# Part 1

## Explain the functionalities of the chatbot and how they will address the needs of the career advisor as described in the scenario.

### Need:

A career advisor has the need to automate career guidance for students who are near graduation for a degree in computer science who also typically have an awareness of their disciplinary preferences and personal strengths. Five computer science undergraduate job types are required in a Pandorabot environment to help these individual students decide on which career options to consider.

### Solution:

The Career Chat Bot is geared to address this main issue with several functions. First, the Career Chat Bot starts with a greeting to engage the student and provide instructions on how to begin. The first piece of information the bot asks for is if the user is ready to start or not. The Career Chat Bot then asks for the student’s name to make the experience more personalized. After that it asks the student to select a topic that they are interested in. After the student has selected a topic the career bot lists off Computer Science careers that include the selected topic. The student is instructed to select a career option from the list, with additional options of select a different preference, list all computer science career options, or to quit.

## Summarize other outside works or articles describing bot implementation that represent each of the key elements of the bot you created. These other works must have been published in the past **5** years.

* 1. “Building Bots on the Pandorabots Platform”, by Pandorabots (Building Bots on the Pandorabots Platform, n.d.)
     1. This article is part of the Pandorabots documentation which is continuously updated and written in the past 5 years. This article describes bot development basics with Pandorabots, the relationships between the chatbot (Career Chat Bot), botmaster (human; in this case Vera Butler), and client (user; in this case the undergraduate). It also describes how to login, create the bot, user reports, edit the bot, train the bot, view the log review, and deploy the bot. The main use I had from this article was the use of the AIML file and the code block examples. The code block examples for the AIML file were used to build a tree structure for the Career Chat Bot’s functionality.
  2. “Building Bots with Pandorabots”, by discover.bot (discover.bot, 2019)
     1. This article was used to research what other companies have done with chatbots to solve business needs. It was also used to weigh the pros and cons of the Pandorabots software. Lastly it was used as an introduction to how machine learning works in Pandorabots. The information gathered from this article was used to consider the best approach to implementation of the functions included in the Career Chat Bot.
  3. “AIML Tutorial: Creating a context aware multi-functional chatbot”, by Steve Worsick (Worsick, 2018)
     1. One key element to the bot is the fact that it can provide career advice based on the user’s selected preferences. In this article Worsick provides code blocks of categories that include buttons with URLs. I used these examples to include buttons in the AIML file. When the user clicks these buttons, it takes them to an external browser location that provides them with more information about the career.
  4. “How can I create a chatbot that organizes input into selected templates?”, unknown (unknown, 2019)
     1. To personalize the experience and give the bot a more human element I included a function that asks for the user’s name. In order to remember the user’s name, the think tag needs to be used along with he set tag. The name then can be used globally throughout the chat bot to make it seemingly more like the user is talking to a career counselor.
  5. “Dealing with Off Topic Input”, by Steve Worsick (Worsick, Dealing With Off Topic Input, 2019)
     1. One last function the bot needed was to handle off topic input. This article goes into a lot of detail about how to do this. Essentially, there needs to be a default reply to any input from the user and this is handled with a plain star within a pattern of a category. The template of this category replies with “Welcome to CS Career Bot! Please say “begin” to get started.”. This prompts the user to go back to the beginning of the AIML tree.

## Identify **5** or more computing job types that your created bot can recommend based on the interaction with the bot. Provide the generated chatbot code files to support the identified job types.

* 1. Topic: Database Administrator  
     File: dba.aiml
  2. Topic: Software Developer  
     File: sd.aiml
  3. Topic: Computer Network Architect  
     File: networkdamin.aiml
  4. Topic: Information Security Analyst  
     File: infosec.aiml
  5. Topic: Web Developer  
     File: webdev.aiml

## Explain how the chatbot training cases were selected and how the AIML or other programming languages were used to enhance the functionality of the bot. Provide examples of the chatbot functionality (that represent the selected case and languages) at the end of the training process in support of your explanation.

* 1. The chatbot training cases were selected based on the requirements of the career counselors needs.
  2. AIML was used to enhance the functionality of the bot using collecting the users name with think, set, and get tags.
  3. **Figure 1** shows an example of the chat bot collecting the users name with the AIML code <think><set name=”name><star/></set></think>. This line of code is nested within its own AIML <category> and <template> tags. The user is prompted to reply with the exact phrase “my name is \*”. When the user replies with this text the bot can save the name for later use within the conversation. What the AIML <set> tag is doing is saving the name (the \* of the phrase) to a variable called ‘name’ in the preferences file. The AIML <think> tag allows the bot to reference the name variable with an AIML <get> tag from any other categories that are called after the ‘name’ variable is set with the AIML <set> tag.
  4. **Figure 2** shows an example of the use of the AIML <get> tag that is used by the QUIT category of the program. The AIML <think> tag retrieves the name variable from the preferences file. The name variable was set by the AIML <set> tag and saved by the AIML <think> tag.

