CA400 Final Year Project

Functional Specification

DCU Campus Chatbot

Sam Wood - 16476066
Mark Agnew - 16315811

Project Supervisor: Gareth Jones

0. Table of Contents

1. Introduction

1.1 Overview

The application we plan to implement for our final year project is a chat bot web app that accommodates queries made about the Dublin City University campus. The areas of computer science we will cover in this project are Web applications, Software Development, Information Retrieval, Machine Learning and also the overall development of a chatbot. An example case of the app being used would be a user texting the app ‘Where is XG?’ and the system would quickly reply, ‘The bottom floor of the Science Building’. The service will not only be aimed at DCU students but also exchange students, secondary school students on open days, guest speakers and visitors. It is intended that the application be as easy to use for our clientele as possible. Our supervisor has informed us that it is acceptable to use software like Dialogflow to develop our app, instead of a machine learning algorithm from scratch, provided there has been difficulty in the project elsewhere. We will also be employing the use of languages like HTML, Javascript, CSS, Python and possibly Java.

For the first stage of our project we would like our system to act as a tool for finding lecture rooms around the campus. We believe that this would be useful given the clear ambiguity for some of the codes for various buildings around the campus. For example it is fairly easy to confuse ‘HG’ (bottom floor of the nursing building) with the Henry Grattan building, which has the code ‘C’.

Fig 1.1 DCU Glasnevin Campus map and Building Codes.

1.2 Business Context

The reason we plan to implement this application as a web service is so it can potentially be linked with other online platforms such as loop.

Market Availability

Chat bots are a very popular topic right now in the Information Technology industry. Given the popularity of instant messaging in recent years through online services instead of standard SMS messaging it seems very unlikely that a person selected at random would be unfamiliar with how to operate a chatbot. For this reason, we believe that having this service would be very intuitive and well suited for the clientele we are intending to have. I am also yet to hear of any universities that employ a chatbot service for their frequently asked questions. Having said that, should this project be a massive success other universities such as UCD and Trinity may take to this new trend and adopt a similar service for their own campus'.

Advertising

If the campus chatbot were to be live, it would be advertised regularly on the DCU website, on Loop, around the Glasnevin campus, also St Pat's, all Hallow's and on the DCU social media pages like Facebook, Instagram, Snapchat and Twitter. This publicity would be enormous for the app given how often these services are used nowadays, particularly by college students.

1.3 Glossary of terms

Dialogflow - This is a Google-owned developer of human-computer interaction technology based on natural language conversations.

Entities - This is a property in Dialogflow which can be used to answer the request from the user. The entity will usually be a keyword within the request like the name, date or location.

Intents - This categorizes an end-user's intention for one conversation turn. You would likely define an intent for questions about locations or directions.

Contexts - Contexts represent the current state of a user's request and allow your agent to carry out information from one intent to another. In other words Context act as a chain for the intents.

HTML - Hypertext markup language.

AIML - Artificial Intelligence markup language

IDE - Integrated development environment

PERT - Program Evaluation and review technique

API - Application Programming Interface

GUI - Graphical User Interface

2. General Description

2.1 Product System Functions

Help Message

The user will be able to message our system saying ‘Help’ and possibly other messages depending on how well we will train it. This message will send the user a concise description of the chatbots functions to guide the user on how to operate it.

Building codes

This is the first issue that we will program given it will most likely be the easiest to implement. As seen in section one, our aim is to eliminate the ambiguity that is found on the DCU campus maps by allowing the customer to send queries about what each one means.

Directions to Buildings and rooms

Most DCU students will tell you that some of the lecture rooms around campus can be very difficult to find. We want our chatbot to be able to answer questions about all lecture rooms , for example ‘Where is cg12?’ and reply with a concise and helpful description of its location.

Conversational Maturity

A good chatbot service has good and specific natural language processing abilities. This means being able to understand the context of a conversation. In other words, advanced conversational capabilities and also being able to ask questions for clarification from the user if the last message sent by the user is unclear or ambiguous.

Free to Explore

We want our application to process and be able to easily search through the vast amounts of data. If it can do this, the service is more likely to be able to gather the relevant text and respond to the user quickly and with the correct response.

Omni-capable

We would want the system to be able converse across multiple digital channels. For example customers hand held devices, laptops, desktops etc. This also means being able to retain data and context over the multiple conversations held over these channels for continuous improvement in its abilities.

Reason Autonomously

The algorithm should be programmed to be able to perform the required complex reasoning on its own, without human intervention. For example, with the aid of free exploration, come to conclusions for questions based on relevant case histories.

Pre-trained

Our application will be pre-trained to understand case-specific terms. For example it will not crash or become confused when a user asks ‘Where would I find CG12?’ as the building codes will be defined within the system. This will include things other than the building codes, as in it will be aware that the user’s reference to ‘The Library’ and ‘The Gym’ would mean the campus facilities.

In Accordance with the Gestalt Principle

A good guideline in user interface design is ‘don’t make the users think’. It can be a hard thing to do. The Gestalt theory implies that the mind understands the whole is greater than the sum of its parts. An example of a messaging service that employs this service is the facebook messenger app ie. an aesthetically pleasing application that the user finds intuitive to operate.

2.2 User Characteristics and Objectives

Target Audience and expertise

The target audience will be any person who wants to get information about DCU campus whether they are Students, guests or even staff. We expect some people to be familiar with chat bots and how to use them, but for those who have little expertise we aim to make it as intuitive as sending a regular text. This means one of the requirements from the users perspective is to have a simple and aesthetically pleasing interface similar to that of something like Facebook or Whatsapp. That being said we may also have to take into account certain visual impairments and cater the look to them allowing users the option to select options that help make the text more easily visible to them. The main overall objective of the system from the user’s perspective is to answer any basic questions about DCU’s main campus for example, “where is XG21?” or “How much are the fees for the year?”. To begin with we will be starting only focusing on queries relating to where a room or building is, to create a working product within the time frame, but would hope to also answer a range of questions in the long term especially ones that currently take the most out of staff member’s time. We could also potentially port it over to be part of loop or its own individual app for mobile devices to make it more accessible.

