



ChatBot

User Guide

Table of Contents

- **Introduction** 1
- **Target Audience** 1
- **Access to the Platform** 2
- **Chatbot Scenario Setup** 3
 - **Create a New Automated Conversation** 3
 - **Build the Scenario Flow Map** 4
 - User Input Unit 5
 - Bot Reply Unit 7
 - Fallback Responses Unit 9
 - Go To Unit 10
 - Enter Value Unit 11
 - API Unit 12
- **Chatbot Editor Tools** 14
 - **Filter Option** 14
 - Variable 14
 - Condition Type 15
 - Logical Operators 16
 - **Sidebar Support Buttons** 17
 - **Keywords and Reference Functional Commands** 19

Introduction

This guide aims to provide a comprehensive understanding of the chatbot service offered by Taqnyat and to assist users in effectively building and managing chatbots through a complete set of tools designed to simplify the creation and management of intelligent, automated conversations.

Target Audience

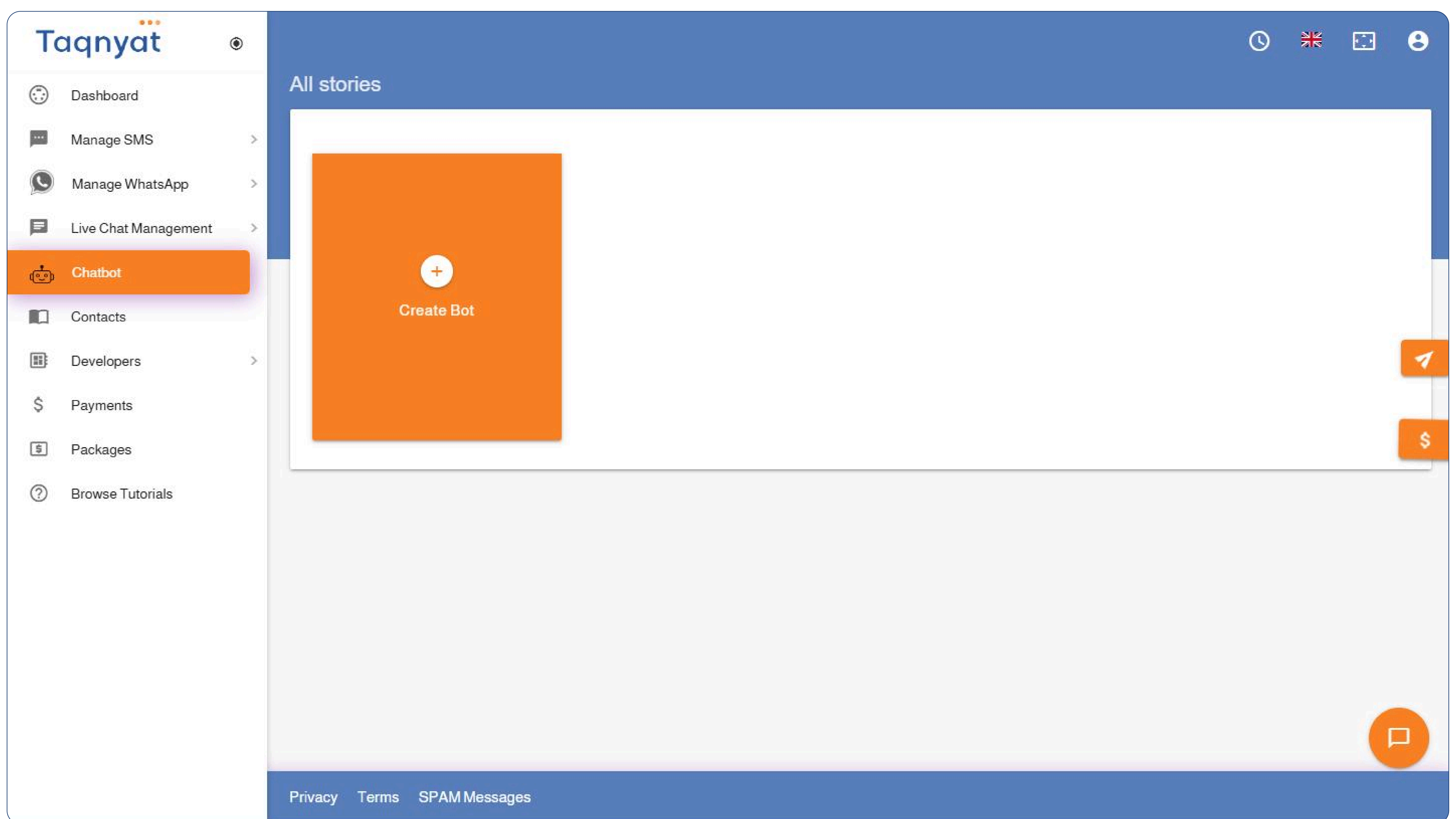


**Supervisors and
Team Leads**

Accessing the Platform

To begin using the chatbot builder, follow these initial steps:

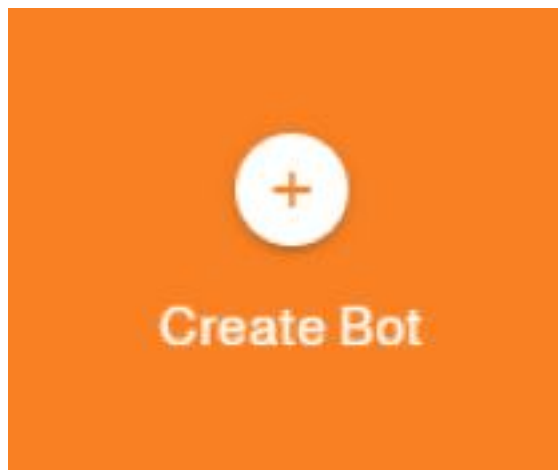
- 1- Log in to the Taqnyat Portal via the link: portal.taqnyat.sa
- 2- From the sidebar menu, select the Chatbot tool.
- 3- Once selected, you will be directed to an interface where you can view existing chatbot scenarios, edit them, or create a new automated conversation



Preparing the chatbot script

Create a New Automated Conversation

1. Customizable Conversation Settings



When you click on “Create Automated Conversation,” a screen will appear asking you to enter a set of chatbot details, which include the following:

Story Name & Description:

The story name is the internal name assigned to distinguish the chatbot scenario within the client’s account on the Taqnyat platform. It will not be visible to end users

The story description is a brief explanation that clarifies the purpose of the scenario and helps team members understand its function more quickly

Conversation Timeout:

This is the time period (in minutes) during which the user is considered inactive. If this period is exceeded without any interaction, the session is automatically reset. If the user sends a message afterward, the conversation restarts according to the configured scenario.

