Cannabot - A Friendly CTS Assistant

Canada School of Public Service - Digital Academy Pilot - Practicum May - June 2019

# Background

The Premium learning program from CSPS Digital Academy ends with a *Practicum* - a 14 day intensive training with participants from various Premium streams (Data, AI/ML, Design and DevOps) are brought together to tackle real business problems put forward by Solution Leaders from the Leadership learning program.

The learners cohort was divided into teams(4-6 people) to work on 9 different problems brought forward by executives from various departments and agencies. Each team included representatives from each of the Premium Streams.

Our team - self declared as the “Cannabis Knights” - was composed of the following members:

* **Allan Phillips** - Natural Sciences and Engineering Research Council of Canada (NSERC)  
  Premium program participant from the Data stream
* **Nidhi Kalia** - Office of the Superintendent of Financial Institutions (OSFI)   
  Premium program Participant from the Data stream
* **Patrick Messier** -Canada School of Public Service (CSPS)   
  Premium program Participant from the Development and Operations (DevOps) stream
* **Sylvain Pâquet** - Canada Border Services Agency (CBSA)  
  Premium program participant from the Machine Learning and Artificial Intelligence (ML/AI) stream
* **Ren**[**é**](https://en.wiktionary.org/wiki/%C3%A9)**e Isaac-Saper** - Canadian Heritage  
  Premium program participant from the Design stream
* **Lindsay Robles** - Health Canada  
  Leadership program participant and Solution Leader

# Introduction

The problem assigned to our team originated from Health Canada and was formulated as follow:

“Illicit cannabis poses health and safety risks for Canadians. How can we better leverage the cannabis tracking system to prevent and detent inversion/diversion of cannabis into/out of the legal supply chain.”

In the first week of Practicum, the team worked with the Solution Leader to refine the problem. The team and Solution Leader ultimately decided to pursue the concept of designing a chat bot to help the CTS reportees on getting responses to their Frequently Asked Questions (FAQs).

# Solution Context and Description

**Note**: The following text is a verbatim of the presentation done at the Digital Academy Premium Show Case held on June 11th 2019 at the Bayview Yard in Ottawa.

On October 17th 2018, recreational cannabis production, sales and use was legalized in Canada. While this event meant new business opportunities for the private sector, it also meant a whole new set of responsibilities and regulations for the Canadian Government.

As a result, Health Canada became responsible for cannabis licensing, and tracking cannabis through the supply chain. To meet these responsibilities, Health Canada implemented a new system called the Cannabis Tracking System (CTS) and every Licensed Producer is required - by regulation - to report their activities on a monthly basis.

As we speak today, there are more than 170 Licensed Producers, and 100’s more waiting to get a license. While few of them are actually large and well established corporations, most of them are in fact small to medium size enterprises.

So picture this. On one side, you have small licensed producers, working day in day out, looking after their crops and operations, and ultimately trying to get their new business afloat and profitable. These people are passionate about cannabis production and transformation...but not so much about regulation. Unfortunately, they typically wait until the eleventh hour to complete their reports. When they run into issues with reporting the specifics of their production through the CTS, they are left with two options: try to find the answer in the 50 page long CTS user guide, or send an email to the generic Health Canada email address, hoping to get an answer by the next business day so that they can report on time. Ultimately, if they don’t find or get an answer on time, they usually end up reporting using their best guess.

On the other side at Health Canada, the small CTS team is literally flooded with information inquiries towards the end of the reporting period, and they rush through the emails, trying to help as many Licensed Producers to report on time as they can. They also run into issues while loading the data into the CTS system as the analysts often discover all sorts of data inconsistencies. The team typically ends up data wrangling, trying to contact the Licensed Producers and have them correct and resubmit their data. The next thing they know, by the time the data is corrected, a new reporting cycle begins.

Well, we believe that it does not have to be this way.

* What if we could provide 24/7 reporting assistance to Licensed Producers?
* What if we could reduce the number of information inquiries sent to Health Canada each month?
* What if, by doing so, we allow the CTS Staff to focus on real data analysis rather than data wrangling?