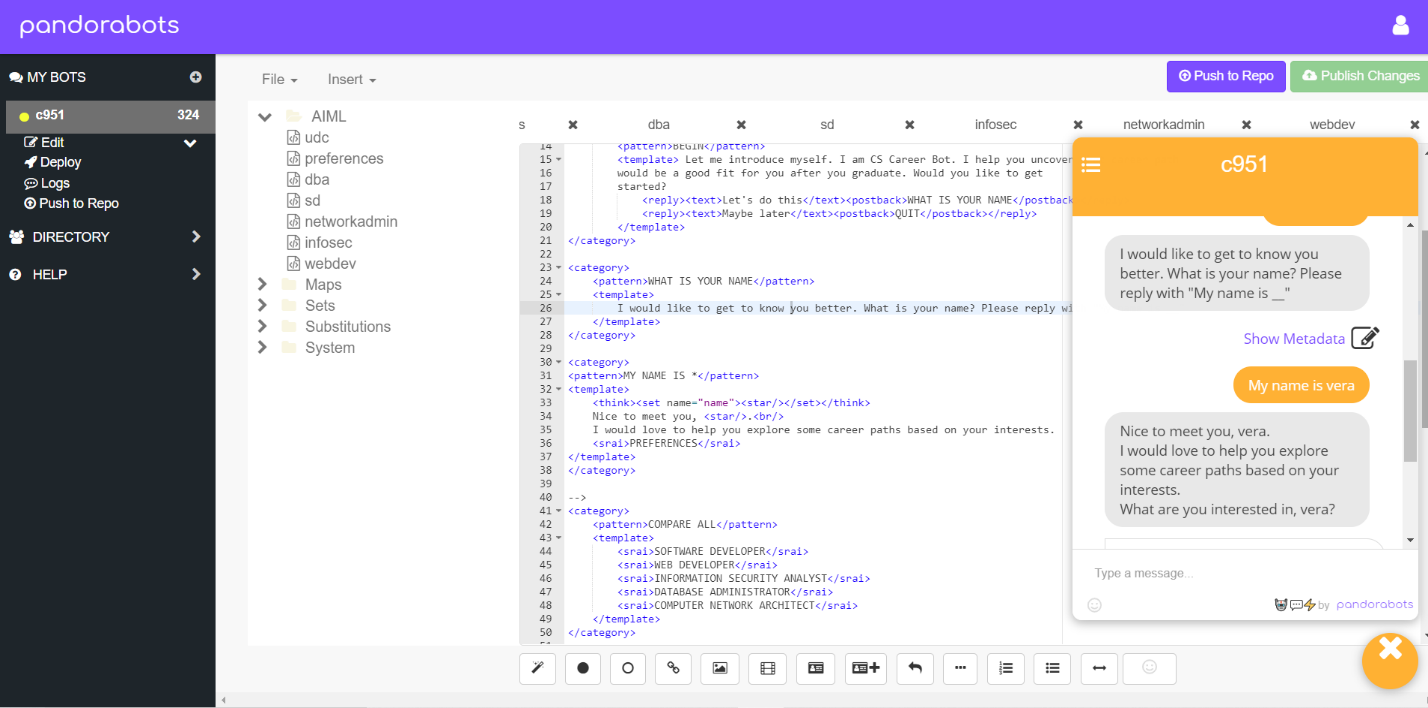


Figure 1 Name Collection