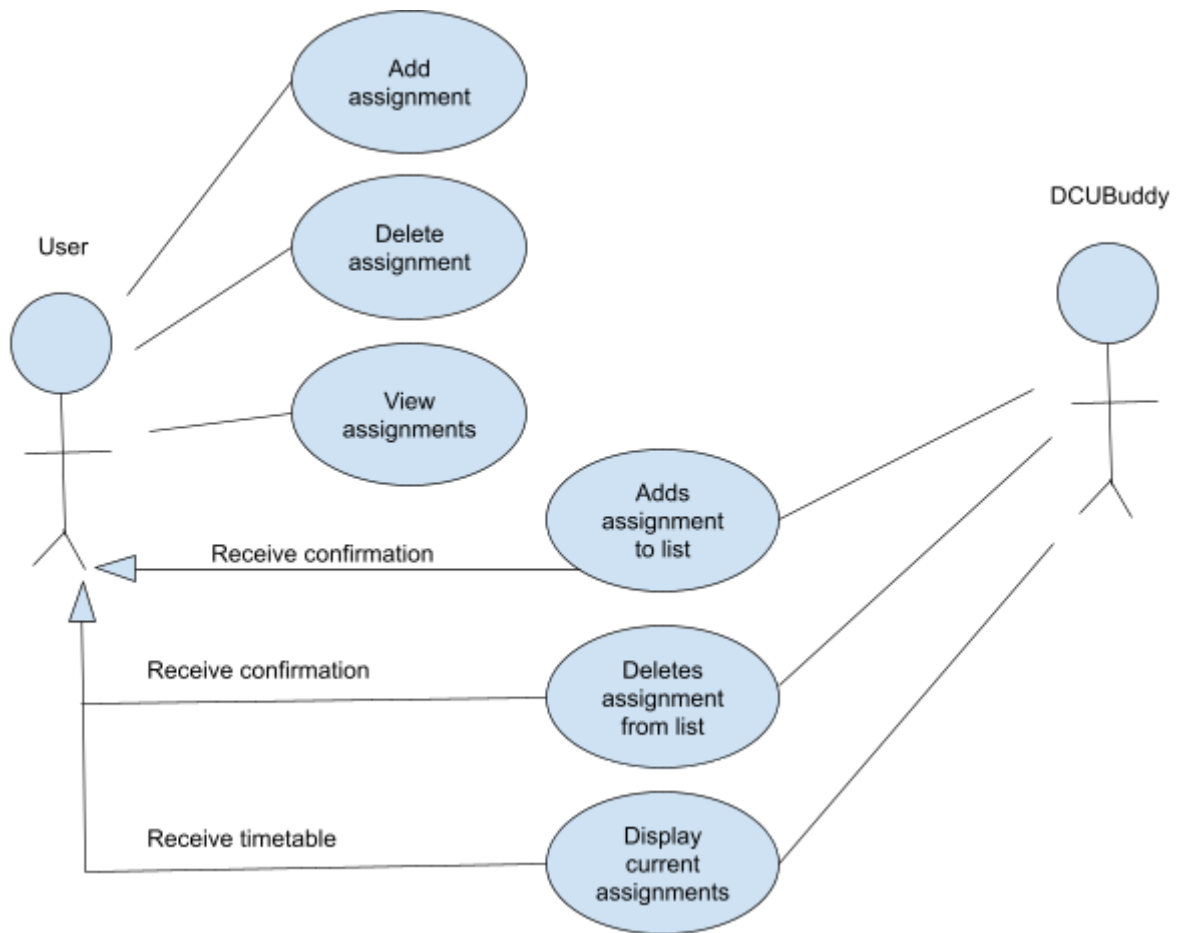
- When user makes a query, the chatter will make reply with an answer to the query and may retrieve answer from DCU's website.