We believe that there is a simple, cost-effective and readily available solution to this... introducing the “Cannabot - the Friendly CTS assistant”.

The Cannabot is simply a Web chat bot loaded with the FAQs extracted from the CTS User Guide and from the most popular inquiry emails - and responses - received and sent by Health Canada.

The Cannabot will stand between the Licensed Producers and Health Canada and will handle the first line of information inquiries. If the Cannabot can’t answer a specific question, the Licensed Producers will be invited to communicate with the CTS staff like before through email. Once the CTS staff have answered a new question, they will simply add it to the bot evolutive knowledge base so that the next time the question is asked, the bot will be able to answer it.

We are very excited about this solution as it represents a “triple win” situation. Business-wise, we believe the Cannabot can reduce the burden on the CTS team, and improve data quality and timeliness. For Licensed Producers, the Cannabot can provide them with 24/7 assistance with their reporting. Finally, on the technical side, the solution is cost-effective, readily available, evolutive and doesn’t require programming skills.

# Solution Implementation

You will find in this section a description of the concepts supporting our solution; the prerequisites to reproduce our solution; the steps to reproduce our solution; and finally a list of the known issues.

## Concepts

We implemented our solution using Microsoft [Azure Bot Service](https://azure.microsoft.com/en-in/services/bot-service/), [QnA Maker](https://azure.microsoft.com/en-ca/services/cognitive-services/qna-maker/) and [Bot Framework Web Chat](https://github.com/microsoft/BotFramework-WebChat) and [Github Pages](https://pages.github.com/).

Conceptually speaking, three layers are involved in a chatbot solution:

|  |  |  |
| --- | --- | --- |
| **Layer** | **Purpose** | **Implemented with** |
| Knowledge Base | Hold and manage the bot “knowledge”. | Azure QnA Maker |
| Bot service | Process the user queries and search the knowledge base for best answers. | Aure Bot Service |
| Channel | Provide the user interface. | Microsoft Bot Framework Web Chat  Github Page |

## Prerequisites

In order to reproduce our solution, one needs the followings:

* **Microsoft Azure Subscription** - In the context of the Digital Academy Practicum, we used [Microsoft Azure free subscription offer](https://azure.microsoft.com/en-ca/free/search/?&OCID=AID719803_SEM_v7EIU8wH&lnkd=Google_Azure_Brand&dclid=CjgKEAjwitPnBRDX-L6Hm8jCogISJACB9OyC4R3q-3MBRKN9jsMOn4R7N2JqDdsldt15oYvrHAjTaPD_BwE) to implement our solution. While this was good enough for demonstration purposes, the free subscription offer is not suitable for a production use case. For production ready implementation, you will need to choose one subscription model amongst the [Azure purchasing options](https://azure.microsoft.com/en-us/pricing/purchase-options/).
* **Microsoft Azure Account** - Once an Azure Subscription has been procured, user accounts must be created or assigned to the subscription to allow people to connect and use the subscription. Depending on the subscription option chosen, the user account creation process may take several forms. This aspect will typically be taken care of by your IT department or whatever entity that provides you with the Microsoft Azure Subscription. In the context of the Practicum, we used the *Guest Accounts* feature to [add guest users to our user directory](https://docs.microsoft.com/en-us/azure/active-directory/b2b/add-users-administrator#add-guest-users-to-the-directory).
* **Azure Resource Group** (Optional) - All services procured from Azure are known as *resources*. A *resource group* is just a logical container which allows you to manage a set of resources as one entity. For instance, budget limits and permissions can be set on a resource group. In the context of the Practicum, we’ve [created a resource group](https://docs.microsoft.com/en-us/azure/azure-resource-manager/manage-resource-groups-portal#create-resource-groups) named “DA-Practicum” to hold all the services supporting our solution.

## Setup

Here are the detailed steps one has to follow in order to reproduce the solution we put together.