Create new story

• Story name

Story description

• Conversation timeout

minute

• msg timeout (lang=ar)

msg timeout (lang=en)

Cancel

Add

Conversation End Message:

This is the automatic message sent to inform the user that the session has ended after the previously set timeout. This message can be written in both Arabic and English, and it will be sent in the language selected for the conversation

2- After filling in all the fields, click the “Add” button to complete the creation of the scenario. The scenario will then appear in the Chatbot Stories list, where you can access and edit it at any time

Preparing the chatbot script

Building the Scenario Flow Map

1- Click on the “New Conversation” box in the Stories interface to be redirected to an interactive interface for customizing the chatbot and designing conversation paths using a set of basic units, which are:

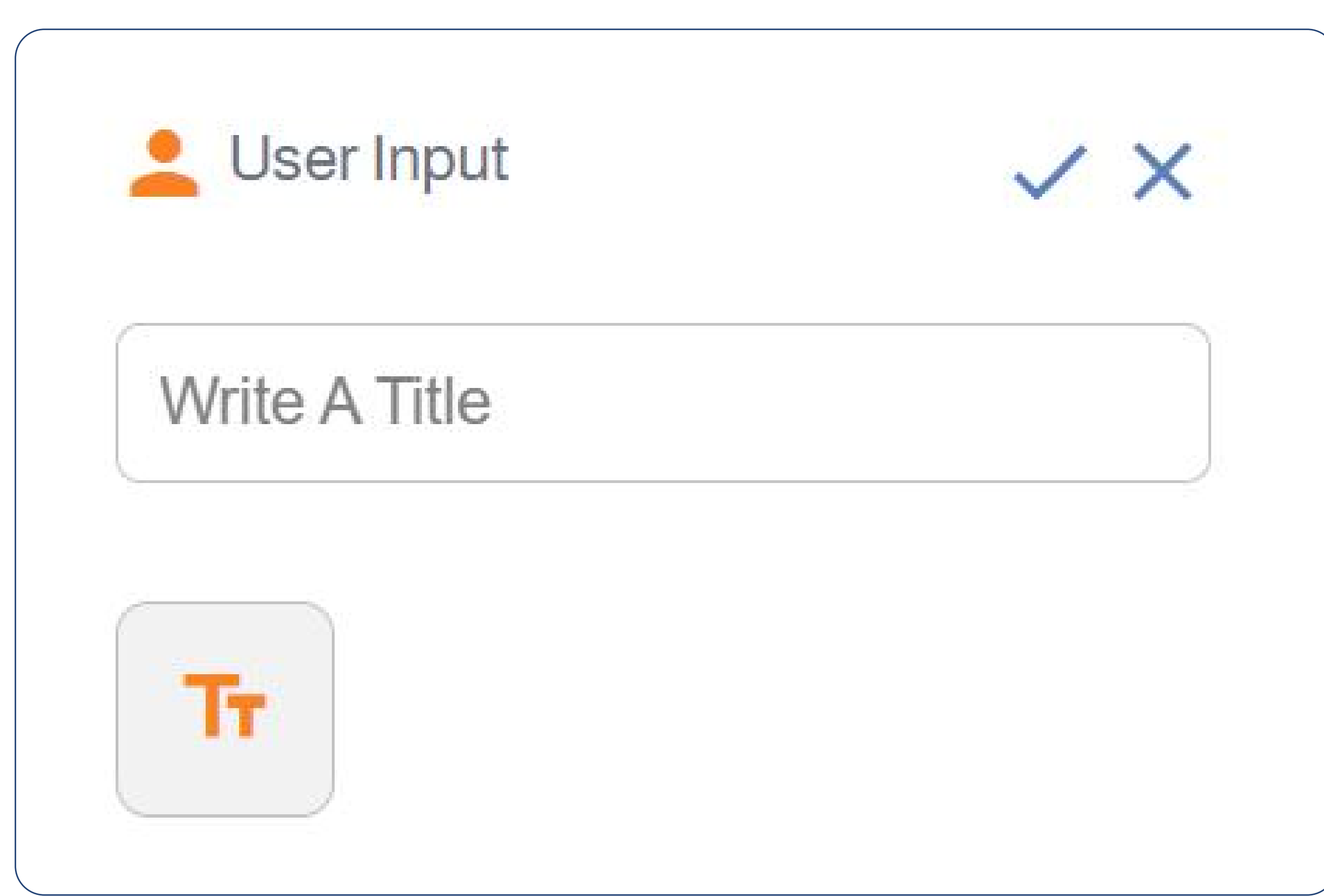


Preparing the chatbot script

Building the Scenario Flow Map – User Input Unit

Adding Inputs to the User Input Unit

The User Input unit is used to determine the chatbot's path based on what the user types. This unit allows linking multiple types and sets of user inputs to a single, specific bot response by filling in the following fields:

A screenshot of the 'User Input' unit configuration interface. It features a title field with the placeholder text 'Write A Title'. Below the title field is a 'Text' input type icon, represented by a blue rounded rectangle with the letters 'Tt' in white. The interface also includes a small orange person icon and a 'User Input' label at the top left, and a checkmark and an 'X' icon at the top right.