with AIML

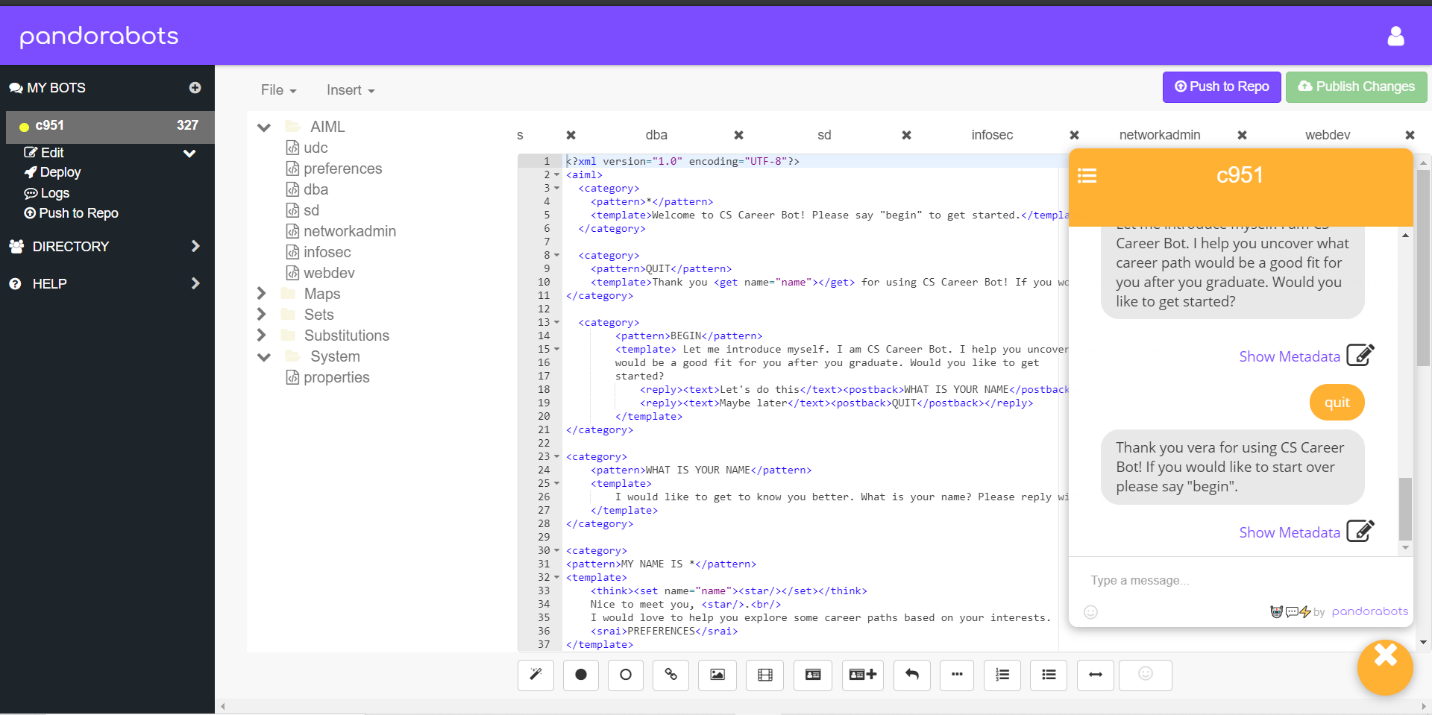