1. [Create an knowledge base](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/how-to/language-knowledge-base) for the English content:
   1. [Create a QnA service in Microsoft Azure](https://portal.azure.com/#create/Microsoft.CognitiveServicesQnAMaker) with the following parameters:
      1. **Name**: Cannabot-QnAService-English
      2. **Subscription**: Free Trial
      3. **Pricing tier**: Fo (3 managed document per month)
      4. **Resource group**: DA-Practicum
      5. **Search pricing tier**: B (15 indexes)
      6. **Search location**: Canada Central
      7. **App name**: Cannabot-QnAService-English
      8. **App service plan**: The App service plan currently defaults to standard(S1) tier. It can be modified by visiting the app service plan resource page once the resource has been created.
      9. **Website location**: Canada Central
      10. **App insights**: Enable
      11. **App insights location**: Canada Central
   2. Connect your QnA service to your knowledge base with the following parameters:
      1. **Microsoft Azure Directory ID**: Default Directory
      2. **Azure subscription name**: Free Trial
      3. **Azure QnA service**: Cannabot-QnAService-English
   3. Name your Knowledge Base
      1. **Name**: Cannabot-English
   4. Populate your Knowledge Base
      1. **Add URL**:
         1. <https://raw.githubusercontent.com/sylvainpaq/cannabis-knights/master/KB/CTLS-Guide-en.xlsx>
         2. <https://raw.githubusercontent.com/sylvainpaq/cannabis-knights/master/KB/Email-FAQ-en.xlsx>
         3. <https://raw.githubusercontent.com/sylvainpaq/cannabis-knights/master/KB/qna_chitchat_friendly_en.xlsx>
      2. **Chit-chat**: none
2. [Publish the Knowledge Base](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/quickstarts/create-publish-knowledge-base#publish-the-knowledge-base)
   1. Once published - and before creating the bot - take down the following parameter from the Postman/Curl HTTP request templates found on the Publish Success page as you’ll need it later.
      1. **QnA Auth Key** from the Authorization statement  
         Ex.: Authorization: EndpointKey [Authorization Key]
3. [Create a bot](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-quickstart?view=azure-bot-service-4.0) with the following parameters:
   1. **Bot name**: cannabot-qnaservice-english-bot
   2. **Subscription**: Free Trial
   3. **Resource group**: DA-Practicum
   4. **Location**: Central Canada
   5. **Pricing Tier**: F0
   6. **App name**: cannabot-qnaservice-english-bot
   7. **SDK language**: C#
   8. **QnA Auth Key**: [the corresponding value from the previous step]
   9. **App servie plan/Location**: Cannabot-QnAService-English
   10. **Application insights**: On
   11. **Application insights location**: Canada Central
   12. **Microsoft App ID and Password**: Auto create App Id and password
4. Repeat the above 3 first steps for the French content.
   1. Basically, simply replace all occurrences of “English” and “en” with “French” and “fr” respectively.
5. [Create a basic web server on Azure](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal) or find an existing one you have access to. In the context of the practicum, we used the [free GitHub Pages offering](https://pages.github.com/) associated with our [project repository](https://github.com/sylvainpaq?tab=repositories) on Github.
6. Download the [Web pages](https://github.com/sylvainpaq/cannabis-knights/tree/master/docs) from the [cannabis-knights project](https://github.com/sylvainpaq/cannabis-knights) on GitHub.
7. Update the Web pages as follow:
   1. index.html
      1. Update the English and French button URLs with your site URL.
   2. index-en.html and index-fr.html
      1. Update all references to the demonstration URL (“https://raw.githubusercontent.com/sylvainpaq/cannabis-knights/master/docs/) with your site URL.
      2. [Get your bot secret key](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-channel-connect-webchat?view=azure-bot-service-4.0#step-1) from azure portal for your English and French bot.
      3. Update the bot secret key (the value of the “s” parameter in the query string below) with your bot secret key in the following line.  
         <iframe title="frame for all interaction with the chatbot" id="catbot" src='[https://webchat.botframework.com/embed/cannabot-qnaservice-english-bot?s=](https://webchat.botframework.com/embed/cannabot-qnaservice-english-bot?s=%5Bbot)qYZEQQjCmU4.0WUA8cB2CCG2kmo9AFwWt8Rg7yaj5q0R0M8F-h8I4kM' style='min-width: 400px; width: 100%; min-height: 500px;'></iframe>
8. Deploy the updated Web pages on your Web server.

