

CA400 Functional Specification

DCU Personal Assistant Chatbot

Ian Anthony Kelly

13480138

Date Completed: 20/11/2017

Table of Contents

Table of contents	1
1. Introduction	1
2. General Description.....	2
3. Functional Requirements.....	4
4. System Architecture	5
5. High-Level Design	6
6. Preliminary Schedule	8
7. Appendices	9

1. Introduction

1.1 Overview

The DCU Personal Assistant Chatbot (DCUPAChatBot) will be an app designed with the purpose of improving the quality of life among DCU students. It will do this by providing them with key information that students often require during their time in college. The DCUPAChatBot will enable them to seek this information from a chat interface.

DCU students also currently have difficulty with gaining answers to basic questions using the existing web services at DCU. When a DCU student needs to find out when and where their next lecture will be, they need to google the DCU website, find the timetable search page, search and select their course, and then find the time of the lecture on timetable. This takes significant time among requiring the student to stop walking and possibly waste phone data.

The student will be able to ask the bot a question regarding DCU, the bot will process what the student is asking and then respond with the appropriate data. This information may include but is not limited to, the time of their next lecture, when their next bus will arrive, or a frequently asked question about the college itself. The app will be implemented into Facebook Messenger for ease of access.

1.2 Glossary

LUIS : Language Understanding Intelligence Service. A product created by Microsoft for natural language processing.

Intent : A part of LUIS. These refer to the function names extracted from the text input e.g. "What lecture is next" might extract an intent GetNextLecture.

Entity : A part of LUIS. These refer to the parts of the text input that were used to find an intent.

API : Application programming interface.

2. General Description

2.1 Product / System Functions

Once the DCU student has set up the chatbot on Facebook Messenger, the user will be able to have a conversation with the bot. The user can ask a question to the bot with natural language and the bot will be able to interpret what the user is asking and answer the question with a suitable response. For this to be a successful replacement to manually looking up information and for user to be able to use this bot daily, the bot needs to have certain functionality:

1. The bot needs to be able to reply to the user's question with the correct response. The user is not interested in incorrect bus times or lectures.
2. The natural language processing needs to be able to handle the vast majority of questions that users might ask. If the user cannot get an answer to their questions they will not use the bot. This will encompass requiring extensive LUIS training and testing before publishing the custom language.
3. The chatbot response times need to be reasonable. If the responses take too long users will not be willing to wait.
4. The bot will have to be familiar with jargon and names relating specifically to DCU that the user might use e.g. 'Nubar', 'Helix', 'Shite Night'.
5. It needs to be able to remember information about the user (bus stops and degree taken) without having to ask each time.
6. The chatbot should not ask for or store private details of the user for security reasons. This would exclude data such as bus stops used or degree being taken.
7. Chatbot responses such as locations of campus buildings need to be concise and unambiguous to prevent user confusion and follow-up questions.

2.2 User Characteristics and Objectives

The chatbot will be specifically for students of Dublin City University. Due to the variety of people that can be a student, expertise and experience with Facebook Messenger and chatbots can vary. DCU students are typically in the age range of 18 to 24 with medium to strong knowledge of phone applications and more than likely have Facebook Messenger already installed on their phone. There are outliers to this but the aim of the chatbot is to be user friendly and easy to use so that the only thing a student needs is to be able to ask it a question and possibly reply to a question if the bot requires certain information.

The goal of the chatbot will be to replace the need for students having to go to websites such as the DCU website and the Dublin Bus website individually in order to look for their next lecture or bus. The chatbot will combine these and potentially other services into a single app.

Functions of the chatbot would ideally include:

- Bus times
- Lecture times and locations
- DCU event calendar
- Campus building and facility locations
- DCU FAQ
- Library Search
- Faculty Information
- Weather
- DCU Updates and Opening/closing times

2.3 Operational Scenarios

Due to the simplistic nature of the chatbot user interface and the focus on easy of use, there will be a restricted amount of operational scenarios that the user can experience.

Setting up the chatbot

Description: The initial setup before the chatbot can be used.

Precondition: The user does not already have the chatbot set up.

Actors: DCU Student / User.

Activation: The user wants to use the chatbot.

Steps:

1. The user logs into Facebook Messenger.
2. They type "DCUPAChatBot" into the search box and hit enter.
3. They select the app in the search results.
4. A chat window will appear.
5. The user can ask a question.

Termination: The chatbot is set up and is ready to be used.

Asking a Question

Description: The general process of the user asking the chatbot a question.

Precondition: The user does not know about something.

