**Flow Diagram Description**

**Scenario 1: No Integration.**

**Input: User Query**

**Output: Response from Dialogflow.**

**Sample Flow: Chat Window 🡪 JavaScript (User) 🡪 Flask (User) 🡪 Dialogflow API 🡪 Speech 🡪 Flask 🡪 JavaScript 🡪 Chat Window**

**Description:** Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent and obtain the respective response or speech. The Dialogflow API sends the speech to flask which then sends it to the JavaScript which appends the speech to the chat window.

**Scenario 2: Integration Required.**

**Input: User Query**

**Output: Response from Python.**

**Sample Flow: Chat Window 🡪 JavaScript (User) 🡪 Flask (User) 🡪 Dialogflow API 🡪 Parameters 🡪 Read/Write CSV 🡪 Speech 🡪 Flask 🡪 JavaScript 🡪 Chat Window**

**Description:** Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent once the intent is obtained the Dialogflow sends the parameters to python using webhook which then either performs a task such as create a claim and return the claim number or looks up the user data from the csv file by performing read/write operations and returns the appropriate output. The returned output is saved in speech and the python sends the speech to flask which then sends it to the JavaScript which appends the speech to the chat window.

**Scenario 3: Answering Unknown Question.**

**Input: User Query Unknown to Bot**

**Output: Response from Human Agent.**

**Sample Flow:**

**1 Chat Window 🡪 JavaScript (User) 🡪 Flask (User) 🡪 Dialogflow API 🡪 Unknown Question not answered 🡪 memcached 🡪 Flask(Agent) 🡪 JavaScript(Agent) 🡪 Agent Input 🡪 Memcached 🡪 Flask(Agent) 🡪 JavaScript(User) 🡪 Chat Window**

**2 Memcached 🡪 CSV(Write)**

**Description:** Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent once the intent cant be determined it goes to the fallback stage the Dialogflow uses webhook sends the parameters to python. The python initiates the memcached hits the flask of the human agent which then hits the REST API of human agent which then goes to the JavaScript of the human agent. When the human agent provides with the response which is then sent to the flask of human agent which then hits memcached which then sends the speech or response to the JavaScript of the user which then appends the speech to the user chat window.

**Scenario 4: Answering Unknown Question already answered.**

**Input: User Query Unknown to Bot**

**Output: Response from Python.**

**Sample Flow:**

**1 Chat Window 🡪 JavaScript (User) 🡪 Flask (User) 🡪 Dialogflow API 🡪 Unknown Question already answered 🡪 Read CSV 🡪 Flask 🡪 JavaScript(User) 🡪 Chat Window**

**Description:** Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent once the intent can’t be determined it goes to the fallback stage the Dialogflow uses webhook sends the parameters to python. Python reads the CSV file to check if the question has already been answered. Then python finds the answer to the requested question in the CSV file. The answer is returned and saved inside speech. The speech is then sent to flask which then sends it to the JavaScript which appends the speech to the user chat window.