## Known Issues

* Answers formatted using [Markdown](https://www.markdownguide.org/) syntax are some time not rendered properly in the Web Chat client. Workaround: Manually edit (ex: add a white space at the end of the answer) all answers that use Markdown syntax directly in QnA Maker and republish the Knowledge Base.

# Operations

## Managing Contributors

In the context of the Practicum, we used the concepts of *Guest Users*, *Groups* and *Roles* to manage the access to the solution underlying Azure resources. Below is what we did:

Steps:

1. [Create Guest User account](https://docs.microsoft.com/en-us/azure/active-directory/b2b/add-users-administrator) for each member of our team.
2. [Add the guest accounts to a basic security group](https://docs.microsoft.com/en-us/azure/active-directory/b2b/add-users-administrator#add-guest-users-to-a-group).
3. [Assign the role “Contributor” to the security group](https://docs.microsoft.com/en-us/azure/role-based-access-control/role-assignments-portal#add-a-role-assignment) created above on all resource groups used by our solution.

In a production context, a similar approach would be used with the difference of using Subscription Account (potentially synchronized with your departement Active Directory service) instead of Guest User accounts.

## Updating the Knowledge Bases

To update the QnA Maker Knowledge Base:

1. Log into the [QnA maker portal](https://www.qnamaker.ai/)
2. Go to “My knowledge bases” and select the appropriate knowledge base to update.
   1. Click on “+ Add QnA pair” to add a new Question and Answer pair;
   2. Click directly on a question or an answer to edit an existing one; or
   3. Click on “+ add alternate phrasing” under a question to add another question to a given answer.

## Exporting a Knowledge Base

There is no backup process associated with the knowledge bases. In order to be safe, we recommend that you export our knowledge bases after major changes. You can easily export a knowledge base directly from the QnA Maker portal. To export a knowledge base:

1. Log into the [QnA maker portal](https://www.qnamaker.ai/)
2. Go to “My knowledge bases” and select the appropriate knowledge base to update.
3. Select the “Settings” menu.
4. Use the “Export knowledge base” to export the current knowledge base to a file.

## Importing a Knowledge Base

Previously exported Knowledge Base file can easily be imported back into your current Knowledge Base. To import a Knowledge base file:

1. Log into the [QnA maker portal](https://www.qnamaker.ai/)
2. Go to “My knowledge bases” and select the appropriate knowledge base to update.
3. Select the “Settings” menu.
4. Use the “Import knowledge base” to import a previously exported knowledge base into the current knowledge.

## Exploring the Bot Analytics

Bot Analytics provide insights on how successful the Cannabot is and how it can be improved. To access the bot knowledge base analytics:

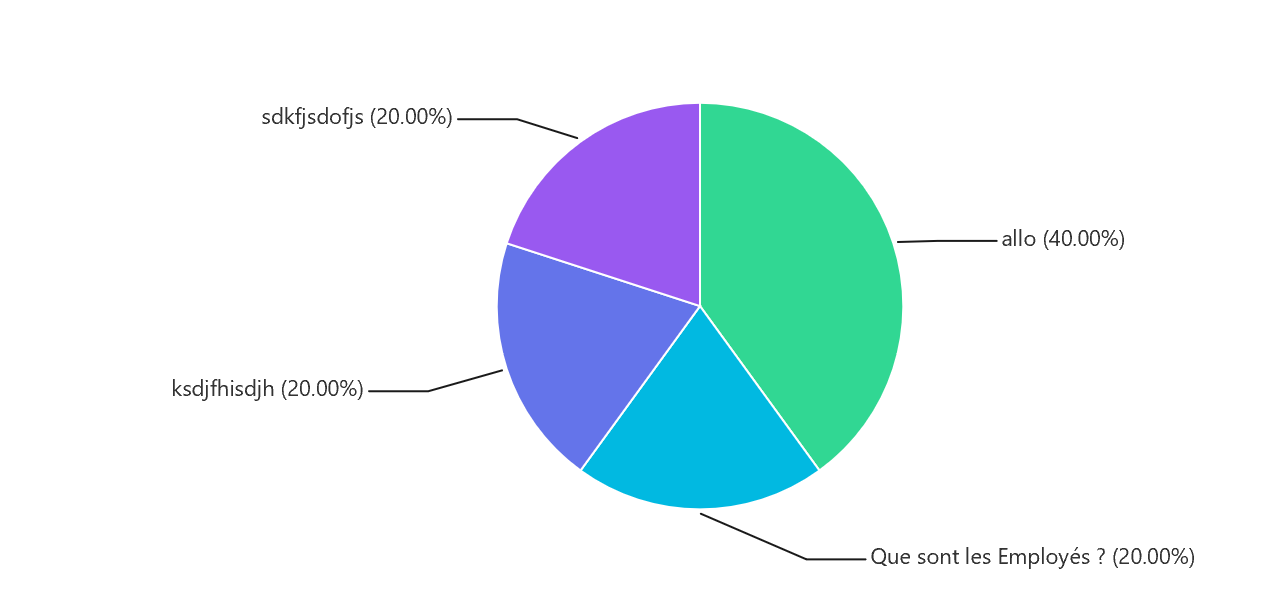
1. Log into the Azure Portal: <https://portal.azure.com>
2. Navigate to the “All resources”.
3. Open the resources of type “Applications Insights” that corresponds to the knowledge base. (If you followed the instructions from the Solution Implementation Setup, this resource is named “Cannabot-QnAService-English-ai”).
4. From the Resource Dashboard, click “Analytics”. (A query window should appear.)
5. Build or paste your query into the editor.
6. Highlight the query text, and click run.

You will find below a list of useful queries.