Actors: DCU Student / User.

Activation: The user would like to ask the chatbot a question.

Steps:

1. The user logs into Facebook Messenger.
2. They open the chat window for the chatbot.
3. They type in their question and hit enter.
4. They observe the bot's response.

Termination: The user has had their question answered satisfactorily.

Possible Steps:

- 4a. The bot has asked the user for additional information.
- 4b. The user types their response to the bot's question.
- 4c. They observe the bot's response.

2.4 Constraints

- There is a deadline for this project to be completed.
- The chatbot cannot be a stand-alone application on Android phones due to the incompatibility of resources and Javascript being unsupported on Android Studios.
- The bot requires internet connection to work.
- The bot cannot be used without installing Facebook Messenger on the phone.
- It is not feasible to cover every part of DCU life that the user might have questions about.

3. Functional Requirements

Natural Language Processing

- Description – The bot needs to be able to take the user input and extract intents.
- Criticality – This is critical to the bot. Without intents the bot cannot function.
- Technical issues – The entities that the user has included in their input may not have been added to the LUIS language and intents cannot be extracted.
- Dependencies with other requirements – Dependent on data storage if the entity from the user input needs to be taken from saved information.

Data Retrieval

- Description – The bot needs to be able to retrieve correct information from external resources to give a suitable reply to the user.

- Criticality – This is very important as replying to the user with incorrect information is detrimental to the usefulness of the chatbot.
- Technical issues – **Retrieving** the correct information using the APIs may be difficult or impossible. What the user is asking for may not be available on the external resource.
- Dependencies with other requirements – This relies on the language processing sending the bot correct intents.

Data Storage

- Description – The bot may need to store certain information in order to prevent asking the user each time.
- Criticality – Not critical to the bot but greatly improves usability.
- Technical issues – Information storage may not be possible. Information may be lost when the bot is closed.
- Dependencies with other requirements – Reliant on language processing to send the correct intents that call for the use of stored information.

Response Format

- Description – The responses to the user need to be concise, unambiguous and present all important information.
- Criticality - Not critical to the bot but greatly improves usability.
- Technical issues – The best response to the user's question may not have a concise answer. High variants in response types may disrupt response quality.
- Dependencies with other requirements - Reliant on language processing to send the correct intents that will produce a suitable text response.

Response Times

- Description – The bot needs to be quick in returning data to the user.
- Criticality - Not critical to the bot but greatly improves usability.
- Technical issues – Highly reliant on internet quality on the phone.
- Dependencies with other requirements – Dependent on data retrieval. The type of data being retrieved may affect speed.

4. System Architecture

Facebook Messenger

Acts as the front-end user interface of the chatbot. User pass text through messenger to the bot and the bot returns data to the user in text format.

Bot Framework

This is used as the connector of the major components of the application and is used to transfer data between Facebook Messenger, LUIS, and the bot.

DCUAPChatbot

The main part of the bot. This contains the code which includes the functions that are called. These functions will then retrieve the relevant data from the

external resources. This along with the bot framework are the back-end of the app.