A- Title Field in the User Input Unit:

This field is used to assign an internal name to the unit, helping the scenario administrator easily identify its purpose while organizing the different stages of the conversation

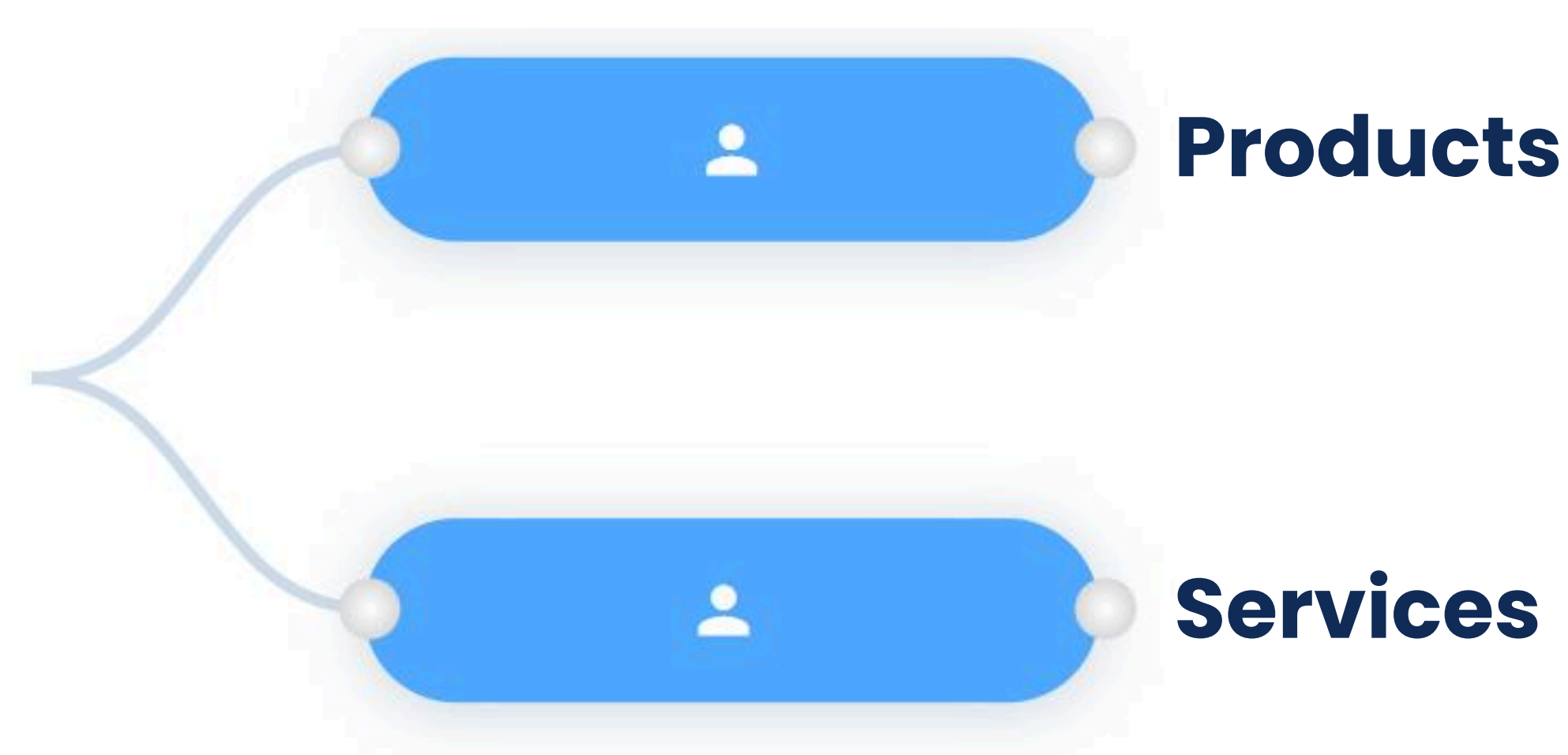


B-Text Input

This is the default input type that allows the user to enter text or a specific value, which can later be used within the conversation path or saved as data. A keyword can also be entered to determine the path the conversation scenario will follow

Example of Keywords:

- If the user types "**Products**," the chatbot will move to the path designated for products.
- If the user types "**Services**," it will go to the services path



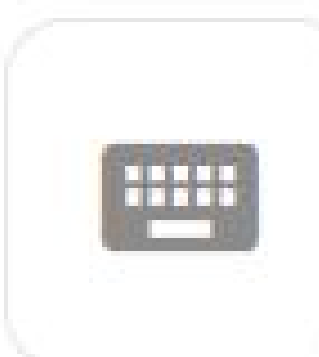
The previous image clearly shows how multiple inputs can lead to different paths within the same unit, based on predefined keywords.

Preparing the chatbot script

Building the Scenario Flow Map – User Input Unit

C– Additional Options

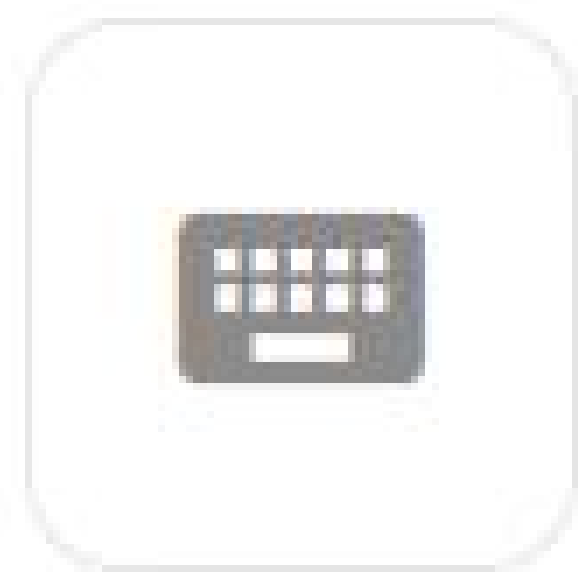






Enter Key

The User Input unit provides a set of additional options that help customize the unit’s behavior and handle inputs more accurately, which are:



Option to save any user input, no matter its content or format.



