Internship Plan

Documentation

Project Title: Conversationalizing the MSK Engage survey experience

In One Sentence: As part of a movement to forge a reliable, all-in-one patient chatbot assistant, this project integrates MSK Engage with an IBM Watson Assistant to dynamically import Engage Q&A text and patient information, for a proof of concept that a mobile chatbot may offer patients a more intuitive, engaging, and accessible survey experience.

Consumer Need: Patient burnout is real and patients do not like navigating multiple portals, many layers of authentication, tedious alerts, or repetitive tasks. Patients would prefer a mobile, on-the-go version of Engage which sends notifications at convenient times directly on their phone interface, is text-able like a real-life friend, and can efficiently help them navigate the numerous forms they need to fill out.

Business Value: Patient engagement is so important for accurate data collection and prevention of post-treatment complications. With a new focus on remote, patient-reported data due to COVID-19, quick survey responses are more important than ever.

Context: See main Confluence page for full background.

> *Summary: MSK is interested in pursuing IBM Watson as a vendor for conversational interfaces. One use case for chatbots is creating an accessible touchpoint for MSK Engage in the form of a mobile survey conversational agent so users can complete required forms on-the-go with an assistant that will authenticate them, direct them to their list of to-do surveys, and collect their responses to the existing Engage database.*

Hypothesis: A mobile survey conversational experience will save patients time on self-reporting and help MSK achieve greater engagement with required surveys.

Short Term Goals: (end of the internship)

- Engineer an IBM Watson Assistant skill that: *(starred are incomplete)*
    - Retrieves question text and answer text/type from the Engage database using Zach Rachlin's Engage survey Q&A text API
    - Uses Q&A text to programmatically create dialog nodes, entities, and slots on the Watson interface
    - *Uses Watson API endpoints to push Q&A dialog nodes while user is interacting with bot
    - *Dumps user response to a dummy SQL database with existing Engage SQL structure
    - *Pulls a list of patient to-do surveys from the Engage SQL database using PHI, which involves connecting to patient info API's from Bing Zhang and Peter Roehrich's team
    - *Connects to a blank HTML front-end which customizes the visual representation of certain types of questions (e.g: buttons, dropdowns, sliders, etc)
    - *If there is time, we can consider multilingual support as Zach Rachlin's API also includes Spanish & Russian translations of Q&A text
- Prototype the full front-end patient experience
    - Decide where such a chatbot would "live"
        - How much user authentication needs to happen within the chatbot vs. already completed in the environment?
        - What kind of mobile format? MSK in-house chat interface?
    - Design notification and alert schedule for users, as well as options menu

Long Term Goals

- This project lives within the greater scope of a reliable, all-in-one patient chatbot assistant with the following features:
    - **Appointment**: CI schedules and answers questions
    - **Forms**: CI conversationalizes the survey experience
    - **Treatment**: CI sends follow-up reminders & prescription pickup
    - **Payment**: CI conducts payment and provides insurance breakdown for total payment transparency
    - **Queries**: CI pulls information from knowledge bank and only contacts physician if necessary
    - **Notifications**: CI provides medication adherence reminders
    - Room for multilingual or voice-activated version!

Risks & Defining Success:

- A successful project outcome includes:
    - A **functional** chatbot that does not confuse the user or become confused by the user
        - The chatbot would not digress due to bad user input, nor would it make the user feel like they have lost control of Engage
        - The chatbot must fail gracefully
    - A **secure** chatbot that can reliably retrieve sensitive information via API, even if we are pulling from "dummy" patients
        - The chatbot should not allow users access to forms they were not assigned
    - A **scalabale** chatbot that involves minimal hard-coding and has documentation for importing future surveys and dealing with survey updates

Design internship projects & deliverables June 8 - Aug 15 2020 (10 weeks)

Week 1, 2: Setup (6/08, 6/15)

- Goal: Access & Context setting
- Outcome: Project Plan
- Resources
    - MSK & Industry
    - Processes & Frameworks
    - Healthcare & Technology
    - Product & Design Development

Week 3, 4: Discovery (6/22, 6/29)

- Goal: Research
- Outcome:
    - Research types: secondary (market, papers, analogous), primary (users, SMEs)
    - Research Findings Lunch & Learn Presentation (1hr)

Week 5, 6: Define & Develop I (7/06, 7/13)

- Goal: Define & develop prototype based on research
- Outcome:
    - Define hypothesis, customer need, business value, short term & long term vision, risks, success
    - Build & demo prototype WIP

Week 7, 8: Define & Develop II (7/20, 7/27)

- Goal: Iterate and refine prototype based on feedback
- Outcome:
    - Test & validate prototype: plan, execute, synthesize
    - Build & demo prototype (80% complete)

Week 9: Deliver (8/03)

- Goal: Refinements & final presentation
- Outcome:
    - Final presentation to team
    - Documentation

Week 10: Presentation & Wrap-up (8/10)

- Goal: Reflections & refinements
- Outcome:
    - Wrap-up
    - 8/14: last day