Business and Project Objectives

Our web app aims to give users easy access to information about DCU and it's campus. We chose this idea as there is currently no such app or web app for a chat bot available for DCU as well as many other colleges across Ireland. This mean that this project could potentially be adapted for other colleges in the future, therefore expanding the business. Another objective is to answer questions that would otherwise take up the time of staff members and therefore allowing them time to focus on other important work.

2.3 Operational Scenarios

USE CASE 1	**Start the Chat bot**
Goal in context	A user should be able to connect to the chat bot
Scope & Level	Subfunction, low level
Preconditions	None
Success end conditions	User receives startup help message
Failure end conditions	Chat bot fails to start, returns error message
Primary, secondary actors	System, User
Trigger	User connects to the chat bots URL
Steps	
1	User connects to URL
2	Call made to start chat bot
3	Chat bot starts
4	Chat bot sends welcome help message to user

USE CASE 2	**Send a query to the chat bot**
Goal in context	A user should be able to send a message with a query to the chat bot
Scope & Level	User goal, High level
Preconditions	Chat bot started successfully
Success end conditions	Message is sent to chat bot for interpretation
Failure end conditions	Message fails to send
Primary, secondary actors	User
Trigger	User types message and hits send
Steps	
1	User types query into text box
2	User hits enter key or send button
3	Query entered passed to the chat bot

USE CASE 3	**Request help message**
Goal in context	The user should be able to display a help message to understand how the chatbot works
Scope & Level	Subfunction, High level
Preconditions	User is on the web app
Success end conditions	User receives help message from chat bot
Failure end conditions	User doesn’t receive help message
Primary, secondary actors	User
Trigger	User wants to understand what the chat bot can do
Steps	
1	User enters help into text box
2	User hits send or enter key
3	Chat bot processes message
4	Help message is returned to user

USE CASE 4	**Open Options menu**
Goal in context	The User should be able to open an options menu to access various settings
Scope & Level	Subfunction, High level
Preconditions	User is on the web app
Success end conditions	An options menu is opened on screen
Failure end conditions	No menu is opened
Primary, secondary actors	User
Trigger	User wants to access the apps settings
Steps	
1	User clicks the options button on the web apps page
2	Web application opens the options menu
3	User is presented with all the various options and settings

USE CASE 5	**Change text font size**
Goal in context	The User should be able to adjust the size of text for the web application
Scope & Level	Subfunction, High level
Preconditions	The user has successfully opened the options menu
Success end conditions	The font size for messages is increased or decreased
Failure end conditions	Text size remains at default
Primary, secondary actors	User
Trigger	User wants to change the font size of messages
Steps	
1	User selects change text size option from menu
2	User selects which font size they would like from a list displaying all possible options
3	Once the correct size is selected the user clicks apply and can return to the chat bot

USE CASE 6	**Change GUI colours**
Goal in context	The User should be able to change the colours to better suit certain visual impairments
Scope & Level	Subfunction, High level
Preconditions	The user has successfully opened the options menu
Success end conditions	The GUI colours get changed to suit the user
Failure end conditions	The GUI colours remain at the default
Primary, secondary actors	User
Trigger	User wants to change the interface colours
Steps	
1	User selects change colours option from menu
2	User selects an option to change background, sent message, received message or text colour.
3	The User then selects a preset colour or can access a colour palette to select one that suits them best
4	Once User has selected their colour options they click apply so save them

2.4 Constraints

Neural Network

Sam has been given the task to implement the Neural Network for the application should one be implemented following our progress on DialogFlow. Given the only experience he has with this area has been Charles Daly's module in third year, this will be a challenge for him. Although given that Mark worked on a machine learning application for his third year project, Sam may refer to him for consultation should he need to.

Processing Natural Language

There can be many ways to ask the same question. Hence processing natural language and handling queries when dealing with a chatbot service may cause a few setbacks in this projects development. For example 'Where do I get my student leap card?', 'Which place on campus do I get my leap card?' and 'What building do I get my leap card in?' would all have to be interpreted as the same question by the chatbot.

Another issue regarding the process of natural language is SMS language. This is the shorthand language that many of us use on mobile phones. An example is people removing vowels and taking the last letter of whatever word they are sending to make their message more concise and thus the language would not be full english. This will have to be taken into account when designing our project.

Web Application

Neither partners for this project have designed a web app with backend functionality or a chatbot for that matter. Therefore it will be a learning experience, however it may prove to be more of a challenge than if one of us had designed one previously. This is not one of the more pressing constraints.

Testing

There will have to be research done for the testing section of our project. We will consider revising some of the work done in our Human Computer Interaction module with Donal Fitzpatrick. This class is a good example of when we have been exposed to interface testing for example Schneiderman's eight golden rules. Also, since neither partner was exposed to any kind of testing while on our INTRA work placements, testing may not be the most simplest of our tasks this year.

3. Functional Requirements

3.1 User start-up

Description

When a member of our clientele selects the chatbots application icon from their mobile, desktop etc. he/she should be subsequently directed to the text message environment with its keyboard and possibly a welcome or greeting message. Our app will not be usable if this environment can't be accessed and it will be perceived as more useful and reliable if it is not frequently or never crashing for its users.

Criticality

This is a critical part of our app as the app is obsolete if queries can not be made in the messaging environment. Were our project live for people on campus to use, it would have to be accessible with ease. The good reputation of this service would also be maintained if it wasn't known to crash or become obsolete on a regular basis.

Dependencies / Technical Issues

The start-up would be dependant on the strength and connectivity capabilities of our server. For our server to live up to its expectation, will take a substantial amount of work and research as this is an unfamiliar topic for both partners.