Deletes the current input configuration from the unit.



A filter used to define conditions for accepting or rejecting input based on specific logic, and it includes:

- Variable: the data you want to check
- Type: The condition used to compare the values
- Value: the value you want to match against
- Logical Operators (and / or): used when combining multiple conditions

variable

type

value

Logical Operators

getChatStatus

is Equal

and

Save

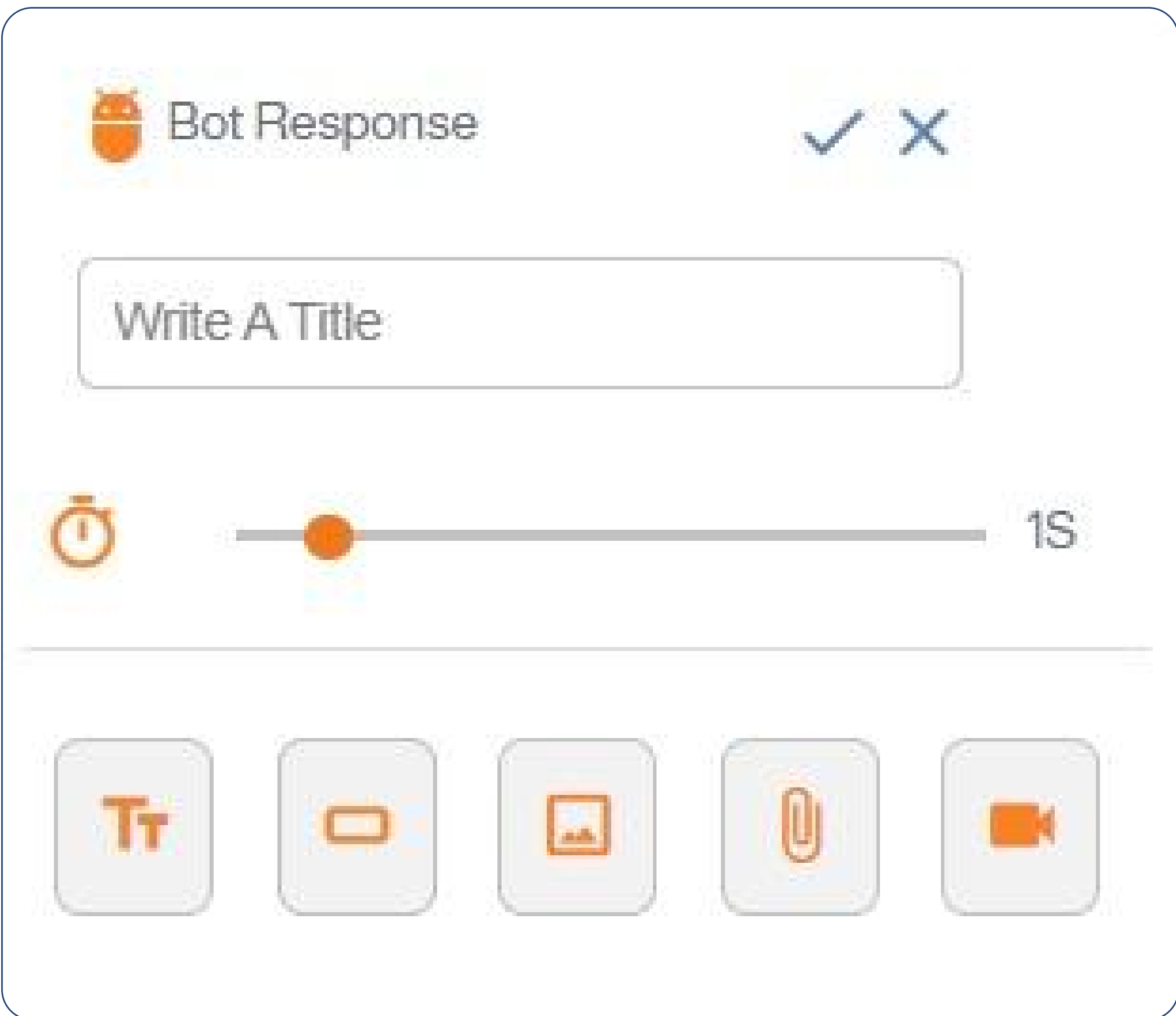
Cancel

Preparing the chatbot script

Building the Scenario Flow Map – Bot Reply Unit

This node is used to send various types of messages to the end user, allowing for rich and engaging communication. The supported response types include:

 **Text Messages:** Used to send simple text replies that provide the user with information or instructions



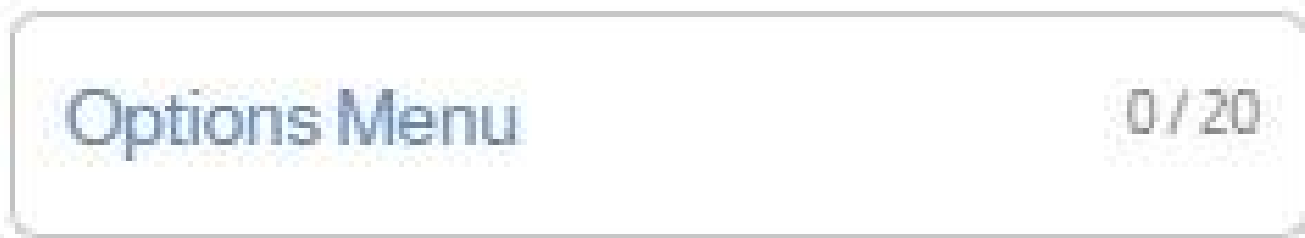
Bot Response

Write A Title

1S

Text Messages icon, Interactive Buttons icon, Image icon, Attachment icon, Video icon

