

**GOVERNMENT COLLEGE OF ENGINEERING BARGUR**

**( AUTONOMOUS)**

**PROJECT TITLE: CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT**

**TEAM MEMBERS:**

**CHANDRU D**

**NAVINKUMAR N**

**MANIKANDAN V**

**BHARATH P**

**JAISURYA G**

**PROBLEM STATEMENT:**

 Create a helpful virtual guide using IBM Cloud Watson Assistant. Customize the chatbot to assist users on popular messaging platforms like Facebook Messenger and Slack. Provide useful information, answer FAQs, and offer a friendly conversational experience. Empower users with quick access to information and create meaningful connections through your virtual guide!

**PROBLEM SOLUTION :**

1. Persona Design: Define the chatbot's persona, including its name, tone, and style of communication.

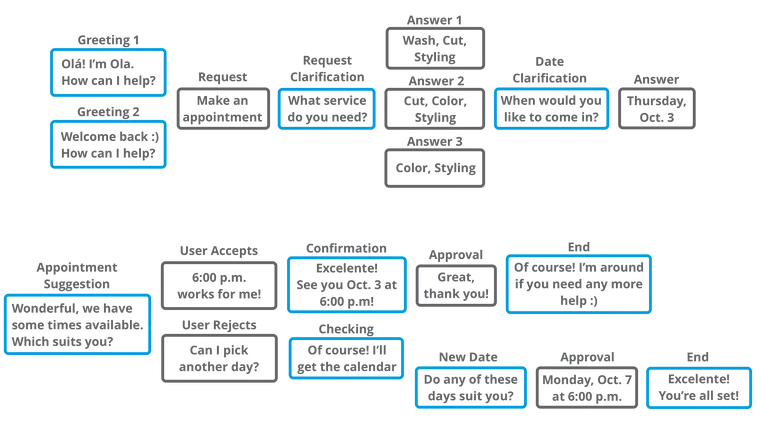
NAME OF THE CHATBOT :**ASSISTIVEBOT**

TONE :**FORMAL**

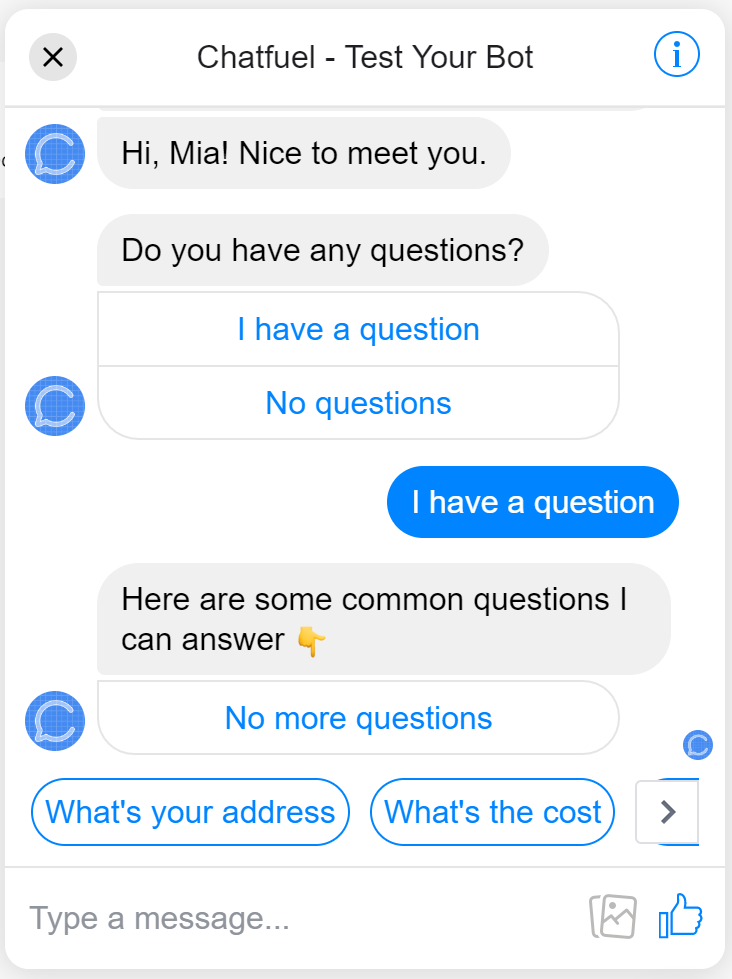
STYLE OF COMMUNICATION :**FRIENDLY**

1. User Scenarios: Identify common user scenarios and FAQs that the chatbot should be able to address.

Example scenario:



Example FAQ:



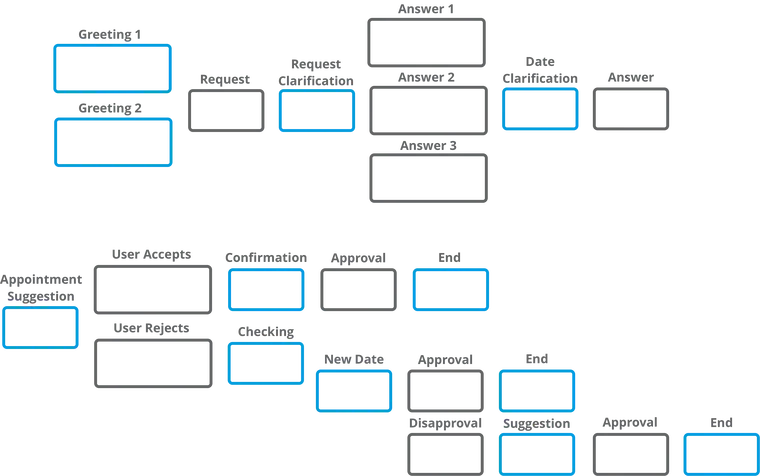
1. Conversation Flow: Design the conversation flow, outlining how the chatbot responds to user queries and prompts.

Write your script in fragments to stay organized and make brainstorming easier. An easy way to do this is by creating a [conversation diagram.](https://miro.medium.com/max/3116/1*Y5DokuM8QuKXaJdz12WtOw.png)

Conversations have elements, and a diagram will help you map out every possibility of what your chatbot could say. The elements you may use in a chatbot conversation are:

* **Greeting:**Used to say hello or start a conversation. Formality is dependent on relationship (return versus new users).
* **Asking:**For engaging or seeking information. Helps keep the conversation going.
* **Informing:**Giving information that is either requested or pertinent to the conversation.
* **Checking:**Testing the user’s understanding. Restating details and information for clarity.
* **Error:**When the chatbot doesn’t understand or fails to fulfill a request.
* **Apologizing:**Politely acknowledging the chatbot’s shortcomings. Should be brief and serve as a bridge to alternative solutions.
* **Suggesting:**Presents the user with relevant actions or options.
* **Conclusion:**A clear end to the conversation.

Visual elements count as well such as GIFs, emoji, pictures or videos.



1. Response Configuration: Configure the chatbot's responses using Watson Assistant's intents, entities, and dialog nodes

**ADD INTENTS:**

For example, in a discussion with the support team, you might gather this set of standard questions that support received from users

* What is the status of the business application? I could not access it.
* How to get access to a business application?
* How to reset my password for a specific application?
* When to renew my workstation?
* How to bring my own device and connect it to enterprise network?

Each of those questions is documented as a frequently asked question in the support team's document repository. Some solutions persist in a relational database in the form of application > problem > solution.

Based on the questions, you can extract these intents:

* Access to a business applications like expense report, AbC.
* Reset password
* Access to supplier on boarding business process
* Bring your own device

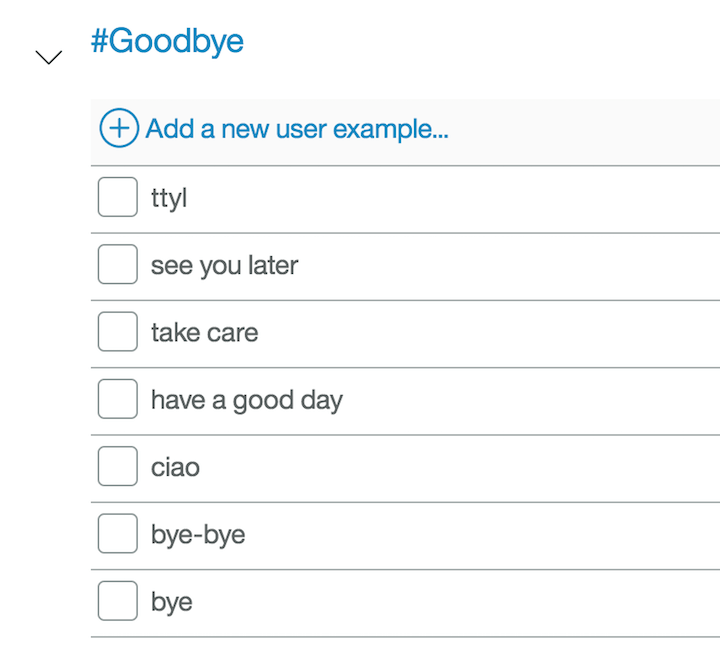
Add those intents to the workspace: From the Build page, click **Intents** and click **Create new**.

For the intent name, type applicationAccess after the number sign (#).

For each intent, add examples to train the conversation for intent recognition.

