

National College of Ireland  
BSc in Computing  
2017/2018

Ciaran Staunton  
X14496288

## NowBot - Artificially Intelligent Chatbot

Technical Report



## Contents

1	Executive Summary .....	4
1.1	Background.....	5
1.2	Aims.....	5
1.3	Technologies .....	7
1.4	System.....	9
1.4	Requirements .....	9
1.4.1	Requirement 1 < Conversation>.....	9
1.4.2	Requirement 2 <Take Requests>.....	12
1.4.3	Requirement 3 <work with LUIS>.....	14
1.4.4	Requirement 4 < Show Packages>.....	15
1.4.5	Requirement 5 <work on messenger > .....	18
1.4.6	Requirement 6 <Contact designer>.....	19
1.4.7	User requirements.....	23
1.4.8	Environmental requirements .....	23
1.4.9	Usability requirements.....	24
1.5	Non-Functional Requirements .....	24
1.5.1	Performance/Response time requirement.....	24
1.5.2	Availability requirement .....	24
1.5.3	Security requirement .....	24
1.5.4	Reliability requirement.....	24
1.5.5	Maintainability requirement .....	25
1.5.6	Extendibility requirement.....	25
1.5.7	Reusability requirement.....	25
2	System Evolution .....	25
2.1	Design and Architecture.....	25
2.2	Implementation .....	27
2.3	Graphical User Interface (GUI) Layout.....	45
2.4	Testing.....	51
2.5	Customer testing.....	56
2.6	Evaluation .....	<b>Error! Bookmark not defined.</b>
3	Conclusions .....	60

4	Further development or research.....	61
5	References .....	62
6	Appendix.....	63
6.1	Project Proposal .....	63
	TITLE .....	63
	Web Design Chatbot.....	63
	Nowbot.....	63
	Objectives .....	64
	Background.....	64
7	Technical Approach .....	65
8	Special resources required .....	65
9	Project Plan .....	65
10	Technical Details.....	66
11	Evaluation .....	67
11.1	Project Plan .....	67
11.2	Monthly Journals.....	67
11.3	Other Material Used.....	75

# 1 Executive Summary

The problem I am aiming to address with my 4<sup>th</sup> year project is information overload that business owners and individuals experience when trying to find out how to get online and the process involved with it. Also, time wasting that web designers experience daily from clients who are interested in websites and all ask the same questions, how do you go about getting online? I know this as I run a digital agency providing clients with websites so have experienced this first hand. The solution which will solve this is a Chatbot Application. My chatbot application will be called the "NowBot". I picked this name because it is current technology providing current information for the now so I feel this name reflects the bots function well. The NowBot will provide instant responses to users helping them get there business online, checking domain availability and showing website, logo and SEO packages. The chat bot will show users possible purchases which they can make , and assist them with any queries they may have. I am building this chatbot using Microsoft Bot Frameworks, this is a modern platform for building chatbots, and really the industry standard. I have never used this framework before or had any experience building chatbots so I realise it will be very challenging but I feel I can overcome this as I am determined to do a good project. The functionality of instant messaging to the user is how I aim to try tackle the problem of the information overload for clients. It will be an interactive information experience for the user who can use it as a one stop shop to finding solutions for their website development. With this considered I feel the commercial potential of this chatbot is excellent as is being built based on the customer's needs. I want this chatbot to act as customer service agent, answering all the queries which a potential customer may have. I also want to impress them with the chatbot and try turn the users into customers.

## **1.1 Background**

I wanted to do something for my final year project that would be up to date and modern for today's world and something I can use in the real-world post college. A chatbot was the most appealing and exciting idea to me. I feel when the project is complete it will have massive commercial potential as it will save time and money for web design companies, and act as a customer service agent. It can also be reprogrammed to suit individual needs, re training the AI to match the company's goals. With this in mind I felt that a chatbot would be the best option for my project, artificial intelligence is the future of technology and already plays a massive role in business. Many businesses use chatbots for customer service, payment gateways etc. AI is a massive part to the future and chatbots are an exciting new technology that many people are still yet to have come into contact with, it is still a fresh, powerful and fun technology for the user. I wanted to do a project relating to chatbots and AI to learn more about both technologies and to gain some experience in addition to the module. I also find this area of technology and web design/ marketing very interesting which is another reason I wanted to do AI chatbot for my final year project.

## **1.2 Aims**

The aim is to provide an engaging experience for the user and take the hassle out of web design. I also aim to provide a fun and engaging experience for the user, conversing with a chatbot rather than scrolling through the typical website or web app. I'm aiming for this chatbot to have complete commercial potential meaning other web design companies can take it for their own use at a cost, but I will use this chatbot on my own web design website once it has been fully complete. I want to use it to save myself time with web design consultations, and also stand out as a web design company with innovative modern technology. I aim to direct users to this application through advertising so that the chatbot can take away the initial

information session that is needed and it can be a constant reliable service for clients. I am aiming for this to be an instant messaging conversation with the user being the one in full control. Rather than searching google and getting results back that can confuse them or not point them in the direction they wish to go or give them the info they want, the chatbot will only return information what the user has requested. The aim of the chat bot is to provide the users with some interesting information and designs for their businesses or brands, helping them brand and build their online presence. I feel there is a need for this kind of application as many business owners find there it very confusing going about the whole web design process and can find it very difficult to know where to start with getting online. The conversation must constantly be in full flow and the user must come away from the experience with a positive attitude and no frustration of not being able to communicate correctly. A website is a major factor in a business success, so when the user has finished with Nowbot, I want the user to be left in a comfortable confident position moving forward with their website. I also want the user to be so impressed they tell a friend or a co worker about the experience. Not many businesses have chatbot applications in production and working for them so I want to be up to date with technology of the artificially intelligent chatbots. The chatbot will take away the consultation time needed with web designers, readying the user for the development of the website, leaving them having a much greater understanding of what exactly needs to be done whilst enjoying the conversation and being impressed by the service. I also want the user to learn more about the services offered than just websites, I am to do this throughout the user's interaction with the bot. These factors are the overall aim of my project.

### **1.3 Technologies**

There are many technologies available out there to make chat bots. Various apps and platforms take a lot of the heavy lifting out of developing the chat bot and give you out of the box environments and require no coding, while these are great for building simple chatbots I plan on building a more complex bot therefore I am using Microsoft Bot Framework. It is the most powerful way to build chat bots currently, it can allow me to achieve the functionality I require for my project such as calling API's and incorporating active learning, allowing the bot to get smarter the more it interacts with users. This is why I will be using Microsoft's newest framework, the bot framework , it allows a lot more flexibility when it comes to developing chatbots unlike many other platforms. The bot will be developed using node.js and the environment I will use will be Microsoft Visual Studio Code. I will be using a program called LUIS which is a language understanding intelligence service. Using LUIS will allow me to program the bot to understand to the best of its ability what the user is inputting. LUIS stores data in a database which I will manual enter. When a user inputs text, LUIS will take this info and check the database and try match it to the most suitable solution it feels it is, it also scores every result based on how well it understood what the user sent. The scoring of the language is sentiment analysis. Another technology I will be using is QnA maker. This is a free REST API that I can use to train the bot to respond and answer questions in a more human like conversational way. It is an extremely useful tool and will allow me to achieve what I need to do with my bot. I will be using Microsoft AZURE for deployment of my bot. I will deploy the bot on azure as a web app, this will act as a server endpoint for me to call from other channels. I will also be deploying the bot on Facebook messenger which will allow for deployment across the social media platform Facebook. Using Facebook will also allow me to use Facebook API and letting the bot have its very own Facebook page. This is where I plan on most user interaction on fakebook through messenger, as it is most convenient to the majority of

users. Using Microsoft's Emulator, I will be able to test the bot locally which is an extremely convenient feature which Microsoft provide. Before deployment all of the bot development will be done here in the emulator. The main language will be node.js for my chatbot again because node will allow me to achieve my objectives. Microsoft bot framework works well with node.js and it allows me to utilise many of the frameworks built in features and packages, which are node.js friendly. These are the most effective ways to make a bot currently. They will contribute to my project by allowing me to develop a conversational chatbot, letting me direct the user to external URL's and program the both the conversations inputs and outputs.



# System

## 1.4 Requirements

Gathering requirements, I done a lot of research online and found that 85% of users who have mobile phones use messaging apps. Mobile phones are easily accessible now and readily available for a low cost to any consumer. This makes the use of chatbots incredibly high as they can be used by anybody anytime on mobile. This knowledge allowed me to come up with my requirements for the bot.

- The user must be able to initiate conversation
- The chatbot must respond to users fast
- Chatbot must be able to take requests
- The chatbot must have LUIS
- The chatbot must show packages
- The chatbot must have QnA
- The chatbot must be on a website
- The chatbot must be accessible through messenger

### 1.4.1 Requirement 1 < Conversation >

Chatbot will start a conversation by welcoming the user to the conversation

This is the most key functional requirement as the chatbot needs to be an instant message service for the user, and needs to welcome them when they join the session.

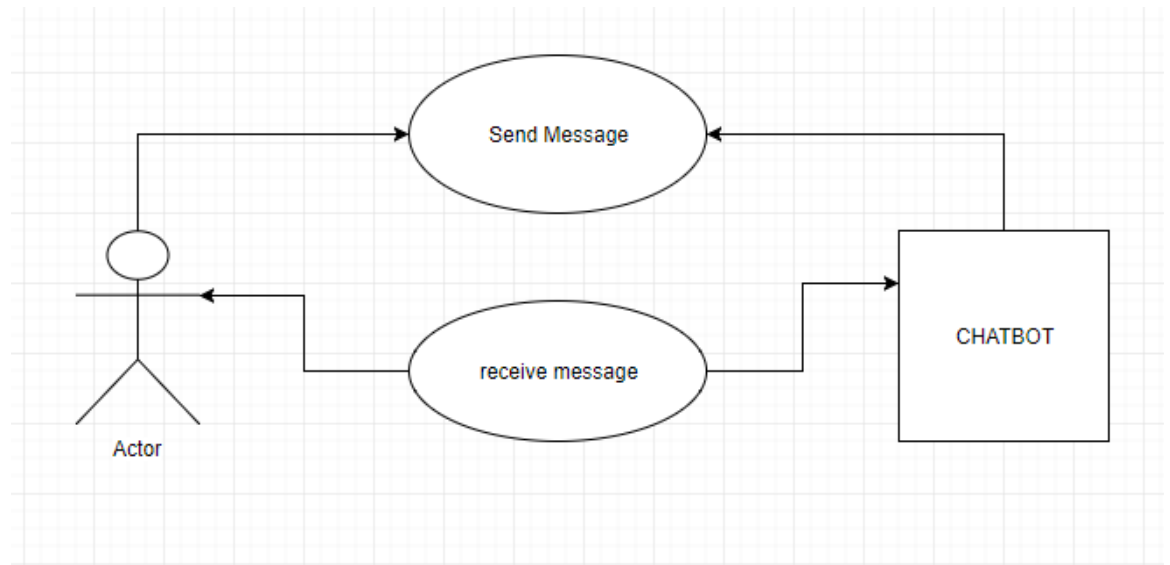
#### **Scope**

The scope of this use case is to make the chatbot initiate the conversation

#### **Description**

This use case shows the start of a conversation and how responses will take place. Once user joins they receive a message, from here it will be back and fourth conversation.

### Use Case Diagram



### Flow Description

The flow description of this diagram is that both the user and chatbot can initiate the conversation and both the user and the chatbot can receive messages. This is the basis of the chat, sending and receiving messages.

### Precondition

The user must have WIFI access. The user must type in the activation word to start a conversation

### Activation

This process will start when user enters a message for example “Hello”

### Main flow

1. The bot starts a conversation
2. The user responds accordingly
3. The bot starts a new conversation by asking question

4. The user answers the question

### Alternate flow

A1 : <Inactive user>

1. The system starts conversation
2. The user does not respond
3. The use case continues at position 3 of the main flow

### Termination

The system presents the data the user requires

### Post condition

The system goes into a wait state

### Code example

```
//this is the welcoming message that the user will be greeted with when they
bot.on('conversationUpdate', function (message) {
  if (message.membersAdded) {
    message.membersAdded.forEach(function (identity) {
      if (identity.id === message.address.bot.id) {
        var reply = new builder.Message()
          .address(message.address)
          .text('Hi! Im NowBot, I will be your assitant today!\n\n')
        bot.send(reply);
      }
    });
  }
});
```

This code segment shows how the conversation will begin. The bot on function sends the user a welcome message to initiate the conversation and greet them when they join the session. This is a key part of the success of the conversation as it will set the tone for how the user feels and chooses to interact.

## 1.4.2 Requirement 2 <Take Requests>

The requirement is that the chatbot must be able to take requests from the user and respond to these requests. If the chatbot fails to take requests this will cause users to leave the session due to inactivity or frustration.