3.2 Algorithm capable to train itself

Description

We want machine learning to be employed in our project. Our potential algorithm would have sections consisting of Intents for what the aim of the question asked will be, Entities for what the question might be regarding for example Building location and building opening times and possibly other sections that we have not yet concluded. We would also hope that our chat bot is robust in its ability to communicate and interpret the language used by the user and learn from it to improve.

Criticality

It is essential that our machine learning aspect of the project is well done. If we are not able to cover as many areas of queries as possible (which we were warned about by our panel in our proposal), we would like to be able to provide a certain service that is functional. We chose the location of lecture rooms around the campus as our fall back. It is important that we meet at least this requirement should the project prove to be too difficult for our abilities.

Dependencies

For our algorithm to fully be usable to the app holders, it will be dependant on the servers capabilities. The standard of the algorithm will be dependant on how much work is put into the project itself and how much research is done on the topic.

Technical issues

In our proposal presentation for this project we were warned that machine learning is a project in itself. For this reason we were asked was the plan to create our own Neural Network or using a software that does it for us. Because of this we thought to use DialogFlow to develop a prototype for our project.

3.3 Server Connectivity

Description

If we want the chatbot to be fully functional it will have to be constantly connected with the server code. This is so the user can use it for their queries on and away from the DCU campus.

Criticality

The Server code and the Machine learning are the 2 largest components of the project. If there is not a sufficiently designed server that allows our users to use the chatbot correctly a huge part of the systems total functionality would be missing from the project.

Dependencies

The server itself has possibly the most responsibility for the campus chatbot. For the server to fill its purpose effectively the back end/ functionality will have to be trained to its optimum point of intelligence. In other words there is no use in having a functional server if what it is intended for is not useful. Much like the algorithm, the standard of the server will be dependant on how much work is put into the project itself and how much research is done on the topic.

Technical issues

This project is the first time either partner will be exposed to programming a server from scratch. Hence there will have to be some research done on this. This is the only issue with this functional requirement that can be recognised at this point given that we were not given any warnings about it by the lecturers in our proposal meeting.

3.4 Parse queries

Description

We will need our system to be able to interpret what the user is asking by parsing for the keywords in the message. For example, if the user was to ask ‘Where is HG21?’, the chat bot would have to recognise the ‘Where’, making the query location related and also the HG as one of the building entities.

Criticality

If the system isn’t able to recognise the keywords in the query, it won’t make reference to the dataset and therefore will be incapable to respond to the user effectively. For this reason it is essential that the keywords and entities are acknowledged by the chatbot. This functional requirement will be needed if the chatbot to recognise the different intents.

Dependencies

The chatbots ability regarding identifying keywords will depend on the structure and quality of whatever data structure we use to contain our intents and entities. For example, contexts are used to identify what the query is about (literally the context of the question) e.g location, directions etc.

Technical issues

The only issue that can be seen to date is lack of Sam’s knowledge on the matter, given he is responsible for the data set and back end. Additional research and revision of tutorials would be useful to deal with the matter.

3.5 Decypher Campus building codes

Description

This will be the first function that we will aim our chat bot to have. As said in the overview, there is a significant amount of ambiguity with the building codes. A very common example is the Henry Grattan building (which has the code C) being mistaken for the bottom floor of the Nursing building (HG). Another is the Sports complex (U) being mistaken for the Students Union building (referred to colloquially as ‘The U’). Our users will be able to ask our system what is the meaning of whichever building code and the chatbot will answer correctly. For this the Buildings would be stored as entities in our dialog system.

Criticality

We were told by lecturers already that we will not be able to answer every query about the campus and also that if we were able to provide this function and possibly directions to lecture rooms that are chatbot would be very useful. Hence or otherwise we will strive to meet this requirement.

Dependencies

The quality of this feature will depend on how we structure our entities, intents and contexts in Dialogflow. With a suitable amount of research on the matter and the right amount of work put in, we trust that this requirement will be working well.

Technical issues

An issue that may arise when programming this requirement is processing the natural language of our user questions. We mean this in the sense that there is more than one way to ask the same question, ie. 'What is PR?', 'What does PR mean?', 'Which building is the PR building?'. It will be important that this issue is addressed so the chatbot is as usable as it can be.

3.6 Directions to facilities and rooms on campus

Description

This feature would let the user ask the chatbot for the location for a certain lecture room or building and the exact location would be provided with a detailed description of how to get there. To do this the building entities in Dialogflow will be put to further use and we will have entities for the rooms also.

Criticality

This is the second requirement we will aim to have in our system. In the feedback we received from our lecturers, we were told that this feature would be a very good feature to have and would most likely be used more than the decode feature. Therefore, to contribute to the usability and therefore likability of our project.

Dependencies

Similar to the last functional requirement, this feature will depend on how well we program our dialog flow with the necessary entities, intents and contexts.

Technical issues

Previously noted, issues may arise with natural language processing with this feature. Also, if a user is looking for directions and they are unfamiliar with areas our chatbot uses in its answer, the chatbot may not be able to help the user. For example if the user asks 'How do I get to the Library?' and the user uses the Invent building as a means of direction, the answer will not be effective.

3.7 Answer general questions about DCU campus

Description

This function would be the most difficult to implement. An example of a general query that someone on the campus might have is 'Where do I enquire about getting an access grant'. As you might guess. A lot of work would have to be put in if this function is to work successfully.

Criticality

This is the feature of our project that we will tackle last, given that it is definitely the most challenging. As having at least one solid function fully implemented this feature will be tackled after that has been seen to. In other words, This is one of the less critical features we will be working on in the project.

Dependencies

This feature will depend on how long it takes for our other functions to be fully implemented.

Technical issues

We were told that no matter what, we will not be able to answer all the queries about the DCU campus. Therefore, we will have to choose what areas would be most useful to people in DCU. This could be hard as people's opinions might vary quite alot on the matter.