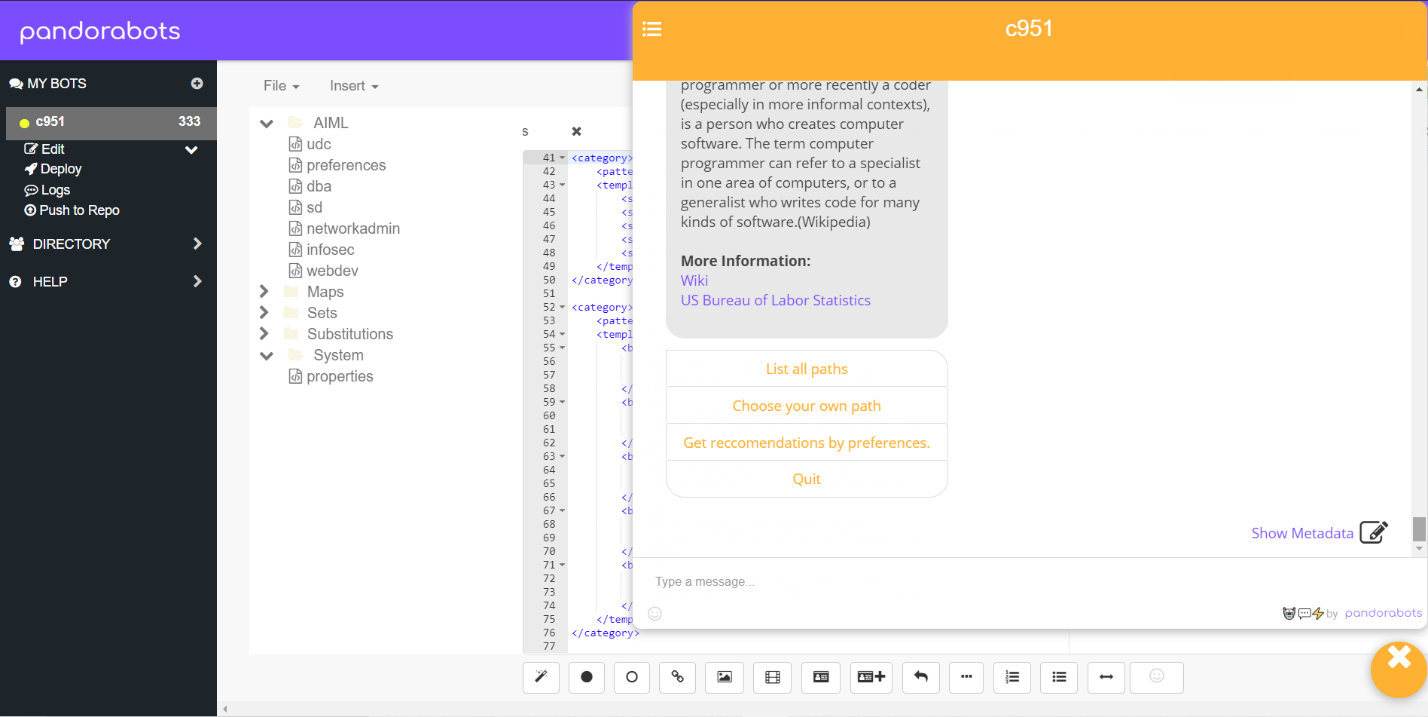
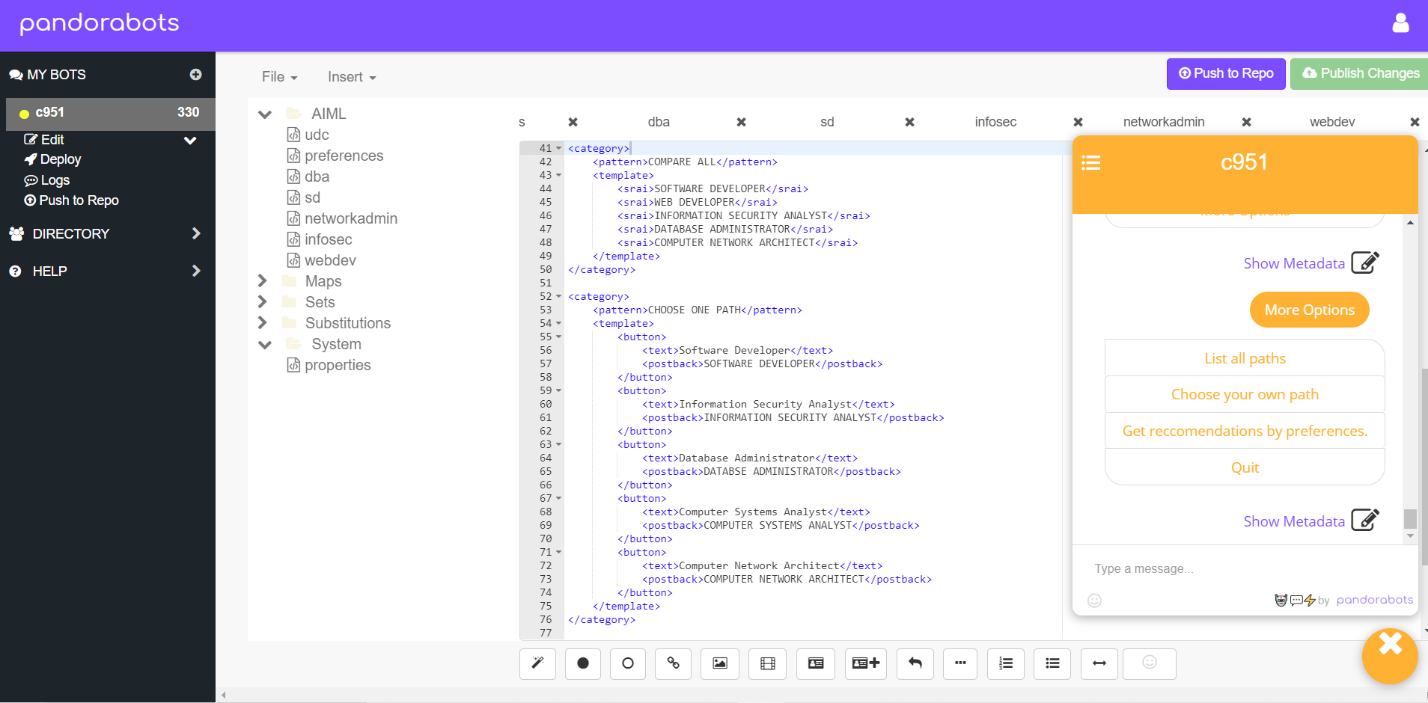