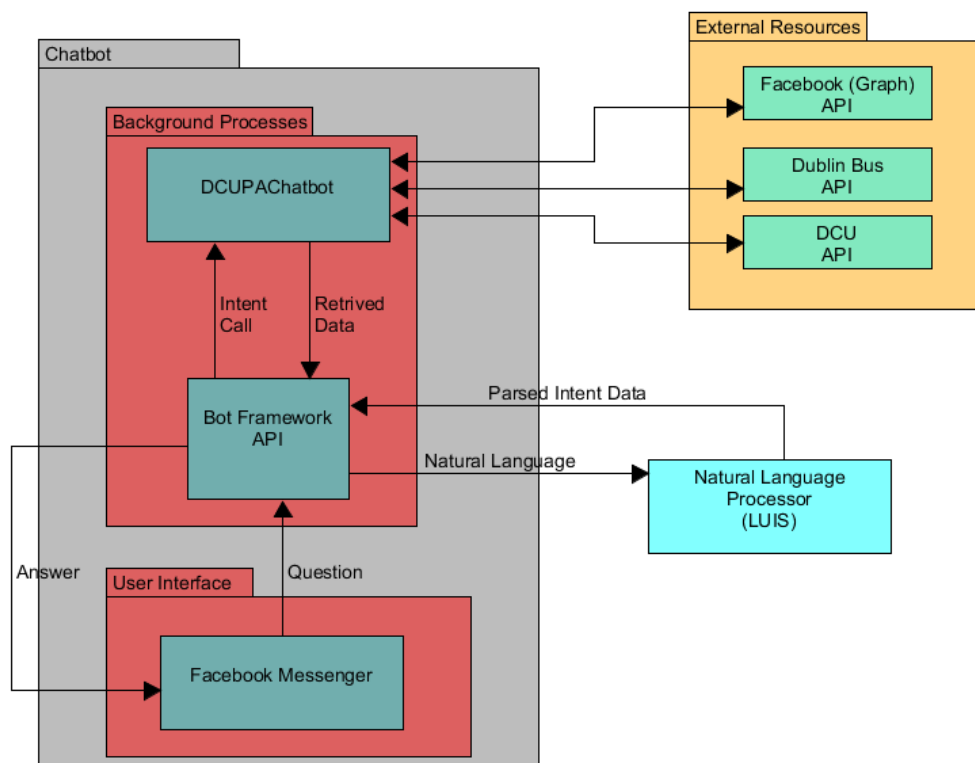
LUIS

This takes the text sent to it from the app and processes it into intents that the bot can use to call the correct function in the code.

External Resources

These are the outside APIs that will be used to get the right information for the bot. For example, the Dublin Bus API will be used to get bus times from the Dublin Bus website.

System Architecture Diagram:



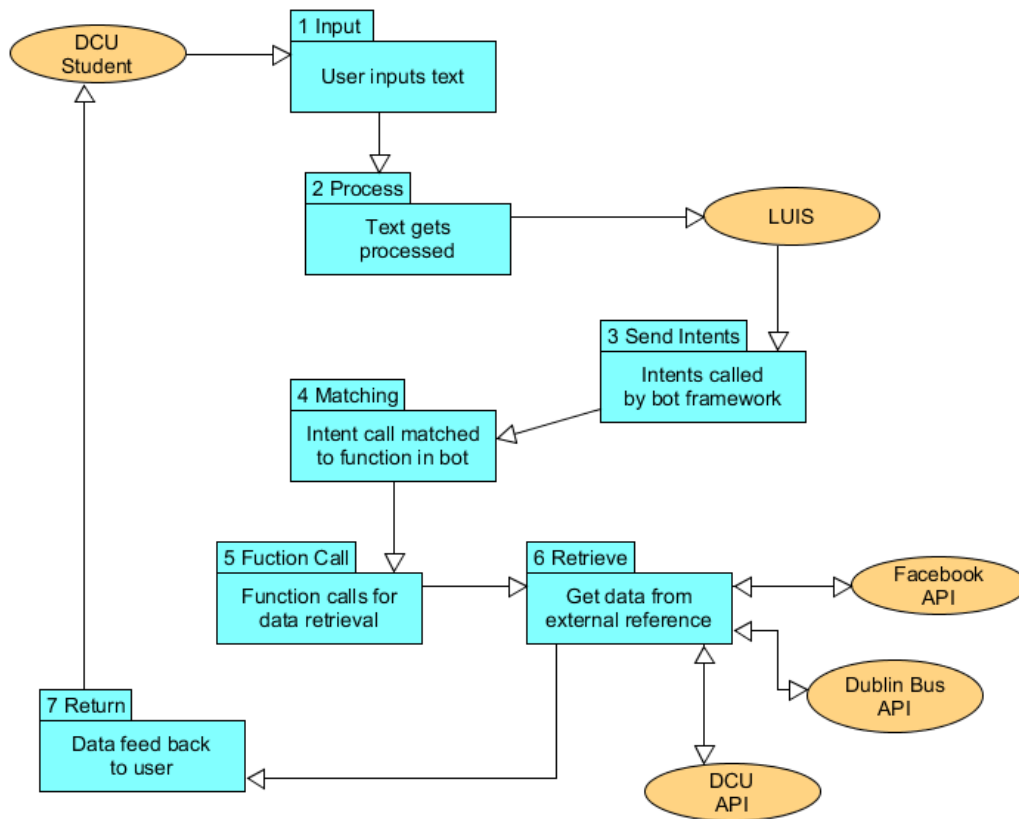
5. High-Level Design

This data flow diagram shows the steps that are taken by the chat bot to take what the user is asking for, processing it, and getting the correct data to successfully answer the question posed.

The text gets input from Facebook Messenger. It is then sent to LUIS for processing which will find the entities in the text and extract intents. These intents are then sent to the bot and are matched with a suitable function in the bot. This function will then fetch the correct data from an external reference.

