SoulTrainRiders: Kevin Moore, Sean Schmidt, Zachery Angelucci, Mayowa Oloyede

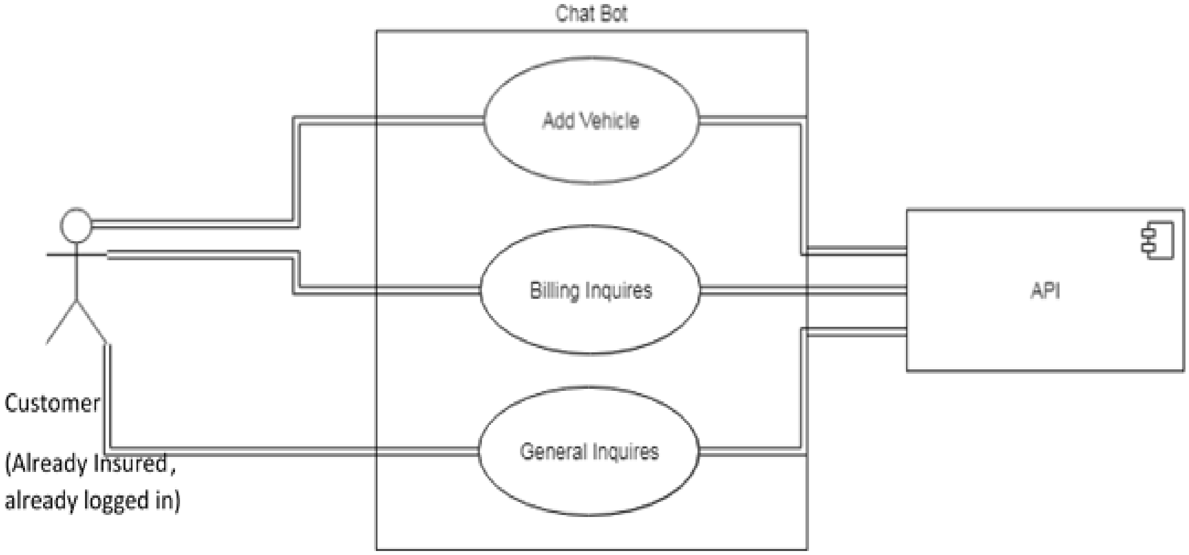
10/4/2018

System Requirements Document

Soul Train Riders AI Customer Service Bot

General Overview:

The AI Customer Service Chatbot is an automated conversational agent, or program, that recognizes questions known as “intents” and produces a response. When launched, the chatbot will display as a user interface with a window containing the current conversation and a text box for typing in customer input. When the bot receives text-based input, the language processing will relate the input to one of several programmed intents and begin a conversation (via response text) with the user to retrieve all necessary information to produce some form of desired result. This bot should accept questions of natural language regarding adding vehicles to an insurance policy, billing and payments, or a select list of generic questions, and pass this information to backend services in order produce either the success of the action or grant the user the requested information. If at any point during a conversation the customer input does not match a known intent, the bot will return a default response that warns the customer of their error before continuing the conversation. If the bot requires more information to complete an action, it should be able to recognize the missing info and request it from the customer. The bot should not ever be unable to return an appropriate response, and always have a default answer at the ready. The bot should not be able to attempt to pass information to the backend API while all required information is still not present. The bot should ultimately be able to respond to natural human language and should not require exact wording to reach an intent.



--Diagram Explanation--

In this use case diagram, we show an end user (a customer) using our system. The three circles within our chatbot system would be our three user epics: Adding a vehicle, billing inquiries, and general inquiries. Each system works accordingly: The end user communicates with the chatbot for a desired purpose, being one of the three epics. Once the chatbot has detected which intent the user is going for, the chatbot will communicate with the backend API to see if all the appropriate information has been collected. If so, the API communicates with the chatbot saying that the user's request has been fulfilled. From there the chatbot sends a confirmation message to the user and the use of the system is complete. On the other hand, if not all the information has been collected for fulfillment of the request, then the API will communicate with the chatbot asking for more information. Which prompts the chatbot to ask for the missing communication. From here, we loop back to the beginning and continue the cycle (user -> chatbot -> API -> chatbot -> user) until all information has been gathered.

User Stories:

(Note: User stories are generic, to be broken down afterwards)

1. Add a vehicle

Customer -> Add a Vehicle -> API assists -> Information is then given out until completion

"As a customer of The Hartford I want to be able to ask a question on how I can add a new vehicle on my policy so that I can instantly accomplish my task efficiently and effortlessly without much human intervention."