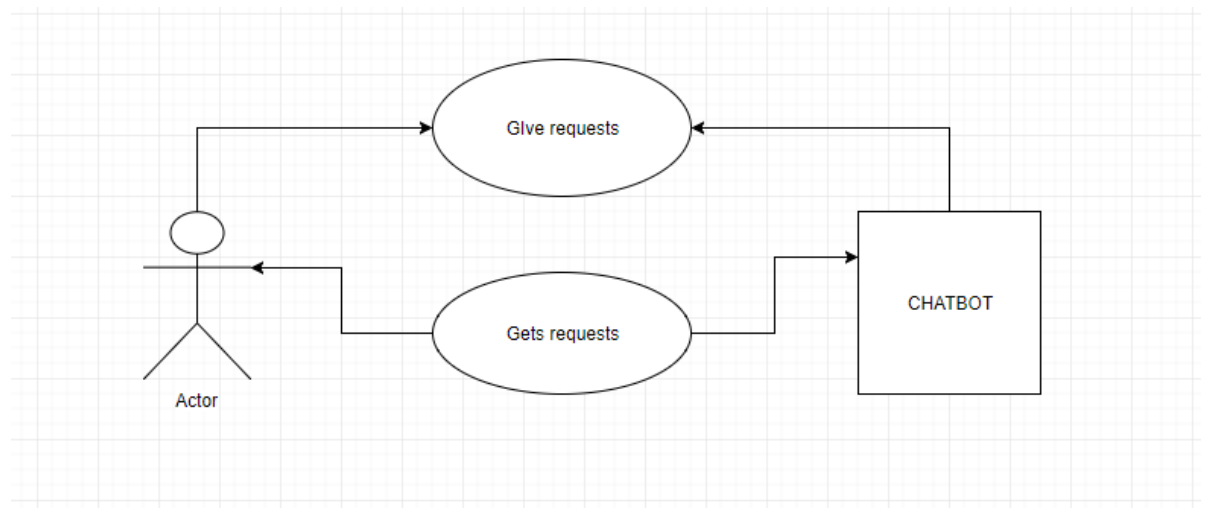
### Scope

The scope of this use case is to allow the chatbot to receive requests from the user.

### Description

This use case describes how the chatbot will take requests and respond to them accordingly.

### Use case diagram



### Flow Description

The flow description here shows how both the user and the chatbot can send and receive requests back and fourth.

### Precondition

The system is in waiting mode waiting to receive a request

### Activation

This use case starts when the user makes a request

### **Main flow**

5. The system identifies the user
6. The user asks a question/request
7. The system responds to the request
8. The user asks another request or terminates the session

### **Alternate flow**

A1 : <no requests>

4. The system waits for request
5. The user does not submit requests
6. The system terminates session

### **Termination**

The user does not respond and closes the session

### **Post condition**

The system goes into a wait state

### **Code Example**

```
bot.dialog('/', intents);
```

This is the root dialog of the bot. It listens to all inputs and chooses the response based on the input. For example, if a user enters "Logo", the root dialog is listening, and will trigger other appropriate dialogs in this case, it is the logo dialog within the bot. The root dialog is the beginning of all conversation.

```
//Logo functionality  
bot.dialog('logo', function (session) {
```

### 1.4.3 Requirement 3 <work with LUIS>

The requirement is that the chatbot must have LUIS incorporated. LUIS is the natural language understanding functionality of the bot.

#### **Scope**

The scope of this use case is to make the chatbot works with LUIS.

#### **Flow Description**

The flow description of this use case shows when a user inputs to the bot chat that the LUIS natural language understanding will understand what the user has inputted and respond effectively.

#### **Precondition**

The system is in waiting mode waiting to receive an input

#### **Activation**

This use case starts when the user inputs

#### **Main flow**

1. The user inputs text
2. The input is scored by LUIS
3. LUIS queries the natural language database
4. Responds to user

#### **Alternate flow**

A1 : <no input>

1. The user inputs text
2. LUIS does not recognize the input
3. Conversation state terminates

#### **Termination**

The user does not respond and closes the session

#### **Post condition**

The system goes into a wait state

## Code Example

In this code example you can see how LUIS is called. The appId and API key are inputted to allow for successful communication between the bot and the LUIS database. It makes requests over HTTPS, ensuring secure communication.

```
var luisAppId="ecfd8615-4745-49f8-84ba-aa814f9dac4c";
var luisAPIKey="7ee2792f7e6c467fb396b0024e3789fd";
var luisAPIHostName="westus.api.cognitive.microsoft.com";

const luisModelUrl = 'https://' + luisAPIHostName + '/luis/v2.0/apps/' +

var recognizer = new builder.LuisRecognizer(luisModelUrl);
var intents = new builder.IntentDialog({

recognizers: [recognizer]

});
```

### 1.4.4 Requirement 4 < Show Packages>

This is a key functional requirement and must happen to allow users to see packages which they may like such as SEO packages, or Logo packages.

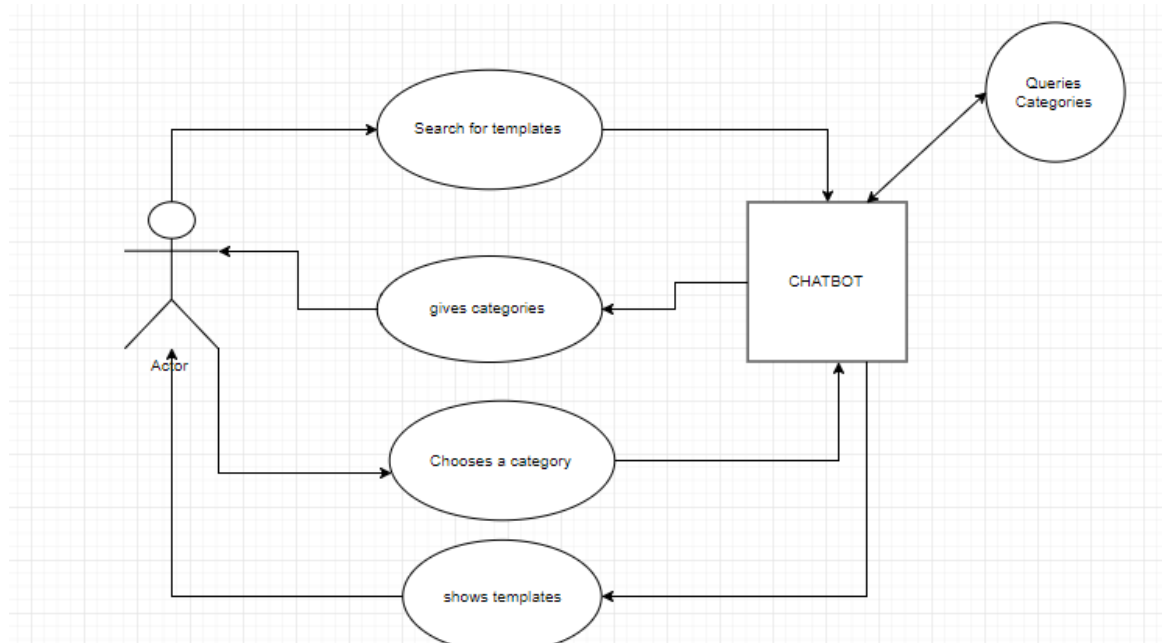
#### Scope

The scope of this use case is to display packages to the users

#### Description

This use case shows how packages will be displayed

#### Use case diagram



### Flow Description

The flow description here shows when the user wants a package they must first choose a category which they feel is most representative of themselves such as blogger, Start up, business or entrepreneur. When a service is requested the chatbot will respond with a list of categories'. Once the user chooses one the chatbot receives this and will allow the user to continue to next stage.

### Precondition

The user must select a category to see packages

### Activation

This process will start when user selects a category eg websites

### Main flow

1. The user requests service
2. The system displays packages
3. The user selects a category
4. The system displays check out options

### Alternate flow



A1 : <Inactive user>

1. The system displays categories
2. The user does not see category of interest
3. The use terminates the session

## Termination

The user ends session

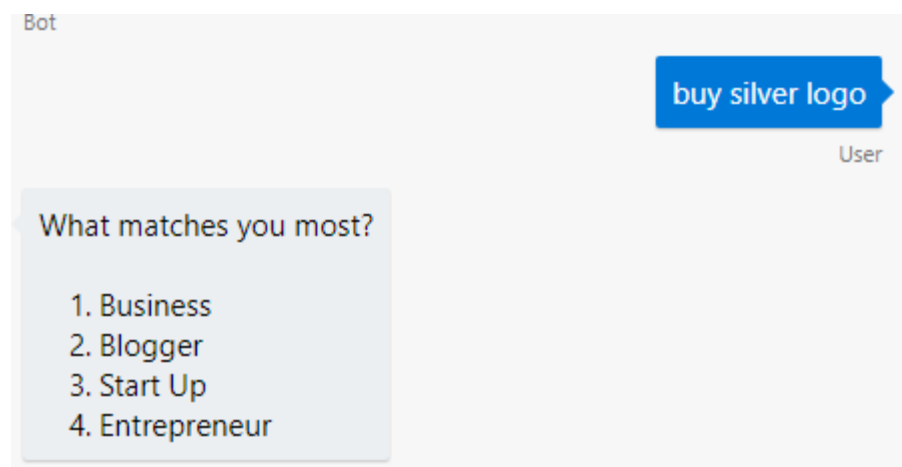
## Post condition

The system asks if the user is happy with package

## Screenshot Example

When a user asks to buy a logo, the bot will prompt them with categories. The user must choose one to get to the next stage of the conversation. This is necessary to gather more details about what exactly the client is looking for and how it can be customized just for them.

```
ce = args.intent.matched[0];  
/(one page|Wordpress|E-commerce|Custom)/i.exec(utterance);  
/\b(Business|Blogger|Entrepreneur|Start up)\b/i.exec(utterance);  
[
```



### **1.4.5 Requirement 5 <work on messenger >**

The requirement is that the chatbot must be available to users on Facebook's messenger platform

#### **Scope**

The scope of this use case is to make the chatbot work on messenger

#### **Description**

The most crucial place for the chatbot to work is on messenger as this is where the majority of user will be and will choose to interact with the bot. It makes it a lot more convenient for the user not to have to download a 3<sup>rd</sup> party app or exit their Facebook/messenger session just to communicate with the bot. For the bot to work on messenger it requires Facebook page ID which means the bot must be installed as an app on a Facebook page in order to work. It also requires App Id of the bot, the App secret key which Microsoft provide when the bot is registered.

#### **Precondition**

The system is in messenger in a wait state until conversation begins with a user

#### **Activation**

This use case starts when the user enters into a session with the bot

#### **Main flow**

1. The user is on facebook
2. The user texts the bot on messenger
3. The system responds
4. The user responds

#### **Alternate flow**

A1 : <does not respond to welcome message>

1. The system greets user
2. The user does not respond to the greeting
3. The user terminates session

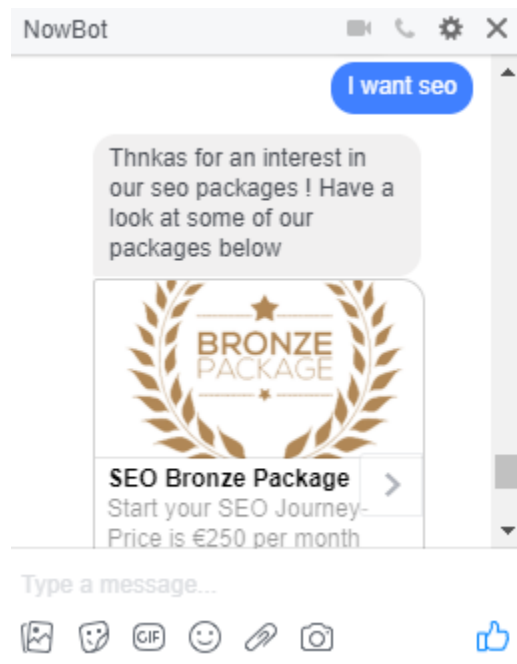
### Termination

The user does not converse with the bot

### Post condition

The system goes into a wait state

### Screenshot



This is the bot functioning within messenger, displaying SEO packages as requested.

### 1.4.6 Requirement 6 <Contact designer>

The requirement is that the chatbot must be able to forward on the users details to the web designer for further development.

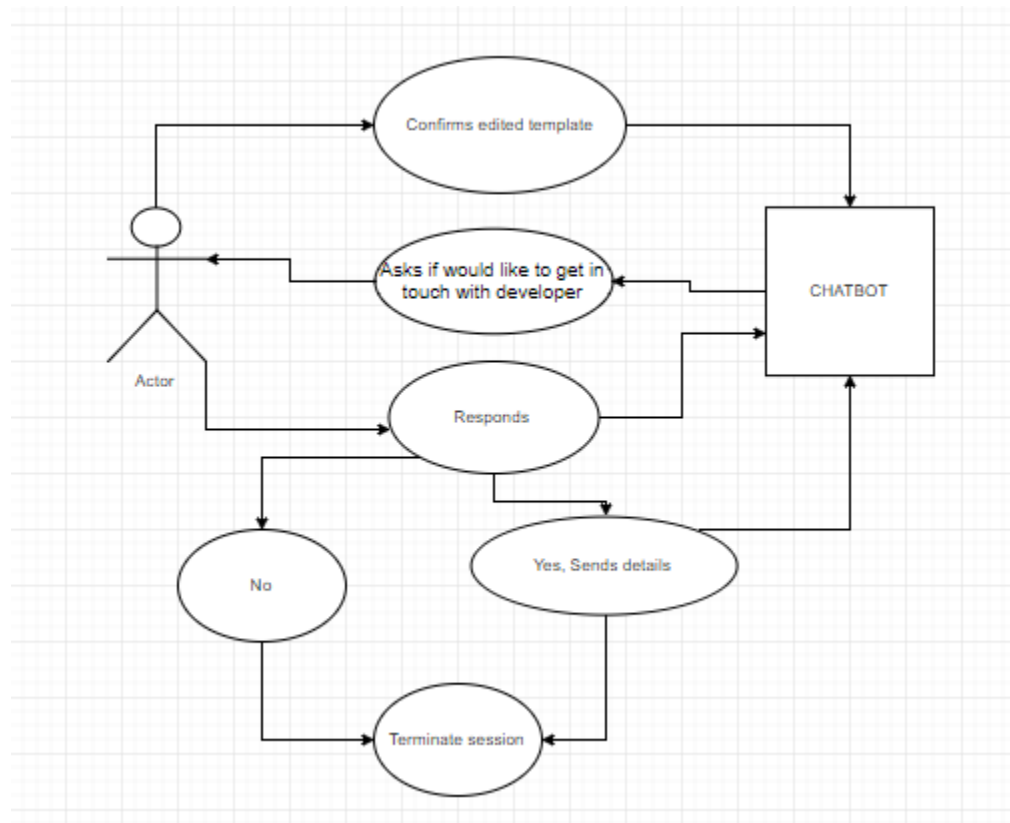
### Scope

The scope of this use case is to allow the chatbot to send users details of a web designer

### Description

This use case describes how the chatbot will send details

### Use case diagram



### Flow Description

this use case shows how the user and bot communicate to send details to a designer. The user will confirm the edited template from here the bot will ask the user if they want to get in touch with a web designer. If the user responds with yes, their details are sent and the session terminates. If the user responds no, the session terminates.