Use Case 3: User wants access today's timetable



- User asks for current date's class timetable
- Chatterbot returns timetable from [opentimetables.dcu.ie](http://opentimetables.dcu.ie)

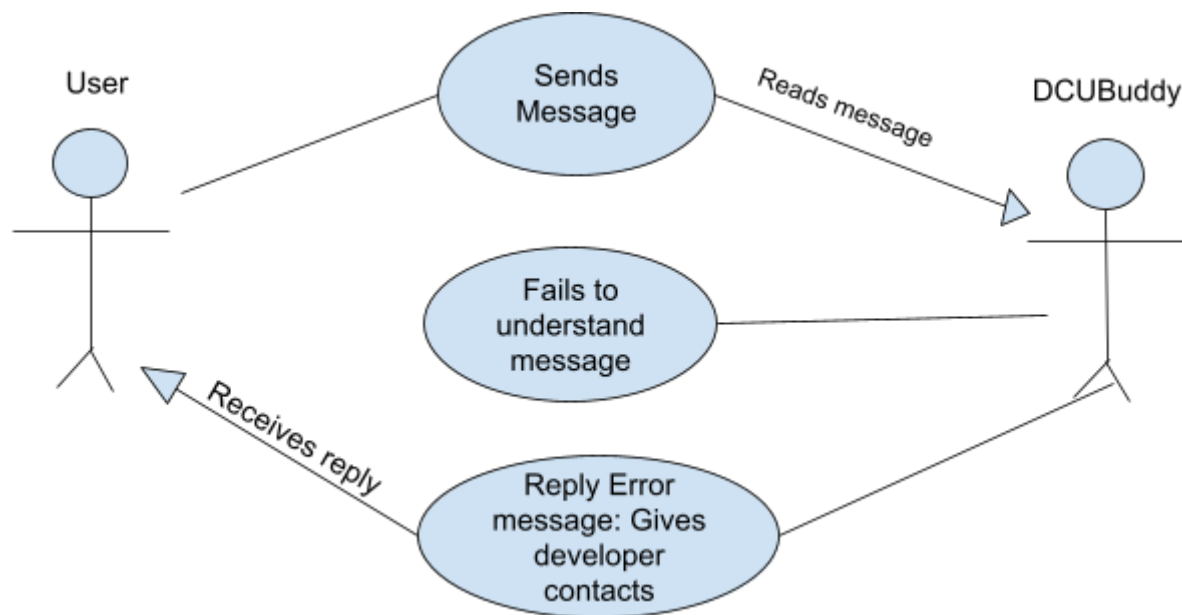
#### Use Case 4: User views, adds or deletes assignment



- User sets/deletes/view assignment(s) or reminder with a due date
- The chatterbot adds the assignment to current list of assignment, deletes assignment from .st or displays the assignment



## Use Case 5: Bot does not understand user input



- When chatbot can't understand a user's message, reply with a message including email of developers to assist with the problem.

## 2.4 Constraints

- Lots of testing and data as different users have different inputs.\
- Technical knowledge on Chatterbots
- Time constraint - we are given a deadline for project deliverables by March 2022. We aim to follow our schedule and get the project done before the deadline.
- Limits of student data or business information that the university can or is willing to provide for GDPR reasons or otherwise

## 3. Functional Requirements

<b>Description</b>	The system must be able to accept a user's input.
<b>Criticality</b>	Essential for the whole system as user input is needed to access all features.
<b>Technical issues</b>	System should reasonably handle voice-based questions with varying qualities of microphone on the end-user's side
<b>Dependencies</b>	Requires implementation of the system in a specific area of the university website or standalone website

<b>Description</b>	The system must be able to connect to the internet
<b>Criticality</b>	Essential for answering user queries and timetable fetching
<b>Technical issues</b>	
<b>Dependencies</b>	

<b>Description</b>	The system must be able to connect to DCU's website.
<b>Criticality</b>	Essential to answer a user's query
<b>Technical issues</b>	Error could arise depending on the state of DCU's website going down
<b>Dependencies</b>	Must connect to internet

<b>Description</b>	The system must be able to connect to opentimetables.dcu.ie
<b>Criticality</b>	Essential for accessing the user's timetable.
<b>Technical issues</b>	Error could arise depending on the state of the website going down
<b>Dependencies</b>	Must connect to internet