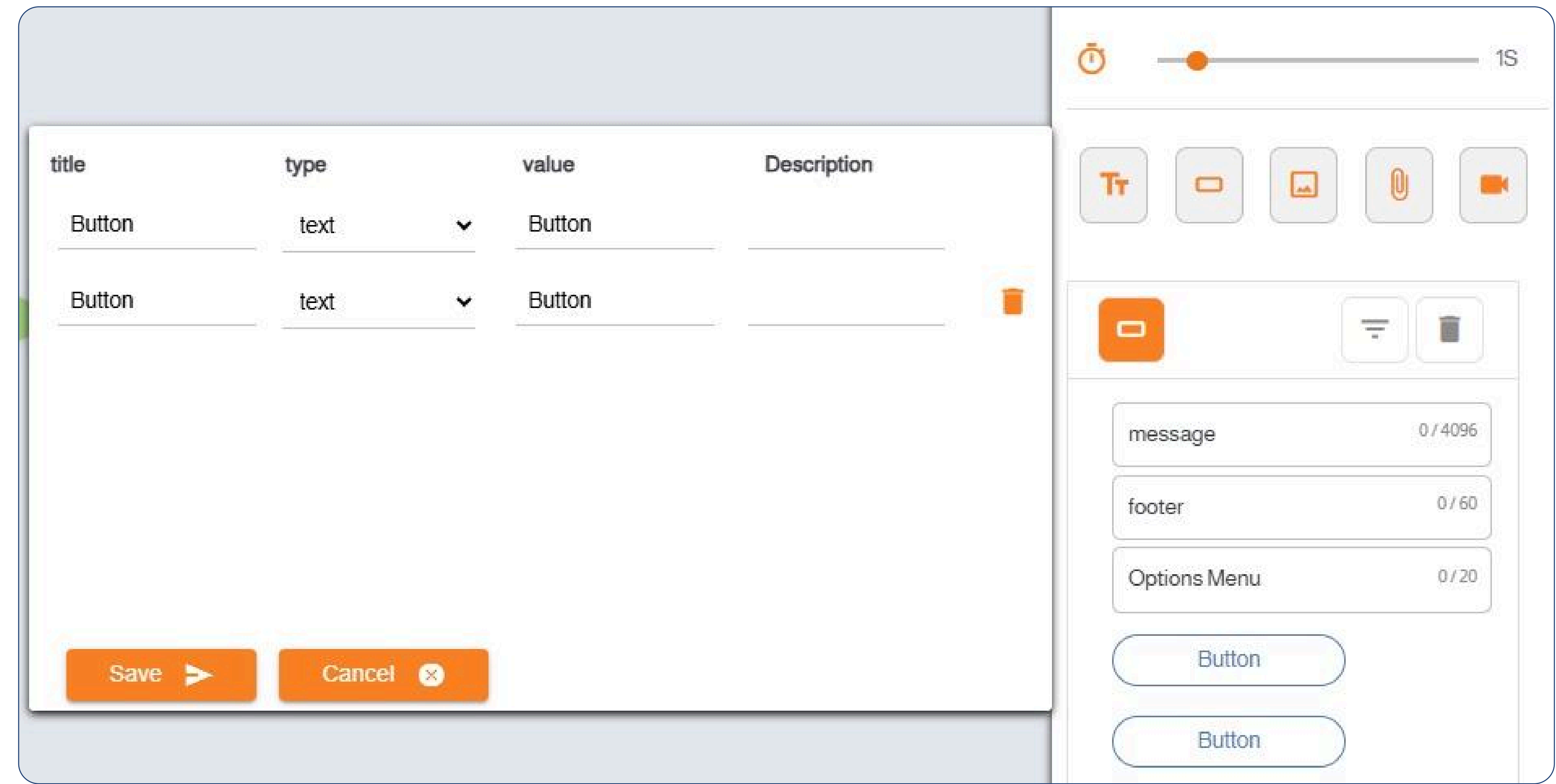
 **Interactive Buttons:**The chatbot sends a message containing text with three response options. If the number of options exceeds three, the system automatically converts them into a list, which can contain up to 10 items. The list name can also be edited from the designated field



Options Menu 0 / 20

Each interactive button message can include:

- Message: The main text shown to the user.
- Footer: An optional line of smaller text displayed below the message.
- Buttons: Up to three quick reply options
- Button Types:
 - **Text** : Sends a specific reply.
 - **URL** : Opens a link



title	type	value	Description
Button	text	Button	
Button	text	Button	

Save Cancel

1S

Text Messages icon, Interactive Buttons icon, Image icon, Attachment icon, Video icon

message 0 / 4096

footer 0 / 60

Options Menu 0 / 20

Button

Button

Preparing the chatbot script

Building the Scenario Flow Map – Bot Reply Unit

Media Messages:

Images, videos, or attached files can be sent to users. These files can either be hosted via publicly accessible links or uploaded directly from the user's device



Image: To add an image, click the edit icon within the Image unit, then paste the image link into the designated field, or upload it from your device. A preview of the image will appear within a frame containing an image icon and a (+) sign.



Files: To add a file, click the edit icon within the File unit, then paste the file link into the designated field, or upload it from your device.



Videos: To add a video, click the edit icon in the Video unit, then enter the video link in the designated field, or upload it from your device. A preview of the video will appear within the unit, complete with a play button and control elements.