## Main flow

1. The system asks user to confirm
2. The user confirms
3. The system sends details to designer
4. The user receives contact from web designer

## Alternate flow

A1 : <no confirmation>

1. The system waits for confirmation
2. The user does not confirm
3. The system terminates session

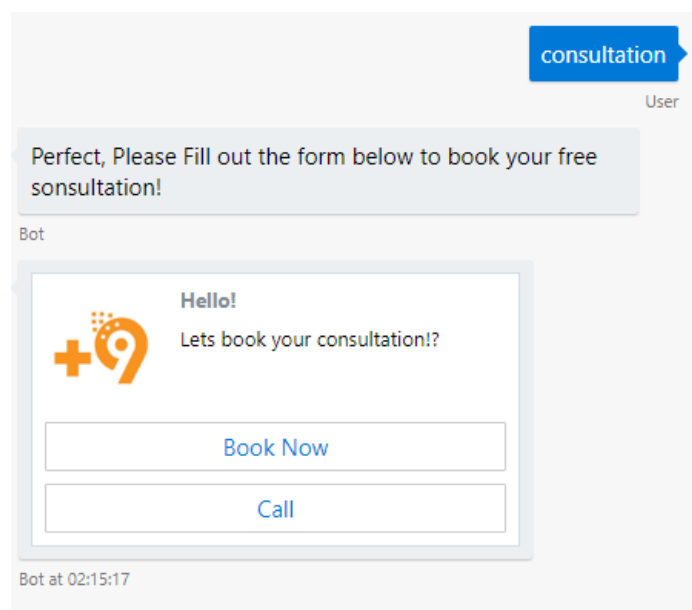
## Termination

The user does not respond and closes the session

## Post condition

The system goes into a wait state

## Screenshots





Hello!

Lets book your consultation!?

Book Now

Call

### Welcome to our consultation booker!

Please enter your Location:

Dublin, Ireland

Choose a date

dd/mm/yyyy

Submit

The contact designer options that the user is prompted with when they request a consultation or to contact designer. When they fill out the form I am notified that a session dialog has ended.

### Requirement 7 <work with QnA>

The chatbot must work with QnA to ensure smooth conversational flow within the bot. I have to make a QnA Microsoft cognitive services account to be able to use this service. Once signed up I get an appId and a knowledge base ID. Entering these keys will allow for successful communication between the bot and QnA API.

```
//QnA Services

var recognizer = new builder_cognitiveservices.QnAMakerRecognizer({
  knowledgeBaseId: '38a927e6-693e-430c-9690-ae3e22594788', // process.e
  subscriptionKey: ' 4e45e350-f5fe-417e-add7-44c083a1722b'}); //process

var basicQnAMakerDialog = new builder_cognitiveservices.QnAMakerDialog({
  recognizers: [recognizer],
  defaultMessage: 'No match! Try changing the query terms!',
  qnaThreshold: 0.3}
);
```

#### 1.4.7 User requirements

The chatbot must respond instantly to hold the customers attention. It must also be visually appealing to get the user to use the bot. Looking at it from the customers point of view the bot must be fully functional and contain no errors or time problems, otherwise the user will exit the session with the chatbot. It must respond fast whilst providing valuable information to the user that will make them stay in active conversation with the bot. The data will need to be both secure so the user trusts the bot to submit information. The data will also need to be easily accessible and not require the user to leave the session to view any data. The bot also must be super mobile friendly as studies show most users are on mobile these days. The chatbot must also available on social platforms such as Facebook as this is where most traffic will come from.

#### 1.4.8 Environmental requirements

The chatbot will be available anywhere in the world once the user has WIFI, this is to ensure the user can gain access and access the chatbot if they require to. The chatbot also must be available on at least one platform so that the user can initiate a conversation if they want. Having the chatbot on Facebook and the website will allow users worldwide to communicate with the bot at any time with constant

instant replies. Due to the bot being deployed on Facebook it will be mobile friendly allowing users to use the bot on mobile devices.

#### **1.4.9 Usability requirements**

The chatbot must look great visually to hold the user's attention, it also must be easy to use and highly user friendly for people who may not be tech literate. It should be a fun and interactive experience for the user so that when the session ends they feel the need to tell somebody and share the experience with friends and peers.

### **1.5 Non-Functional Requirements**

#### **1.5.1 Performance/Response time requirement**

The chatbot must respond instantly to users when it receives a message. This is key to the chatbots success. It must provide instant replies.

#### **1.5.2 Availability requirement**

The chatbot must be available always to suit the needs of the user. I don't want my bot to only be accessible at certain times. Once my bot is live and deployed it will there for the user to use whenever they require it.

#### **1.5.3 Security requirement**

I will need my chatbot to be secure so the user trusts the bot to share information over the chat. Users generally share sensitive data over messaging apps, so this chatbot will be required to be secure. I will have full access to the conversations had over the chat bot. I will add various security features to the code. The code will be stored securely by Azure. People also react emotionally to bots, so I need to take into account what I train the AI to do to ensure I do not make them sad, angry or make them feel as if I am trying to breach privacy with questions.

#### **1.5.4 Reliability requirement**

The bot must always respond and never end the dialog when the user does is not finished. It must stay consistent with the responses and dialogs it is sending to the



user. If it is not consistent the user will get annoyed and leave the session so this is a very important requirement.

### **1.5.5 Maintainability requirement**

The chatbot will not need to much maintenance as all the 3<sup>rd</sup> parties I am using have scheduled automatic updates.

### **1.5.6 Extendibility requirement**

Nowbot is fully extendable and the AI can continuously be improved and trained better. This will happen over time, hence the bot will get better over time.

### **1.5.7 Reusability requirement**

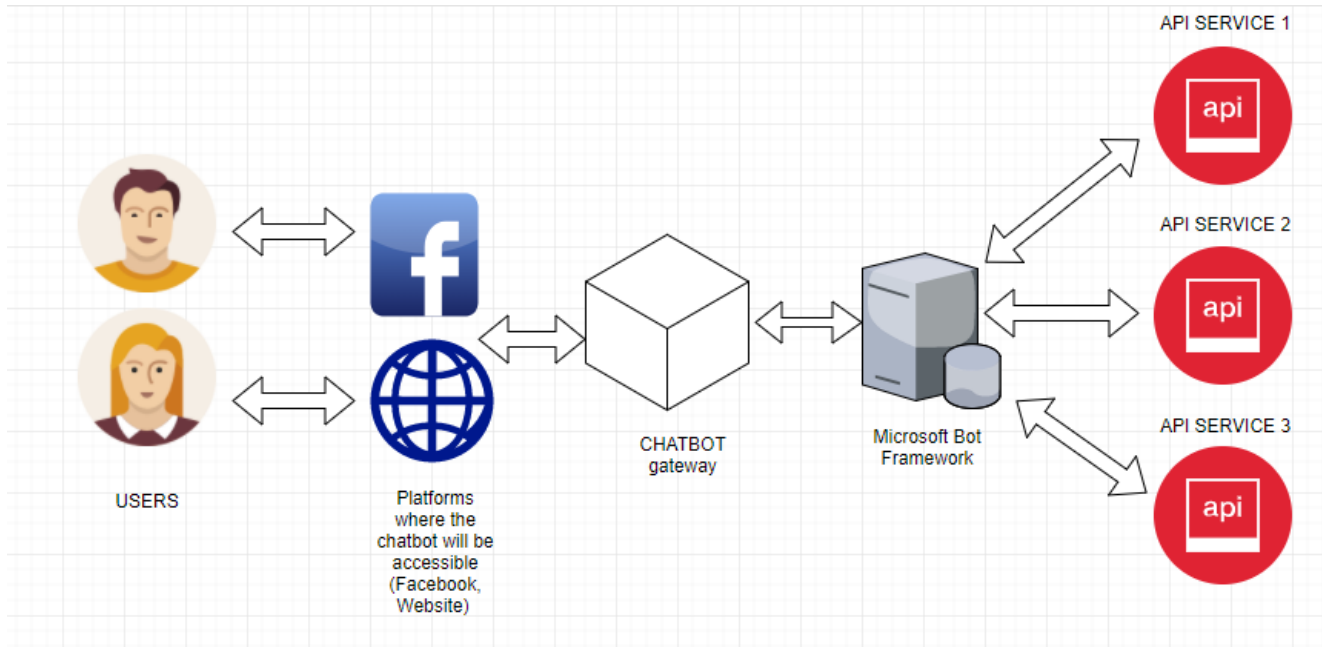
The bot will be reusable to the user as many times as they need it to be.

## **2 System Evolution**

Over time this chatbot will continue to improve and improve. The AI can constantly be trained to a higher standard allowing it to have more extensive conversations with the user. I think this is a great aspect of the chat bot, keeping it relevant and modern for the ultimate user experience.

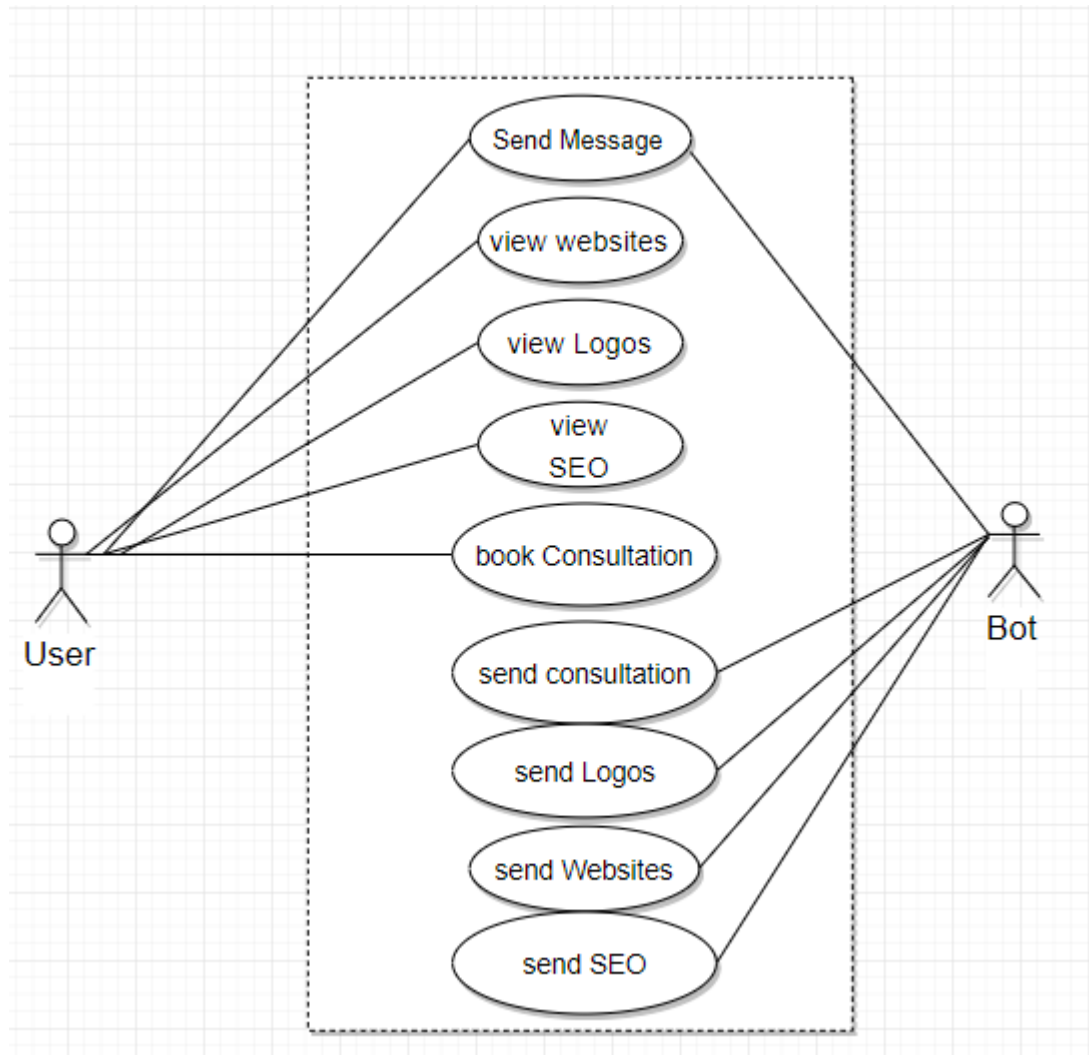
### ***2.1 Design and Architecture***

One of the main coding techniques I will use in my project is Waterfalls. This is how the conversation will flow in node.js., it will allow me to program the bot's messages to the user and also the responses to the user based off their previous message. The name waterfall comes from the idea of flowing and moving from one question to the next. It controls the flow of the conversation and allows me to prompt the user with new options, end or begin conversation. All of the waterfall dialogs and other bot methods will be inside the "app.js" file. This file is the main component to the bot and is where all the dialogs will be coded.