Pre-conditions:

- The questions are asked through natural language as if it's a normal conversion
- The question must be able to match the intent from the chatbot
- The user must be logged in and already a policy holder

Post-conditions:

- The chatbot will then provide the information that was requested based on the question
- If there's not enough information given beforehand for the API to read, the bot will acknowledge it and ask the consumer to provide more information

2. Billing Inquiries

Customer -> billing inquiry -> API assists -> information given/drives conversation until actionable information is given.

“As a customer, I want to be able to ask questions about billing so that I may know the status of my billing.”

Note: We expect this epic to encompass 3-4 Particular questions, to be presented at a later date

Pre-conditions:

- Question must be on the topic of billing in some way
- The customer must provide enough information about their inquiry so that the API may perform the necessary actions

Post-conditions:

- The bot will return the proper information about the customer’s billing and confirm that is the end of the requests
- If not enough information is given to the chatbot in that the API cannot perform any actions, the bot will ask for more specifics of the customer’s request

3. General Inquiries

Customer -> “Natural Language” question -> API assists -> follow-up generated

“As a customer, I want to be able to ask questions in a natural language or format like “Am I covered if...” so that I can obtain information about my concerns as quickly as possible without relying on another person or my own searches.”

Note: We expect this epic to encompass 3-4 Particular questions, to be presented at a later date

Pre-conditions:

- Question must be asked in a natural language (I.e. non-specific wording like a typical conversation).
- Question must match a covered intent.
- If an intent requires more information, when prompted the user must provide the bot with required information.

Post-conditions:

- The bot will return information in a natural language format as requested by the original question.
- If any additional content would help the customer, such as a link to another page, this content will be included in the response.
- If more information was required to produce an appropriate response, the bot will recognize the lack of information and prompt the customer for additional information.
- Based on the intent that was responded to, the bot will be able to determine if any predictions of the customers’ actions can be made and, if so, produce a second response providing the customer with a chance to agree to a follow-up action. Example: Customer asks: “Driving to Canada for a week, am I covered?” and bot responds: “Yes, your coverage extends to Canada and Mexico as well. Will you be needing a Canadian auto id card?”

Complexity:

User story #1 is straightforward, with the only necessary breakdown being the extensive querying of the customer for all the necessary variables. User story 3 and 6 both require a more thorough breakdown, as they are simply generic instances that could potentially encompass any number of desires. For these intents the bot will be able to cover 3 to 4 possible questions for each intent, creating several new user stories. These breakdowns are to be developed currently and presented later.

Developer Stories:

1. As a developer, I want to be able to leverage natural language processing, machine learning, and other AI techniques so that I can provide maximum automated functionality to my system and provide the best service to a potential customer. This requires learning how to best utilize AI features found in DialogFlow.
2. As a developer I want to be able to access and extract entities from a DialogFlow chatbot response so that the system can automate tasks for the customer. This requires understanding the process of queries and responses to/from chatbots and how to leverage that return data.
3. As a developer I want to be able to mimic The Hartford's backend API with my own functions so that the system can verify that user information is valid but does not touch upon actual components of The Hartford's architecture. This is to avoid any concerns, legal or otherwise, that occur when dealing with private systems and potentially sensitive information. This requires knowledge in the development and implementation of APIs.

Nonfunctional Requirements:

1. The system shall be broken down into three parts: chatbot, website, & API
2. The system shall be written in JS for the chatbot, JSON for the API, and HTML for the website
3. The website shall be plain in design with the only thing of note being a tab for the chatbot
4. The chatbot shall be embedded into the website code
5. The system shall be clear in how it replies, and responds with proper English, set font, and sentence structure
6. The system shall be responsive within a given time frame

Glossary:

Chatbot- The application used to interact with customers and provide information customers requested. A chatbot is an embedded program with a window to view chats, a textbox to type a query, and a submit button to send a query.

Intent- Words that triggers the chatbot to return information. Essentially, a customers' potential question that the chatbot can answer.

Customer- Owns an account with The Hartford is worried about their insurance policy

DialogFlow- Developer tool for human-compute interaction used to form the chatbot

AI- (Artificial intelligence) How the chatbot gains its intelligence and learns when adding new information. Involves self-teaching programs and automated services provided for free by Google and affiliates.

NLP- (Natural language processing)- Accepting and understanding text as standard human language. What allows the chatbot to interact with a as customer as if it's an actual person.

API- (Application Programming Interface)- Used to display the interaction of the chatbot. Will accept information processed by the chatbot and mimic backend features

Integration- The connection between the chatbot and API to have full functionality

Validate- Used to see if the chatbot can send out the right response for the question requested. In practice, testing to see if customer information is proper.