**Assignment 2**

**Azhan Saleem**

**11635219**

**PART I: Request GCP Free Credit Coupon and Redeem It**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**PART II: GCP: Dialogflow CX: Fundamental Concepts**

**Report: Key Concepts in GCP Dialogflow CX**

Dialogflow CX, a cloud-based conversational AI platform, provides a structured approach to designing and managing complex dialogue systems. This report focuses on eight fundamental concepts that play a crucial role in building AI-powered dialogue systems in Dialogflow CX: Conversation Session, Agent, Flow, Page, State, State Handler, Intent, and Entity.

1. **Conversation Session**

* **Concept:** A conversation session refers to the period during which an interaction occurs between the user and the agent. It is temporary, beginning when a user initiates communication and ending when the conversation is completed or timed out. Each session has a unique identifier.
* **Role in AI Dialogue System**: The conversation session is the backbone of the interaction. It helps the system maintain continuity by tracking the conversation’s context, storing session-specific data, and facilitating personalized responses. It allows the AI to "remember" what has occurred earlier in the conversation and respond accordingly.

2. **Agent**

* **Concept:** An agent in Dialogflow CX is the central component that simulates a virtual conversation partner. It is configured with multiple flows, intents, entities, and state handlers to carry out dynamic conversations.
* **Role in AI Dialogue System**: The agent is essentially the orchestrator of the dialogue system. It manages the various aspects of the conversation, such as detecting user inputs, understanding intents, executing flows, and providing responses. All flows and dialogues revolve around the agent, making it the core system component.

3. **Flow**

* **Concept:** A flow is a sequence of related conversational states and transitions, defining how the conversation moves forward in different scenarios. Each flow handles specific tasks, such as addressing a specific customer query or managing a service request.
* **Role in AI Dialogue System**: Flows enable the conversation to be broken into manageable segments. In complex systems, agents can contain multiple flows to manage different interactions (e.g., booking appointments, and handling complaints). By dividing conversations into flows, the system can maintain clarity and manage user journeys more effectively.

4. **Page**

* **Concept**: Pages represent specific points in a conversation within a flow. Each page captures a distinct stage of interaction, where the system awaits input from the user or takes some predefined action.
* **Role in AI Dialogue System**: Pages help structure the conversation by marking specific points where users are within a flow. When a user interacts with the system, they are guided from page to page based on their input or the intents recognized by the system. Pages ensure smooth transitions and provide a clear path for the dialogue.

5. **State**

* **Concept:** A state refers to the current status of the conversation. It reflects the agent’s understanding of where the user is within a particular flow or page at any given moment.
* **Role in AI Dialogue System**: States are critical for maintaining conversation logic. They help the system keep track of the user’s progress in the conversation, allowing the agent to manage transitions effectively. The state ensures that the system responds appropriately, depending on the current context and conversation history.

6. **State Handler**

* **Concept:** A state handler is a mechanism that determines how the system moves from one state (or page) to another. It is responsible for managing transitions based on user input or detected intents.
* **Role in AI Dialogue System**: The state handler brings interactivity to the system by controlling how the agent reacts to different inputs. For instance, if a user requests a booking, the state handler decides whether to move to the confirmation page or prompt for more details. This flexibility allows the system to adjust dynamically based on the conversation flow.

7. **Intent**

* **Concept:** An intent is a mapping of what the user wants to accomplish. It captures the purpose behind the user’s message, allowing the agent to understand and act upon it.
* **Role in AI Dialogue System**: Intents are the key to understanding the user's needs and driving the conversation forward. Once the system detects an intent, it triggers the appropriate response or flow transition. For instance, if a user asks to "schedule a meeting," the system recognizes the intent and initiates the booking process. Without intent, the system would not be able to interpret user requests meaningfully.

8**. Entity**

* **Concept:** Entities represent specific details within the user’s input, such as names, dates, numbers, or other critical pieces of information. They serve as variables that capture important data within a conversation.
* **Role in AI Dialogue System**: Entities help the system extract actionable data from the user’s message. For example, in a restaurant booking scenario, the entity might capture the number of people, the date, or the time for the reservation. Entities allow the system to be more intelligent and contextual, turning user input into information that the system can use to fulfill requests.

**Conclusion:**

These eight concepts form the core framework of an AI dialogue system created using Dialogflow CX. Together, they enable the design of dynamic and responsive conversational agents capable of handling complex user interactions. From managing the structure and flow of conversations to understanding user intents and capturing key data, each concept plays a pivotal role in delivering a seamless conversational experience. By leveraging these concepts effectively, developers can create powerful, AI-driven dialogue systems that meet diverse user needs.