## Use Case Diagram

The use case diagram displays how the bot and the user will interact with each other. It shows the options available to the user and also how the bot will respond if the user chooses these options.



## 2.2 Implementation

The toughest part of this project was by far the implementation. The reason why it was so tough was because I was using Microsoft bot frameworks, which is a framework I've never worked with before, it is also fairly new so there are not many resources available online to help with errors I was getting in the code. This made

it extremely difficult to learn and gain knowledge into how to go about building a chatbot within the framework. I had to research many different techniques and ways of implementation and try each one to see what would work best. It took me a few weeks alone roughly 30-40 hours before I had the chatbot running smoothly and connecting to my local test environment, which was a frustrating process as this was just the initial local testing set up. It was a lot of trial and error deleting and re editing code until I eventually discovered how to do this. This was the way the entire project, trial and error. Since the framework was so new and the language I used node.js which is also a fairly new language to me, and a difficult one it was a tough process to discover how to get the bot to run smoothly and functioning correctly. I used Microsoft's emulator desktop application to set up my local environment. This emulator allowed me to test my bot locally and remotely if I choose to however, to begin my project I just wanted to work locally. I was a great resource to me throughout the project and was a highly exiting time as it was a totally new experience for me and had not worked with environments like this before so seeing it work correctly was rewarding. I started by making the server so the emulator could connect to the local host. Restify is a node.js service framework which is specially built for working with web services and API's.

I created my app.js file and inputted all the relevant code to allow for the restify server to be created and for the bot to listen to the correct endpoint. When I want to begin testing this bot I must run the app.js file through my command prompt which will then if working correctly listen to the port 3798.

Once the app.js is listening I can then connect to the emulator and begin conversing with the bot.

```
// Setup Restify Server
var server = restify.createServer();

server.listen(process.env.port || process.env.PORT || 3978, function () {
  console.log('%s listening to %s', server.name, server.url);
});
```

You can see that the port number is 3798 and the console is now listening to any activity on that port. Within the emulator app I connected to the correct port and URL. I am a cyber security student so I also ensured that the communications are secure by adding in HTTPS which will add good security practice to my project.

```
var https = require('https');
```

```
var bot = new builder.UniversalBot(connector) ;
server.post('/api/messages', connector.listen());
```

In the emulator I enter in my Microsoft app id and password, to allow for successful communication between the bot server I created and the emulator. The URL `api/messages` then listens and receives any messages that are sent to that end and responds with the correct dialog. Once all the code is correctly in the `app.js` file I can run it through the command prompt.

```
C:\Users\hp\Desktop\Chatbot>node app.js
restify listening to http://[::]:3978
```

The screenshot shows the Bot Framework Emulator interface. At the top, the URL is `http://localhost:3978/api/messages`. Below this, there are input fields for 'Microsoft App ID' (containing `c8928f39-dac0-46e7-831e-0f7b845616ab`), 'Microsoft App Password' (masked with dots), and 'Locale' (empty). A blue 'CONNECT' button is positioned below these fields.

The main chat area displays a card titled 'Welcome to our Domain Checker'. The card features an image of a person in a black hat and trench coat holding a magnifying glass over a blue background. Below the image, the text reads: 'Welcome to our Domain Checker', 'Lets see if your domain is available!', and 'Having a good domain is crucial to the success of your website! if your domain is not already taken, make sure your secure yours now!'. A 'Lets Go' button is at the bottom of the card. The chat area also shows a 'User' name and a 'Bot at 16:19:42' timestamp.

On the right side, there is a 'Details' panel showing a JSON message object:

```
{
  "type": "message",
  "text": "blog",
  "from": {
    "id": "default-us",
    "name": "User"
  },
  "locale": "en-GB",
  "textFormat": "plain",
  "timestamp": "2018-12-15T18:09.562Z",
  "channelData": {
    "clientActivityId": "1526133084870.91288954"
  },
  "id": "retry"
}
```

Below the JSON is a 'Log' panel showing a series of network requests and responses with timestamps and status codes (e.g., `[16:24:30] <- POST 200`).

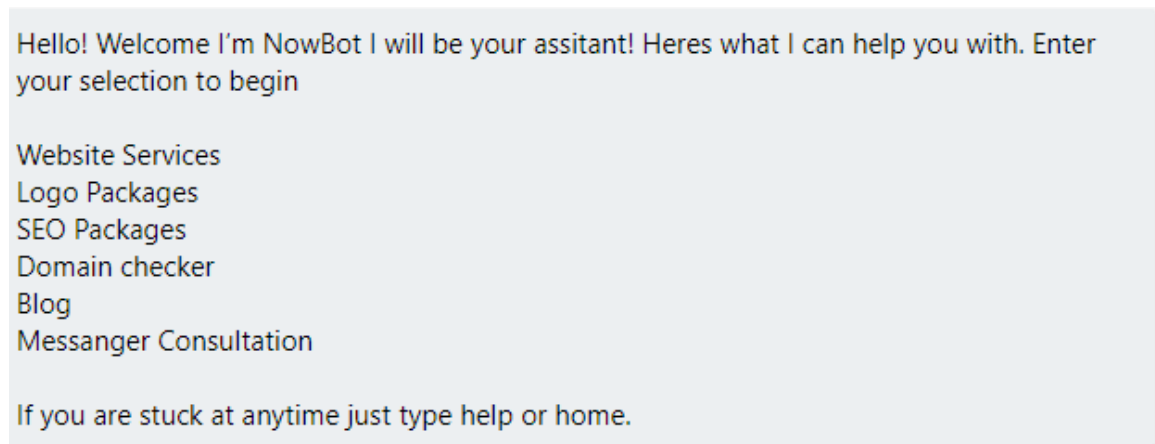
The emulator also has a very beneficial function of displaying JSON data of the input and outputs which allowed me to see how the back end of the bot was working for every message sent, this was extremely useful for debugging. It is an amazing process really and was a great skill to learn, when I finally got everything working it was highly rewarding.

After I got my test environment linked correctly, the fun part began which was to begin the waterfall dialogs, and message handlers and create the bot to the best of my ability. The first part of the implementation was to add a greeting message when the user joins the session. This greeting was what started the bot. with "Bot.on" as you can see in the screenshot below. The bot.on activates the bot and

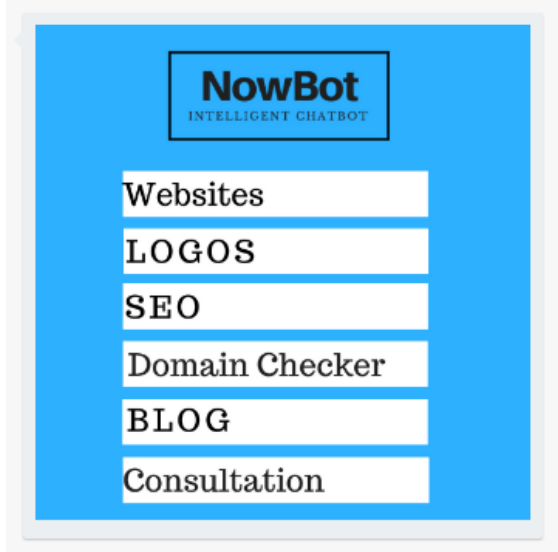
starts the dialog, attached to this bot.on dialog Is the greeting massager as you can see here “ Hi, I’m knowbot”

```
//this is the welcoming message that the user will be greeted with when they
bot.on('conversationUpdate', function (message) {
  if (message.membersAdded) {
    message.membersAdded.forEach(function (identity) {
      if (identity.id === message.address.bot.id) {
        var reply = new builder.Message()
          .address(message.address)
          .text('Hi! Im NowBot, I will be your assitant today!\n\n');
        bot.send(reply);
      }
    });
  }
});
```

I wanted the bot to greet the user first, as then I can take control of the flow of the conversation and direct the user to the functionality of the bot which will benefit them the most. This improves the overall experience for the user and does not leave them in the dark as to what exactly the bot will be able to help them with.



As you can see here I am showing the user straight from the beginning what exactly I can help with , this ensures that there is no confusion between the bot and the user. I also added an image in to make it clearer what options are available to the user. I like this image in the bot as I feel it is visually nice and beneficial to the user.



Another great benefit of greeting the user and providing instructional content is that the user will find it hard to get lost in dialog. Getting lost in dialog was a very hard problem to address but I used smart code implementation and smart dialog use to avoid this as much as possible. Using LUIS also allowed me to avoid getting the user lost as it recognizes the input from the user and prompts them with instructions.

The returning of the user to the main menu was how I address the getting lost problem. Attached to the greeting messages were instructions what to do if they get stuck

If you are stuck at anytime just type help or home.

Bot

The next part of implementing this was to make LUIS recognize home and help. I made a "intent" function for "GoHome" so when the user enters help the bot will recognize that they are stuck or lost and prompt them with solutions of how to get back to normal conversational flow.



```
intents.matches('GoHome', (session, args, next ) => {  
  session.send("Hey! I noticed your stuck! not to worry , ");  
});
```

Within LUIS I made a GoHome intent, within this intent I had utterances which I believed the user would enter when lost.

## GoHome

Utterance

---

help

---

i want to go home

---

As you can see above I trained the bot to understand help and I want to go home. Once a user enters these values into the chatbot it will then bring them to the gohome intent function above.

**help**  
User

Hey! I noticed your stuck! not to worry , just type any of these options to get out of here.

Websites

SEO

Logo

Book Consultation


Explore

Bot at 14:53:03

**logos**  
User

Thanks for an interest in our logos!

Bot




**Silver Logo Package**

A logo to make your brand instantly regognisable

Price is €75

[Buy](#)



**Gold Logo Package** ➤

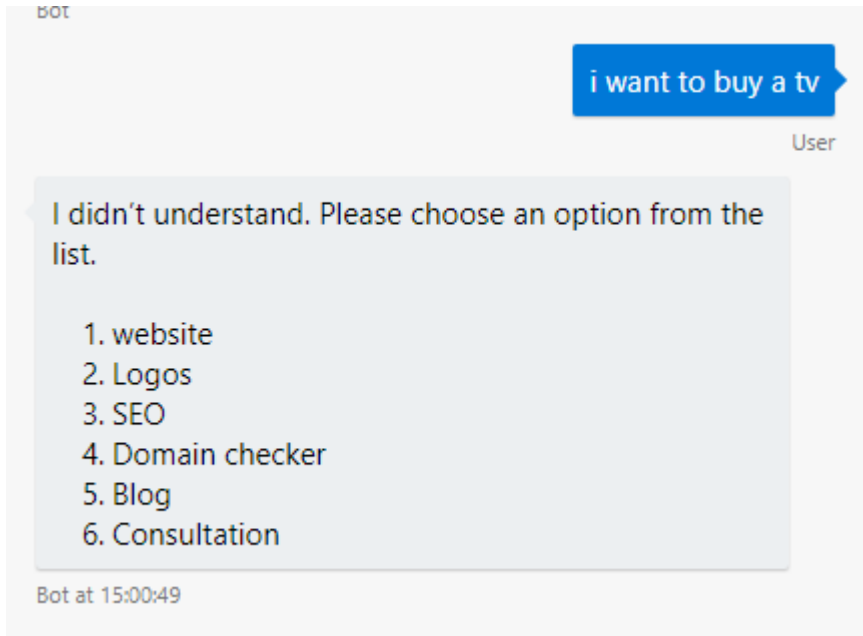
100% design from sketch.