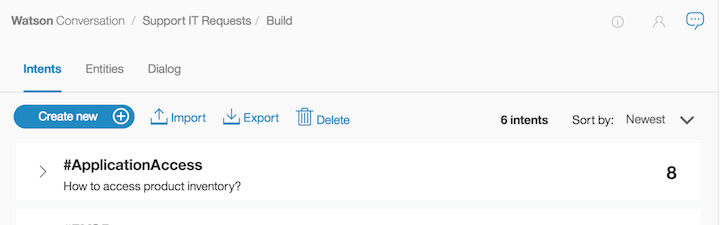
Create the Goodbyes intent and add examples for it. Because many intents can be reused from conversation to conversation implementations, you can define .csv files and import them in the Conversation Tool Intents.

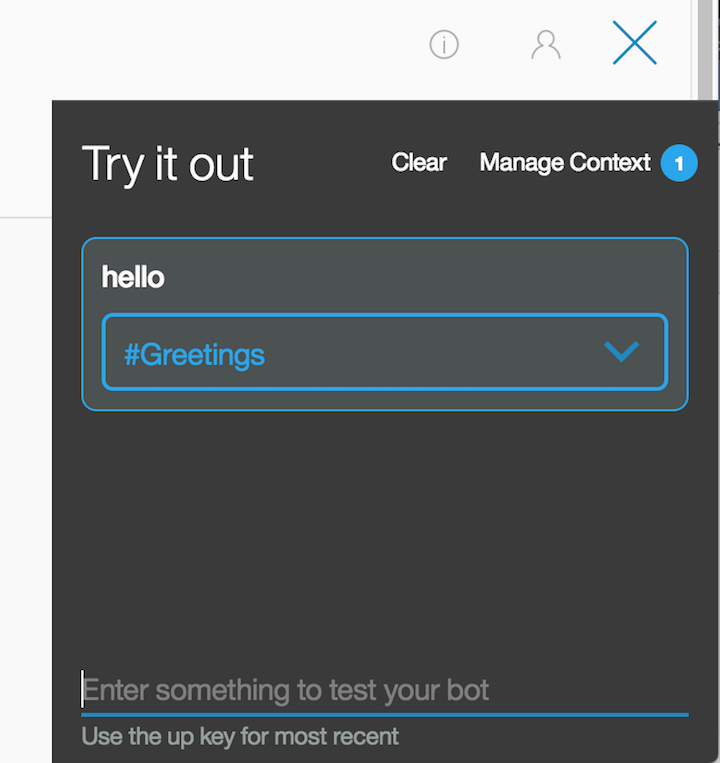
The .csv format is shown in this example with one intent per line:

To get the IT support demonstration intents, click the **Import** link on the Intents page to import the wcs-workspace/ITSupport-Intents.csv file from the GitHub repository.

Next, test your conversations.

1. As soon as you create an intent, you can test it by clicking **Ask Watson** icon in the top, right-hand side of the conversation editor.



Enter one of the examples. You should get the #greetings intent identified by Watson. Enter other greetings to test the #greetings intent.

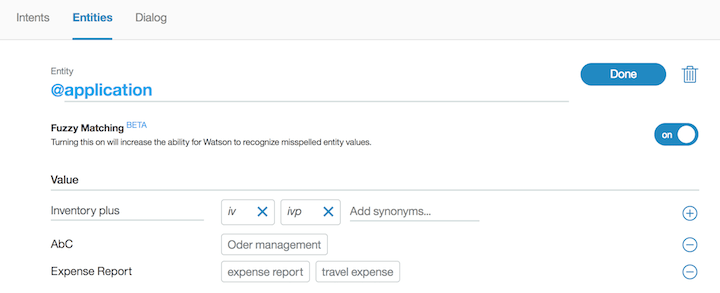
**ADD ENTITIES:**

Click **Entities**. On the Entities page, click **Create new**.

1. Adding values and synonyms to entities helps your chatbot learn important details that your users might mention.

Each entity definition includes a set of specific entity values that can be used to trigger different responses. Each value can have multiple synonyms that define different ways that the same value can be specified in user input.

1. Create entities to represent to the application what the user wants to access.



*Fuzzy logic* is a feature that allows Watson Assistant to accept misspelled words. You can enable this feature at the entity level.

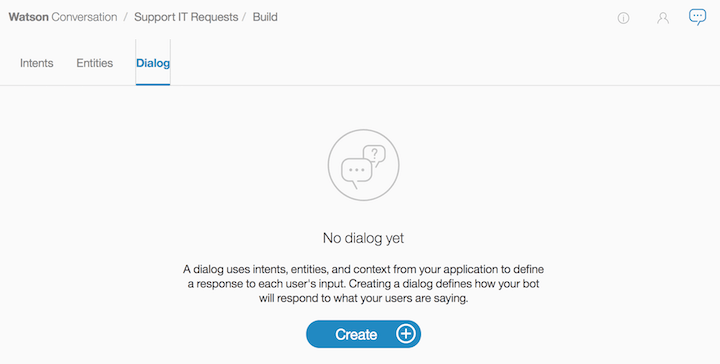
As you did for intents, you can reuse entities' definitions through the export and import capabilities. Import the wcs-workspace/ITSupport-Entities.csv file.