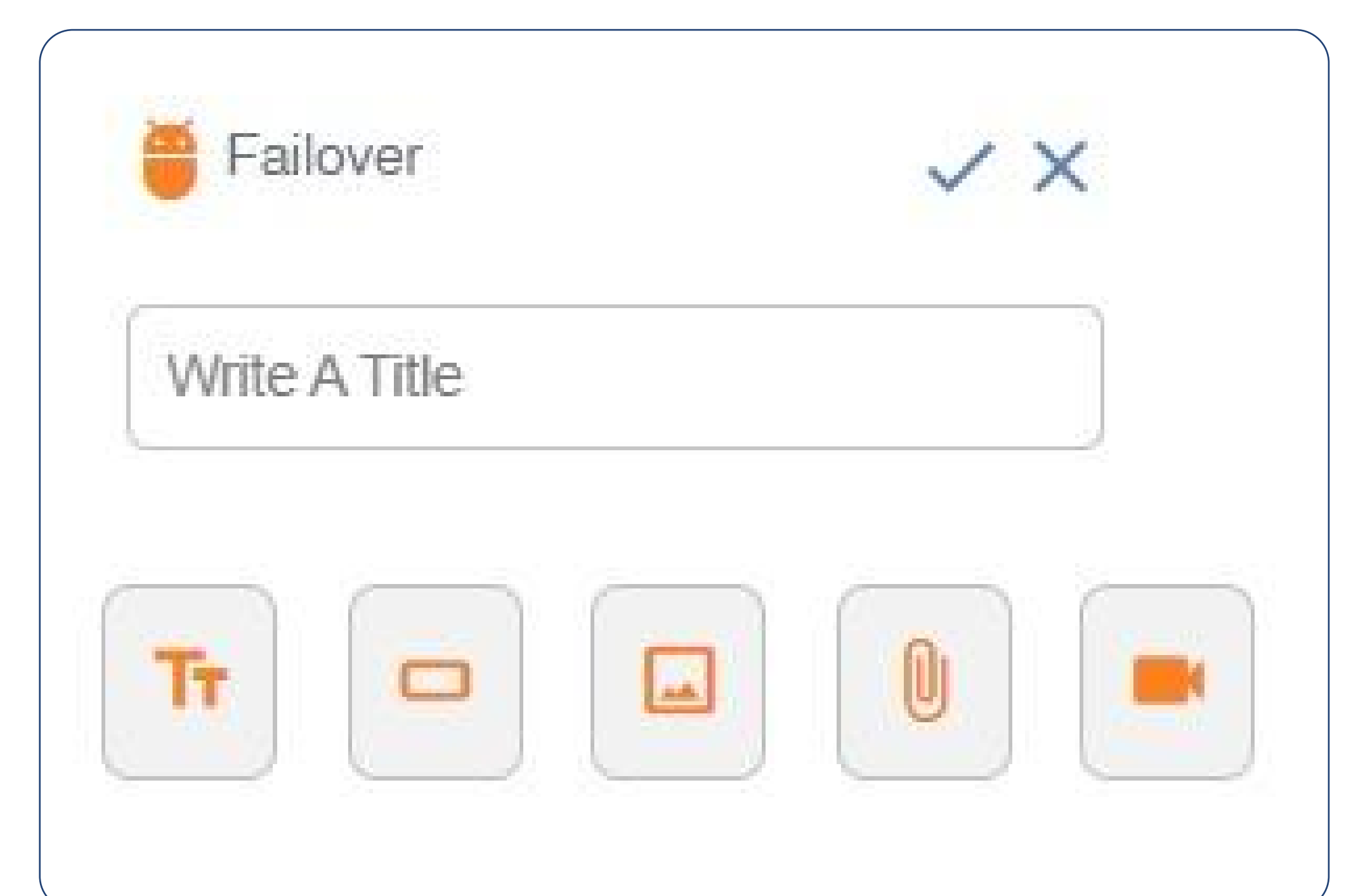
Preparing the chatbot script

Building the Scenario Flow Map – Fallback Responses Unit

This unit is used when the chatbot is unable to match the user's input with any of the predefined conditions. In this case, the user is automatically redirected to a fallback path by adding a "Go To" unit after this unit. A specific message can also be sent to the user before the "Go To" action through the following icons:



Text Messages: Used to send simple text replies that provide the user with information or instructions



Interactive Buttons: Used to send a message accompanied by three response options. If the number of options exceeds three, the system automatically converts them into a list, which can contain up to 10 items. The list name can also be edited from the "Options List" field.



Image: To add an image, click the edit icon within the Image unit, then paste the image link into the designated field, or upload it from your device. A preview of the image will appear within a frame containing an image icon and a (+) sign.



Files: To add a file, click the edit icon within the File unit, then paste the file link into the designated field, or upload it from your device.



Videos: To add a video, click the edit icon in the Video unit, then enter the video link in the designated field, or upload it from your device. A preview of the video will appear within the unit, complete with a play button and control elements.

Preparing the chatbot script

Building the Scenario Flow Map – Go To Unit

The "Go To" module is used to redirect the user to a specific point within the chatbot scenario based on user input, scenario conditions, or by reusing existing predefined paths

Example: If the user types "Request" and no condition is defined to handle this input in the current path, the "Go To" module can redirect the user to a predefined path, such as a fallback path or service-related responses.

How It Works:

When selecting the "Go To" unit, a menu will appear on the left side of the screen containing a dropdown list of all the preconfigured units

- If the preconfigured units have not been named by the user, they can be identified through the system-generated ID. If the ID does not appear, click the Sync icon to display it within the unit.
- The unit name cannot be entered manually at this stage, as the selection of units that the scenario will navigate to must be made only from the available list.
- New units cannot be created directly from the "Go To" unit interface.



Preparing the chatbot script

Building the Scenario Flow Map – Input Value Unit

The Input Value module is used to define variables and store data collected from the user during the conversation, allowing it to be used later within the same scenario, sent to external systems, or used for reporting. These values help to:

- Create personalized chatbot experiences that adapt to each user's needs.
- Track user details such as name, phone number, interaction intent, or interest in a specific product.
- Enable integration with backend systems, such as Customer Relationship Management (CRM) systems or external APIs

How to use this data:

- Gain insights into user behavior and preferences.
- Identify high-intent customers based on the captured data.
- Make informed decisions regarding follow-ups, marketing campaigns, and support priorities.