Price is €199

[Buy](#)

This worked very effectively but was difficult to address. However since I successfully address it made the flow of the conversation a lot better and made

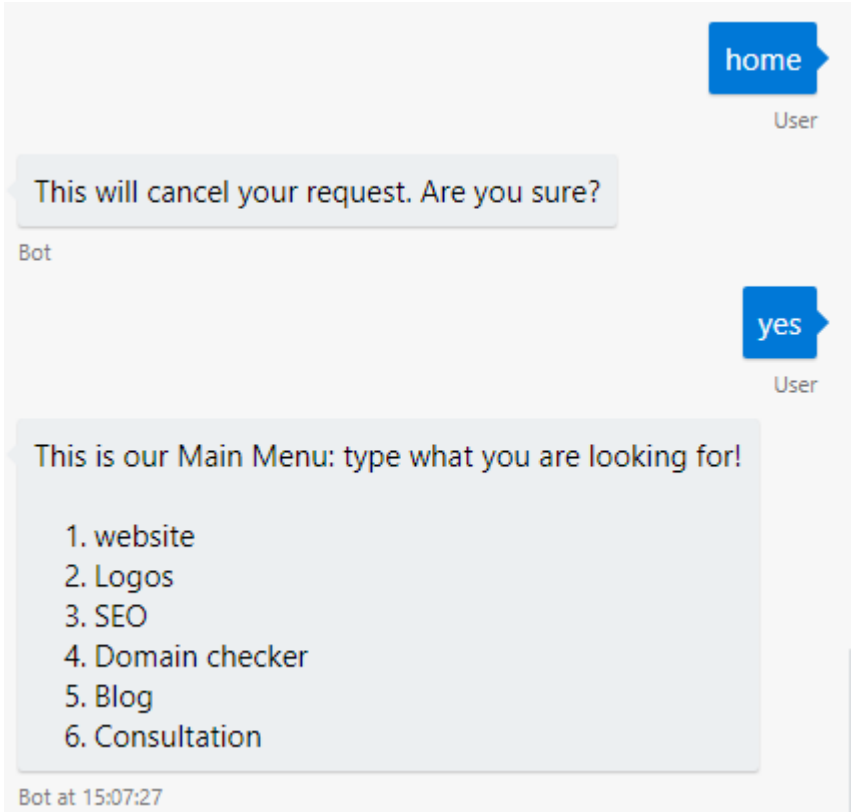
the chances of the user or bot getting confused minimal. So, any time a user entered a message that the bot would not recognize it responded appropriately see the screenshot below.



This reduces the chances of the bot not understanding things and keeps the conversation on track. I also made a “home” menu which I instructed user they can enter at any time if they feel stuck or get brought to a service they are not interested in. This again was a tough part of the project to implement as there is so many different ways the conversation can go, and so many different inputs to expect from the user.

```
bot.dialog("mainMenu", [
  function(session){
    builder.Prompts.choice(session, "This is our Main Menu: type what you are looking for! ", menuItems);
  },
  function(session, results){
    if(results.response){
      session.beginDialog(menuItems[results.response.entity].item);
    }
  }
])
.triggerAction({
  // The user can request this at any time.
  // Once triggered, it clears the stack and prompts the main menu again.
  matches: /^main menu$|^Main Menu$|^Home$|^home$/i,
  confirmPrompt: "This will cancel your request. Are you sure?"
});
```

You can see here with this messenger handler the “matches” part. This is for when the user will input main menu or Home that the bot will understand what they require and prompt the Main menu dialog. I added a TriggerAction to the mainMenu dialog, meaning that when them words are trigged(entered) the bot will display the main menu.



I added menus by doing the following.

```
var menuItems = {  
  "website": {  
    item: "buywebsitehome"  
  },  
  "Logos": {  
    item: "logo"  
  },  
  "SEO": {  
    item: "buyButtonSEO"  
  }  
}
```

Declaring a new menu and adding the items to it, Microsoft bot framework will then display the menu in a nice format for me, which is another useful feature of the framework.

Again, this was a very effective way to help ensure the flow of the conversation. This was a very tough part of the project, however I addressed it first so that it would be easier for me to work with later within the project. Overall the implementation of adding these error handling dialogs took me about 2/3 days in total. However, it was a good time to do it as it taught me how to work with input from the user and direct them to where I want them to go. After about 20 hours of coding and research I correctly implemented the home and main menu options, along with the intent recognizers from LUIS.

Another very difficult part of the project was when the user entered what they wanted, displaying these services to them correctly. For instance, when a user enters "websites" that it was displayed nicely and functioned fully to optimize user experience. I utilized the use of Microsoft bot framework built-in cards. These cards allowed me to achieve what I wanted to. I could add attachments, titles, text, and images to the card. Just like setting up the environment, the cards were a lot of

trial and error and I could not get them to work correctly with the bot. eventually I learnt how to incorporate them correctly. I create a new “session” when replying to the user. Within the session I send a reply and choose the Card layout I want to use, in this example it is the “HeroCard”. Once the user enters in logos the bot will recognize the input and respond with a new session, within the response the card is built.

```
var msg = new builder.Message(session);
msg.attachmentLayout(builder.AttachmentLayout.ca
msg.attachments([
  new builder.HeroCard(session)
    .title("Silver Logo Package")
    .subtitle("A logo to make your brand ins
    .text("Price is €75")
    .images([builder.CardImage.create(sessio
    .buttons([
      builder.CardAction.imBack(session,
    ]),
  ]),
```

This is how the card looks within the chat.



I utilised these cards as they look slick and clean and visual appealing to the user. This was the option complete as I needed to make the button function to give te user the option to buy a logo within the bot. To do this again , much like everything within the bot , I needed to create a new dialog and a new session for when the

button is clicked, and call it from the correct button of what the user is trying to purchase

```
bot.dialog('buyButtonClick', [
  function (session, args, next) {
    // Get color and optional size from users utterance
    var utterance = args.intent.matched[0];
    var color = /(silver|gold)/i.exec(utterance);
    var cat = /\b( Entrepreneur|Start Up|Blogger|Business)\b/i.exec(utterance);
```

I done this by adding colour variables to the packages, as you can see here silver and gold packages. The bot will understand that silver logo buy button was selected and begin the buybuttonClick dialog. Also adding in category variables allowed me to gainer a better insight into the type of customer that is buying the logo. The category variables will be prompted after the user selects the logo package they want.

```
.buttons([
  builder.CardAction.imBack(session, "buy silver logo", "Buy")
])
```

Now that the user can choose to see the packages and also hit the buy button but this is not the process over. I know must collect details off the user so that I can contact them about their interest in the logo. To do this I used more waterfall steps to gather the info.

Here is the waterfall process for collecting the information post logo selection.

```

session.send("Thank you for your interest in purchasing a logo , please answer these questions
    builder.Prompts.time(session, "What date do you need your Logo for?");
},
function (session, results) {
    session.dialogData.conDate = builder.EntityRecognizer.resolveTime([results.r
    builder.Prompts.text(session, "What is your email address");
    var re = /^[^@]+\.[^@]+*|([a-zA-Z0-9]+\.[a-zA-Z]{2,})$/;
    email = session.message.text;
    if (re.test(email)) {
        session.userData.email = email;
    } else {
        session.send("Please type a valid email address. For example: test@hotmail.com");
    }
},
function (session, results) {
    session.dialogData.Size = results.response;
    builder.Prompts.text(session, "what is your full name??");
},
function (session, results) {
    session.dialogData.conName = results.response;

    // Process request and display reservation details
    session.send("Logo confirmed. We will contact you via email to discuss further");
    session.endDialog();
}

```

I used a 3 step waterfall process here. You can see this with the “Builder prompts”, I am asking for the user’s data which they require the logo for, their email, and also their full name. This is sufficient information for me to be able to reach out to the user and contact them about their interaction with the bot.

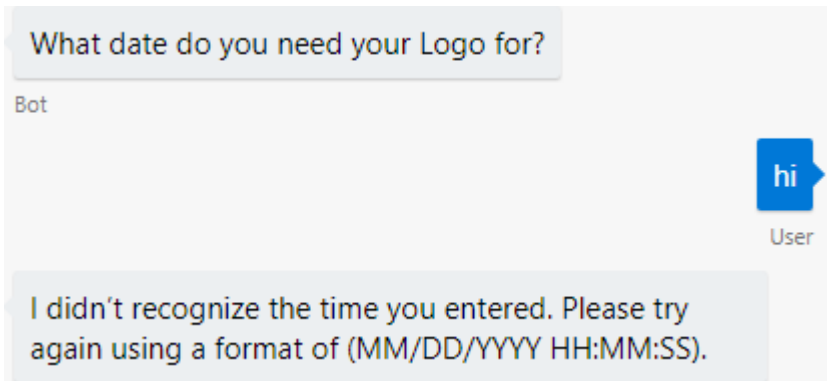
As I am a security student, I choose to validate the fields such as email. Ensuring that the user must enter a valid email address. As you can see it requires the user to enter special character @ sign and when they don’t they are prompted with the error message, as you can see in the screenshot above “session.send” sends the message asking for the user to enter a valid email. Implementing security features like this into the bot was a difficult task , and required me to research node js security techniques such as the requirement of the @. Although validation is a basic feature and easily implemented into html, php or other languages within the bot framework there is a restriction as to what you can do with the code so again trial and error was used here until I successfully got the validation to work.

Please type a valid email address. For example:  
[test@hotmail.com](mailto:test@hotmail.com)

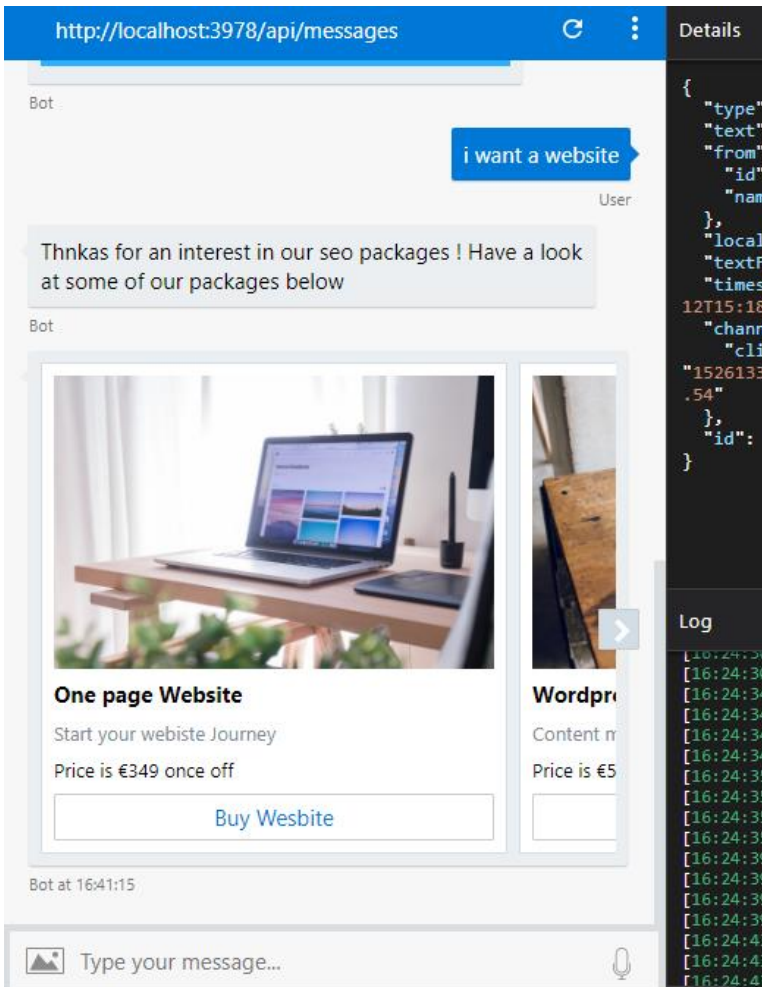


Above you can see the validation message that shows to the user within the bot.

Another validation field I entered was on the Date for logo prompt. Ensuring the user must enter date in the format MM/DD/YYYY, you can see this validation in action in this screenshot below.



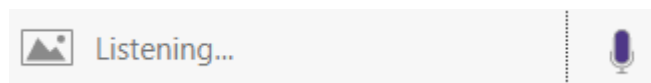
I



A cool feature which I added into the bot is that it can speak. I was able to do this using adaptive cards. Adaptive cards allowed me to attach the 'speak' tag so I code add what I want the bot to say. You can see in this screenshot below

```
// Display adaptive card to help user get in touch with designer
var card = {
  'contentType': 'application/vnd.microsoft.card.adaptive',
  'content': {
    '$schema': 'http://adaptivecards.io/schemas/adaptive-card.json',
    'type': 'AdaptiveCard',
    'version': '1.0',
    'body': [
      {
        'type': 'Container',
        'speak': '<strong>Welcome to consultation booker</strong>',
        'items': [
```

I tested this in the emulator and it worked correctly. The emulator has a listening option, I hit the microphone button in the emulator and then I can talk to the bot. For example I can say “websites” and it prompts the website functionality. When the bot hears what you say into it, it matches with what it hears best and then speaks the response it has. I am proud of this functionality and I think it makes the bot a lot better.



It was a tough feature to add and involved a lot of testing but it was a nice feature to add as it makes the bot a lot more flexible and scalable. I feel users will enjoy the option to speak to the bot.

After all dialogs were added and the conversation flow was fully in place the next stage was to put the app live on Facebook and on a website so that users can start using it. Deployment was definitely one of the most difficult parts of this project , it took me a few days to get it correctly deployed but once I did I could then update the bot easily using git push commands from my command prompt which is a fast and convenient way to update the chatbot.

To begin my deployment, I needed to register for a web app on azure. This acts as a messaging endpoint for the bot. The url is displayed here below. Api/messages is added to the url so it is listening to any messages sent here.