3.8 Data Store

Description

At the minute we are using DialogFlow which means it will act as our datastore. Should we choose to use AIML we may need a different data store for replies to queries. AIML also provides this function.

Criticality

Obviously, it is essential for our project to have a data store since without one there can be no possible answers to give. It is also important to have one to allow for expansion going forward, this will enable us to add more responses for different query topics.

Dependencies

Our data store will be dependant on the information given by the possible surveys we may distribute to DCU students and also research done ourselves. This data store will rely on the information we provide in order to answer a user's query. Both are important attributes of the project.

Technical issues

There are certain machine learning algorithms used by DialogFlow that are used for responding to user queries, these are done in the background and not visible to the developer. AIML however allows us access to more information but is more technically complex. These are the areas that we can see technical issues arising

4. System Architecture

4.1 architecture diagram

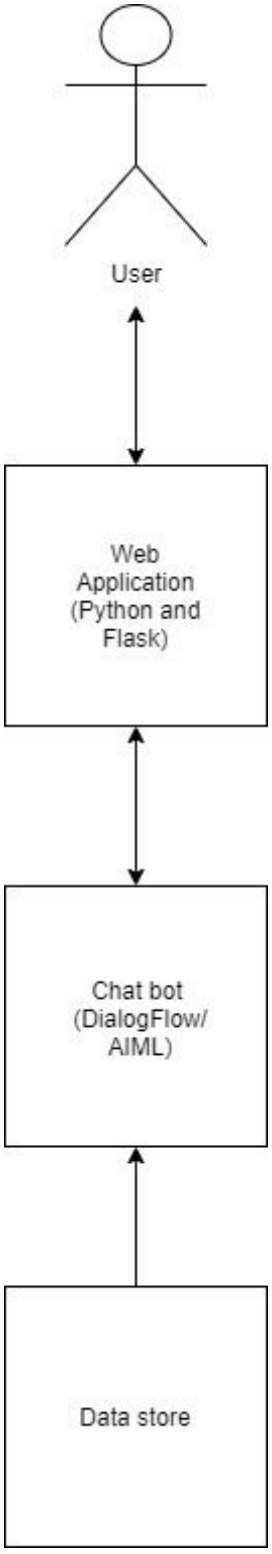


Fig 4.1

4.2 Web Application

The web application will be how the user interacts with our chatbot, it will provide a GUI to the user so make interaction with the bot as intuitive as possible. It will provide the user with a text box to send messages to the chat bot and allow them to access the options menu to change various visual settings. The web app will also allow the chatbot to show its response to the user. There are many different languages and API's that can be used to build web applications, for this application we are hoping to use python and flask to build our web application but may look into other options.

4.3 Chat bot

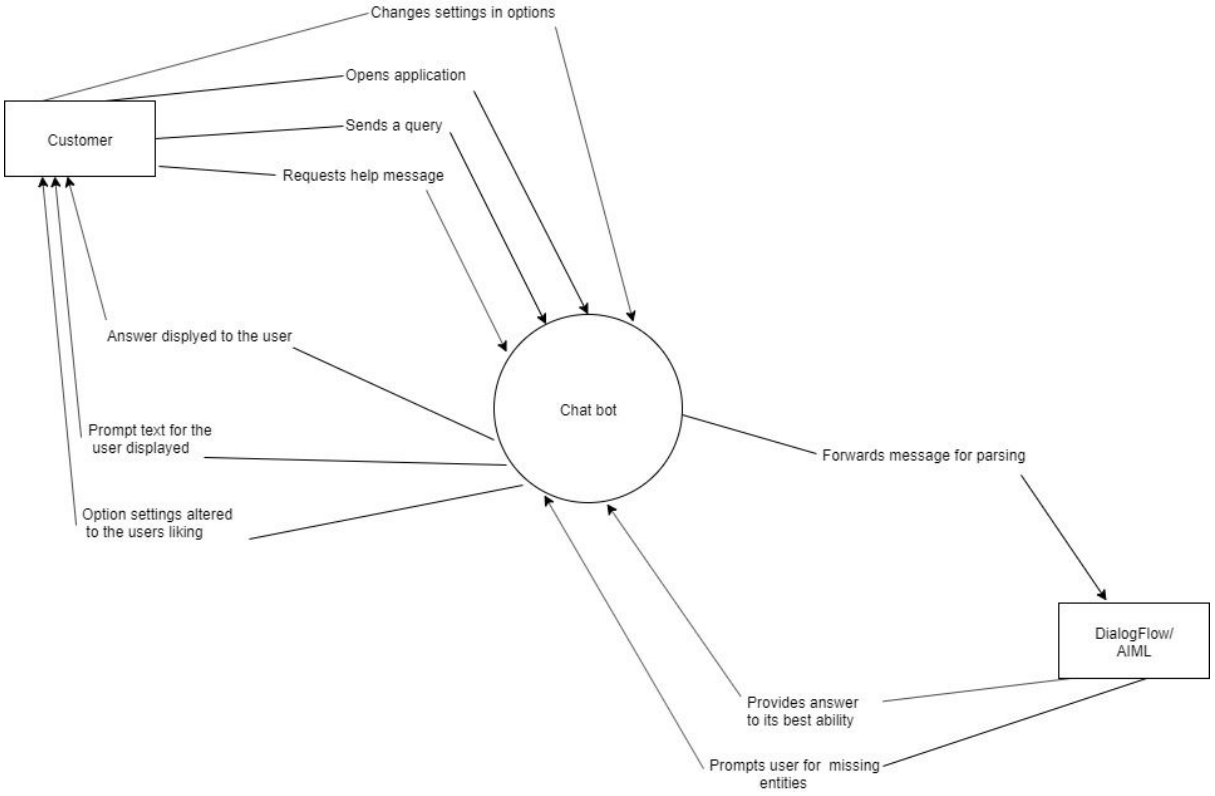
The chat bot is what will handle users questions and provide answers to those questions. It will be used to interpret those queries then use a store of information to give the most relevant reply. To begin we will be using Google's API called DialogFlow but we are also considering using AIML to create our chat bot as it allows for more functionality as it's not a third party API.