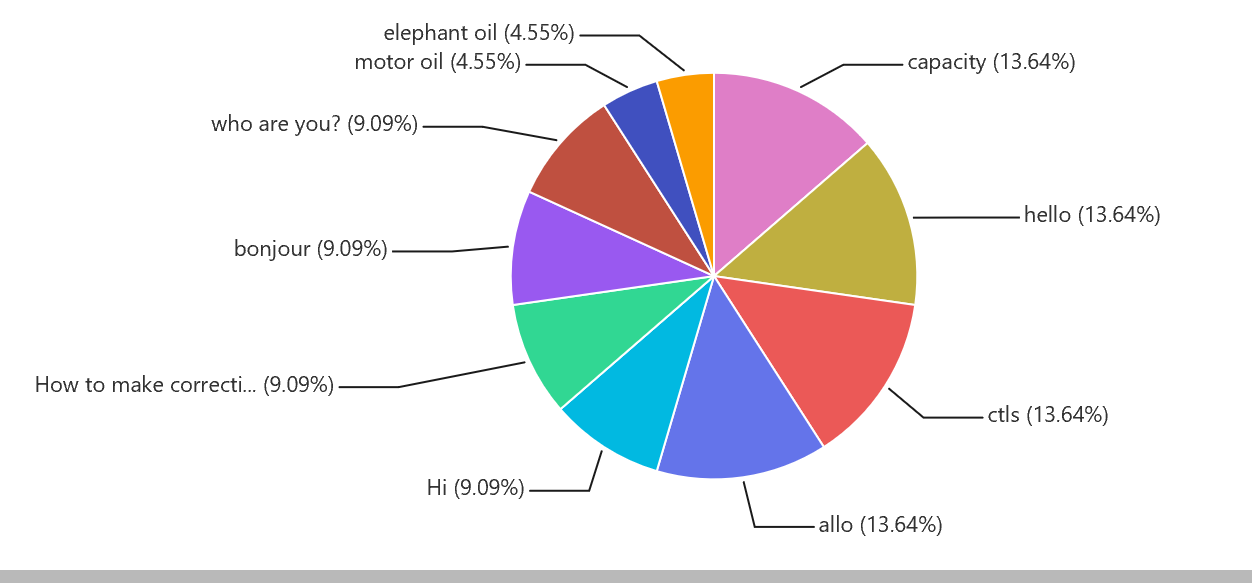
|  |  |
| --- | --- |
| **Purpose** | **Query** |
| Top 10 most asked questions | requests  | where timestamp > datetime(2019-06-01T09:30:00Z)  | where url endswith "generateAnswer"  | project timestamp, id, name, resultCode, duration  | parse kind = regex name with \*"(?i)knowledgebases/"KbId"/generateAnswer"  | join kind= inner (  traces | extend id = operation\_ParentId  ) on id  | where isnotempty(tostring(customDimensions['Question']))  | extend question = tostring(customDimensions['Question'])  | extend answer = tostring(customDimensions['Answer'])  | project KbId, timestamp, resultCode, duration, question, answer  | summarize event\_count=count() by question  | top 10 by event\_count  //| order by event\_count desc  | render piechart |
| Top 10 most unanswered questions | requests  | where timestamp > datetime(2019-06-01T09:30:00Z)  | where url endswith "generateAnswer"  | project timestamp, id, name, resultCode, duration  | parse kind = regex name with \*"(?i)knowledgebases/"KbId"/generateAnswer"  | join kind= inner (  traces | extend id = operation\_ParentId  ) on id  | where isnotempty(tostring(customDimensions['Question']))  | extend question = tostring(customDimensions['Question'])  | extend answer = tostring(customDimensions['Answer'])  | where answer == "No good match found in KB." //This needs to be set to the internal  | project KbId, timestamp, resultCode, duration, question, answer  | summarize event\_count=count() by question  | top 10 by event\_count  | render piechart |
| List of distinct questions asked to the bot | requests  | where timestamp > datetime(2019-06-01T09:30:00Z)  | where url endswith "generateAnswer"  | project timestamp, id, name, resultCode, duration  | parse kind = regex name with \*"(?i)knowledgebases/"KbId"/generateAnswer"  | join kind= inner (  traces | extend id = operation\_ParentId  ) on id  | extend question = tostring(customDimensions['Question'])  | extend answer = tostring(customDimensions['Answer'])  | project KbId, timestamp, resultCode, duration, question, answer  | order by question, answer asc // Ordered by question then answer  | summarize by question , answer // Returns a distinct list of questions and answers |

Below are examples of the graphs produced by the above queries:

**Top 10 most unanswered questions**



**Top 10 most asked questions**



# Usability and Accessibility

## Usability

We conducted interviews with stakeholders including Health Canada employees who use the CTS system from the inside, and we spoke with frontline workers who gave us insights into the kinds of problems and questions licensed producers have with cannabis production and the CTS system.

We sat in on a live demo of the CTS reporting system, given to us by Health Canada employees to help us gain a better understanding of how the reporting system works.

We have demoed our first cut of the Cannabot to the Health Canada folks, as well as a resource representing the LPs (Licence Producers).

We also ran two preliminary user testing sessions, and gained insights on how the Cannabot responds to questions outside of the FAQ’s, including more general inquiries such as questions related to reporting deadlines.

## Accessibility

We started with the Microsoft Azure bot service template, and built it to suit our design criteria and ran it through a Chrome extension called “Lighthouse Report”. Our chatbot received a score of 100%.

We tested the Mac screen reader tool called “voice over” with the chatbot and there is still lots of room for improvement. It did not immediately pick up the chatbot screen, but was able to read out the text in the chat. Going forward, more work will need to be done to ensure screen readers can easily pick up the chatbot application.