Bot Service pricing tier

S1

Messaging endpoint

<https://ncinowbot.azurewebsites.net/api/messages>

Once the web app was deployed the next part of the deployment process was to register an app with azure. In this case it was a chatbot application. I registered the chatbot application on the azure portal, and used the web app url I had previously registered with azure. Once I have registered the web app and the chatbot applications I then connected my github to azure and push my application from git hub to azure. When I started I thought this would go smoothly however I got many errors and it took me a few days to get working, everytime I pushed with git it returned me errors in the code and didn't seem to understand the bot dialogs. You can see the various deployments here

—	auth erros GitHub Inactive 1:48 PM
—	old keys GitHub Inactive 12:17 PM
—	hhttp server code added GitHub Inactive 4:01 AM
—	First commit GitHub Inactive 3:34 AM
—	Merge branch 'master' of <a href="https://github.com/stannci/Nowbot">https://github.com/stannci/Nowbot</a> GitHub Inactive 3:24 AM
—	Delete saveData.php GitHub Inactive 3:15 AM
—	143 GitHub Inactive 3:11 AM

Every time I make a change in the app.js file it is noted, to check I just use git status command to see what changes have been made before I push live. I found this very useful as it allowed me to double check changes before push the app live to the master branch.

```
C:\Users\hp\Desktop\Chatbot>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
  (commit or discard the untracked or modified content in submodules)

        modified:   Nowbot (untracked content)
        modified:   app.js

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\hp\Desktop\Chatbot>
```

When I got it working I could then deploy onto web client and Facebook messenger. Which I done successfully. These were the two main places I wanted to deploy my application and was very happy that I got it deployed on both of them. Now the bot is live and can I can demonstrate it on both web client and Facebook.

Web client URL - <http://plusninedesign.com/chatbot/>

Facebook url - <https://www.facebook.com/NowBot-1355910854555169/>

### **2.3 Graphical User Interface (GUI) Layout**

Screenshot of welcome message user will be greeted with when they enter the conversation with the bot.

Hi, Welcome! Im NowBot, I will be your assistant today!

Here is what I can help you with

Website Services

Logo Packages

Domain checker

SEO Packages

Online Consultation

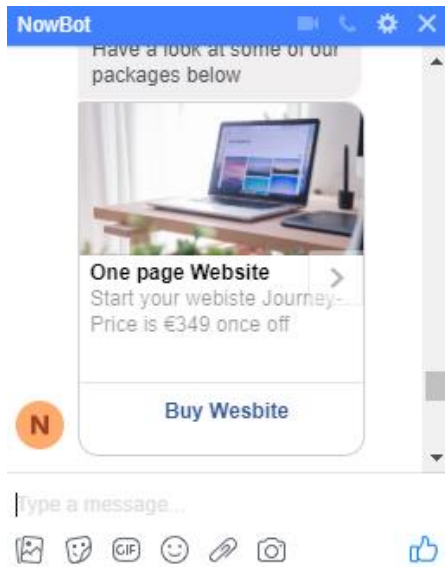
If you get lost at anytime just say help

Bot at 20:31:35



Type your message...

This is the bot working in messenger.



This is the bot GUI on the website, the same as the emulator.



## CHATBOT


Chat

Hi! Im NowBot, I will be your assitant today!

Here is what I can help you with means you have extra features!  
such as our Discovery Section! To access just type Discover.

- Website Services
- Logo Packages
- Domain checker
- SEO Packages
- Online Consultation

nowbotnci at 9:08:29 PM

 | type your message...






This is the website dialog within the chatbot.

I want a website


User

Thank you for an interest in our website packages  
! Have a look at some of our packages below

Bot



**One page Website**  
Start your webiste Journey  
Price is €349 once off  
[Buy Wesbite](#)



**Wordpress Website**  
Content managemen  
Price is €549 once off  
[Buy Webs](#)

Bot at 20:21:38

Type your message...

buy one page Website

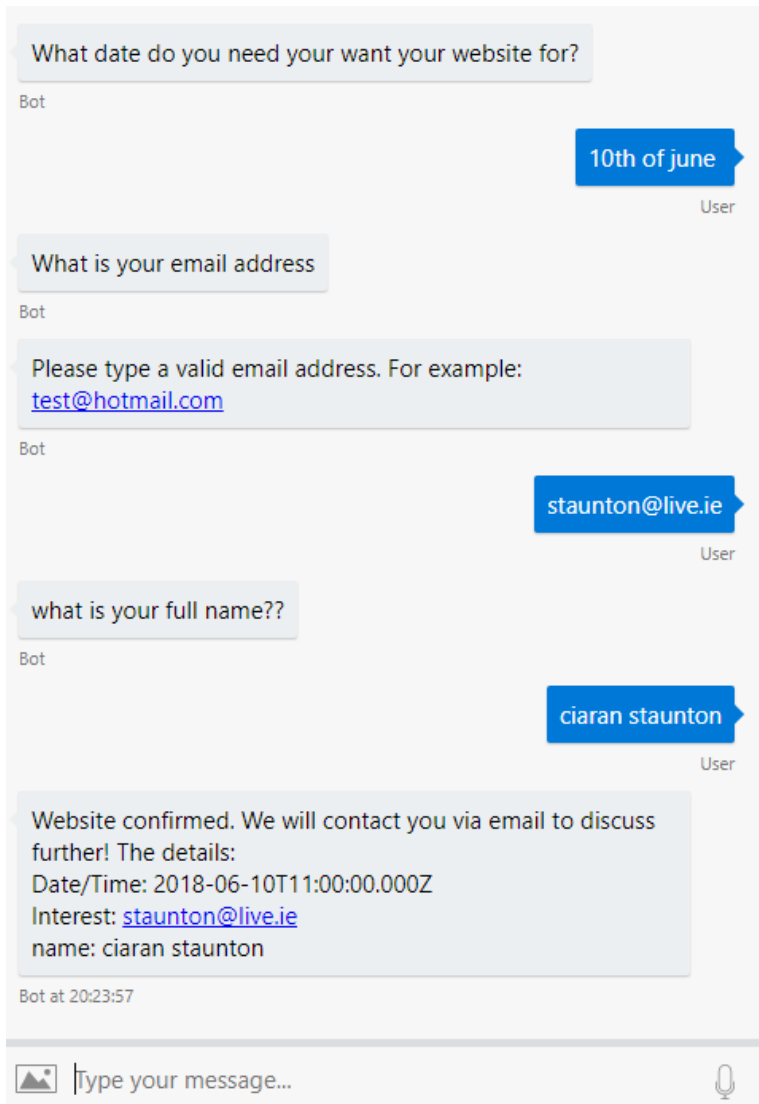
User

Please choose what matches you most?

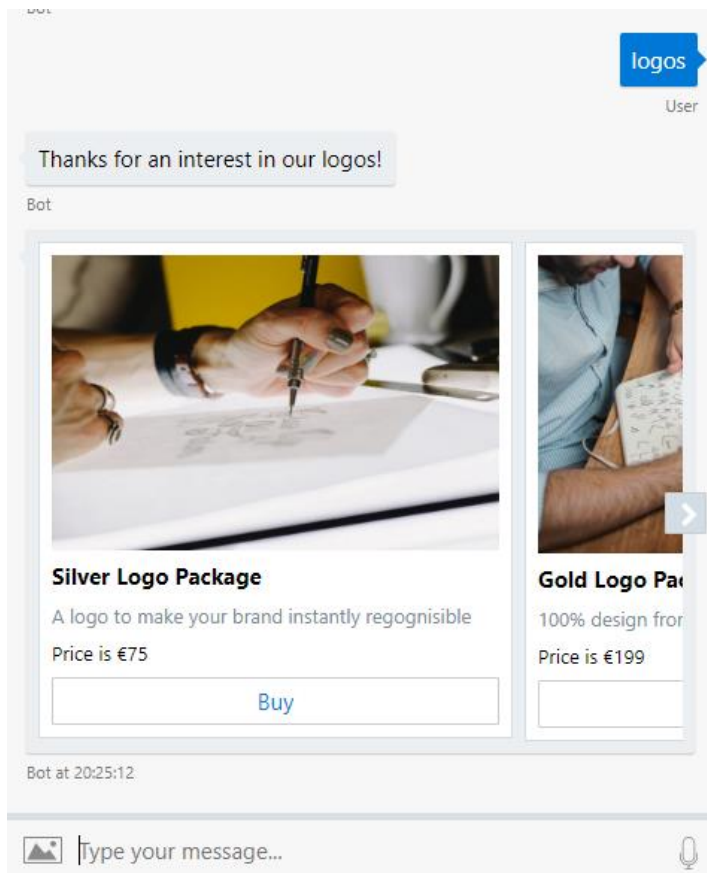
1. Business
2. Blogger
3. Entrepreneur
4. Start up

Bot at 20:22:51

Type your message...



This is the logo section within the bot.

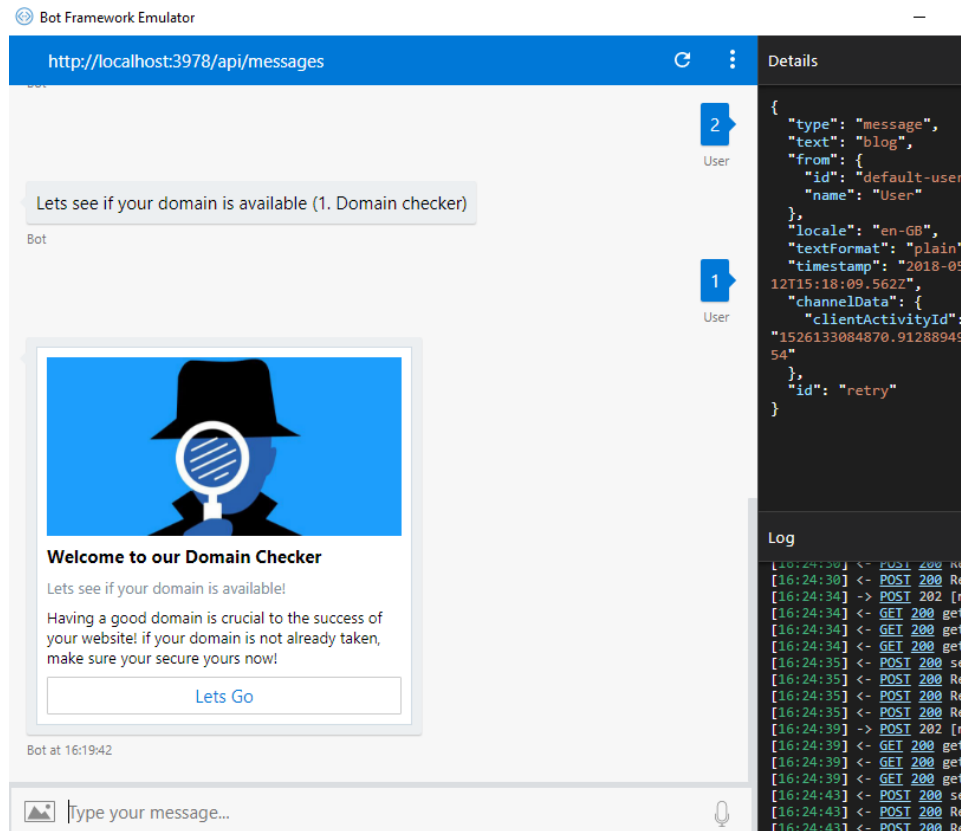


## 2.4 Testing

To test the bot, I will use Microsoft emulator. It is a Microsoft desktop application. It allowed me to test my bot locally and run appropriate tests. I will also have to test my bot from the view of a user, and think like a user asking it questions which I feel people may ask the bot. This will require a large amount of testing to prepare for the huge variety of questions that anyone may ask. I will also have friends and family test out my Chatbot. I will ask them to try test it out for a few minutes and find bugs, take notes of any pros and also take note of any cons that they found whilst using the bot. I will then take these pros/cons and bugs, go back and apply necessary changes that are needed to the chatbot. This will be a very effective way to improve the usability of the bot and a great way to test and find bugs.

The emulator was really one of the key parts of this entire project. It is an excellent service provided by Microsoft. It lets you imitate a conversation in the bot as if it is

deployed live. Testing locally like this so conveniently allowed me to work much more efficiently and let me test my code immediately. Here you can see me testing the domain checker functionality within the bot.



Building a chatbot was a new development experience for me, it was not like other development that I have done, it involved a lot of repetitiveness. I would implement new code and would need to begin the conversation from the start all over again and follow the whole dialog until I get to where I want to go in the conversation, only to discover it does not work "oops, something went wrong here" is an error message I seen many times throughout this project. This message sends when the bot code is broken. Throughout the testing of this chatbot I hit the end points many times as you can see from the screenshot below. I was continually restarting the emulator and beginning conversations again to test the dialog and functionality within the bot.