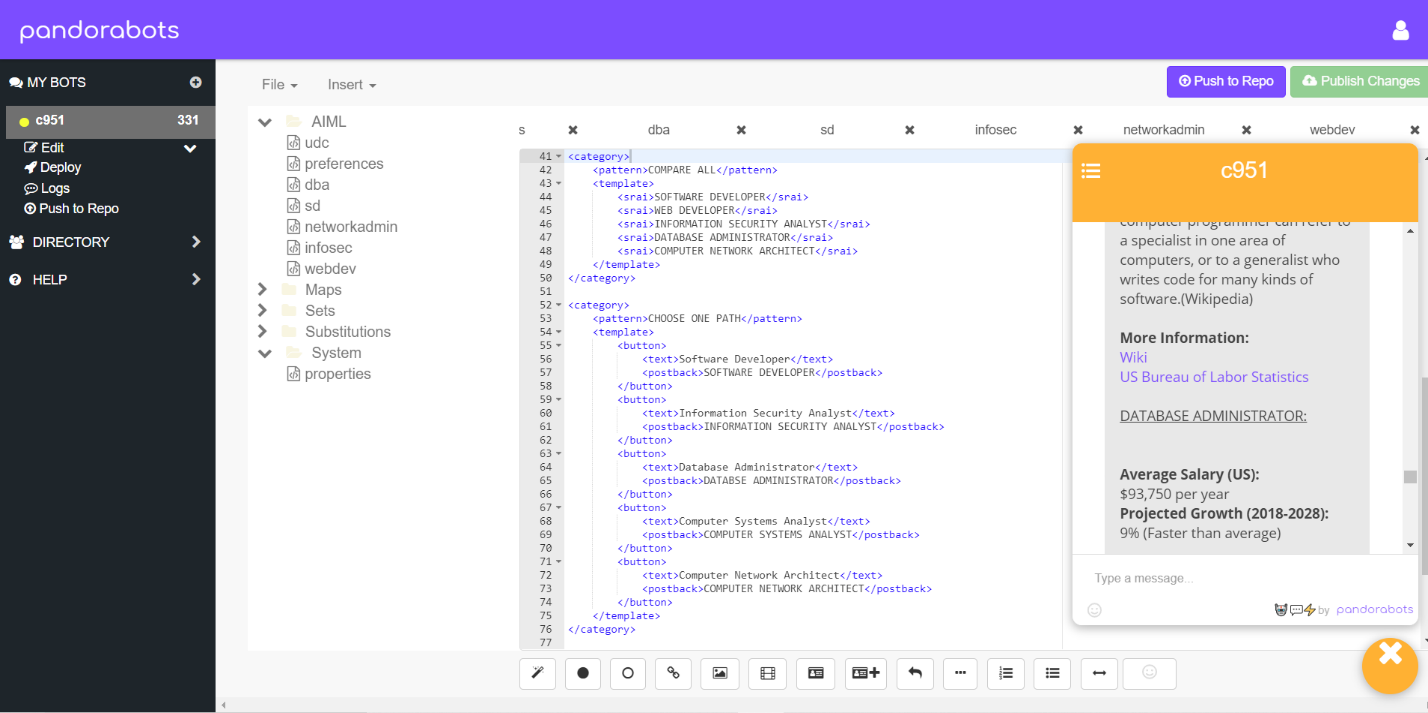