**PART III: GCP: Dialogflow CX: Create a Virtual Agent**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Introduction**

Google Cloud Platform (GCP) provides Dialogflow CX, a powerful tool for building conversational AI systems. A **virtual agent** is the core element of any AI-based system on Dialogflow CX, which interacts with users and handles conversations based on predefined workflows. In this report, I will walk through the steps of enabling the Dialogflow CX API and creating a virtual agent using GCP, accompanied by screenshots that capture each critical step.

**Step 1: Access the GCP Console**

To begin, we access the Google Cloud Platform (GCP) Console and ensure that the correct project is selected. In this example, we used **"My First Project"** for enabling the Dialogflow CX API and creating the agent.

**Steps:**

1. Log in to your GCP account at https://console.cloud.google.com/.
2. Make sure the correct project is selected. You can confirm the project name on the top bar of the dashboard. For this task, the project name is **"My First Project"**.

**Step 2: Enable the Dialogflow CX API**

Next, we enable the Dialogflow CX API to ensure the virtual agent can be created.

**Steps:**

1. Navigate to the **APIs & Services** section in the GCP Console.
2. Click on **Library** to search for APIs.
3. In the search bar, type **Dialogflow CX API**.
4. Click on the **Dialogflow CX API** from the results.
5. Press the **Enable** button to activate it.

Once enabled, the API will be active and ready for use with your project.

**Step 3: Create a Virtual Agent**

After the API is enabled, we proceed to create a virtual agent.

**Steps:**

1. In the GCP Console, navigate to **Dialogflow CX Console**.
2. Click on **Create Agent**.
3. Fill in the following details:
   * **Agent Name**: (e.g., “Agent”).
   * **Location**: Choose **us-central1 (Iowa)** for optimal latency based on location.
   * **Default Language**: Select **English**.
   * **Time Zone**: Choose **America/Chicago** for Dallas.
4. Verify the **GCP Project** is selected (in this case, "My First Project").
5. Click **Create**.

Once the agent is created, you’ll see a confirmation message at the bottom of the screen indicating that the agent was created successfully.

**Screenshot:**

The screenshot above shows the successful creation of the virtual agent named "Agent".

**Step 4: Verify Agent Creation**

After the agent creation, you will be redirected to the agent’s dashboard, where you can view the agent’s **flows** and **pages** for further configuration.

**Verification Screenshot:**

In this dashboard, we can begin building out the conversational flows for the virtual agent.

**Conclusion**

This report outlined the steps for enabling the Dialogflow CX API and creating a virtual agent within GCP. The process included accessing the GCP Console, enabling the required API, creating an agent, and verifying the creation through the agent dashboard. The virtual agent is now ready for further development, such as defining flows, intents, and pages for user interaction.

**PART IV: GCP: Dialogflow CX: A Simple Conversation**

**Question 3.1:**

(a) How many turns are there in the above conversation?

The total number of turns in the conversation refers to each instance where either Customer C or Staff A speaks.

1. A: Good morning. How can I help you?
2. C: Good morning. I want to buy a shirt. What color do you have?
3. A: It’s green.
4. C: Great! How much is it?
5. A: $30.00
6. C: What size?
7. A: Medium. Is it OK for you?
8. C: Yes. I like it. I’ll get to the store and buy one.
9. A: We’re open until 7:00 PM tonight.
10. C: Thanks. Bye.
11. A: Thank you very much. Bye.

**Thus, there are 11 turns in this conversation.**

**(b) Which turns should be included in the “Default Start Flow”?**

In Dialogflow CX, the Default Start Flow typically handles the initial user interaction and provides the foundation for guiding the conversation. The turns included in the Default Start Flow should be those that start the conversation and gather the necessary information from the user.

For this simple scenario, the following turns should be included in the Default Start Flow:

1. A: Good morning. How can I help you?
   * This is the agent’s initial greeting.
2. C: Good morning. I want to buy a shirt. What color do you have?
   * This is the customer initiating the query and seeking information about the product.
3. A: It’s green.
   * This is the response to the customer’s first question.
4. C: Great! How much is it?
   * The customer continues the query by asking about the price.
5. A: $30.00
   * The agent provides the price.

**These turns form the core of the conversation.**

**3.2 Report on the Conversation Flow Design**

**1. Agent Creation**

* The agent "Apparel-X" was created to handle customer inquiries about the store’s medium-size green shirts.
* The default language was set to English, and the time zone was configured to match the store’s operational hours.