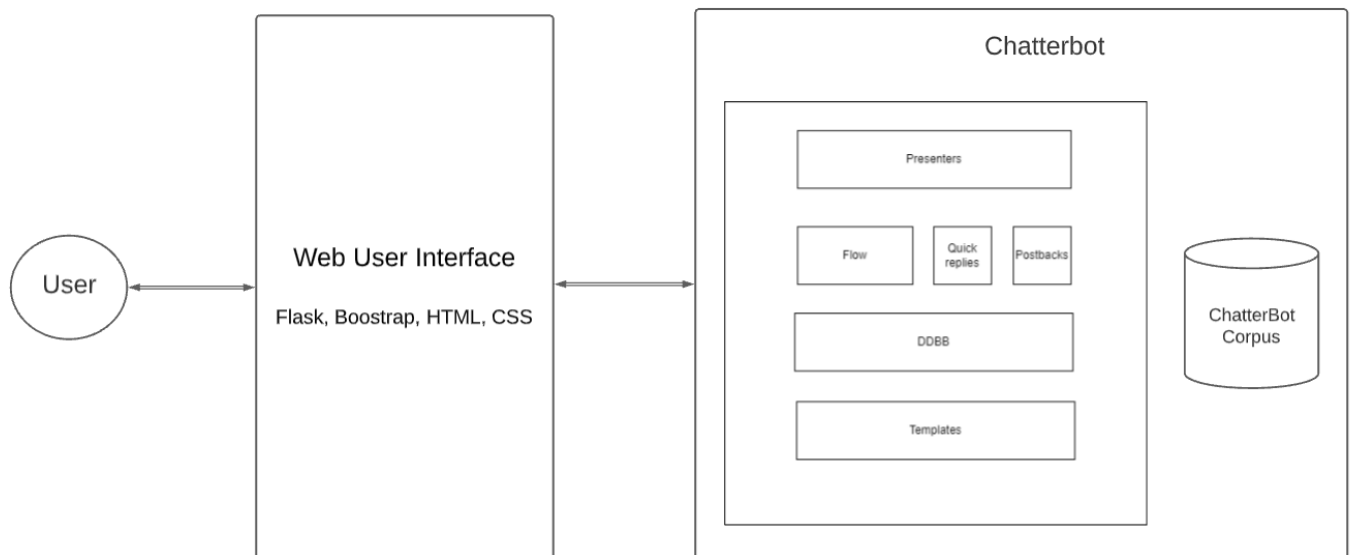
<b>Description</b>	The system must send a reply for every input.
<b>Criticality</b>	User must always receive a reply from the system whether it successfully reads or fails to read user input.
<b>Technical issues</b>	System should have means of clarifying or recycling unclear questions in a way to allow the chat to continue
<b>Dependencies</b>	

<b>Description</b>	The system must be able to add, store or delete assignments.
<b>Criticality</b>	Requirement for assignment feature.
<b>Technical issues</b>	
<b>Dependencies</b>	Assignments should be viewed.

<b>Description</b>	The system must be able to display current stored assignments.
<b>Criticality</b>	Requirement for assignment feature and viewing student's assignments.
<b>Technical issues</b>	
<b>Dependencies</b>	The system must be able to add, store and delete assignments to display a user's assignment.

## 4. System Architecture

### Elements of Chatbot Architecture



#### **Presenters**

This level is responsible for making a call on the given channel on which the chatbot operates to show text, an image, button, video or other feature.

#### **Flow**

This layer executes the logical operations of the chatbot by using repositories to store data, extracting information from web services, and generally putting available data to use.

#### **Quick Replies**

Makes use of the functions declared by the Flow to control quick replies that arrive from the channel by different bot actions.

#### **Postbacks**

Triggers action from the Flow module by getting the user's input.

#### **Repositories**

Contains the most common elements to obtain data and templates from our database or web services we are using.

### Templates

Predefined actions that our chatterbot is going to respond with.

### Speech recognition

Libraries responsible for handling the information provided by the user through voice-based commands/questions.

### Speech synthesis

Libraries responsible for enabling an audible text-to-speech response from the chatterbot.

### Conversation manager

The central element that decides the flow and style of the conversation.