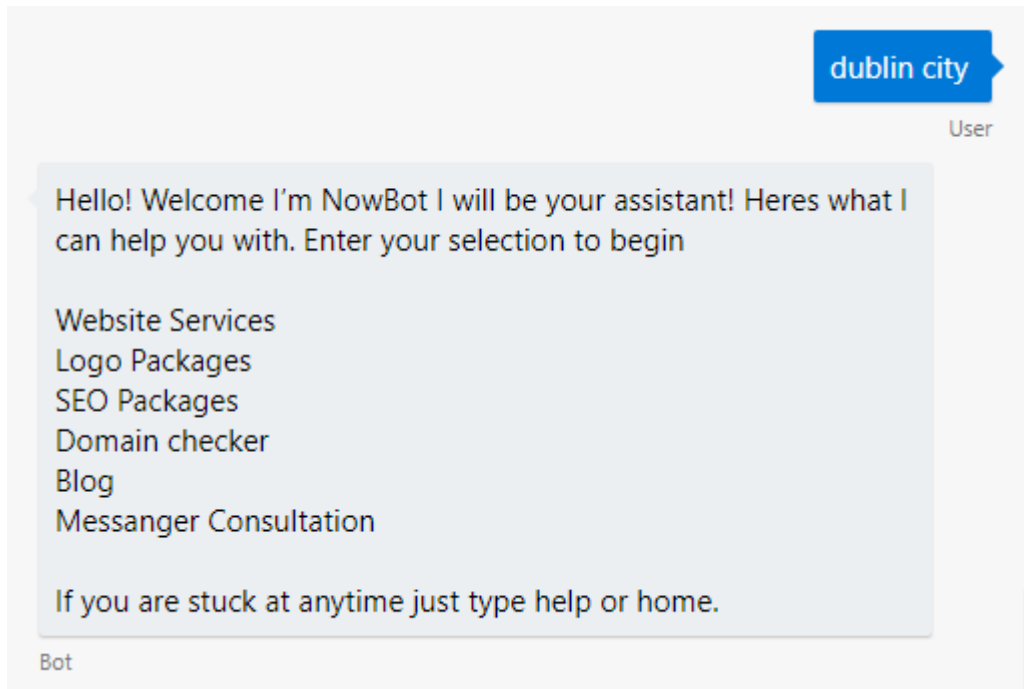
Quota Scope	Calls	Call Quota
Subscription	-	-
API: LUIS Endpoint API	751.00	1,000.00

I had to do a lot of manual testing through the emulator and basically tried to end the dialog in every way I could. I made response dialog which takes in any response that the bot is not familiar and sends the user to home. So any time the bot gets a message it is not familiar with it will send them back to the home dialog, you can see this demonstrated in the screenshots below.

```
//language understanding for general words/phrases that a user
intents.matches('Repsonse', (session, args, next) => {

    //this is the prompt for Facebook/messenger users
    session.send("Hello! Welcome I'm NowBot I will be your a
    var msg = new builder.Message(session)
    .addAttachment({
        contentUrl: 'http://plusninedesign.com/wp-content/up
        contentType: 'image/jpg',
        name: 'plusninedesign.jpg',
    });

    session.send(msg);
};
```



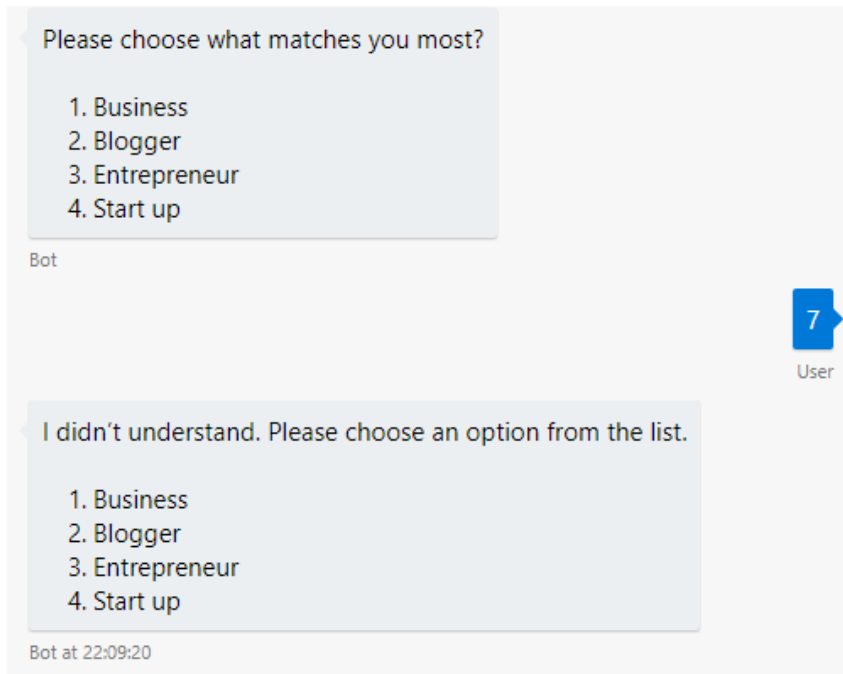
In this case I entered in Dublin city which is not a command or dialog within the bot , so the bot does not understand it. This is where the response dialog kicks in , and sends the user the welcoming message.

I also tried to break the dialogs like you can see here. By selecting options that the bot does not offer and to see how the bot will handle them. When the user enters an option that is not valid the bot will simply respond “I didn’t understand” and reiterate the original options.

I also done some manual case testing for the project. The manual cases test covered all of the key dialogs and functionality within the bot that are crucial to the bot performance. You can see the results from the tests below.

Test case ID	Test type	Steps	Expected Action	Actual action	Pass/fail
1	Welcome message <u>Bot.on test</u>	Go to <a href="http://plusninedesign.com/chatbot/">http://plusninedesign.com/chatbot/</a> -Join session -Wait for a welcome message	Send welcome message	Welcome message sent	Pass
2	Test for website packages	Go to <a href="http://plusninedesign.com/chatbot/">http://plusninedesign.com/chatbot/</a> -Ask the bot for a website	Shows website packages	Bot showed website packages	pass
3	Test Logo packages	Go to <a href="http://plusninedesign.com/chatbot/">http://plusninedesign.com/chatbot/</a> -Ask the bot for a logo	Shows logo packages	Bot showed logo packages	pass
4	Test bot on messenger	Go to <u>NowBot facebook page</u> Hit send message Send message Wait for response	Bot responds	Bot responded in messenger	pass
5	Test bot on website	Got to <a href="http://plusninedesign.com/chatbot/">http://plusninedesign.com/chatbot/</a> -See if the bot is working	Bot fully functional	Bot was fully functional	pass
6	Test LUIS intents	Got to <a href="http://plusninedesign.com/chatbot/">http://plusninedesign.com/chatbot/</a> -Type Domain Wait to be sent to domain options	Bot should send user to the domain function	Bot prompted domain functionality	pass

I also tested the bot via voice. I could do this in the emulator. It has a listening option much like cortana. This was a useful feature to test my bot and see how it works with speech.



## ***2.5 Customer testing***

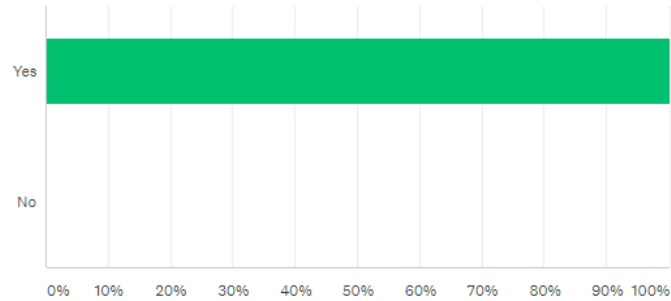
I done a survey to find out how users felt with interacting with the bot and to see if they found talking to the bot beneficial or interesting. I will post the survey results here. I sent the survey out to friends and family and got back 16 responses from users who had interacted with the bot. I began by send the url to the bot first followed by the survey so they knew exactly what they were answering in the survey. I will post the most interesting responses to the survey here. I was very pleased to see the answers.

To begin with I found the results from this question very motivating. It shows that people are inclined to talk to bots now and the technology is moving forward and so are we as people, moving forward with it.



Would you be comfortable talking to the chatbot in order to help you with web design services?

Answered: 16 Skipped: 0

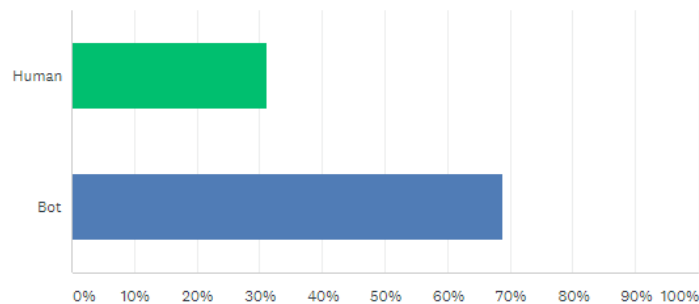


ANSWER CHOICES	RESPONSES	
▼ Yes	100.00%	16
▼ No	0.00%	0
<b>TOTAL</b>		<b>16</b>

A good follow up to this question is this one here. 68% of people said they would rather talk to a chatbot, I would also answer the same. If the bot is built correctly and can understand the inputs from users it can be programmed to answer all of the questions user may have, and it will do this instantly unlike many live chats that company's use as customer service.

Would you rather talk to the chatbot or a member of customer support?

Answered: 16 Skipped: 0

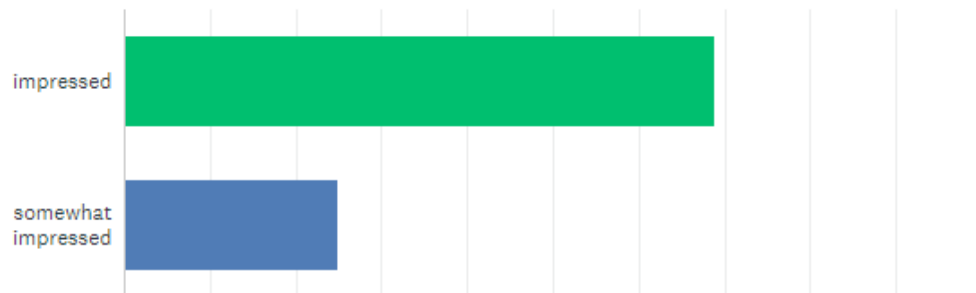


ANSWER CHOICES	RESPONSES	
▼ Human	31.25%	5
▼ Bot	68.75%	11
<b>TOTAL</b>		<b>16</b>

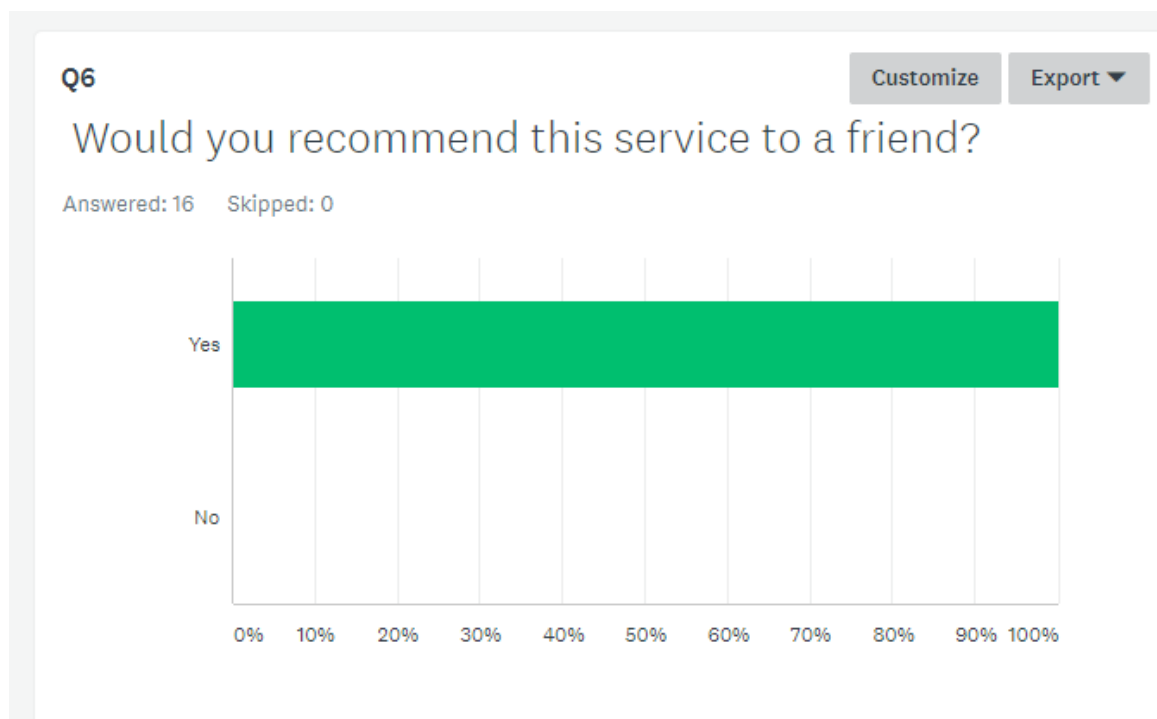
Another pleasing result was to see that majority of users were impressed with the chatbots functionality.

## Did you find the chatbots help impressive or useful?

Answered: 16 Skipped: 0



Another positive answer is this one here, that 100% of users would recommend using this chatbot to a friend.



The chatbot understood 100% of the participants messages which again is very pleasing to see.

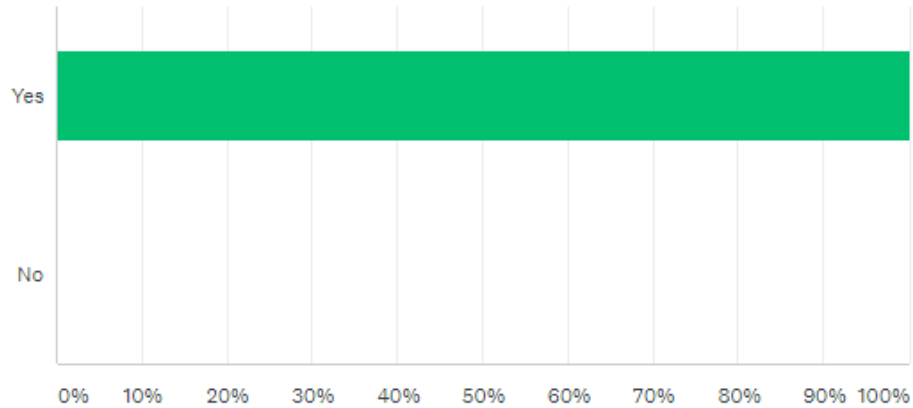
Q5

Customize

Export ▼

Did it understand your messages?