**2. Default Start Flow**

* The Default Start Flow includes the initial greeting and product inquiry, which captures the first few turns of the conversation.
* These interactions are critical as they lay the foundation for guiding the customer through the conversation.

**3. Intent Creation**

* Intents were created to capture specific user requests: greeting, product inquiry, price inquiry, size inquiry, and confirmation.
* Each intent was matched to appropriate responses relevant to Apparel-X's products and services.

**4. Pages**

* Pages were set up to handle each part of the dialogue. As the conversation progresses, the customer is guided through different pages such as product inquiry, price inquiry, and confirmation.
* The transitions between these pages ensure a smooth conversation flow without the agent asking unnecessary questions.

**5. Testing**

* After designing the flow, the built-in simulator tested the conversation.
* The conversation worked as expected, with the agent responding accurately to customer queries and providing relevant information at each step.

**A screenshot of a computer

Description automatically generated**

**Question 3.3: Based on the Conversation Design**

**(a) What conversation session does the design visualize?**

The conversation session visualizes a **purchase inquiry interaction** between a customer (Customer C) and a staff member (Staff A) at Apparel-X. The session handles a straightforward customer inquiry about a product (medium-size green shirts), including the shirt’s color, price, size, and store hours.

**(b) How many pages? What does each page refer to?**

There are **5 pages** in the conversation flow:

1. **Greeting Page**:
   * Refers to the initial interaction where the agent greets the customer and asks how they can assist.
2. **Product Inquiry Page**:
   * Refers to the part of the conversation where the customer asks about the available product, and the agent provides information on the product's color (green).
3. **Price Inquiry Page**:
   * Refers to the section where the customer asks about the price, and the agent responds with the cost ($30.00).
4. **Size Inquiry Page**:
   * Refers to the inquiry where the customer asks about the size, and the agent confirms the product's size (medium).
5. **Confirmation Page**:
   * Refers to the final part of the conversation, where the customer confirms they are satisfied with the product, and the agent provides store hours before closing the conversation.

**(c) How many states? What does each state refer to?**

There are **5 states**, each corresponding to a specific stage of the conversation flow:

1. **Greeting State**:
   * The initial state where the agent greets the customer and waits for their query.
2. **Product Inquiry State**:
   * The state where the agent provides information on the product's color.
3. **Price Inquiry State**:
   * The state where the agent provides the price of the shirt.
4. **Size Inquiry State**:
   * The state where the agent confirms the shirt's size.
5. **Confirmation State**:
   * The final state where the agent closes the conversation by confirming the customer’s satisfaction and providing store hours.

**(d) Are there any state handlers? If YES, what are they?**

Yes, there are **state handlers** in the conversation flow. These are transitions between states based on customer inputs:

1. **Greeting State Handler**:
   * This handler transitions from the greeting to the product inquiry based on the customer's response (e.g., asking about the product).
2. **Product Inquiry State Handler**:
   * This handler transitions to the price inquiry based on the customer’s question about the price.
3. **Price Inquiry State Handler**:
   * This handler moves to the size inquiry after the customer asks about the product’s size.
4. **Size Inquiry State Handler**:
   * This handler moves to the confirmation after the customer agrees that the size is suitable.
5. **Confirmation State Handler**:
   * This handler closes the session after the customer expresses satisfaction and is informed about the store hours.

**(e) Are there intents? What are they?**

Yes, there are **5 intents**:

1. **Greeting Intent**:
   * Captures customer greetings like "Good morning," "Hi," or "Hello."
2. **Product Inquiry Intent**:
   * Captures queries like "I want to buy a shirt," "What color do you have?"
3. **Price Inquiry Intent**:
   * Captures questions like "How much is it?" or "What's the price?"
4. **Size Inquiry Intent**:
   * Captures inquiries like "What size?" or "Is it medium?"
5. **Confirmation Intent**:
   * Captures confirmations like "Yes," "I like it," or "That’s fine."

**(f) Are there entities? What are they?**

In this conversation, there are **no custom entities** defined since the conversation is very straightforward. However, if we wanted to introduce entities to make the conversation more dynamic in other scenarios, we could potentially use the following:

1. **Product Entity**:
   * If the store sold more than one product, a "shirt" could be captured as an entity for product type.
2. **Size Entity**:
   * If various sizes were available, we could capture the size ("medium") as an entity.