4.4 Data store

The information store will be some sort of file or database containing all relevant information for the chat bot to provide to the user. This information will then be given to the chat bot for use in answering users queries. This information store can then be later expanded on to allow the chat bot to deal with more varying queries and new topics entirely. To begin this information store will be housed within DialogFlow itself but may change moving forward.

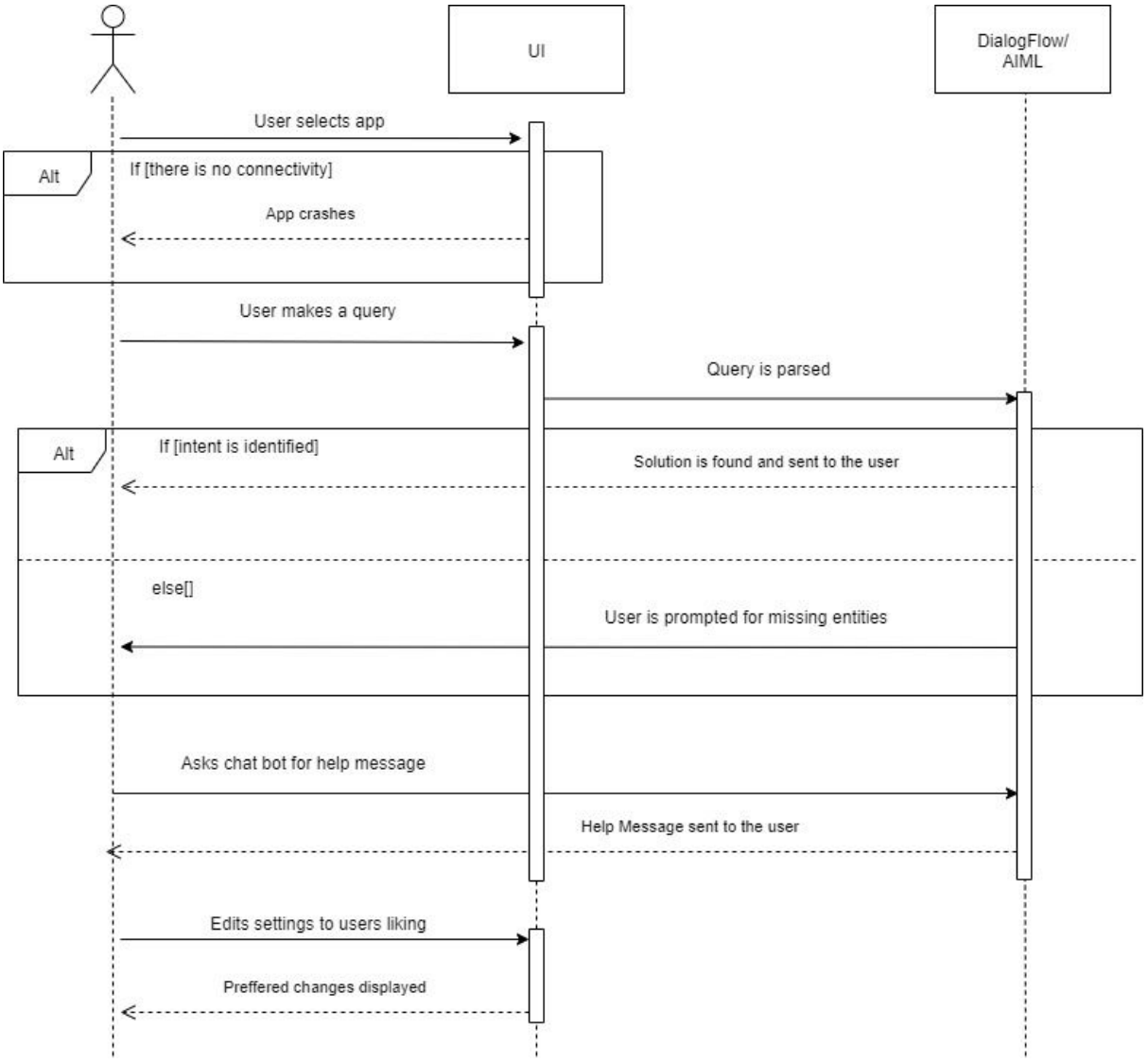
5 High Level Design

5.1 Context Diagram



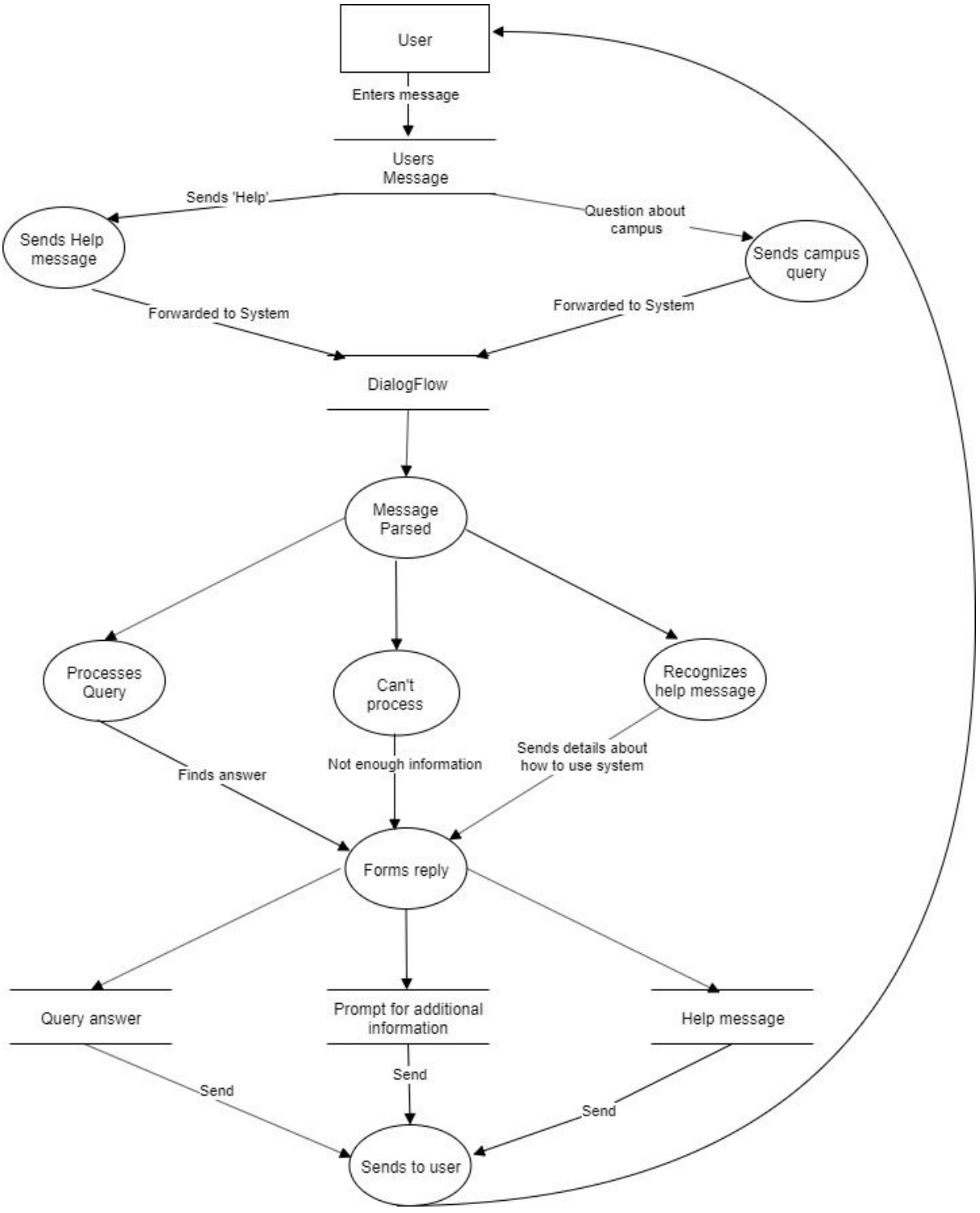
The above is our context diagram or our level 0 data-flow diagram. Our system is viewed as one big process, the chat bot, the circle in this case. The external entities are the Customer and Dialogflow, represented in rectangles.

5.2 Sequence Diagram



Here is our sequence diagram. We have depicted the most common interactions for a user with our chatbot. In this case if the user selects the app and it crashes, the user makes a query, the user sends the help message and if the user changes the settings in the options.