### Natural language understanding

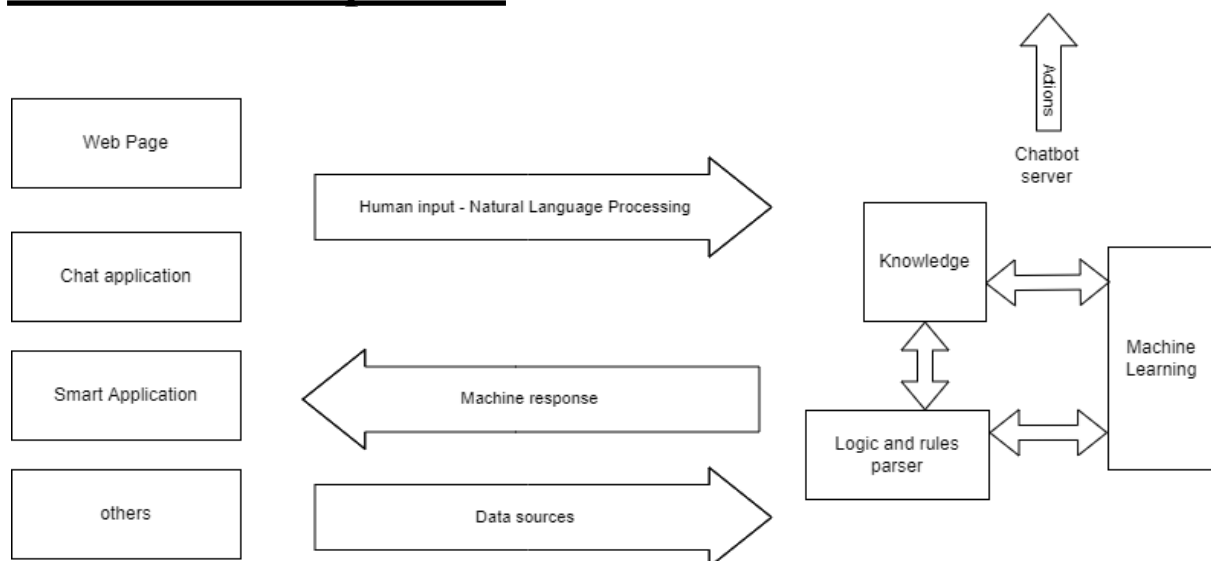
Responsible for understanding the meaning and intent of what the user is trying to say.

### Chatterbot Corpus

The ChatterBot Corpus is a project containing user-contributed dialog data that can be used to train chat bots to communicate.

## 5. High-Level Design

### Chatterbot Composition



## 6. Preliminary Schedule

	Task Name	Duration	Start	End
1	<b>3rd Year Project</b>	<b>71 days</b>	<b>08/12/2021</b>	<b>16/03/2022</b>
2	Exam Period	8 days	08/12/2021	17/12/2021
3	Project Start	0 days	17/12/2021	17/12/2021
4	Christmas Holidays	5 days?	20/12/2021	26/12/2021
5	<u>Backend Code (Prototype 1)</u>	15 days	26/12/2021	16/01/2022
6	Design GUI	0 days	17/01/2022	17/01/2022
7	<u>FrontEnd Code (Prototype 1)</u>	7 days	18/01/2022	26/01/2022
8	Testing (Prototype 1)	2 days	28/01/2022	31/01/2022
9	Create And Finish Prototype 2	14 days	01/02/2022	20/02/2022
10	Testing	3 days	21/02/2022	23/02/2022
11	Documentation	3 days	24/02/2022	28/02/2022
12	Project End	0 days	01/03/2022	01/03/2022
13	Submission Of Final Project...	1 day	04/03/2022	04/03/2022
14	Demonstration	8 days?	07/03/2022	16/03/2022

	Tas...	Task Name	Duration	Start	End	Dec 21	13 Dec 21	20 Dec 21	27 Dec 21	3 Jan 22	10 Jan 22
1	🚩	3rd Year Project	71 days	08/12/2021	16/03/2022						
2	🚩	Exam Period	8 days	08/12/2021	17/12/2021						
3	🚩	Project Start	0 days	17/12/2021	17/12/2021						
4	🚩	Christmas Holidays	5 days?	20/12/2021	26/12/2021						
5	🚩	Backend Code (Prototype 1)	15 days	26/12/2021	16/01/2022						
6	🚩	Design GUI	0 days	17/01/2022	17/01/2022						
7	🚩	FrontEnd Code (Prototype 1)	7 days	18/01/2022	26/01/2022						
8	🚩	Testing (Prototype 1)	2 days	28/01/2022	31/01/2022						
9	🚩	Create And Finish Prototype 2	14 days	01/02/2022	20/02/2022						
10	🚩	Testing	3 days	21/02/2022	23/02/2022						
11	🚩	Documentation	3 days	24/02/2022	28/02/2022						
12	🚩	Project End	0 days	01/03/2022	01/03/2022						
13	🚩	Submission Of Final Project de...	1 day	04/03/2022	04/03/2022						
14	🚩	Demonstration	8 days?	07/03/2022	16/03/2022						





## 7. Appendices

- Flask: <https://flask.palletsprojects.com/en/2.0.x/>
- Chatterbot Corpus: <https://chatterbot.readthedocs.io/en/stable/#process-flow-diagram>