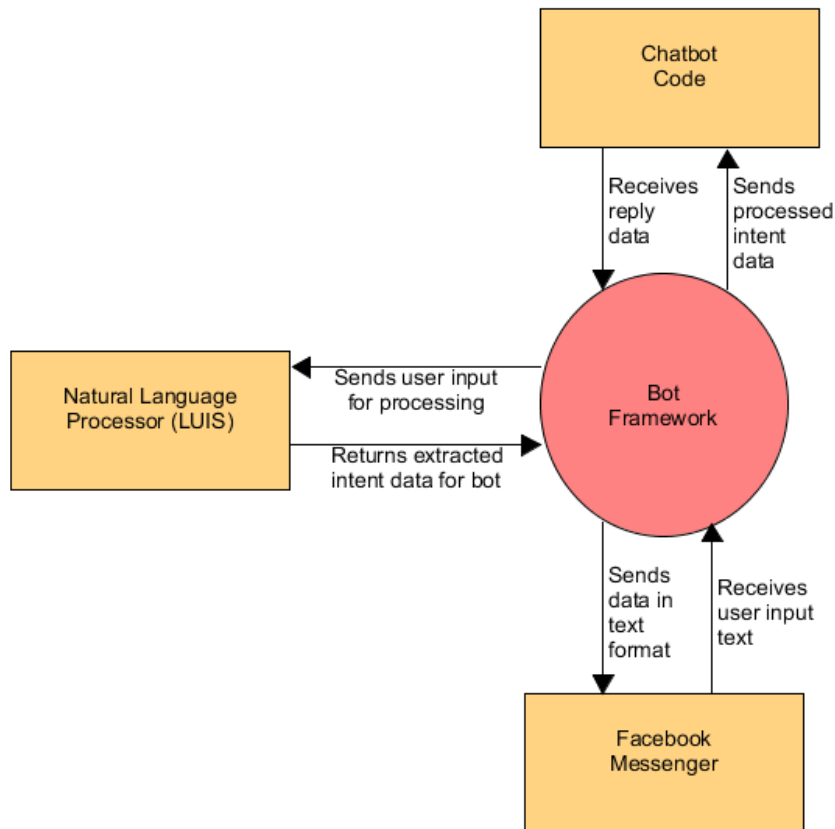
This data is then passed all the way back to Messenger in text format for the user and answering the initial question.

Data Flow Diagram:



Next is a context diagram for the bot framework that highlights in more detail how it takes the input with Facebook Messenger, processes it with LUIS, and then passes it along to the bot. After it will the pass back to messenger the data retrieved by the bot. The bot diagram was chosen as it acts as the connector for the majority of the bot and as the main transporter of data.

Context Diagram of the Bot framework:

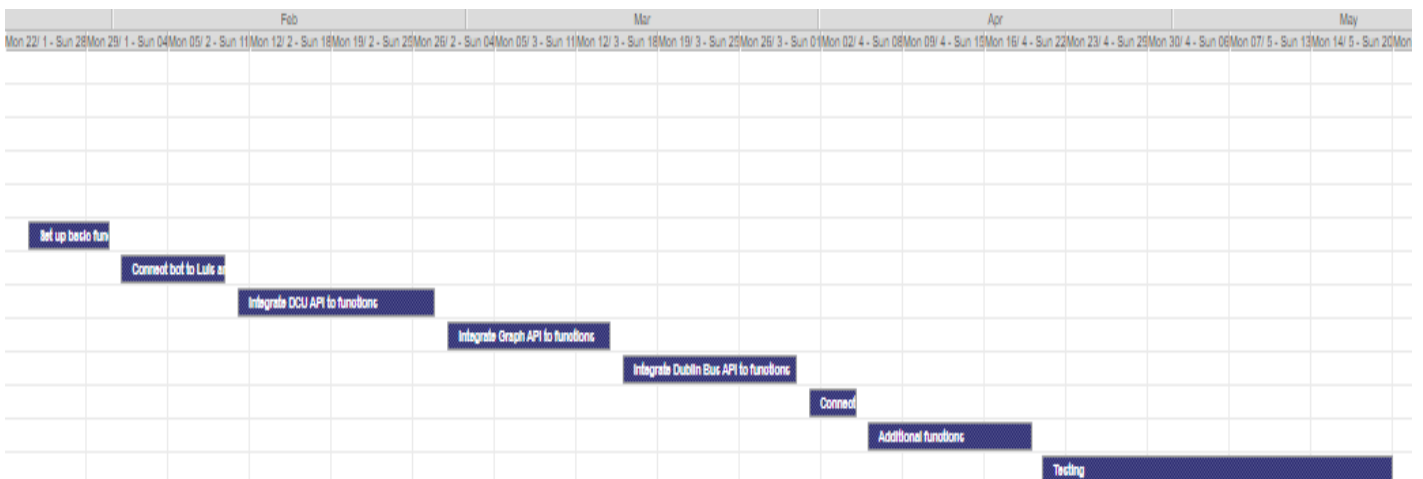
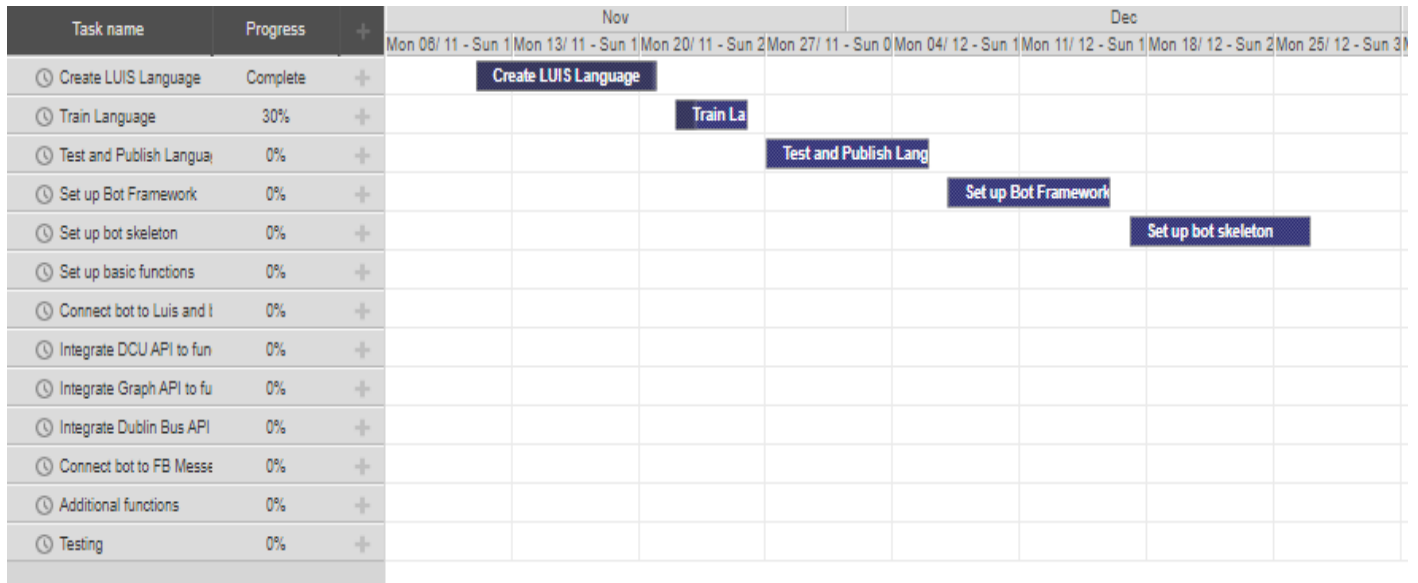


6. Preliminary Schedule

11/11/2017 – 21/11/2017: Create LUIS Language
22/11/2017 – 26/11/2017: Train Language
17/11/2017 – 06/12/2017: Test and Publish Language
07/12/2017 – 16/12/2017: Set up Bot Framework
17/12/2017 – 27/12/2017: Create bot skeleton
<u>Break for examination period</u>
24/01/2018 – 31/01/2018: Set up basic functions
01/02/2018 – 10/02/2018: Connect bot to Luis and bot framework
11/02/2018 – 28/02/2018: Integrate DCU API to functions
01/03/2018 – 15/03/2018: Integrate Graph API to functions
16/03/2018 – 31/03/2018: Integrate Dublin Bus API to functions
01/04/2018 – 05/04/2018: Connect bot to FB Messenger

06/04/2018 – 20/04/2018: Create additional functions if available

21/04/2018 – 21/05/2018: Testing



7. Appendices

- [Overview of Microsoft's Bot Framework](https://docs.microsoft.com/en-us/bot-framework/overview-how-bot-framework-works) (https://docs.microsoft.com/en-us/bot-framework/overview-how-bot-framework-works)
- [Overview of LUIS](https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home) (https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home)
- [Overview of Facebook's Graph API](https://developers.facebook.com/docs/graph-api/overview/) (https://developers.facebook.com/docs/graph-api/overview/)