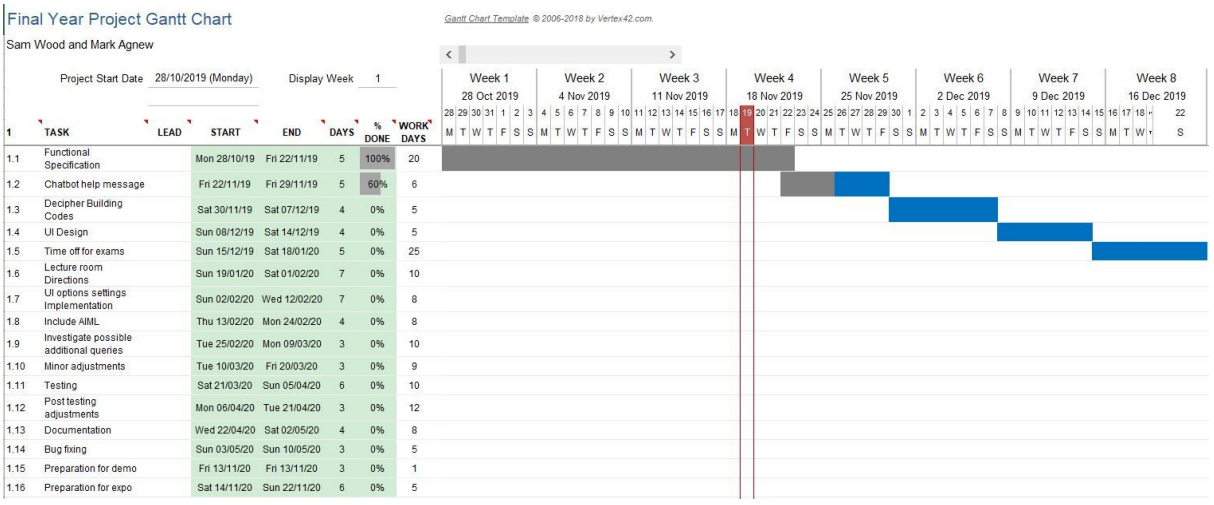
5.3 Logical Data Flow diagram



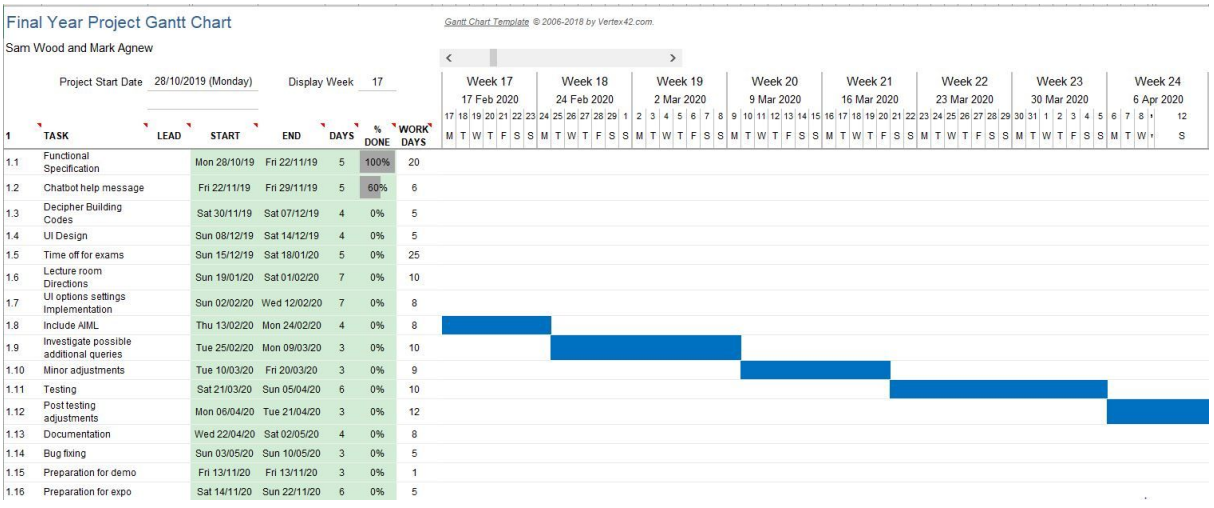
This is our projects data flow diagram. This is the representation of the flow of our data from our entities to the various data stores through the processes. Once the system has responded to the user the process will repeat itself.