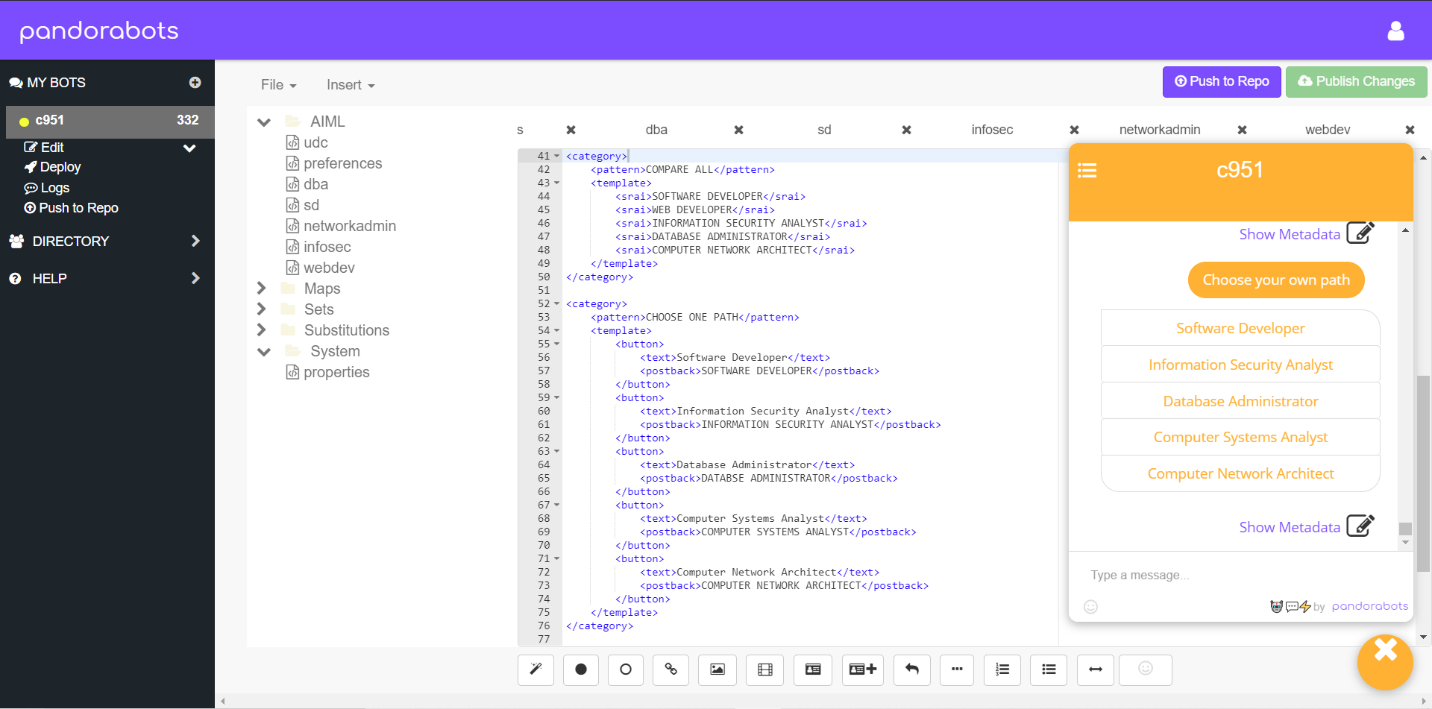
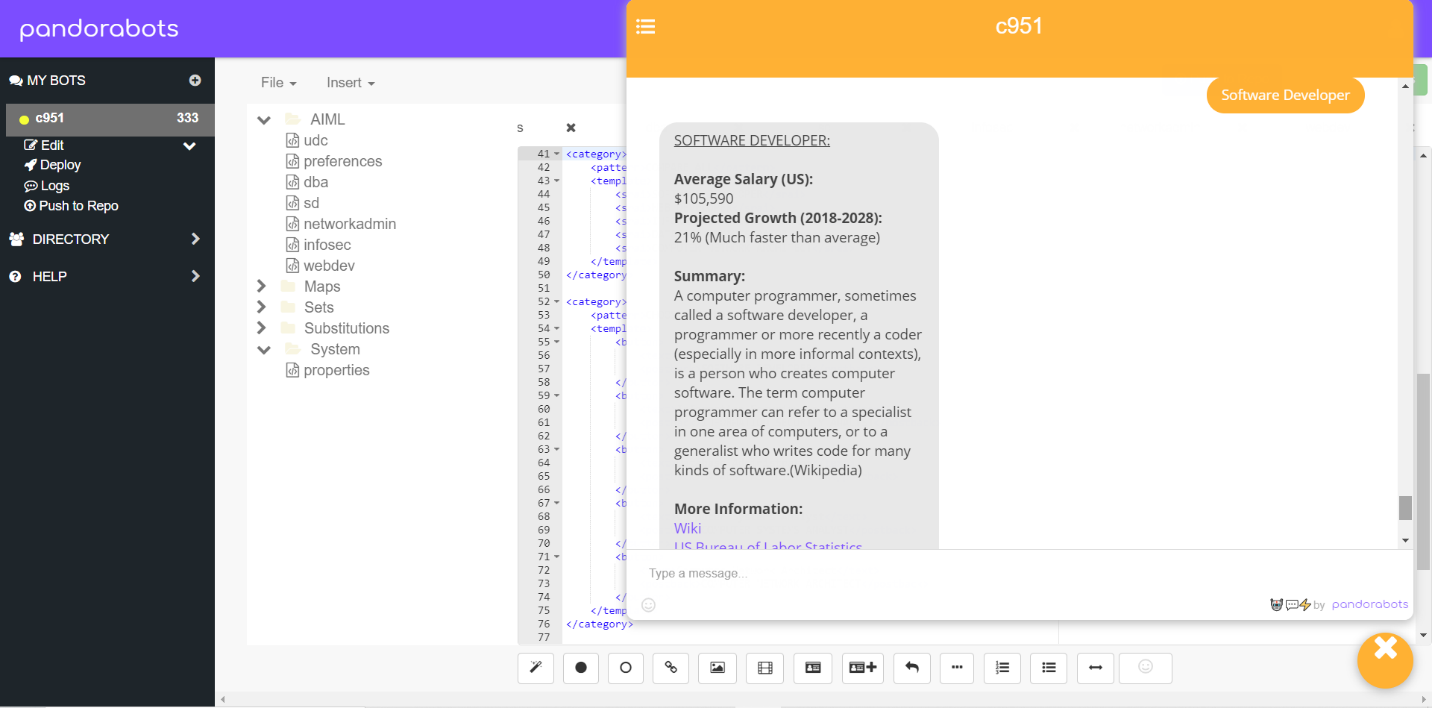


