Holiday Chat Agent Design Document

The aim of the Holiday Chat Agent (further referred to as HCA) project is to deliver a chat agent that will recommend holidays based on user input. The specific requirements for the project are:

**General**

* Ask the user a minimum of 2 questions to identify possible destinations
* Provide user feedback to questions that the Chat agent cannot respond to
* Provide a recommended set of holidays to the user based on their responses

**Data**

* Data is to be loaded from a local file into application
* Responses to questions need to be stored temporarily to provide recommendations

**User Interface**

* Simple GUI interface
* GUI should show a trail of chat conversation for the session

I am planning to build this project in a NodeJS / Javascript environment, as I feel that a web app would work perfect for this project and Javascript is currently my preferred environment in general. I am planning to use a RESTful API for the backend, which will handle all the logic for the questions and will communicate to the frontend with HTTP requests. I feel like this is perfect for scaling up if necessary, as more questions can easily be added without a lot of work on the frontend.

The backend will use a MongoDB database to manage the data, and Express to run our API.

The frontend will use VueJS, as this is perfect for HTTP request based applications, and can be easily exported to iOS / Android apps if necessary. In VueJS as well, we have the ability to turn the front-end into a “Component” and then it can be easily added to other VueJS based applications, only needing the URL of the backend server.

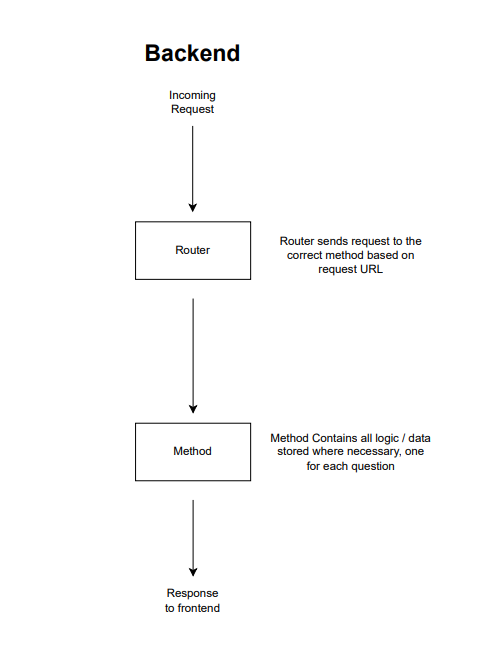
The backend of the application will host a server and via HTTP requests will serve / collect information from / to the user, and store it temporarily for use in logic and chat logs.

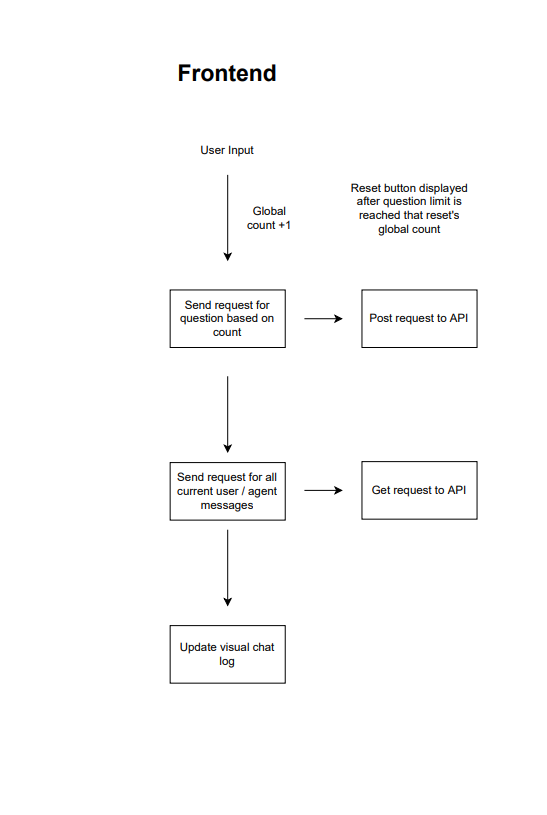
The frontend will have a little logic based on which question to serve, but apart from that will just send HTTP requests and display the information in a nicely styled, user experience oriented way.