6. Preliminary Schedule

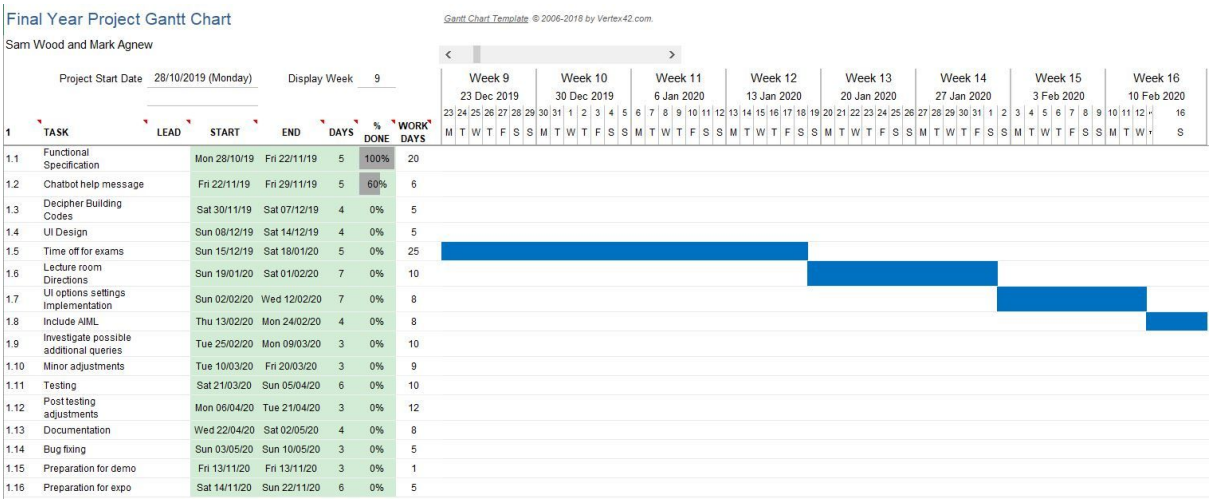
6.1 Gantt chart

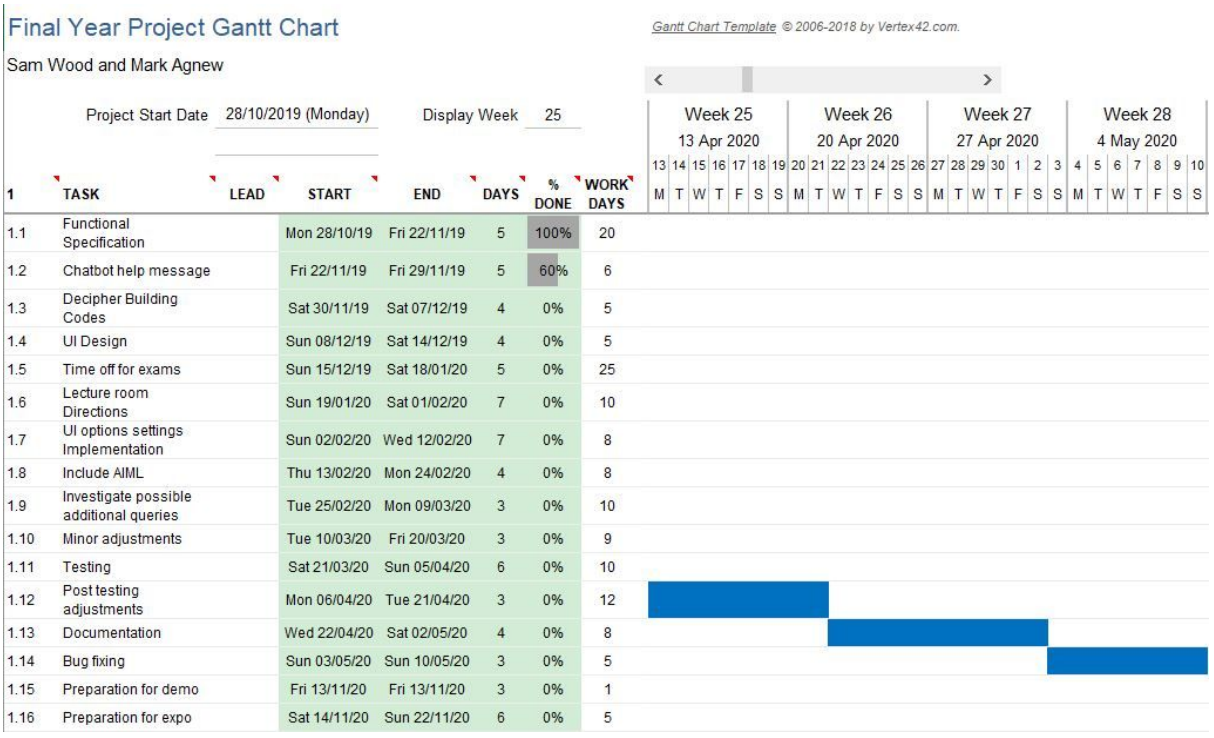


1	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1.1	Functional Specification		Mon 28/10/19	Fri 22/11/19	5	100%	20
1.2	Chatbot help message		Fri 22/11/19	Fri 29/11/19	5	60%	6
1.3	Decipher Building Codes		Sat 30/11/19	Sat 07/12/19	4	0%	5
1.4	UI Design		Sun 08/12/19	Sat 14/12/19	4	0%	5
1.5	Time off for exams		Sun 15/12/19	Sat 18/01/20	5	0%	25
1.6	Lecture room Directions		Sun 19/01/20	Sat 01/02/20	7	0%	10
1.7	UI options settings Implementation		Sun 02/02/20	Wed 12/02/20	7	0%	8
1.8	Include AIML		Thu 13/02/20	Mon 24/02/20	4	0%	8
1.9	Investigate possible additional queries		Tue 25/02/20	Mon 09/03/20	3	0%	10
1.10	Minor adjustments		Tue 10/03/20	Fri 20/03/20	3	0%	9
1.11	Testing		Sat 21/03/20	Sun 05/04/20	6	0%	10
1.12	Post testing adjustments		Mon 06/04/20	Tue 21/04/20	3	0%	12
1.13	Documentation		Wed 22/04/20	Sat 02/05/20	4	0%	8
1.14	Bug fixing		Sun 03/05/20	Sun 10/05/20	3	0%	5
1.15	Preparation for demo		Fri 13/11/20	Fri 13/11/20	3	0%	1
1.16	Preparation for expo		Sat 14/11/20	Sun 22/11/20	6	0%	5



1	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1.1	Functional Specification		Mon 28/10/19	Fri 22/11/19	5	100%	20
1.2	Chatbot help message		Fri 22/11/19	Fri 29/11/19	5	60%	6
1.3	Decipher Building Codes		Sat 30/11/19	Sat 07/12/19	4	0%	5
1.4	UI Design		Sun 08/12/19	Sat 14/12/19	4	0%	5
1.5	Time off for exams		Sun 15/12/19	Sat 18/01/20	5	0%	25
1.6	Lecture room Directions		Sun 19/01/20	Sat 01/02/20	7	0%	10
1.7	UI options settings Implementation		Sun 02/02/20	Wed 12/02/20	7	0%	8
1.8	Include AIML		Thu 13/02/20	Mon 24/02/20	4	0%	8
1.9	Investigate possible additional queries		Tue 25/02/20	Mon 09/03/20	3	0%	10
1.10	Minor adjustments		Tue 10/03/20	Fri 20/03/20	3	0%	9
1.11	Testing		Sat 21/03/20	Sun 05/04/20	6	0%	10
1.12	Post testing adjustments		Mon 06/04/20	Tue 21/04/20	3	0%	12
1.13	Documentation		Wed 22/04/20	Sat 02/05/20	4	0%	8
1.14	Bug fixing		Sun 03/05/20	Sun 10/05/20	3	0%	5
1.15	Preparation for demo		Fri 13/11/20	Fri 13/11/20	3	0%	1
1.16	Preparation for expo		Sat 14/11/20	Sun 22/11/20	6	0%	5





The above figures are of the tasks to complete our project by way of a Gantt chart. We have allocated each task an amount of time we have considered fit. The duration for each task may change depending on unforeseen obstacles or how difficult the task may turn out to be.

6.2 Implementation Details

Our plan thus far