1. If you click the **Ask Watson** icon immediately after you import the entities, the Watson is training message is displayed. Watson Assistant classifies the entities. You can unit-test the entities by entering I want to access application AbC. The following figure shows both the intent and entity (@application:AbC) extracted by Watson Assistant:

You are now ready to create the dialog flow.

**ADD DIALOG FLOW:**

After you specify your intents and entities, you can construct the dialog flow.



A dialog is made up of nodes that define steps in the conversation.

The first node is the standard welcome message. The other node is a catch-all node named "Anything else." Dialog nodes are chained in a tree structure to create an interactive conversation with the user. The evaluation starts at the top, so the welcome node is assessed before the "Anything else" node.

If you click the welcome node, the standard Watson response is "Hello. How can I help you?" To validate how the flow works, you can click the **Ask Watson** icon.

1. Platform Integration: Integrate the chatbot with popular messaging platforms like Facebook Messenger and Slack.

**FACEBOOK INTEGRATION:**

1. Go to the **Integrations** page by clicking the integrations icon (Integrations icon) in the left menu.
2. Click **Add** on the **Facebook Messenger** tile.
3. Click **Confirm**.
4. Follow the instructions that are provided on the screen to complete the integration process.

**Action considerations**

The rich responses that you add to the action are displayed in a Facebook app as expected, with the following exceptions:

* **Connect to live agent**: This response type is ignored.
* **Image**: This response type embeds an image in the response. A title and description are not displayed before the image, whether or not you specify them.
* **Option**: This response type shows a list of options that the user can choose from.
  + A description is not displayed, whether you specify one or not.
  + After a user clicks one of the buttons, the button choices disappear and are replaced by the user input that is generated by the user's choice. If the assistant or the user enters new input, then the button-generated input disappears. Therefore, if you include multiple response types in a single response, position the option response type last. Otherwise, content from subsequent responses, such as text from a text response type, will replace the button-generated text.
  + The title is automatically taken from the text of the relevant step of the action where options are listed.

**SLACK INTEGRATION:**

1. Go to the **Integrations** page by clicking the integrations icon (Integrations icon) in the left menu.
2. Click **Add** on the **Slack** tile.
3. Click **Confirm**.
4. You need to have a Slack app to connect to.

If you don’t have a Slack app, create one now. See [Starting with Slack apps](https://api.slack.com/start).

1. Go to the [Your Apps](https://api.slack.com/apps) page on the Slack website, and then click the app you want to use.
2. From the settings page for your Slack app, open the **App Home** page.
3. Add access scopes for your Slack app.
4. Assign bot token scopes to your Slack app. At a minimum, apply the following scopes:
   * app\_mentions:read
   * chat:write
   * im:history
   * im:read
   * im:write
5. Click *Install App to Workspace*, and then allow the installation when prompted.

If you are editing scopes for an existing application, reinstall it.

1. From the Slack settings App Home page, enable the *Always Show My Bot As Online* setting.
2. Go to the *OAuth and Permissions* page in Slack, copy the *Bot User OAuth Access Token*.
3. From the Watson Assistant Slack integration configuration page, paste the token that you copied in the previous step into both the **OAuth access token** and **Bot user OAuth access token** fields.
4. On the Slack app settings page, go to the *Basic Information* page, and then find the *App Credentials* section. Copy the app credential verification token.
5. From the Watson Assistant Slack integration configuration page, paste the verification token that you copied in the previous step into the **Verification token** field.
6. Click **Generate request URL**, and then copy the generated request URL.
7. Return to the Slack app settings page. Open the *Event Subscriptions* page, and then turn on *Enable Events*. Paste the request URL that you copied in the previous step into the field.
8. On the *Event Subscriptions* page in Slack, find the *Subscribe to Bot Events* section. Click *Add Bot User Event*, and then select the event types you want to subscribe to. You must select at least one of the following types:
   * message.im: Listens for message events that are posted in a direct message channel.
   * app\_mention: Listens for only message events that mention your app or bot.
9. Click *Save Changes*.
10. Optional: To add support for showing buttons, menus, and disambiguation options in the Slack app, go to the *Interactive Components* tab and enable the feature. Paste your request URL in the provided text entry field, and then click *Enable Interactive Components*.
11. User Experience: Ensure a seamless and user-friendly experience, with clear prompts and informative responses.

Will conclude after the completion of ASSISTIVEBOT.