When selecting the Input Value module, a menu will appear on the left side of the screen containing the following icons:



To assign a predefined value to a specific variable within the scenario.



To save the actual input entered by the user in the variable for later use in the same session or for analytical purposes.

Note: All values saved through the "Input Value" module can later be retrieved from the "WhatsApp Leads" tab in the sidebar menu of the admin account on the livechat.taqnyat.sa platform

Preparing the chatbot script

Building the Scenario Flow Map – API Unit

The API module allows the chatbot to interact with external systems, such as Customer Relationship Management (CRM) or Enterprise Resource Planning (ERP) systems, by sending or receiving data through application programming interfaces (APIs)

API Call Structure:

- **Request URL:**

Define the endpoint the bot will call. Supports GET and POST methods.

- **Header & Data:**

Include required tokens, credentials, or payloads for the request.

- **Response Handling:**

- **Key:** The name of the variable where the returned value will be stored.
- **Value:** The actual data received from the client's system.

"This module enables the implementation of a dynamic integration capable of:

- Retrieving live data such as user status or account details
- Sending the data collected from the user in real time to external systems"

Example: Sending Data to an External System (POST Request)

To better understand how the API Call node works, here is a real-world example of a POST request. It sends user feedback from the chatbot to an external system, using the same format shown in tools like Postman or within the API node.

Preparing the chatbot script

Building the Scenario Flow Map – API Unit

Request Details

- **Method**

POST

- **Endpoint**

https://api.example.com/feedback

- **Headers**

- access-token: your_API_key
- mobile: 966508292156
- Content-Type: application/json

- **Payload**

```
{
  "number": "12021626",
  "feedback": [
    {
      "category_id": "1",
      "category_value": "1",
      "category_note": "I was happy"
    }
  ]
}
```

- **Response**

```
{
  "jsonrpc": "2.0",
  "id": null,
  "result": {
    "status": "OK"
  }
}
```

The figure above illustrates an example of sending structured data (such as notes or orders) to an external system easily, enhancing the chatbot’s ability to interact with other enterprise systems

Chatbot Editor Tools

Filter Option (Variable)

Within the units **(User Input, Bot Reply, Go To)**, there is a Filter option  used to define conditions for accepting or rejecting input based on specific logic. It includes a set of fields: Variable, Type, Value, and Logical Operators

First: Variable

The variable dropdown includes the following variables:

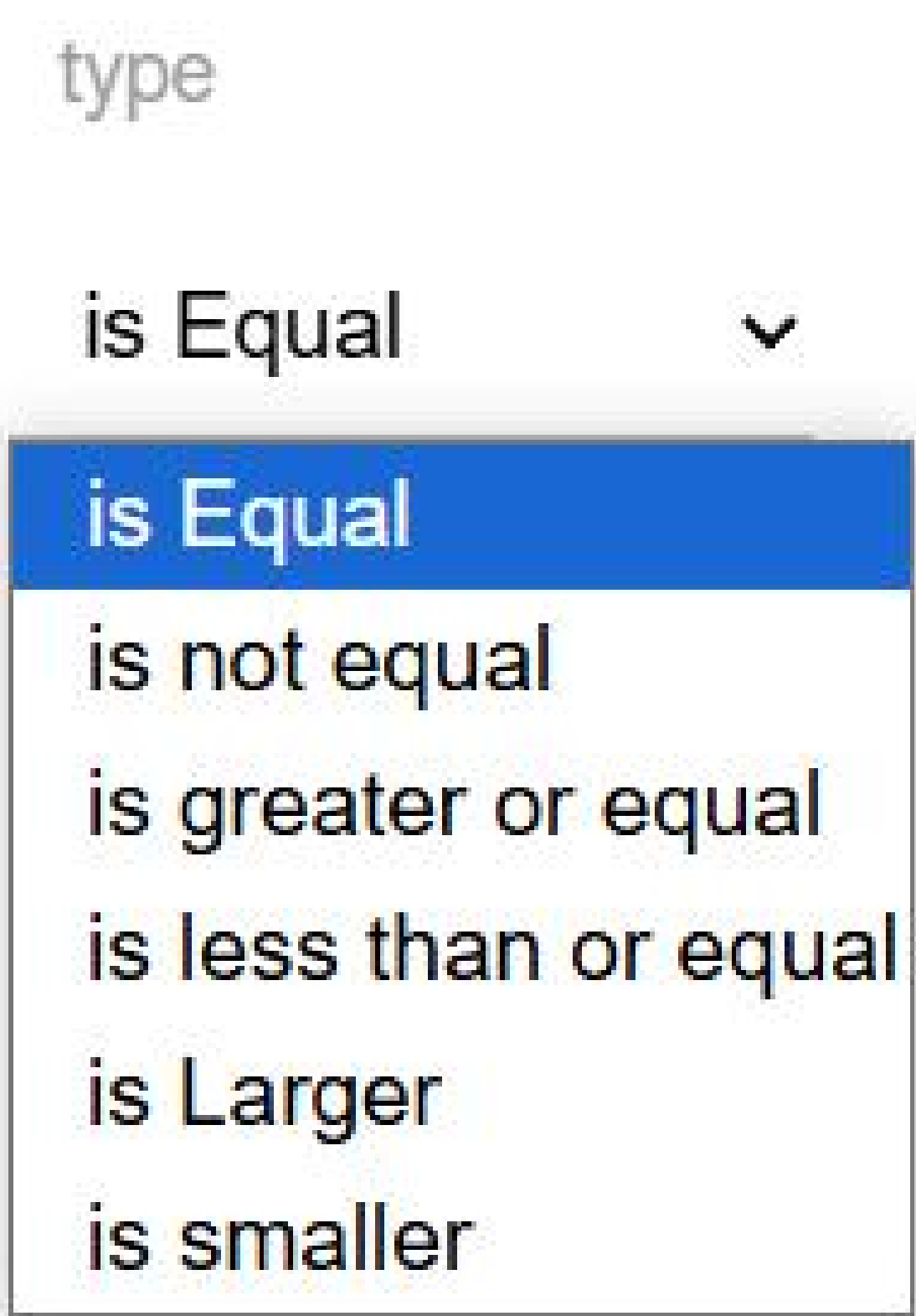
Variable Name	Description	Example Values
getChatStatus	Retrieves the status of the chat after it ends.	movetoticket, lostconnection, solved
getReplyType	Returns the type of the user's input.	video, audio, interactive, image, text
getName	Retrieves the name associated with the user's WhatsApp account.	
getMobile	Returns the user's phone number used in the conversation.	
getHour	Returns the current hour in 24-hour format when the user interacts with the bot.	0 to 23
getMin	Returns the current minute of the interaction time.	0 to 59
getDayName	Returns the name of the day the user contacted the bot.	Monday, Tuesday, ..., Sunday
getLastInput	Retrieves the last input or message sent by the user.	
getMediaId	Returns the ID of the last media file sent by the user.	
getReplyData	Returns the value (payload) of a button or quick reply selected by the user.	
getUserType	Identifies the source of the conversation.	web, whatsapp
getTime	Retrieves hour and minute	1507 -> 3:07