The Gantt chart depicts the order of the tasks for our project. They have been ordered in the way we have thought most suitable regarding difficulty and if required for the tasks that are at a later date. For example we thought it suitable to have tasks like building codes and directions saw to first in DialogFlow before implementing AIML in the project. We thought it fit to have these functions ready before we try to improve the Artificial intelligence in the chatbot as we fear it may lead to complications.

Also, from past experience with our third year projects we believe that it is necessary to have three time slots for making changes. One after finishing implementation, one after testing and one for fixing bugs. This is because there are always minor complications revealing themselves over the course of software projects. This alas, we cannot prevent. These time slots were made to try and produce the cleanest and best working project that we can provide.

6.3 Further Constraints

Collection of Data

With regard to the dataset for our project, we have already contacted the registry for frequently asked questions by people on the campuses. They replied with six or seven questions hence we have not been able to obtain a huge dataset. For this reason, If we do choose to broaden the usage for our chat bot, we will more than likely have to make our dataset manually. This may be difficult.

Cost of collection of Data

We will have to be careful with how much time we spend when we are collecting our data. Since there are so many questions that can be asked, it may take an unexpectedly long time to collect the right data and implement it in our DialogFlow. Our other tasks may suffer as a result.

Scope

We have been warned about the scope for our project being too ambitious. We intend for the chatbot to be as useful as possible ie. answer as many questions as possible. In our proposal, the panel told us we will simply not be able to able to answer all questions about the campus and that we must draw the line somewhere. For example, we may begin with answering where all of the lecture rooms are.

7. Appendices

Youtube tutorials
https://www.youtube.com/watch?v=yT58gTXdQb8
https://www.youtube.com/watch?v=gWNUg_v25dw&t=274s

Diagram Tutorials
https://www.youtube.com/results?search_query=dfd+diagmram
https://www.youtube.com/watch?v=pCK6prSq8aw
https://www.youtube.com/watch?v=fWNrc6GNK14

DialogFlow pages
https://chatbotslife.com/dialogflow-restaurant-bot-tutorial-1-45ce1d3c0ab5

DialogFlow API
https://dialogflow.com/