General

The agent will start by prompting the user to enter a keyword. When the keyword has been successfully entered (the agent will notify if it is incorrect), the agent will then ask questions, displaying and filtering the results based on responses. When the max limit of questions has been reached the agent will thank the user for using the agent, and display a reset button to start the application again.

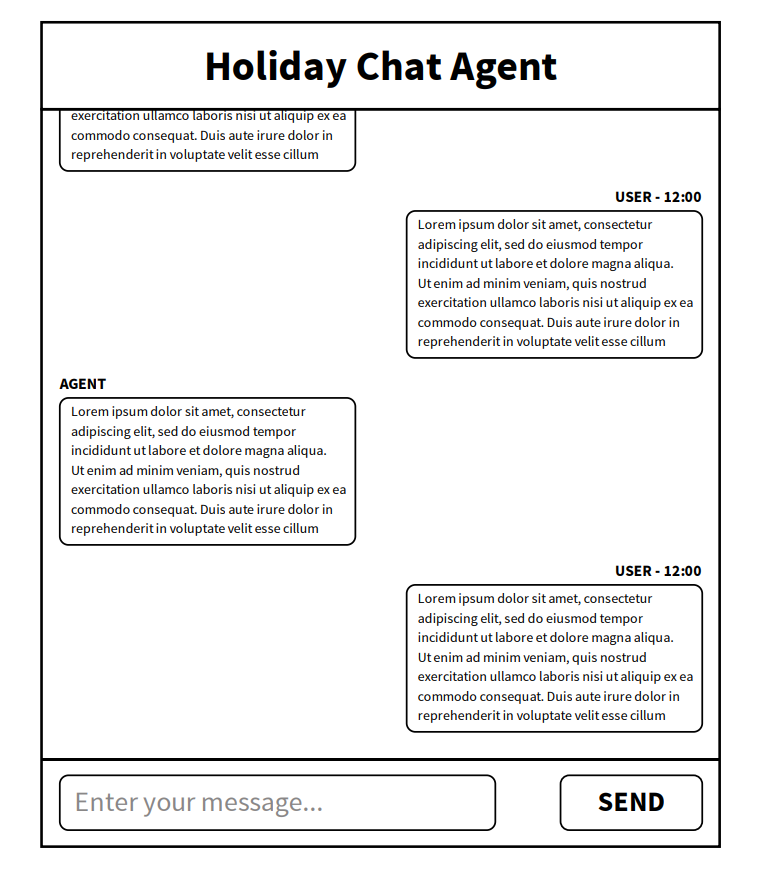
A flowchart of the general process for a request is below:





User Interface

I have wireframes and designed a user-friendly chat interface that should be perfect for this application:



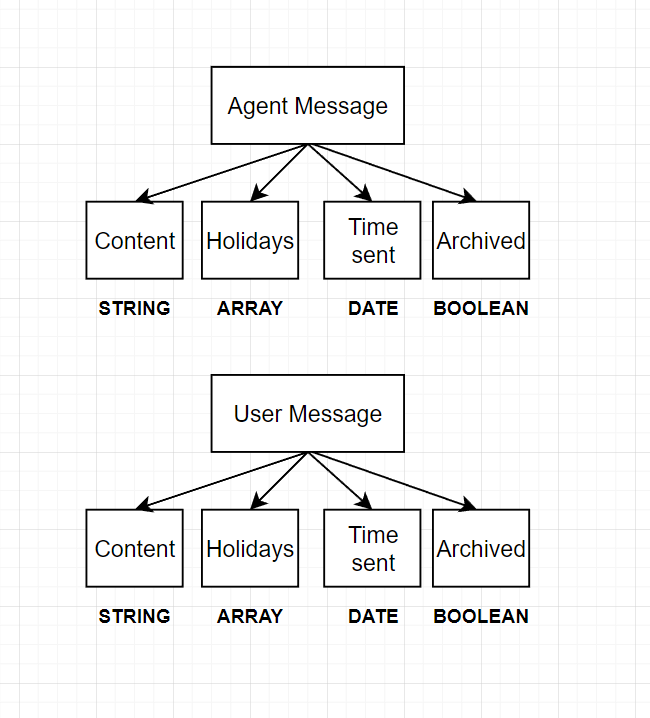
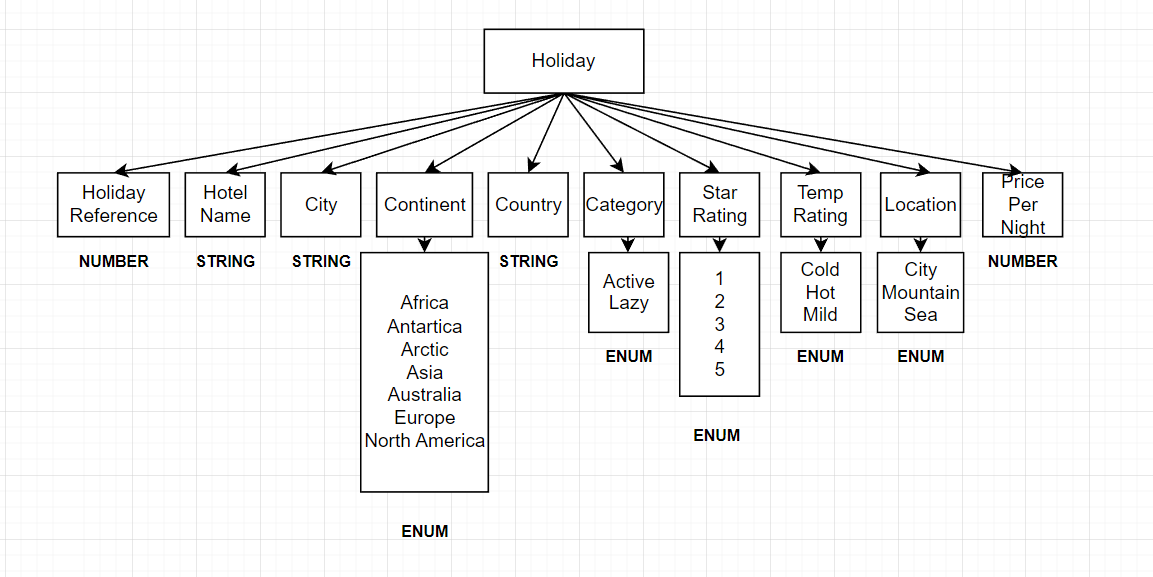


Expected issues

The only real issue I can see being presented is the rendering of the chat messages on the frontend. Maybe I need an algorithm or something to correctly space the messages between user and agent.

Data Models

Below are the data models of how I plan to structure the database for this application.



Functional / Non-functional requirements

The functional requirements for this project are all the requirements listed further up:

**General**

* Ask the user a minimum of 2 questions to identify possible destinations
* Provide user feedback to questions that the Chat agent cannot respond to
* Provide a recommended set of holidays to the user based on their responses

**Data**

* Data is to be loaded from a local file into application
* Responses to questions need to be stored temporarily to provide recommendations

**User Interface**

* Simple GUI interface
* GUI should show a trail of chat conversation for the session
* Whereas the non-functional requirements haven’t been specified. The difference between functional and non-functional requirements is that functional requirements define what an application must do, the non-functional requirements define requirements for the properties of the system that the application is running upon.  
    
  For the project, I may recommend such non-functional requirements as:
* Ensuring that the chat agent can serve multiple users at once
* Ensuring that the webpage for the chat agent loads fast
* Ensuring that the system is reliable for hosting this type of application
* Make the application easily scalable (easy to add more questions etc)