Figure 2 Name on Quit with AIML

## Explain how AI optimization methods were used to optimize the chatbot by providing examples that represent the optimization methods used at the end of the optimization process.

### AI optimization was used by recursively calling categories with <postback> and <srai> tags to provide a loop that the user can continue in to explore careers.­­­­­­­

* 1. 
  2. 
  3. 
  4. 
  5. 

## Create an installation manual for the chatbot.

* 1. Retrieve the files to upload
     1. Step 1: Download the zip file attached to this submission
     2. Step 2: Unzip the file to your desired location
  2. Sign up for an account.
     1. Step 3: got to <https://home.pandorabots.com/home.html>
     2. Step 4: Click Sign Up Free
     3. Step 5: Follow the prompts to complete registration
  3. Create new bot
     1. Step 6: Click the + symbol next to My Bots in the top left corner of your screen
     2. Step 7: Bot Settings
        1. Name the new bot cs-career-bot
        2. Select English for Language
        3. Select Blank Bot for Content
     3. Step 8: Upload Files to Bot Folders
        1. Click the Bot’s Edit menu underneath it’s name
        2. Click the Code Editor menu item
        3. Click the File menu above the folders
        4. Select Upload
        5. Click Select Files
           1. Navigate to the location of the unzipped folder named ‘cs-career-bot’
           2. Navigate to the ‘files’ folder
           3. Select all files in the files folder
           4. Click Open
        6. Click Select Files again
           1. Navigate to the ‘substitutions’ folder within your cs-career-bot folder
           2. Select all files within the substitutions folder
           3. Click Open
        7. Click Select Files again
           1. Navigate to the ‘system’ folder within your cs-career-bot folder
           2. Select all files within the system folder
           3. Click Open
        8. Click Upload
           1. NOTE: Do NOT click the ‘x’ where you have warning messages – we want to over write the existing files
        9. When files are done uploading, click the Done button
  4. Run the bot
     1. Click the yellow and white chat bubble in the bottom right corner of the screen
     2. Type ‘Hi’
     3. Follow the prompts from the bot

# Part Two

## G.  Explain how you measured the effectiveness of the bot and how the bot will be monitored and maintained to improve the final user experience.

### 1. Measuring Effectiveness

The effectiveness of the bot was measured by testing the flow of the program, exploring the career path links, and getting feedback from real people. Currently the chatbot’s effectiveness as a career advisor is basic and can be improved on with future monitoring and maintenance.

One feature that contributes to the current effectiveness of the bot includes a welcoming tone that an effective career advisor would also have. By mimicking a the welcoming tone of an effective human career advisor, the user will be more comfortable to continue engaging with the bot.

Another effective approach to keep the user engaged and allow for career path discovery is the bot moves into the assessment portion of the conversation to find out what topics the user prefers. This feature contributes to the effectiveness because based on the user’s selection it replies with careers that include the type of work the user is interested in.