# Lessons Learned

* **Bot must be knowledgeable** - Because the current prototype of the chatbot only has the nine FAQ’s loaded into the knowledge base, the people who tested the bot often got “N/A” for a response when they asked any questions unrelated to the FAQ’s. It is important to consider that if this bot would be implemented, it should have enough answers to the more general questions related to the CTS so that users do not get too many “N/A” answers. A possible pitfall that could occur is that if LP’s use the Cannabot for the first time and get too many “N/A” answers, they may not see value in this application and might not use it in the future.
* **Bot must be friendly and concise** - Currently, when the Cannabot sends a response based on the nine FAQ’s, often the answers are quite long. For future iterations, it would be advisable to explore how to go about creating a more user friendly and concise answers to LPs so they can easily get the correct information they need to complete their reports. Some ways to go about this could be consulting with User Experience Designers and carrying out more robust user testing scenarios to ensure users are getting answers that meet their needs, which in turn could create better quality data.
* **Change the Default Answer -** Whenever the bot is unable to find a suitable answer to a question, a default answer (“No QnA Maker answers were found.”) is systematically returned. This is less than ideal as the default answer is 1) not very helpful, and 2) always in English. There is apparently a way to change the Default Answer without doing any coding at the bot service level but it doesn’t work with the Web Chat Client we used. Developing a custom Web Chat client (or reusing the one from CRA) could address issues.
* **Adjust the Bot Confidence Score Threshold -** The Confidence Score Threshold determine whether or not the bot should return a partially matching answer from the knowledge base to the end user. A threshold of 0 will return virtually anything whereas a threshold of 100 will only return perfectly matching answers. In the context of the Practicum, we used a generic Web Chat Bot with a pre-determined Confidence Score threshold of 0. As per the Confidence Score Best Practices suggest, we recommend that the threshold be raised to at least 50 to avoid ridiculous answers from the bot.
* **Chit Chat** - Carefully review the chatbot personality files (English and French) to make sure they match your communication standards.
* **TSV Files** - The English Chit chat file originally came in Tab Separated Value (TSV) format. Working with .tsv file format is cumbersome as these need to be converted by Excel before being open. We recommend working with Excel files directly.

# Possible Extension Points

You will find below a list of suggestions to improve our base solution.

## Enabling Multi-turn Conversation

Multi-turn conversation is interaction where the bot actually responds with a question or a list of options to clarify the user’s intent. In order to enable multi-turn conversation, one needs to customize the Web Chatbot client currently used. See [Use follow-up prompts to create multiple turns of a conversation](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/how-to/multiturn-conversation) for implementation details.

## Follow up with Health Canada

Instead of simply replying something like “I don’t know.” when the bot can’t answer a specific question, it should invite the the user to follow up with Health Canada at [hc.ctls-bi-sscdl-ie.sc@canada.ca](mailto:hc.ctls-bi-sscdl-ie.sc@canada.ca).

# 

# Appendix A - Contacts

## Charlie - CRA Beta Chatbot

In the spring of 2019, the Canada Revenue Agency (CRA) implemented [Charlie: a chatbot to answer questions about the climate action incentive (CAI) payment](http://www.cra-arc.gc.ca/chtbt/chatbot-en.html). In the course of our project, we had the opportunity to talk to the following people about their experience with chatbot development. We recommend that Health Canada further exchange with CRA before implementing their own bot as collaboration and sharing would be preferable to duplication.

* **David Young**, A/Director General - Data & Business Intelligence Directorate. [David.Young2@cra-arc.gc.ca](mailto:David.Young2@cra-arc.gc.ca)

## Digital Academy - Practicum 2019 - Team 1

Also known as the *Cannabis Knights*, we are the team that actually worked on developing this solution. You will find below our contact information. Should you need to, don’t hesitate to contact us for clarification and feedback.

