MILESTONE2

In Milestone2 , We made great strides in improving user inputs, introducing outfit recommendations based on weather forecasts, and revamping the weather API class and data models of our chatbot project. Our main goals were to lay a strong framework for user interactions and offer insightful advice for organizing travel.

1. User Input:
   Predefined Journeys: Following deliberation, we made the decision to manage the user's journey at first to guarantee a strong base for the chatbot. There were two primary predetermined routes established:
   -Verify the Weather in a Place
   -Plan a trip, taking into account the weather.

   User Input Validation: Using the UserInputValidator class, a strong user input validation method was implemented. This guarantees the accuracy of data submitted by users before it is processed.

   Improved Trip Destination Validation: The validateTripDestination method was extended to fully validate trip destination information, such as the inputs for the city, country, and date.

   GeoLocation Validation: Added geolocation validation to the UserInputValidator to guarantee location-based queries are accurate.

2. Suggestions for Clothing:

   Logic Development: Using retrieved weather forecasts as a basis, we created logic to offer precise outfit recommendations. Important prerequisites comprised:
   ideas for attire based on the weather prediction
   Basic flowcharts, activity diagrams, and class diagrams are made to direct the application of logic.

   Important Weather Variables:
   We were able to derive broad weather forecasts in a legible way by utilizing the weather codes that Open-Meteo made available.
   Important meteorological factors taken into account:

   The apparent and atmospheric temperatures
   Probability of precipitation and total amount
   Wind gusts and speed
   Anticipated weather code

   Important Issues Solved:
   How to read weather codes so that we can draw conclusions.
   using data from weather forecasts to make customized outfit suggestions.

3. Weather API Class and Data Models Redesigned :

   Effective Location Fetching: Developed and put into practice an effective way to retrieve several locations from OpenMeteo's Geo-coding service. This enhancement makes sure that weather forecasts for different places are retrieved more quickly, which improves the chatbot's general responsiveness.

   Models for the Effective Storage of Forecast Data: These models were developed through iteration and design. We enhance accessibility and data management by structuring forecast data, making it easier to integrate with the chatbot's features.

   Class Diagram Refinement: Detailed class diagrams were made for each class that was utilized in the data models and weather API. The architecture of the system is clearly visualized by these diagrams, which makes it easier to comprehend and maintain the codebase.

   Implementation: To seamlessly integrate the new features, we updated the weather API class and data models. This involved streamlining the deployment procedure to guarantee smooth interaction with our chatbot technology.

In summary, during Milestone 2, notable progress was achieved in enhancing the functionality and dependability of our chatbot project. By concentrating on three main areas—user inputs, weather-forecast-based clothing recommendations, and the redesign of the weather API class and data models—we've laid the groundwork for a more responsive and user-friendly chatbot experience. We have improved the overall performance and accuracy of our chatbot by thoroughly validating user inputs, implementing effective location fetching, and honing prediction data storage models.