An effective career advisor would point undergraduate students to outside resources with information about the career paths that may be a good for the undergraduate. The bot behaves as an effective career counselor by returning quality information about careers. The information returned includes salary, expected job market growth, a summary of what the career involves, and links to reliable outside sources.

There is an OPTIONS feature that is built into all career replies by the bot. This feature may be more effective than a human career advisor by allowing the user to bypass the preferences selection with options that include buttons that link to all available career paths, select a specific career, go back to preferences selection, or to quit.

### 2. Monitoring the Bot

The bot will be monitored after deployment. After the bot has been deployed Pandorabots offers a log file of the user interactions. This log file will need to be reviewed regularly to assess for any changes that would improve the bot for future users. The potential improvements should be documented and dispersed to the appropriate stake holders for review and approval.

### 3. Maintaining the Bot for Improving Final User Experience

Maintenance of the bot to improve final user experience will come after monitoring the bot. After the improvements have been approved by stake holders the improvements should be implemented, tested, and reviewed. Maintenance of the bot is an ongoing process and the person responsible for maintaining the bot should then start back at Part 2.G.1 of this document and work though Part 2.G.3.

## 

## H.  Describe the challenges faced during the development process and summarize their resolution.

### Challenge 1: Learning the AIML architecture

The AIML programming language has many built in features with their own special architecture. One of the biggest challenges was ensuring that the architecture was what the programming language was expecting.

### Challenge 2: Providing a Human Like Conversation

The bot’s primary job is to automate an experience that a computer science undergraduate would typically receive from a career counselor at their school. It was important to have a human element to the bot for a more personalized experience for the undergraduate. To emulate this experience, it was decided to include a question about the user’s name for a couple of reasons. Reason one was so the user felt more welcome. Reason two was so the bot could use the user’s name throughout the rest of the conversation.

## I.  Assess the strengths and weaknesses of the bot development environment and explain how they supported or impeded the construction of the chatbot.

### Strengths

The strengths of the bot are the human element of the conversation, the quality of the information provided to the user, and the options list that allows the user to start over, quit, or select a different flow of conversation.

### Weaknesses

One weakness of the bot is that the information provided to the user for each career path is kind of dense and the user must scroll back up to see the top of the returned card. Another weakness is that the bot does not have a welcoming statement without the user typing something first so they may feel confused without initial prompt.

## J.  Provide a Panopto video recording that includes a verbal summary of the capabilities of your bot and an example of human interaction with the bot where it provides meaningful career advice.

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=2537972d-2988-4fc8-a82d-abf6013e8753>

Note: For instructions on how to access and use Panopto, use the "Panopto How-To Videos" web link provided below. To access Panopto's website, navigate to the web link titled "Panopto Access", and then choose to log in using the “WGU” option. If prompted, log in using your WGU student portal credentials, and then it will forward you to Panopto’s website.

Panopto's system, retrieve the URL of the recording from Panopto and copy and paste it into the Links option. Upload the remaining task requirements using the Attachments option.To submit your recording, upload it to the Panopto drop box titled “INTRODUCTION TO ARTIFICIAL INTELLIGENCE – NIP1 Task 1 | C951.” Once the recording has been uploaded and processed in Panopto's system, retrieve the URL of the recording from Panopto and copy and paste it into the Links option. Upload the remaining task requirements using the Attachments option.

## Works Cited

*Building Bots on the Pandorabots Platform*. (n.d.). Retrieved from pandorabots: https://www.pandorabots.com/docs/building-bots/tutorial/

discover.bot. (2019, June 18). *Buidling Bots with Pandorabots*. Retrieved from discover.bot: https://discover.bot/bot-talk/guide-to-bot-buiding-frameworks/pandorabots/

unknown. (2019, April). *How can I create a chatbot that organizes input into selected templates?* Retrieved from Stackoverflow.com: https://stackoverflow.com/questions/55433380/how-can-i-create-a-chatbot-that-organizes-input-into-selected-templates?rq=1

Worsick, S. (2018, August 9). *AIML Tutorial: Creating a context aware multi-fucntional chatbot*. Retrieved from Medium.com: https://medium.com/pandorabots-blog/aiml-tutorial-creating-a-context-aware-multi-functional-chatbot-e5e82c027a6a

Worsick, S. (2019, September 5). *Dealing With Off Topic Input*. Retrieved from Medium.com: https://medium.com/pandorabots-blog/dealing-with-off-topic-input-3857af3e5024