Chatbot Editor Tools

Filter Option (Condition Type)

Second: Condition Type

Conditions are used to determine whether the input or variable matches a specific value or falls within a defined range
The supported condition types are as follows:



Condition Type	Description
is Equal	Checks if the variable exactly matches the specified value.
is not equal	Checks if the variable does not match the specified value.
is greater or equal	Checks if the variable is greater than or exactly equal to the specified value.
is less than or equal	Checks if the variable is less than or exactly equal to the specified value.
is Larger	Checks if the variable is strictly greater than the specified value (excludes equal).
is smaller	Checks if the variable is strictly less than the specified value (excludes equal).

Chatbot Editor Tools

Filter Option (Logical Operators)

Third: Logical Operators

Logical operators are used when combining multiple conditions within the same filter to determine how the conditions should be evaluated together

Logical Operators

and ▼

and

or

Logical Operators

Operator	Description
and	All combined conditions must be true for the filter to pass.
or	At least one of the combined conditions must be true to pass.

Chatbot Editor Tools

Sidebar Support Buttons

The help **Sidebar Support Buttons** on the chatbot platform allow users to enhance performance and manage the interface easily while building the scenario



Conversation Test: Used to run the scenario and track its steps to verify the logical flow of the scenario and the user experience



Zoom In: Enlarges the elements displayed on the platform



Zoom Out: Reduces the size of the elements



Sync: Reloads the page elements and reorganizes their updated structure



Center Screen: Automatically arranges and moves the scenario elements to be neatly centered on the screen



Lock Screen: Restricts the ability to edit elements within the editor during testing or review, preventing unintended changes



Connect to WhatsApp: Activates the desired scenario using the client's verified WhatsApp account

Chatbot Editor Tools

Keywords and Reference Functional Commands

This section provides a comprehensive reference for keywords, placeholders, and internal commands used within the chatbot platform. These elements are typically utilized within bot responses, API calls, or as unit headers to perform specific technical functions such as ending a conversation, saving user data, or linking reporting tracking points

Note: These keywords are not visible to the end user, but they play a crucial role in scenario tracking, data logging, and controlling the platform’s behavior

Keyword	Usage / Function	Additional Notes
<code>{{survey1}}</code> إلى <code>{{survey10}}</code>	This keyword is added inside the Bot Response node and is not visible to end users. It is used to track how many times users reach this specific point in the scenario, and the data appears in the daily Live Chat reports. The format uses <code>{{surveyN}}</code> , where N is a number from 1 to 10 (e.g., <code>{{survey1}}</code>).	Can later be utilized in the daily Live Chat report available through the Taqnyat portal.
<code>{{endChat}}</code>	Used to automatically end the conversation through a Bot Response.	Alternative phrases like end conversation or انهاء المحادثة can also be used. These are written directly in the message field in the bit response node (instead of the placeholder that says “enter your message”).
<code>{{apiResult}}</code>	Used to display the result of an API query. It is intended for testing purposes during the chatbot’s API integration phase.	Can be added in the bot response node next to the message

Chatbot Editor Tools

Keywords and Reference Functional Commands

Keyword	Usage / Function	Additional Notes
<code>{{live agent}}</code>	Used to transfer the conversation from the chatbot to a live human agent.	Alternatives include: "talk to live agent" and "خدمة العملاء". These can be written next to the message text inside the Bot Response node.
<code>{{getToday}}</code> , <code>{{getTomorrow}}</code> , <code>{{getAfterDay1}}</code> , ...	Used to retrieve the current date or dynamically generate future dates.	Supports up to <code>{{getAfterDay7}}</code> . These can be written next to the message text inside the Bot Response node.
<code>afterCloseChat</code>	It is used to indicate that the customer has exited the Live Chat and has been returned to the chatbot, in order to direct them to a specific scenario (e.g., a post-service survey or feedback flow).	A label added in the title of a Bot Response node

We're Here to Support You

At Taqnyat, we believe that building and managing effective conversational experiences through chatbots is a pivotal element in your customer communication strategy.

This guide has been prepared to provide you with full support at every stage of use ; from initial setup to optimization and development aiming to empower you to make the most of the platform with clarity

Taqnyat: Your Trusted Technology Partner on Your Digital Transformation Journey

**Our specialized team is always ready
to assist you whenever needed**



Technical Support Team

support@taqnyat.sa

966920015404