Answered: 16 Skipped: 0



### **3 Conclusions**

My conclusion of this project is that I enjoyed the process of the development and building of this project. I liked working with the new technologies I used and I am happy to have learnt the skill of building chatbots. I am also proud of my project and proud of myself for designing and building it. I had doubt in my mind at the beginning as I knew nothing about developing chatbots and also new it would be a very difficult build. I was a tough project no doubt, but I am happy I challenged myself and pushed myself to do something a bit tougher than the average web app. I don't see any disadvantages with this project, as it is a project which is current, modern and can constantly be evolved and updated using artificial intelligence. It has many advantages such as its commercial potential, with its ability to both make and save money for businesses and represent the company in a high manner. I felt it was a massive learning experience and a great learning curve for me which is why I enjoyed it so much. I am happy I can now build bots and have a project to demonstrate to friends and family and use for my freelance web design that I am proud of.

## 4 Further development or research

With more resources, where could the results of this project lead to?

With more resources the project could prepell to an all in one Branding / marketing and digital agency. Ideally you could just message the chatbot and ask about any topic of internet marketing and

Further development of this app , I would like to add in the functionality of taking credit card details through the app

## 5 References

- Sciencedirectcom. 2018. Sciencedirectcom. [Online]. [Available from: <https://www.sciencedirect.com/science/article/pii/S0747563215001247>].
- Researchgatenet. 2018. ResearchGate. [Online]. Available from: [https://www.researchgate.net/profile/Eric\\_Atwell/publication/220046725\\_Chatbots\\_Are\\_they\\_Really\\_Useful/links/00b7d518ab7329add2000000/Chatbots-Are-they-Really-Useful.pdf](https://www.researchgate.net/profile/Eric_Atwell/publication/220046725_Chatbots_Are_they_Really_Useful/links/00b7d518ab7329add2000000/Chatbots-Are-they-Really-Useful.pdf)
- Chatbotslifecom. 2018. Chatbot's Life. [Online]. Available from: <https://chatbotslife.com/which-are-the-best-on-site-chatbot-frameworks-3dbf5157fb57>
-

## 6 Appendix

### 6.1 *Project Proposal*

# Project Proposal

**TITLE**

**Web Design Chatbot**

**Nowbot**

Student Name, Number, email address

Ciaran Staunton x14496288 , x14496288@student.ncirl.ie

Degree Programme Name e.g. BSc (Hons) in Computing

Specialisation (if applicable, e.g., Software Systems, Cloud Computing, Gaming  
and Multimedia, Networking and Mobile Technologies)

Cyber Security

## **Objectives**

I have many objectives for this project which I will discuss here, I feel these objectives are important to my project succeeding. My main objective for my fourth-year project is to make a fully functioning chatbot using Microsoft bot frameworks. The chatbot will be called "NowBot" as it is current developed from new technologies and providing current new relevant data to the user. Providing fast and instant replies to the user is one of my main objectives for this project, I feel if the bot is lagging or slow the user will just exit the session so having this objective is crucial to making this project succeed. Making the application suitable for real world use is also another objective of mine, I feel there is a need for this kind of application which is a one of the reasons I'm deciding to build a chatbot. Another objective I have it to make the is to make the Nowbot available across web and mobile devices and integrate it with Facebook. My final objective for this project is to make the bot fun and engaging experience that the user will enjoy, which I feel I can achieve throughout development.

## **Background**

I wanted to do something for my final year project that would be up to date and modern for today's world and something I can use in the real-world post college. With this in mind I felt that a chatbot would be the best possible solution for my project, I feel that this is future of technology and already plays a massive role in business, many businesses use chatbots for customer service, payment gateways etc. AI is a massive part to the future and chatbots are an exciting new technology that many people are still yet to have come into contact with, it is still a fresh, powerful and fun technology for the user. I wanted to do a project relating to chatbots and AI to learn more about the technology and to gain some experience in addition to the module. I also find this area of technology and web design/ marketing very interesting which is another reason I wanted to do AI chatbot for my final year project.



## **7 Technical Approach**

I had no prior knowledge to AI or chatbots. So, I needed to do a lot of research to help me gain a better understanding of this field. I watched a lot of YouTube videos on Microsoft Chatbot Frameworks to understand how this framework works. I also had to research which languages are the best to use to build chatbots. After my research, I have chosen to Use Microsoft Chatbot Frameworks and node js. I picked Microsoft Chatbot Frameworks because it is the best environment out there for building chatbots, I also have used many Microsoft environments before such as Microsoft Visual Studio and Visual Studio Code which I became comfortable with after my internship as this was the main environment all of the senior developers used, which is another contributing factor as to why I have chosen this environment's. My requirements for this project are to make the chatbot user friendly and have it help the user out with their website development and by the time they are finished chatting to the bot feel comfortable moving forward with development of their website.

## **8 Special resources required**

I require no hardware requirements for my project but I do require a hosting platform, I will be hosting my bot on Microsoft azure. Hosting it live will enable me to get users to use my chatbot and this will allow me to find any bugs or errors within my application.

## **9 Project Plan**

<b>Project Proposal</b>	<b>30 days</b>	<b>Mon 18/09/17</b>	<b>Fri 27/10/17</b>
research	14 days	Mon 25/09/17	Thu 12/10/17
project pitch	1 day	Fri 06/10/17	Fri 06/10/17
project proposal document	11 days	Mon 25/09/17	Mon 09/10/17
Journal	1 day	Sat 30/09/17	Sat 30/09/17
<b>Requirements</b>	<b>17 days</b>	<b>Wed 01/11/17</b>	<b>Thu 23/11/17</b>
gather requirements	3 days	Thu 02/11/17	Mon 06/11/17
research	2 days	Tue 03/10/17	Wed 04/10/17
requirements document	12 days	Wed 04/10/17	Thu 19/10/17
journal	1 day	Mon 30/10/17	Mon 30/10/17
<b>Mid Point Presentation</b>	<b>14 days</b>	<b>Mon 23/10/17</b>	<b>Thu 09/11/17</b>
tech document	10 days	Mon 13/11/17	Fri 24/11/17
Prototype	6 days	Mon 20/11/17	Mon 27/11/17
presentation	1 day	Mon 06/11/17	Mon 06/11/17
journal	1 day	Fri 01/12/17	Fri 01/12/17

## 10 Technical Details

There are many technologies available out there to make chat bots. Various apps and platforms take a lot of the heavy lifting out of developing the chat bot and give you out of the box environments and require no coding, while these are great for building simple chatbots I plan on building a more complex bot therefore I am using Microsoft Bot Framework. It is the most powerful way to build chat bots currently, it can allow me to achieve the functionality I require for my project such as calling API's. Microsoft bot framework allows you a lot more flexibility when it comes to developing chatbots unlike many other platforms. The technologies I plan on using the chatbot are node.js and c# and the environment I will use will be Microsoft Visual Studio Code. I will be using a program called LUIS which is a language understanding intelligence service. Using LUIS will allow me to program the bot to try understand to its best of its ability what the user is inputting. LUIS stores data in a database which I will manual enter. When a user enters text LUIS will take this info and check the database and try match it to the most suitable solution it feels it is, it also scores every result so for example a user may enter "I would like a web

site” and in LUIS I have entered “I need a website” so LUIS will receive the text and try match it to the best matches in the database, this one example may get ranked with a 80% match so it will register and understand what the user is saying and respond appropriately. Another technology I will be using is QnA maker. This is a free REST API that I can use to train the bot to respond and answer questions in a more human like conversational way. IT is an extremely useful tool and will allow me to achieve what I need to do with my bot. I will be using Microsoft AZURE for deployment and also Facebook messenger for deployment across social platforms. Using Microsoft’s Emulator, I will be able to test the bot locally which is an extremely convenient feature which Microsoft provide. The main language will be node.js for my chatbot again because node will allow me to achieve my objectives, I will also be using some c#. These are the most effective ways to make a bot currently. They will contribute to my project by allowing me to develop a conversational chatbot, letting me program the conversation outputs which will take place and allowing me to carry out the functions which are required.

## **11 Evaluation**

Describe how you will evaluate the system with real technical data using system tests, integration tests etc. In addition, where possible describe how you will evaluate the system with an end user.

Signature of student and date

### ***11.1 Project Plan***

### ***11.2 Monthly Journals***

#### **September**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: September

## **My achievements**

This month, I was able to achieve some research into different ideas and gain an understanding of the software project module and how exactly it is graded and what is expected in the coming months. I also got an idea to pitch which was a great achievement.

My contributions to the projects included research and pitching my idea

## **My Reflection**

I felt, it worked well to research on youtube and various websites to try and discover the most modern technologies available. I also felt my pitch went well as my idea got approved I am happy with that.

However, I feel I need to add more functionality to my idea to ensure it is complex enough

## **Intended Changes**

Next month, I will try to find out in more detail of the technologies required to build the chat bot which I pitched.

I realized that I need to think greatly into the technologies to use to ensure I choose one which will allow me to build and develop exactly what I need.

## **Supervisor Meetings**

Meetings did not take place until October however emails have been exchanged

Date of Meeting:

Items discussed:

Action Items:

## **October**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: October

## **My achievements**

This month, I was able to achieve some further research into technologies I need to build my chat bot and also finally decide on the environment I will be using which will be visual studio code.

My contributions to the projects included more research and also completed the project proposal. This helped me to understand more clearly which direction I am going in with my project and felt it was beneficial. I have also install my environments and am ready to code and begin work on the prototype.

## **My Reflection**

I felt October was a good month for project development, I gained great knowledge into the technologies needed and also the environment. I also completed my project proposal and now feel in a comfortable position moving forward with the project.

However, I feel I need to start development as soon as possible as I have a lot of work to do.

## **Intended Changes**

Next month, I will begin coding my chatbot

I realized that I need to time manage as fourth year is consistently exams and CA's so good time management will be essential to the success of the project and the year.

## **Supervisor Meetings**

Date of Meeting: 27-10-17

Items discussed: project proposal, requirements specification further meetings and position the supervisor plays in the project development.

Action Items: organized a time and day for weekly meetings

## **November**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: November

### **My achievements**

This month, I uploaded my requirement specification document and am working on the technical report doc which will include the requirement's specification document and the project proposal so it was handy to have both of them done it made the tech doc easier. I have also begun to code for the prototype and make my presentation for the presentations next week.

### **My Reflection**

I felt November so far has been ok, it has been somewhat stressful with the term coming to an end trying to reach project deadlines and submit all CA's on time.

However, I feel after this month project development

### **Intended Changes**

Next month, I will continue with development of the bot.

I realized now how to add dialog to the bot and will continue to do this and grow the bot over the coming weeks and months

### **Supervisor Meetings**

Date of Meeting: 17-11-17

Items discussed: tech document and presentations

Action Items: worked on tech doc

### **December**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: September

### **My achievements**

This month, I focused on gaining more knowledge of dialogs within the bot so I can be ready for development next month.

My contributions to the projects included research and pitching my idea

### **My Reflection**

It was a good month for research, I learnt about dialogs and LUIS intents

### **Intended Changes**

Next month, I will try to implement what I learnt

### **Supervisor Meetings**

I had 2 supervisor meetings in december

### **January**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: September

### **My achievements**

Began to expand on dialogs in the bot

### **My Reflection**

I would like to understand them even more for next month

### **Intended Changes**

Next month, I will try to find out in more detail of the technologies required to build the chat bot conversation.

### **Supervisor Meetings**

3 meetings took place this month

### **February**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: September

### **My achievements**

Implemented QnA Microsoft cognitive services

### **My Reflection**

I had a good month this month and learnt a lot to benefit for next month

### **Intended Changes**

Next month, I will try to find out in more detail about LUIS

### **Supervisor Meetings**

I had 3 meetings this month with my supervisor

### **March**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: September

### **My achievements**

This month, I was able to achieve adding LUIS to my chatbot. My chatbot now understands natural language processing

### **My Reflection**

I felt this month went successfully and am looking forward to next month development stages

### **Intended Changes**

Next month, I will try to deploy my application live

### **Supervisor Meetings**

I had 2 meetings this month

Date of Meeting:



Items discussed:

Action Items:

## **April**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: September

### **My achievements**

This month, I was able to put my application live on Facebook and my website

### **My Reflection**

I had a great month this month and am happy with the work that was done

### **Intended Changes**

Next month, I will try to add in more code to my dialog and waterfalls

### **Supervisor Meetings**

I had 2 meetings this month

Date of Meeting:

Items discussed:

Action Items:

## **May**

Student name: Ciaran Staunton

Program (e.g., BSc in Computing): BSc in Computing: cyber security

Month: September

### **My achievements**

This month, I was able to work on my report and get ready for upload and presentations and showcase

## **My Reflection**

I felt, it worked well to research on youtube and various websites to try and discover the most modern technologies available. I also felt my pitch went well as my idea got approved I am happy with that.

However, I feel I need to add more functionality to my idea to ensure it is complex enough

## **Intended Changes**

Next month, I will try to find out in more detail of the technologies required to build the chat bot which I pitched.

I realized that I need to think greatly into the technologies to use to ensure I choose one which will allow me to build and develop exactly what I need.

## **Supervisor Meetings**

I had 5 meetings this month

Date of Meeting:

Items discussed:

Action Items:

### **11.3 Other Material Used**

Any other reference material used in the project for example evaluation surveys etc.