* **Allan Phillips**, Natural Sciences and Engineering Research Council of Canada (NSERC) and participant to the Data stream of the Digital Academy - Premium Stream [allan.phillips@nserc-crsng.gc.ca](mailto:allan.phillips@nserc-crsng.gc.ca)  
  613-697-2634
* **Nidhi Kalia**, Office of the Superintendent of Financial Institutions (OSFI) and participant to the Data stream of the CSPS - Digital Academy - Premium Stream  
  [nidhi.kalia@osfi-bsif.gc.ca](mailto:nidhi.kalia@osfi-bsif.gc.ca)
* **Patrick Messier**,Canada School of Public Service (CSPS) and participant to the Development and Operations (DevOps) stream of the CSPS - Digital Academy - Premium Stream  
  [Patrick.Messier@canada.ca](mailto:Patrick.Messier@canada.ca)  
  613-698-3298
* **Sylvain Pâquet**, Canada Border Services Agency (CBSA) and participant to the Machine Learning and Artificial Intelligence (ML/AI) stream of the CSPS - Digital Academy - Premium Stream  
  [sylvain.paquet@cbsa-asfc.gc.ca](mailto:sylvain.paquet@cbsa-asfc.gc.ca)
* **Ren**[**é**](https://en.wiktionary.org/wiki/%C3%A9)**e Isaac-Saper**, from Canadian Heritage and participant to the Design stream of the CSPS - Digital Academy - Premium Stream   
  [renee.isaac-saper@canada.ca](mailto:renee.isaac-saper@canada.ca)

# Appendix B - Related Resources

* [Cannabot GitHub Repository](https://github.com/sylvainpaq/cannabis-knights)
* [Cannabot Demo - the CTS Assistant](https://sylvainpaq.github.io/cannabis-knights/cannabot-en.html#q) - Please note that the infrastructure supporting the demonstration bots (French and English) uses “free tier” from Azure. As a result, the bot servers will “fall into sleep mode” automatically if not use for few hours. In order to “wake” them up, simply persist in sending messages, even if you get a “couldn’t send retry” message. After a minute or two, the bot will resume working.
* [Best practices of a QnA Maker knowledge base](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/concepts/best-practices)
* [QnA Multi Language Support](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/overview/language-support)
* [How to verify the language setting of a knowledge base index](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/how-to/language-knowledge-base)
* [Colour Review](https://color.review)
* [Government of Canada Digital Standards](https://www.canada.ca/en/government/system/digital-government/government-canada-digital-standards.html)

# 

# Appendix C - Solution Alignment with the Digital Standards

*The following text outlines how the Cannabot meets the Government of Canada Digital Standards.*

Given the tight time frame, it was tough to achieve the **design of the chatbot with users**. We have, however, demonstrated the look and feel to the Health Canada team. In addition, there was interaction within the other teams in the practicum, as well as the instructors and other Solution Leaders. The look and feel of the Cannabot will be improved with future iterations.

Throughout the design of the prototype, the team strived to **iterate and improve frequently**. This was achieved by reframing the problem statement, using an agile methodology, and by obtaining feedback from Health Canada and through other testers.

The team was able to **work in the open by default** by keeping the knowledge base in GitHub.

We have **addressed security and privacy risks** by leveraging the built in security from Microsoft Azure in the cloud.

We considered **building in accessibility from the start** but this is still a work in progress. We explored adding a speaker function to the chatbot so that the text conversations could be read aloud to the user, but have yet to implement this. We tested the chatbot with screen readers and some improvements would need to be made in the future. For visual design, we created graphic options and tested the colour contrast ratios to ensure all text and icons are easy to read.

We have **empowered staff to deliver better services,** by providing a means to potentially reduce the number of support emails that they need to go through. In addition, the staff will be able to augment the knowledge base on their own.

This chat bot will allow the Health Canada staff to **be good data stewards** by helping the producers to enter accurate data.

The design of our chatbot is intended to provide **ethical** service to LPs, and the “learning” aspect of the chatbot is managed by people who will be able to control how the chatbot interacts with users.

The team was able to **collaborate widely** across the the Dev Ops, Data, Design and AI streams. Each member brings to the table a wealth of knowledge in terms of life skills and stories as well as an expertise in each of their domains. In addition, the team shares a willingness to pass on their knowledge and skill sets to other team members and teams.

**Use open standards and solutions:** this standard is not currently applicable to the current prototype of the Cannabot because the software used (Microsoft Azure) is not open source. Currently, the Cannabot can be used on desktop and mobile devices but improvements need to be made in future iterations.